

HEURISTIC OPTIMIZATION METHODS FOR THE CHARACTERIZATION OF  
DYNAMIC MECHANICAL PROPERTIES OF COMPOSITE MATERIALS

Except where reference is made to the work of others, the work described in this dissertation is my own or was done in collaboration with my advisory committee. This dissertation does not include proprietary or classified information.

---

Klaus H. Hornig

Certificate of Approval:

---

Malcolm J. Crocker  
Distinguished University Professor  
Mechanical Engineering

---

George T. Flowers, Chair  
Professor  
Mechanical Engineering

---

Gerry V. Dozier  
Associate Professor  
Computer Science and Software  
Engineering

---

Subhash C. Sinha  
Professor  
Mechanical Engineering

---

Joe F. Pittman  
Interim Dean  
Graduate School

HEURISTIC OPTIMIZATION METHODS FOR THE CHARACTERIZATION OF  
DYNAMIC MECHANICAL PROPERTIES OF COMPOSITE MATERIALS

Klaus H. Hornig

A Dissertation  
Submitted to  
the Graduate Faculty of  
Auburn University  
in Partial Fulfillment of the  
Requirements for the  
Degree of  
Doctor of Philosophy

Auburn, Alabama  
May 10, 2007

HEURISTIC OPTIMIZATION METHODS FOR THE CHARACTERIZATION OF  
DYNAMIC MECHANICAL PROPERTIES OF COMPOSITE MATERIALS

Klaus H. Hornig

Permission is granted to Auburn University to make copies of this dissertation at its discretion, upon request of individuals or institutions and at their expense. The author reserves all publication rights.

---

Signature of Author

05/10/2007

---

Date of Graduation

## VITA

Klaus Harald Hornig Hollstein, son of Carlos E. Hornig and Inge A. Hollstein, was born on June 1, 1976, in Puerto Montt, Chile. He obtained his Bachelor of Science degree (Summa cum Laude) in Acoustical Engineering from Universidad Austral de Chile in July 2001. Working initially in acoustic room design, and projects in environmental noise control, he joined the Mechanical Engineering Department at Auburn University as a Research Assistant in August 2001, and became a Doctor of Philosophy candidate in November 2005. His scientific interests cover vibration analysis, nonlinear systems, optimization, and acoustics, in general. He is a member of the International Institute of Acoustics and Vibration (IIAV), and the American Society of Mechanical Engineers (ASME). He married Carolina D. Rojas on December 17, 2005.

DISSERTATION ABSTRACT

HEURISTIC OPTIMIZATION METHODS FOR THE CHARACTERIZATION OF  
DYNAMIC MECHANICAL PROPERTIES OF COMPOSITE MATERIALS

Klaus H. Hornig

Doctor of Philosophy, May 10, 2007  
(B.Sc. Universidad Austral de Chile, July, 2001)

138 Typed Pages

Directed by George T. Flowers

Generally speaking, of the fundamental dynamic mechanical properties - mass, damping, and stiffness, damping is usually the most difficult to quantify. This is perhaps particularly true for composite materials which tend to have substantially higher damping than comparable isotropic materials and therefore having an accurate representation is correspondingly more important. Accordingly, some heuristic optimization techniques for the identification of the dynamic characteristics of honeycomb-core sandwich composite materials have been suggested. More specifically, Genetic Algorithms (GA) and Particle Swarm Optimization (PSO) have been used in the present work as the optimization techniques for the system identification process. Experimental measurements of the dynamic responses (in the form of hysteresis loops) of simply-supported composite beam samples have been carried out, and a simplified semi-empirical mathematical model has

been developed for such a system, tailored from individual experimental observations of the dynamic behavior of the samples when they are excited at their mid-points by sinusoidal displacement waves. The hysteresis loops that were obtained are for several frequencies and excitation amplitudes around the first mode of vibration. The analytical model contains four unknown system parameters, which must be identified by both optimization techniques utilized. The performance of these optimization methods are compared with computer-generated and experimental hysteresis loops. In addition, the effect of noise contamination in the signals has been studied in order to assess the search accuracy of the optimization algorithms under such conditions.

## ACKNOWLEDGMENTS

I would like to thank my advisor Dr. George T. Flowers for his wise guidance, encouragement, and support in this research work, while I was pursuing my Doctoral studies in Mechanical Engineering at Auburn University. I also would like to thank Dr. Malcolm J. Crocker, Dr. Subhash C. Sinha, and Dr. Gerry V. Dozier, mainly for their directions in the areas of vibration analysis, and artificial intelligence (focused on optimization techniques), which have been very important for the fulfillment of my degree. I also thank Dr. Winfred A. Foster for his valuable comments and corrections, and for his great help for being my outside reader. I would like to give thanks in a very special way to my parents, family, and friends, that without their constant support this “journey” would have been very difficult to accomplish. And finally, I would like to express my greatest gratitude and gratefulness to my wife, Carolina Rojas, whose company, love, and perpetual support have been invaluable in the final steps of my degree.

Style manual or journal used: International Journal of Acoustics and Vibration

Computer software used: Microsoft Office 2003, Matlab R2006a, LabView 8.0,

Adobe Photoshop CS2

## TABLE OF CONTENTS

LIST OF FIGURES .....	xi
LIST OF TABLES .....	xv
CHAPTER 1. INTRODUCTION .....	1
1.1. Composite Materials and Damping .....	1
1.2. Scope of Study .....	5
CHAPTER 2. LITERATURE REVIEW .....	6
2.1. Hysteresis Models and Applications.....	6
2.2. Damping, and Measurement Techniques.....	12
2.3. Experimental Considerations .....	20
2.4. Heuristic Optimization Algorithms in the Study of Composites .....	22
CHAPTER 3. HEURISTIC OPTIMIZATION METHODS .....	26
3.1. Genetic Algorithms (GA) .....	26
3.2. Particle Swarm Optimization (PSO).....	37
3.2.1. PSO: The Basic Algorithm .....	40
CHAPTER 4. MATHEMATICAL MODEL DEVELOPMENT .....	42
4.1. Experimental Setup.....	42
4.2. Mathematical Model Baseline .....	47
4.3. Preliminary Experimental Observations .....	49
4.4. Final Model.....	52

CHAPTER 5. IMPLEMENTATION AND RESULTS.....	54
5.1. Optimization Algorithms Implementation.....	54
5.2. Benchmark Test of GA and PSO.....	58
5.2.1. Benchmark Test Results.....	60
5.3. Material Samples Description and Experimental Results.....	75
5.3.1. Optimization Results from Experimental Data.....	77
CHAPTER 6. DISCUSSION AND CONCLUSIONS.....	89
6.1. Suggestions for Future Research.....	95
REFERENCES.....	96
APPENDIX A.....	101
APPENDIX B.....	120

## LIST OF FIGURES

Figure 1.1. Honeycomb sandwich composite .....	2
Figure 1.2. Mechanical hysteresis loops examples .....	4
Figure 2.1. Plot of a hystereron of the second kind.....	8
Figure 2.2. Chua-Stromsmoe hysteresis loop.....	8
Figure 2.3. Input-output behavior of a hysteretic relay .....	9
Figure 2.4. Weighted summation of 3 hysteretic relays.....	9
Figure 2.5. Geometric interpretation of the Preisach model .....	10
Figure 2.6. Hysteresis loops from Bouc-Wen model .....	12
Figure 2.7. Loss factor ( $\eta$ ) as a function of frequency, of a 2024-T4 aluminum alloy cantilever beam, obtained with the Zener's damping model .....	15
Figure 2.8. Compliance FRF, and half power bandwidth method .....	16
Figure 2.9. General behavior of hysteresis in a system, when crossing resonance ( $f_0$ )...19	19
Figure 2.10. Categories of optimization techniques.....	23
Figure 3.1. Schematic of a parameter estimation method .....	26
Figure 3.2. The two versions of GA chromosome representation.....	28
Figure 3.3. Main steps for a typical Genetic Algorithm (GA) .....	31
Figure 3.4. Pseudo-code of a typical Genetic Algorithm (GA).....	31
Figure 3.5. Roulette wheel analogy, as a parent selection strategy .....	33
Figure 3.6. Two types of crossover points .....	34

Figure 3.7. Mating process, when the crossover point is between genes.....	35
Figure 3.8. Mating process, when the crossover point lies over a gene.....	35
Figure 3.9. PSO method of particle position correction at j-th iteration.....	38
Figure 3.10. PSO neighborhood topology types .....	39
Figure 3.11. PSO pseudo-code for synchronous implementation.....	41
Figure 3.12. PSO pseudo-code for asynchronous implementation .....	41
Figure 4.1. Concept drawing of the test rig for hysteresis loops generation.....	42
Figure 4.2. Pictures of the test rig for hysteresis loops generation .....	43
Figure 4.3. Experimental setup schematic, for hysteresis loops generation.....	44
Figure 4.4. Pictures of the experimental setup for hysteresis loops generation.....	45
Figure 4.5. Compliance FRF comparison, between a mid-point excited beam, and an equivalent 1-DOF system.....	48
Figure 4.6. Observed behavior of a sandwich composite beam; plot of $f_n$ vs. $X_0$ .....	49
Figure 4.7. Observed behavior of a sandwich composite beam; plot of $\xi$ vs. $X_0$ .....	50
Figure 4.8. Observed behavior of a sandwich composite beam; plot of $\xi$ vs. $f$ at different excitation amplitudes .....	50
Figure 4.9. Equivalent experimental mechanical system, considering the stiffness of the supports.....	51
Figure 4.10. Overlay plot example of the compliance frequency response of a sandwich composite beam sample around its first resonance, at different excitation amplitudes.....	52
Figure 5.1. Comparison of hysteresis loops for error calculation (displacement loading).....	56
Figure 5.2. Scatter plots from the results of the Synth set, for $\xi_0$ .....	63
Figure 5.3. Scatter plots from the results of the Synth set, for $S_\xi$ .....	64
Figure 5.4. Scatter plots from the results of the Synth set, for $\omega_0$ .....	65

Figure 5.5. Scatter plots from the results of the Synth set, for $S_{\omega}$ .....	66
Figure 5.6. Convergence plots from the results of the Synth set, for 0% noise .....	67
Figure 5.7. Convergence plots from the results of the Synth set, for 10% noise .....	68
Figure 5.8. Convergence plots from the results of the Synth set, for 20% noise .....	69
Figure 5.9. Convergence plots from the results of the Synth set, for 30% noise .....	70
Figure 5.10. Comparison plots for some of the hysteresis loops of the Synth set; 0% noise.....	71
Figure 5.11. Comparison plots for some of the hysteresis loops of the Synth set; 10% noise.....	72
Figure 5.12. Comparison plots for some of the hysteresis loops of the Synth set; 20% noise.....	73
Figure 5.13. Comparison plots for some of the hysteresis loops of the Synth set; 30% noise.....	74
Figure 5.14. Sketch of a sandwich composite beam sample .....	75
Figure 5.15. Parameters scatter plots from the optimization results of beam sample 1 ....	80
Figure 5.16. Parameters scatter plots from the optimization results of beam sample 2 ....	81
Figure 5.17. Parameters scatter plots from the optimization results of beam sample 3 ....	82
Figure 5.18. Optimization convergence plots for beam sample 1 .....	83
Figure 5.19. Optimization convergence plots for beam sample 2.....	84
Figure 5.20. Optimization convergence plots for sample beam 3.....	85
Figure 5.21. Optimization comparison plots for some of the hysteresis loops of sample 1 .....	86
Figure 5.22. Optimization comparison plots for some of the hysteresis loops of sample 2.....	87
Figure 5.23. Optimization comparison plots for some of the hysteresis loops of sample 3.....	88

Figure 6.1. Overlay compliance FRF plots of all 3 beam samples utilized,  
for low, medium, and high excitation amplitudes.....91

## LIST OF TABLES

Table 3.1. Parameters properties for decoding the binary chromosome of Fig. 3.2.....	29
Table 5.1. Mutated parameters of the best partial solution (of 4 parameters), for the creation of an additional set of candidate solutions (L=3) for local search .....	55
Table 5.2. Excitation frequencies of the computer-generated Synth hysteresis set .....	59
Table 5.3. System parameters used to generate the Synth hysteresis set .....	59
Table 5.4. Search space for the Synth hysteresis set .....	60
Table 5.5. Optimization results for the Synth data set; GA, 0% noise.....	61
Table 5.6. Optimization results for the Synth data set; GA, 10% noise.....	61
Table 5.7. Optimization results for the Synth data set; GA, 20% noise.....	61
Table 5.8. Optimization results for the Synth data set; GA, 30% noise.....	61
Table 5.9. Optimization results for the Synth data set; PSO, 0% noise .....	62
Table 5.10. Optimization results for the Synth data set; PSO, 10% noise .....	62
Table 5.11. Optimization results for the Synth data set; PSO, 20% noise .....	62
Table 5.12. Optimization results for the Synth data set; PSO, 30% noise .....	62
Table 5.13. Differences between the resultant averages and the actual global optimum, for GA and PSO, at different noise levels.....	75
Table 5.14. Properties of the sandwich composite beams studied .....	75
Table 5.15. Excitation frequencies used with sample 1 .....	76
Table 5.16. Excitation frequencies used with sample 2 .....	76

Table 5.17. Excitation frequencies used with sample 3 .....	77
Table 5.18. Excitation displacement amplitudes used with the beam samples studied ....	77
Table 5.19. Search spaces for all 3 beam samples studied .....	77
Table 5.20. GA optimization results for beam sample 1 .....	78
Table 5.21. GA optimization results for beam sample 2 .....	78
Table 5.22. GA optimization results for beam sample 3 .....	78
Table 5.23. PSO optimization results for beam sample 1 .....	79
Table 5.24. PSO optimization results for beam sample 2 .....	79
Table 5.25. PSO optimization results for beam sample 3 .....	79

# CHAPTER 1

## INTRODUCTION

### 1.1. Composite Materials and Damping

Composite materials are being increasingly used as alternatives to conventional materials for highly demanding structural applications, such as the construction of marine, automobile, and spacecraft structures, which include the equipment panels, payload platforms, solar panels, and antenna reflectors <sup>[1]</sup>. Composites are mainly used for these applications because they have the desired properties of high specific strength, specific stiffness and tailorable design, along with a low mass. However, they can be quite different from isotropic materials, because the former may have (for instance) certain failure characteristics, such as core or matrix crazing, delamination, and interface debonding, which typically do not occur in more conventional materials.

Damping is an important parameter related to the study of the dynamic behavior of composite structures in general. There are numerous types of composites currently available, but the present study deals with the use of a specific kind of composite construction called “sandwich composite”. A sandwich composite (or “sandwich panel”) is a structure consisting of two thin faces bonded to a thick lightweight core <sup>[2-3]</sup>. The materials used to construct the faces are usually aluminum or a composite laminate, and the core can be a lightweight foam or a honeycomb structure of some other material, such as shown in Fig. 1.1.



**Figure 1.1.** Honeycomb sandwich composite; (a) detail of the core, (b) composite beam

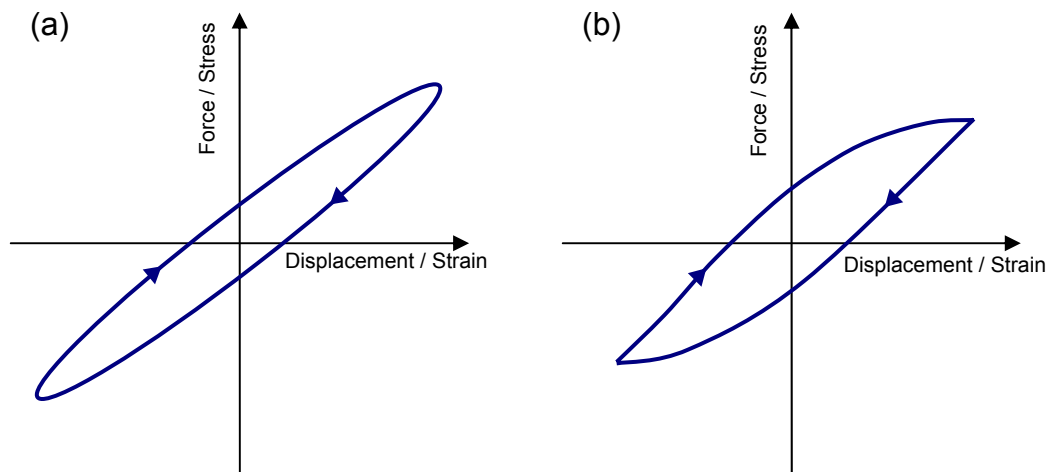
Chandra *et al* <sup>[4]</sup> make an extensive review of damping mechanisms in composites, some of which are detailed below in order to enhance the understanding of this important physical phenomenon:

- *Viscoelastic nature of matrix and/or fiber materials*: the matrix shows the most significant contribution to the damping. In the case of high damping fibers, such as in carbon and kevlar composites, the effect of the fibers on damping should be included in the analysis.
- *Damping due to interfaces*: the interfaces are the regions where adjacent surfaces exist. In fiber-reinforced composites, the interfaces are the areas where the fibers are in direct contact with the matrix. In sandwich composites, on the other hand, the interfaces are the areas where the different layers are in contact with one other.
- *Damping due to damage*: this may be caused by frictional damping due to slip in the unbound regions between the fiber/matrix interface, and may also be caused by delaminations, and damping due to energy dissipation in the area of matrix cracks, broken fibers, etc.

There are also other mechanisms of energy dissipation in composites, including viscoplastic and thermoelastic damping, which depend on the nonlinearities seen for high amplitude vibration levels, and on the cyclic heat flow from regions of compressive to tensile stresses, respectively.

Several damping measurement techniques have been developed, and the applicability of each of these methods is normally dependent on the frequency range of interest, and whether or not the damping in a resonance frequency is to be measured. Among the several approaches available for damping measurement, the half power method, logarithmic decrement, and hysteresis loops analysis are generally the most popular. This topic will be discussed in more detail in section 2.2.

A mechanical hysteresis loop consists of a phase plot of the force vs. the displacement usually at a specific point in a structure. Equivalently, it may be also considered as a plot of stress vs. strain. For dynamical systems that include damping, the displacement/force relationship is typically out of phase near resonance, generating an elliptically-shaped hysteresis loop, for a linear system, or a sharp-edged loop, for a non-linear system, as shown in Fig. 1.2. The area enclosed by the hysteresis loop is proportional to the damping in the system.



**Figure 1.2.** Mechanical hysteresis loops examples; (a) linear system, (b) non-linear system

## 1.2. Scope of Study

The present work examines the hysteresis loop technique for the characterization of the vibrational dynamics of three beam samples of sandwich composite materials, consisting of a paper honeycomb core filled with foam and external layers of interlaced strips of carbon fibers, such as was shown in Fig. 1.1b. This combination provides great bending stiffness and endurance to a very light structure. These characteristics, along with the great number of interfaces in a sandwich composite of this nature, typically results in higher damping than standard homogeneous materials.

The main goal is to fit a parametric mathematical model to several experimental hysteresis loops obtained for specific sandwich composite materials, simultaneously for different displacement amplitudes and frequencies around the first resonance of the beam sample. Experimental observations of the behavior of the materials are made, in order to tailor a simple baseline mathematical model to another that includes such effects. Due to the large scale of the system identification process, and the number of parameters to identify, two popular heuristic optimization methods have been used in this problem: Genetic Algorithms (GA), and Particle Swarm Optimization (PSO). Before using these optimization techniques with actual experimental data obtained from the composite samples with an unknown global optimum solution, computer-generated hysteresis loops, with a known global optimum solution, and with similar vibrational behavior of actual composites, are created. This is done to test the search performance of both optimization techniques, with added Gaussian random noise of different levels.

## CHAPTER 2

### LITERATURE REVIEW

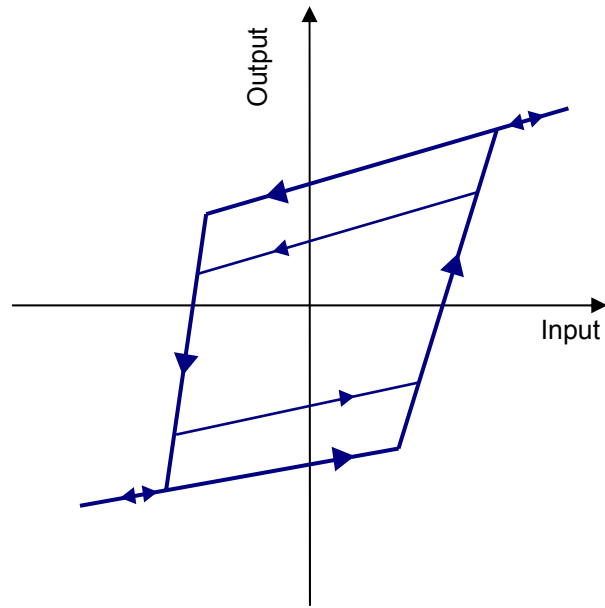
#### 2.1. Hysteresis Models and Applications

The hysteresis phenomenon appears in several physical systems, such as mechanical and magnetic materials and devices. In some cases, it can cause some undesirable effects, such as loss of stability robustness; but it is mostly a way in how energy is dissipated in a system, whose behavior typically characterizes damping. There are several mathematical models of hysteresis, which are useful for different types of systems. Their usefulness depends on the field of study to which they are applied, and also depends on the type of materials to be characterized and the amount of nonlinearities present. Sain <sup>[5]</sup> presents a survey on hysteresis models, focusing on four popular representations, which have been used with success on general control designs. They are: hysterons, the Chua-Stromsmoe, Preisach, and the Bouc-Wen model.

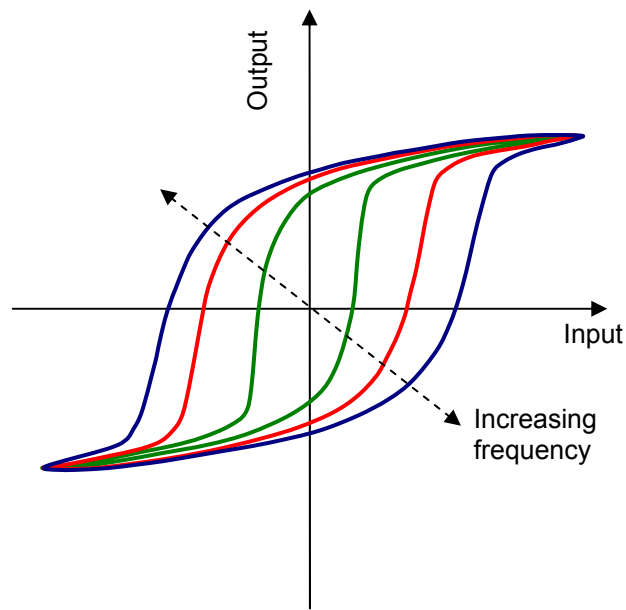
Hysterons were developed by Krasnosel'skii and Pokorovskii <sup>[6]</sup>, with the intent of establishing a basic and general model of hysteresis. In essence, a hysteron represents a transducer, which maps input functions into output functions. These output functions are characterized by a family of curves, used to determine the output value of a specific input, depending on its previous coordinate position and direction. In this way, the model allows the state of the system to oscillate in a closed-loop manner, as pictured in Fig. 2.1.

The Chua-Stromsmoe hysteresis model <sup>[7]</sup> was created primarily with the intent of modeling the behavior of ferromagnetic hysteretic inductors, with some suitable characteristics, such as frequency-dependence control of the width of the loops. It is given by the expression:  $\dot{x} = w(\dot{u})h(x)g[u(t) - f(x)]$ , with  $x(t_0)=x_0$ , where  $u$  is the input, and  $x$  is the output. The functions named  $f$  and  $g$  represent dissipative and restorative phenomena, respectively;  $h$  controls the width of the loop, and  $w$  controls the frequency-dependent widening behavior. A plot describing this frequency widening effect is shown in Fig. 2.2, for  $w(\dot{u})=h(x)=1$ ,  $f(x)=100x^3$ ,  $g(u)=u^5$ , and  $u(t)=\sin(\omega t)$ .

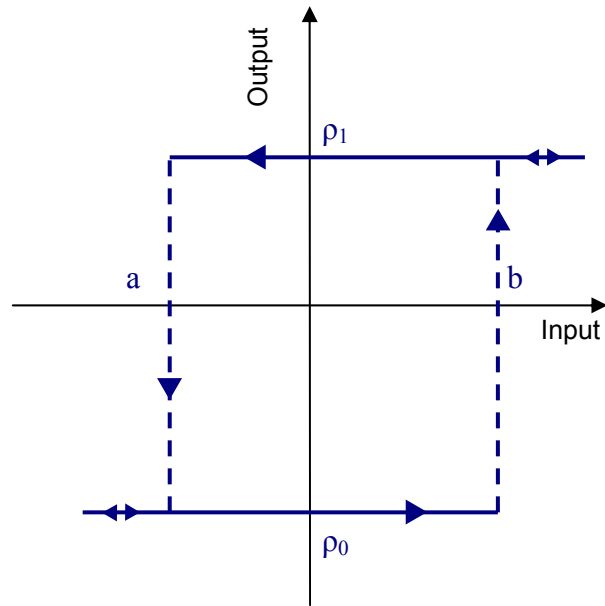
The Preisach model of hysteresis <sup>[8]</sup> is based on a weighted summation of the outputs of a set of hysteretic relays, each of them integrated over a set of individual switching thresholds,  $a$  and  $b$ . This means that, when a threshold is attained for a single relay, the output changes between  $\rho_1$  and  $\rho_0$ , (which are the high and low levels, respectively), and remains at a specific level until the next threshold is attained, where it switches back to the initial level. The basic behavior of such relay is shown in Fig. 2.3. A first discrete approximation of the behavior of the Preisach model can be seen on Fig. 2.4, where the weighted summation of the outputs of three parallel relays generates a very basic hysteresis loop. In the limit, where an infinite number of relays are working in parallel, a continuous response can be attained, and it can be interpreted into an intuitive geometrical representation. If we consider an  $a$ - $b$  plane, and for a monotonically increasing input value, the output of the Preisach model corresponds to the area obtained from the bottom left corner of a minimum value over an  $a=b$  line up to the current value of the input (Fig. 2.5a). Then, for a monotonically decreasing input, it vertically wipes this area, from the maximum value reached up to the current input value (Fig. 2.5b).



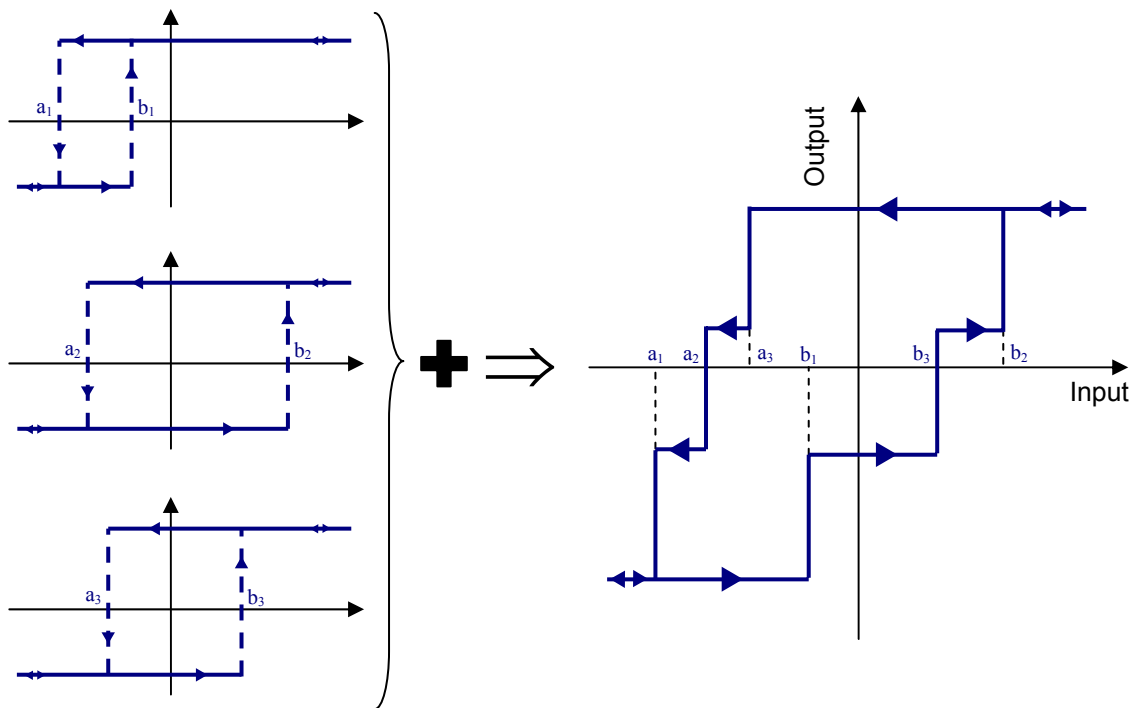
**Figure 2.1.** Plot of a hysteron of the second kind



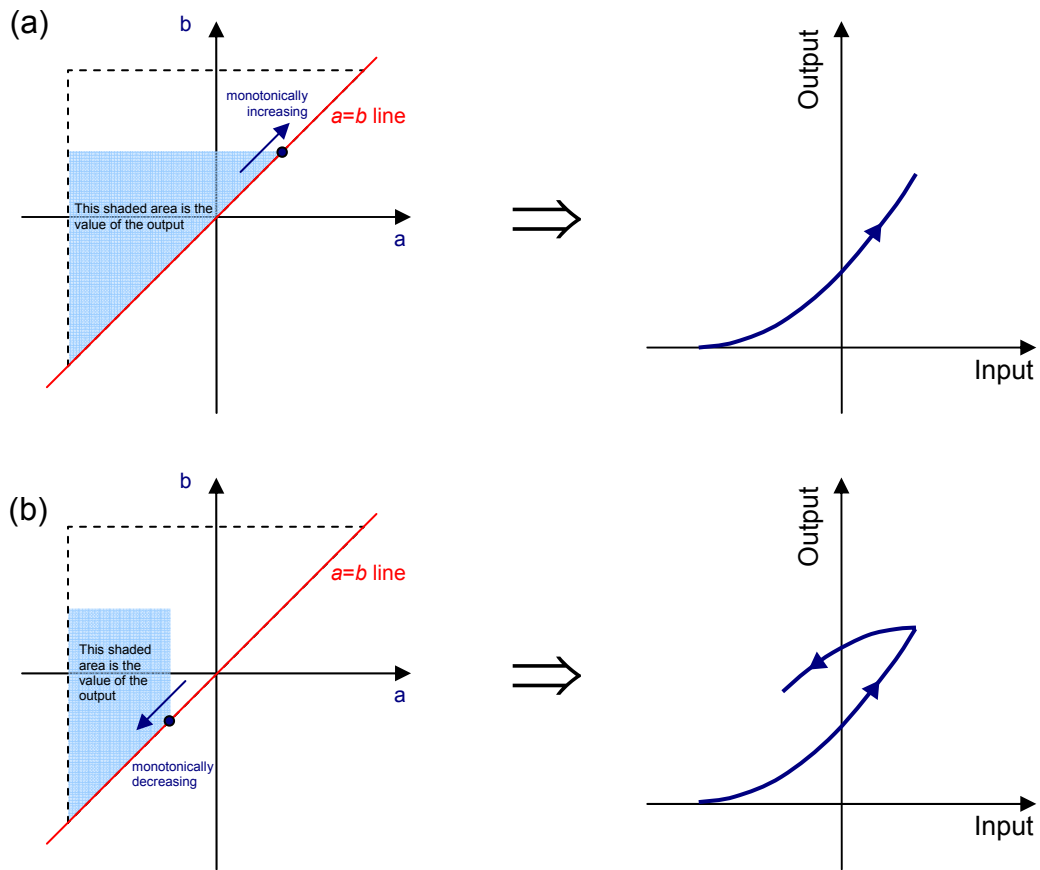
**Figure 2.2.** Chua-Stromsmoe hysteresis loop: Effect of increasing the loop area, for increasing excitation frequencies



**Figure 2.3.** Input-output behavior of a hysteresis relay



**Figure 2.4.** Weighted summation of 3 hysteresis relays



**Figure 2.5.** Geometric interpretation of the Preisach model;  
 (a) monotonically increasing input, (b) monotonically decreasing input

Finally, the Bouc/Wen model <sup>[9-10]</sup> is one of the most popular mathematical representations of several hysteresis phenomena, but it is mostly used in vibrations problems, as it is based on the differential equation of motion of a spring-mass-viscous damper mechanical system. The difference from the standard linear model, is that it adds an extra term in the equation, referred as the hysteretic nonlinear term. This model can be expressed as a system of two coupled nonlinear differential equations, defined by Eq. (2.1).

$$\begin{aligned}
\text{a) } & \ddot{x} + 2\xi\omega_n\dot{x} + \alpha\omega_n^2x + (1-\alpha)\omega_n^2z = f(t) \quad , \\
\text{b) } & \dot{z} = -\gamma|\dot{x}||z|^{(n-1)}z - \beta\dot{x}|z|^n + ax \quad ,
\end{aligned}
\tag{2.1}$$

where the parameters of the system are:

$\alpha$  : rigidity ratio ( $0 \leq \alpha \leq 1$ ),

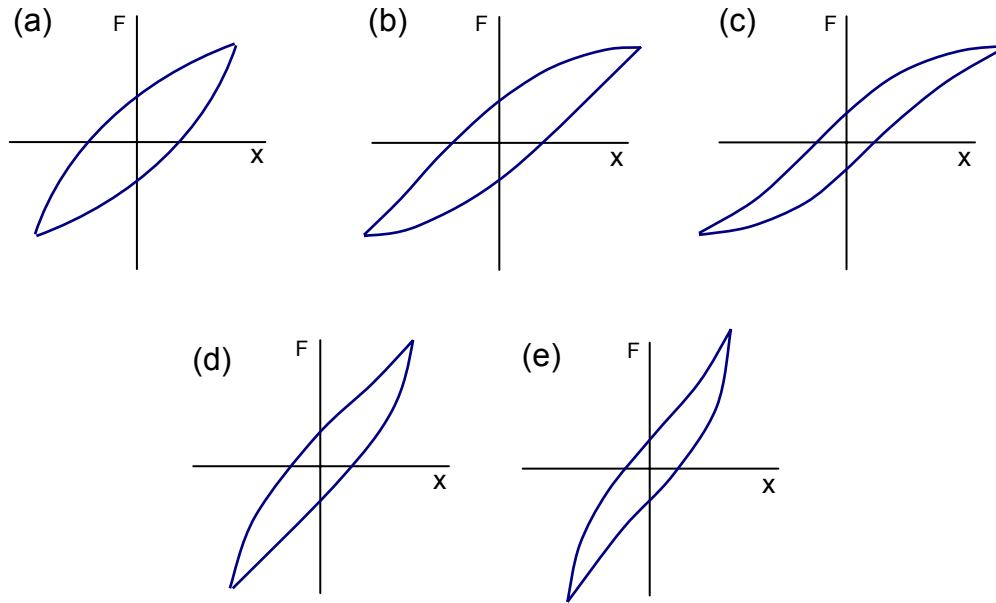
$\xi$  : linear elastic viscous damping ratio ( $0 \leq \xi \leq 1$ ),

$\omega_n$  : pseudo-natural frequency of the system (rad/s),

$a$  : parameter controlling hysteresis amplitude,

$\beta, \gamma, n$  : parameters controlling hysteresis shape ( $\gamma > 0, n \geq 1$ ).

The  $f(t)$  term in Eq. (2.1a) is a mass normalized forcing function (N/kg). The hysteretic (or memory) variable  $z$  is a “fictitious” displacement related to the actual displacement,  $x$ . This variable accounts for the dependence of the nonlinear restoring force (last term on the left-hand side of Eq. (2.1a)) on not only the current displacement value, but on the previous time history of the motion, as well. Different combinations of the parameters yield different shapes for the hysteresis loops. To illustrate this let  $n=1$  and vary only the parameters  $\beta$  and  $\gamma$ . Different possible stable hysteresis loops <sup>[11]</sup> are shown in Fig. 2.6. The versatility of the system of differential equations described in Eq. (2.1) can be observed. Numerous researchers have successfully used this model. For example, Ni <sup>[12]</sup> and Constantinou <sup>[13]</sup> used it for the study of nonlinear hysteretic isolators; Smyth <sup>[14]</sup> for an adaptive application and Heine <sup>[15]</sup> for an optimization approach to degrading hysteretic joints with slack behavior.



**Figure 2.6.** Hysteresis loops from Bouc-Wen model, obtained by varying  $\beta$  and  $\gamma$ ;  $n=1$ ;  
 (a)  $\beta+\gamma>0$ ,  $\beta-\gamma<0$ , (b)  $\beta+\gamma>0$ ,  $\beta-\gamma=0$ , (c)  $\beta+\gamma>\beta-\gamma>0$ ,  
 (d)  $\beta+\gamma=0$ ,  $\beta-\gamma<0$ , (e)  $\beta-\gamma<\beta+\gamma<0$

## 2.2. Damping, and Measurement Techniques

Extensive research has been performed on the topic of energy dissipation of materials, whose damping mechanisms and modeling depend on specific types of materials. For metals, viscoelastic materials, etc. Macioce <sup>[16]</sup> gives a brief but precise explanation of damping and some of its measurement techniques, as follows.

A material or structure subjected to vibrations has a combination of potential and kinetic energy, along with a dissipative element, which converts some of this mechanical energy into heat during each cycle of motion, which is then lost when it is transferred to the surrounding environment. The amount of dissipated energy is related to the damping of the structure, which is a measure of the ability of the system to dissipate mechanical energy into thermal energy. Several forms of damping have been identified, like

Coulomb friction, fluid viscosity, air damping, particle damping, magnetic hysteresis, and acoustic radiation damping. As it can be seen, the environment can also play a role in providing a mechanism for added external damping, but it is insignificant in most cases. The most common models of damping used in solving vibrations problems, are the viscoelastic damping (also called “structural”, “material”, or “hysteretic” damping), viscous damping, and Coulomb damping.

Viscoelastic damping is found in a wide range of materials (known as “viscoelastic materials”), such as rubbers, epoxies, and foams. One main characteristic is that it is rate-independent, which means that the energy stored per cycle remains constant when changing the excitation frequency. And another important characteristic is that its modulus of elasticity (a function of the temperature  $T$  and frequency  $f$ , in general) is expressed as a complex term, which is a sum of a conservative part ( $E_1$ ) and a dissipative part ( $E_2$ ), as shown in Eq. (2.2).

$$E(T, f) = E_1 + iE_2 = E_1(1 + i\eta) \quad , \quad (2.2)$$

where  $\eta$ =loss factor (a descriptor of damping). So, if the behavior of this type of material is included in a 1-DOF (Degree of Freedom) system, its equation of motion is represented in Eq. (2.3).

$$m\ddot{x}(t) + k \cdot (1 + i\eta)x(t) = F(t) \quad . \quad (2.3)$$

On the other hand, the viscous damping model is rate-dependent, and it may generally be used for some non-viscoelastic materials. A 1-DOF equation of motion including this type of damping is shown in Eq. (2.4), where  $\xi = \eta/2$ , and  $\omega_n = (k/m)^{1/2}$ .

$$m\ddot{x}(t) + c\dot{x}(t) + kx(t) = F(t) \quad \Rightarrow \quad \ddot{x}(t) + 2\xi\omega_n\dot{x}(t) + \omega_n^2x(t) = F(t) \quad . \quad (2.4)$$

Finally, Coulomb damping (also called “dry damping”) <sup>[17-18]</sup> is produced from the sliding friction of two adjacent dry surfaces. The dry friction force that appears is parallel to the surface, and its magnitude is proportional to the normal force on the body (such as its own weight), but in a direction opposite to that of the velocity. A 1-DOF equation of motion with this type of damping is shown in Eq. (2.5), with  $F_d = \mu \cdot N$  = magnitude of the damping force,  $\mu$  = coefficient of friction,  $N$  = force normal to the sliding surface, and  $\text{sgn}(\dot{x}) = \dot{x}/|\dot{x}|$  = sign of the velocity.

$$m\ddot{x}(t) + F_d \text{sgn}(\dot{x}) + kx(t) = F(t) . \quad (2.5)$$

Some other models of damping have been developed, such as Zener’s damping model for metals <sup>[19]</sup>. As explained by Crandall <sup>[20]</sup>, this model is frequency dependent, and is based on the assumption of the existence of heat flow from warmed compression fibers in a material to cooled tension fibers. The mathematical expression for the loss factor,  $\eta$ , for this model is shown in Eq. (2.6) <sup>[20-21]</sup>. A plot of the modeled loss factor for a 2024-T4 aluminum alloy cantilever beam is shown in Fig. 2.7.

$$\eta = \Delta_r \frac{\omega \tau_s}{1 + \omega^2 \tau_s^2} , \quad (2.6)$$

where:

$$\Delta_r = \frac{\alpha_T^2 E_T T}{\rho C_p} = \text{relaxation strength},$$

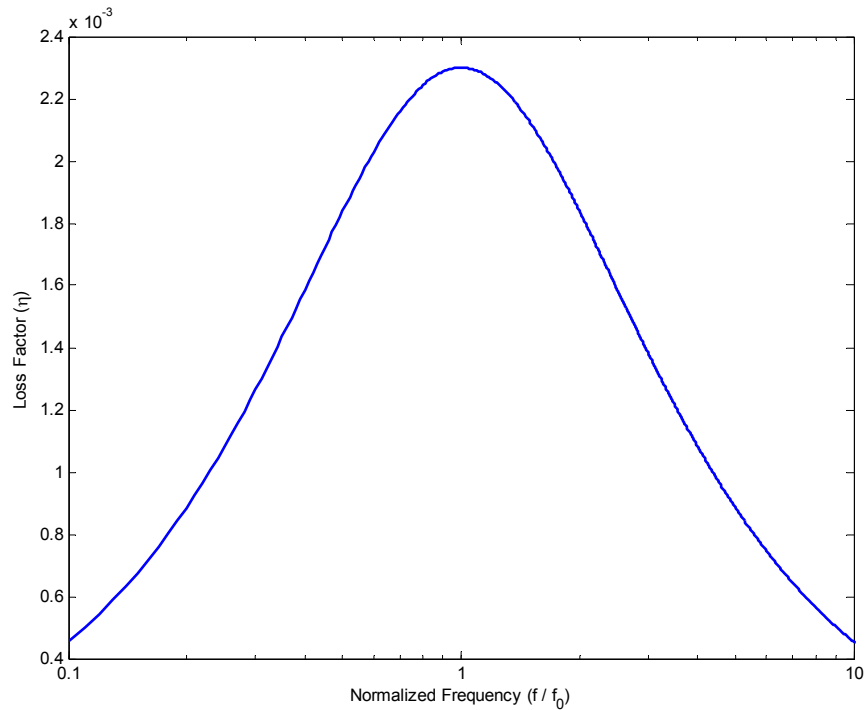
$$\tau_s = \left( \frac{d}{\pi} \right)^2 \frac{\rho C_p}{K} = \text{thermal relaxation time},$$

$$\omega = \text{frequency of vibration},$$

$$\alpha_T = \text{temperature expansion coefficient},$$

$$E_T = \text{Young’s modulus at constant temperature},$$

- $T$  = absolute temperature,
- $\rho C_p$  = specific heat per unit volume,
- $d$  = material sample thickness,
- $K$  = thermal conductivity.



**Figure 2.7.** Loss factor ( $\eta$ ) as a function of frequency, of a 2024-T4 aluminum alloy cantilever beam, obtained with the Zener's damping model

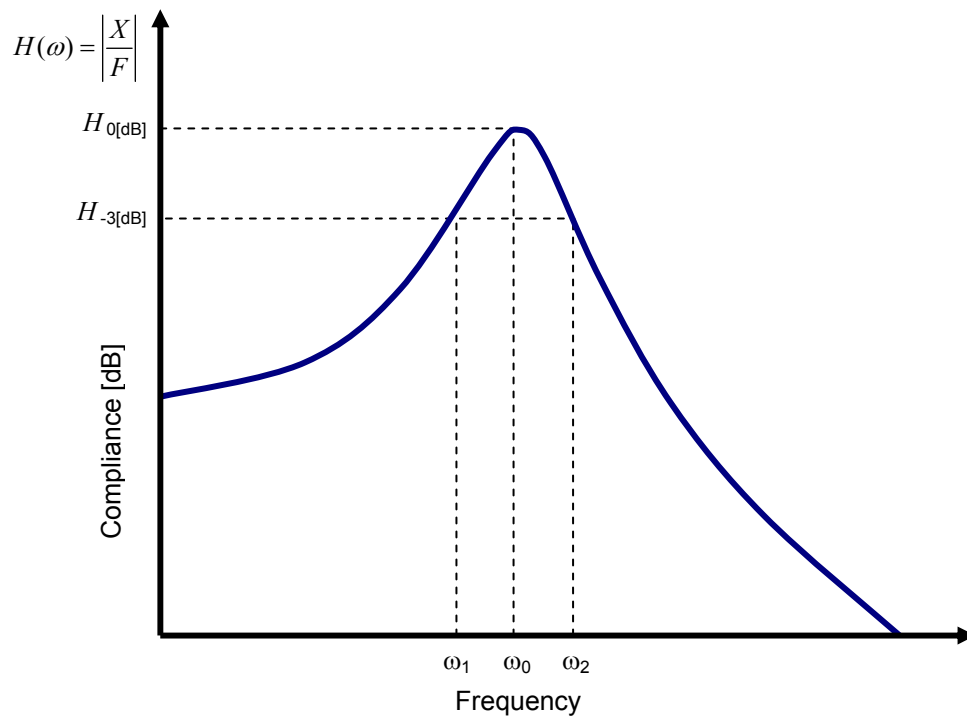
In several applications, the damping is usually very light, and its effect is observable only around resonances. Also, in many practical situations, it is common to represent the damping of a system with an equivalent viscous damper (dashpot), which simplifies the mathematical calculations, and provides a sufficiently accurate approximation to the total damping. In order to experimentally quantify the amount of damping present in a material or structure, there are several measurement techniques developed by researchers in the area of vibration analysis. Three of the most popular

techniques used are <sup>[16,17,21]</sup>: the half power bandwidth method, logarithmic decrement, and hysteresis loops analysis.

The half power bandwidth method consists of determining the modal damping from a classic frequency response function (FRF), such as the compliance response (displacement/force), also known as admittance, or receptance. The amount of damping for a specific mode of vibration will depend on the sharpness of the resonance peak, i.e. the rounder the peak, the larger the damping. From the compliance function shown in Fig. 2.8, the damping ratio  $\xi$  can be determined with Eq. (2.7).

$$\xi = \frac{\eta}{2} = \frac{\omega_2 - \omega_1}{2 \cdot \omega_0} , \quad (2.7)$$

where  $\omega_0$ =frequency of resonance peak,  $\omega_1, \omega_2$ =lower and upper frequencies of -3dB from the resonance peak. This method is also often referred as the “3dB method”.



**Figure 2.8.** Compliance FRF, and half power bandwidth method

The logarithmic decrement method consists of computing the damping of an unforced underdamped system from the rate of decay of its response in the time domain. The amount of amplitude that has decreased in the system between two consecutive cycles (at times  $t_1$  and  $t_2$ , respectively) is shown in Eq. (2.8) <sup>[17]</sup>, and since  $t_2 = t_1 + T = t_1 + 2\pi/\omega_d$ , the expression reduces to Eq. (2.9).

$$\frac{x_1}{x_2} = \frac{Ae^{-\xi\omega_n t_1} \cos(\omega_d t_1 - \phi)}{Ae^{-\xi\omega_n t_2} \cos(\omega_d t_2 - \phi)} , \quad (2.8)$$

$$\frac{x_1}{x_2} = \frac{e^{-\xi\omega_n t_1}}{Ae^{-\xi\omega_n (t_1 + T)}} = e^{\xi\omega_n T} . \quad (2.9)$$

Because of the exponential nature of Eq. (2.9), it is proper to introduce the notation:

$$\delta = \ln \frac{x_1}{x_2} = \xi\omega_n T = \frac{2\pi\xi}{\sqrt{1-\xi^2}} , \quad (2.10)$$

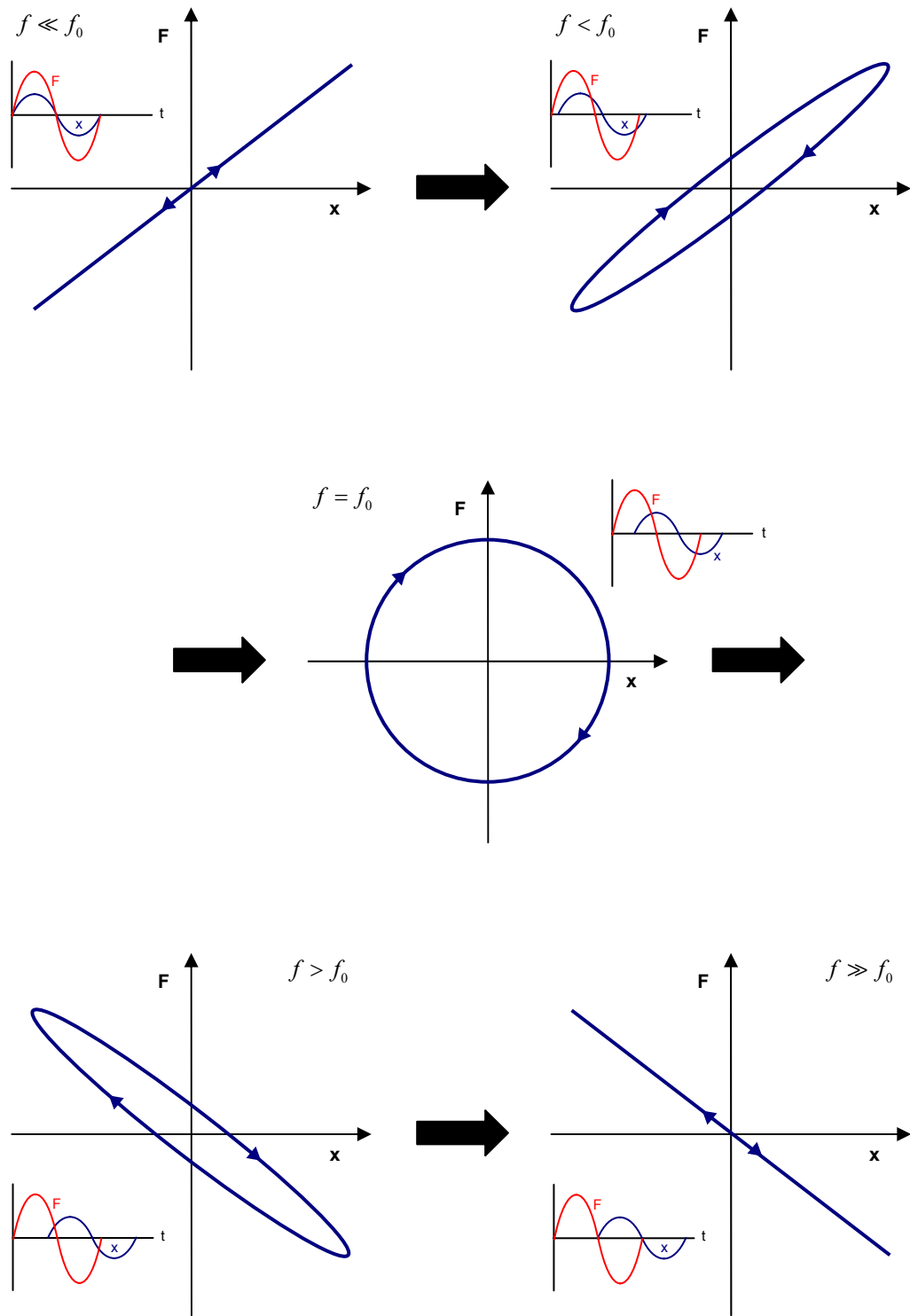
where  $\delta$  is the logarithmic decrement. Now, solving for  $\xi$ , the damping ratio (in terms of  $\delta$ ) is given by Eq. (2.11).

$$\xi = \frac{\delta}{\sqrt{(2\pi)^2 + \delta^2}} \cong \frac{\delta}{2\pi} = \frac{1}{2\pi} \ln \left( \frac{x_1}{x_2} \right) . \quad (2.11)$$

The approximation given in Eq. (2.11) is true for small values of damping, which is often encountered in most systems. It is also possible to determine the damping ratio by measuring the logarithmic decrement of two displacements separated by  $N$  cycles, utilizing Eq. (2.12).

$$\delta = \frac{1}{N} \ln \left( \frac{x_1}{x_{N+1}} \right) \Rightarrow \xi \cong \frac{1}{N \cdot 2\pi} \ln \left( \frac{x_1}{x_{N+1}} \right) . \quad (2.12)$$

The hysteresis loop method of damping analysis consists in calculating the energy loss per cycle at steady-state harmonic loading. This may be achieved by creating a plot of displacement vs. force (or equivalently, strain vs. stress) of the system at a fixed excitation frequency, and determining the area enclosed by this hysteresis loop, which is precisely the energy dissipated per cycle, and is proportional to the damping in the system. This type of plot is equivalent to the Lissajous figures of two signals being (in general) out of phase. Below resonance (and far from it), the hysteresis plot usually shows a straight line with positive slope. This means that both, the displacement and the force waves, are in phase. When the excitation frequency is close to resonance and below it, both signals are slightly out of phase (displacement lags the force), and the hysteresis loop is a tilted ellipse with positive slope (for linear systems, as in the viscous and viscoelastic damping models), as shown in Fig. 1.2a, or a closed but not-elliptical loop for the nonlinear case, as shown in an example in Fig. 1.2b. At resonance, both signals are exactly  $90^\circ$  out of phase, and the hysteresis is a non-tilted closed loop, or in some cases, a circle, for the linear case. Above resonance and close to it, the hysteresis loop is a closed loop with negative slope. Finally, above resonance and far from it, the hysteresis turns again into a straight line, but with negative slope, which means that the displacement lags the force in  $180^\circ$ . A summary of all this behavior is shown in Fig. 2.9. An equivalent methodology of damping analysis using this technique consists of performing a curve-fitting of the mathematical model of motion of the system to the actual loops obtained experimentally, this way finding a set of parameters from the model that is able to represent the actual dynamics of the system in reality.



**Figure 2.9.** General behavior of hysteresis in a system, when crossing resonance ( $f_0$ ); hysteresis loops, along with time trace overlays of displacement ( $x$ ) and force ( $F$ )

A variety of novel ideas for damping measurement have been introduced by researchers, who have also described the critical factors for specific problems. Hooker <sup>[22]</sup>, for example, explains how damping measurements for biaxial hysteresis loops yield apparent loss factors, which may be related to the true loss factors of the system. On the other hand, So, *et al* <sup>[23]</sup> shows a method for frequency-response-based damping measurement using a periodic chirp excitation, similar to pseudo-random noise excitation, but with a much higher peak-to-rms ratio. Also, McDaniel, Dupont, and Salvino <sup>[24]</sup> introduce an approach for the determination of a frequency-dependent damping from transient loadings.

### **2.3. Experimental Considerations**

The common methodology used as a first approach in the development of mathematical models is assuming that the system is “perfect”, in the sense that its behavior is supposed to be completely represented by the proposed equations. In reality, this is almost impossible to achieve, as several additional factors related to the system, such as effects from the environment, material damage, or attachments, may have not been considered. This may alter the accuracy of the model predictions. Hence, experimental considerations must be made in order to incorporate these effects into the equations, with the purpose of getting as close as possible to the exact solutions. Several researchers have addressed these practical issues as a way to improving the predictions of their models. Some of them are presented next, which are relevant to the present work.

Renji and Shankar Narayan <sup>[1]</sup>, along with Brown and Norton <sup>[25]</sup>, explained the importance of considering the effect of mass of the force transducer (or impedance head)

in their experiments. They argue that the measured force by the transducer will be actually different from the actual applied force. Its mass causes a variation in the impedance at the attachment point. Thus, the driving point admittance  $Y$ , as well as the transducer admittance  $Y_m$ , must be considered in order to obtain the actual force  $f_a$  from the measured force  $f_m$ , as shown in Eq. (2.13).

$$f_a = \frac{f_m}{1 + (Y/Y_m)} . \quad (2.13)$$

The admittance introduced by the mass of the transducer is simply  $Y_m = 1/(j\omega M)$ , and the driving point admittance is  $Y = \phi_{fv} / \phi_{ff}$ , where  $\phi_{ff}$  is the auto-spectral density of the force, and  $\phi_{fv}$  is the cross-spectral density between the force and the velocity. The direct use of the measured force (without corrections) in models may lead to errors in the modal damping values calculated, although it has been seen that this effect is more significant at high frequencies rather than low frequencies, where in the latter the effect may be neglected without significant loss of accuracy.

On the other hand, Boyaci <sup>[26]</sup> explains the effects of non-ideal boundary conditions as follows. Beam samples are commonly used for the study of physical properties of materials, and their supports may have a direct effect on the experimental results. Different types of boundary conditions are defined, such as clamped, free, simply supported, etc. However, there may be small deviations from these ideal conditions, such as small gaps and friction in the attachments, which may introduce small deflections and moments at the ends. These non-ideal boundary conditions are modeled using perturbation theory. More specifically, the method of multiple scales is used. It was found

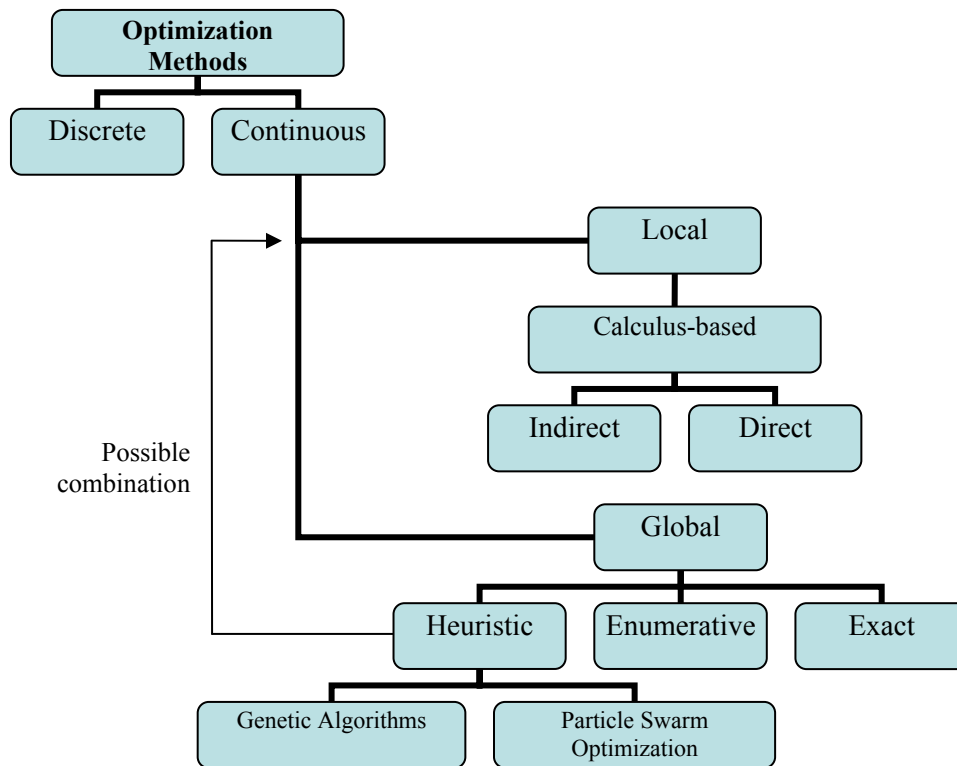
that while the stretching effect increases the frequencies, the non-ideal boundary conditions may increase or decrease them.

Finally, Qiao, Pilpchuk, and Ibrahim <sup>[27]</sup> made a similar analysis, focusing on parameter uncertainties and relaxation of joints. They suggest that relaxation effects in the attachments at the boundaries will cause time-dependent boundary conditions, and will depend on the level of structural vibration. The main problems that are encountered in those cases are the existence of random eigenvalues, and probability of failure. The analysis of stiffness uncertainties at the joints under dynamic excitation were performed, with and without the effect of the relaxation process. It has been found that the frequency of resonance of a material is reduced when the “level of clamping” is reduced. Also, the relaxation of the joints produces a gradual decrease in the resonance frequency.

#### **2.4. Heuristic Optimization Algorithms in the Study of Composites**

The design of materials and system identification of model parameters provide an optimization problem which happens to be difficult in most cases. First because of the functional forms of the optimization descriptors often encountered in engineering problems, and secondly, because of large dimensionalities of the search spaces to be analyzed. Several computational tools are available for optimization, which in turn may be categorized as shown in Fig. 2.10 <sup>[15]</sup>. As can be seen, heuristic optimization corresponds to the continuous and global search methods, which makes this category very powerful and versatile as compared to other techniques. Their most important characteristics are: (1) they don't need a starting search point (initial guess), as they generate populations of candidate solutions, usually spanning a great area of the search

space simultaneously, and (2) they work in a random fashion (i.e. not using calculus operations in the search), looking for areas of interest in the search space, and gradually converging to optimum solutions within these areas. Because of this random nature of the methodology, different runs of the optimization computer routines yield also different results, but generally close to each other.



**Figure 2.10.** Categories of optimization techniques

Heuristic optimization methods have been extensively used for solving problems in different areas, such as in mechanics, electronics, economy, and several others. Focusing on the topics relevant to this work, we have found publications on the utilization of Genetic Algorithms (GA) and Particle Swarm Optimization (PSO) in

composite materials, for design optimization, extraction of physical properties, and system identification of model parameters.

Liu, Han, and Lam <sup>[28]</sup> used a combination of GA with a nonlinear least squares method (LSM) in a problem of material characterization, where the elastic constants, (Young's moduli, shear modulus, and Poisson's ratios) of some composite plates were required to be determined. They used GA to find a solution close to the optimum, which is then used as an initial guess for LSM to finally converge to that point. Matsuoka, Yamamoto, and Takahara <sup>[29]</sup> used a combination of GA and finite element stress analysis on the search of material structures that give way to desired elastic properties. Muc and Gurba <sup>[30]</sup> used a combination of GA and finite element analysis for layout optimization of composite structures, which consists in finding information on stacking sequences, shape, and sizes of the material layers, in order to achieve optimum mechanical properties. Gantovnik, Gürdal, and Watson <sup>[31]</sup> also used GA for the optimal design of laminated sandwich composite panels, in which memory is included for continuous variables in order to preserve data from previous analyzed designs, thus avoiding the repetition of an analysis already performed, and then reducing the computational costs.

Regarding the use of PSO in composites design, Kovacs, Groenwold, Jarmai, and Farkas <sup>[32]</sup> have used this technique for the optimal design of a carbon-fiber-reinforced plastic (CFRP) sandwich-like structure with aluminum webs, in order to achieve a minimum cost and maximum stiffness. The CFRP plate is optimized with respect to ply arrangement, while the complete structure is optimized with respect to a combination of manufacturing and material cost. Sekishiro, and Todoroki <sup>[33]</sup> also used PSO for the optimal design of laminated composite structures, focused in achieving a minimum

weight of a hat-stiffened composite panel subjected to a buckling constraint. In order to reach their goal, they used the fractal branch and bound method (a stacking sequence optimization method) integrated into PSO.

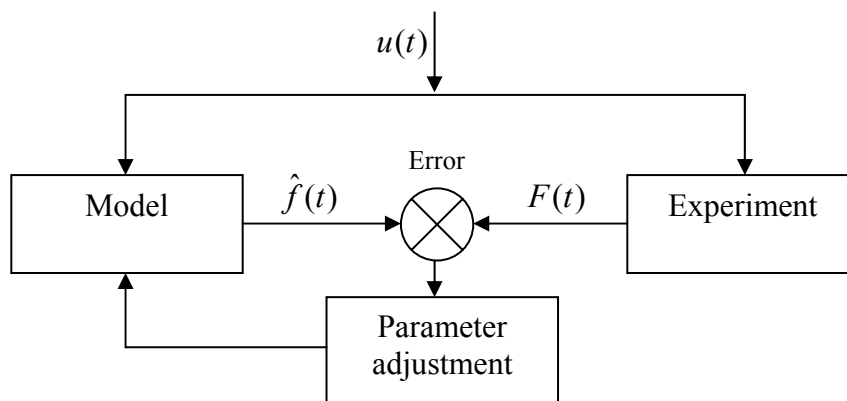
In the topic of heuristic optimization methods applied to mechanical hysteresis, the publications available are focused more on GA, rather than PSO, but in neither case focused specifically on composites. Heine <sup>[15]</sup>, however, studied the response of degrading hysteretic joints with slack behavior in connected timber members, by using GA for the identification of the parameters of a modified Bouc-Wen hysteresis model. Kyprianou and Worden <sup>[34]</sup> carried out a generalized study of the performance of GA on system identification of the Bouc-Wen hysteresis model. Zhang, *et al* <sup>[35]</sup>, also presented a system identification problem on the parameters of the Bouc-Wen hysteresis model by utilizing GA, in which the crossover and mutation operators are adopted in between of the gene code strings of the same parameters, improving the convergence of the algorithm. On the other hand, Peng, Li, and Suzuki <sup>[36]</sup> successfully used PSO in the identification of the parameters of a nonlinear differential hysteresis model with pinching, based on the Duhem hysteresis operator.

In general, both GA and PSO have been proven to perform well in most engineering problems, but several researchers <sup>[37-38]</sup> have reported that PSO has been much more efficient than GA in terms of both, quality of the solutions, and simplicity of implementation.

**CHAPTER 3**  
**HEURISTIC OPTIMIZATION METHODS**

**3.1. Genetic Algorithms (GA)**

As explained in section 2.4, characterizing the parametric values of a mathematical model is some times a challenging problem due to the relative complexity of the functional forms of the models often used, especially in nonlinear problems. A possible solution is the application of an optimization strategy to determine the specific parametric values that best fit the observed behavior. In general, a parameter estimation method is an iterative process that takes a known input  $u(t)$  of a system, evaluates it into a mathematical model with a starting set of parameter values, compares its result with the trial data (thus obtaining a value of error between them), and then modifies the values of the model parameters, expecting the new set to get closer to the optimum solution. This process is summarized in Fig. 3.1 <sup>[15]</sup>.



**Figure 3.1.** Schematic of a parameter estimation method

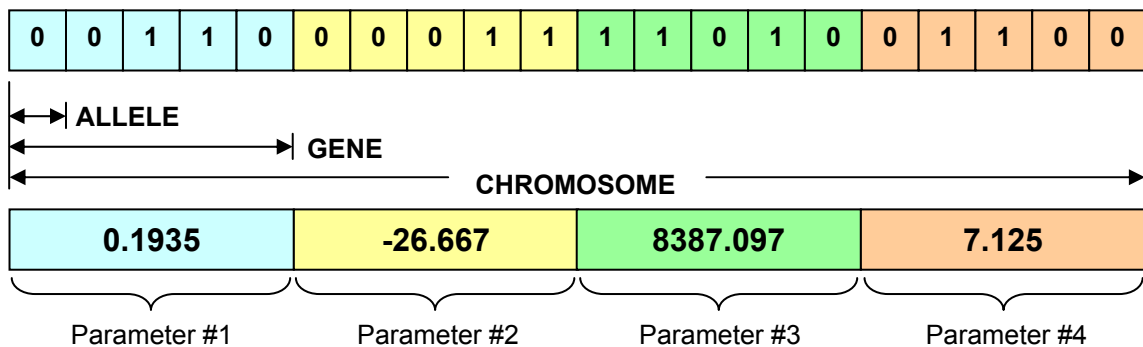
The complexity and high dimensionality of some models lead to the use of a heuristic search method. For those problems, Genetic Algorithms (GA) has proven to be a suitable optimization tool for a wide selection of problems.

This method was developed by John Holland <sup>[39]</sup> over the course of the 1960's and 1970's and finally popularized by one of his students, David Goldberg, who was able to solve a difficult problem involving the control of a gas-pipeline transmission for his dissertation <sup>[40]</sup>. Some of the advantages of a GA include that it <sup>[41]</sup>:

- Optimizes with continuous or discrete parameters,
- Doesn't require derivative information,
- Simultaneously searches from a wide sampling of the cost surface,
- Deals with a large number of parameters,
- Is well suited for parallel computers,
- Optimizes parameters with extremely difficult cost surfaces; it can jump out of a local minimum,
- Provides a list of optimum parameters, not just a single solution,
- May encode the parameters so that the optimization is done with the encoded parameters, and
- Works with numerically generated data, experimental data, or analytical functions.

GAs may be also used to identify an initial guess for calculus-based optimization techniques, in order to locally refine the search.

GAs have been a breakthrough to optimization theory because of their versatility, performance, and ease of use. The GA method is based on the ideas associated with the natural evolution of species, in which the “fittest” organisms in a population survive, passing their genes to the next generation, and so forth. Basically, the numerical version of this mechanism is a population composed of a large set of *chromosomes* (set of parameters), which generate results from a parametric mathematical model. These results are then compared to an objective function, which is usually the Mean Square Error (MSE) function, in order to determine the fitness of a particular chromosome. This yields a set of *fitness* values, which are used to sort the population, eliminate the badly-fitted individuals, mate the best-fitted chromosomes, and then propagate the “good” *genes* (parameter combinations) to the upcoming generation. This process may be repeated a fixed number of generations or indefinitely, until a certain error threshold is attained. There are two versions of GA: one with binary encoding (all parameters are encoded by long strings of ones and zeros), and another one with floating-point-number parameters (“Differential Evolution Algorithm”, or “Real-Coded GA”). An example of these two versions is shown in Fig. 3.2, for a search space with 4 parameters.



**Figure 3.2.** The two versions of GA chromosome representation; upper, binary representation, with 5 bits for each gene (parameter); lower, real-coded representation

In the binary-coded representation, a population of several random candidate solutions (chromosomes) is first generated, where all parameters are encoded into strings of ones and zeros. The algorithm works in this “encoded environment” (genotype space) while modifying the population. Then, it must decode each parameter in each chromosome into real numbers (phenotype space) in order to evaluate them with a suitable fitness function. This decoding may be carried out by using Eq. (3.1) [42].

$$decode_i(ub_i, lb_i, L_i, P_i) = \frac{(ub_i - lb_i) \cdot d(P_i)}{(2^{L_i} - 1)} + lb_i, \quad (3.1)$$

where:

$ub_i$  = upper bound of parameter  $i$ ,

$lb_i$  = lower bound of parameter  $i$ ,

$P_i$  = binary string of parameter  $i$ ,

$d(P_i)$  = decimal conversion of binary string  $P_i$ ,

$L_i$  = length of binary string of parameter  $i$ .

In Fig. 3.2, the values presented in the example of a real-coded GA, correspond to the decoded values of each binary string in the binary GA. The decoding was performed assuming the values given by Table 3.1.

Property	Parameter #1	Parameter #2	Parameter #3	Parameter #4
Upper bound (ub)	1	40	10000	20
Lower bound (lb)	0	-40	0	1
String length (L)	5	5	5	5

**Table 3.1.** Parameters properties for decoding the binary chromosome of Fig. 3.2

The real-coded representation works in a similar way as in the binary version, but no conversions are done. Instead, it works just in the phenotype space, so this method is usually much easier to implement.

Both versions perform the same genetic operations, but with different mechanisms. The operations are, in order of occurrence: (1) initial population generation (generates randomly a set of chromosomes), (2) fitness evaluation (determines how fit is each of the chromosomes to the objective function), (3) selection (determines pairs of chromosomes for mating/combination, either depending statistically on their fitness values, ranking, or tournament results), (4) crossover (for each pair of parents, it sets a parameter to be combined), (5) mating (combines the parameters determined by the crossover stage and swaps a certain number of parameters between each of the selected parents), and (6) mutation (randomly changes the value of a certain amount of parameters in the new offspring; this mutation version is called “uniform mutation”). Usually the best chromosome is allowed to propagate to the next generation unchanged by mutation. Such an approach is called an “elitist strategy”. Fig. 3.3 summarizes the main operations of the algorithm, where Fig. 3.4 shows it in pseudo-code. The aforementioned steps are explained next in more detail.

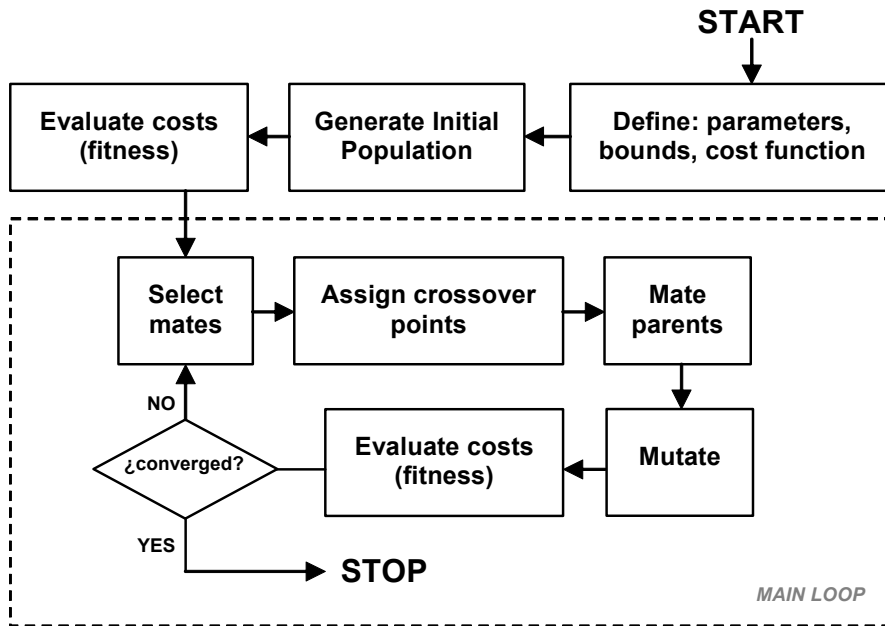
The initial population is generated first by defining the search space (i.e. the bounds  $ub$  and  $lb$  for each parameter  $j$ ), and then using Eq. (3.2).

$$p_{ij}(t) = (ub_j - lb_j) \cdot \text{rand}(0,1) + lb_j \quad , \quad (3.2)$$

where:

$p_{ij}(t)$  = parameter  $j$  of chromosome  $i$  at iteration  $t$ ,

$\text{rand}(0,1)$  = uniform random number between 0 and 1.



**Figure 3.3.** Main steps for a typical Genetic Algorithm (GA)

```

Procedure GA {
t=0;
Create_Population(P(t));
Evaluate_Fitness(P(t));
while (not done) {
    t=t+1;
    M(t)=Select_Mates(P(t));
    Assign_Xover(M(t));
    Offspring(t)=Mate(M(t));
    P(t)=Add_to_Population(Offspring(t));
    Mutate(P(t));
    Evaluate_Fitness(P(t));
    Select_Survivors(P(t));
}
}
  
```

**Figure 3.4.** Pseudo-code of a typical Genetic Algorithm (GA)

The fitness of a candidate solution (chromosome) is evaluated by a suitable error mathematical expression, such as the Ordinary Least Squares (OLS) error function, given by Eq. (3.3), with the goal of finding a chromosome with the lowest possible value of OLS.

$$\text{OLS}_i(t) = \sum_{k=1}^N \left[ \hat{f}_k(p_1, p_2, \dots, p_n; u(t)) - F_k(u(t)) \right]^2, \quad (3.3)$$

where:

$\text{OLS}_i(t)$  = ordinary least square error of chromosome  $i$  at iteration  $t$ ,

$\hat{f}_k$  = model output data point number  $k$ ,

$F_k$  = experimental output data point number  $k$ ,

$N$  = total number of data points (samples) to be evaluated.

The parents selection may be carried out by several ways. The most popular techniques are: fitness-based, ranking-based, and tournament selection. In the fitness-based selection, a probability  $P_i$  of being chosen as a parent for recombination is calculated for each chromosome (at iteration  $t$ ) depending on their fitness value, by using Eq. (3.4).

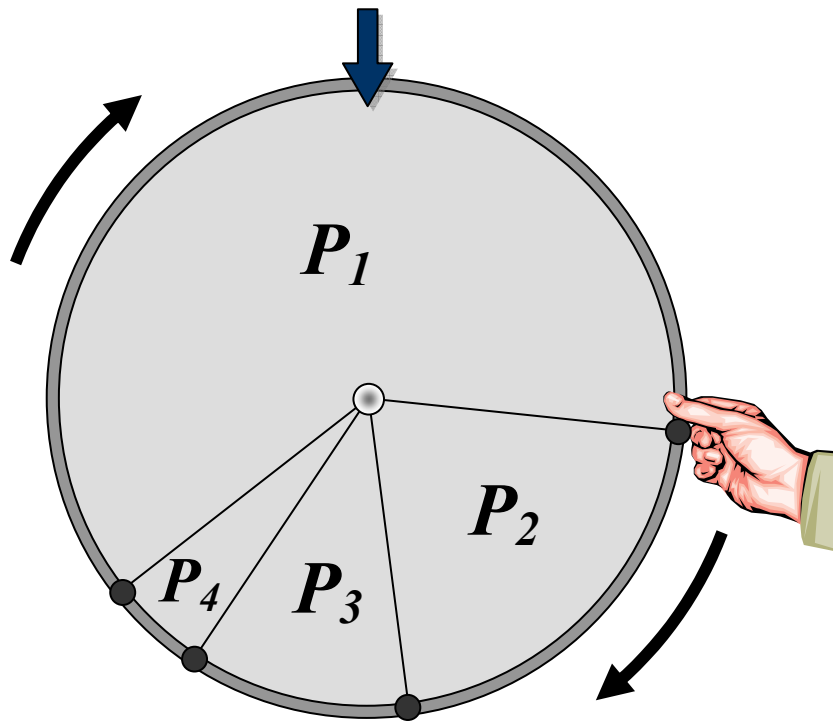
$$P_i(t) = \frac{|C_i(t)|}{\left| \sum_{q=1}^M C_q(t) \right|}, \quad (3.4)$$

where  $C_i(t) = [\text{Fitness}_i(t) - \text{Fitness}_{M+1}(t)]$ ,  $\text{Fitness}_i(t)$  is the fitness of chromosome  $i$  at iteration  $t$ , and  $M$  is the total number of chromosomes to keep from the population, for the general case of discarding a specific amount of “badly-fitted” chromosomes at each iteration. In the ranking-based selection, however, the probability of being chosen is

calculated from the ranking position, where the chromosome with the best fitness (lowest error) is ranked #1, and so forth. The probability in this case is given by Eq. (3.5).

$$P_i(t) = \frac{M - i + 1}{\sum_{r=1}^M r} = \frac{2 \cdot (M - i + 1)}{M(1 + M)} . \quad (3.5)$$

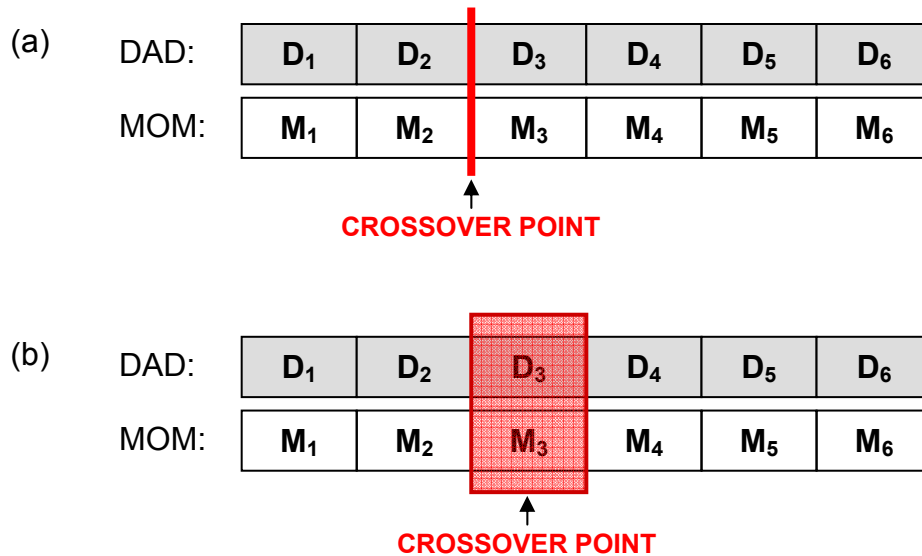
In both, fitness-based, and ranking-based selection, a technique called “roulette wheel” is used in order to pick pairs of parents for recombination, which consists in an analogy of spinning a roulette wheel which has  $M$  sections, each of them with a size corresponding to the probability determined by either Eq. (3.4) or (3.5), as shown in Fig. 3.5 for a population where 4 individuals are to be kept. After spinning the wheel a first time, an individual will be randomly picked as the first parent. Similarly, the second parent is then picked after spinning the wheel a second time.



**Figure 3.5.** Roulette wheel analogy, as a parent selection strategy

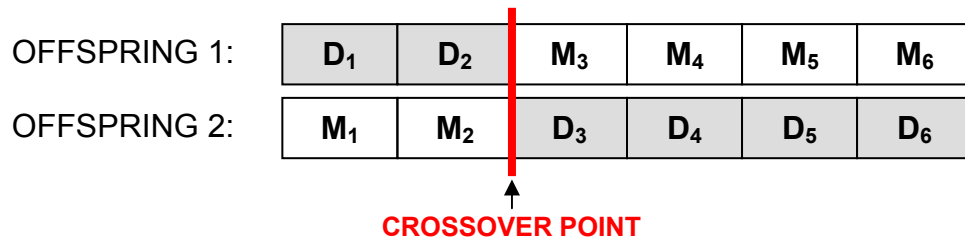
The tournament selection is the easiest way of obtaining couples from the population. It consists of simply selecting a random subset (of  $q$  individuals, where  $q$  is known as “tournament size”) from the mating pool, and the best one (with the lowest value of error) is selected as the first parent. Then, the process is repeated in order to obtain the second parent.

The crossover operator is utilized to determine swapping points within the chromosomes of the selected couples, with the purpose of using this information for the mating process. Particularly, in the real-coded chromosome representation, the crossover point may be selected between genes, or it may be over a gene, as shown in Fig. 3.6. Notice that multiple crossover points may be selected for each couple, although single point crossover is often employed.



**Figure 3.6.** Two types of crossover points; (a) between genes, (b) over a gene

The mating process is used to recombine each couple in order to generate offsprings. In this step, the genes at one side of the crossover point selected for a couple are swapped, thus creating a new pair of chromosomes. This process is shown in Fig. 3.7.

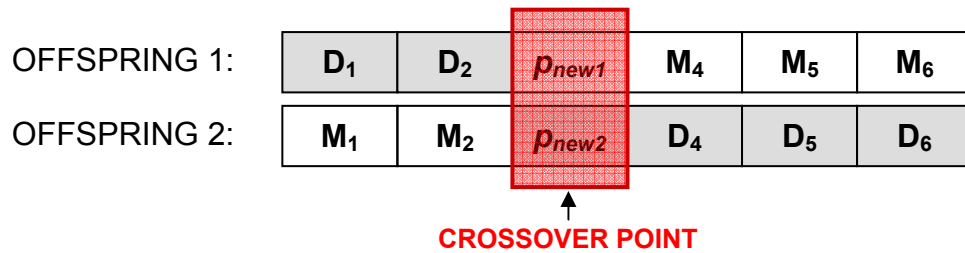


**Figure 3.7.** Mating process, when the crossover point is between genes

In the case where the crossover points are implemented to lie over the  $i$ -th gene in a couple, the process is similar as in Fig. 3.7 for all other genes. The crossover genes are then blended using the generalized Michalewicz<sup>[43]</sup> crossover method (Eq. (3.6)), which performs a weighted average on them, having the capacity of extending the new values out of the range defined by both genes at the crossover point in a proportion defined by the optimization parameter  $\alpha$ . This type of crossover method is known as BLX- $\alpha$ . The procedure is shown in Fig. 3.8.

$$\begin{aligned} p_{new1} &= D_i - \alpha \cdot \eta (D_i - M_i) \quad , \\ p_{new2} &= M_i + \alpha \cdot \eta (D_i - M_i) \quad , \end{aligned} \quad (3.6)$$

where  $\eta = \text{rand}(0,1)$ , a uniform random number between 0 and 1, in which the same value must be used for both expressions.



**Figure 3.8.** Mating process, when the crossover point lies over a gene

After the mating process, mutation is applied to some genes in the whole new population, in order to avoid letting the algorithm converge to suboptimal solutions, thus being able to jump out of these local minima and keep searching for the global optimum. It may be either applied just to genes of the newly created offspring, or to both, the population of parent chromosomes and offspring. There can be mutation of a specific percentage of genes of the population, or a mutation probability may be set in order to determine, for each gene, if mutation is going to be performed. The value to be set for a specific mutated gene can be either a uniform random number within the bounds defined for it (uniform mutation), or a number around the current value, at a distance defined by a mutation amount factor, the parameter bounds, and a Gaussian normally distributed random number (Gaussian mutation).

In the final steps of GA, the fitnesses of the new chromosomes in the population (offspring and chromosomes with mutated genes) are calculated, followed by a process of survival, in order to keep a certain amount of the best performing chromosomes. The survival can be either by using a  $(\mu, \nu)$  strategy (where  $\nu$  newly-generated chromosomes substitute the  $\nu$  worst performing ones from the parent population of size  $\mu$ ), or a  $(\mu+\nu)$  strategy (where the survivors are the  $\mu$  best performing chromosomes from a pool composed by the parent population of size  $\mu$ , and the  $\nu$  newly-generated ones). The process is then repeated (from the selection step) until a certain amount of iterations or a fitness threshold is reached.

### 3.2. Particle Swarm Optimization (PSO)

The Particle Swarm Optimization (PSO) is an optimization technique for continuous nonlinear functions. It was originally created for the simulation of a simplified social model, such as bird flocking or fish schooling, where a set of individuals (particles) interact in order to find an optimal solution for either maximization or minimization problems. It was first introduced by James Kennedy and Russ Everhart in 1995<sup>[44]</sup>. However, there are many variations of this approach, each created by different researchers in order to improve the search performance for specific problems. Shi and Eberhart<sup>[45]</sup> have shown that PSO searches wide areas effectively, but usually lacks precision for local search. Their suggestion for solving this issue is to introduce an inertia factor ( $\Omega$ ) in the standard equation, that adjusts the velocity correction over time, gradually concentrating the algorithm into a local search, as shown in Eq. (3.7).

$$\begin{aligned} \text{a) } v_{id} &= \Omega \cdot v_{id} + \varphi_1 \cdot \text{rand}(0,1) \cdot (p_{id} - x_{id}) + \varphi_2 \cdot \text{rand}(0,1) \cdot (p_{gd} - x_{id}) , \\ \text{b) } x_{id}' &= x_{id} + v_{id} , \end{aligned} \quad (3.7)$$

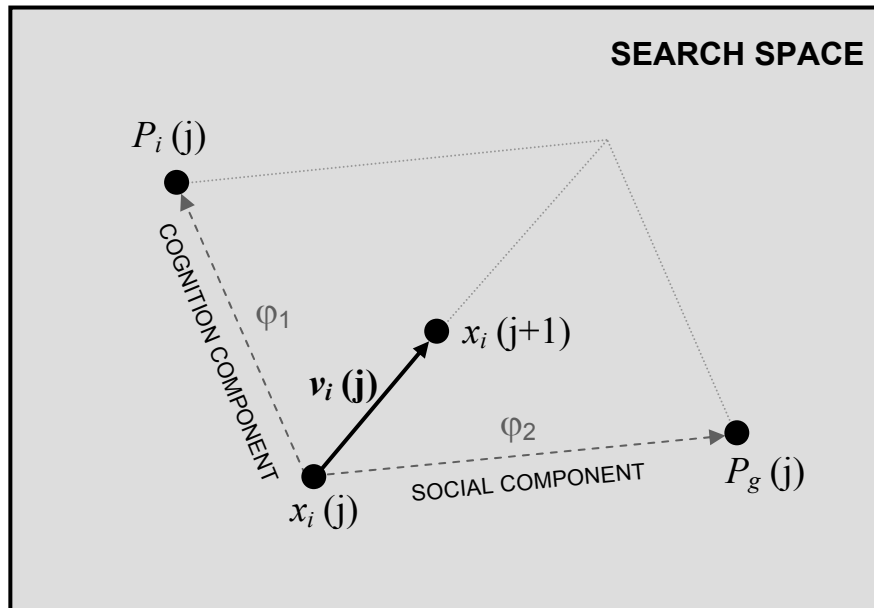
where  $i$  is the particle index,  $d$  is the dimension index (parameter),  $g$  stands for “global”,  $v$  is the velocity (correction),  $\varphi_1$  and  $\varphi_2$  determine the relative influence of the “cognition” and “social” components of the velocity,  $p$  is the location of the previous best position,  $x$  is the current particle position, and  $\text{rand}(0,1)$  is a uniform random number between 0 and 1. Maurice Clerc<sup>[46]</sup> introduced a so-called “constriction factor” ( $K$ ), that improves the control of the velocities, given by Eq. (3.8).

$$K = \frac{2}{\left| 2 - \varphi - \sqrt{\varphi^2 - 4\varphi} \right|} , \quad (3.8)$$

where  $\varphi = \varphi_1 + \varphi_2$ ;  $\varphi > 4$ . The velocity correction equation for the PSO is given by Eq. (3.9).

$$v_{id} = K \cdot \left( v_{id} + \varphi_1 \cdot \text{rand}(0,1) \cdot (p_{id} - x_{id}) + \varphi_2 \cdot \text{rand}(0,1) \cdot (p_{gd} - x_{id}) \right) . \quad (3.9)$$

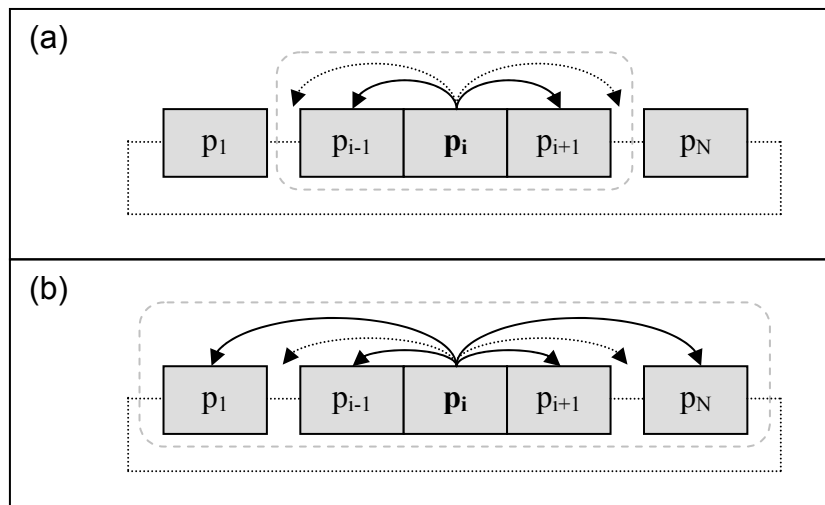
From Eqs. (3.7) to (3.9), it can be seen that PSO moves each particle of the swarm (population) in a direction determined by the current particle's best and the global best, each weighted by the cognition and social constants ( $\varphi_1$  and  $\varphi_2$ , respectively), as can be seen in Fig. 3.9.



**Figure 3.9.** PSO method of particle position correction at  $j$ -th iteration

The main advantage of PSO is that it is simple to implement, as compared to other evolutionary computation optimization techniques and generally has an outstanding performance. For example, for some popular testing functions <sup>[47]</sup>, like the Sphere (De Jong's F1), Rosenbrock, generalized Rastrigin, generalized Griewank, and Schaffer's F6 functions, success rates of 100% and low numbers of function evaluations may be easily obtained.

PSO's can have different neighborhood topologies. A particle's neighborhood is defined as a set of particles that influence an individual particle movement through the search space. It defines what is considered as  $P_g$  (global best position) for a specific particle being evaluated. The larger the neighborhood size, the more particles included in a particular particle's neighborhood <sup>[47]</sup>. There are two basic topologies for PSO: the ring topology and the star (or global) topology. In the ring topology, the particles are thought to be held in a wrap-around array, with each particle's neighborhood consisting of themselves, the particle to their left, and the particle to their right. The number of particles to include in the neighborhood set can expand or contract, adding the particles that are two positions away, three positions away, etc., thus increasing the neighborhood size. If the neighborhood set includes every single particle in the swarm, then the neighborhood is "fully-connected". A fully-connected neighborhood is said to be in a star (or global) topology. Fig. 3.10 shows these two types of topology for a swarm of  $N$  particles, where  $p_1$  to  $p_N$  are the individual particles, and  $p_i$  is the current particle to be analyzed.



**Figure 3.10.** PSO neighborhood topology types;  
(a) Ring topology, (b) Star topology (global neighborhood)

There are two ways of determining the global best ( $P_g$ ) position. The first is to only update  $P_g$  after every iteration. The particles are all updated in parallel, and only then a  $P_g$  is determined. This method is called a “synchronous” update. The second method is to determine the  $P_g$  after each particle has been updated. This form is called an “asynchronous” update. Asynchronous updates tend to find solutions quicker (in terms of number of iterations), due to their ability to use a newly found  $P_g$  in subsequent particle updates immediately in the same iteration, instead of having to wait for the next iteration, as is done in synchronous updates. This advantage usually makes asynchronous updates a better choice for a standard PSO.

### **3.2.1. PSO: The Basic Algorithm**

As it was mentioned in section 3.2, there are two methods of velocity update in PSO: synchronous, and asynchronous. The pseudo-codes describing the baseline algorithms for these PSO's are shown in Fig. 3.11 and 3.12, respectively. Notice that the synchronous method is easier to implement, since the algorithm works with blocks of data (when all particle positions have been just updated), after which the global best ( $P_g$ ) is determined. The asynchronous method is somewhat more difficult to implement, as the algorithm must evaluate the particles one by one, where  $P_g$  is updated as soon as a particle is determined to be better than the current global best. As was mentioned previously, the asynchronous method generally provides a better performance, and so it is the usually recommended update method.

```

Procedure PSO_Sync {
Initialize x; /* create swarm */
while (t <= number_of_iterations){
    Evaluate(x_fitness);
    P = best x's so far;
    Pg = best P so far;
    Calculate v;
    x = x + v;
    t = t + 1; }
}

```

**Figure 3.11.** PSO pseudo-code for synchronous implementation

```

Procedure PSO_Async {
Initialize x; /* swarm of size S */
Evaluate(x_fitness);
P = x;
Pg = best_P;
t = 1;
while (t <= number_of_iterations){
    for (i=1, i<=S, i++){
        /* evaluate every particle */
        Calculate vi;
        xi = xi + vi;
        Evaluate(xi_fitness);
        Pi = best xi so far;
        Pg = best P so far; }
    t = t + 1; }
}

```

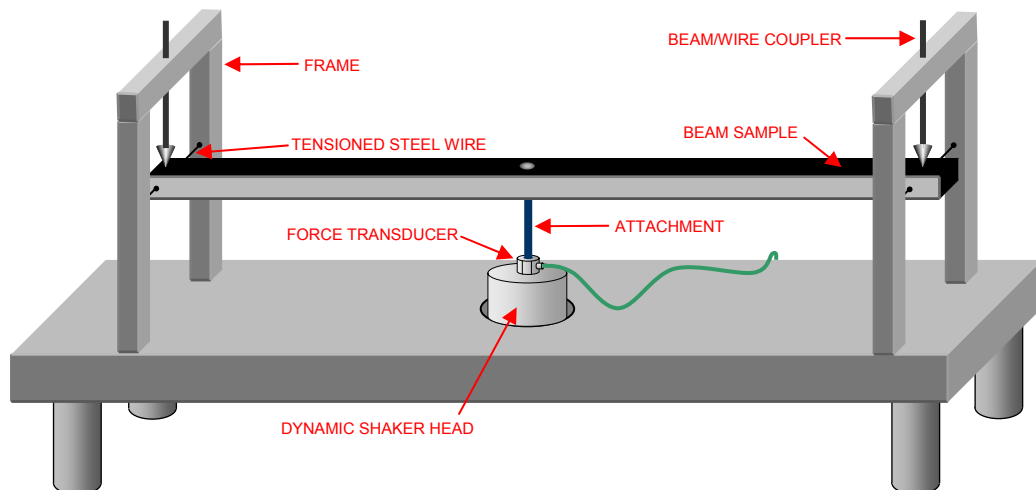
**Figure 3.12.** PSO pseudo-code for asynchronous implementation

## CHAPTER 4

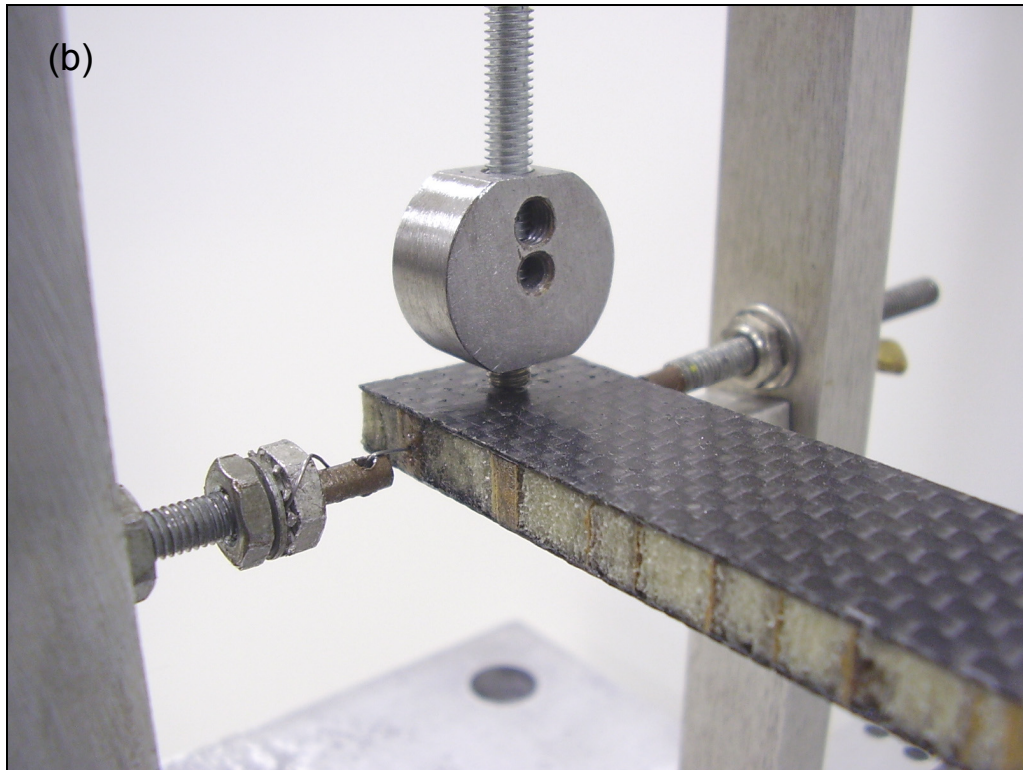
### MATHEMATICAL MODEL DEVELOPMENT

#### 4.1. Experimental Setup

In order to develop a suitable mathematical model, able to predict the dynamic response of honeycomb sandwich composites, observations of individual experimental hysteresis loops were first performed for numerous beam samples of these materials. These experimental hysteresis loops were obtained by constructing a customized test rig that allows the mounting of the beam samples with simply supported boundary conditions, and mid-point excitation. A concept drawing and pictures of the test rig are shown in Fig. 4.1 and 4.2, respectively.

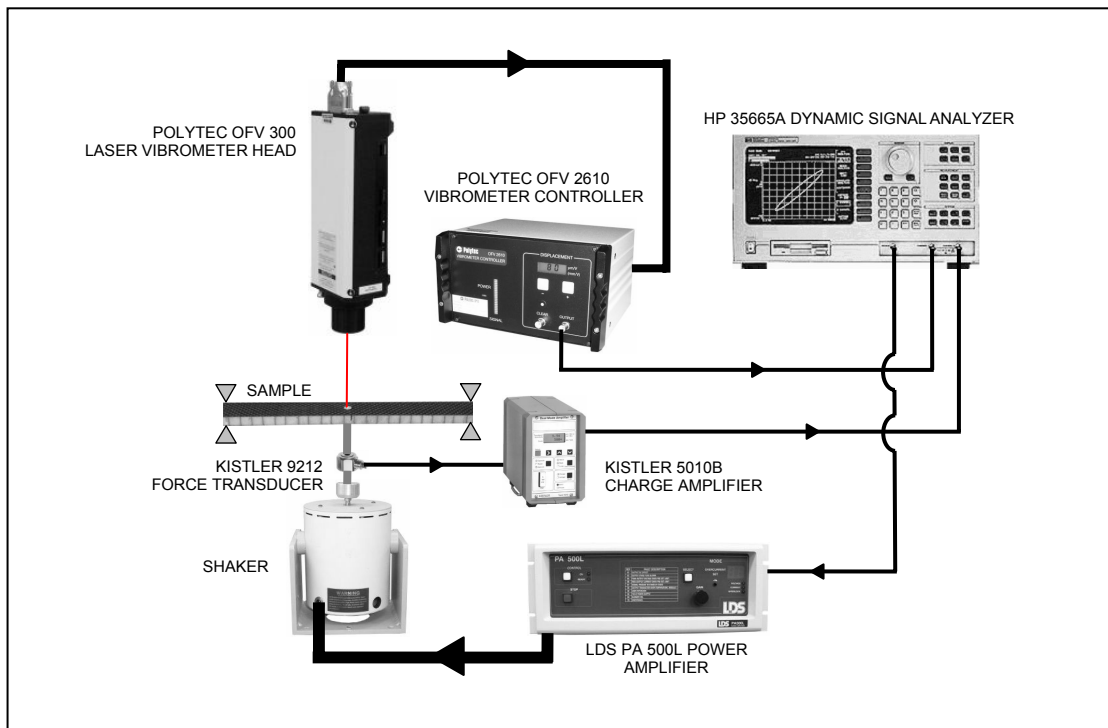


**Figure 4.1.** Concept drawing of the test rig for hysteresis loops generation

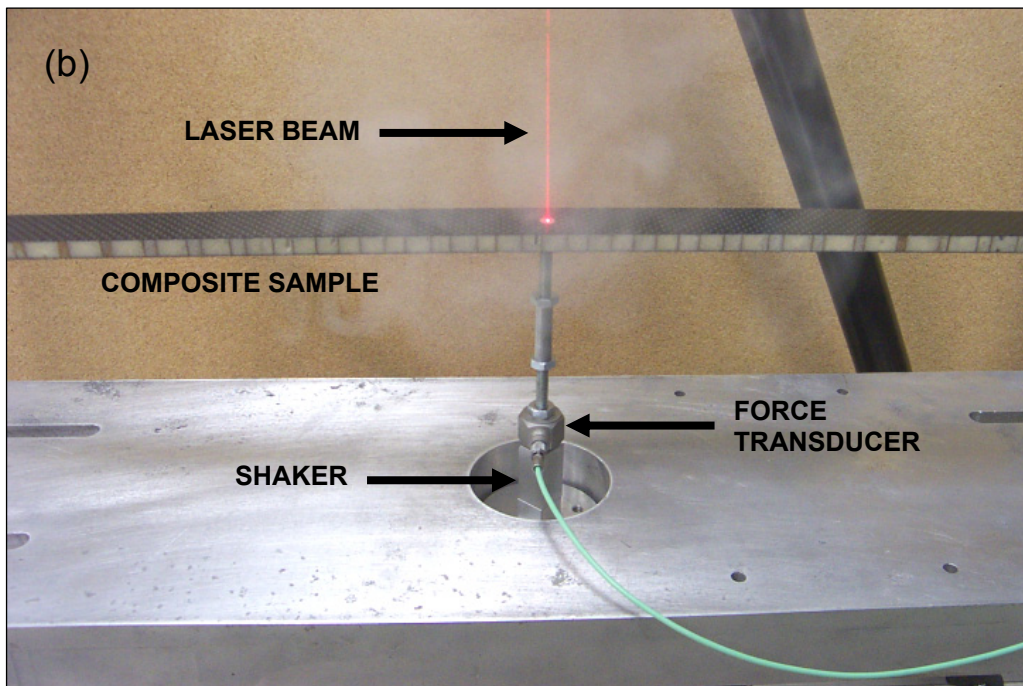
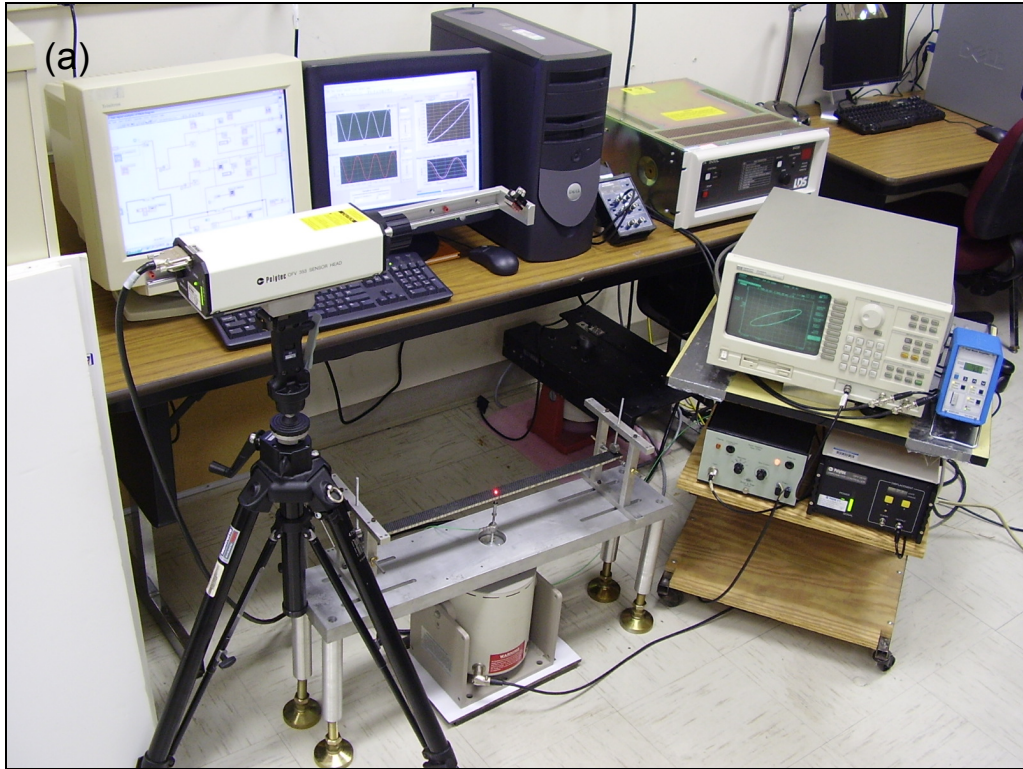


**Figure 4.2.** Pictures of the test rig for hysteresis loops generation;  
(a) test rig, (b) wire attachment detail

Ni, Ko, and Wong <sup>[12]</sup> have performed system identification experiments with hysteretic systems, where the displacement-controlled tests were shown to have a much better accuracy than the force-controlled ones. Accordingly, the experimental tests performed in this work consist of a simply-supported (by steel wires at both ends) beam sample of the composite material, excited by a pure sinusoidal displacement wave with a dynamic shaker. This applied displacement is measured at the mid-point of the beam with a laser displacement vibrometer, and the resulting force is measured by a force transducer positioned in-line between the shaker contact and the test sample. The coupling is attained by using a rigid point connector, bonded to the beam with Cyanoacrylate. The signals are then collected using a dynamic signal analyzer, and also alternatively using a PC with a DAQ card and National Instrument's LabView v8.0. A schematic and a picture of the complete experimental setup are shown in Fig. 4.3 and 4.4, respectively.



**Figure 4.3.** Experimental setup schematic, for hysteresis loops generation



**Figure 4.4.** Pictures of the experimental setup for hysteresis loops generation; (a) complete setup, (b) detail of the excitation/measurement contact point

A very important experimental consideration is explained as follows. A charge-mode signal conditioning amplifier has a time constant (TC) determined by  $TC=C_r \cdot R_t$ , where  $C_r$ =range capacitor capacitance, and  $R_t$ =time constant resistor resistance. This time constant has a direct effect on the output of the amplifier, in relation to the expressions given by Eqs. (4.1) and (4.2), for its voltage transfer and phase responses, respectively.

$$\frac{V_{out}}{V_{in}} = \frac{2\pi f \cdot (TC)}{\sqrt{1+(2\pi f \cdot (TC))^2}} \quad , \quad (4.1)$$

$$\phi = \tan^{-1}\left(\frac{1}{2\pi f \cdot (TC)}\right) \quad . \quad (4.2)$$

The time constant has the same effect as a single-pole high-pass filter, whose cutoff frequency is defined by Eq. (4.3).

$$f_c = \frac{1}{2\pi \cdot (TC)} \quad . \quad (4.3)$$

According to Eqs. (4.1) to (4.3), the higher the time constant, the better response the charge amplifier will have, considering that the frequency range of operation increases (as it improves the low frequency response), the voltage transfer approaches to unity, and the phase approaches to zero. This last result (regarding the phase output) is critical for hysteresis analysis, as the width of the loops are directly related to the phase between both signals. So, any external source of signal phase shift may cause the experimental data to be incorrect, hence leading to erroneous system identification results. If the frequency of the signal is sufficiently high, these issues barely affect the response of the amplifier, but considering that the experimental analysis of hysteresis is usually

performed at relatively low frequencies, the time constant must be always set to the largest possible value.

## 4.2. Mathematical Model Baseline

There have been a number of earlier attempts at representing the dynamics of sandwich composites with nonlinear models, such as the Bouc/Wen hysteresis model [2]. However, the almost perfectly elliptical hysteresis loops that were obtained experimentally, along with a lack of consistent convergence in many of the system parameters for the nonlinear model, led to the conclusion that a simpler semi-empirical equivalent linear model would yield a much better representation of these materials behavior. A baseline model consisting of a linear 1-DOF mechanical system with an equivalent viscous damping has been used as a starting point. The differential equation for such system is shown in Eq. (4.4), where  $\xi$ =damping ratio,  $\omega_n$ =natural frequency of vibration, and  $M_{eq}$ =equivalent mass.

$$\ddot{x} + 2\xi\omega_n\dot{x} + \omega_n^2x = \frac{F(t)}{M_{eq}} \quad . \quad (4.4)$$

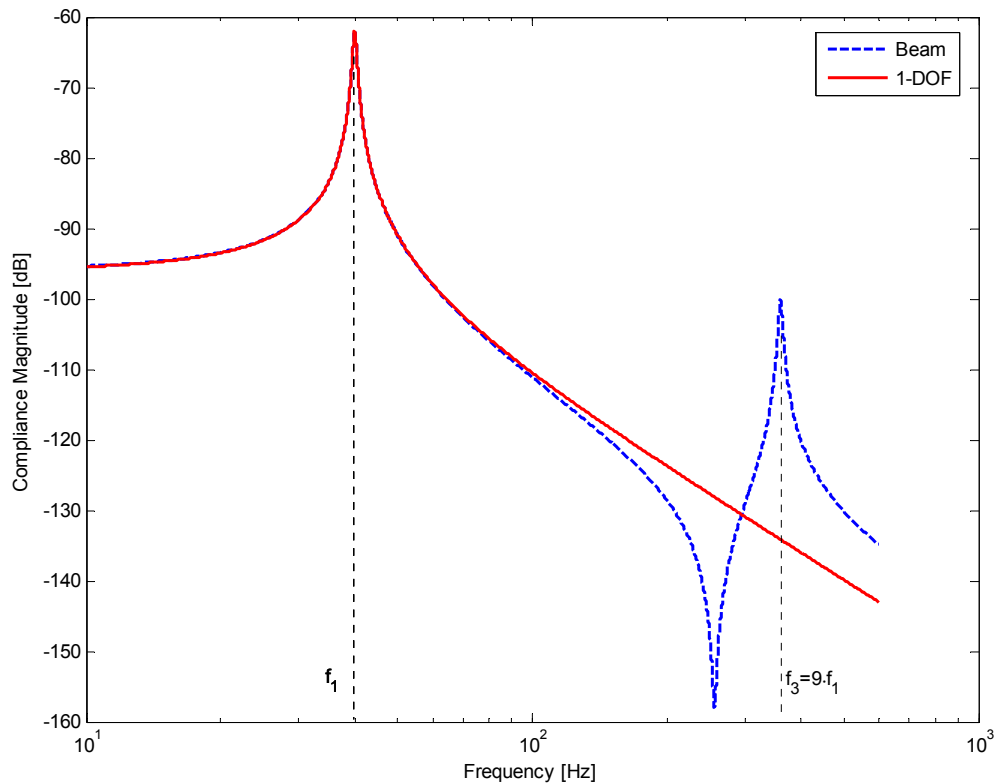
Since a sinusoidal displacement excitation is to be used in this study, the time functions for  $x$ ,  $\dot{x}$ , and  $\ddot{x}$  are given by Eq. (4.5), where  $X_0$ =amplitude, and  $\omega$ =excitation frequency.

$$\begin{aligned} x &= X_0 \cdot \sin(\omega t) \quad , \\ \dot{x} &= X_0\omega \cdot \cos(\omega t) \quad , \\ \ddot{x} &= -X_0\omega^2 \cdot \sin(\omega t) \quad . \end{aligned} \quad (4.5)$$

Substituting Eq. (4.5) into Eq. (4.4), a closed-form solution for  $F(t)$  is given by Eq. (4.6).

$$F(t) = M_{eq} \cdot X_0 \cdot \sqrt{(\omega_n^2 - \omega^2)^2 + (2\xi\omega_n \cdot \omega)^2} \cdot \sin\left(\omega t + \tan^{-1}\left(\frac{2\xi\omega_n \cdot \omega}{\omega_n^2 - \omega^2}\right)\right) \quad . \quad (4.6)$$

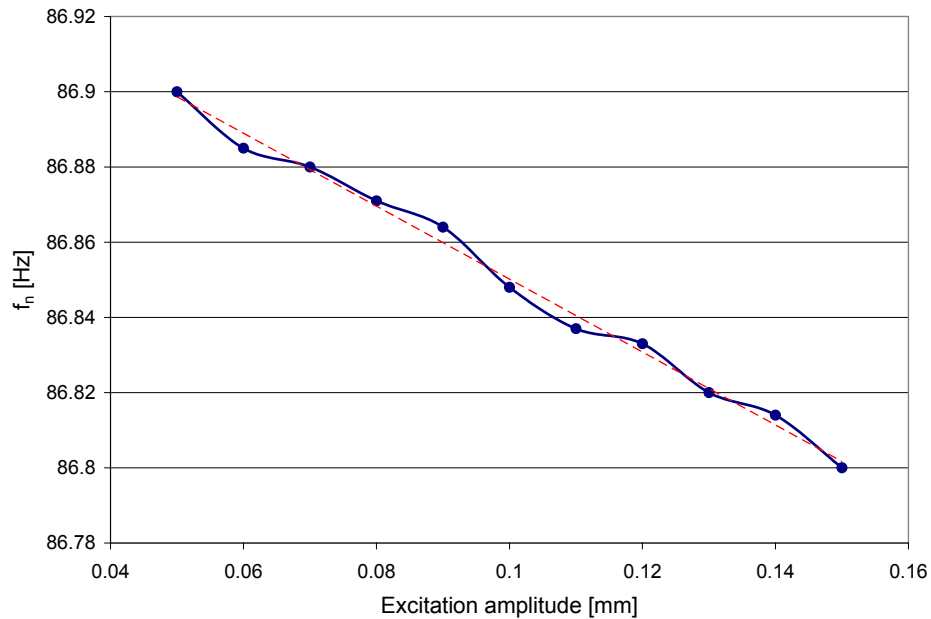
Additional tailoring of this expression to the observed experimental behavior has been performed, as explained in the following. It should be pointed out first that, since the experimental setup consists of a point excitation at the beam center, and the sample is simply-supported at both ends, the Euler's beam equation has been approximated into this equivalent 1-DOF equation. The mid-point excitation eliminates the presence of the second natural mode of vibration of the beam, thus having the first and third modes well separated, by a factor of nine. This allows the infinite-DOF system that characterizes a beam to be approximated into a 1-DOF system (around the first mode of vibration) with sufficient accuracy, as can be observed in the compliance FRF for this case, in Fig. 4.5.



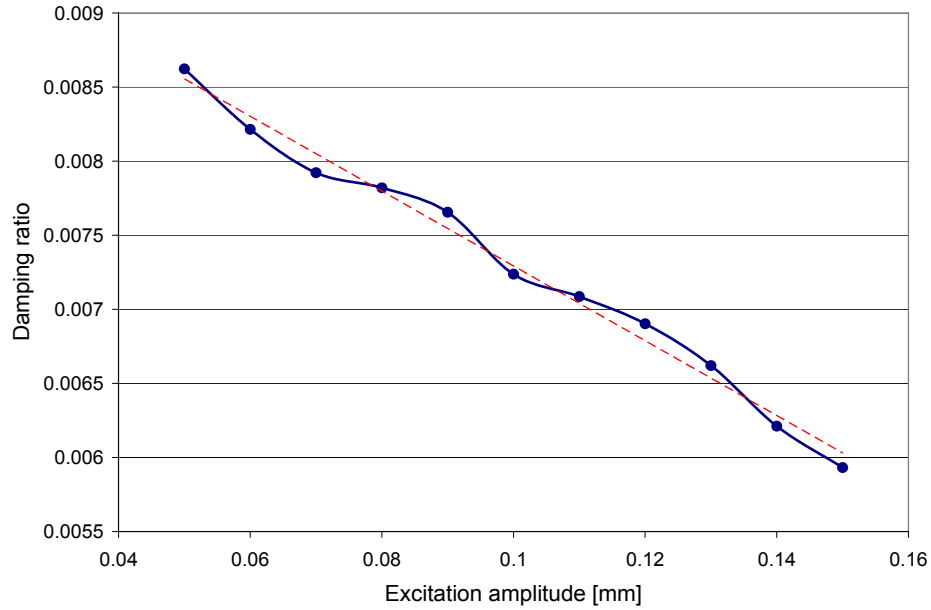
**Figure 4.5.** Compliance FRF comparison, between a mid-point excited beam, and an equivalent 1-DOF system

### 4.3. Preliminary Experimental Observations

As a first step in identifying appropriate modifications to the baseline model, several hysteresis loops have been obtained experimentally for different material samples, at different excitation frequencies and displacement amplitudes. Two unusual phenomena have been noted after the preliminary identification of the system parameters for individual loops, using an implementation of Genetic Algorithms, as it will be described in more detail in section 5.1. Both, the resonance frequency ( $\omega_n$ ) and the damping ratio ( $\xi$ ) are linear functions of the excitation amplitude, as can be seen in the graphs of Figs. 4.6 and 4.7, respectively, as an example of the results obtained for one of the composite beam samples to be studied. Similar behavior has been observed for several other samples as well.

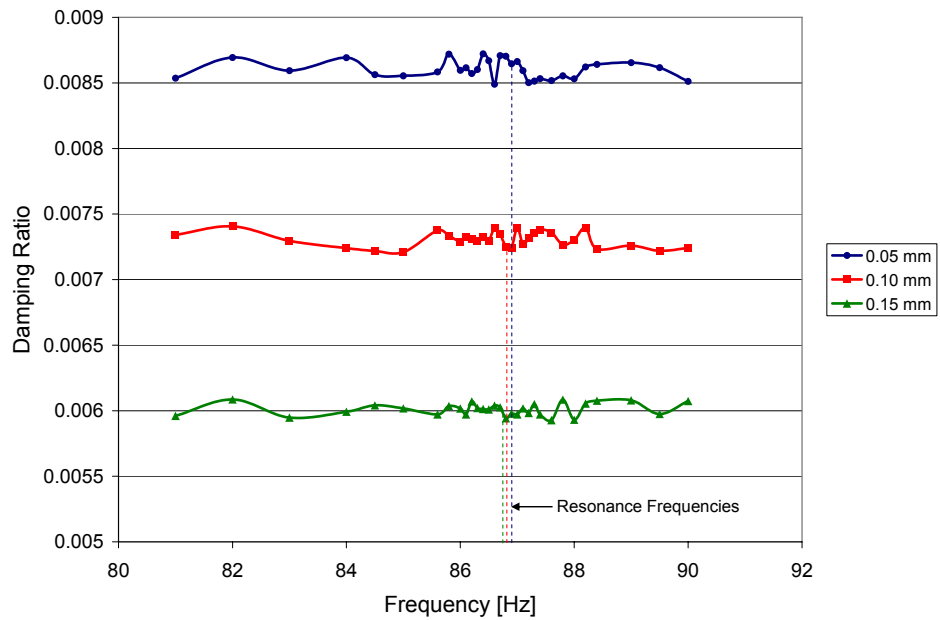


**Figure 4.6.** Observed behavior of a sandwich composite beam; plot of  $f_n = \omega_n / 2\pi$  vs.  $X_0$



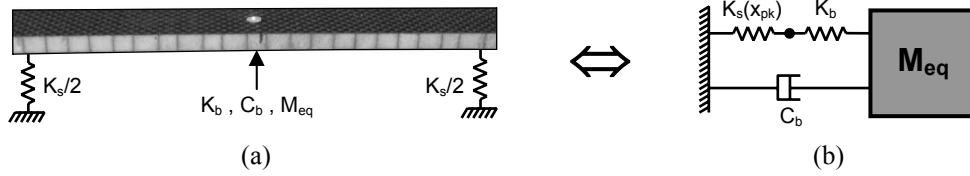
**Figure 4.7.** Observed behavior of a sandwich composite beam; plot of  $\xi$  vs.  $X_0$

The damping ratios at fixed excitation amplitudes, however, remained relatively constant for frequencies around the first resonance, as in the example shown in Fig. 4.8.



**Figure 4.8.** Observed behavior of a sandwich composite beam; plot of  $\xi$  vs.  $f$  at different excitation amplitudes

The probable cause of the amplitude dependence of the resonance frequency are the nonlinear effects, which arise from the flexing of the wire supports at both ends of the beam. This produces a decrease in the effective stiffness of the system. This effect is illustrated in Fig. 4.9, with both (a) the beam sample, and (b) an equivalent system consisting of a mass attached to two springs in series, and a viscous damper.



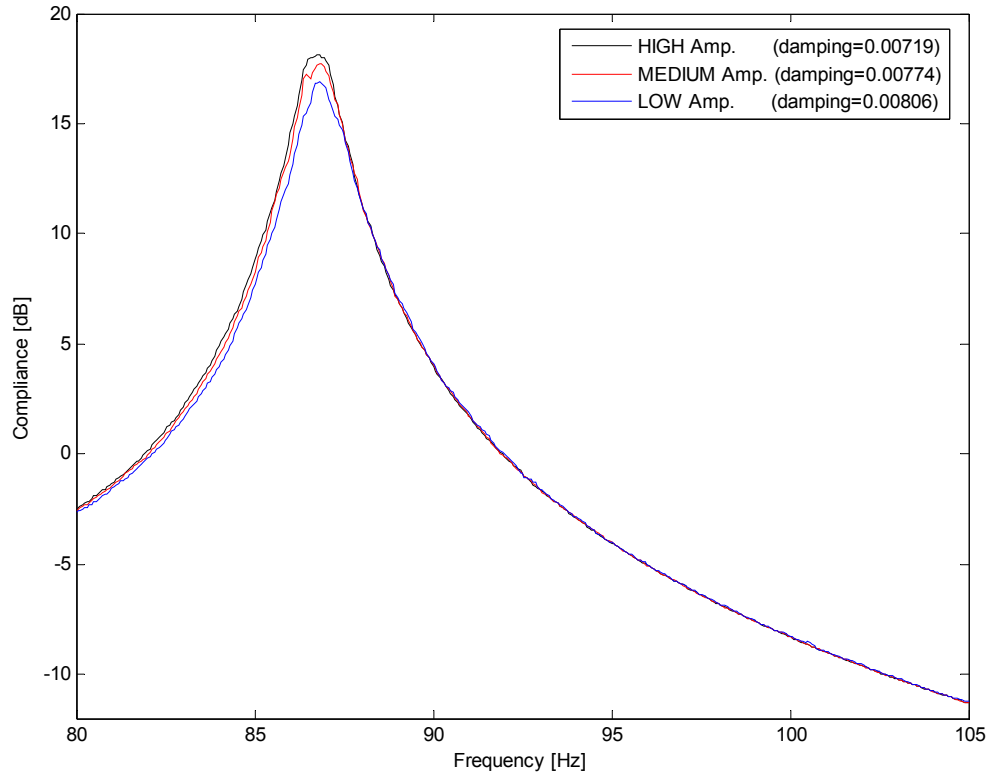
**Figure 4.9.** Equivalent experimental mechanical system, considering the stiffness of the supports

Then the total effective stiffness of the system is given by Eq. (4.7), where  $K_b$  is the beam equivalent stiffness,  $K_s$  is the supports nominal stiffness, and  $A$  is a stiffness decay factor.

$$K = \left( \frac{1}{K_b} + \frac{1}{K_s - A \cdot X_0} \right)^{-1} . \quad (4.7)$$

This expression for the stiffness leads to a behavior similar to that shown in Fig. 4.6.

As for the amplitude dependence of the damping ratio, this effect was further verified (by generating compliance frequency response plots of some beam samples) that, for increasing vibration amplitudes, the peaks of the first resonance of the system also increased in level, as in the example shown in Fig. 4.10, for one of the beam samples. Hence, according to the half-power method damping equation given by Eq. (2.7), the -3dB point frequencies are located progressively in narrower bandwidths, reducing the value of the damping ratio as the excitation level is increased.



**Figure 4.10.** Overlay plot example of the compliance frequency response of a sandwich composite beam sample around its first resonance, at different excitation amplitudes

From the behavior observed in Figs. 4.6 to 4.8, functional expressions for the damping ratio ( $\xi$ ) and the resonance frequency ( $\omega_n=2\pi f_n$ ) can be developed.

#### 4.4. Final Model

For the values of the damping ratio  $\xi$ , and for the resonance frequency  $\omega_n$ , linear functions in  $X_0$  will be used (Eq. (4.8) and (4.9), respectively, where  $S_\xi$  and  $S_\omega$  are referred as the slopes of these linear functions, respectively). As a result, the force response can be represented by the steady-state expression shown in Eq. (4.10), for a sinusoidal displacement excitation, as given by Eq. (4.5).

$$\xi(X_0) = \xi_0 + S_\xi \cdot X_0 \quad , \quad (4.8)$$

$$\omega_n(X_0) = \omega_0 + S_\omega \cdot X_0 \quad , \quad (4.9)$$

$$F(t) = M_{eq} \cdot x_{pk} \cdot \sqrt{\left(\omega_n(X_0)^2 - \omega^2\right)^2 + \left(2\xi(X_0) \cdot \omega_n(X_0) \cdot \omega\right)^2} \cdot \sin\left(\omega t + \tan^{-1}\left(\frac{2\xi(X_0) \cdot \omega_n(X_0) \cdot \omega}{\omega_n(X_0)^2 - \omega^2}\right)\right) \quad . \quad (4.10)$$

Therefore, the system parameters to be identified is the set composed by:  $(\xi_0, S_\xi, \omega_0, S_\omega)$ .

## CHAPTER 5

### IMPLEMENTATION AND RESULTS

#### 5.1. Optimization Algorithms Implementation

In order to identify the system parameters from the model given by Eqs. (4.8) to (4.10) for specific beam samples of honeycomb sandwich composite materials, computer programs have been written in Matlab R2006a, implementing customized codes for GA and PSO (as shown in Appendix A). It is expected that, in general, the search algorithms will find parametric solutions relatively close to each other for different runs of the computer codes, simultaneously for different excitation conditions. It should be pointed out that the equivalent mass of the system ( $M_{eq}$ ) is a 1-DOF approximation of the continuum of the simply-supported beams, which happens to be equal to the total mass of the samples being analyzed.

Some additional features were also implemented from the baseline characteristics of both algorithms, which greatly improved their search performance. First, a large initial population (at iteration 0) is initially created, in order to find good candidates for the upcoming iterations. For PSO, another feature is that when a “particle position update” attempts to go beyond the search bounds set by the user, the particle remains motionless only for those parameters (not set to  $x_{min}$  or  $x_{max}$ , as is more typically done), and  $V_{id}$  is set to zero for the next velocity calculation. In addition to the regular offsprings that are created in both algorithms, an additional set of candidate solutions (CS) is generated,

which consists of several different combinations of Gaussian-mutated versions of the current best partial solution (BPS), with the intention of performing an exhaustive local search around the BPS for an even better solution. The number of members of this group depends on an “add-on level” ( $L$ ) set by the user, which is the maximum number of parameters to be mutated in the BPS in order to create additional candidate solutions, as in the example shown in Table 5.1, for a search space of 4 parameters, and  $L=3$ .

Mutated parameters from the best partial solution			
$p_1$	$p_2$	$p_3$	$p_4$
X			
	X		
		X	
			X
X	X		
X		X	
X			X
	X	X	
	X		X
		X	X
X	X	X	
X	X		X
X		X	X
	X	X	X

LEVEL 1

LEVEL 2

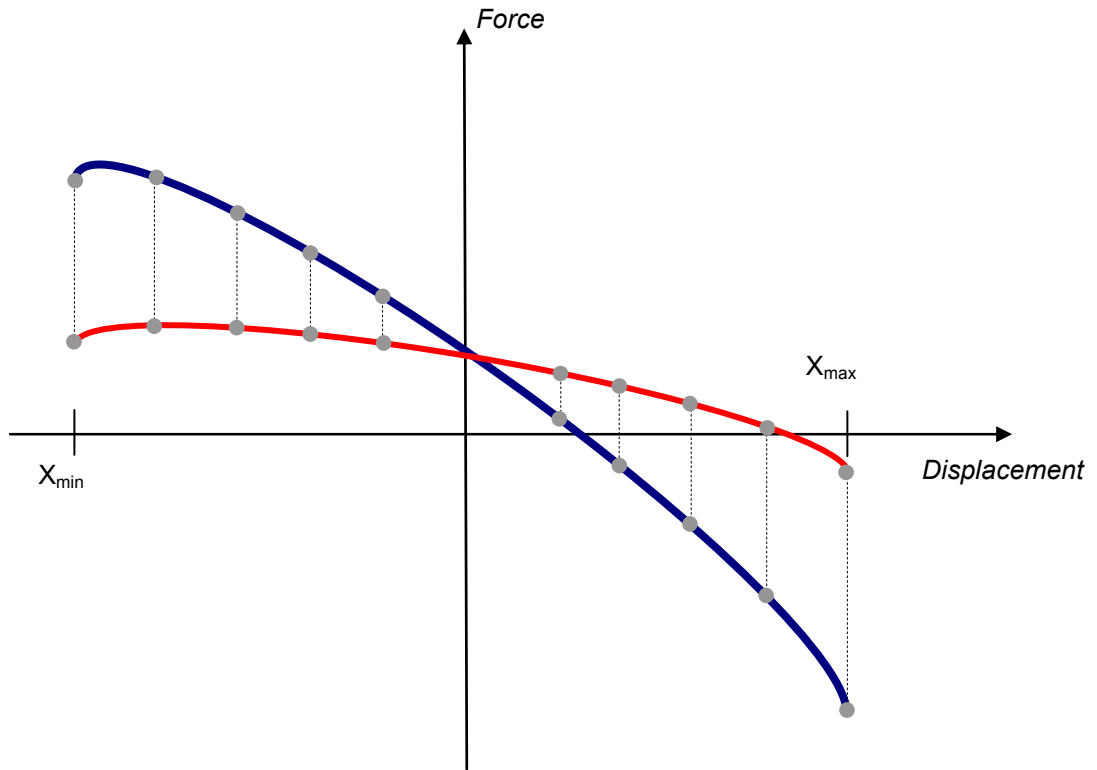
LEVEL 3

**Table 5.1.** Mutated parameters of the best partial solution (of 4 parameters,  $p_i$ , where  $i=1,2,3,4$ ), for the creation of an additional set of candidate solutions ( $L=3$ ) for local search

For a search space of size  $N$ , the number of added individuals is given by Eq. (5.1).

$$\text{AddPopNumber} = \sum_{i=1}^L \binom{N}{i} = \sum_{i=1}^L \frac{N!}{i!(N-i)!} . \quad (5.1)$$

In the extreme case, where  $L=N$  (as used in the present study), the result may be simplified as:  $\text{AddPopNumber}=2^N-1$  (i.e. for  $N=4 \Rightarrow \text{AddPopNumber}=15$ ). Finally, the method of error calculation between the trial and model was performed by using the standard Ordinary Least Squares (OLS) error function given by Eq. (3.3), comparing the difference between the force data for every displacement sample value (Fig. 5.1) in both, the top and bottom parts of the hysteresis loop. Interpolations are performed in order to enforce the requirement that the model must match the trial data sample points.



**Figure 5.1.** Comparison of hysteresis loops for error calculation (displacement loading);  
— trial, — model, • sample points, ..... error

The characteristics and optimization parameters in the PSO routine created were taken mainly from the recommendations of Carlisle and Dozier <sup>[47]</sup>, typical values used by other researchers, and some specific values considered appropriate for this problem.

These characteristics are:

- Asynchronous PSO
- Global topology
- $\varphi_1=\varphi_2=2.05$

As for the GA, several versions of it have been tested, but the best results were obtained for the following definitive implementation:

- Real-coded GA
- Rank-based parent selection
- Single-point BLX- $\alpha$  crossover (with  $\alpha=0.3$ )
- $(\mu + (\mu + 1))$  survival selection strategy
- Elitist strategy
- Mutation rate=5%
- Gaussian mutation amount=30% (amount allowed for the mutated parameters to move).

Both algorithms shared the following optimization parameters, which provided with sufficiently accurate results:

- Initial population/swarm=6000 (at iteration 0)
- Population/swarm size=500 (after iteration 0)

- Iterations=18
- Additional-population Gaussian mutation amount=20% (amount allowed for the mutated parameters of the additional population to move).
- Additional-population “add-on” level:  $L=4$

It should be noted that the main steps the GA takes at each iteration, are as follows:

1. From the parent population of size  $\mu$ , generate an offspring also of size  $\mu$ .
2. Generate a set of “AddPopNumber” chromosomes for the additional population, based on the best current solution available.
3. Identify the best chromosome from the additional population generated.
4. Select the best chromosomes from a pool of size  $(\mu+\mu+1)$ , composed of the parent population, the offspring, and the best performing chromosome from the additional population.

## **5.2. Benchmark Test of GA and PSO**

Before making use of the computer optimization programs with actual experimental data, computer-generated hysteresis loops (*Synth* set) have been created, at 40 different frequencies, and at 3 different excitation amplitudes each, making a total of 120 loops. The purpose of this is to carry out a benchmark test of the search performance of both PSO and GA with a known global solution. Also, the influence of Gaussian random noise present in the signals has been studied, in order to verify its effect on the optimization results.

The equivalent mass used to generate the *Synth* data set is  $M_{eq}=40e-3$  kg, with excitation amplitudes ( $X_0$ ) of 0.5, 1.0, and 1.5mm, and the excitation frequencies given in Table 5.2.

Excitation Frequencies [Hz]									
25	26	26.5	27	27.5	28	28.2	28.4	28.6	28.8
29	29.1	29.2	29.3	29.4	29.5	29.6	29.7	29.8	29.9
30	30.1	30.2	30.3	30.4	30.5	30.6	30.7	30.8	30.9
31	31.2	31.4	31.6	31.8	32	32.5	33	33.5	34

**Table 5.2.** Excitation frequencies of the computer-generated *Synth* hysteresis set

The system parameters used (global optimum) are shown in Table 5.3, where  $\omega_0=2\pi 30.1=189.1239$ , and  $S_\omega=2\pi(-200)=-1256.64$ .

$\xi_0$	$S_\xi$	$\omega_0$	$S_\omega$
0.0042	-0.4	189.1239	-1256.64

**Table 5.3.** System parameters used for generating the *Synth* hysteresis set

The data given in Table 5.3 correspond to the following constant values of resonance frequency ( $\omega_n$ ) and damping ratio ( $\xi$ ) of the general baseline model solution (Eq. (4.6)) at specific excitation amplitudes:

- At  $X_0=0.5$ mm :  $\omega_n=2\pi 30.0$  [rad/s] ;  $\xi=0.0040$
- At  $X_0=1.0$ mm :  $\omega_n=2\pi 29.9$  [rad/s] ;  $\xi=0.0038$
- At  $X_0=1.5$ mm :  $\omega_n=2\pi 29.8$  [rad/s] ;  $\xi=0.0036$

The noise levels used for this test (with respect of the peak values of force of each hysteresis loop) were: 0% (clean signals), 10%, 20% and 30%. Finally, the search space used (parameter bounds) is shown in Table 5.4.

Parameter	Min	Max
$\xi_0$	0	0.1
$S_\xi$	-10	10
$\omega_0$	100	500
$S_\omega$	-10000	10000

**Table 5.4.** Search space for the *Synth* hysteresis set

### 5.2.1. Benchmark Test Results

A total of 4 runs of each optimization routine were performed for each noise level case of the *Synth* hysteresis set. The parametric results for each case, along with their average values ( $\bar{x}$ ), standard deviations ( $\sigma$ ), and normalized standard deviations ( $\sigma/\bar{x}$ , for a relative comparison among the results of each parameter), are shown in Tables 5.5 to 5.8 for GA, and Tables 5.9 to 5.12 for PSO.

In order to determine the effect of the added Gaussian random noise on the optimization results, scatter plots describing the spread ranges of the results of each parameter, and for each optimization algorithm, are shown in Figs. 5.2 to 5.5, where the dot over each line represents the average solution, and the red dotted line represents the global optimum. Also, convergence plots (i.e. the fitness histories over all iterations) for both algorithms and at each noise level (overlying all 4 runs in each case, and with matching scales, for comparison purposes), are shown in Figs. 5.6 to 5.9. Some plots showing comparisons between trial hysteresis loops and model results (typical for both optimization techniques), are shown in Figs. 5.10 to 5.13, for all 3 excitation amplitudes, and all 4 noise levels. Finally, as a way of measuring the quality of the solutions of both optimization techniques utilized, the differences of their averages from the actual global optimum are compared in Table 5.13, for all 4 noise levels considered.

GA 0% noise	ACTUAL	RUN 1	RUN 2	RUN 3	RUN 4	$\bar{x}$	$\sigma$	$\sigma/\bar{x}$
$\xi_0$	<b>0.0042</b>	0.00413	0.004444	0.004306	0.003081	<b>0.00399</b>	0.00062	0.155336
$S_\xi$	<b>-0.4</b>	-0.33347	-0.58421	-0.42632	0.372055	<b>-0.24299</b>	0.42289	1.740382
$\omega_0$	<b>189.1239</b>	189.1282	189.1831	189.222	189.1089	<b>189.1605</b>	0.051628	0.000273
$S_\omega$	<b>-1256.64</b>	-1256.03	-1311.81	-1309.71	-1298.78	<b>-1294.08</b>	26.00358	0.020094

Table 5.5. Optimization results for the *Synth* data set; GA, 0% noise

GA 10% noise	ACTUAL	RUN 1	RUN 2	RUN 3	RUN 4	$\bar{x}$	$\sigma$	$\sigma/\bar{x}$
$\xi_0$	<b>0.0042</b>	0.00379	0.003934	0.004319	0.004104	<b>0.004037</b>	0.000227	0.056337
$S_\xi$	<b>-0.4</b>	-0.00494	-0.22136	-0.52046	-0.32666	<b>-0.26836</b>	0.214907	0.800830
$\omega_0$	<b>189.1239</b>	189.0764	189.1619	189.2238	189.1275	<b>189.1474</b>	0.061878	0.000327
$S_\omega$	<b>-1256.64</b>	-1199.93	-1303.86	-1327.24	-1256.3	<b>-1271.83</b>	56.29433	0.044262

Table 5.6. Optimization results for the *Synth* data set; GA, 10% noise

GA 20% noise	ACTUAL	RUN 1	RUN 2	RUN 3	RUN 4	$\bar{x}$	$\sigma$	$\sigma/\bar{x}$
$\xi_0$	<b>0.0042</b>	0.004398	0.004363	0.004103	0.004045	<b>0.004227</b>	0.000179	0.042303
$S_\xi$	<b>-0.4</b>	-0.61225	-0.50622	-0.33247	-0.25812	<b>-0.42726</b>	0.161295	0.377506
$\omega_0$	<b>189.1239</b>	188.9366	189.1497	189.0965	189.1741	<b>189.0892</b>	0.106813	0.000565
$S_\omega$	<b>-1256.64</b>	-1092.83	-1243.7	-1226.6	-1303.53	<b>-1216.66</b>	88.90028	0.073069

Table 5.7. Optimization results for the *Synth* data set; GA, 20% noise

GA 30% noise	ACTUAL	RUN 1	RUN 2	RUN 3	RUN 4	$\bar{x}$	$\sigma$	$\sigma/\bar{x}$
$\xi_0$	<b>0.0042</b>	0.004771	0.003594	0.003973	0.003385	<b>0.003931</b>	0.000611	0.155381
$S_\xi$	<b>-0.4</b>	-0.71675	0.208465	-0.25031	0.246317	<b>-0.12807</b>	0.452737	3.535067
$\omega_0$	<b>189.1239</b>	189.0589	189.052	189.1316	189.0822	<b>189.0812</b>	0.035996	0.000190
$S_\omega$	<b>-1256.64</b>	-1200.27	-1201.69	-1228.52	-1233.67	<b>-1216.04</b>	17.52276	0.014410

Table 5.8. Optimization results for the *Synth* data set; GA, 30% noise

PSO 0% noise	ACTUAL	RUN 1	RUN 2	RUN 3	RUN 4	$\bar{x}$	$\sigma$	$\sigma/\bar{x}$
$\xi_0$	<b>0.0042</b>	0.004437	0.004611	0.004557	0.004614	<b>0.004555</b>	8.27E-05	0.018151
$S_\xi$	<b>-0.4</b>	-0.56995	-0.81059	-0.70964	-0.67819	<b>-0.69209</b>	0.099097	0.143185
$\omega_0$	<b>189.1239</b>	189.1421	189.0642	189.0937	189.1477	<b>189.1119</b>	0.039992	0.000211
$S_\omega$	<b>-1256.64</b>	-1302.46	-1192.85	-1245.96	-1288.61	<b>-1257.47</b>	49.3357	0.039234

Table 5.9. Optimization results for the *Synth* data set; PSO, 0% noise

PSO 10% noise	ACTUAL	RUN 1	RUN 2	RUN 3	RUN 4	$\bar{x}$	$\sigma$	$\sigma/\bar{x}$
$\xi_0$	<b>0.0042</b>	0.004785	0.004713	0.003923	0.003881	<b>0.004326</b>	0.00049	0.113324
$S_\xi$	<b>-0.4</b>	-0.79555	-0.81552	-0.23406	-0.14349	<b>-0.49716</b>	0.358097	0.720290
$\omega_0$	<b>189.1239</b>	188.9828	189.0352	189.1041	189.088	<b>189.0525</b>	0.054989	0.000291
$S_\omega$	<b>-1256.64</b>	-1131.53	-1192.03	-1253.12	-1236.37	<b>-1203.26</b>	54.32264	0.045146

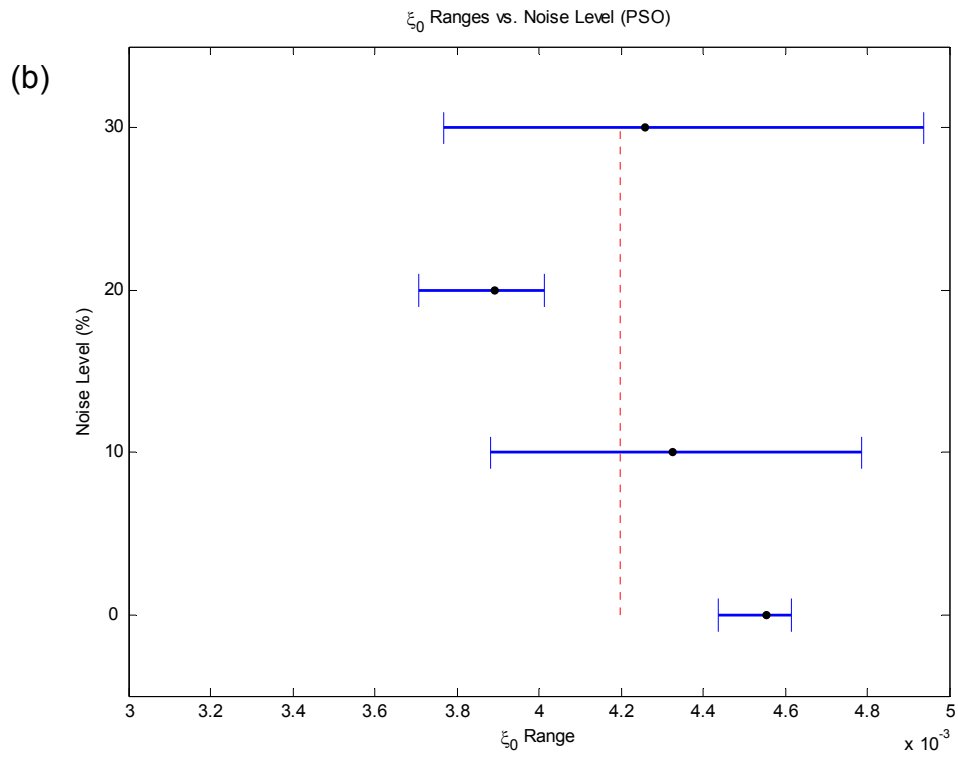
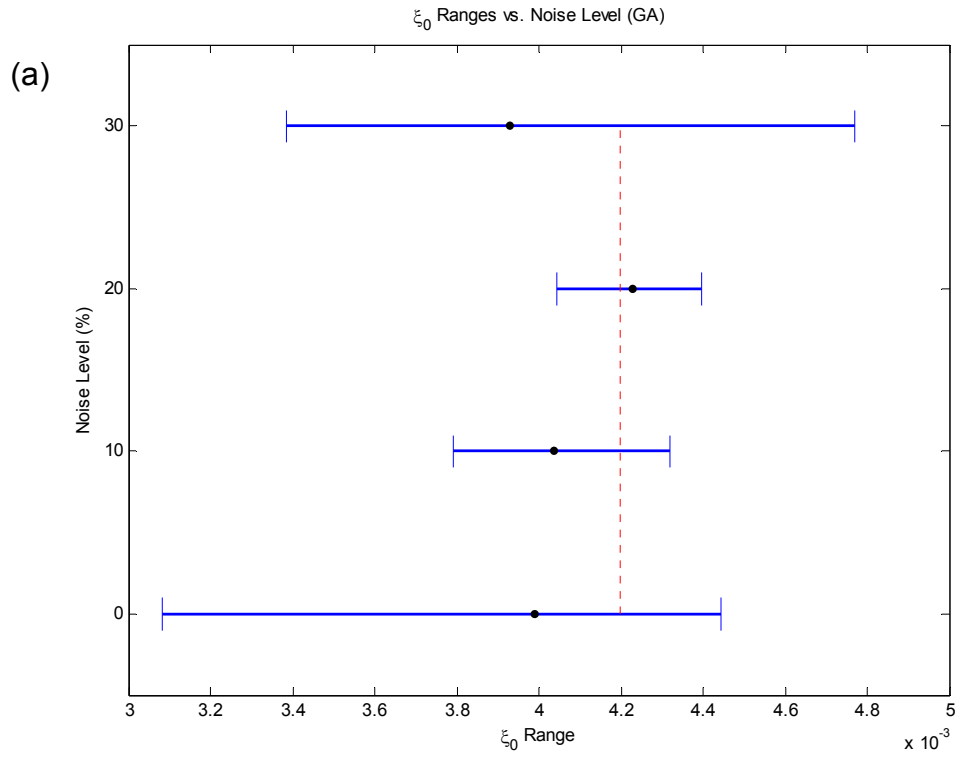
Table 5.10. Optimization results for the *Synth* data set; PSO, 10% noise

PSO 20% noise	ACTUAL	RUN 1	RUN 2	RUN 3	RUN 4	$\bar{x}$	$\sigma$	$\sigma/\bar{x}$
$\xi_0$	<b>0.0042</b>	0.003963	0.004014	0.003709	0.00389	<b>0.003894</b>	0.000134	0.034335
$S_\xi$	<b>-0.4</b>	-0.14437	-0.26865	-0.02538	-0.24365	<b>-0.17052</b>	0.110643	0.648877
$\omega_0$	<b>189.1239</b>	189.0428	189.1218	189.1098	189.1164	<b>189.0977</b>	0.036953	0.000195
$S_\omega$	<b>-1256.64</b>	-1194.78	-1245.47	-1245.32	-1240.31	<b>-1231.47</b>	24.57532	0.019956

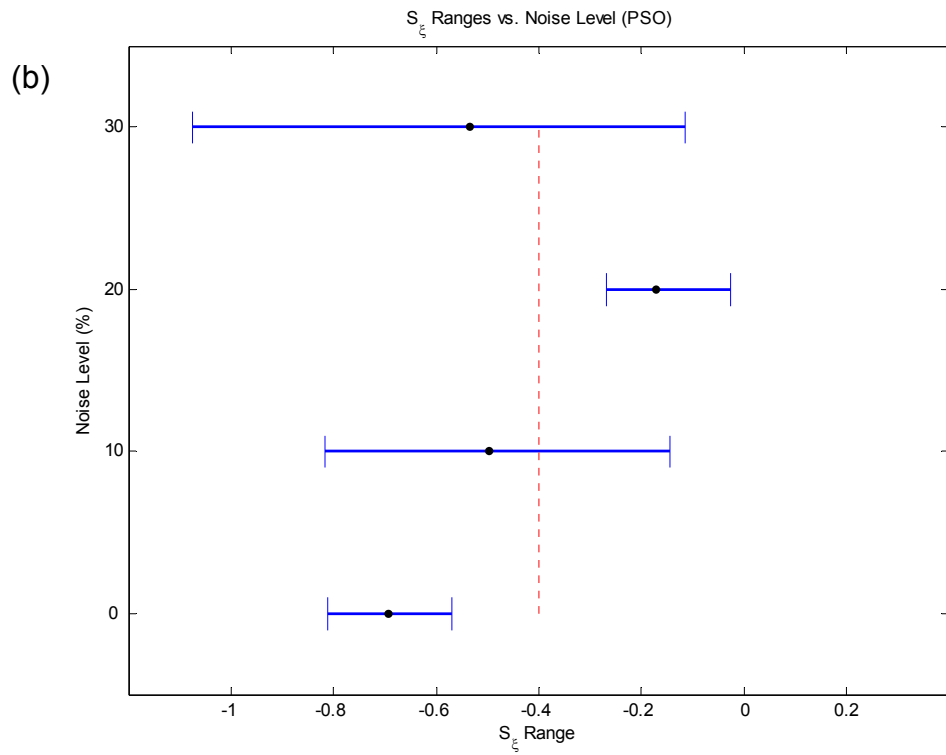
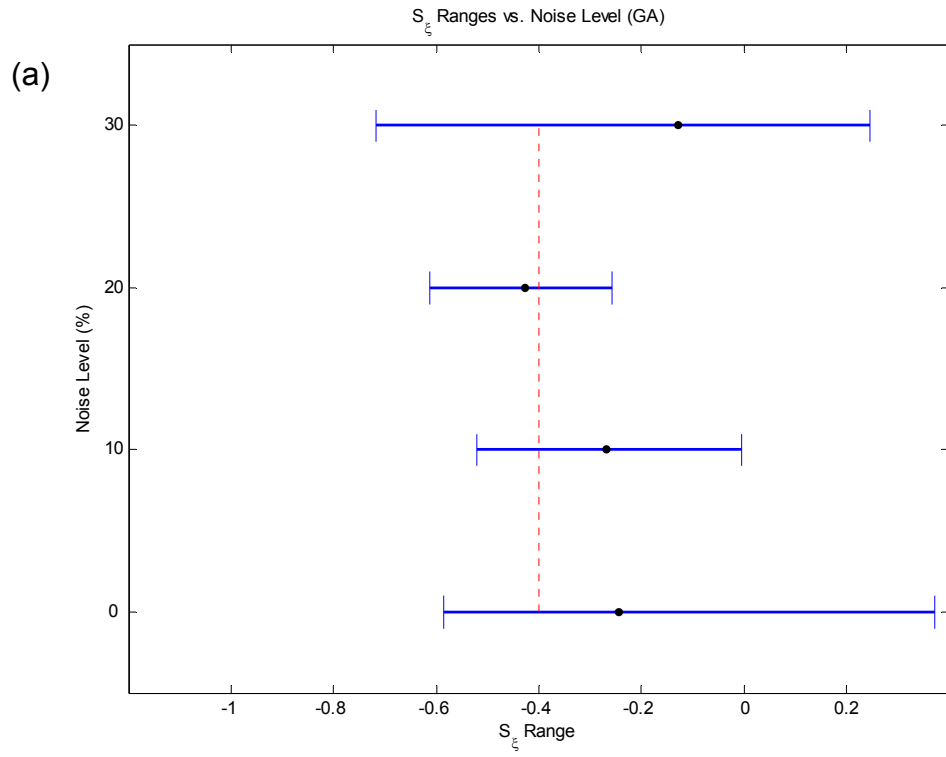
Table 5.11. Optimization results for the *Synth* data set; PSO, 20% noise

PSO 30% noise	ACTUAL	RUN 1	RUN 2	RUN 3	RUN 4	$\bar{x}$	$\sigma$	$\sigma/\bar{x}$
$\xi_0$	<b>0.0042</b>	0.004226	0.004108	0.004937	0.003767	<b>0.004259</b>	0.000492	0.115522
$S_\xi$	<b>-0.4</b>	-0.4518	-0.4938	-1.07537	-0.1138	<b>-0.53369</b>	0.399173	0.747946
$\omega_0$	<b>189.1239</b>	189.1569	189.1508	189.3972	189.1963	<b>189.2253</b>	0.116378	0.000615
$S_\omega$	<b>-1256.64</b>	-1264.21	-1262.62	-1442.37	-1347.87	<b>-1329.27</b>	85.26707	0.064146

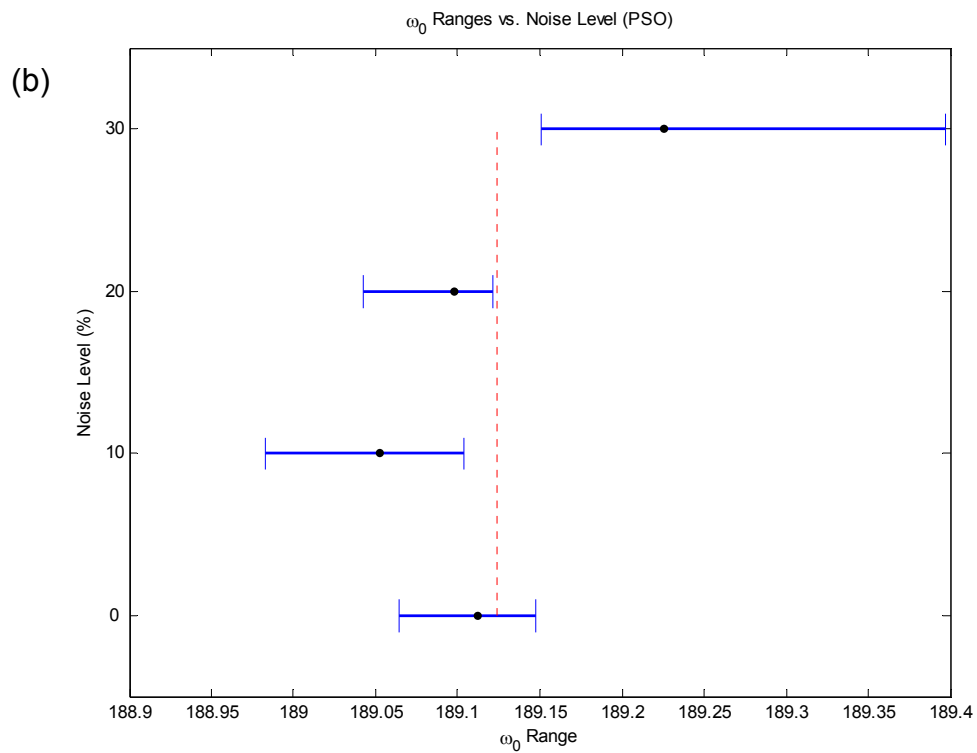
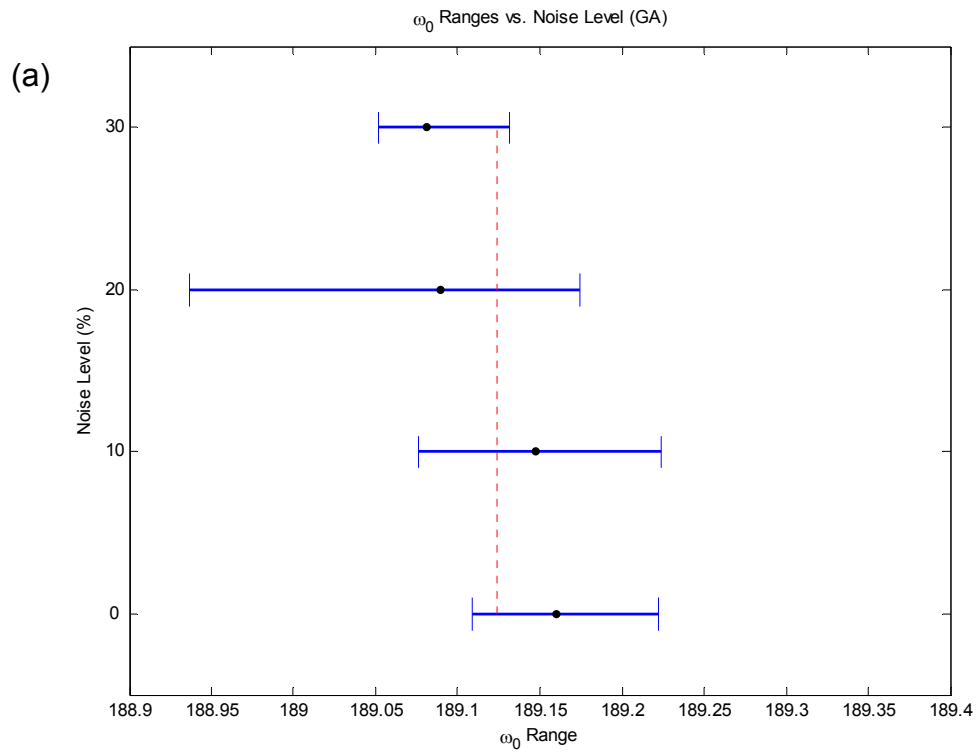
Table 5.12. Optimization results for the *Synth* data set; PSO, 30% noise



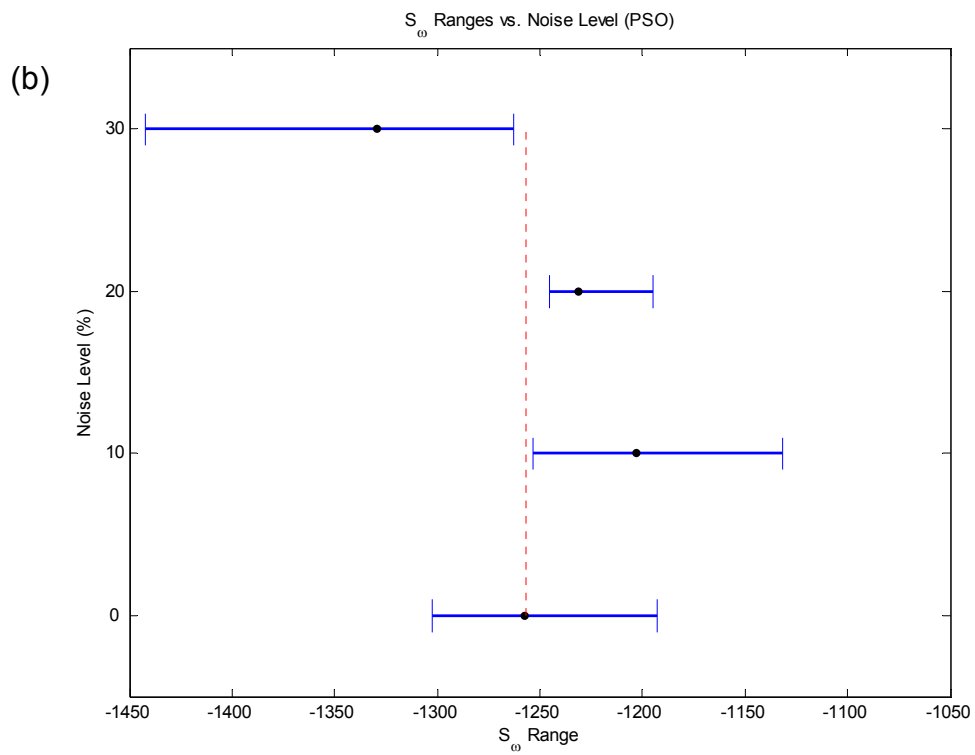
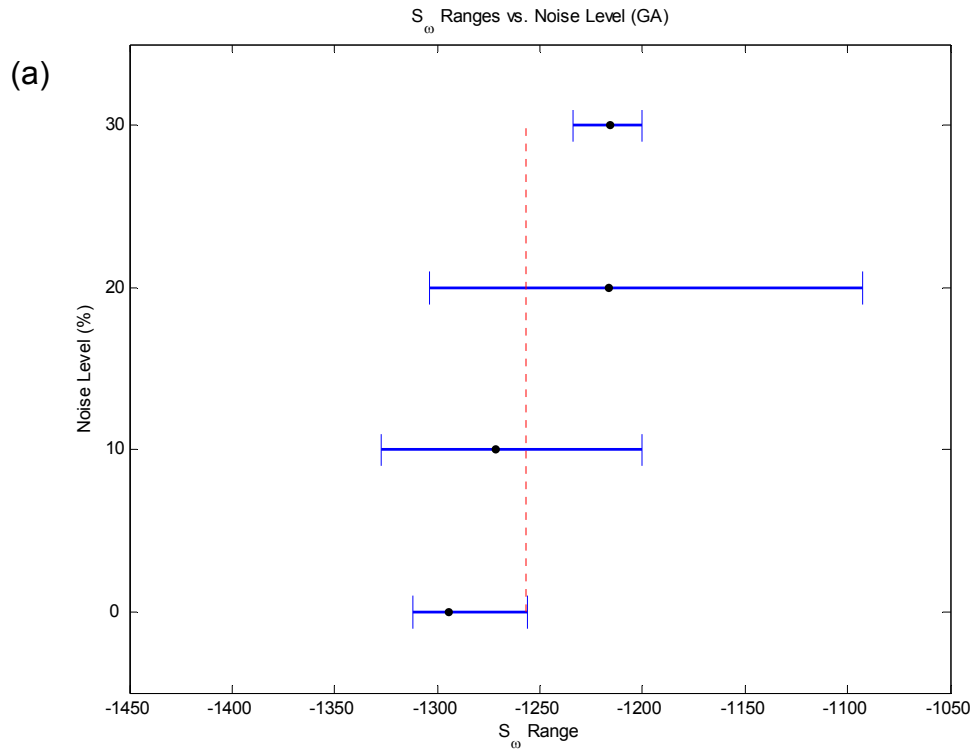
**Figure 5.2.** Scatter plots from the results of the *Synth* set, for  $\xi_0$ ; (a) GA, (b) PSO



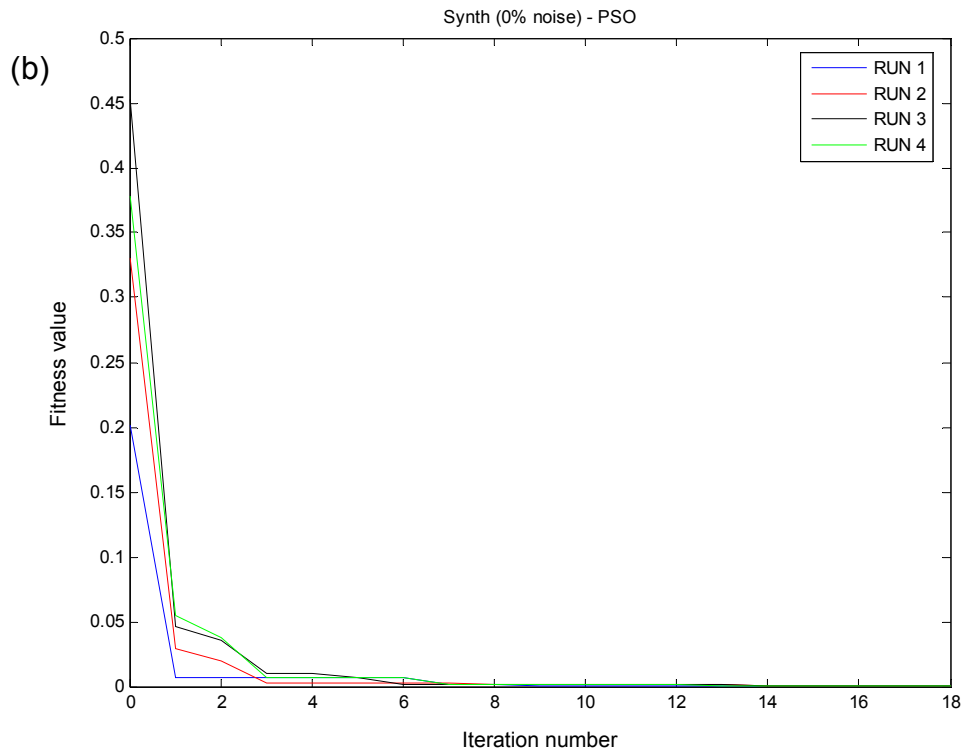
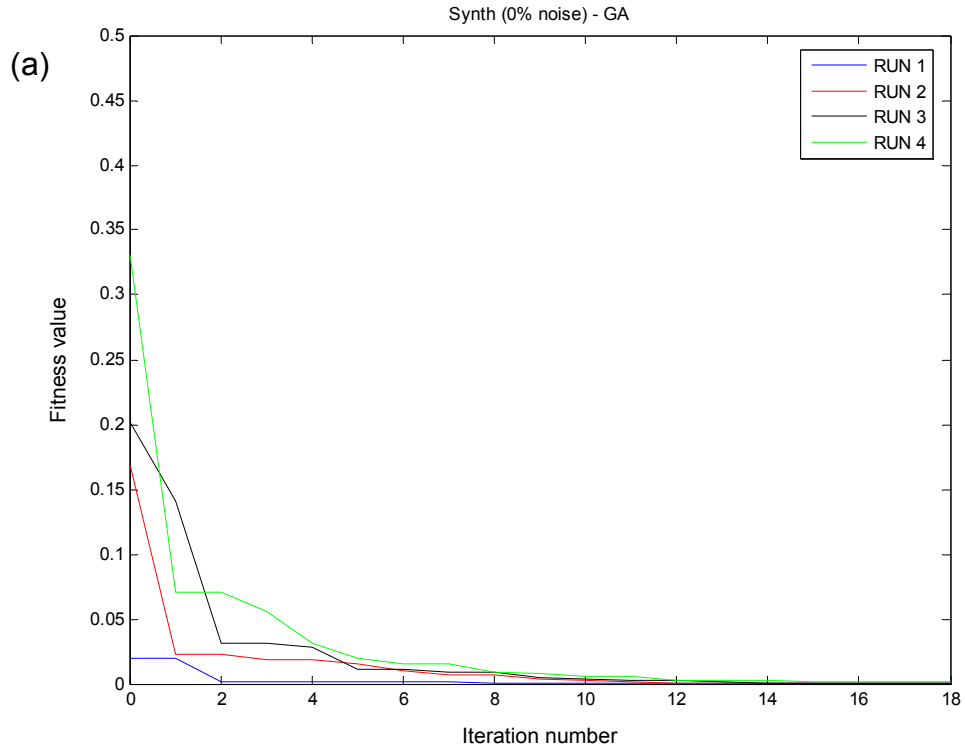
**Figure 5.3.** Scatter plots from the results of the *Synth* set, for  $S_{\xi}$ : (a) GA, (b) PSO



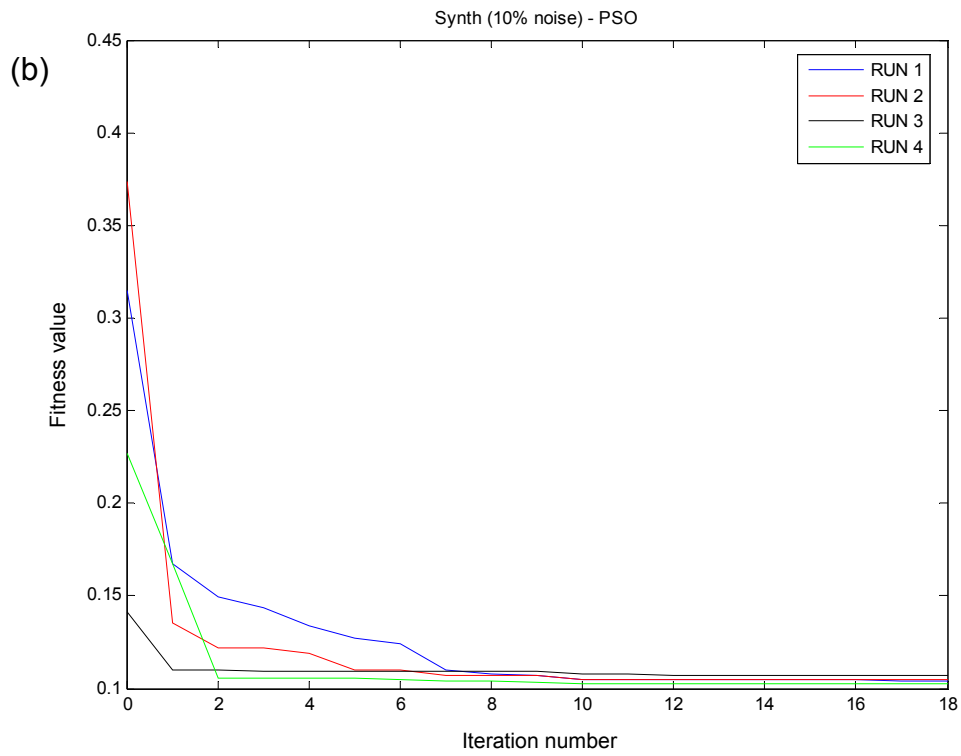
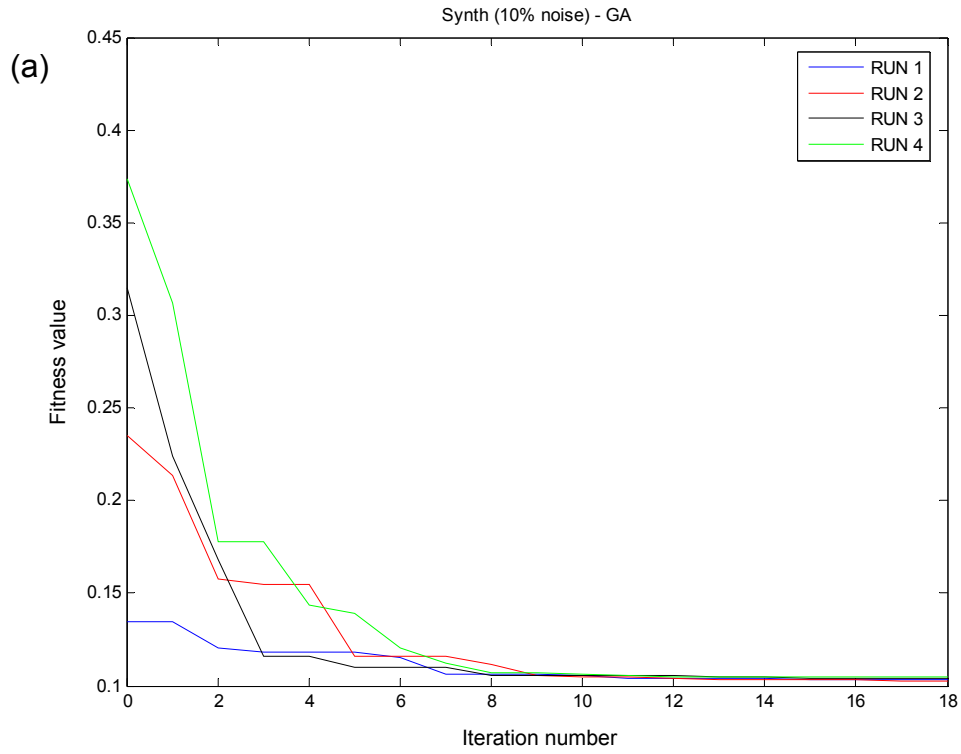
**Figure 5.4.** Scatter plots from the results of the *Synth* set, for  $\omega_0$ ; (a) GA, (b) PSO



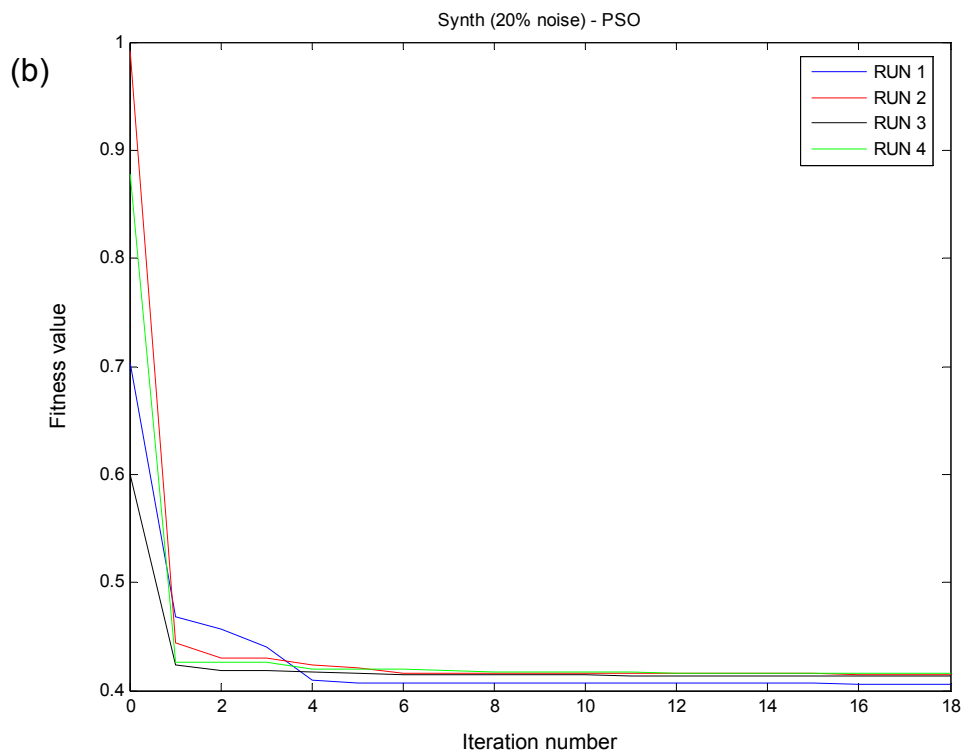
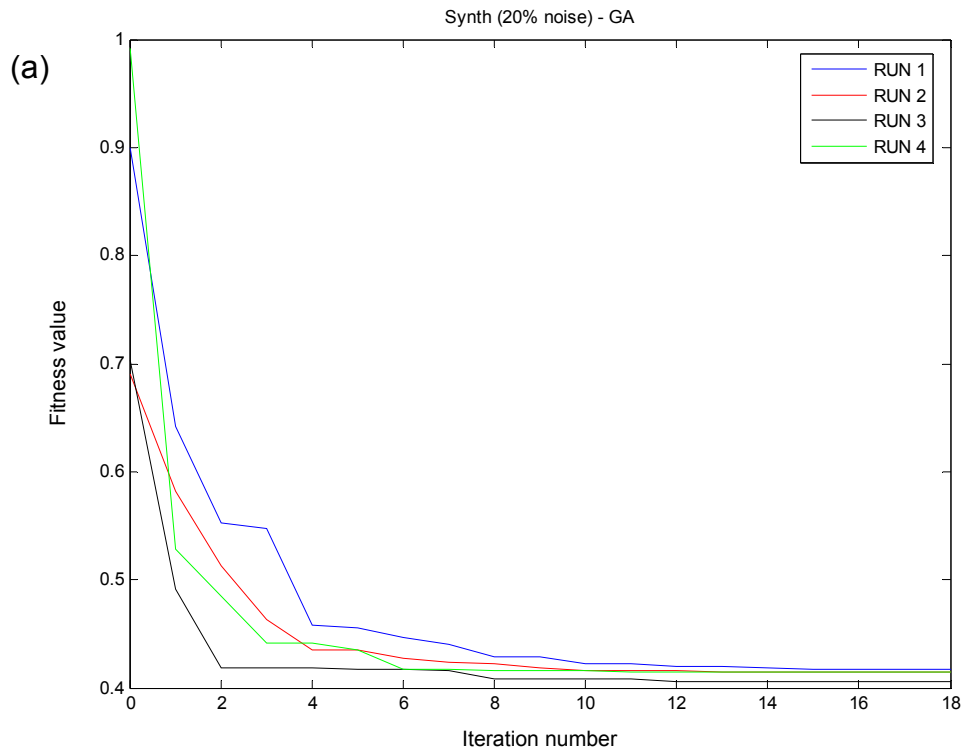
**Figure 5.5.** Scatter plots from the results of the *Synth* set, for  $S_{\omega}$ : (a) GA, (b) PSO



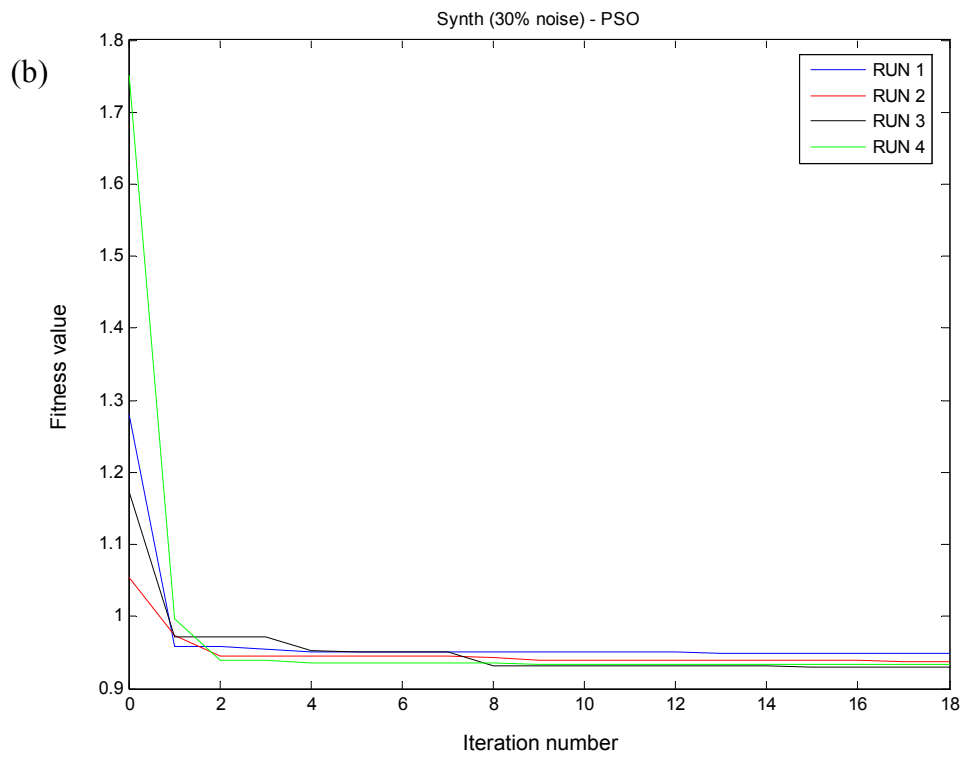
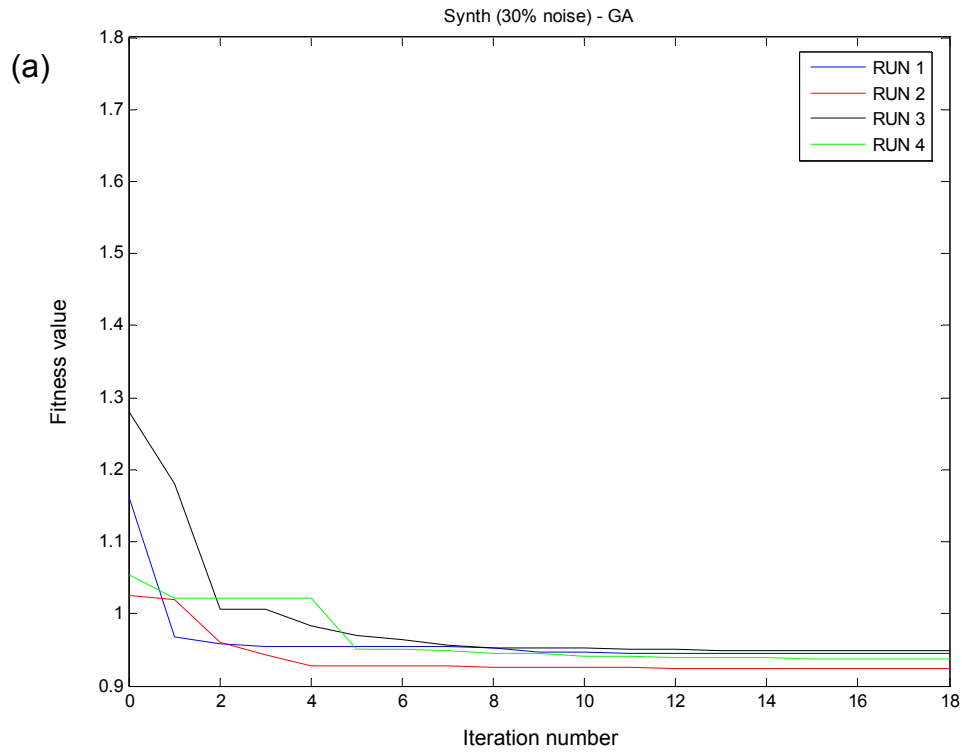
**Figure 5.6.** Convergence plots from the results of the *Synth* set, for 0% noise; (a) GA, (b) PSO



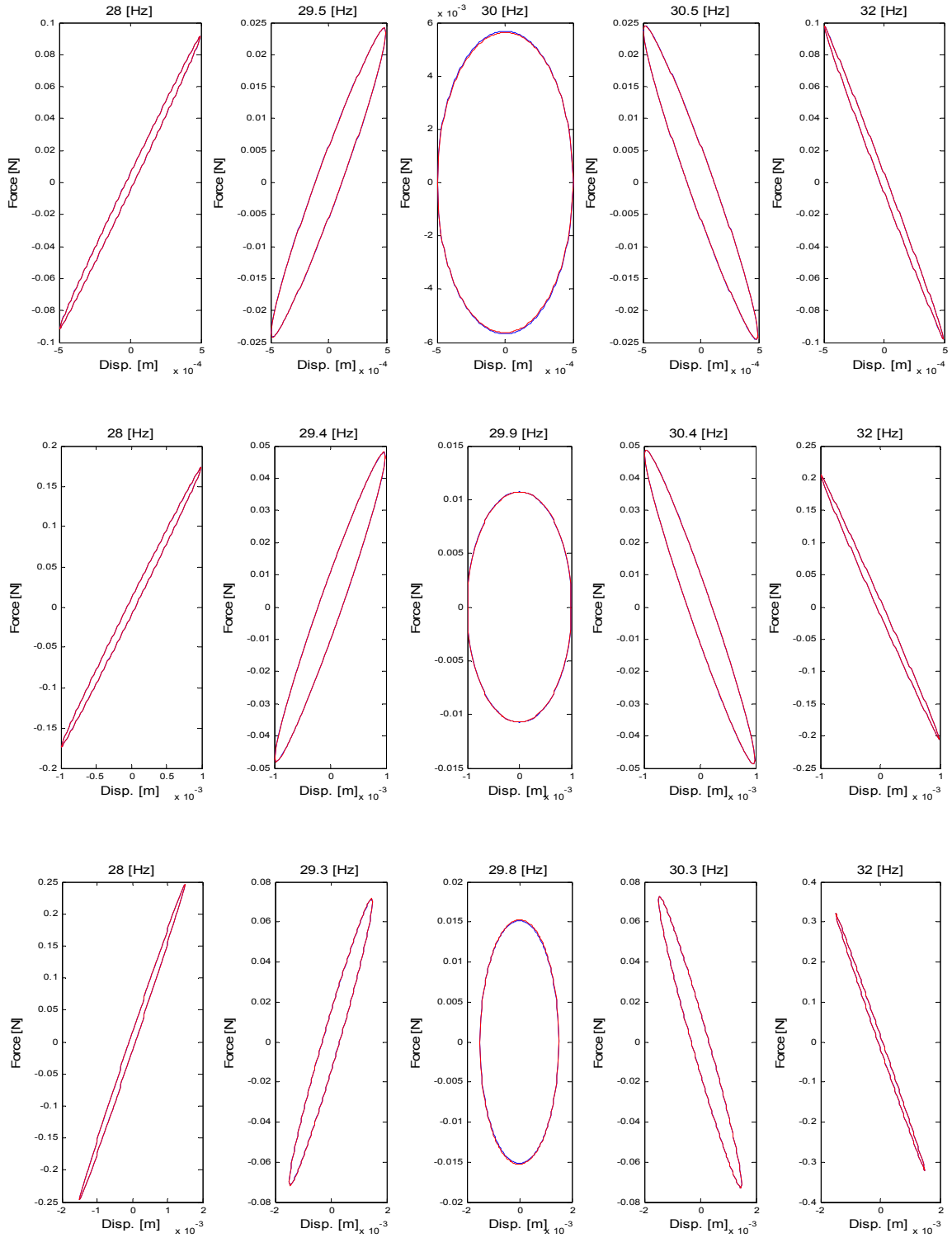
**Figure 5.7.** Convergence plots from the results of the *Synth* set, for 10% noise; (a) GA, (b) PSO



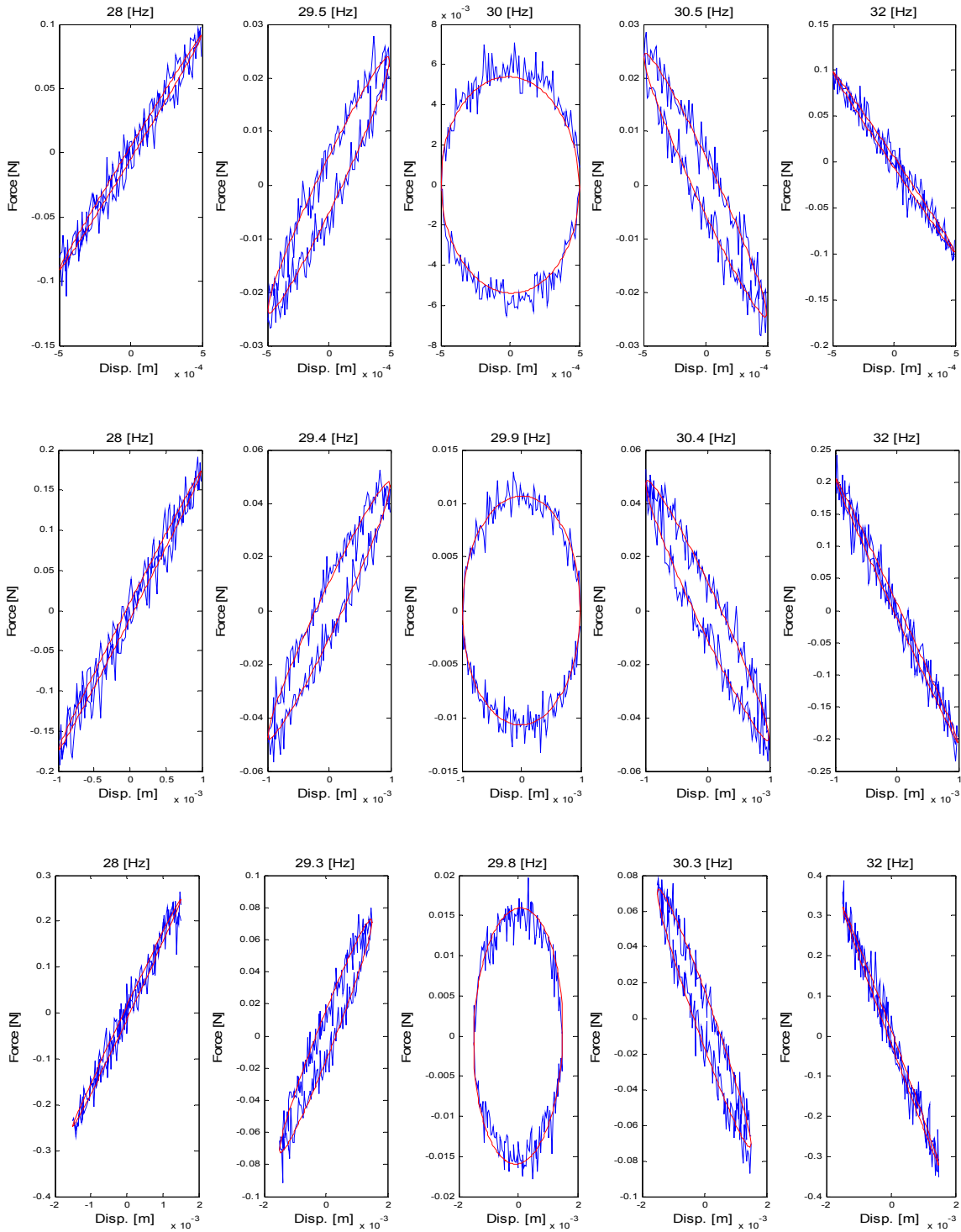
**Figure 5.8.** Convergence plots from the results of the *Synth* set, for 20% noise; (a) GA, (b) PSO



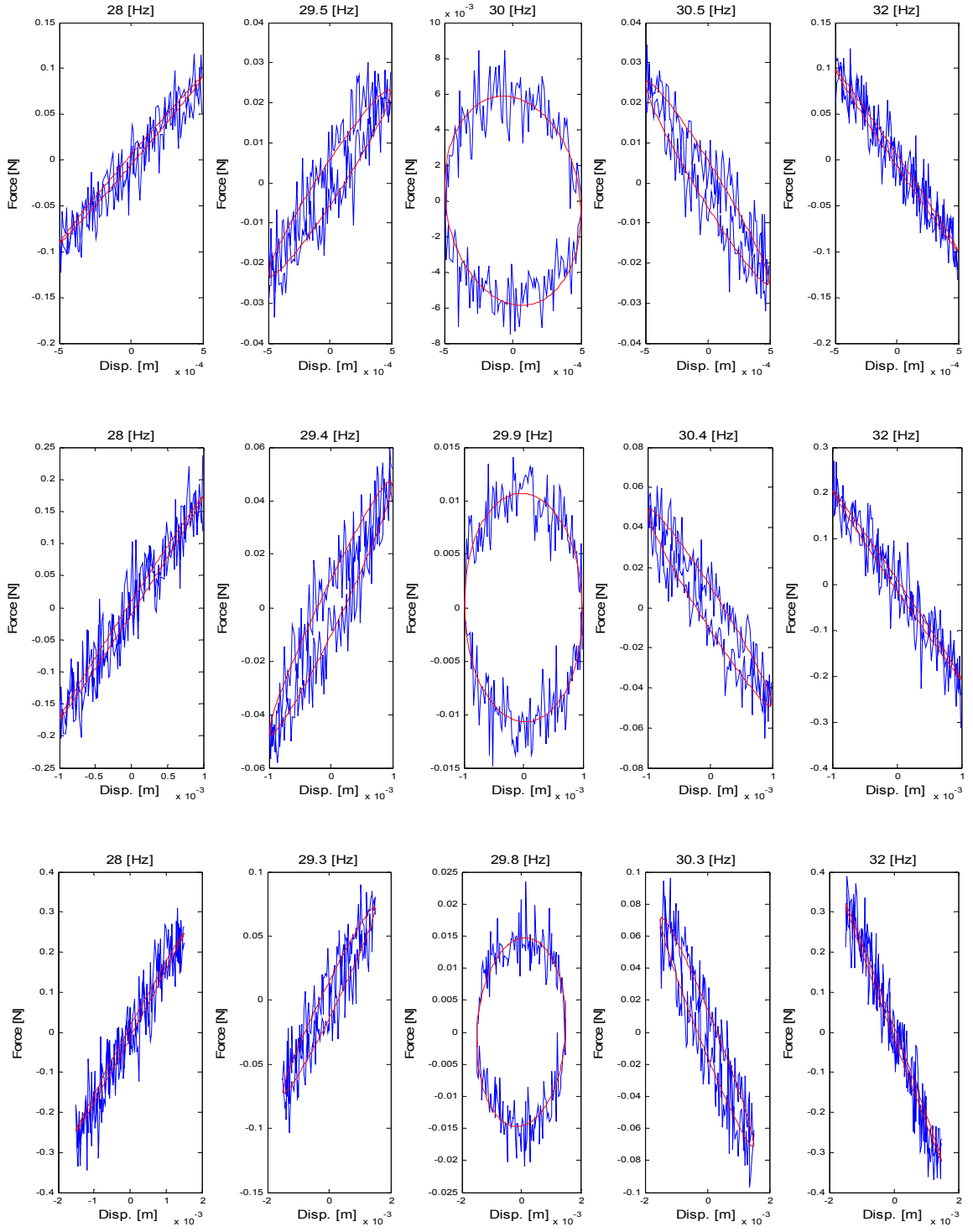
**Figure 5.9.** Convergence plots from the results of the *Synth* set, for 30% noise; (a) GA, (b) PSO



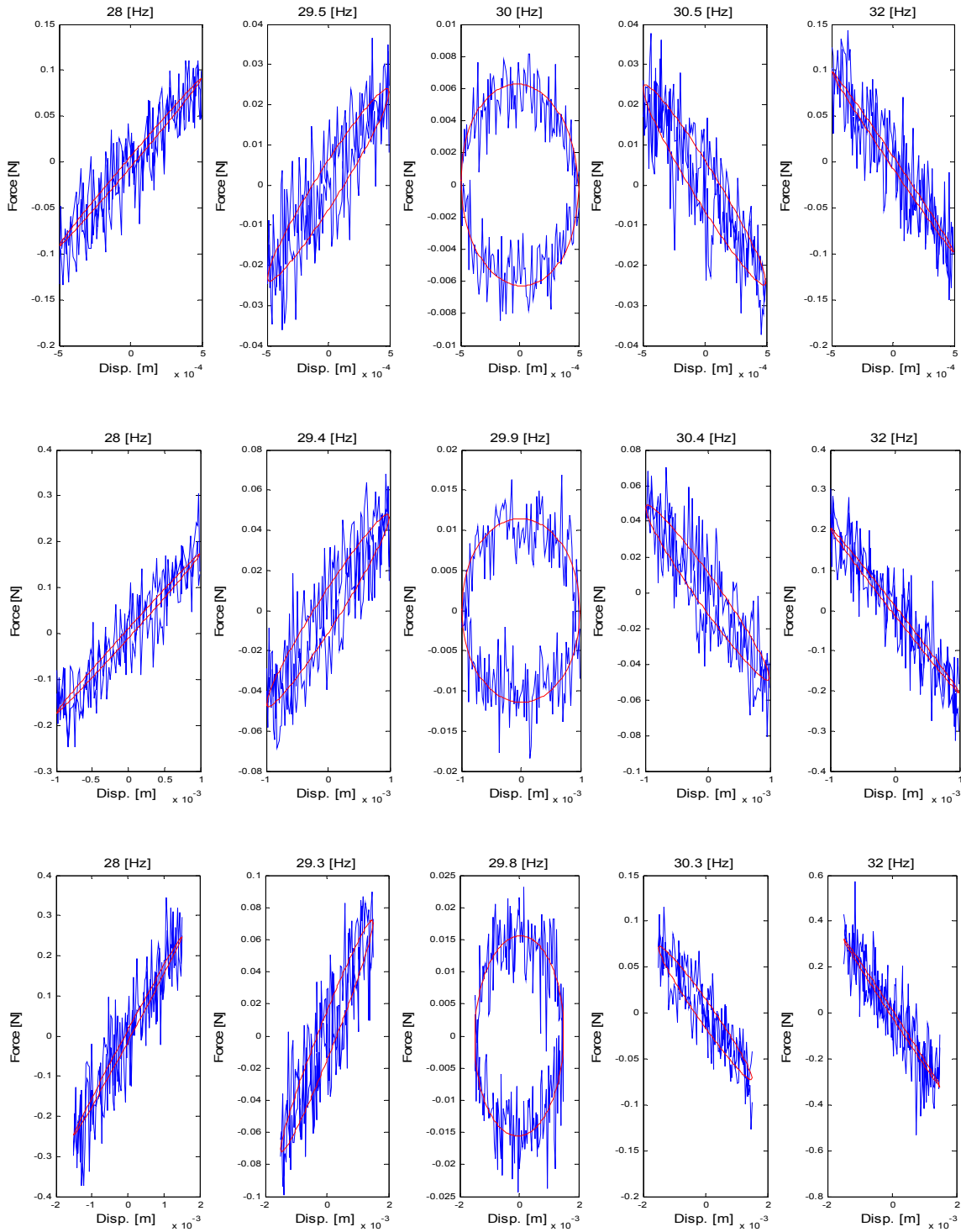
**Figure 5.10.** Comparison plots for some of the hysteresis loops of the *Synth* set; 0% noise; 1st.row: 0.5mm; 2nd.row: 1.0mm; 3rd.row: 1.5mm; — trial, — model



**Figure 5.11.** Comparison plots for some of the hysteresis loops of the *Synth* set; 10% noise; 1st.row: 0.5mm; 2nd.row: 1.0mm; 3rd.row: 1.5mm; — trial, — model



**Figure 5.12.** Comparison plots for some of the hysteresis loops of the *Synth* set; 20% noise; 1st.row: 0.5mm; 2nd.row: 1.0mm; 3rd.row: 1.5mm; — trial, — model



**Figure 5.13.** Comparison plots for some of the hysteresis loops of the *Synth* set; 30% noise; 1st.row: 0.5mm; 2nd.row: 1.0mm; 3rd.row: 1.5mm; — trial, — model

Parameter	0% noise		10% noise		20% noise		30% noise	
	GA	PSO	GA	PSO	GA	PSO	GA	PSO
$\xi_0$	2.10E-4	3.55E-4	1.63E-4	1.26E-4	0.27E-4	3.06E-4	2.69E-4	0.59E-4
$S_\xi$	0.1570	0.2920	0.1320	0.0972	0.0273	0.2290	0.2720	0.1340
$\omega_0$	0.0366	0.0120	0.0235	0.0714	0.0347	0.0262	0.0427	0.101
$S_\omega$	37.4	0.834	15.2	53.4	40.0	25.2	40.6	72.6

**Table 5.13.** Differences between the resultant averages and the actual global optimum, for GA and PSO, at different noise levels

### 5.3. Material Samples Description and Experimental Results

As explained in section 1.2, three sandwich composite material beams have been studied. They consist of a paper honeycomb core filled with foam and external layers of interlaced strips of carbon fibers, as shown in Fig. 1.1b. The samples used are described in Table 5.14, based on the sketch shown in Fig. 5.14.



**Figure 5.14.** Sketch of a sandwich composite beam sample

PROPERTY	SAMPLE 1	SAMPLE 2	SAMPLE 3
Length (L) [mm]	590	590	590
Width (W) [mm]	25	25	25
Core Thickness ( $H_1$ ) [mm]	6.5	6.5	13
Face Thickness ( $H_2$ ) [mm]	0.2	0.2	0.2
Faces per side	1	2	1
Mass [g]	28.7	36.8	48.0

**Table 5.14.** Properties of the sandwich composite beams studied

In a way similar to that described in section 5.2 for the *Synth* set, several hysteresis loops have been experimentally obtained for all 3 beam samples. For samples 1 and 2, a total of 40 frequencies have been used, at 3 different displacement excitation amplitudes each, making a total of 120 hysteresis loops for each of these samples. For sample 3, however, 31 frequencies have been used, at 3 different displacement excitation amplitudes each, making a total of 93 hysteresis loops for that sample. The excitation frequencies used for each sample sweep a range that crosses their first resonance frequency, and they are shown in Tables 5.15 to 5.17. The displacement excitation amplitudes used are indicated in Table 5.18, and the search spaces used for the beam samples are shown in Table 5.19.

Excitation Frequencies [Hz] - Sample 1									
28	29	30	30.5	31	31.2	31.4	31.6	31.8	32
32.2	32.4	32.6	32.7	32.8	32.9	33	33.1	33.2	33.3
33.4	33.5	33.6	33.7	33.8	33.9	34	34.2	34.4	34.6
34.8	35	35.2	35.4	35.6	36	36.5	37	38	39

**Table 5.15.** Excitation frequencies used with sample 1

Excitation Frequencies [Hz] - Sample 2									
37	38	38.5	39	39.5	40	40.2	40.4	40.6	40.8
41	41.2	41.4	41.5	41.6	41.7	41.8	41.9	42	42.1
42.2	42.3	42.4	42.5	42.6	42.7	42.8	42.9	43	43.2
43.4	43.6	43.8	44	44.2	44.4	45	45.5	46	47

**Table 5.16.** Excitation frequencies used with sample 2

Excitation Frequencies [Hz] - Sample 3							
81	82	83	84	84.5	85	85.6	85.8
86	86.1	86.2	86.3	86.4	86.5	86.6	86.7
86.8	86.9	87	87.1	87.2	87.3	87.4	87.6
87.8	88	88.2	88.4	89	89.5	90	

**Table 5.17.** Excitation frequencies used with sample 3

Amplitude [mm]	Sample 1	Sample 2	Sample 3
Low	0.5	0.2	0.05
Medium	0.8	0.5	0.10
High	1.0	0.8	0.15

**Table 5.18.** Excitation displacement amplitudes used with the beam samples studied

Parameters	Sample 1		Sample 2		Sample 3	
	Min	Max	Min	Max	Min	Max
$\xi_0$	0	0.05	0	0.1	0	0.05
$S_\xi$	-20	20	-10	10	-100	100
$\omega_0$	100	400	100	500	400	700
$S_\omega$	-4000	4000	-10000	10000	-10000	10000

**Table 5.19.** Search spaces for all 3 beam samples studied

### 5.3.1. Optimization Results from Experimental Data

A total of 4 runs of each optimization routine were performed, for the identification of the system parameters of all 3 beam samples studied. As it has been done in section 5.2.1, the parametric results for each case, along with their average values ( $\bar{x}$ ), standard deviations ( $\sigma$ ), and normalized standard deviations ( $\sigma/\bar{x}$ ), are shown in

Tables 5.20 to 5.22 for GA, and Tables 5.23 to 5.25 for PSO. Scatter plots, comparing the GA and PSO spread ranges in the results for each parameter, are shown in Figs. 5.15 to 5.17. Also, convergence plots for both algorithms (overlying all 4 runs in each case, and with matching scales, for comparison purposes), and for each beam sample, are shown in Figs. 5.18 to 5.20. Some plots showing comparisons between trial hysteresis loops and model results (typical for both optimization techniques), are shown in Figs. 5.21 to 5.23, for all 3 excitation amplitudes.

GA SAMPLE 1	RUN 1	RUN 2	RUN 3	RUN 4	$\bar{x}$	$\sigma$	$\sigma/\bar{x}$
$\xi_0$	0.00554	0.005378	0.005608	0.004806	<b>0.005333</b>	0.000364	0.068322
$S_\xi$	-2.41365	-2.22411	-2.46943	-1.77114	<b>-2.21958</b>	0.316861	0.142757
$\omega_0$	209.9339	209.9262	209.8417	210.3053	<b>210.0018</b>	0.206592	0.000984
$S_\omega$	-1257.99	-1249.85	-1160.66	-1702.94	<b>-1342.86</b>	244.0698	0.181754

**Table 5.20.** GA optimization results for beam sample 1

GA SAMPLE 2	RUN 1	RUN 2	RUN 3	RUN 4	$\bar{x}$	$\sigma$	$\sigma/\bar{x}$
$\xi_0$	0.004618	0.004864	0.005017	0.004822	<b>0.00483</b>	0.000164	0.034017
$S_\xi$	-0.08142	-0.31777	-0.61774	-0.29375	<b>-0.32767</b>	0.220629	0.673327
$\omega_0$	265.8243	265.8762	265.7912	265.7695	<b>265.8153</b>	0.046433	0.000175
$S_\omega$	-1479.36	-1512.6	-1409.21	-1378.99	<b>-1445.04</b>	61.61231	0.042637

**Table 5.21.** GA optimization results for beam sample 2

GA SAMPLE 3	RUN 1	RUN 2	RUN 3	RUN 4	$\bar{x}$	$\sigma$	$\sigma/\bar{x}$
$\xi_0$	0.009875	0.009658	0.00995	0.009857	<b>0.009835</b>	0.000125	0.012664
$S_\xi$	-25.7341	-24.0335	-26.4077	-25.7074	<b>-25.4707</b>	1.01142	0.039709
$\omega_0$	546.2193	546.3223	546.3425	546.389	<b>546.3183</b>	0.071626	0.000131
$S_\omega$	-5584.3	-6259.44	-6498.74	-6853.79	<b>-6299.07</b>	535.4282	0.085001

**Table 5.22.** GA optimization results for beam sample 3

PSO SAMPLE 1	RUN 1	RUN 2	RUN 3	RUN 4	$\bar{x}$	$\sigma$	$\sigma/\bar{x}$
$\xi_0$	0.004985	0.005251	0.005896	0.00591	<b>0.00551</b>	0.000466	0.084637
$S_\xi$	-1.81823	-2.04107	-2.87999	-3.05194	<b>-2.44781</b>	0.609253	0.248897
$\omega_0$	210.035	209.8965	209.9815	209.9329	<b>209.9615</b>	0.060139	0.000286
$S_\omega$	-1376.72	-1206.84	-1314.3	-1258.59	<b>-1289.11</b>	73.04978	0.056667

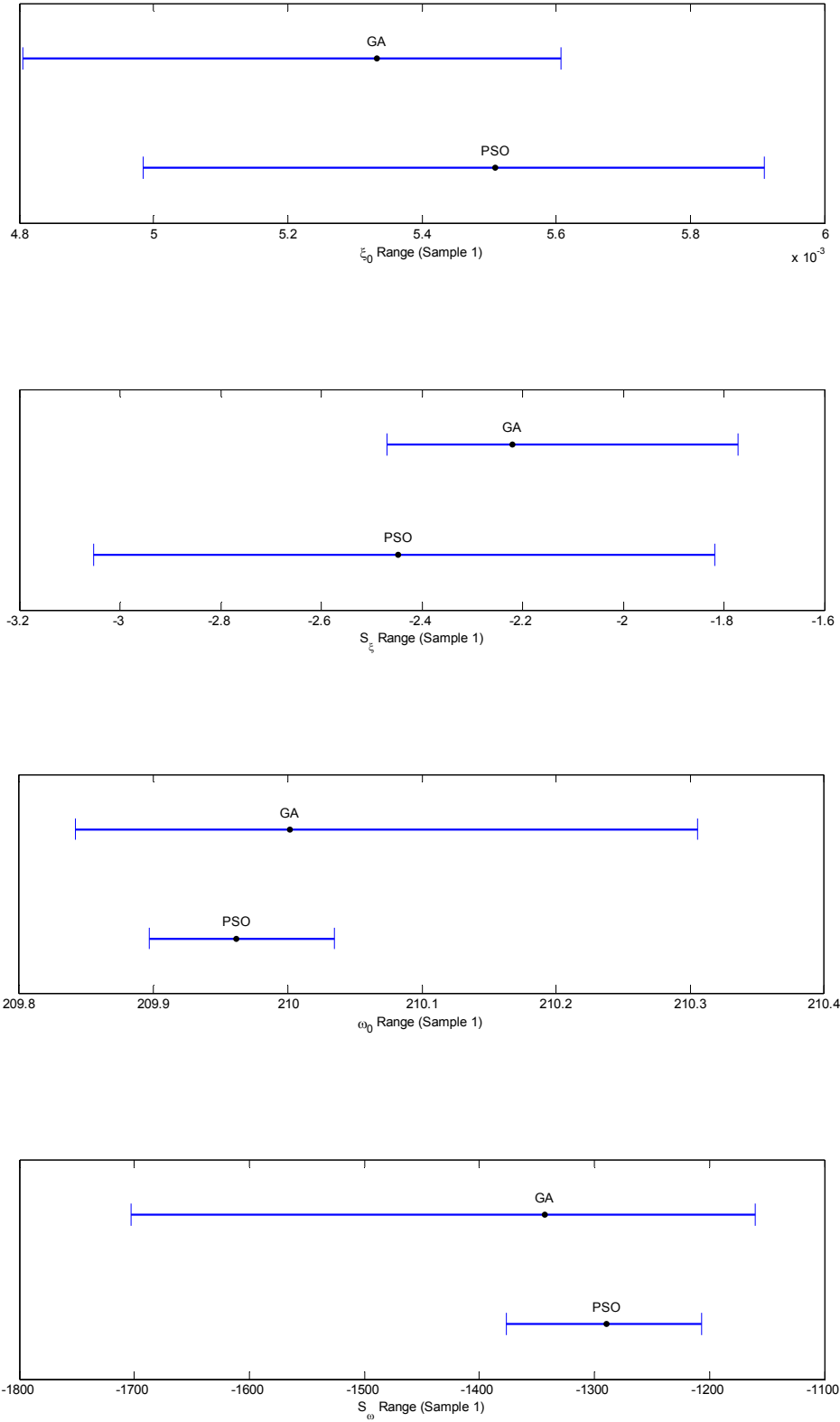
**Table 5.23.** PSO optimization results for beam sample 1

PSO SAMPLE 2	RUN 1	RUN 2	RUN 3	RUN 4	$\bar{x}$	$\sigma$	$\sigma/\bar{x}$
$\xi_0$	0.004922	0.004993	0.004957	0.004672	<b>0.004886</b>	0.000146	0.029812
$S_\xi$	-0.31858	-0.6121	-0.3394	-0.10522	<b>-0.34382</b>	0.207813	0.604417
$\omega_0$	265.8805	265.8835	265.8225	265.8239	<b>265.8526</b>	0.034008	0.000128
$S_\omega$	-1559.42	-1508.21	-1481.76	-1462.2	<b>-1502.9</b>	42.13699	0.028037

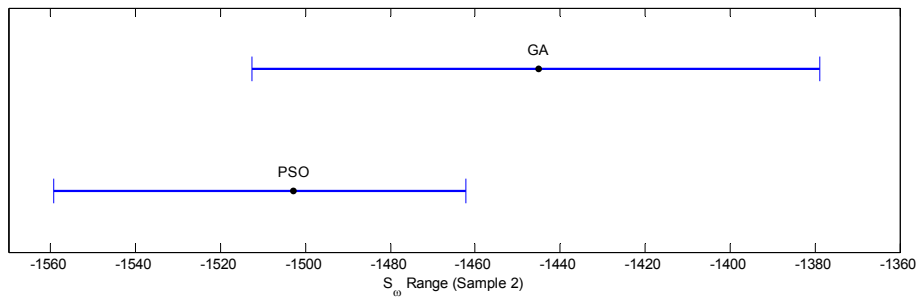
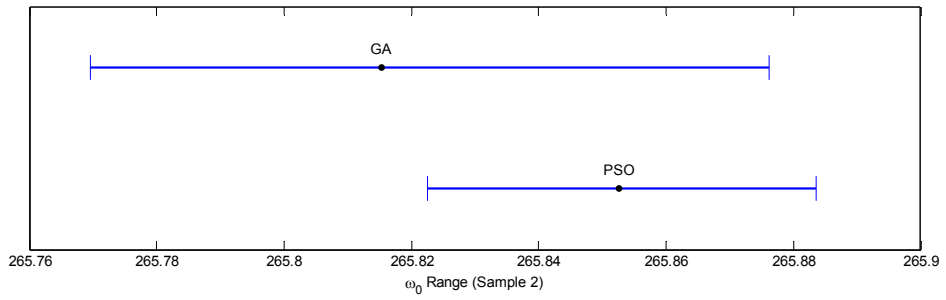
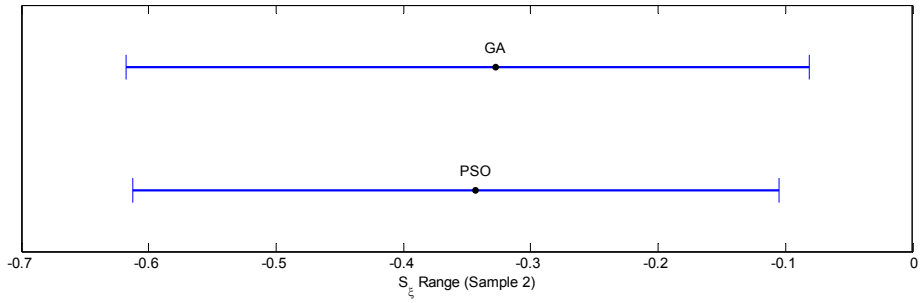
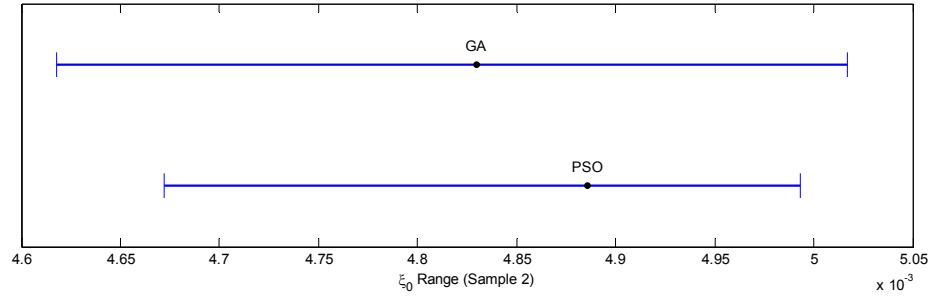
**Table 5.24.** PSO optimization results for beam sample 2

PSO SAMPLE 3	RUN 1	RUN 2	RUN 3	RUN 4	$\bar{x}$	$\sigma$	$\sigma/\bar{x}$
$\xi_0$	0.009784	0.009884	0.009892	0.009974	<b>0.009884</b>	7.79E-05	0.007879
$S_\xi$	-25.0854	-25.9808	-26.1065	-26.3107	<b>-25.8709</b>	0.540977	0.020911
$\omega_0$	546.2392	546.2741	546.2434	546.4486	<b>546.3013</b>	0.099383	0.000182
$S_\omega$	-5740.92	-5792.44	-5850.91	-7493.71	<b>-6219.49</b>	850.6676	0.136774

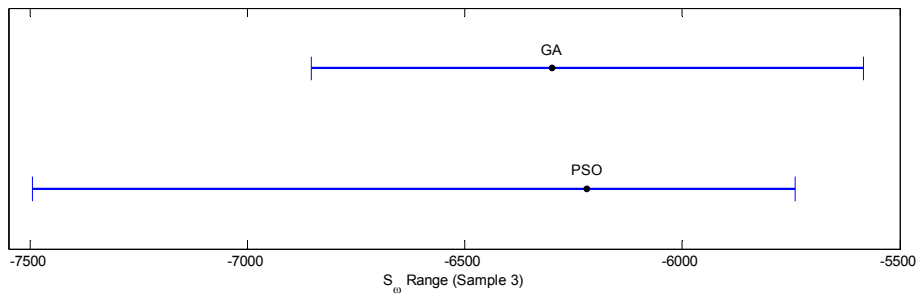
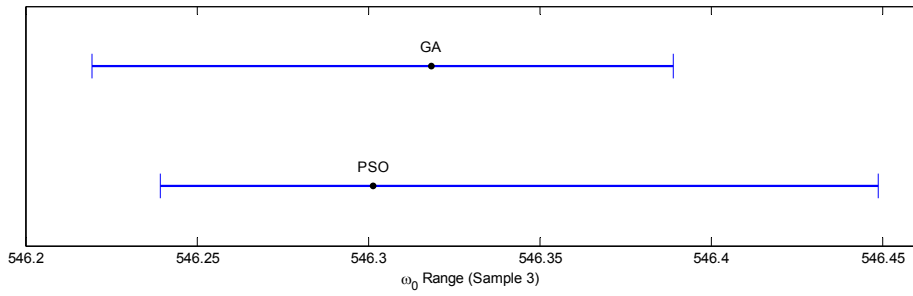
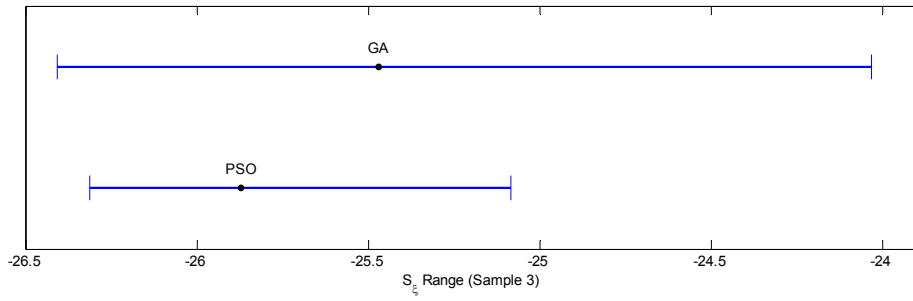
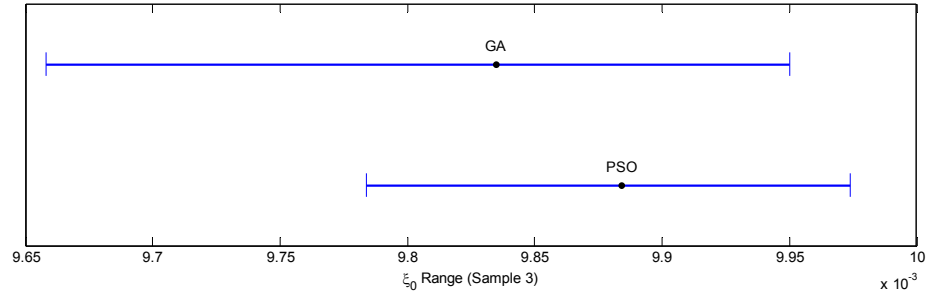
**Table 5.25.** PSO optimization results for beam sample 3



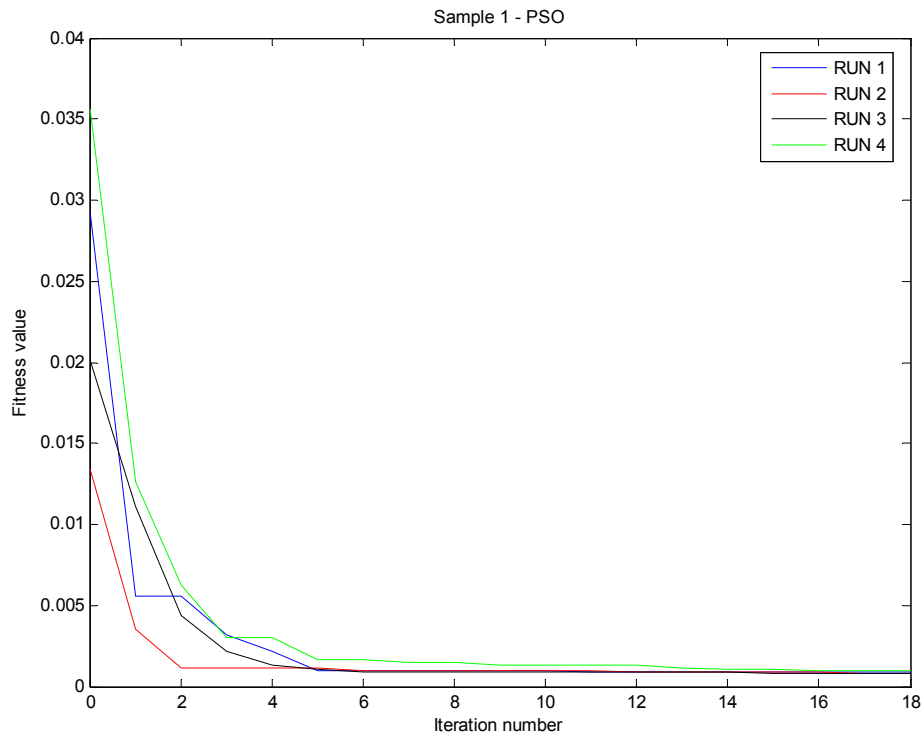
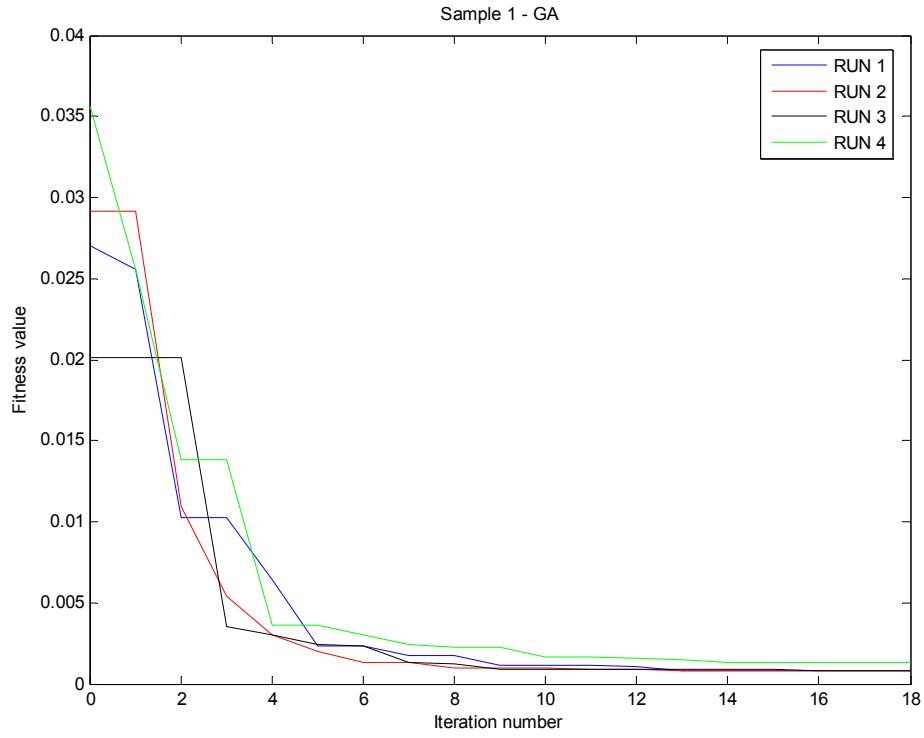
**Figure 5.15.** Parameters scatter plots from the optimization results of beam sample 1



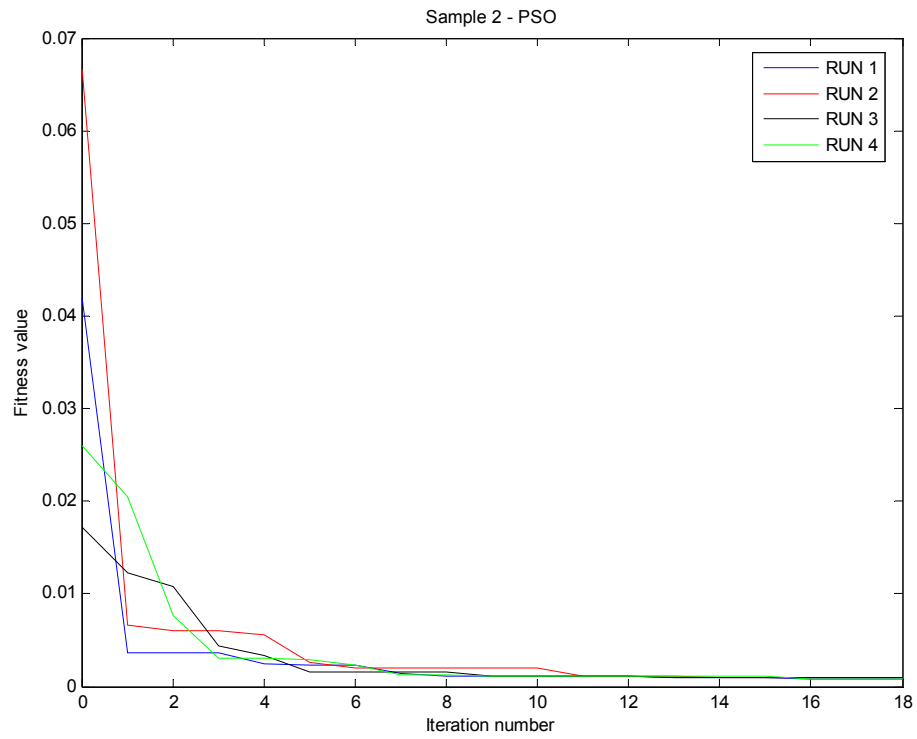
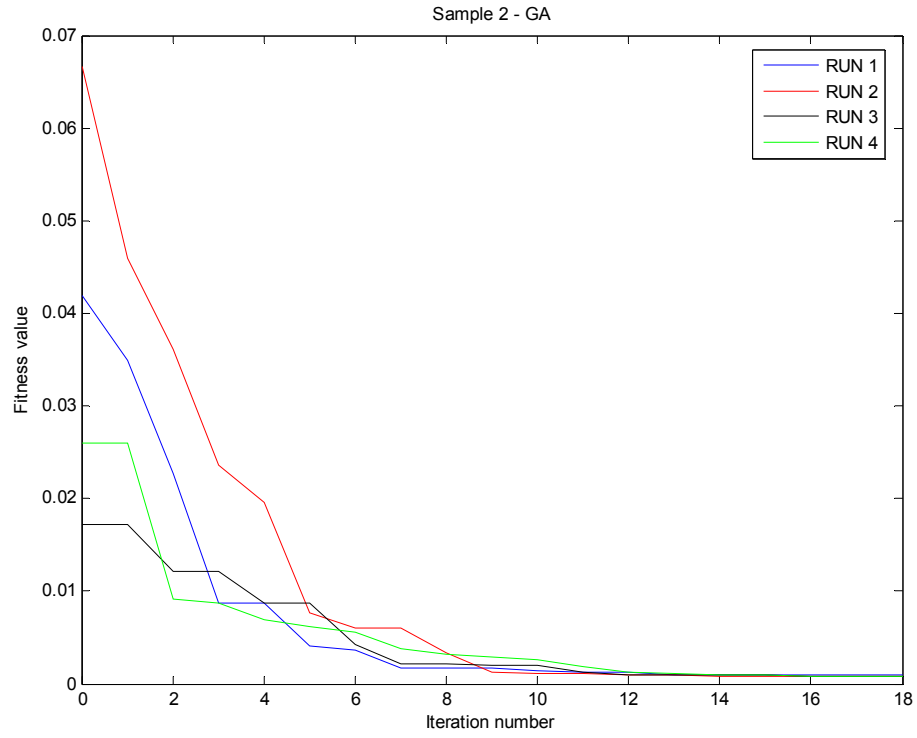
**Figure 5.16.** Parameters scatter plots from the optimization results of beam sample 2



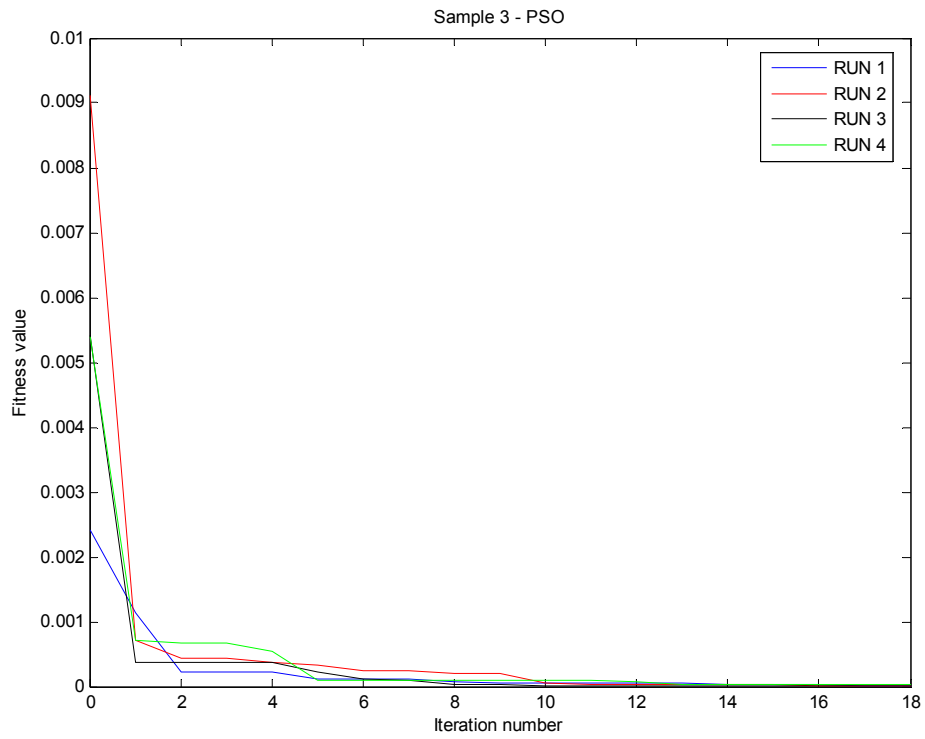
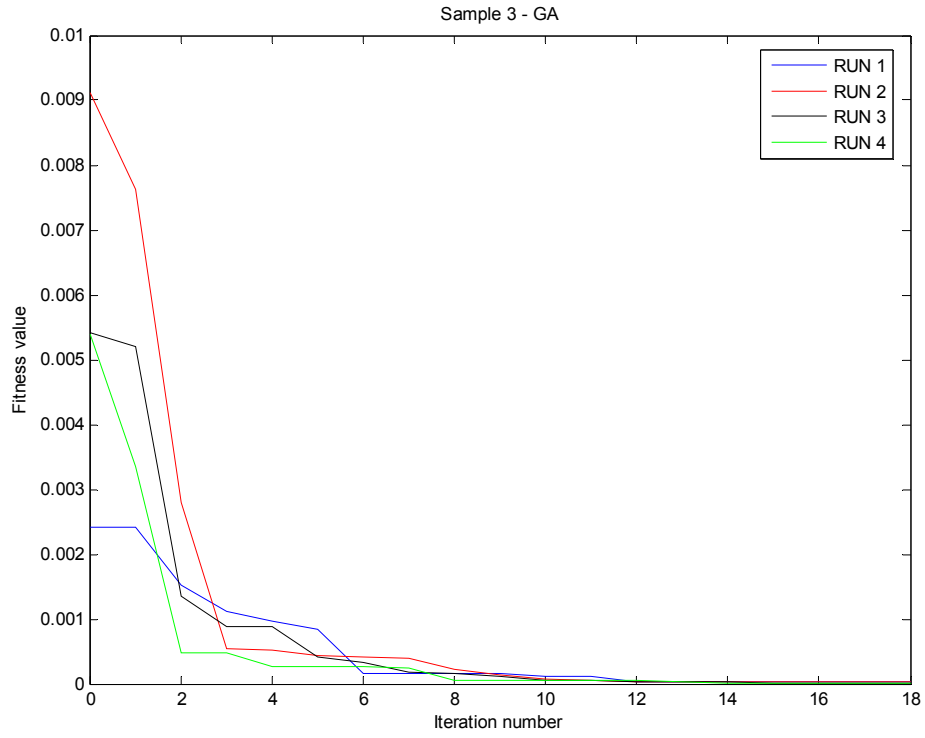
**Figure 5.17.** Parameters scatter plots from the optimization results of beam sample 3



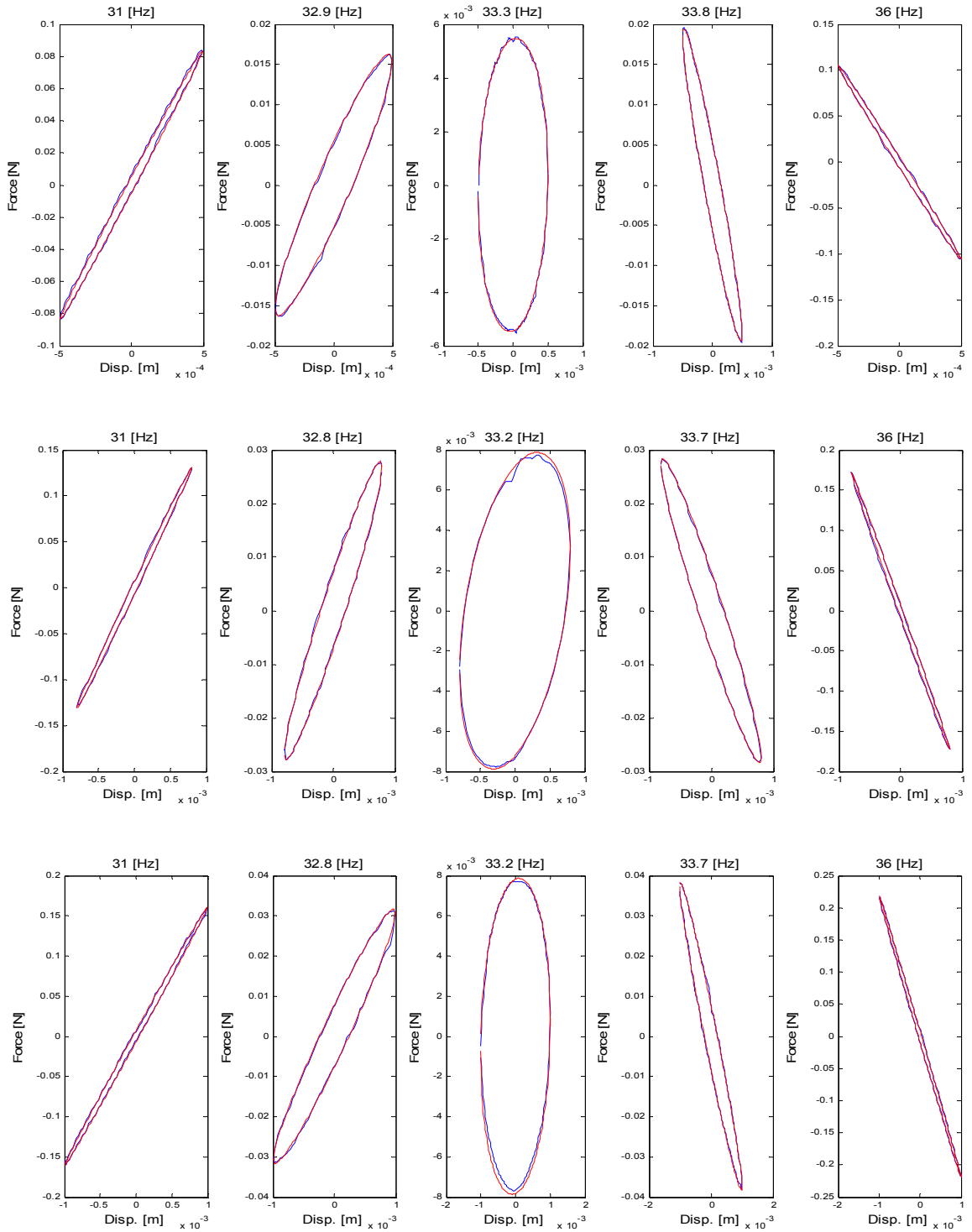
**Figure 5.18.** Optimization convergence plots for beam sample 1



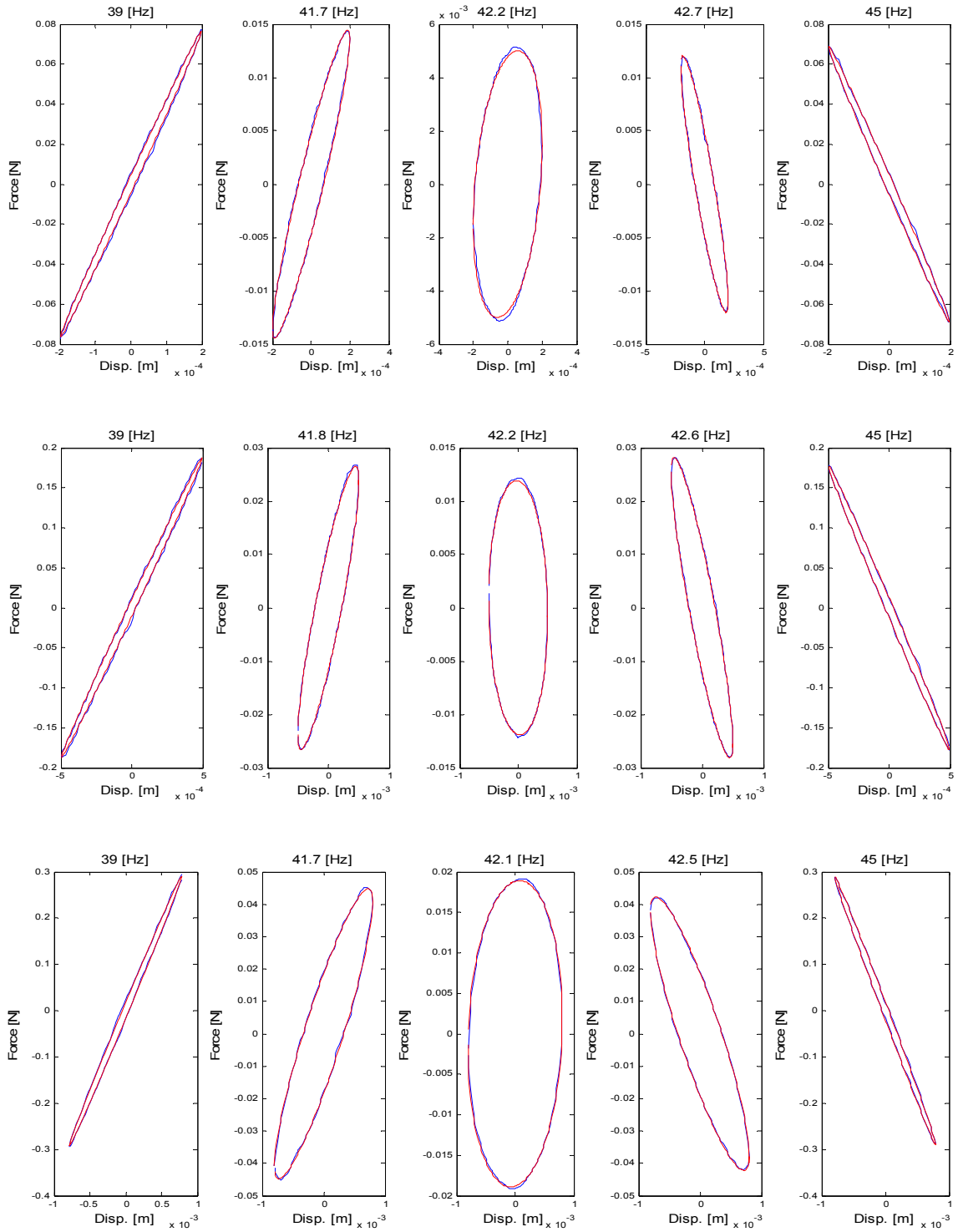
**Figure 5.19.** Optimization convergence plots for beam sample 2



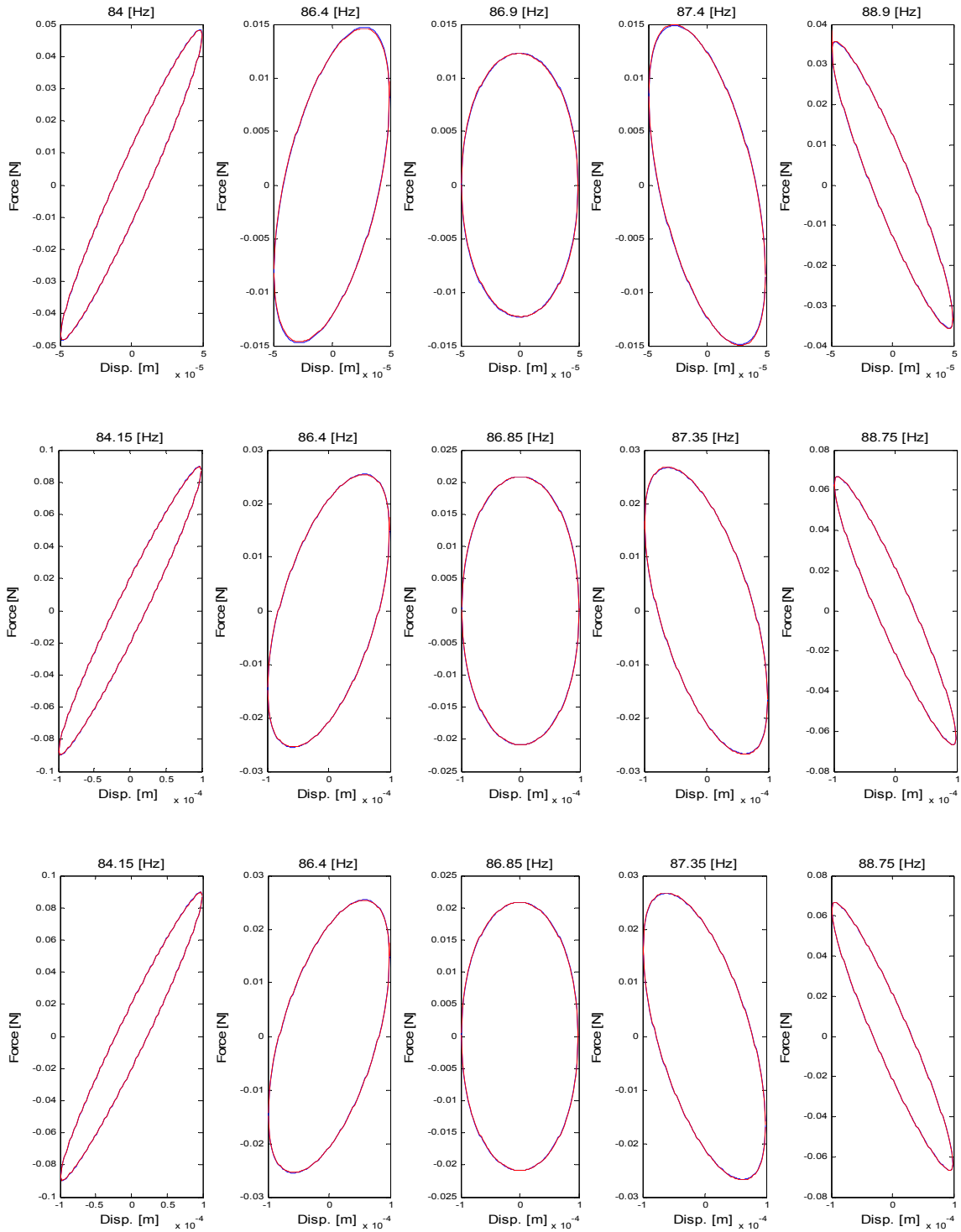
**Figure 5.20.** Optimization convergence plots for sample beam 3



**Figure 5.21.** Optimization comparison plots for some of the hysteresis loops of sample 1; 1st.row: 0.5mm; 2nd.row: 0.8mm; 3rd.row: 1.0mm; — trial, — model



**Figure 5.22.** Optimization comparison plots for some of the hysteresis loops of sample 2; 1st.row: 0.2mm; 2nd.row: 0.5mm; 3rd.row: 0.8mm; — trial, — model



**Figure 5.23.** Optimization comparison plots for some of the hysteresis loops of sample 3; 1st.row: 0.05mm; 2nd.row: 0.10mm; 3rd.row: 0.15mm; — trial, — model

## CHAPTER 6

### DISCUSSION AND CONCLUSIONS

In this dissertation, a study to identify model parameters related to beams of sandwich composite materials has been presented. The experimental testing apparatus and some representative hysteresis loops that were obtained have been also presented. A simplified mathematical model for the representation of the dynamic response of a honeycomb-core sandwich composite material beam sample under a displacement excitation at its mid-point has been developed and tailored based upon experimental observations of the system behavior.

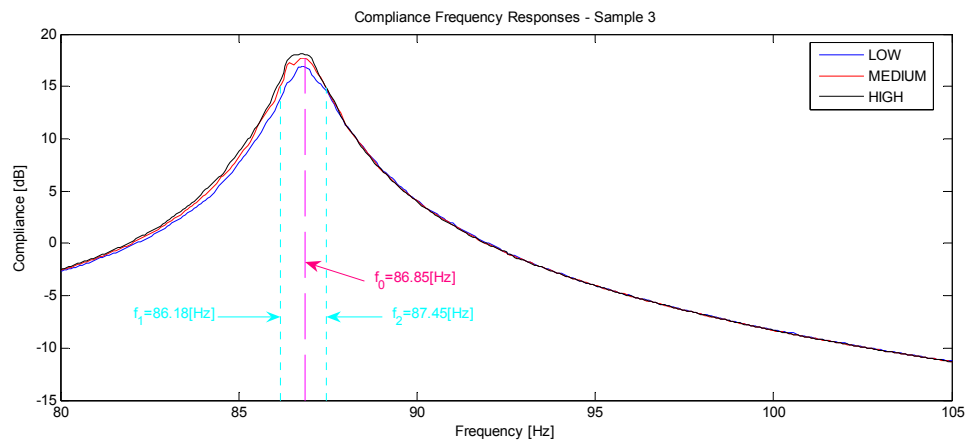
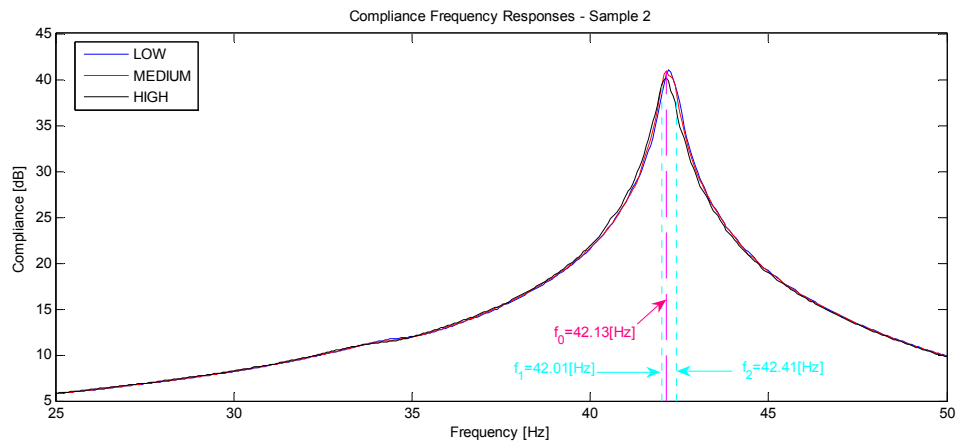
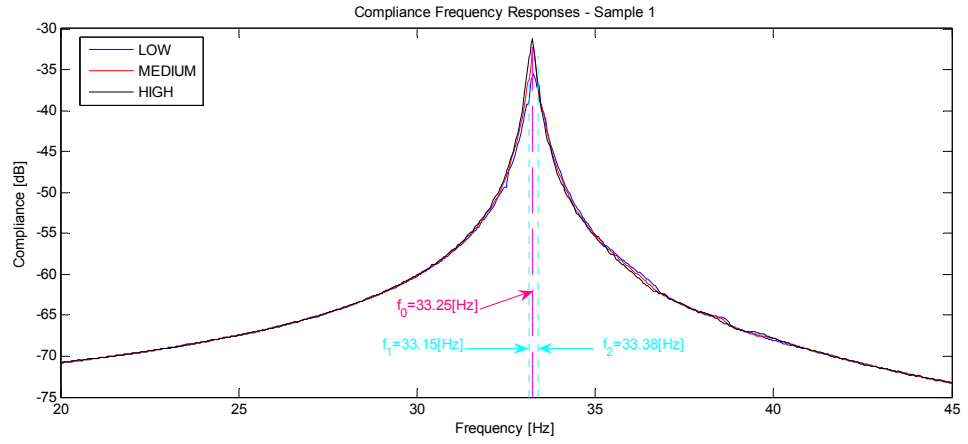
It can be seen from the simulation results obtained from both the PSO and GA, over a set of computer-generated hysteresis loops (*Synth* set), that the parametric solutions that were found are usually quite close (among different runs) to the actual values of the parameters when the global optimum was known. In addition, it has been observed that in spite of the addition of high levels of Gaussian random noise to the simulated signals, the solutions in these cases still remain close to the global optimum. However, the noise in many cases in the simulation study was deliberately set at very high levels, and the solutions obtained under these conditions are still quite consistent and acceptable for most applications.

The experimentally identified parameters are also quite consistent and lend confidence to both the proposed mathematical model and to the identification algorithms.

From the experimental results obtained with the optimization techniques in Tables 5.20 to 5.25, it can be observed that nominal values of the damping ratio for each sample (i.e. for samples 1, 2, and 3, at “medium” excitation amplitudes) are 0.0035, 0.0047, and 0.0073, respectively, when the results are evaluated using Eq. (4.8). These damping ratios are relatively high even for composite materials and this phenomenon is probably caused by the relative motion at the various interfaces within the composite and their interaction with the foam filling in the honeycomb core structure. This makes this type of material a good choice when low weight and good vibration energy dissipation performance are of importance for the construction of a structure, such as what is needed in aircrafts.

Considering sample 1 as a standard for comparison with the other samples utilized, this result shows that the addition of an extra face layer (represented by sample 2) increased the damping ratio by about 34%, whereas doubling the core thickness (represented by sample 3) led to an increase of the damping ratio by about 108%.

The resulting damping ratios from the hysteresis loop approach are consistent with the results obtained by using the half-power bandwidth method, described in section 2.2. Compliance frequency response plots for each beam sample, near their first resonance frequency, are shown in Fig. 6.1. Each of these plots are shown as an overlay of the frequency response of a beam at different excitation levels, generically named “low”, “medium”, and “high”, respectively. The resonance frequency (in magenta color) and the -3dB frequency points (in cyan color), for a medium excitation amplitude, are also shown in each case, whose values (when evaluated into Eq. (2.7)) lead to similar damping ratio results as with the hysteresis loops technique.



**Figure 6.1.** Overlay compliance FRF plots of all 3 beam samples utilized, for low, medium, and high excitation amplitudes

It can be also verified analytically from Fig. 6.1 that the higher damping ratios correspond to the cases with the smoothest resonance peaks. This result was expected, bearing in mind the explanation in section 2.2, concerning the relationship between damping and the sharpness of the resonance peaks.

With regard to the effect of Gaussian random noise in the system response signal (force, in this case), it was expected that, in general, both optimization algorithms would perform well in the search for an “average” hysteresis loop. The explanation for this lies in the nature of Gaussian random noise, as explained next. The total mean square error of a specific candidate solution (CS) to a set of noisy sample points (from a trial set of size  $N$ ) is given by:

$$MSE_{CS} = \frac{1}{N} \sum_{i=1}^N (\hat{F}_i - f_i)^2, \quad (6.1)$$

where  $\hat{F}_i = (F_i + n_i)$  is a noise-contaminated trial sample point,  $F_i$ =clean trial point  $i$ ,  $n_i$ =Gaussian random noise of point  $i$ , and  $f_i$ =corresponding sample point  $i$  from the model solution. We have then, omitting the  $i$  subscripts, for convenience:

$$[\hat{F} - f]^2 = [(F + n) - f]^2 = [F - f]^2 + 2(F - f) \cdot n. \quad (6.2)$$

Since Gaussian random noise has an expected value (i.e. the estimated average for an infinitely long data set) of  $E(n)=0$ , and a standard deviation of 1, we have:

$$MSE_{CS} = \frac{1}{N} \sum_{i=1}^N (F_i - f_i)^2 + 2(F_i - f_i) \cdot E(n) = \frac{1}{N} \sum_{i=1}^N (F_i - f_i)^2. \quad (6.3)$$

This result shows that, in theory, Gaussian random noise present in a system response signal should not interfere in the quality of the system identification results. In other words, the optimization algorithms should be able to find solutions similar to those of the

corresponding noiseless signals. This conclusion was verified from the results of the *Synth* computer-generated hysteresis set, shown in Tables 5.5 to 5.12, and Figs. 5.10 to 5.13.

An interesting optimization performance result can be observed from all the convergence plots shown in this work, in Figs. 5.6 to 5.9 (for the *Synth* hysteresis set), and 5.18 to 5.20 (for the experimental data). In spite of the fact that both optimization algorithms used provided sufficiently accurate results, and relatively low standard deviations, PSO produced a much better search performance than GA. This may be verified by observing the slopes of the convergence plots in the initial iterations, which are much steeper in PSO than in GA, in all cases. This means that PSO is able to find lower-error candidate solutions much faster than GA. Besides, PSO is much easier to implement than GA, as it requires fewer steps to update the candidate solutions during iterations, therefore requiring several fewer lines of computer code. This makes PSO a definite choice between both algorithms, which is in agreement with what has been reported by several researchers<sup>[37-38]</sup>, as explained in section 2.4.

The large scale of the present study (considering the amount of hysteresis loops simultaneously used in each test), and the relatively high dimensionality of the model, makes it a difficult problem to solve for any optimization algorithm, in general. It seems fair then to consider that both, GA and PSO, have been shown to be good choices for solving the problems presented in this study, with low error values and few iterations.

Some fundamental contributions of the presented research work can be identified as follows:

- Developed techniques for the determination of the damping characteristics of sandwich composite materials from hysteresis curves.
- Created computer programs for the implementation of different customized heuristic optimization methods, such as GA and PSO.
- Tested the performance and reliability of these optimization methods when used for system characterization of the semi-empirical model that has been developed.
- Validated the experimental results utilizing these two different optimization methods.
- Analyzed the related theory in order to develop all the necessary mathematical expressions needed for the present research.

Finally, some potentially important applications of this research work are:

- The extraction of important information of the physical/mechanical properties of the vibratory dynamics of composite materials (or any type of material, in general) that may be used for design and for further research in vibration analysis.
- The ability to more accurately predict the dynamical behavior in time of such materials through computer simulation once the system has already been characterized (parameters of the system are finally known).
- A better understanding of the application of optimization methods in solving engineering problems, focused on two powerful techniques, GA and PSO, which have shown to be of popular usage, and proven to be successful in a wide array of problems. This knowledge may be utilized for future research as well, in order to substantially reduce the effort required to implement such techniques, this way being

able to create optimized computer programs, thus reducing the CPU time required to converge to good solutions.

### **6.1. Suggestions for Future Research**

1. Although the present work deals with the study of a generalized set of honeycomb-core sandwich composite beams, several more samples should be studied (following the presented approach), having different combinations of core materials and structures, number of face layers, and dimensions. Composite materials of different nature should also be studied, outside of the honeycomb-core category.
2. A study on the variation of the optimization parameters for both algorithms should be carried out, in order to analyze their effect on the optimization performances. More specifically, for GA, different population sizes, number of iterations, parent selection techniques, mutation rates, and survivor selection strategies should be tested, in order to find combinations that lead to fewer function evaluations and less CPU time required. As for PSO, different swarm sizes, in conjunction with different cognition and social component factor values should also be tested.
3. In order to improve the performance of the time consuming task of finding optimal solutions for large sets of trial hysteresis loops, a networked collaboration of computer systems may be implemented. In other words, several networked computers could run the same optimization routines simultaneously, sharing all the best partial solutions found, then incorporating them into their own populations and continue their search. Thus, the time needed to get optimal solutions may be dramatically reduced.

## REFERENCES

- <sup>1</sup> Renji, K., and Shankar Narayan, S. Loss Factors of Composite Honeycomb Sandwich Panels, *J. of Sound and Vib.*, **250**(4), 745-761, (2002).
- <sup>2</sup> Hornig, K.H., and Flowers, G.T. Parameter Characterisation of the Bouc/Wen Mechanical Hysteresis Model for Sandwich Composite Materials using Real Coded Genetic Algorithms, *Intl. J. of Acoustics and Vib. (IJAV)*, **10**(2), 73-81, (2005).
- <sup>3</sup> Nilsson, E. Some Dynamic Properties of Honeycomb Panels, Report, Royal Inst. of Tech., Stockholm, Sweden, (2000).
- <sup>4</sup> Chandra, R., Singh, S.P., and Gupta, K. Damping Studies in Fiber-Reinforced Composites - a Review, *Composite Structures*, **46**(1), 41-51, (1999).
- <sup>5</sup> Sain, P.M., Sain, M.K., and Spencer, B.F. Models of Hysteresis and Application to Structural Control, *Proc. of the Amer. Control Conf.*, Albuquerque, NM, 16-20, (1997).
- <sup>6</sup> Krasnosel'skii, M.A., and Pokorovskii, A.V. *Systems with Hysteresis*, Springer-Verlag, New York, (1989).
- <sup>7</sup> Chua, L.O., and Bass, S.C. A Generalized Hysteresis Model, *IEEE Trans. Circuit Theory*, **19**, 36-48, (1972).
- <sup>8</sup> Preisach, P. Über die Magnetische Nachwirkung, *Zeitschrift für Physik*, **94**, 277-302, (1938).
- <sup>9</sup> Bouc, R. Forced Vibration of Mechanical Systems with Hysteresis, *Proc. of the 4th Conf. on Nonlinear Oscillation, Prague, Czechoslovakia*, (1967).
- <sup>10</sup> Wen, Y.K. Method for Random Vibration of Hysteretic Systems, *J. Engr. Mech.* **102**, 249-263, (1976).

- <sup>11</sup> Spencer, B.F. *Reliability of Randomly Excited Hysteretic Structures*, Springer-Heidelberg, New York, (1986).
- <sup>12</sup> Ni, Y.Q., Ko, J.M., and Wong, C.W. Identification of Non-Linear Hysteretic Isolators form Periodic Vibration Tests, *J. of Sound and Vib.*, **217** (4), 737-756 (1998).
- <sup>13</sup> Constantinou, M.C., and Tadjbakhsh, I.G. Hysteretic Dampers in Base Isolation: Random Approach, *J. of Struct. Eng., ASCE* **111** (ST4), 705-721, (1998).
- <sup>14</sup> Smyth, A.W., Masri, S.F., Kosmatopoulos, E.B., Chassiakos, A.G., and Caughey, T.K. Development of Adaptive Modeling Techniques for Non-linear Hysteretic Systems, *Intl. J. of Non-Linear Mech.*, **37**(8), 1435-1451, (2002).
- <sup>15</sup> Heine, C.P. Simulated Response of Degrading Hysteretic Joints With Slack Behavior, Ph.D. Dissertation, Virginia Polytech. Inst. and State Univ., (2001).
- <sup>16</sup> Macioce, P. Viscoelastic Damping 101, *Sound and Vibration Magazine*, **4**, 4-5, (2003).
- <sup>17</sup> Meirovitch, L. *Elements of Vibration Analysis*, McGraw-Hill, New York, (1986).
- <sup>18</sup> Sun, C.T., and Lu, Y.P. *Vibration Damping of Structural Elements*, Prentice Hall, New Jersey, (1995).
- <sup>19</sup> Zener, C. *Elasticity and Inelasticity of Metals*, University of Chicago Press, Chicago, (1948).
- <sup>20</sup> Crandall, S.H. The Role of Damping in Vibration Theory, *J. of Sound and Vib.*, **11**(1), 3-18, (1970).
- <sup>21</sup> Lazan, B.J. *Damping of Materials and Members in Structural Mechanics*, Pergamon Press, UK, (1968).
- <sup>22</sup> Hooker, R.J. On the Interpretation of Biaxial Hysteresis Loops, *J. of Sound and Vib.*, **76**(3), 463-466, (1981).
- <sup>23</sup> So, C.K., Lai, T.C., and Tse, P.C. The Measurement of Material Damping by Free-Vibration Technique with Periodic Excitation, *Experimental Techniques*, **14**(3), 41-42, (1990).

- <sup>24</sup> McDaniel, J.G., Dupont, P., and Salvino, L.A. Wave Approach to Estimating Frequency-Dependent Damping under Transient Loading, *J. of Sound and Vib.*, **231**(2), 433-449, (2000).
- <sup>25</sup> Brown, K.T., and Norton, M.P. Some Comments on the Experimental Determination of Modal Densities and Loss Factors for Statistical Energy Analysis Applications, *J. of Sound and Vib.*, **102**, 588-594, (1985).
- <sup>26</sup> Boyaci, H. Vibrations of Stretched Damped Beams under Non-Ideal Boundary Conditions, *Sādhanā*, **31**(1), 1-8, (2006).
- <sup>27</sup> Qiao, S.L., Pilipchuk, V.N., and Ibrahim, R.A. Modeling and Simulation of Elastic Structures with Parameter Uncertainties and Relaxation of Joints, *J. of Vib. and Acoustics*, **123**, 45-52, (2001).
- <sup>28</sup> Liu, G.R., Han, X., and Lam, K.Y. A Combined Genetic Algorithm and Nonlinear Least Squares Method for Material Characterization using Elastic Waves, *Comput. Methods in Appl. Mech. and Eng.*, **191**(17), 1909-1921, (2002).
- <sup>29</sup> Matsuoka, T., Yamamoto, S., and Takahara, M. Prediction of Structures and Mechanical Properties of Composites using a Genetic Algorithm and Finite Element Method, *J. of Materials Science*, **36**, 27-33, (2001).
- <sup>30</sup> Mac, A., and Gurba, W. Genetic Algorithms and Finite Element Analysis in Optimization of Composite Structures, *Composite Structures*, **54**, 275-281, (2001).
- <sup>31</sup> Gantovnik, V.B., Gürdal, Z., and Watson, L.T. A Genetic Algorithm with Memory for Optimal Design of Laminated Sandwich Composite Panels, *Composite Structures*, **58**, 513-520, (2002).
- <sup>32</sup> Kovacs, G., Groenwold, A.A., Jarmai, K., and Farkas, J. Analysis and Optimum Design of Fibre-Reinforced Composite Structures, *Struct. and Multidisc. Optimization*, **28**(2), 170-179, (2004).
- <sup>33</sup> Sekishiro, M., and Todoroki, A. Optimizations of Stiffened Composite Panel Using Fractal Branch and Bound Method, *Proc. of the 46th Struct. Dynamics and Materials Conf.*, Austin, TX, 1525-1534, (2005).

- <sup>34</sup> Kyprianou, A., and Worden, K. Identification of Hysteretic Systems using the Differential Evolution Algorithm, *Proc. of the 23rd Intl. Conf. on Noise and Vibration Engineering (ISMA)*, Leuven, Belgium, 557-568, (1998).
- <sup>35</sup> Zhang, X., Huang, Y., Liu, Y., Wang, X., and Gao, F. Method Identifying the Parameters of Bouc-Wen Hysteretic Nonlinear Model Based on Genetic-Algorithm, *Proc. of the IEEE Intl. Conf. on Intelligent Proc. Systems (ICIPS)*, Beijing, China, 602-605, (1997).
- <sup>36</sup> Peng, J.Y., Li, H., and Suzuki, Y. Modeling of Nonlinear Hysteresis with Pinching, *Journal of Shenyang Jianzhu University*, **21**(4), 325-328, (2005).
- <sup>37</sup> Fu, J., Hou, C., and Dou, L. Application of Particle Swarm Optimization Algorithm in the Design of Multilayered Planar Shielding Body, *Chinese Journal of Electronics*, **14**(2), 357-360, (2005).
- <sup>38</sup> Chen, J., Ge, R., and Wei, J. Optimization of the Reliability of Laminated Plates Based on the PSO, *J. of Huazhong Univ. of Sc. and Technology*, **34**(4), 96-98, (2006).
- <sup>39</sup> Holland, J.H. *Adaptation in Natural and Artificial Systems*, Ann Arbor, Michigan, The University of Michigan Press, (1975).
- <sup>40</sup> Goldberg, D.E. *Genetic Algorithms in Search, Optimization, and Machine Learning*, Reading, MA, Addison-Wesley, (1989).
- <sup>41</sup> Haupt, R.L., and Haupt, S.E. *Practical Genetic Algorithms, 2nd Ed.*, Wiley, New York, (2004).
- <sup>42</sup> Zilouchian, A., and Jamshidi, M. (Editors). *Intelligent Control Systems Using Soft Computing Methodologies*, CRC Press, USA, (2001).
- <sup>43</sup> Michalewicz, Z. *Genetic Algorithms + Data Structures = Evolution Programs, 3rd Ed.*, Springer-Verlag, New York, (1996).
- <sup>44</sup> Kennedy, J., and Eberhart, R. Particle Swarm Optimization, *Proc. of the IEEE Intl. Conf. on Neural Networks*, Perth, Australia: IEEE Service Center, Piscataway, NJ, 1942-1948, (1995).

- <sup>45</sup> Shi, Y.H., and Eberhart, R.C. Parameter Selection in Particle Swarm Optimization, *Proc. of the 7th Annual Conf. on Evolutionary Programming*, New York, NY, 591-600, (1998).
- <sup>46</sup> Clerc, M. The Swarm and the Queen: Towards a Deterministic and Adaptive Particle Swarm Optimization, *Proc. of the 1999 ICEC*, Washington, DC, 1951-1957, (1999).
- <sup>47</sup> Carlisle, A., and Dozier, G. An Off-The-Shelf PSO, *Proc. of the 2001 Workshop on Particle Swarm Optimization*, Indianapolis, IN, 1-6, (2001).

## **APPENDIX A**

# **PROGRAMS CREATED IN MATLAB R2006a FOR THE IMPLEMENTATION OF GENETIC ALGORITHMS (GA) AND PARTICLE SWARM OPTIMIZATION (PSO) FOR SYSTEM IDENTIFICATION OF SANDWICH COMPOSITE MATERIALS**

```

%Optimization Program for the Hysteresis model using GA
%(c)2006 - Klaus H. Hornig
%


---


clear all; clc;

%Parameters: [xi0, S_xi, w0, Sw]
pb=[0, .1; -10, 10; 100, 500; -10000, 10000]; %parameter bounds
np=size(pb,1); %np=number of parameters

tspan=.4*ones(1,120); %time span
Meq=36.8e-3; %equivalent mass (sample #2)

%GA operators
ni=18; %Number of iterations of the whole GA code (main loop)
nipop=6000; %Number of initial population
rempop=500; %Number of remaining population
alphaBLX=0.3; %alpha parameter for X-over
mutrate=.05; %Mutation rate
mutamount=.3; %Gaussian mutation amount
addpopmut=.2; %Additional population mutation amount
addpoplevel=4; %Additional population "add-on" level (L)
interppts=100; %Resampling resolution (initial interpolation)

randn('state',0);
rand('state',0);

%Generate the initial population
ipop=genipop(pb,nipop);
ipop=[ipop,zeros(nipop,1)];

%load experimental loops (files that contain both variables, HexpT and HexpB,
%for TOP and BOTTOM parts, respectively)
hysloopsnames=['SAMPLE3_02mm_370Hz';... %0.2mm
'SAMPLE2_02mm_380Hz'; 'SAMPLE2_02mm_385Hz'; 'SAMPLE2_02mm_390Hz';...
'SAMPLE2_02mm_395Hz'; 'SAMPLE2_02mm_400Hz'; 'SAMPLE2_02mm_402Hz';...
'SAMPLE2_02mm_404Hz'; 'SAMPLE2_02mm_406Hz'; 'SAMPLE2_02mm_408Hz';...
'SAMPLE2_02mm_410Hz'; 'SAMPLE2_02mm_412Hz'; 'SAMPLE2_02mm_414Hz';...
'SAMPLE2_02mm_415Hz'; 'SAMPLE2_02mm_416Hz'; 'SAMPLE2_02mm_417Hz';...
'SAMPLE2_02mm_418Hz'; 'SAMPLE2_02mm_419Hz'; 'SAMPLE2_02mm_420Hz';...
'SAMPLE2_02mm_421Hz'; 'SAMPLE2_02mm_422Hz'; 'SAMPLE2_02mm_423Hz';...
'SAMPLE2_02mm_424Hz'; 'SAMPLE2_02mm_425Hz'; 'SAMPLE2_02mm_426Hz';...
'SAMPLE2_02mm_427Hz'; 'SAMPLE2_02mm_428Hz'; 'SAMPLE2_02mm_429Hz';...
'SAMPLE2_02mm_430Hz'; 'SAMPLE2_02mm_432Hz'; 'SAMPLE2_02mm_434Hz';...
'SAMPLE2_02mm_436Hz'; 'SAMPLE2_02mm_438Hz'; 'SAMPLE2_02mm_440Hz';...
'SAMPLE2_02mm_442Hz'; 'SAMPLE2_02mm_444Hz'; 'SAMPLE2_02mm_450Hz';...
'SAMPLE2_02mm_455Hz'; 'SAMPLE2_02mm_460Hz'; 'SAMPLE2_02mm_470Hz';...
'SAMPLE2_05mm_370Hz';... %0.5mm
'SAMPLE2_05mm_380Hz'; 'SAMPLE2_05mm_385Hz'; 'SAMPLE2_05mm_390Hz';...
'SAMPLE2_05mm_395Hz'; 'SAMPLE2_05mm_400Hz'; 'SAMPLE2_05mm_402Hz';...
'SAMPLE2_05mm_404Hz'; 'SAMPLE2_05mm_406Hz'; 'SAMPLE2_05mm_408Hz';...
'SAMPLE2_05mm_410Hz'; 'SAMPLE2_05mm_412Hz'; 'SAMPLE2_05mm_414Hz';...
'SAMPLE2_05mm_415Hz'; 'SAMPLE2_05mm_416Hz'; 'SAMPLE2_05mm_417Hz';...
'SAMPLE2_05mm_418Hz'; 'SAMPLE2_05mm_419Hz'; 'SAMPLE2_05mm_420Hz';...
'SAMPLE2_05mm_421Hz'; 'SAMPLE2_05mm_422Hz'; 'SAMPLE2_05mm_423Hz';...
'SAMPLE2_05mm_424Hz'; 'SAMPLE2_05mm_425Hz'; 'SAMPLE2_05mm_426Hz';...
'SAMPLE2_05mm_427Hz'; 'SAMPLE2_05mm_428Hz'; 'SAMPLE2_05mm_429Hz';...
'SAMPLE2_05mm_430Hz'; 'SAMPLE2_05mm_432Hz'; 'SAMPLE2_05mm_434Hz';...
'SAMPLE2_05mm_436Hz'; 'SAMPLE2_05mm_438Hz'; 'SAMPLE2_05mm_440Hz';...

```

```

'SAMPLE2_05mm_442Hz'; 'SAMPLE2_05mm_444Hz'; 'SAMPLE2_05mm_450Hz'; ...
'SAMPLE2_05mm_455Hz'; 'SAMPLE2_05mm_460Hz'; 'SAMPLE2_05mm_470Hz'; ...
'SAMPLE2_08mm_370Hz'; ... %0.8mm
'SAMPLE2_08mm_380Hz'; 'SAMPLE2_08mm_385Hz'; 'SAMPLE2_08mm_390Hz'; ...
'SAMPLE2_08mm_395Hz'; 'SAMPLE2_08mm_400Hz'; 'SAMPLE2_08mm_402Hz'; ...
'SAMPLE2_08mm_404Hz'; 'SAMPLE2_08mm_406Hz'; 'SAMPLE2_08mm_408Hz'; ...
'SAMPLE2_08mm_410Hz'; 'SAMPLE2_08mm_412Hz'; 'SAMPLE2_08mm_414Hz'; ...
'SAMPLE2_08mm_415Hz'; 'SAMPLE2_08mm_416Hz'; 'SAMPLE2_08mm_417Hz'; ...
'SAMPLE2_08mm_418Hz'; 'SAMPLE2_08mm_419Hz'; 'SAMPLE2_08mm_420Hz'; ...
'SAMPLE2_08mm_421Hz'; 'SAMPLE2_08mm_422Hz'; 'SAMPLE2_08mm_423Hz'; ...
'SAMPLE2_08mm_424Hz'; 'SAMPLE2_08mm_425Hz'; 'SAMPLE2_08mm_426Hz'; ...
'SAMPLE2_08mm_427Hz'; 'SAMPLE2_08mm_428Hz'; 'SAMPLE2_08mm_429Hz'; ...
'SAMPLE2_08mm_430Hz'; 'SAMPLE2_08mm_432Hz'; 'SAMPLE2_08mm_434Hz'; ...
'SAMPLE2_08mm_436Hz'; 'SAMPLE2_08mm_438Hz'; 'SAMPLE2_08mm_440Hz'; ...
'SAMPLE2_08mm_442Hz'; 'SAMPLE2_08mm_444Hz'; 'SAMPLE2_08mm_450Hz'; ...
'SAMPLE2_08mm_455Hz'; 'SAMPLE2_08mm_460Hz'; 'SAMPLE2_08mm_470Hz'];

load efSample2; %load variable "ef", with the excitation frequency of each loop

numloops=length(ef);
userweights=ones(1,numloops);

for loopsload=1:numloops %create cell array, storing all loops {TOP,BOTTOM}
load(hysloopsnames(loopsload,:));
[HexpT,HexpB]=reinterpolate(HexpT,HexpB,interppts); %reinterpolate loops
maxF(loopsload)=max(abs([HexpT(:,2);HexpB(:,2)]));
rnT=(maxF(loopsload)*noiselevel/100)*randn(size(HexpT,1),1);
rnB=(maxF(loopsload)*noiselevel/100)*randn(size(HexpB,1),1);
HexpT=[HexpT(:,1),HexpT(:,2)+rnT];
HexpB=[HexpB(:,1),HexpB(:,2)+rnB];
hysteresis_loops{loopsload,1}=HexpT;
hysteresis_loops{loopsload,2}=HexpB;
end
maxFall=max(maxF);

for i=1:numloops %get peak displacement for each loop
Xpk(i)=max(abs([hysteresis_loops{i,1}(:,1);hysteresis_loops{i,2}(:,1)]));
amplitudescaling(i)=maxFall/maxF(i);
end

loopswweights=userweights.*amplitudescaling;

nap=0;
for lvl=1:addpoplevel %calculate number of additional population (AddPopNumber)
nap=nap+nchoosek(np,lvl);
end

%CALCULATE FITNESS FOR EACH CHROMOSOME
for i=1:nipop
fprintf('Calculating fitness for chromosome N°: %g\n',i);
p=ipop(i,1:np);
lasterr='a';
while lasterr~=' '
lasterr=' ';
try
ipop(i,np+1)=0;
for j=1:numloops
HexpT=hysteresis_loops{j,1};

```



```

end
%check to avoid a parent to mate with itself
for i=1:rempop/2 %because there are rempop/2 couples
    while mates(2*i-1)==mates(2*i)
        temprandmate=rand;
        for ii=1:rempop
            jj=1; %initialize variable 'jj'
            while temprandmate>cumprob(jj)
                jj=jj+1;
            end
            mates(2*i)=jj;
        end %for ii
    end %while mates
end %for i

%MATE THE PARENTS
offspring=zeros(rempop,np); %initialize variable "offspring"
for i=1:(rempop/2) %couple number
    crosspnt=ceil(np*rand); %CROSSOVER POINT!!!
    for parameter=1:np
        if parameter<crosspnt
            offspring(2*i-1,parameter)=ipop(mates(2*i-1),parameter);
            offspring(2*i,parameter)=ipop(mates(2*i),parameter);
        end
        if parameter==crosspnt %apply BLX-alphaBLX blend X-over
            temprand=(1+alphaBLX)*rand;
            offspring(2*i-1,parameter)=ipop(mates(2*i-1),parameter)...
                -temprand*(ipop(mates(2*i-1),parameter)...
                    -ipop(mates(2*i),parameter));
            %check if out of bounds
            if offspring(2*i-1,parameter)<pb(parameter,1) ||...
                offspring(2*i-1,parameter)>pb(parameter,2)
                offspring(2*i-1,parameter)=ipop(mates(2*i-1),parameter);
            end
            offspring(2*i,parameter)=ipop(mates(2*i),parameter)...
                +temprand*(ipop(mates(2*i-1),parameter)...
                    -ipop(mates(2*i),parameter));
            %check if out of bounds
            if offspring(2*i,parameter)<pb(parameter,1) ||...
                offspring(2*i,parameter)>pb(parameter,2)
                offspring(2*i,parameter)=ipop(mates(2*i),parameter);
            end
        end
        if parameter>crosspnt
            offspring(2*i-1,parameter)=ipop(mates(2*i),parameter);
            offspring(2*i,parameter)=ipop(mates(2*i-1),parameter);
        end
    end %for parameter
end %for i

%Create a new population of (Gaussian) mutated versions of BC
APcount=0; %additional population count
addpop=ones(nap,1)*BC;
for j=1:addpoplevel
    C=nchoosek(1:np,j)';
    for ap=1:size(C,2)
        APcount=APcount+1;
        for i=C(:,ap)'
            newvalue=addpop(APcount,i)+randn*addpopmut*(pb(i,2)-pb(i,1));

```

```

        if newvalue>pb(i,1)&&newvalue<pb(i,2) %mutate just within bounds
            addpop(APcount,i)=newvalue;
        end
    end
    if addpop(APcount,')==BC %if no change from BC..
        addpop(APcount,)=genipop(pb,1); %..then generate new chromosome
    end
end
end

%ADD OFFSPRING TO THE POPULATION
ipop2=[offspring,zeros(rempop,1);addpop,zeros(nap,1);ipop];

%SELECT GENES TO MUTATE
mutations=round(rempop*np*mutrate); %number of mutations
randmutparams=rand(mutations,2);
randmutparams(:,1)=ceil(randmutparams(:,1).*rempop); %row numbers
randmutparams(:,2)=ceil(randmutparams(:,2).*np); %column numbers

%APPLY GAUSSIAN MUTATION
for i=1:mutations
    rpr=randmutparams(i,1); %random parameter row (chromosome)
    rpc=randmutparams(i,2); %random parameter column (parameter)
    newvalue=ipop2(rpr, rpc)+randn*mutamount*(pb(rpc,2)-pb(rpc,1));
    if newvalue>pb(rpc,1)&&newvalue<pb(rpc,2) %mutate just if within bounds
        ipop2(rpr, rpc)=newvalue;
    end
end

%RANK CHROMOSOMES
for par=1:rempop+nap %chromosome index
    fprintf('%g) Calculating fitness for chromosome N°: %g\n',loop,par);
    p=ipop2(par,1:np);
    lasterr='a';
    while lasterr~=' '
        lasterr=' ';
        try
            ipop2(par,np+1)=0;
            for j=1:numloops
                HexpT=hysteresis_loops{j,1};
                HexpB=hysteresis_loops{j,2};
                [NOS,dummy]=size([HexpT;HexpB]); %NOS=Number Of Samples
                ipop2(par,np+1)=ipop2(par,np+1)+loopsweights(j)*...
                    erreval(HexpT,HexpB,p,ef(j),Xpk(j),Meq,tspan(j))/NOS;
            end %for j
            fprintf('%g\n\n',ipop2(par,np+1));
        catch
            fprintf('\nCouldn't calculate. Generating new chromosome\n\n');
            ipop2(par,1:np)=genipop(pb,1);
            p=ipop2(par,1:np);
            lasterr='a';
        end %try-catch
    end %while
end %for par

APbestrow=find(ipop2(rempop+1:rempop+nap,np+1)==...
    min(ipop2(rempop+1:rempop+nap,np+1)));

```

```

%Update ipop2, to keep the offspring, best AddPop, and original pop...
ipop2=[ipop2(1:rempop,:);...
      ipop2(rempop+APbestrow,:);ipop2(rempop+nap+1:2*rempop+nap,:)];

%SORT THE CHROMOSOMES
ipop2=sortrows(ipop2,np+1);
ipop=ipop2(1:rempop,:);
BC=ipop(1,1:np);
BCf=ipop(1,np+1);

fprintf('The best chromosome so far is:\n-----\n');
bp_string_values=sprintf('[ %g ',BC(1));
if np>1
    for i=2:np
        bp_string_values=[bp_string_values,sprintf(' , %g',BC(i))];
    end
end
fprintf([bp_string_values,' ]\n\n\n']);

%Fitness history value
fhv(loop+1)=BCf;

save allvariables;

%*****
end %           END OF MAIN LOOP  !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!*
%*****

```

---

```

%Optimization Program for the Hysteresis model using PSO
%(c)2006 - Klaus H. Hornig
%


---


clear all; clc;

%Parameters: [xi0, S_xi, w0, Sw]
pb=[0, .1; -10,10; 100,500; -10000,10000]; %parameter bounds (for constant damping)
np=size(pb,1); %np=number of parameters

tspan=.4*ones(1,120);
Meq=36.8e-3; %equivalent mass (sample #2)

%GA operators constants
ni=18; %Number of iterations of the whole GA code (main loop)
nipop=6000; %Number of initial population
rempop=500; %Number of remaining population
interppts=100; %resampling resolution (initial interpolation)
addpopmut=.2; %Additional population mutation amount
addpoplevel=4; %max. number of parameters to be changed in addpop

randn('state',0);
rand('state',0);

%Initialize PSO variables
Pi=zeros(nipop,np);
Pif=zeros(nipop,1);
Vi=zeros(rempop,np);
phi1=2.05; %PSO cognition factor
phi2=2.05; %PSO social factor
phi=phi1+phi2;
K=2/abs(2-phi-sqrt(phi^2-4*phi)); %PSO constriction factor

%Generate the initial population
ipop=genipop(pb,nipop);
ipop=[ipop,zeros(nipop,1)];

%load experimental loops (files that contain both variables, HexpT and HexpB,
%for TOP and BOTTOM parts, respectively)
hysloopsnames=['SAMPLE3_02mm_370Hz';... %0.2mm
'SAMPLE2_02mm_380Hz'; 'SAMPLE2_02mm_385Hz'; 'SAMPLE2_02mm_390Hz';...
'SAMPLE2_02mm_395Hz'; 'SAMPLE2_02mm_400Hz'; 'SAMPLE2_02mm_402Hz';...
'SAMPLE2_02mm_404Hz'; 'SAMPLE2_02mm_406Hz'; 'SAMPLE2_02mm_408Hz';...
'SAMPLE2_02mm_410Hz'; 'SAMPLE2_02mm_412Hz'; 'SAMPLE2_02mm_414Hz';...
'SAMPLE2_02mm_415Hz'; 'SAMPLE2_02mm_416Hz'; 'SAMPLE2_02mm_417Hz';...
'SAMPLE2_02mm_418Hz'; 'SAMPLE2_02mm_419Hz'; 'SAMPLE2_02mm_420Hz';...
'SAMPLE2_02mm_421Hz'; 'SAMPLE2_02mm_422Hz'; 'SAMPLE2_02mm_423Hz';...
'SAMPLE2_02mm_424Hz'; 'SAMPLE2_02mm_425Hz'; 'SAMPLE2_02mm_426Hz';...
'SAMPLE2_02mm_427Hz'; 'SAMPLE2_02mm_428Hz'; 'SAMPLE2_02mm_429Hz';...
'SAMPLE2_02mm_430Hz'; 'SAMPLE2_02mm_432Hz'; 'SAMPLE2_02mm_434Hz';...
'SAMPLE2_02mm_436Hz'; 'SAMPLE2_02mm_438Hz'; 'SAMPLE2_02mm_440Hz';...
'SAMPLE2_02mm_442Hz'; 'SAMPLE2_02mm_444Hz'; 'SAMPLE2_02mm_450Hz';...
'SAMPLE2_02mm_455Hz'; 'SAMPLE2_02mm_460Hz'; 'SAMPLE2_02mm_470Hz';...
'SAMPLE2_05mm_370Hz';... %0.5mm
'SAMPLE2_05mm_380Hz'; 'SAMPLE2_05mm_385Hz'; 'SAMPLE2_05mm_390Hz';...
'SAMPLE2_05mm_395Hz'; 'SAMPLE2_05mm_400Hz'; 'SAMPLE2_05mm_402Hz';...
'SAMPLE2_05mm_404Hz'; 'SAMPLE2_05mm_406Hz'; 'SAMPLE2_05mm_408Hz';...
'SAMPLE2_05mm_410Hz'; 'SAMPLE2_05mm_412Hz'; 'SAMPLE2_05mm_414Hz';...
'SAMPLE2_05mm_415Hz'; 'SAMPLE2_05mm_416Hz'; 'SAMPLE2_05mm_417Hz';...

```

```

'SAMPLE2_05mm_418Hz'; 'SAMPLE2_05mm_419Hz'; 'SAMPLE2_05mm_420Hz'; ...
'SAMPLE2_05mm_421Hz'; 'SAMPLE2_05mm_422Hz'; 'SAMPLE2_05mm_423Hz'; ...
'SAMPLE2_05mm_424Hz'; 'SAMPLE2_05mm_425Hz'; 'SAMPLE2_05mm_426Hz'; ...
'SAMPLE2_05mm_427Hz'; 'SAMPLE2_05mm_428Hz'; 'SAMPLE2_05mm_429Hz'; ...
'SAMPLE2_05mm_430Hz'; 'SAMPLE2_05mm_432Hz'; 'SAMPLE2_05mm_434Hz'; ...
'SAMPLE2_05mm_436Hz'; 'SAMPLE2_05mm_438Hz'; 'SAMPLE2_05mm_440Hz'; ...
'SAMPLE2_05mm_442Hz'; 'SAMPLE2_05mm_444Hz'; 'SAMPLE2_05mm_450Hz'; ...
'SAMPLE2_05mm_455Hz'; 'SAMPLE2_05mm_460Hz'; 'SAMPLE2_05mm_470Hz'; ...
'SAMPLE2_08mm_370Hz'; ... %0.8mm
'SAMPLE2_08mm_380Hz'; 'SAMPLE2_08mm_385Hz'; 'SAMPLE2_08mm_390Hz'; ...
'SAMPLE2_08mm_395Hz'; 'SAMPLE2_08mm_400Hz'; 'SAMPLE2_08mm_402Hz'; ...
'SAMPLE2_08mm_404Hz'; 'SAMPLE2_08mm_406Hz'; 'SAMPLE2_08mm_408Hz'; ...
'SAMPLE2_08mm_410Hz'; 'SAMPLE2_08mm_412Hz'; 'SAMPLE2_08mm_414Hz'; ...
'SAMPLE2_08mm_415Hz'; 'SAMPLE2_08mm_416Hz'; 'SAMPLE2_08mm_417Hz'; ...
'SAMPLE2_08mm_418Hz'; 'SAMPLE2_08mm_419Hz'; 'SAMPLE2_08mm_420Hz'; ...
'SAMPLE2_08mm_421Hz'; 'SAMPLE2_08mm_422Hz'; 'SAMPLE2_08mm_423Hz'; ...
'SAMPLE2_08mm_424Hz'; 'SAMPLE2_08mm_425Hz'; 'SAMPLE2_08mm_426Hz'; ...
'SAMPLE2_08mm_427Hz'; 'SAMPLE2_08mm_428Hz'; 'SAMPLE2_08mm_429Hz'; ...
'SAMPLE2_08mm_430Hz'; 'SAMPLE2_08mm_432Hz'; 'SAMPLE2_08mm_434Hz'; ...
'SAMPLE2_08mm_436Hz'; 'SAMPLE2_08mm_438Hz'; 'SAMPLE2_08mm_440Hz'; ...
'SAMPLE2_08mm_442Hz'; 'SAMPLE2_08mm_444Hz'; 'SAMPLE2_08mm_450Hz'; ...
'SAMPLE2_08mm_455Hz'; 'SAMPLE2_08mm_460Hz'; 'SAMPLE2_08mm_470Hz'];

load efSample2; %load variable "ef", with the excitation frequency of each loop

numloops=length(ef);
userweights=ones(1,numloops);

for loopsload=1:numloops %create cell array, storing all loops {TOP,BOTTOM}
load(hysloopsnames(loopsload,:));
[HexpT,HexpB]=reinterpolate(HexpT,HexpB,interppts); %reinterpolate loops
maxF(loopsload)=max(abs([HexpT(:,2);HexpB(:,2)]));
rnT=(maxF(loopsload)*noiselevel/100)*randn(size(HexpT,1),1);
rnB=(maxF(loopsload)*noiselevel/100)*randn(size(HexpB,1),1);
HexpT=[HexpT(:,1),HexpT(:,2)+rnT];
HexpB=[HexpB(:,1),HexpB(:,2)+rnB];
hysteresis_loops{loopsload,1}=HexpT;
hysteresis_loops{loopsload,2}=HexpB;
end
maxFall=max(maxF);

for i=1:numloops %get peak displacement for each loop
Xpk(i)=max(abs([hysteresis_loops{i,1}(:,1);hysteresis_loops{i,2}(:,1)]));
amplitudescaling(i)=maxFall/maxF(i);
end

loopsweights=userweights.*amplitudescaling;

nap=0;
for lvl=1:addpoplevel %calculate number of additional population (AddPopNumber)
nap=nap+nchoosek(np,lvl);
end

%CALCULATE FITNESS FOR EACH CHROMOSOME
for i=1:nipop
fprintf('Calculating fitness for particle N°: %g\n',i);
p=ipop(i,1:np);
lasterr='a';
while lasterr~=''

```



```

        phi2*rand2.*(Pg-ipop(par,1:np)); %velocity
    xt=ipop(par,1:np)+Vi(par,:); % temporal x-positions

    for i=1:np %update particle position
        if (xt(i)>pb(i,1)) && (xt(i)<pb(i,2)) %within bounds
            ipop(par,i)=xt(i); %update particle position
        else
            Vi(par,i)=0;
        end
    end

    %Evaluate particle fitness
    fprintf('%g) Calculating fitness for particle N°: %g\n',loop,par);
    p=ipop(par,1:np);
    lasterr='a';
    while lasterr~=' '
        lasterr=' ';
        try
            ipop(par,np+1)=0;
            for j=1:numloops
                HexpT=hysteresis_loops{j,1};
                HexpB=hysteresis_loops{j,2};
                NOS=size([HexpT;HexpB],1); %NOS=Number Of Samples
                ipop(par,np+1)=ipop(par,np+1)+loopsweights(j)*...
                    erreval(HexpT,HexpB,p,ef(j),Xpk(j),Meq,tspan(j))/NOS;
            end %for j
            fprintf('%g\n\n',ipop(par,np+1));

            if ipop(par,np+1)<Pif(par) %Update Pi if necessary
                Pi(par,:)=ipop(par,1:np);
                Pif(par)=ipop(par,np+1);
                if Pif(par)<Pgf %update Pg if necessary (ASYNCHRONOUS PSO)
                    Pg=Pi(par,:);
                    Pgf=Pif(par);
                end
            end %if ipop

        catch
            fprintf('\nCouldn't calculate. Generating new particle\n\n');
            ipop(par,1:np)=genipop(pb,1);
            p=ipop(par,1:np);
            lasterr='a';
        end %try-catch
    end %while

end %for par

%Create a new population of (Gaussian) mutated versions of Pg
APcount=0; %additional population count
addpop=ones(nap,1)*Pg;
for j=1:adddpoplevel
    C=nchoosek(1:np,j)';
    for ap=1:size(C,2)
        APcount=APcount+1;
        for i=C(:,ap)'
            newvalue=addpop(APcount,i)+randn*adddpopmut*(pb(i,2)-pb(i,1));
            if newvalue>pb(i,1)&&newvalue<pb(i,2) %mutate if within bounds
                addpop(APcount,i)=newvalue;
            end
        end
    end
end

```

```

        end
        if addpop(APcount,')==Pg %if no change from Pg..
            addpop(APcount,)=genipop(pb,1); %..then generate new chromosome
        end
    end
end

%Evaluate ADDITIONAL POPULATION fitness
addpop=[addpop,zeros(nap,1)];
for par=1:nap
    fprintf('%g) Calculating fitness for ADD.POP. N°: %g  ***\n',loop,par);
    p=addpop(par,1:np);
    lasterr='a';
    while lasterr~=' '
        lasterr=' ';
        try
            addpop(par,np+1)=0;
            for j=1:numloops
                HexpT=hysteresis_loops{j,1};
                HexpB=hysteresis_loops{j,2};
                NOS=size([HexpT;HexpB],1); %NOS=Number Of Samples
                addpop(par,np+1)=addpop(par,np+1)+loopsweights(j)*...
                    erreval(HexpT,HexpB,p,ef(j),Xpk(j),Meq,tspan(j))/NOS;
            end %for j
            fprintf('%g\n\n',addpop(par,np+1));

            if addpop(par,np+1)<Pgf %update Pg if necessary (ASYNCH. PSO)
                Pg=addpop(par,1:np);
                Pgf=addpop(par,np+1);
            end

        catch
            fprintf('\nCouldn't calculate. Generating new particle\n\n');
            addpop(par,1:np)=genipop(pb,1);
            p=addpop(par,1:np);
            lasterr='a';
        end %try-catch
    end %while
end %for par

##### PSO ---////

fprintf('The best particle so far is:\n-----\n');
bp_string_values=sprintf('[ %g ',Pg(1));
if np>1
    for i=2:np
        bp_string_values=[bp_string_values,sprintf(' , %g',Pg(i))];
    end
end
fprintf([bp_string_values,' ]\n\n\n']);

%Fitness history value
fhv(loop+1)=Pgf;

save allvariables;

end % END OF MAIN LOOP !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

```

---

```

function ipop=genipop(pb,nipop)
%genipop(pb,nipop):
%-----
%
% Generates the initial population.
%
% where: pb    = parameters bounds vector
%         nipop = number of individuals to be created

np=size(pb,1);          %np=number of parameters
ipop=zeros(nipop,np); %initialize ipop
for parameter=1:np
    ipop(:,parameter)=(pb(parameter,2)-pb(parameter,1))...
        *rand(nipop,1)+pb(parameter,1);
end

```

---

```

function fitness=erreval(HexpT,HexpB,p,ef,Xpk,Meq,tspan)
% ERREVAL function to evaluate the error of a hysteresis loop
% from the hysteresis model compared to experimental data.
%
% Syntax: erreval(HexpT,HexpB,p,ef,Xpk,Meq,tspan)
%
% where: HexpT  = experimental input data (TOP part of curve)
%         HexpB  = experimental input data (BOTTOM part of curve)
%         p      = chromosome (set of 'np' parameters)
%         ef     = excitation frequency [Hz]
%         Xpk    = peak displacement amplitude [m]
%         Meq    = 1-DOF equivalent mass of the beam (kg)
%         tspan  = time span of numerical integration [sec]
%
%(c)2006 - Klaus H. Hornig

w=2*pi*ef;
t=0:1e-4:tspan;
t=t';

xi=p(1)+p(2)*Xpk;
wn=p(3)+p(4)*Xpk;

X=Xpk*sin(w*t);
Fcf=Xpk*Meq*((wn^2-w^2)*sin(w*t)+2*xi*wn*w*cos(w*t));

hys1=[X,Fcf];
[modelT,modelB]=getloopdisp(hys1);

YmodT=interp1(modelT(:,1),modelT(:,2),HexpT(:,1),'linear'); %interp. TOP data
%Remove Nan's from YmodT
YmodT1=length(YmodT);
if isnan(YmodT(1))
    YmodT=YmodT(2:YmodT1);
    HexpT=HexpT(2:YmodT1,:);
end
YmodT1=length(YmodT);
if isnan(YmodT(YmodT1))
    YmodT=YmodT(1:YmodT1-1);

```

```

    HexpT=HexpT(1:YmodTl-1,:);
end

YmodB=interp1(modelB(:,1),modelB(:,2),HexpB(:,1),'linear'); %interp. BOTTOM data
%Remove Nan's from YmodB
YmodBl=length(YmodB);
if isnan(YmodB(1))
    YmodB=YmodB(2:YmodBl);
    HexpB=HexpB(2:YmodBl,:);
end
YmodBl=length(YmodB);
if isnan(YmodB(YmodBl))
    YmodB=YmodB(1:YmodBl-1);
    HexpB=HexpB(1:YmodBl-1,:);
end

fitness=leastsqdisp([HexpT;HexpB],[YmodT;YmodB]); %get the OLS error

```

---

```

function value=leastsqdisp(A,B)

```

```

[NRA,NCA]=size(A);
[NRB,NCB]=size(B);

if NRA==NRB
    valuevector=(A(:,2)-B).^2;
    value=sum(valuevector);
else
    disp('Least-squares error: both matrices must have the same dimensions');
end

```

---

```

function [top,bot]=getloopdisp(hys1)

```

```

[NR,NC]=size(hys1);
i=0;
while hys1(NR-i,1)==hys1(NR-i-1,1)
    i=i+1;
end

%
while hys1(NR-i,1)<hys1(NR-i-1,1)
    i=i+1;
end
F1=NR-i; %1st. end
%advance if next is the same
F1f=0;
while hys1(NR-i,1)==hys1(NR-i-1,1)
    i=i+1;
    F1f=F1f+1;
end
%
while hys1(NR-i,1)>hys1(NR-i-1,1)

```

```

        i=i+1;
    end
    F2=NR-i; %2nd. end
    %advance if next is the same
    F2f=0;
    while hys1(NR-i,1)==hys1(NR-i-1,1)
        i=i+1;
        F2f=F2f+1;
    end
    %
    _____
    while hys1(NR-i,1)<hys1(NR-i-1,1)
        i=i+1;
    end
    F3=NR-i; %3rd. end
    %advance if next is the same
    F3f=0;
    while hys1(NR-i,1)==hys1(NR-i-1,1)
        i=i+1;
        F3f=F3f+1;
    end
    %
    _____
    while hys1(NR-i,1)>hys1(NR-i-1,1)
        i=i+1;
    end
    F4=NR-i; %4th. end

    if (hys1(NR,1)<hys1(NR-1,1)) & (hys1(NR,2)<hys1(NR-1,2))
        TOPmin=F2; %=> positive slope loop / end bottom
        TOPmax=F1-F1f;
        BOTmin=F3;
        BOTmax=F2-F2f;
    elseif (hys1(NR,1)>hys1(NR-1,1)) & (hys1(NR,2)<hys1(NR-1,2))
        TOPmin=F4; %=> negative slope loop / end top
        TOPmax=F3-F3f;
        BOTmin=F3;
        BOTmax=F2-F2f;
    elseif (hys1(NR,1)>hys1(NR-1,1)) & (hys1(NR,2)>hys1(NR-1,2))
        TOPmin=F4; %=> positive slope loop / end top
        TOPmax=F3-F3f;
        BOTmin=F3;
        BOTmax=F2-F2f;
    elseif (hys1(NR,1)<hys1(NR-1,1)) & (hys1(NR,2)>hys1(NR-1,2))
        TOPmin=F2; %=> negative slope loop / end bottom
        TOPmax=F1-F1f;
        BOTmin=F3;
        BOTmax=F2-F2f;
    end

    top=hys1(TOPmin:TOPmax,:);
    bot=hys1(BOTmin:BOTmax,:);

```

---

```

function [HexpT,HexpB]=reinterpolate(HexpTo,HexpBo,ints,plotflag,inttype)
%
%Program to reinterpolate the trial hysteresis curves, adding
%a specified discretization of the displacement axis.
%
%HexpTo   : original trial TOP curve
%HexpBo   : original trial BOTTOM curve
%ints     : interpolation size
%plotflag : comparison plot flag (1=yes)
%inttype  : interpolation type
%          (nearest,linear,spline,pchip,cubic,v5cubic)

if nargin==2
    ints=100;
    plotflag=0;
    inttype='linear';
elseif nargin==3
    plotflag=0;
    inttype='linear';
elseif nargin==4
    inttype='linear';
end

%TOP curve
Xmin=min(HexpTo(:,1));
Xmax=max(HexpTo(:,1));
X=linspace(Xmin,Xmax,ints);
X=X';
F=interp1(HexpTo(:,1),HexpTo(:,2),X,inttype,'extrap');
HexpT=[X,F];

%BOTTOM curve
Xmin=min(HexpBo(:,1));
Xmax=max(HexpBo(:,1));
X=linspace(Xmin,Xmax,ints);
X=X';
F=interp1(HexpBo(:,1),HexpBo(:,2),X,inttype,'extrap');
HexpB=[X,F];

if plotflag
    %PLOT comparison
    figure %TOP
    plot(HexpTo(:,1),HexpTo(:,2))
    hold on
    plot(HexpTo(:,1),HexpTo(:,2),'kx','MarkerSize',14)
    plot(HexpT(:,1),HexpT(:,2),'r')
    plot(HexpT(:,1),HexpT(:,2),'+g')

    figure %BOTTOM
    plot(HexpBo(:,1),HexpBo(:,2))
    hold on
    plot(HexpBo(:,1),HexpBo(:,2),'kx','MarkerSize',14)
    plot(HexpB(:,1),HexpB(:,2),'r')
    plot(HexpB(:,1),HexpB(:,2),'+g')
end

```

```

%PrepareExpData.m
%Program to create files with HexpT and HexpB variables (which are the TOP and
%BOTTOM parts of each hysteresis loop) from experimental results

clear all; clc;

%Sample #2
freqs=[370,380,385,390,395,400,402,404,406,408,410,412,414,415,416,417,418,...
      419,420,421,422,423,424,425,426,427,428,429,430,432,434,436,438,440,...
      442,444,450,455,460,470];

j=0;
for i=1:40
    j=j+1;
    file=sprintf('TRAC%g',i); %import TRAC***.txt files (experimental data)
    eval(['load ',file,'.lvm;']);
    eval(['trac=',file,'];]);
    trac(:,1)=trac(:,1)*1e-3; %transform Displacement from MM to METERS

    overagain=0;
    while overagain==0
        Xs=smooth(trac(:,1)); %smooth displacement (moving average - 5 pts)
        F=smooth(trac(:,2));

        loopchoice=0;
        while loopchoice==0
            try
                [HexpBo,HexpTo,TOPmin,TOPmax,BOTmin,BOTmax]=getloopdisp([Xs,F]);
            catch
                fprintf('\nThis is the last loop! What do you want to do?\n');
                overagain=input('OVER AGAIN(0), SELECT THIS ONE(1): ');
                fprintf('\n');
                break
            end

            [HexpT,HexpB]=centercurve1loop(HexpTo,HexpBo);
            figure(1); clf;
            plot(trac(:,1),trac(:,2),'k');
            hold on;
            plot(HexpT(:,1),HexpT(:,2),'b','LineWidth',2);
            plot(HexpB(:,1),HexpB(:,2),'r','LineWidth',2);
            title(sprintf('SAMPLE3_02mm_trac%g_%gHz.mat;',...
                i,freqs(j)), 'Interpreter','none');
            loopchoice=input('Like it? NO(0) / YES(1) / EDIT(2): ');
            if loopchoice==1
                overagain=1;
            elseif loopchoice==2
                break
            end
            Hmin=min([TOPmin,TOPmax,BOTmin,BOTmax]);
            Xs=Xs(1:Hmin); F=F(1:Hmin);
        end %while loopchoice

        if loopchoice==2 %Input points with mouse
            epoa=0; %EditPoints OverAgain
            while epoa==0
                figure(1); clf; plot(trac(:,1),trac(:,2),'k');
                title(sprintf('SAMPLE2_02mm_%gHz.mat;',freqs(j)),...

```

```

        'Interpreter','none');
fprintf('\nPick points with LEFT mouse button\n');
fprintf('When ready, click RIGHT mouse button\n');
fprintf('(Create OVER a whole loop, go CLOCKWISE!)\n');
button=1; k=1; clear xi yi;
while button==1
    [x,y,button]=ginput(1);
    if button==1
        xi(k)=x; yi(k)=y;
        if k==1
            hold on;
            plot(xi,yi,'xm','MarkerSize',8);
        else
            hold on;
            plot(xi(k),yi(k),'xm','MarkerSize',8);
            plot(xi(k-1:k),yi(k-1:k),'r','LineWidth',2);
        end
        k=k+1;
    end
end
[HexpTo,HexpBo, TOPmin, TOPmax, BOTmin, BOTmax]=...
    getloopdisp([xi',yi']);
[HexpT,HexpB]=centercurveloop(HexpTo,HexpBo);
figure(1); clf;
plot(trac(:,1),trac(:,2),'k');
hold on;
plot(HexpT(:,1),HexpT(:,2),'b','LineWidth',2);
plot(HexpB(:,1),HexpB(:,2),'r','LineWidth',2);
title(sprintf('SAMPLE2_02mm_%gHz.mat;',freqs(j)),...
    'Interpreter','none');
epoa=input('Like it? NO(0) / YES(1): ');
if epoa==1
    overagain=1;
end
end
end

end %while overagain

clear HexpTo HexpBo;
eval(sprintf('save SAMPLE2_02mm_%gHz HexpT HexpB;',freqs(j)));
end

```

---

```

function [HexpT,HexpB]=centercurveloop(HexpTo,HexpBo)
%This function centers a hysteresis loop

```

```

Tlength=length(HexpTo);
Blength=length(HexpBo);

x=[HexpTo(:,1);HexpBo(:,1)];
F=[HexpTo(:,2);HexpBo(:,2)];

xmax=max(x);
xmin=min(x);
Fmax=max(F);

```

```
Fmin=min(F);

xpk=(xmax-xmin)/2;
Fpk=(Fmax-Fmin)/2;
dx=xmax-xpk;
dF=Fmax-Fpk;

Hc=[HexpTo(:,1)-dx , HexpTo(:,2)-dF;...
    HexpBo(:,1)-dx , HexpBo(:,2)-dF];

HexpT=Hc(1:Tlength,:);
HexpB=Hc(Tlength+1:Tlength+Blength,:);
```

---

## **APPENDIX B**

**PROGRAM CREATED IN NATIONAL INSTRUMENTS LABVIEW 8.0 FOR  
THE ACQUISITION OF EXPERIMENTAL HYSTERESIS LOOPS**

**FRONT PANEL:**

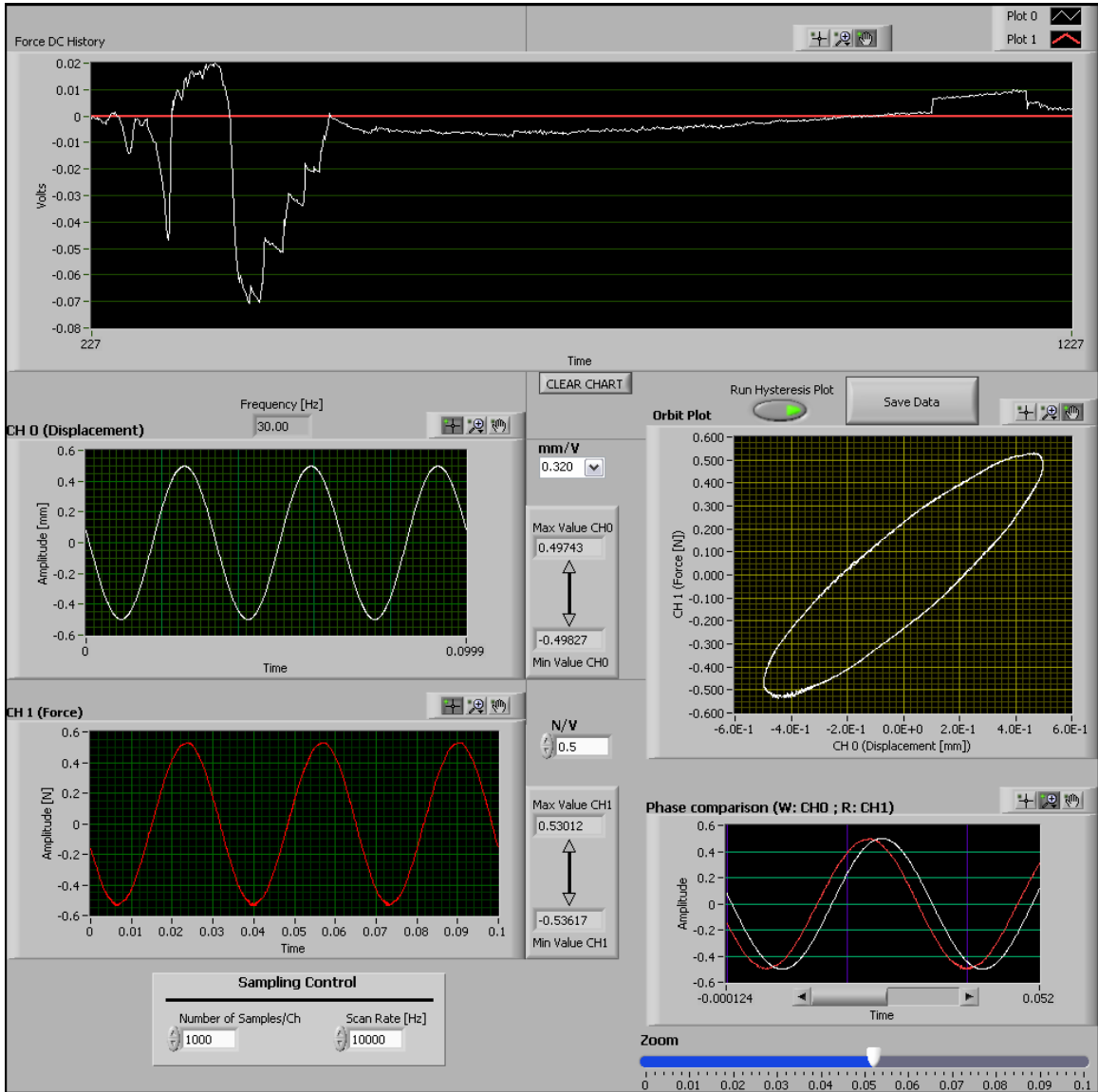


DIAGRAM:

