The University Of Sheffield.

Department
Of
Mechanical
Engineering

MSc Advanced Mechanical Engineering

Nonlinear Systems Superharmonics Cancellation

Hakan **DOGAN**

August 2017

Dr. Nikolaos Dervilis

Dissertation submitted to the University of Sheffield in partial fulfilment of the requirements for the degree of Master of Science

ABSTRACT

Almost all engineering systems have nonlinear characteristics to some extent. However, they are generally approached as linear systems by making assumptions when they are being analysed or designed. This is because of the easiness of dealing with linear systems. Still, the nonlinearity in systems becomes dominant when the systems are getting lighter and more flexible. In this case, the linear theory cannot be applied to understand the systems since the linear properties, which make possible the linear theory, start to break down and some new nonlinear phenomena might occur. This paper offers a novel approach to remove superharmonics, which is one of the nonlinear phenomena, in a nonlinear system. The motivation of the study can be stated as expressing a nonlinear response which has superharmonics with a new superharmonic-free from.

The novelty in the study is to seek a transformation which transforms a nonlinear response with superharmonics into a new form without superharmonics. The transformation is in a polynomial expansion form and the coefficients of the transformation are defined by using Self-Adaptive Differential Evolution so as to cancel superharmonics. A single degree-of-freedom Duffing's equation is used to create a nonlinear response with superharmonics. The superharmonics which exist in the response are tried to remove by using the transformation and Self-Adaptive Differential Evolution. Also, the Monte Carlo simulation is conducted to see the possible scenarios and all results are presented.

Next, the Gaussian process is employed to obtain the initial signal from the transformed signal as a pattern recognition application. The general pattern of the initial signal is obtained in this way.

The results show that this approach proposed in this paper can be utilised to remove superharmonics in a nonlinear system. Thus, this novel method provides an easiness of analysis of the nonlinear system, especially for modal analyses and system identification problems.

NOMENCLATURE

Greek Symbols

Φ	Phase angle (radian)
ω_n	Natural frequency (rad/s)
ε	Small parameter
$\boldsymbol{\Phi}$	Eigenvectors of A
ε_{noise}	Noise term
σ_0	Signal standard deviation
λ	Length scale
ξ	Damping ratio

Other Symbols

m	Mass (kg)
c	Viscous damping (Ns/m)
k	Linear stiffness (N/m)
$f_n(\dot{x}, x)$	Velocity-displacement-dependent nonlinear term
w	Forcing frequency (rad/s)
$x(t)$	Time-dependent displacement (m)
$F(t)$	Time-dependent force (N)
k'	Cubic stiffness (N/m^3)
X_l	Amplitude of the displacement (m)
x_0	Linear response
M	Mass matrix (kg)
C	Damping matrix (Ns/m)
K	Stiffness matrix (N/m)
F_f	Amplitude matrix of force (N)

$\boldsymbol{r}$	Time-dependent forcing vector
A	System matrix
$\tilde{f}_n(\boldsymbol{x})$	Nonlinear term
$\boldsymbol{X}_{i,G}$	Parameter vector
$\boldsymbol{V}_{i,G}$	Mutant vector
$\boldsymbol{X}_{r,G}$	Randomly chosen parameter vector
$\boldsymbol{X}_{best,G}$	Best vector to fit the cost function
F	Scaling Factor
CR	Crossover constant
y	Observed noisy output
N	Gaussian distribution
σ^2	Variance of the noise
$f(\boldsymbol{x})$	Prior
$K_{\boldsymbol{x},\boldsymbol{x}}$	Covariance function
$\boldsymbol{x}_*$	Future input
$\boldsymbol{f}_*$	Output which is desired to find
$p(\boldsymbol{y}\|\boldsymbol{x})$	Likelihood
$\boldsymbol{f}_*\|\boldsymbol{y}, x, x_*$	Posterior distribution
k_2	Quadratic stiffness(N/m^2)
k_3	Cubic stiffness (N/m^3)
a	Nonlinearity parameter
z	Transformed signal
a_n	Coefficients of the transformation

CONTENTS

ACKNOWLEDGEMENT

I would like to thank my supervisor Dr Nikolaos Dervilis for his advices and support throughout the study. I have been lucky to have a supervisor who does not hesitate to share his knowledge and experience with me. His positive attitude and encouragement gave me confidence during the project.

1 INTRODUCTION

Engineering applications are essential parts of the modern world. They have been improved continuously and consequently, they differ from the earliest forms with the help of the knowledge which has been obtained. Therefore, it is crucial to understand theories behind the systems to create safer, more efficient and low-cost applications. One of the significant subjects for mechanical systems which work under dynamical loads is vibration. In the vibration theory, the linear approach is well established and it is sufficient to explain the behaviour of the most systems by making some assumptions such as small deformation and ignoring friction. However, as the systems become lighter and more flexible, the linear theory starts to break down and the real outputs of the systems differentiate from the analytical or numerical results. Some new phenomena which are not expected in the linear systems occur such as superharmonics, subharmonics, jumps, bifurcation, saturation[1]. They make the analysis of the nonlinear system more complicated and difficult.

The focus of the paper is mechanical systems rather than chemical or electrical systems. The study is concerned with the cancellation of the superharmonics in a nonlinear system. Superharmonics can be defined as higher harmonics of the forcing frequency which do not appear in a linear system. The elimination of the superharmonics in a system provides easiness for especially areas of nonlinear system identification and nonlinear modal analysis. Moreover, it can be used in the modal analysis as a part of the modal analysis of a nonlinear system. Consequently, it gives a quick picture of the system which is easier to understand the overall behaviour of the system. The importance of the study can be defined that it offers a simple approach, which does not contain any complex analytical analysis, to simplify the analyse of a nonlinear system which contains superharmonics. The motivation of the study is to find an expression of the nonlinear response in a new form by transforming it in a similar way as in normal form transformation which transforms the nonlinear equations into a simple form.

One approach to cancel superharmonics in a nonlinear system can be self-adaptive differential evolution (SADE) algorithm which is an optimisation method. The cancellation process functions as a simplified transformation. The output signal of a nonlinear system which contains superharmonics can be transformed and cleanse from the superharmonics by using SADE. This method is not required complex analysis so it is relatively straightforward to apply.

Only one result may not be enough to decide the success of SADE in terms of superharmonics cancellation so the Monte Carlo simulation must be run. For this,

SADE parameters and settings are adjusted until obtaining effectual results. After that, the algorithm is run as many as possible by changing some parameter in the equation of motion to see the possible outputs. In that way, it is possible to have a sufficient data set to evaluate the algorithm in terms of cancellation of superharmonics and the robustness of the algorithm becomes tested. In general, the Monte Carlo simulation provides to see the best and worst scenarios by randomly changing the parameters. This simulation is also presented in this paper.

Another study which is given in this paper is a pattern recognition application. After obtained the transformed signal in which superharmonics are not included, acquisition of the original signal from the transformed signal might be desired. Hence, a pattern recognition algorithm can be applied to obtain the original signal. The Gaussian process can be a reasonable choice to succeed this objective. It is processed with matrix manipulations and gives the results with a confidence region. Therefore, any validation is not required as a difference from the other pattern recognition methods such as Neural networks.

As briefly explained above, this paper mainly investigates a transformation to cancel superharmonics in a nonlinear system by using SADE. Also, it seeks an application of a reverse process to obtain the original signal from the transformed signal by using Gaussian process. The motivation of this study is to reduce the degree of the difficulty of the analysis of a nonlinear system which stems from superharmonics with a simple transformation. It investigates an expression of the nonlinear response in a new form by transforming it in a similar way as in normal form transformation which transforms the nonlinear equations into a simple form. The study is based on [2] and it seeks a cancellation which is mentioned above. The objectives of the study can be stated as:

- Remove or reduce the amplitudes of the second, third and more superharmonics as many as possible,
- An inverse process which gives the original signal from the transformed signal.

Thus, the aim of the study is to provide a novel approach which can be used in the modal analysis and nonlinear system identification in the analysis of nonlinear systems. The novelty is to transform the response of the system into a new form without superharmonics by using a transformation, whose parameter can be found by SADE.

This current study is built upon the findings of previous studies as all studies are. Therefore, the next chapter will give the literature review about the nonlinear analysis, solution methods of the nonlinear equations, self-adaptive differential evolution algorithm and the Gaussian process. The basic concepts underlying the

study are provided, besides the previous works. Chapter 3 will present the methodology and explain some critical points and adjustments in the processes. It presents how the decision of the transformation and the cost function were made. Moreover, it gives some details about the Monte Carlo simulation and the Gaussian process. Chapter 4 will demonstrate the results of the study. These results, their robustness and the study as an overall will be discussed in Chapter 5. In addition, the future works related to this study will be evaluated in the same chapter. Finally, Chapter 6 will draw a conclusion of the study.

2 LITERATURE REVIEW

This chapter introduces the previous works and the basic concepts underlying the study in this paper. The study presented in this paper is an application of SADE to a nonlinear system's response and a pattern recognition application. Thus, the study can be seen as a combination of three subjects: nonlinear vibration, SADE and the Gaussian process. Though numerous studies can be found in each individual subject, a study by Dervilis et al. [2] is directly related to this study in terms of the application of SADE to a nonlinear system's response. Also, a statistical analysis of [2] was conducted in [3]. Each subject is reviewed individually in regard to the roles in the study and therefore, the chapter consists of three sections. In the first section, the general properties of the linear and nonlinear systems are given. Because of the analysis of the nonlinear system is required different methods from the linear theory, the analysis methods are presented at the end of the first section. Understanding the solution methods gives an idea about how complicated the nonlinear analyses are. The second section mentions about SADE starting from differential evolution algorithm. Finally, the third section is about the Gaussian process.

2.1 General Properties of the Systems

2.1.1 Linear Systems

Vibration analysis can be used for different purposes such as the decision of critical parameters in a design project, condition monitoring, and system identification problem to estimate the parameters. Modal analysis is frequently used to define critical parameters of a system such as the natural frequencies, mode shapes, damping ratios and frequency response function (FRF) [1], [4], [5]. The analysis of a linear system is easier than the analysis of a nonlinear system. Furthermore, the linear vibration theory has been studied for many and thus, it is a settled theory [1]. Therefore, although it is difficult to find a purely linear system in nature, engineers tend to use linear system theory by making some assumptions.

Linear systems have some properties such as superposition, homogeneity and reciprocity which make the linear analysis easier. If superposition can be said to exist in a system, the output of that system for the sum of the different inputs can also be expressed the sum of the outputs of the same system for each input. Homogeneity shows that the ratio of the output to the input in the frequency domain, which gives the frequency response function, is independent of the level of excitation. Finally, reciprocity can be defined as that the ratio of the output of a system at a point to the input of the system at another point is the same as the case where the points are exchanged. These all three properties can be also used to detect nonlinearity in

a system [6]. The detection of the nonlinearity in a system is crucial because the nonlinearity causes the differences in the results between the real output and the output which is obtained from the linear theory.

2.1.2 Nonlinear Systems

Nonlinearity can cause deviation in the output of a system so it must be taken into consideration in the cases of that nonlinear terms are dominant in a system. It is important to be able to detect nonlinearity. Otherwise, the mathematics applied will be invalid owing to the nonlinear behaviour of the equations of motion of the corresponding system. The detection of the nonlinearity can be made by investigating harmonic distortion in results (displacement, velocity or acceleration) of the system, distortion in FRF, the Hilbert transform and Nyquist plots due to the loss of homogeneity [1], [4], [6]. Different types of nonlinearities such as cubic stiffness, nonlinear damping and Coulomb friction might show different distortions from the linear system. A book by Worden and Tomlinson [6] comprehensively introduces the detection, identification and modelling methods for the nonlinear structures. Figure 2.1 is a good example for both the detection of nonlinearity and the dependency on the level of excitation [7]. It illustrates the first frequency in an FRF of a forced nonlinear system due to cubic stiffness and nonlinearity causes distortion in FRF. Moreover, the increase in the amplitude of the force increases the distortion from the linear one.

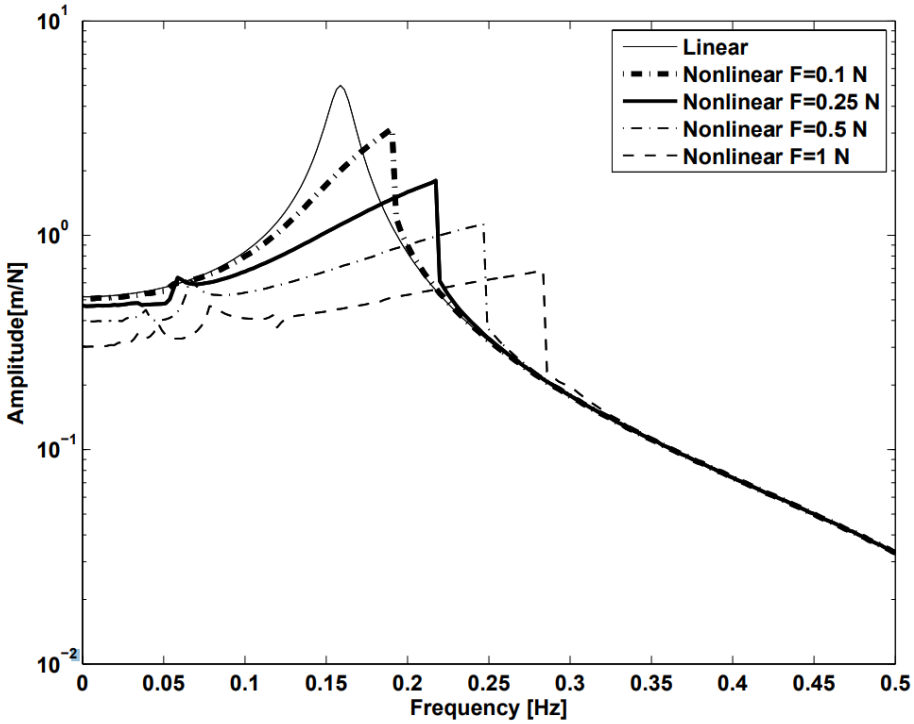

Figure 2.1 An example of FRF showing effects of nonlinearity [7].

In a real structural system, different factors can lead to nonlinearity. These might result from geometric nonlinearity due to large deformation, nonlinear material behaviour, boundary conditions such as loose joints and clearance, polynomial stiffness and damping, and friction [1], [6], [8]. Nonlinearity comes with its own difficulties which stem from the nature of the equations of motion when looked from a mathematical perspective. It can be possible to find different approaches to solve nonlinear equations and estimate the behaviour of the nonlinear structural system.

The equation of motion for a single-degree of freedom mass-damper-spring model with nonlinear terms can be expressed as

$$m\ddot{x}(t) + c\dot{x}(t) + kx(t) + f_n(\dot{x}, x) = F(t)$$

Equation 2.1

where m, c and k are mass, damping and stiffness, and $f_n(\dot{x}, x)$ is the term which causes nonlinearity. If $f_n(\dot{x}, x)$ is zero, the system becomes linear. $F(t)$ is a time-dependent force and $x(t)$, $\dot{x}(t)$ and $\ddot{x}(t)$ in the equation are time-dependent displacement, velocity and acceleration, respectively. The solution of the nonlinear differential equations (Equation 2.1), which are used to describe the mathematical model of the nonlinear system, is difficult when compared linear differential equations. Some methods have been developed which are taken the nonlinearity of the system into consideration to estimate the solution or some critical parameters such as natural frequencies.

2.1.3 Solution Methods for Nonlinear Systems

2.1.3.1 Harmonic Balance Method

Harmonic balance method is one of the solution methods for nonlinear problems. It is possible to find the proof of that the method can be used for the steady-state solution of the nonlinear systems [8]–[13]. This method also has some advantages and disadvantages as all nonlinear solution methods have. The major benefit is that Harmonic balance method can be used for not only weakly nonlinear systems but also highly nonlinear systems [8], [10], [13]. This gives a general applicability to the method. Most of the methods are restricted to weakly nonlinear systems. Harmonic balance method gives a fast approximated view about the behaviour of the system. However, it might be concluded with an inconsistent result because of the higher harmonics which are not balanced [8]. The method is simple to apply and it has general applicability whether the nonlinearity is weak or not. Hence, Harmonic balance method is a useful approach to obtain the steady-state solution.

To show the implementation of Harmonic balance method, Equation 2.1 can be modified as a nonlinear equation with a cubic stiffness term by taking $f_n(\dot{x}, x) = k'x^3(t)$ and $F(t) = F_f sin(wt - \Phi)$. Therefore,

$$m\ddot{x}(t) + c\dot{x}(t) + kx(t) + k'x^3(t) \ = F_f sin(wt - \Phi)$$

Equation 2.2

where w is the forcing frequency, Φ is the phase difference and k' is nonlinear stiffness. A trial solution can be assumed to have the form

$$x(t) = \sum X_l \sin(lwt) \quad l = 1,2, \dots$$

Equation 2.3

If Equation 2.3 for $l = 1$ is substituted into Equation 2.2, the new equation becomes

$$-mw^2 X_1 \sin(wt) + cwX_1 \cos(wt) + kX_1 \sin(wt) + k'{X_1}^3 sin^3(\text{wt}) \ = F_f sin(wt - \Phi)$$

Equation 2.4

After trigonometric transformations, the equation yields

$$-mw^2 X_1 \sin(wt) + cwX_1 \cos(wt) + kX_1 \sin(wt) + k'{X_1}^3 \left[\frac{3}{4}\sin(wt) - \frac{1}{4}\sin(3wt)\right] = F_f sin(wt)\cos(\Phi) - F_f cos(wt)\sin(\Phi)$$

Equation 2.5

As it can be seen in Equation 2.5, there is a harmonic term $\sin(3wt)$, which is not balanced. To obtain an exact solution, what it has to be done is to update the initial solution assumption and increase the value of l in Equation 2.3. Yet, the higher l is taken, the higher harmonics will appear in Equation 2.5. Thus, some higher harmonics are neglected and the balance is provided. The approximated gain of the system, phase and FRF can be found in that way.

2.1.3.2 Averaging and Perturbation Methods

Averaging and perturbation method are two techniques which can be applied to weakly nonlinear systems to obtain the response of the system [8]. Averaging and perturbation method might give an erroneous result if the nonlinearity is high. This is basically because of the small nonlinearity assumption in the methods. However, it can be seen in the literature that perturbation method can provide a good approximation for highly nonlinear problems in some cases [14], [15]. Both methods have different approaches to approximate the response.

In averaging method, the equation of motion of the system can be expressed as [8]

$$\ddot{x} + {\omega_n}^2 x = \varepsilon F_n(x, \dot{x})$$

Equation 2.6

where ω_n is natural frequency, $F_n(x, \dot{x})$ includes nonlinear terms and ε is a small parameter which means that the right-hand side of the equation is small. Thus, the nonlinear term, force and damping must be small. The method offers a trial solution for Equation 2.6 in the form

$$x = x_c(t)\cos(\omega_n t) + x_s(t)\sin(\omega_n t)$$

Equation 2.7

where $x_c(t)$ and $x_s(t)$ are calculated by being averaged over a cycle [8]. If Equation 2.7 is substituted into Equation 2.8, the transient response of the system can be obtained.

In perturbation method, it is considered that the solution can be expressed as a power series in ε

$$x = x_0 + \varepsilon x_1 + \varepsilon^2 x_2 + \cdots$$

Equation 2.8

In the form of Equation 2.8, the response can be considered as the sum of the linear response (x_0) and perturbations around the linear response [8]. Although there is a method called regular perturbation theory, it will not be mentioned here. This is because it does not work well for a large time. Multiple scales method approximates well by using an approach which includes different time-scales (fast and slow time-scales). In light of this information, the solution can be assumed as [8]

$$x = X_c(\varepsilon t)\cos(\omega t) + X_s(\varepsilon t)\sin(\omega t)$$

Equation 2.9

As it can be seen in Equation 2.9, ε leads to a slow evolution of the amplitude. In that way, an accurate response of the system can be provided.

Both averaging and multiple scales techniques have been explained briefly as the focus of this paper is the cancellation of the superharmonics rather than solution techniques of nonlinear problems. A book by Wagg and Neild [8] covers a comprehensive explanation of both techniques.

2.1.3.3 Normal Form Transformation

Another approximation technique to estimate the response of a nonlinear system is normal form transformation. It can be said that this method has a higher importance than other methods in terms of this study since this study is based on the idea

underlying normal form transformation. Normal form transformation provides a simpler form by transforming the nonlinear system's equations of motion [8], [16]–[18]. A journal article by Neild et al. [19] and a book by Wagg and Neild [8] present an introduction and the use of the normal form transformation. The initial approach for normal form transformations includes the conversion of the second-order form (Equation 2.1) to first-order state-space form. Basically, the method of normal form uses a transformation of the form of the equations which cannot be decoupled because of the nonlinear terms. First, linear decomposition is applied to the equations but it is not possible to decouple all terms due to nonlinearity. Hence, a second nonlinear decomposition is applied. Eventually, the method provides an approximate response of the nonlinear system.

If the implementation of the method can be showed, the equation of motion of a second-order multi-degree of freedom nonlinear system can be expressed as

$$M\ddot{\boldsymbol{x}} + C\dot{\boldsymbol{x}} + K\boldsymbol{x} + f_n(\boldsymbol{x}, \dot{\boldsymbol{x}}) = F_f \boldsymbol{r}$$

Equation 2.10

where M, C and K are mass, damping and stiffness matrix, F_f is the amplitude matrix of force, $\boldsymbol{r}$ is time-dependent forcing vector and $f_n(\boldsymbol{x}, \dot{\boldsymbol{x}})$ includes nonlinear terms in the corresponding system. For free vibration ($F_f = 0$), first-order state-space form can be written [8]

$$\dot{\boldsymbol{x}} = A\boldsymbol{x} + \tilde{f}_n(\boldsymbol{x})$$

Equation 2.11

where $\boldsymbol{x}$ is the state vector, $A = \begin{bmatrix} 0 & I \\ -M^{-1}K & -M^{-1}C \end{bmatrix}$ and $\tilde{f}_n(\boldsymbol{x}) = \begin{Bmatrix} 0 \\ -M^{-1}f_n(\boldsymbol{x}) \end{Bmatrix}$. The first transformation, linear modal transformation, is applied at this point by substituting $\boldsymbol{x} = \boldsymbol{\Phi q}$ into Equation 2.11 and multiplying by $\boldsymbol{\Phi}^{-1}$, where $\boldsymbol{\Phi}$ is eigenvectors of A. Thus, Equation 2.11 is transformed into [8]

$$\dot{\boldsymbol{q}} = \boldsymbol{\Lambda q} + \boldsymbol{f}(\boldsymbol{q})$$

Equation 2.12

where $\boldsymbol{f}(\boldsymbol{q})$ contains the nonlinear terms. It can be said that Equation 2.12 consists of two parts as linear and nonlinear part. If the system of interest is linear, $\boldsymbol{f}(\boldsymbol{q})$ will be zero. One more transformation, nonlinear transformation, is required for nonlinear systems to try to convert Equation 2.12 into $\dot{\boldsymbol{u}} = \boldsymbol{\Lambda u}$ which is linear form. However, this is not possible in most cases. Hence, instead of directly seeking a transform in the form of $\dot{\boldsymbol{u}} = \boldsymbol{\Lambda u}$, the form as [8]

$$\dot{\boldsymbol{u}} = \boldsymbol{\Lambda u} + \boldsymbol{g}(\boldsymbol{u})$$

Equation 2.13

is looked for by using nonlinear transformation as [8]

$$\boldsymbol{q} = \boldsymbol{u} + \boldsymbol{h}(\boldsymbol{u})$$

Equation 2.14

Note that the terms which stand for nonlinearity in Equation 2.12, 2.13 and 2.14, $\boldsymbol{f}(\boldsymbol{q})$, $\boldsymbol{g}(\boldsymbol{u})$, and $\boldsymbol{h}(\boldsymbol{u})$ can be expressed in a series of function similar to Equation 2.8. Even though the analysis was made for free vibration response, the forced vibration analysis is also achievable. The basic idea of normal form analysis in the first-order form has been given above. Neild and Wagg [16] demonstrated that a second-order approach of a normal form analysis is also possible for vibration problems.

The idea in normal forms transformation is to create a simplified form from a second-order equation. The form is simplified as much as possible in an analytical way. Similarly, this study investigates a transformation which delivers a simplified version of the response without superharmonics.

2.1.3.4 Nonlinear Normal Modes

Nonlinear systems have a different characteristic from linear systems which makes the analysis difficult. For instance, FRF is dependent on the level of excitation in a nonlinear system. Similarly, the mode shapes in a nonlinear system vary with the change of the energy or amplitude and nonlinear normal modes relies on this change [1], [20], [21]. A common definition for nonlinear normal modes is the nonlinear extension of the normal mode in classical vibration theory [19], [21], [22]. The first development of the method is the article by Rosenberg [23] and many studies have been conducted to analyse of nonlinear systems [10], [14], [18], [20], [22], [24]–[28]. An article by Vakakis [21] presents an overview of the method and its application. Most of the literature about the nonlinear normal modes is related to low-order systems so it has not tested with complex engineering structures and highly nonlinear systems. Therefore, Kerschen et al. [22] encourage the implementation of the method numerically.

It is important to highlight the importance of modal analysis in vibration problem. Nonlinear normal modes makes possible modal analysis and also, it is beneficial for system identification problems. Though nonlinear normal modes is crucial, this paper will not go further in the method as the scope of the paper is cancellation of the superharmonics.

2.2 Self-Adaptive Differential Evolution

The idea of this study to eliminate superharmonics is to transform the result which contains superharmonics into a simple form which does not have superharmonics by using SADE. The transformation can be in the form of a polynomial expansion and the problem turns an optimisation problem in terms of finding the parameters. Therefore, differential evolution (DE) algorithm can be used to remove the higher harmonics components from the result. Differential evolution, proposed by Storn and Price, is an optimisation method with advantages such as easiness to use and the capability to deal with non-differentiable, nonlinear and multimodal function [29]. The major benefits of the algorithm are its rapidness and simplicity [30], [31]. Complex analyses are not required to apply. The optimisation problem of the interest turns into a simple form and it can keep being a powerful method, despite the simplicity. Moreover, the space complexity of the algorithm and the number of control parameters are low [31]. These increase the capability of dealing with demanding optimisation problem and support the easiness of the algorithm. Owing to all these advantages of DE, it has been used as an optimisation method in many engineering applications in the literature. Worden and Manson [32] used DE to improve the parameter estimates in a hysteretic system. Dervilis et al. [2] found the parameters of the polynomial expansion to remove the first superharmonic. Rogalsky et al. [33] employed the algorithm as an aerodynamic design optimizer. It has been utilised in not only mechanical engineering applications but also a wide range of areas such as chemical engineering, pattern recognition, bioinformatics and electrical power systems [31], [34], [35]. DE has demonstrated that it can be served as a powerful optimisation method in many disciplines where optimisation is required.

The concept of DE is given in numerous articles [29], [30], [36]–[39]. More detailed information and the progress of DE are given in two surveys [31], [40].

DE is a population-based optimisation method. Thus, the algorithm creates new populations from the initial population until the stopping criterion is satisfied. Figure 2.2 illustrates the schematic of the algorithm and processes to create a new population (called new generation) from the current population. The whole process consists of three operations: mutation, crossover and selection. First, a population is created randomly and in the manner that it covers all parameter space by using initial parameters. The parameter vectors are chosen

$$\boldsymbol{X}_{i,G} = \left[x_{1,G}, x_{2,G}, \dots, x_{n,G}\right], i = 1,2, \dots, n$$

Equation 2.15

where index G shows generation number and n is the number of parameter vectors.

Next step is mutation where the mutant vector is created by using the parameter vectors. Though there are different rules to generate a mutant vector, only one known as DE/rand/1 is given below to make the description of the operation simple. Other rules will be touched on later in this and next chapter. The mutant vector can be expressed as

$$\boldsymbol{V}_{i,G+1} = \boldsymbol{X}_{r_1^i,G} + F.\left(\boldsymbol{X}_{r_2^i,G} - \boldsymbol{X}_{r_3^i,G}\right), i = 1,2,\dots,n$$

Equation 2.16

where $r1, r2, \dots$ are indexes chosen randomly and F is the scaling factor which is a constant between 0 and 2.

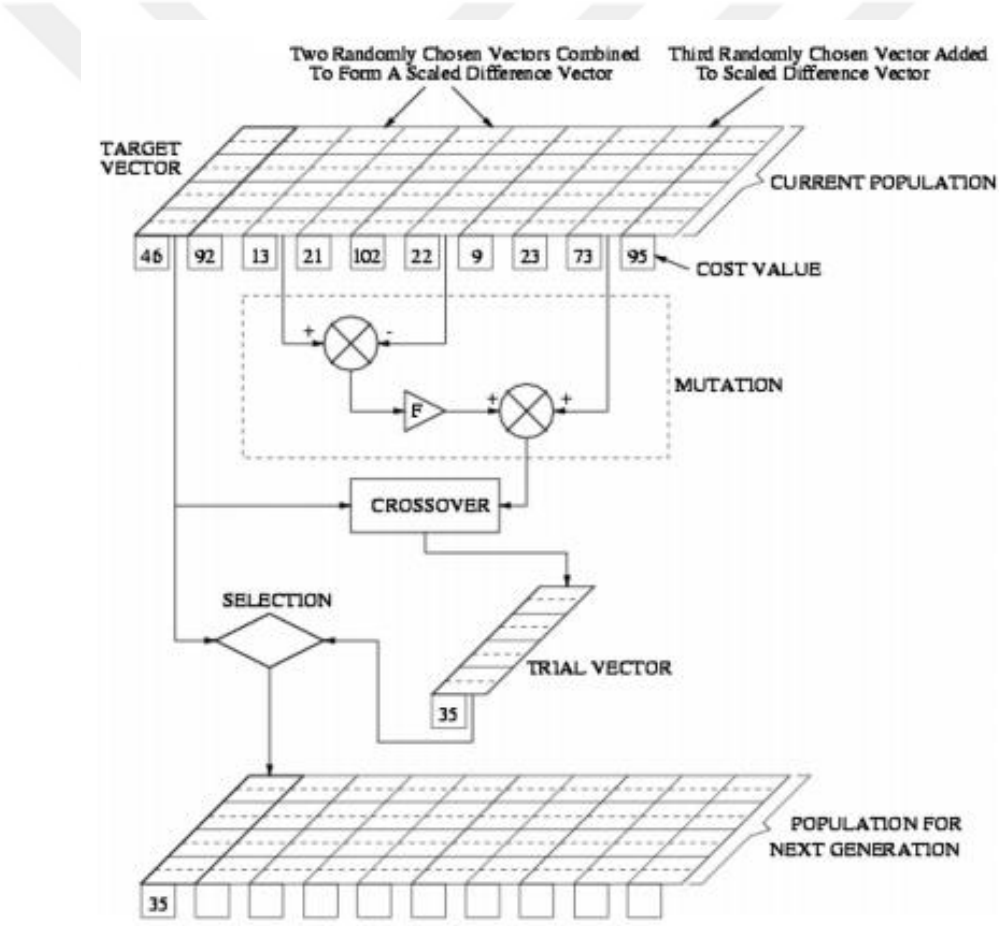

Figure 2.2 The schematic of DE and its processes [32]

In crossover operation, a trial vector is generated from the mutant vector and target vector which is chosen from the initial population. For this, a random value $\in [0, 1]$, which is generated for each parameter, is compared with the crossover constant (CR) $\in [0, 1]$. The comparison is made for each parameter between the mutant and

target vectors. If crossover constant is smaller than the random value, the parameter of the target vector is kept for the trial vector. Otherwise, the parameter of the mutant target vector is kept. It is assured that at least one parameter from the mutual vector is chosen for the trial vector to sustain the diversity [29]. At the end of the crossover process, a trial vector is obtained for the selection operation.

Selection is the last step of the algorithm where the members of the new generation are chosen. The one which has the smallest cost function value is transferred to the new generation population. These processes are repeated until the stopping criterion is satisfied. It can be seen that all procedure is straightforward. There is no need challenging analysis. It must be noted that the choice of the mutation rule (or strategy), scaling factor, crossover constant and size of the initial population are decided by the user in the standard differential evolution algorithm. These parameters are called control parameters in literature.

The mutant vector can be generated by different mutation strategies. Each strategy bears different properties. For instance, the strategy in Equation 2.16 has a high capacity of detection but low converge speed. Another one is named DE/best/1 is formulated as

$$\boldsymbol{V}_{i,G+1} = \boldsymbol{X}_{best,G} + F.\left(\boldsymbol{X}_{r_1^i,G} - \boldsymbol{X}_{r_2^i,G}\right), i = 1,2,\dots,n$$

Equation 2.17

where $\boldsymbol{X}_{best,G}$ is the best vector to fit the cost function of the problem in the corresponding population. Even though its converge speed is fast, it is inclined to be trapped at a local optimum [37]. Hence, building an appropriate strategy pool is vital to obtain the best result from the algorithm.

Even though DE can successfully deal with a numerical optimisation problem, the success of the standard differential evolution algorithm is highly dependent on the strategies and control parameters settings [36], [37]. Therefore, different parameter studies for DE have been conducted to use the algorithm more efficiently [41]–[44]. It is possible to find a good mutation strategy and parameter settings. However, this can be time-consuming and it might not end up with a result which belongs to a global minimum. Thus, the decision of mutation strategy and control parameters can be seen as the main drawback of DE.

Self-adaptive differential evolution algorithm, which can adjust the control parameters and the strategy during evolution, is first proposed by Qin and Suganthan [36]. SADE reduces the effects of users' decisions on the output of the process. It adapts the strategy to generate the mutant vector among mutation strategies defined, as well as adapting scaling factor and crossover constant. It

adapts the strategy and parameters according to the previous results observed in a learning period. The size of the population is defined by users as it is in standard DE. This is because it is related to the structure of the problem of interest. Together with this, the crossover constant has a higher effect on the output and the scaling factor affects the speed of converging [37].

A detailed description of SADE also contains the analysis of strategies of trial vector generation, is given in [37]. It is obvious that the mutation strategy and two control parameters, the scaling factor and crossover constant, are adapted in the algorithm. To update the strategy, first, a strategy pool where all strategies are included is created. The choice of one of the strategies happens according to the probabilities which are defined depending on the success of that strategy in the previous learning period. Initially, all choosing probabilities are taken equal to each other and they are updated as regards success of the strategies.

In parameter adaptation, the scaling factor and crossover constant are assigned randomly by a normal distribution with a mean and standard deviation. The mean value of crossover constant is updated using a set of crossover constants of the vectors which are transferred the next generation in the previous learning period. The updated mean value is used in the current learning period [37]. In that way, SADE decides the possible optimum control parameters during evolution.

SADE algorithm offers a strong and fast tool to solve a numerical optimisation problem with a minimum user effect which makes the algorithm more effective. It can be said that, with these all properties, SADE is ideally suited to this study.

2.3 Gaussian Process

One of the objectives of this study is to obtain the original signal which contains superharmonics from the transformed signal. This problem can be approached as a pattern recognition application. Hence, the Gaussian process for regression can be practised to reach the original signal. When considered neural networks applications, the Gaussian process does not have problems like overfitting and the decisions of architecture, activation functions and learning rate [45]. Therefore, a validation is not required in the Gaussian process. The Gaussian process relies on the prior distribution over functions rather than the prior distribution over parameters[46]. Hence, in order to practice Bayesian approach, a set of functions are taken as a prior and the likelihood is built by using data observed [46]. The posterior probability can be expressed

$$Posterior \ \alpha \ prior \ x \ likelihood$$

Equation 2.17

The estimation for unknown inputs is defined with that posterior probability. It can be found analytically in the Gaussian process with matrix manipulations.

Although the key points of the Gaussian process for regression are given below, one of the main references for the Gaussian process can be found in [45] for those who go in further detail. Furthermore, different studies for the Gaussian process for regression exist in the literature [46]–[49]. In Gaussian process, observed noisy output can be expressed as

$$y = f(\boldsymbol{x}) + \varepsilon_{noise}$$

Equation 2.18

where $\boldsymbol{x}$ is input vector, f is the function value and ε_{noise} is the noise which has a Gaussian distribution $N{\sim}(0,\ \sigma^2)$ where σ^2 is variance. The Gaussian process provides a distribution over the function. Thus, prior $f(\boldsymbol{x})$ can be expressed with the mean and variance in Gaussian process

$$f(\boldsymbol{x}){\sim}GP(m_{\boldsymbol{f}}, K_{\boldsymbol{f},\boldsymbol{f}})$$

Equation 2.19

where $m_{\boldsymbol{f}}$ and $K_{\boldsymbol{f},\boldsymbol{f}}$ are mean and covariance function, respectively. The mean function can be taken zero for the simplicity. The covariance function can be chosen as squared exponential so that it becomes an expression which shows distances between the random variables. Therefore, the covariance function

$$K_{\boldsymbol{x},\boldsymbol{x}} = k(\boldsymbol{x}, \boldsymbol{x}') = {\sigma_0}^2 \exp(-\frac{1}{2\lambda^2}|\boldsymbol{x} - \boldsymbol{x}'|^2)$$

Equation 2.20

where $\boldsymbol{x}$ and $\boldsymbol{x}'$ are different points, and σ_0 and λ are hyperparameters called amplitude (signal standard deviation) and length scale, respectively. The covariance represents how related two points are: the value of the covariance which closes to 1 shows a high relation between corresponding points and the value of the covariance which closes to 0 means low relation. If the noise is taken into account in the observations, then the covariance function becomes

$$cov(\boldsymbol{y}) = K_{\boldsymbol{x},\boldsymbol{x}} + \sigma^2 I$$

Equation 2.21

where I is the identity matrix. A training set can be specified $D_1(\boldsymbol{x}, \boldsymbol{y})$ where $\boldsymbol{x}$ and $\boldsymbol{y}$ are input and output data observed, and a test set can be described $D_2(\boldsymbol{x}_*, \boldsymbol{f}_*)$ where $\boldsymbol{x}_*$ is future input and $\boldsymbol{f}_*$ is the value which is desired to find. The joint prior distribution becomes

$$\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{f}_* \end{bmatrix} \sim N\left(0, \begin{bmatrix} K_{x,x} + \sigma^2 I & K_{x,x_*} \\ K_{x_*,x} & K_{x_*,x_*} \end{bmatrix}\right)$$

Equation 2.22

The likelihood is written

$$p(\boldsymbol{y}|\boldsymbol{x}) \sim N(\boldsymbol{x},\ \sigma^2 I)$$

Equation 2.23

Finally, the posterior (predictive) distribution from Bayes' theorem, which provides the Gaussian distribution of $\boldsymbol{f}_*$, is found

$$\boldsymbol{f}_* | \boldsymbol{y}, x, x_* \sim N(\bar{\boldsymbol{f}}_*, cov(\boldsymbol{f}_*))$$

Equation 2.24

where $\bar{\boldsymbol{f}}_* = K_{x_*,x}\left[K_{x,x} + \sigma^2 I\right]^{-1}\boldsymbol{y}$ and $cov(\boldsymbol{f}_*) = K_{x_*,x_*} - K_{x_*,x}\left[K_{x,x} + \sigma^2 I\right]^{-1}K_{x,x_*}$ [45], [48]. The mean and covariance of the posterior provide the estimation with an error bar.

Another explanation of the Gaussian process can be done graphically. Figure 2.3 shows graphical expression of the prior and the posterior. Figure 2.3(a) represents a huge number of function which is considered that one of them is the one is sought. This is the prior which can be built as it is in Equation 2.19. There is no knowledge about the data in the prior. The crosses on the figure indicate the data observed. They are known and consequently, shape the posterior. The functions fit the crosses are kept as shown in Figure 2.3(b) and the posterior is created.

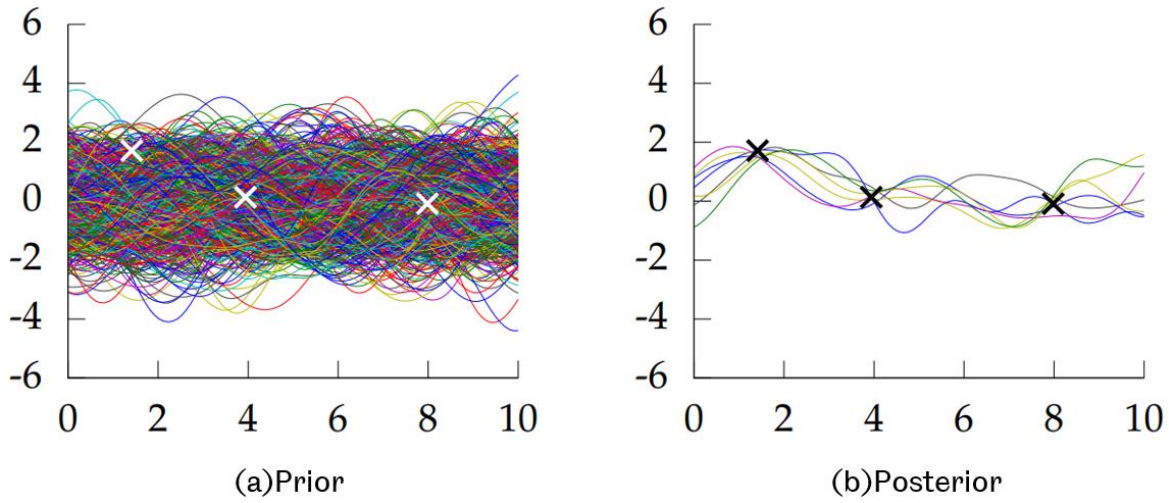

Figure 2.3 Graphical representation of the prior and the posterior [50]

The key point is that the reduction in the number of functions from the prior to the posterior happens in an analytical way rather than a computational way [50]. It uses matrix manipulations as it was stated above. This also saves computation time. It is obvious that a more specific prediction can be possible with the increase in the number of observations (the crosses in the figure). What it obtains at the end of the process is the posterior means and standard deviations. These provide a prediction of the model with a confidence region. The function predicted passes over the posterior mean and the borders of the region are created by the standard deviation.

3 METHODOLOGY

The chapter gives the details about how the study is conducted. First of all, a definition of the superharmonic is given and secondly, the application of SADE is explained comprehensively. The role of the transformation and the cost function are defined. How the mutation strategies and the parameters are adapted is clarified. Finally, the Gaussian process is detailed. One thing must be borne in mind is that this chapter is built over the information in Chapter 2.

3.1 Superharmonic

Superharmonic is one of the nonlinear phenomena which leads to the presence of the energy (amplitude) at the higher harmonics of the forcing frequency in vibration response. This is because of the nature of the nonlinear equations of the motion. Although the system of interest is forced at a single forcing frequency, the higher harmonics of that forcing frequency appears in the response due to nonlinear terms in the equation. To show this phenomenon, an equation of motion can be written

$$m\ddot{x}(t) + c\dot{x}(t) + k_1x(t) + k_2x^2(t) + k_3x^3(t) = F_f sin(wt - \Phi)$$

Equation 3.1

where m, c, k, k_2 and k_3 are mass, damping, linear stiffness, quadratic stiffness and cubic stiffness, respectively. The system is forced at a single forcing frequency w with an amplitude F_f and a phase Φ. If a trial solution $x(t) = X\sin(wt)$ is substituted into Equation 3.1, the equation turns by using trigonometric transformation

$$-mw^2Xsin(wt) + cwXcos(wt) + k_1Xsin(wt) + \frac{1}{2}X^2k_2(1 - \cos(2wt)) + k_3X^3\left(\frac{3}{4}\sin(wt) - \frac{1}{4}\sin(3wt)\right) = F'sin(wt)cos\Phi - F'sin\Phi\cos(wt)$$

Equation 3.2

Equation 3.2 contains $(2wt)$ and $(3wt)$ which are the higher harmonics of the forcing frequency (wt). This situation compels the initial trial solution to be updated. However, even if the higher harmonics are included by the trial solution, far higher harmonics appear in the response. This is the explanation of where superharmonics come. Therefore, the response of Equation 3.2 is expected in a similar form to Equation 2.3. To obtain an accurate result, the trial solution must contain all harmonics of the forcing frequency.

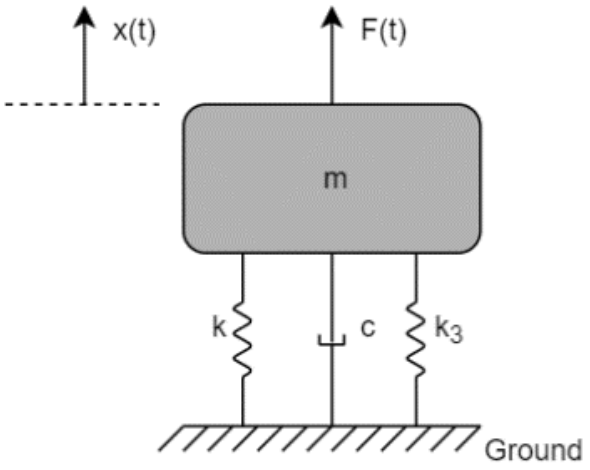

Figure 3.1 A nonlinear single degree-of-freedom system with a cubic stiffness

A similar expression can be derived from an equation with only cubic stiffness term as it was derived to explain Harmonic balance method in section 2.1.3.1. Therefore, a nonlinear single degree-of-freedom system with a cubic stiffness was taken as the system of interest as shown in Figure 3.1. The equation of motion can be written as

$$m\ddot{x}(t) + c\dot{x}(t) + kx(t) + k_3x^3(t) = F_f cos(wt)$$

Equation 3.3

or the equation can be expressed as the natural frequency and damping ratio

$$\ddot{x}(t) + 2\xi w_n\dot{x}(t) + w_n^2x(t) + ax^3(t) = fcos(wt)$$

Equation 3.4

where ξ is the damping ratio, w_n is the natural frequency, a is the nonlinearity parameter, w is the forcing frequency and f is the force parameter. This equation is named Duffing's equation in the literature. It is one of the most common single degree-of-freedom equations which is studied on since it reflects most of the behaviour of general nonlinear systems [6]. In the Duffing's equation case, only odd harmonics appear in the response as superharmonics as derived in Equation 2.5.

The coefficients of the equation were defined as $m = 1, \xi = [0.001, 1], w_n = 1, a = [1000\ 100000]$, $w = 1$ and $f = 100$. To test the robustness of the findings, the algorithm was run many times and the damping ratio and the nonlinearity parameters were chosen randomly between the values defined above in each time. The solution of Equation 3.4 was obtained numerically using a Runge-Kutta formula [51] and the displacement $x(t)$ was evaluated in the following processes.

The displacement results were investigated in the frequency domain rather than in the time domain. In this way, the amplitudes in each harmonic could be seen clearly. Power spectral density estimation was obtained using Welch's method. The method provides the obtainment of the power of the signal at frequencies defined.

Figure 3.2 illustrates an example of the displacement response of Equation 3.4 in the time domain. It can be easily observed that the form of the response is different from a single harmonic response. Higher harmonics can be detected even in the time domain response. This is also one of the nonlinearity detection methods as mentioned about in Chapter 2 since the response is not in single harmonic form for a single harmonic force. More convenient way to see the amplitude at each frequency is power spectral density estimation. Figure 3.3 shows the power spectral density estimation of the signal in Figure 3.2. The higher harmonics, superharmonics, in the response can be easily observed. The peaks at around 1, 3, 5 and 7 Hz, only odd harmonics as expected, exist since the forcing frequency is 1 Hz. Furthermore, the natural frequency of the system is 1 Hz and this causes higher peak at that frequency due to resonance.

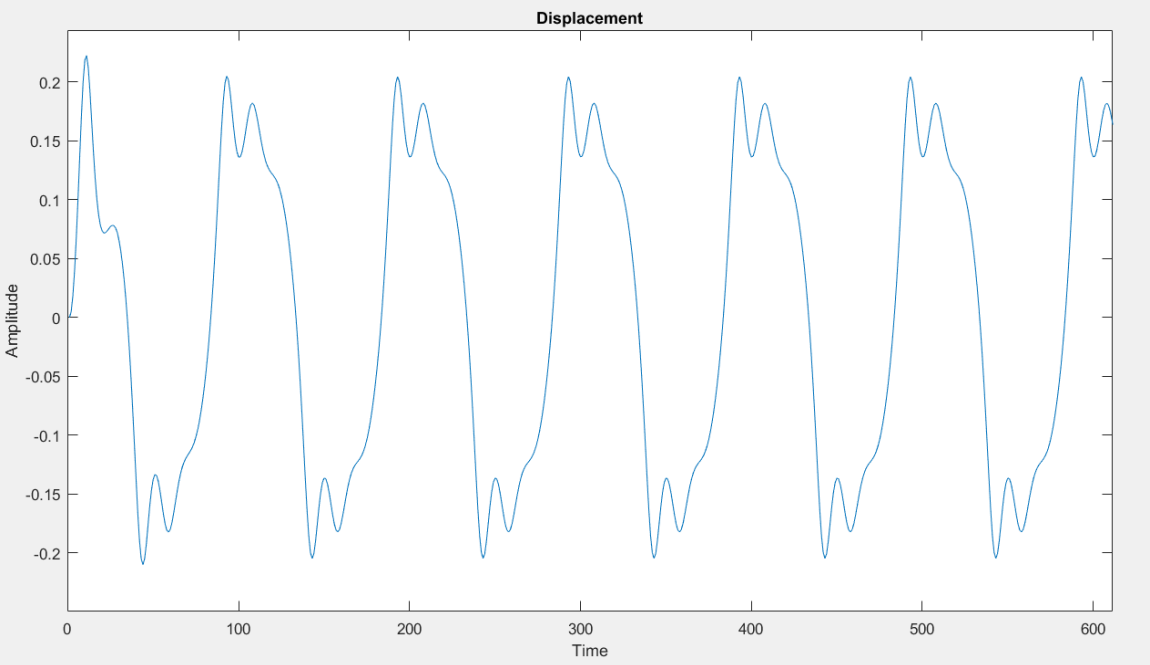

Figure 3.2 Displacement response of the forced Duffing's equation.

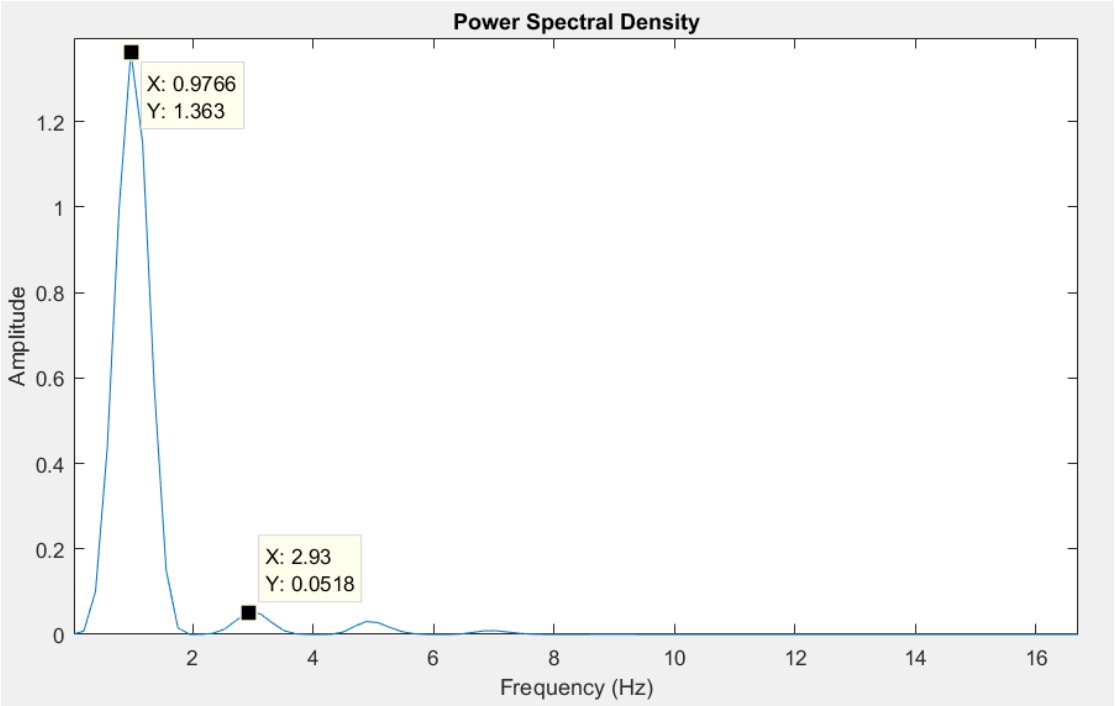

Figure 3.3 Power spectral density of the displacement response of the forced Duffing's equation.

Until this point, what it has been done is that the response of a nonlinear system which contains superharmonics has been captured with a numerical method. The removal of these superharmonics happens with the application of SADE.

3.2 Application of SADE

The response of a nonlinear system might contain superharmonics as it is in Figure 3.3. The idea of this study is to remove or reduce the amplitude of these superharmonics by using a transformation. The transformation can be formed in a polynomial expansion and thus, the coefficients of the transformation can be found using SADE. SADE is a self-adaptive algorithm which means that the algorithm can adapt its own parameters. Still, there are some crucial parameters have to be decided by users as it was stated before. The mutation strategies, cost function, population size and initial range for coefficients of the transformation are considered by users. Also, the transformation itself is an important role to remove harmonics.

The transformation can be written

$$z(t) = a_0 + a_1 x(t) + a_2 x(t)^3 + \cdots + a_n x(t)^{2n-1}, \quad n: integer$$

Equation 3.5

where $x(t)$ is the original displacement signal, $z(t)$ is the transformed signal, a_n is the coefficients of the transformation. SADE seeks a set of parameters $[a_0\ a_1\ \dots\ a_n]$ which provides a transformed signal with the amplitudes of higher harmonics reduced or without higher harmonics.

3.2.1 Mathematical Expression

A mathematical approach to the transformation is helpful to show the duty of the transformation and SADE in the cancellation of superharmonics, as well as to give an idea about the structure of the cost function. The mathematical approach can be made by assuming a response with only one superharmonic. If the response $x(t) = X_1\cos(wt) + X_2\cos(3wt)$ is substituted into Equation 3.5 for $n = 3$, the transformation becomes

$$z(t) = a_0 + a_1[X_1\cos(wt) + X_2\cos(3wt)] + a_2[X_1\cos(wt) + X_2\cos(3wt)]^3$$

Equation 3.6

After expansion of the cubic term and trigonometric transformations, the equation can be expressed as

$$\begin{aligned} z(t) = a_0 &+ \left[a_1X_1 + a_2\left(\frac{3X_1^3}{4} + \frac{3X_1^2X_2}{4} + \frac{3X_1X_2^2}{2}\right)\right]\cos(wt) \\ &+ \left[a_1X_2 + a_2\left(\frac{X_1^3}{4} + \frac{3X_2^3}{4} + \frac{3X_1^2X_2}{2}\right)\right]\cos(3wt) \\ &+ a_2\left(\frac{3X_1^2X_2}{4} + \frac{3X_1X_2^2}{4}\right)\cos(5wt) + a_2\left(\frac{3X_1X_2^2}{4}\right)\cos(7wt) \\ &+ a_2\frac{X_2^3}{4}\cos(9wt) \end{aligned}$$

Equation 3.7

Or

$$\begin{aligned} z(t) = a_0 &+ [a_1C_1 + a_2C_2]\cos(wt) + [a_1C_3 + a_2C_4]\cos(3wt) + a_2C_5\cos(5wt) \\ &+ a_2C_6\cos(7wt) + a_2C_7\cos(9wt) \end{aligned}$$

Equation 3.8

where $C_1, C_2, \dots, C_7$ are constants which depend on the original response signal and, a_0, a_1 and a_2 are the coefficients of the transformation. What the transformation does here is to provide a polynomial expansion and in that way, the coefficients of the transformations are set to reduce or remove the amplitude of superharmonics. In other words, what is tried to do is to set zero to the coefficients of the higher harmonics, $\cos(3wt)$, $\cos(5wt)$, $\cos(7wt)$ and $\cos(9wt)$, in Equation 3.8. It is obvious that setting zero to the coefficients of the higher harmonics is not possible without

zeroing first harmonic in this example. This is because a_2 must be zero to cancel the fifth, seventh and ninth harmonics and consequently, a_1 must be zero to cancel the third harmonics but the first harmonic also becomes zero in this case. The point is that this is just an example to explain the duty of the transformation in terms of mathematics. In a real situation, it is highly likely that the original signal has more than one superharmonic. Moreover, the transformation might have higher order terms, instead of third-order transformation. This means the increase the number of terms and harmonics in Equation 3.8 so it becomes possible to have zero in the amplitude of the superharmonics while the first harmonic is not zero. However, the choice of the appropriate set of the parameters becomes more complicated and demanding. Differential evolution algorithm can be made use of to find the optimum parameters.

3.2.2 Cost Function

The expression in Equation 3.8 also gives an indication about how the cost or objective function must be built. The focus of the cost function must be the amplitudes of the superharmonics. Hence, the cost function can be chosen as the sum of the amplitudes in a frequency range around superharmonics frequencies. For instance, the frequency range can be from 2 Hz to 8 Hz in a case which is similar to Figure 3.3 because superharmonics appear between these frequencies. The peaks are not like delta functions, which are very sharp. They have the width so ±1 Hz can be added to the frequency of the first and last superharmonics. Therefore, the cost function can be taken as the sum of the amplitudes from 2 Hz to 8Hz. The aim is to find a set of parameters of the transformation which provides the minimum sum by using SADE algorithm. That is why the problem can be described as an optimization problem or more specifically, a minimization problem.

There is no doubt that one of the best ways to calculate the sum of the amplitudes at corresponding frequencies is to use power spectral density. This provides the amplitude values at the frequencies depending on the sensitivity. The sensitivity also refers to the resolution of the power spectral density. The frequency sensitivity for power spectral density was 0.0488 Hz during the study. This is an adequate sensitivity when the natural frequency of the system and the forcing frequency are considered. One of the methods to obtain amplitudes in the frequency domain is power spectral density as stated before. Hence, each amplitude is obtained by using power spectral density and the amplitudes from 2 Hz to 8 Hz are added to calculate the cost function value. The cost function can be formulated as

$$f_{cost} = \sum_{n=2\ Hz}^{8\ Hz} A_n$$

Equation 3.9

where f_{cost} is the cost function or objective function and A_n is the amplitude at each frequency. n starts from 2 Hz and increases by the sensitivity, 0.0488 Hz.

The parameters of the transformation define the result of the cost function. This means the configurations of SADE are crucial to have a valid transformed result. The configurations of SADE are consisted of mutation strategies, population size and initial range.

3.2.3 Mutation Strategy and Parameter Adaptation

The mutation strategy has a key role to define the mutant vector and different strategies have different characteristics. Therefore, a mutation strategy pool is created in SADE by using different strategies and the algorithm adapts its strategy according to previous results. It is important to eliminate the ineffective strategies and use a limited number of strategies for the mutation strategy pool [37]. In this way, the algorithm does not use these ineffective algorithms and it can be less time-consuming. Four different strategies, whose success were stated in [37], were used in this study. These are:

- DE/rand/1:

$$\boldsymbol{V}_{i,G+1} = \boldsymbol{X}_{r_1^i,G} + F.\left(\boldsymbol{X}_{r_2^i,G} - \boldsymbol{X}_{r_3^i,G}\right)$$

Equation 3.10

- DE/rand-to-best/2:

$$\boldsymbol{V}_{i,G+1} = \boldsymbol{X}_{i,G} + F.\left(\boldsymbol{X}_{best,G} - \boldsymbol{X}_{i,G}\right) + F.(\boldsymbol{X}_{r_1^i,G} - \boldsymbol{X}_{r_2^i,G}) + F.(\boldsymbol{X}_{r_3^i,G} - \boldsymbol{X}_{r_4^i,G})$$

Equation 3.11

- DE/rand/2:

$$\boldsymbol{V}_{i,G+1} = \boldsymbol{X}_{r_1^i,G} + F.\left(\boldsymbol{X}_{r_2^i,G} - \boldsymbol{X}_{r_3^i,G}\right) + F.\left(\boldsymbol{X}_{r_4^i,G} - \boldsymbol{X}_{r_5^i,G}\right)$$

Equation 3.12

- DE/current-to-rand/1:

$$\boldsymbol{V}_{i,G+1} = \boldsymbol{X}_{i,G} + K.\left(\boldsymbol{X}_{r_1^i,G} - \boldsymbol{X}_{i,G}\right) + F.(\boldsymbol{X}_{r_2^i,G} - \boldsymbol{X}_{r_3^i,G})$$

Equation 3.13

where $\boldsymbol{X}$ is the parameter vector, $\boldsymbol{V}$ is the mutant vector, F is the scaling factor, K is a random control parameter $\in [0,1]$, $\boldsymbol{X}_{best}$ is the parameter vector which gives the minimum cost value and $\boldsymbol{X}_i$ is the corresponding target vector as explained in Section 2.2. Indexes $r1, r2, \ldots$ mean that those vectors are chosen randomly and G represents the generation. The mutant vector generated via one of the strategies above goes to crossover operation and then, selection operation happens. The processes for these operations occur as stated in Section 2.2.

Both the scaling factor (F) and crossover constant (CR) is defined by a normal distribution. Even though their mean values are adapted by the algorithm, an initial mean value is required to define. Initially, the scaling factor and crossover constants were created according to the normal distribution $N(0.5,0.3)$ and $N(0.5,0.1)$, respectively. The means were 0.5, and the standard deviations were 0.3 and 0.1 for the scaling factor and crossover constant, respectively. The standard deviations were kept constant during the evaluation while the means were adjusted according to a set of the previous vectors which succeeded to be chosen for next process. The mean of scaling vector was forced to take a value $\in [0,2]$. Therefore, if the mean of the scaling factor was adapted as smaller than 0, it was taken 0. Similarly, in case that it exceeded 2, it was taken 2. The length of the set of the previous vectors is depended on the learning period. It was 10 runs which means that the mutation strategy, the scaling factor and crossover constant were updated according to last 10 generations. As an instance, Figure 3.4 illustrates the change in scaling factor during evolution. The scaling factor took the same value during the consecutive 10 generations and then, it was updated. The crossover constant and the mutation strategy also follow the rule.

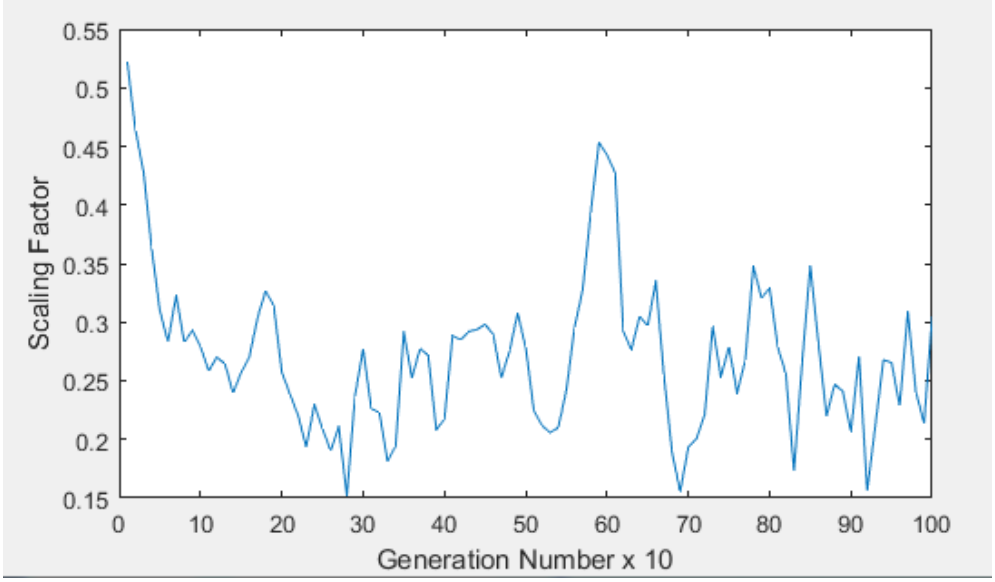

Figure 3.4 The change in the scaling factor during the evolution

The initial population, where all the next generations come from, is randomly built by uniformly distributed numbers with a range. The boundary of this initial range is set before the run of the algorithm. The algorithm also checks for whether the mutated solutions are inside the range or outside the range. If one of them is larger than the upper boundary of the range, that parameter is assumed the upper boundary value of the range. If it is smaller than the lower boundary of the range, it is taken the lower boundary value of the range. At the end of the evolution, parameters are checked and if any of the parameters of the transformation equal to one of the boundaries of the range, the initial range can be expanded. A favourable initial range can be provided with the help of this trial-and-error method. By following similar way, the initial range was taken ±500.

The population size is another user defined parameter. It is reasonable to expect that the population size increases with the number of the parameters of the transformation. Hence, the size of the population matrix was taken 10 times the number of parameters by the number of parameters. It was a 60 by 6 matrix for 6 parameters.

The differential evolution algorithm is normally run until the stopping criteria are reached. The stopping criterion for this specific problem could be considered that all amplitudes from 2 Hz to 8 Hz must be zero. However, this would be a criterion which is very difficult to be satisfied. Even small values of the sum of the amplitudes might be a problem for the stopping criterion. It would still be troublesome due to the difficulty of the decision how small it must be. It would be a time-consuming process. Hence, instead of choosing a cost value as a stopping criterion, a stopping generation number was decided. The evolution was stopped when that generation number was reached. It was considered that the generation number is enough large to deal with the local minimum.

Figure 3.5 demonstrates the change in the minimum cost value with the generation. In each attempt, the algorithm started from the beginning during 1000 generations. As it can be clearly seen in the 3 attempts, there is no change in the cost value after around 600th generation. Hence, 1000 generations were taken as the stopping criterion with a safety distance.

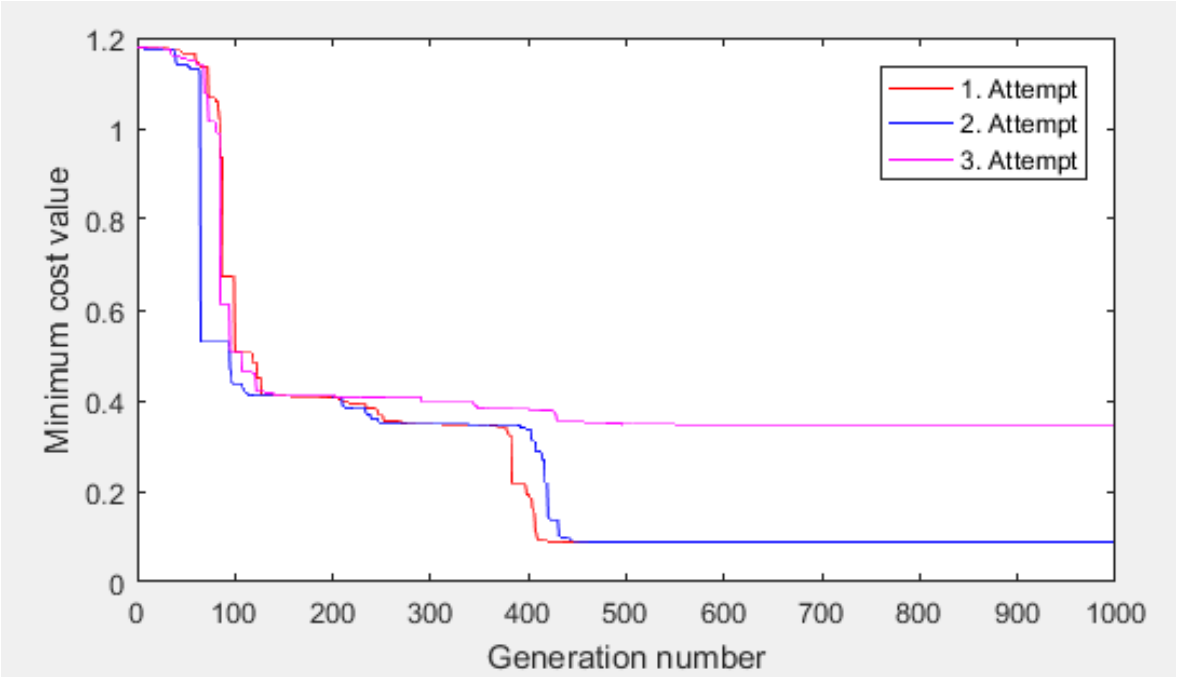

Figure 3.5 The minimum cost value for different attempts

SADE can be seen as a partly random method to use for optimization problems. It is random because the initial population and decisions in the mutation and crossover operations are made randomly. It is partly since the algorithm is guided by the strategy and the control parameters. This partly random method might be required to run more than once because the solution may stick around a local minimum or another initial population may perform a better result. Therefore, SADE ran three times for the same m, ξ, w_n, a, w and f in Equation 3.4. The parameters were decided among the three results.

3.2.4 Decision of the Number of Parameters of the Transformation.

Although it was stated that the transformation is in a polynomial expansion form, the exact transformation has not been given yet. There are different options to decide the exact transformation. One option is Equation 3.5, whose form can be named as Transformation A with the number of the coefficients in this study. Therefore, Transformation A6 refers to a transformation with 6 coefficients from a_0 to a_5 in the form of Equation 3.5

$$z(i) = a_0 + a_1 x(i) + a_2 x(t)^3 + a_3 x(i)^5 + a_4 x(i)^7 + a_5 x(i)^9$$

Equation 3.14

where i represents for the sample, instead of time in Equation 3.5. Likewise, Transformation A10 refers to a transformation with 10 coefficients from a_0 to a_9 in

the same form. It is possible to increase the number of coefficients with the same logic. Another option can be written as

$$z(i) = a_0 + a_1 x(i) + a_2 x(t)^2 + a_3 x(i)^3 + a_4 x(i)^4 + a_5 x(i)^5$$

Equation 3.15

and can be named Transformation B6 with a similar approach. This form was also tested to see whether it causes a big difference in terms of removal of the superharmonics. To decide the exact transformation, some selected possible transformations were tested and the power spectral density of the result of the transformations are given in Figure3.6.

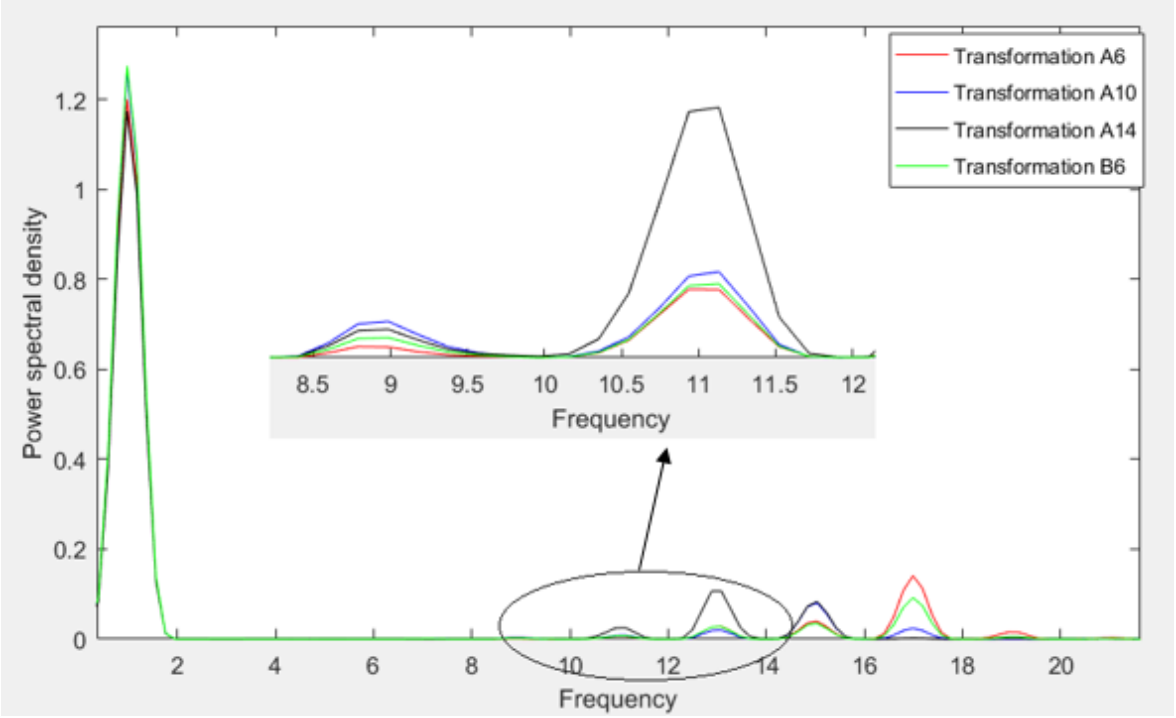

Figure 3.6 Comparison of the power spectral density of the results of the different transformations

All transformations tested in Figure 3.6 could successfully remove the first three harmonics but Transformation A6 (Equation 3.14) showed a slightly better performance for the rest. This is not the only reason why Transformation A6 was chosen as the transformation. The differential evolution can be processed less time with the less number of coefficients. When the number of coefficients increases, the size of the population and the number of the operations increase and as a result, the evolution takes longer time. The solution of the differential evolution algorithm might take serious time depending on the number of generation and coefficients of the transformation. The computation time is specifically one of the important

factors to decide transformation in this study as a Monte Carlo simulation would be conducted as a part of the study. When these all are considered, the choice of Equation 3.14 as the transformation might be reasonable. Thus, the results related to SADE in the next chapter were generated by using Equation 3.14 as the transformation.

3.2.5 Summary of the Procedure

The application of SADE is the most significant part of this study. Therefore, it might be beneficial to summarize the procedure with the light of the information discussed above. It can be summarized as follows:

- An initial population is created with randomly chosen numbers which are uniformly distributed by using the initial range, from -500 to 500. It is a 60 by 6 matrix can be shown:

$$Initial\ population = \begin{bmatrix} a_{1,0} & a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} & a_{1,5} \\ a_{2,0} & a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} & a_{2,5} \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \\ a_{n,0} & a_{n,1} & a_{n,2} & a_{n,3} & a_{n,4} & a_{n,5} \end{bmatrix}$$

Equation 3.16

  where n equals to the number of the coefficients of the transformation times 10 so $n = 60$.

- $[a_{i,0}, a_{i,1}, a_{i,2}, a_{i,3}, a_{i,4}, a_{i,5}]$; $i = 1,2, \dots, n$ is a set of coefficients. Each set in the initial population is substituted into Equation 3.14 and the cost value of each of them is calculated by using Equation 3.9.

- Mutation operation is operated by using the strategies from Equation 3.10 to Equation 3.13. The same strategy is applied to all coefficients in the same set. The probability of choosing strategy is initially 0.25 for each strategy since there are 4 strategies in the strategy pool. The scaling factors, which initially obey the normal distribution $N(0.5,0.3)$, randomly are generated and a mutation pool (60 by 6 matrix) is created.

- Crossover operation is conducted by using the mutation pool. For this, a binomial matrix (60 by 6 matrix) whose all elements $\in [0,1]$ are randomly chosen is generated. Each element in the binomial matrix is compared with the crossover constant. The initial crossover constants are also randomly generated with the normal distribution $N(0.5,0.1)$ and each set has the same crossover constant.

  If the crossover constant is higher than the binomial value in the comparison, the parameter in the mutation pool is chosen. If the crossover constant is

smaller than the binomial value, then the corresponding parameter in the initial population is taken. Together with this, it is made sure that at least one parameter in each set comes from the mutation pool. Therefore, a new matrix (60 by 6 matrix) which contains 60 trial sets/vectors is obtained.

- The next is the selection operation where the next generation population is built. The cost values of these trial sets are calculated and compared with the corresponding cost values in the previous population (in this case, the initial population). The sets/vectors are transferred to next generation. In that way, a second-generation population is created.
- The probability of choosing strategy, the means of the normal distribution of the scaling factor and the crossover constant are updated in each learning period according to success of the parameters in the previous generations. The learning period is chosen 10 generations.
- The same steps are repeated until the 1000^{th} generation is reached.
- The set of parameters which gives the minimum cost value is chosen as the coefficients of the transformation.

The aim is to find the parameters which provide the minimum sum of the amplitudes at frequencies from 2 Hz to 8 Hz.

3.3 Monte Carlo Simulation

The differential evolution method is not an analytical method and it contains random processes, though a framework is drawn. Thus, the Monte Carlo simulation is required to evaluate the reliability and robustness of the algorithm. The performance of the algorithm is tested for the different cases of Equation 3.4 rather than one case. The results were obtained by randomly taking $\xi = [0.001, 1]$ and $a = [1000\ 100000]$ in Equation 3.4.

The algorithm was run 540 times to see the scenarios as many as possible. The gains at the frequencies of from the first superharmonic to the fifth superharmonic are the interest of Monte Carlo simulation. The gain is calculated in the following form

$$Gain = \frac{The\ amplitude\ after\ transformation}{The\ amplitude\ before\ transformation}$$

Equation 3.17

The ratio of the amplitude after transformation to the amplitude before transformation at the same harmonic gives the gain at that harmonic. To evaluate the Monte Carlo simulation result, the gain can be used because it shows the changes in the amplitude at the corresponding harmonic after transformation. Each

amplitude at the corresponding frequency was defined with a range. Although the range was different for each harmonic, it was around 1.5 Hz. For instance, to calculate the total amplitude at the third harmonic, the sum of the amplitude was taken from 2.34 Hz to 3.9 Hz at intervals which were the sensitivity of the power spectral density. The gains at the corresponding frequencies will be given in the next chapter.

The Monte Carlo simulation provides a set of data which contain different possible results of a system. In this study's case, the results differ because of the random choice of the damping ratio and the nonlinearity parameter. Also, SADE might give different results in some cases due to the initial population which is created randomly.

3.4 Gaussian Process

The next step is to find the original signal from the transformed signal after successful application of SADE. The Gaussian process was performed for this purpose as the next step in this study. The Gaussian process estimates the future output of a system. It gives a Gaussian distribution of the output at each future point. This is basically done by creating a covariance function and Bayes' theorem as stated in Section 2.3. The code to operate the Gaussian process algorithm has been built by Matlab software in [52]. This code was used to perform the Gaussian process to obtain the original signal in the study. Different sets of data such as input vector, target vector and test vector were used for this purpose. These all three vectors were generated from the transformed signal. Figure 3.7 demonstrates a transformed signal. It contains high-order superharmonics (from fourth order to eleventh order), as it can be seen.

The original signal is normally in the time domain but it can be considered that the values in the transformed signal become dependent on the sample rather than the time. The time domain losses its meaning after the transformation. Therefore, x-axis in Figure 3.7 represents samples. The value for each sample in the transformed signal (y-axis in Figure 3.7) coincides with a value for the corresponding sample in the original signal (y-axis in Figure 3.2). Therefore, it can be said that the Gaussian process gives a normal distribution of the original signal value for each unknown future transformed signal values.

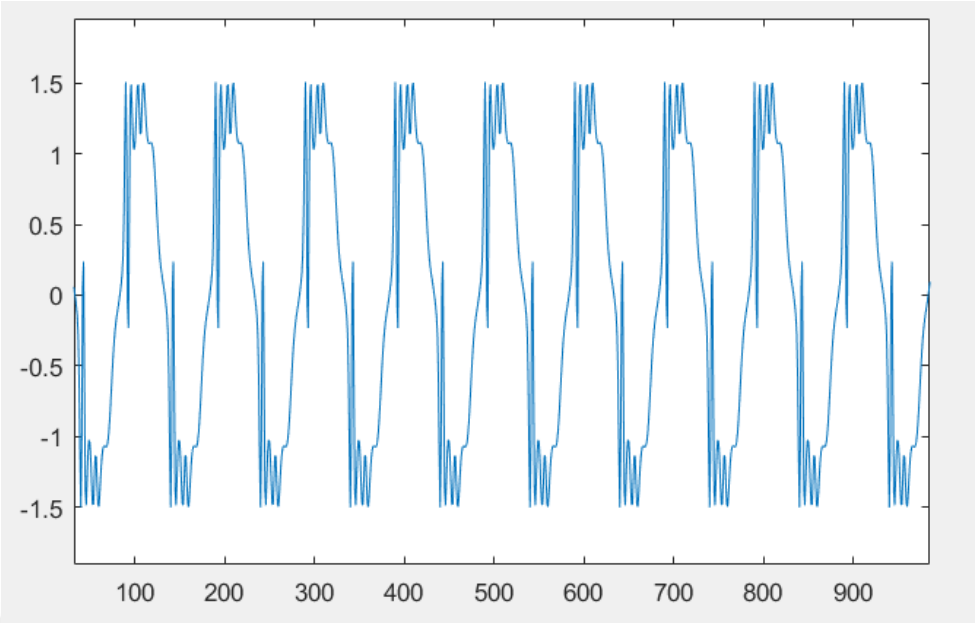

Figure 3.7 An example of the transformed signal

The Gaussian process was applied by creating vector following:

The training data were created. The inputs were entered as

$$\boldsymbol{x} = [x_1 \quad x_3 \quad \dots \quad x_n]$$

Equation 3.18

where x is the amplitude of the transformed signal. n index is the odd integers sample. Each input in the input vector coincides with an output or a target value and the target vector was built as

$$\boldsymbol{y} = [y_1 \quad y_3 \quad \dots \quad y_n]$$

Equation 3.19

where y is the amplitude of the original signal. Finally, the test vector, whose each element shows the future input which is desired to predict, was created as

$$\boldsymbol{xs} = [x_1 \quad x_2 \quad \dots \quad x_m]$$

Equation 3.20

where x is again the amplitude of the transformed signal. This time, m index also takes the even integers.

The original signal and transformed signal contain 10001 samples. As the signals consist of harmonics, taking only a part of the signal is sufficient to investigate. The rest of the signal is the repetitive form of the part of the signal taken. Therefore, 3001 samples were taken into consideration during the Gaussian process. In other

words, m index in Equation 3.19 was taken 3001 and only the odd indexes were considered in training.

The Gaussian process uses Bayes' theorem and consequently, the matrix operations in Section 2.3. The covariance function was created by using squared exponential (Equation 2.20) so the length-scale λ and the signal standard deviation σ_0 in Equation 2.20, and the noise standard deviation σ must be initialized. As they are generally unknown parameters, it is common to optimise them using the marginal likelihood. The hyperparameters were determined with this method as it is shown in [45]. The mean of the noise was taken zero as it was stated in Section 2.3.

With all these stated above, the prediction was plotted with 95% confidence by using the mean and the standard deviation of the posterior distribution. The results will be given in Chapter 4.

4 RESULTS

Self-adaptive differential evolution algorithm run to remove superharmonics in the response of the nonlinear system. The procedure which was followed has been given in the previous chapter. Moreover, the Monte Carlo simulation conducted to see the success of the algorithm to remove superharmonics. Transformation A6 was used for the analysis as the transformation. As the final step, the Gaussian process was applied to obtain the original signal from the signal transformed. This chapter presents all results have obtained during the study. It can be divided into three sections and in the first section, results of the SADE algorithm is given. The second section demonstrates results of the Monte Carlo simulation. Finally, the last section introduces result of the Gaussian process.

4.1 Result of SADE Algorithm

The model was created contains superharmonics, as seen Figure 3.3. After applied Self-Adaptive Differential Evolution algorithm as stated in Section 3.2. Figure 4.1 gives the power spectral densities of the response of the system and the output of the algorithm together. The blue line in the figure shows the original signal, which contains superharmonics. The red line is the transformed signal, which is the output of the algorithm.

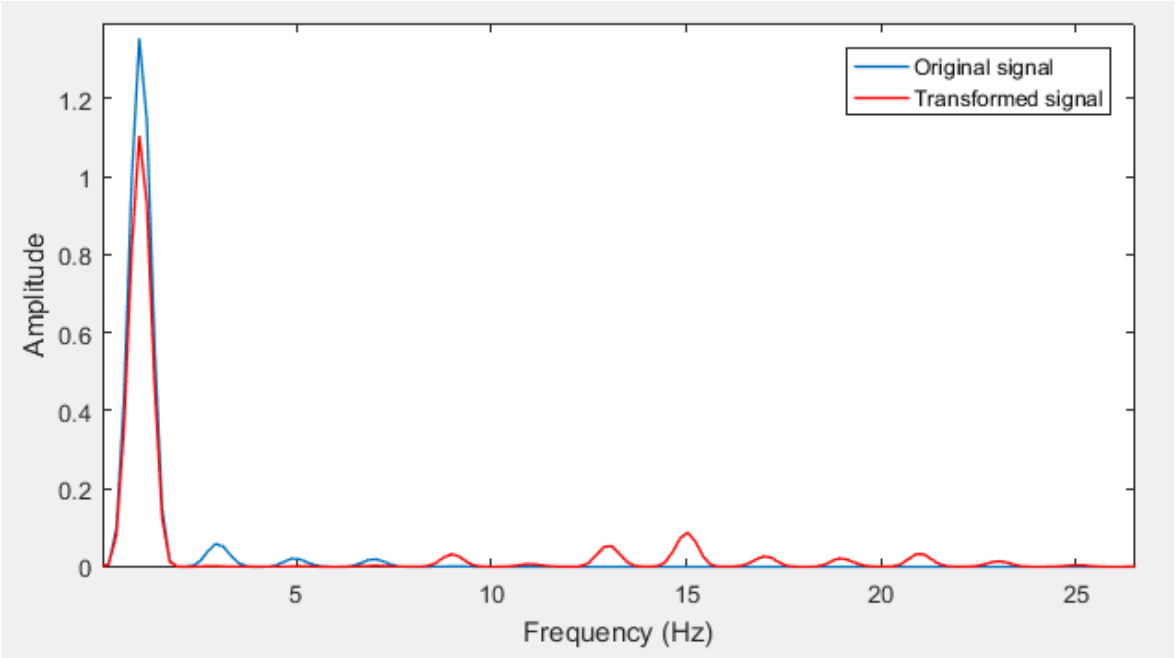

Figure 4.1 The power spectral density of the output of SADE

In the figures, the first, second and third superharmonics are removed in the transformed signal while the first harmonic is preserved. However, there is an increase in the amplitude at the harmonics after the third superharmonics. Figure 4.2 illustrates the first, second, third, fourth and fifth superharmonics zoomed. Another power spectral density of the output of SADE is given in Figure 4.3. In this figure, the peak of the first harmonics of the original and transformed signals are the same after transformation. Furthermore, the higher harmonics of the transformed signal have smaller harmonics.

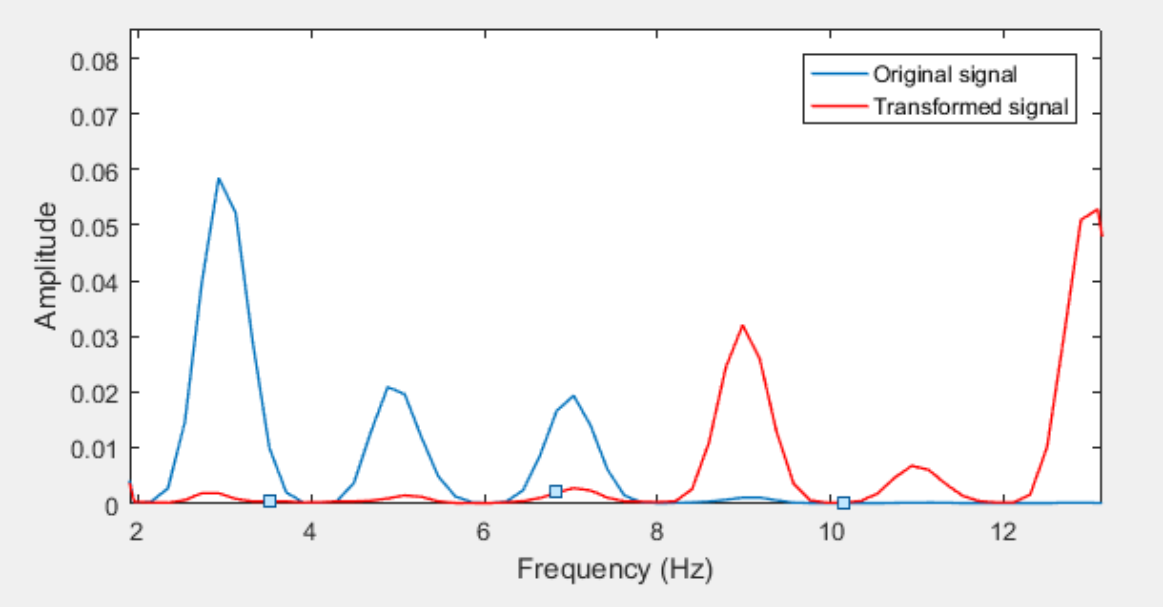

Figure 4.2 The zoomed view of the superharmonics from first order to fifth order

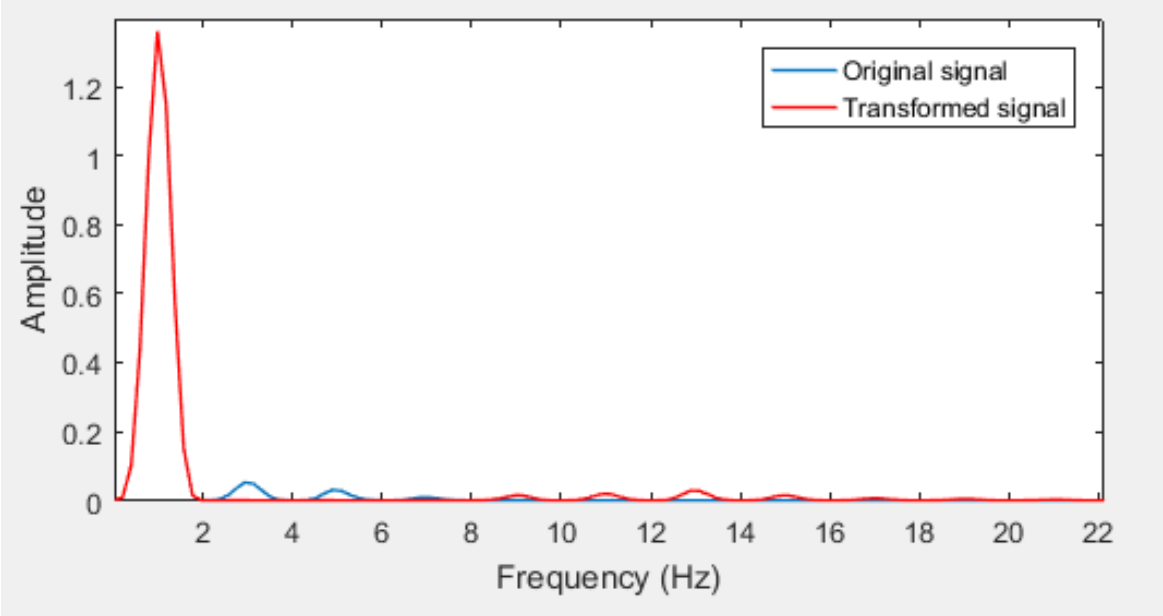

Figure 4.3 Another power spectral density of the output of SADE

The two different results have been given here are examples of the good results have been obtained. They are good examples to see that the algorithm really works. However, the algorithm does not always give results in such quality. The Monte Carlo simulation helps to see a set of results so a better perspective is obtained to evaluate the success of the algorithm. The next section will give the Monte Carlo simulation results. The further discussion about the results in this section will be drawn in the next chapter.

4.2 Result of Monte Carlo Simulation

These two figures above are separate outputs of the algorithm. The Monte Carlo simulation was also conducted to test the robustness of the algorithm and to see possible scenarios as many as possible. The higher number of runs is provided, the better inference about the success of the algorithm can be reached. Hence, the algorithm was run 540 times. The gain of each run at each corresponding harmonic was saved and then, they were classified to observe the frequencies of the gains.

The interest of the Monte Carlo simulation were the gains at the frequencies of from the first superharmonic to the fifth superharmonic. This is because, in the most cases, the first three superharmonics were seen in the original signal and also, an increase was observed at the frequencies after the harmonics have been removed or reduced. Therefore, fourth and fifth superharmonics were also investigated.

The results for each harmonic are given here with two figures. The former demonstrates the gains after transformation for each run, respectively. The latter figure classifies the gains in terms of the magnitude so it gives the frequencies at the gains as a histogram. Figure 4.4 and Figure 4.5 present the gains of the first superharmonic (third harmonic) given in order of run and according to the frequency of the gains as a histogram, respectively.

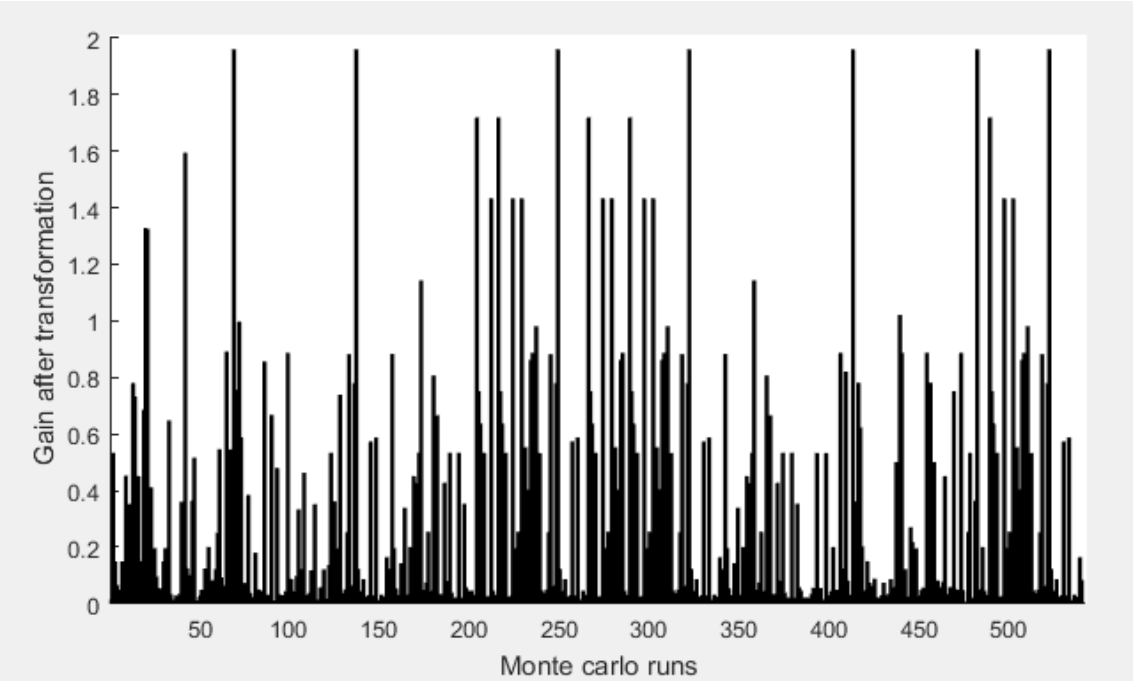

Figure 4.4 The gains at the third harmonic given in order of run

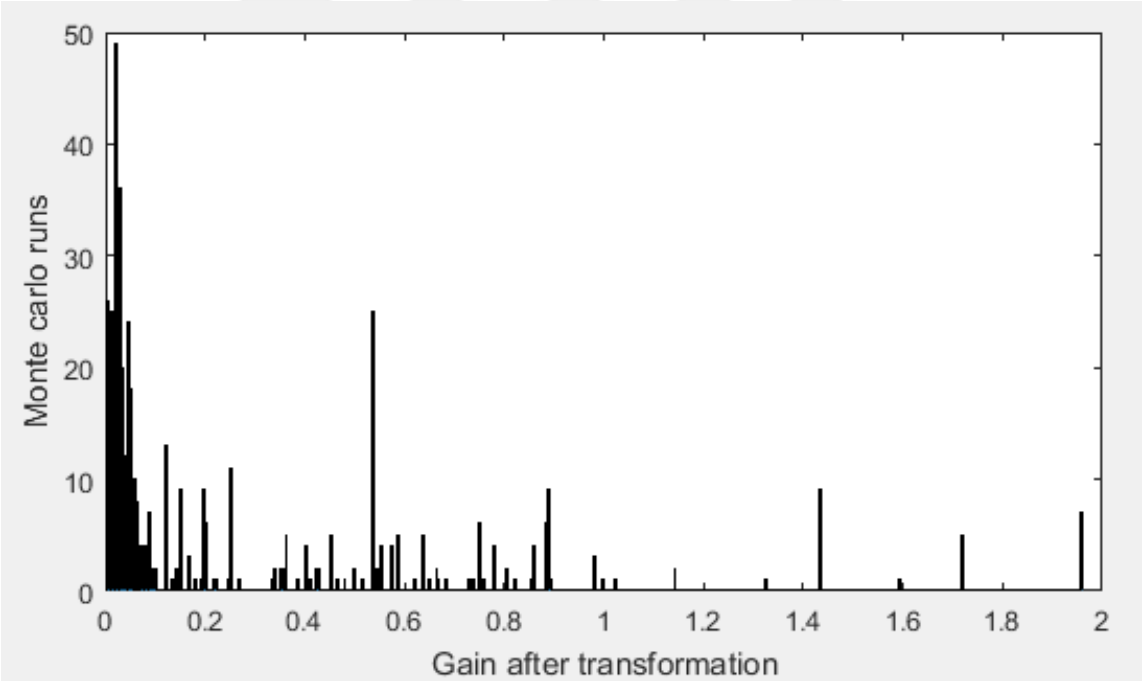

Figure 4.5 The frequency of the gains at the third harmonic

Similar figures for the second and third superharmonics are provided. Figure 4.6 and Figure 4.7 demonstrate the gains at the second superharmonic in order of run and as a histogram, respectively.

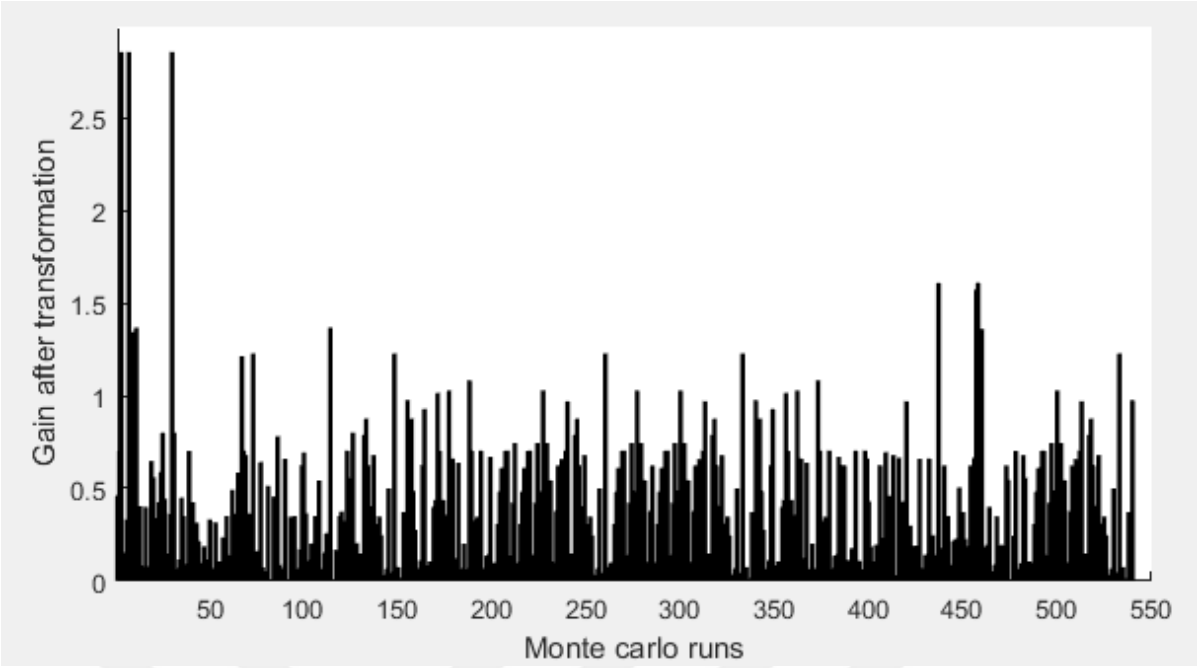

Figure 4.6 The gains at the fifth harmonic given in order of run

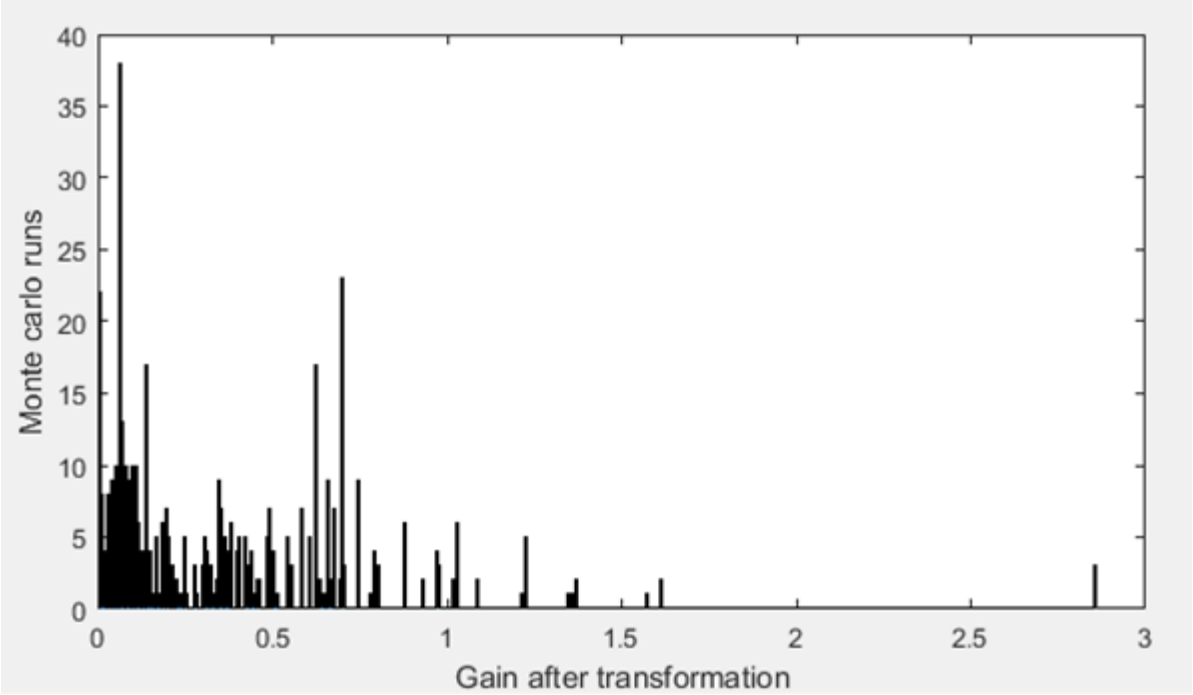

Figure 4.7 The frequency of the gains at fifth harmonic

Figure 4.8 and Figure 4.9 illustrate the gains at the third superharmonic in order of run and as a histogram, respectively. There are some high values of the gain in Figure 4.8. However, it can be seen in Figure 4.9 that the percentage of these high values are very small. Figure 4.9 focuses on the gains between 0 and 5. More specifically, the percentage of the gains which are smaller than 0.2 is around 85%. It is almost

93% for the gains which are smaller than 1. Also, these high values of the gain do not mean high values of the amplitude after transformation since the amplitudes of the original signal at the frequency of third superharmonics are already quite small.

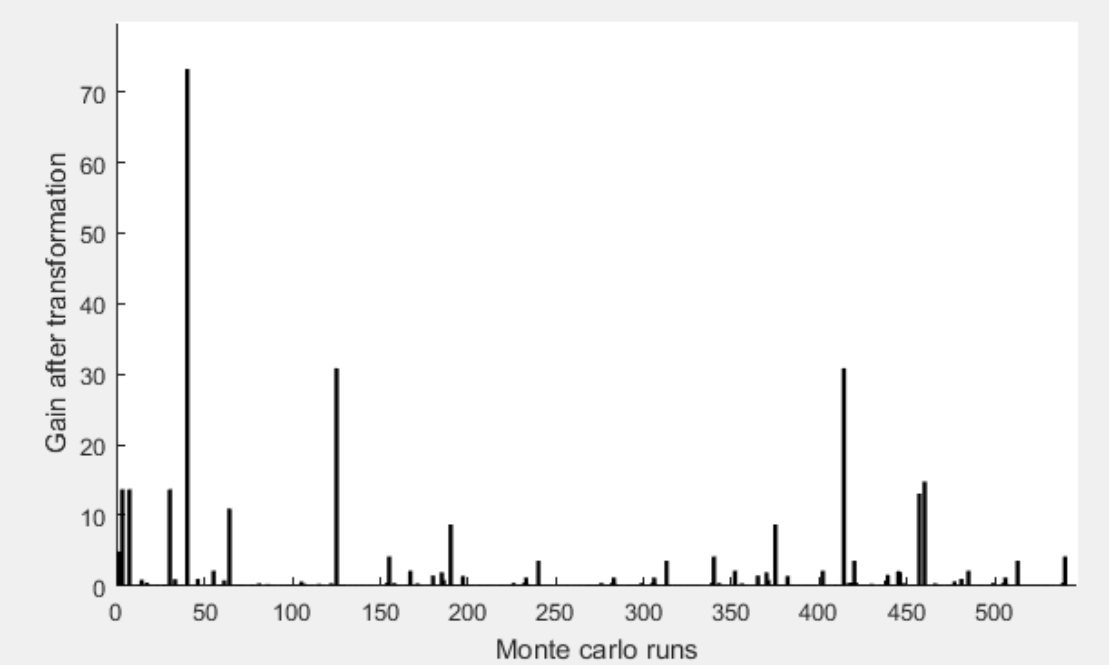

Figure 4.8 The gains at seventh harmonic given in order of run

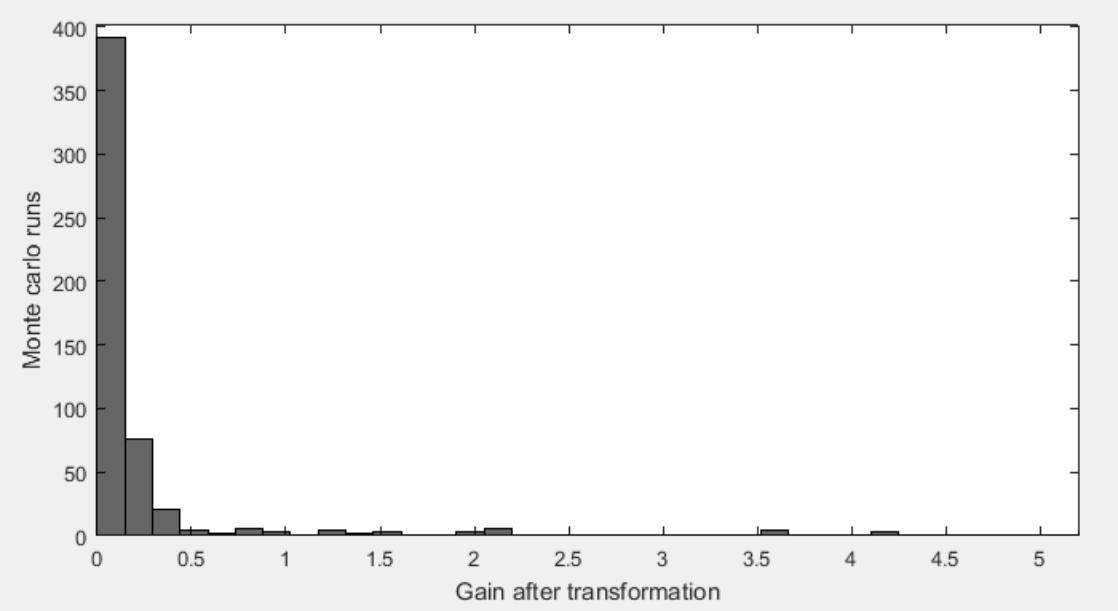

Figure 4.9 The frequency of the gains at seventh harmonic

The next results are about the ninth harmonic (the fourth superharmonic). This harmonic was not included to remove by the algorithm because there was no peak

at the harmonic exists in the original signal. However, after transformation, the signal had a peak at the corresponding frequency so the Monte Carlo simulation was conducted for this harmonic. Figure 4.10 shows the gains at the frequency of the fourth superharmonic in order of run. It can be said that after transformation, there is an increase in the amplitude at the fourth superharmonics when compared with the original signal. Some very high gains can be seen in the figure. This is because the amplitudes at corresponding harmonic were close to zero. Figure 4.10 demonstrates the frequency of the gains between 0 and 50 as a histogram. The percentage of the gains which are between 0 and 50 is almost 83%.

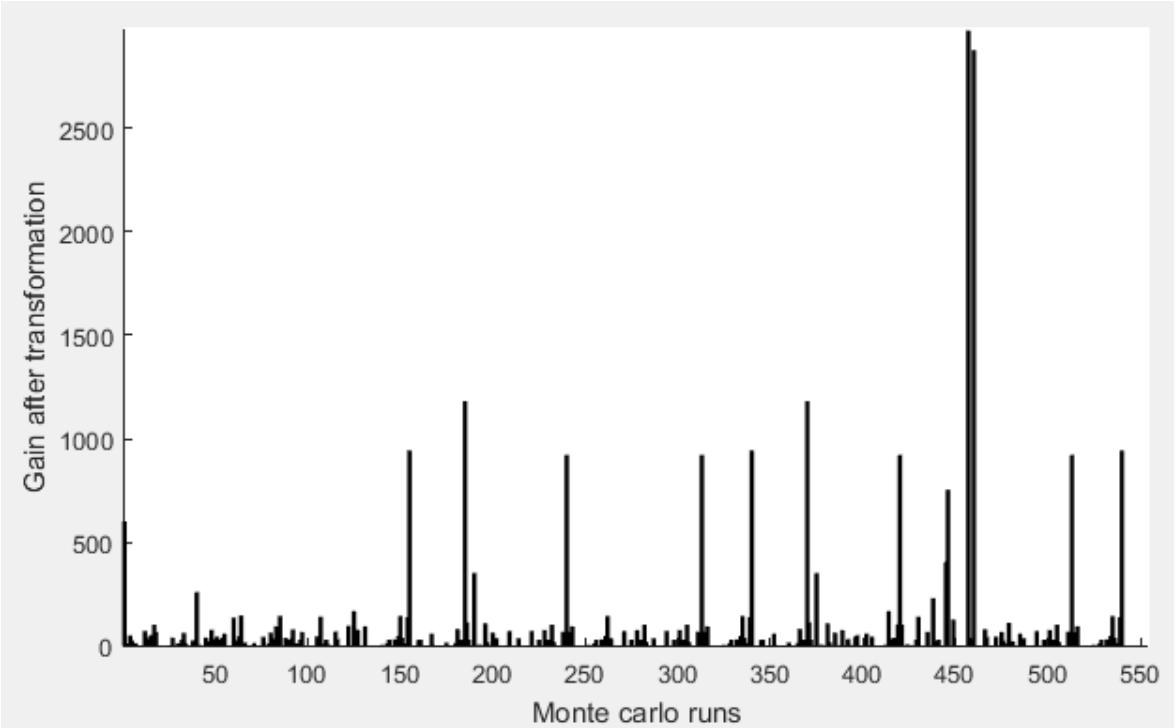

Figure 4.10 The gains at the ninth harmonic given in order of run

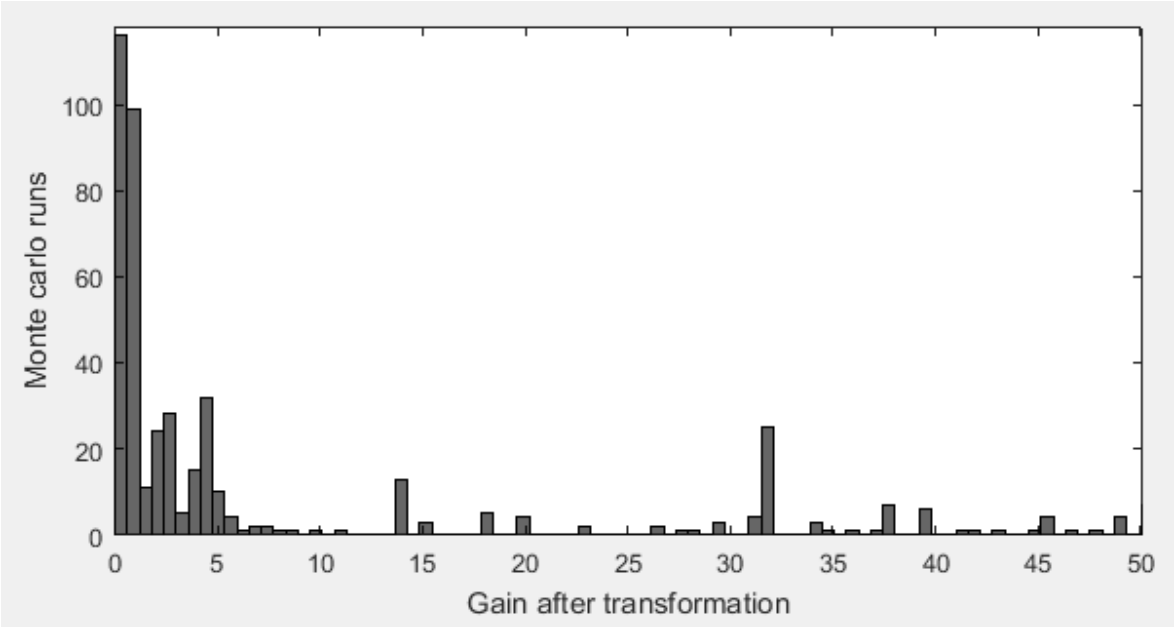

Figure 4.11 The frequency of the gains at the ninth harmonic

The final results in the Monte Carlo simulation are about the eleventh harmonic (the fifth superharmonic). Similarly, Figure 4.12 presents the gains at the fifth superharmonics in order of run. Figure 4.13 gives the frequency of the gains between 0 and 200. This corresponds 75% of the all gains. Same explanation as it is for fourth superharmonics can be given for the high gains at the fifth superharmonics.

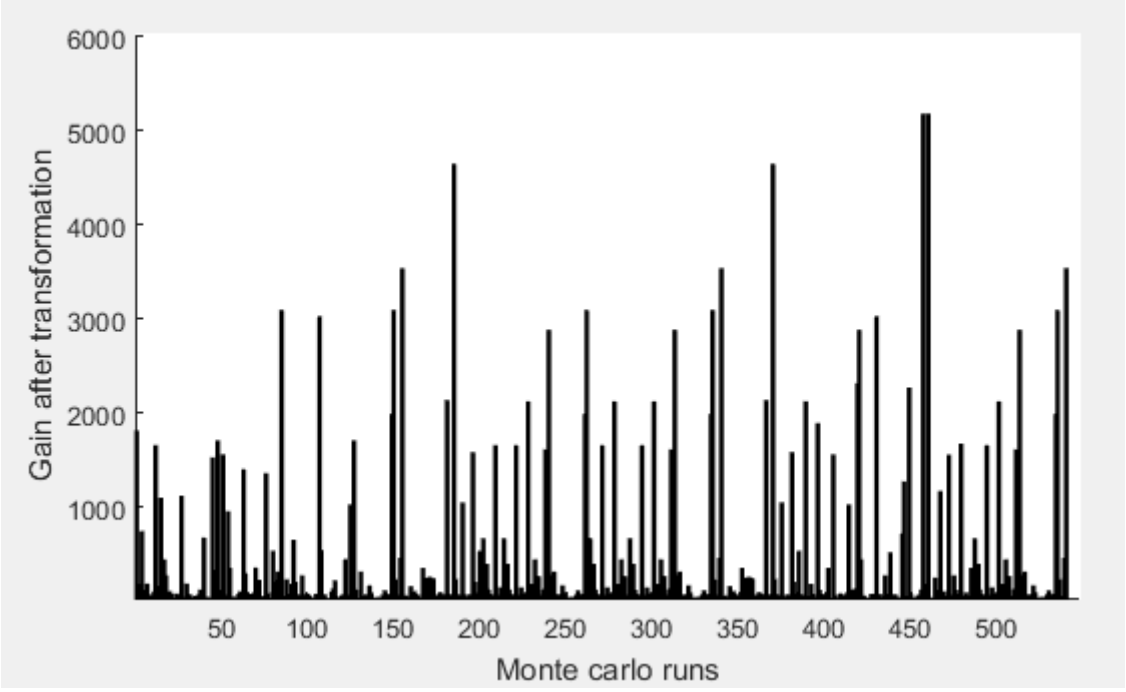

Figure 4.12 The gains at the eleventh harmonic given in order of run

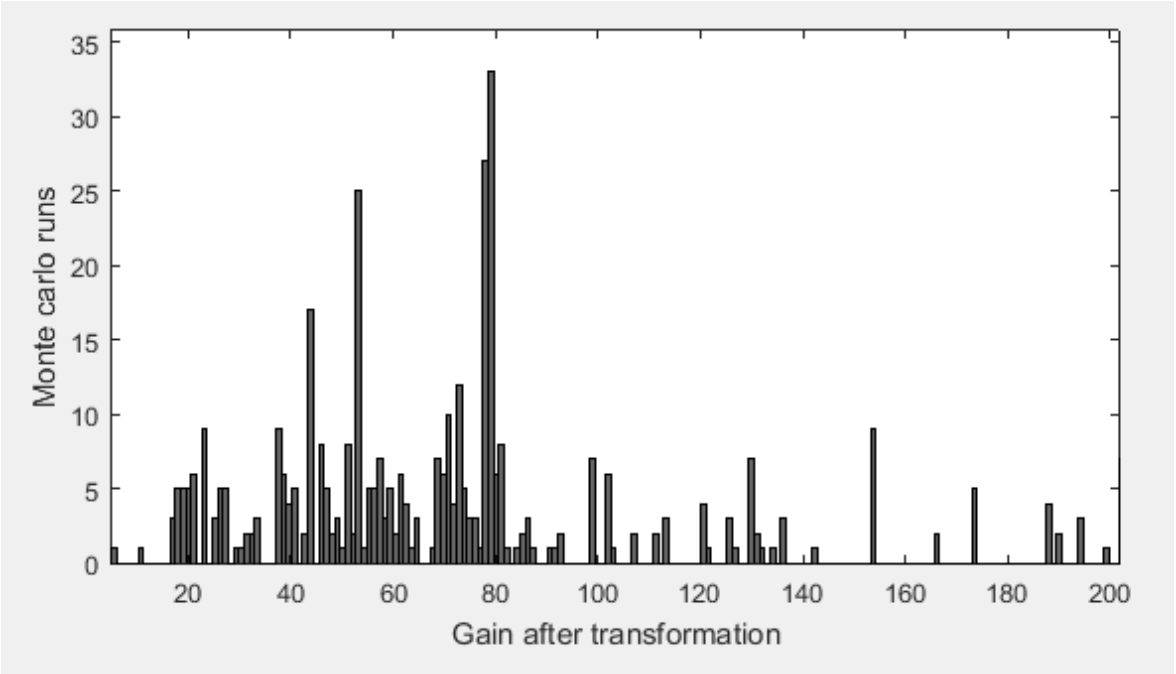

Figure 4.13 The frequency of the gains at the ninth harmonic

4.3 Result of Gaussian Process

The aim to employ the Gaussian process in this study is to obtain the original signal from the transformed signal. The Gaussian process was applied to the transformed signal in the time (sample) domain and the original signal in the time (sample) domain was tried to obtain.

Figure 4.14 shows the output of the Gaussian process. The red line in the figure is the original signal and the blue line is the predictive result which has been given by the Gaussian process. The grey zone gives the 95% confidence area. Only two cycles are presented in the figure to be see the result clearer and the rest of them is repeated itself.

The blue line (the output of the Gaussian process) was created by following the mean value of the posterior which is obtained as the output of the process. The 95% confidence region (the grey zone) was built by using the standard deviation of the posterior. The expectation was that the mean value would follow the original signal throughout the data in an ideal result or it would have some little deviation in the confidence region. However, the result shows that there are some parts of the output which are not in the confidence region. Nevertheless, the result gives the general pattern of the output signal.

All results including the Gaussian process (Figure 4.14) have been given in this chapter will be evaluated in the next chapter.

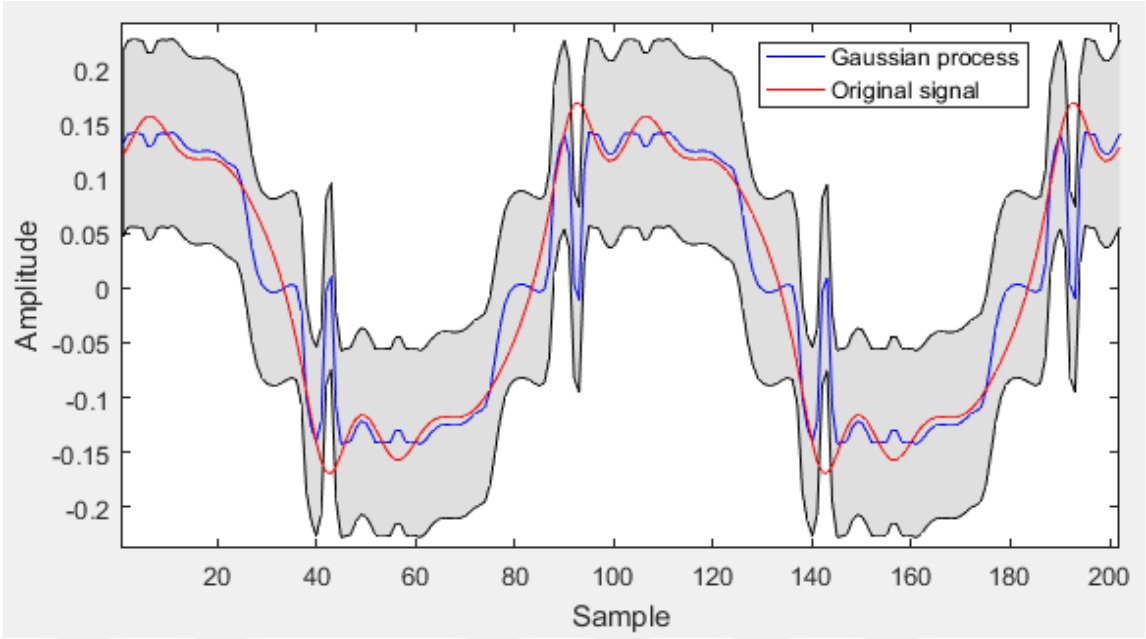

Figure 4.14 The result of the Gaussian process

5 DISCUSSION

5.1 Cancellation of the Superharmonics

The study in this paper offers SADE to eliminate superharmonics in the response of a nonlinear system. For this, a single degree-of-freedom Duffing's equation was employed to create a response which contains superharmonics. It was chosen as Duffing's equation as discussed in Section 3.1. Thus, the response had only the odd order harmonics. After getting the results, the next step was to cancel or reduce the harmonics except first one.

The study offered SADE to eliminate the superharmonics by using a transformation in the form of a polynomial expansion. The transformation was chosen as it is Equation 3.14 and it succeeded to remove the harmonics of the interest. However, it is possible to decide a different form of transformation which would probably work, as well. One of the crucial criteria must be the effect of the transformation on the computation time. In this study, the Monte Carlo simulation was also conducted so the algorithm was run 540 times to observe all scenarios as many as possible. Hence, the computation time had to be considered primarily. Otherwise, the completion of 540 runs of the algorithm would take massive time. The higher number of parameters of the transformation means that the algorithm has to deal with larger matrices and more operations. This costs more computation time. That is why the number of parameters of the transformation was chosen 6. It has provided a good output of the algorithm with a small number of parameters.

The number of generations defined as the stopping criterion is another factor to describe the computation time. It was chosen as 1000 generations. It would be taken further and it would end up with a low-cost value as a final outcome. Still, 1000 generations gave stable results in several tries so it was taken 1000 generations to keep the number of generations small.

To develop new mutation strategies and parameters adaptation are required a different and comprehensive study. Hence, the mutation strategies and parameters adaptation were set according to reference articles as stated in Section 2.2 and Section 3.2.3.

When the power spectral density of the result of SADE is looked in Figure 4.1, it can be seen that the superharmonics which exist in the original signal are removed after the transformation. In the original signal, the third, fifth and seventh harmonics appear as superharmonics. Even though these harmonics are removed in the transformed signal, the higher harmonics such as ninth, eleventh and thirteenth

ones emerge. This situation can be commented as that the energy at relatively lower harmonics transfers to the higher harmonics. Nevertheless, the elimination of all superharmonics in the original signal can be succeeded. As for the first harmonic, it is significant to preserve an obvious peak at the first harmonic of the power spectral density of the response. Although it seems slightly decreased in Figure 4.1, it is still obvious as most of the power spectral densities of the transformed signals are.

The decision of the cost or objective function guided the algorithm during the evolution. Therefore, the build of the cost function had a vital role. It was built as it is Equation 3.9 in this study. The scope of the cost function was chosen from 2 Hz to 8 Hz. It covered the superharmonics which appeared in the original signal in initial trials. The results showed that the algorithm with this cost function causes other superharmonics. One solution of this might be to increase the scope of the cost function in terms of the harmonics be covered. It would be enhanced to 10 Hz or further. The algorithm would seek the minimum cost value by including the superharmonics such as fifth and sixth ones which appear later on. In this case, even though the algorithm might be able to reduce the amplitudes of the superharmonics, it would be difficult to cancel the superharmonics. This might be improved by increasing the number of generations and parameters of the transformation but this needs a further investigation. The time limitation of this study did not allow to enhance the scope of the study. This can be the subject of a future study.

It can be stated that SADE has been achieved to find the parameters which remove the superharmonics in the original signal, as an overall assessment of this part of the study. However, Figure 4.1 shows only one result given by the algorithm and also, extra superharmonics which do not exist in the original signal appear in the transformed signal. Accordingly, the Monte Carlo simulation was conducted to evaluate possible results at the corresponding harmonics as a whole.

The SADE was run 540 times and at each time, the damping ratio ξ and the nonlinearity parameter a were randomly chosen. The results of the Monte Carlo simulation were given in Section 4.2. In Monte Carlo simulation, the gain, which is the ratio of the amplitude after transformation to the amplitude before the transformation, was considered. The figures which give the frequency of the gains at the corresponding harmonics provide a better view of the overall results. Figure 4.5, Figure 4.7 and Figure 4.9 display that most of the gains at the first, second and third superharmonics are under 0.2. This can be clearly understood from the density of the line between 0 and 0.2. This means that SADE has managed to cancel or reduce the superharmonics in the most cases.

The data in Figure 4.5, Figure 4.7 and Figure 4.9 can be expressed in Table 5.1. The table gives the percentages of the gains under specific values at the first, second and third superharmonics. The ideal condition can be accepted that all gains are under 0.2. Therefore, it can be said that all superharmonics in the original signal have been removed in the half of the total run. In the almost quarter of the runs, important reductions have been succeeded.

Superharmonics/Gains	0.2	0.4	0.6	0.8	1
First (%)	67.59	73.89	84.81	89.81	95.00
Second (%)	49.63	64.63	75.56	91.85	95.19
Third (%)	84.63	88.15	91.30	91.85	93.15

Table 5.1 The percentages of the gains at superhamonics frequencies

Owing to the random changes of the damping ratio and the nonlinearity parameter, the natural frequencies (the peaks in the power spectral density of the original signal) might differ from run to run. In some cases, the maximum points of the peaks might remain out of the range which was used to calculate the total amplitude to find the gain. Though this situation is not valid in the most cases, it causes a higher gain value than the real gain value. Therefore, it can be expected that the real percentages for especially small gains are slightly higher than the percentages presented in Table 5.1.

The same investigation was carried out for ninth and eleventh harmonics, as well. Figure 4.11 and Figure 4.13 show that the gains at these harmonics are higher than 1. This indicates the increase in the amplitudes. There are some extreme values of the gains in the figure but these can be explained as a result of the very small values at the harmonics in the original signal.

The approach in this paper to remove superharmonics by using SADE provides good results in the most cases. For the results which any removal or reduction was not provided, the solution might be to set a new parameter adaptation or increase the number of generations to stop the algorithm. As it was stated before, these all parameter adaptation and the number of generations were set for some initial trials which gave good results. As an overall, the performance of the method was fulfilling.

5.2 Gaussian Process

The next operation was to obtain the original signal from the transformed signal by using the Gaussian process. Basically, it can be said that the process can be applied

by matrix manipulations. Although it is a straightforward method to apply, the result obtained is not as satisfied as expected. The Gaussian process results in Figure 4.14 predicts the lower harmonics properly and thus, it gives the general pattern of the original signal. However, it does not reflect the high harmonics and that is why the original signal goes out of the confidence region at some points. For that reason, the result is required to improve. Nonetheless, the result of the Gaussian process provides the general pattern of the original signal.

5.3 Future Works

It is possible to widen the current study in terms of improving SADE to eliminate superharmonics and getting a better Gaussian process result. The removal of more superharmonics is obviously desired so one further study can be about the cancellation of the superharmonics which appear later on. This might be achieved with a larger number of parameters and generations so it seems to take larger time. Therefore, another study which focuses on the performance of the algorithm in terms of the computation time can be recommended. Finally, an investigation is required to improve the result of the Gaussian process.

Apart from the further investigations stated above, a conference paper from the study as it stands is projected to join International Conference on Noise and Vibration Engineering, ISMA2018 in Belgium. After a satisfying result has been obtained in the Gaussian process, a journal article is planned to write as another future work.

6 CONCLUSION

Nonlinear systems have different phenomena as distinct from the linear system. One of them is that superharmonics might occur in the response of a nonlinear system. The study offered a novel approach to deal with these superharmonics. A single degree-of-freedom Duffing's equation was used to create the response with superharmonics. After that, a transformation was sought to express the response in a new form which does not contain superharmonics. This appeared as a polynomial expansion with unknown coefficients. Self-Adaptive Differential Evolution was made use of to find these coefficients which provide superharmonics-free form.

To set the transformation, the cost function, and user-defined parameters, the algorithm was run several times by changing the settings. The algorithm managed to remove superharmonics in the original signal. However, some extra superharmonics which did not exist in the original signal appeared.

The Monte Carlo simulation was also made to test the robustness of the algorithm in terms of superharmonics removal. For this, the algorithm was run 540 times by randomly changing the damping ratio and the nonlinearity parameter in the equation of the motion. In that way, a set of data which included different cases was created. The mutation strategies and parameter settings were kept same as them in the initial trials. All results were presented and discussed.

The results showed that the approach in this paper provides removal or reduction of the superharmonics in the most cases. Thus, it can be said that one of the objectives of this paper has been achieved.

The other objective of the study was to obtain the original signal from the transformed signal. The Gaussian process was employed for this pattern recognition problem. The result obtained from the Gaussian process gave the general pattern of the original signal but it cannot be said that it managed to predict high harmonics. In some points, the predicted curve by the Gaussian process stays out of the confidence region. Thus, the second objective has been partly completed. It needs more improvements.

The importance of the study is to show that a nonlinear response with superharmonics can be expressed in a new form without superharmonics. Moreover, this can be done with a transformation and by using SADE which does not contain any complicated analyis when compared with solution of the nonlinear differential equations. This is a novel approach to the analysis of nonlinear systems with superharmonics.

One of the major limitations was the time restraint because the solution of the algorithm takes serious time due to the size of the matrices and the number of operations and generations. Also, because of the Monte Carlo simulation, the algorithm was run 540 times. After improvement of the Gaussian process result, one recommendation as a future work can be a study which focuses on increasing the number of the superharmonics removed. This will make possible the removal of the superharmonics which appear later in the transformed signal.

REFERENCES

[1] G. Kerschen, K. Worden, A. F. Vakakis, and J.-C. Golinval, “Mechanical Systems and Signal Processing Past, present and future of nonlinear system identification in structural dynamics,” *Mech. Syst. Signal Process.*, vol. 20, pp. 505–592, 2006.

[2] N. Dervilis, K. Worden, D. J. Wagg, and S. A. Neild, “Simplifying transformations for nonlinear systems: Part I, an optimisation-based variant of normal form analysis,” in *Conference Proceedings of the Society for Experimental Mechanics Series*, 2016.

[3] N. Dervilis, K. Worden, D. J. Wagg, and S. A. Neild, “Simplifying transformations for nonlinear systems: Part II, statistical analysis of harmonic cancellation,” in *Conference Proceedings of the Society for Experimental Mechanics Series*, 2016.

[4] K. Worden and G. R. Tomlinson, “Nonlinearity in experimental modal analysis,” *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.*, vol. 359, no. 1778, pp. 113–130, 2001.

[5] K. C. Park, “Methods for Vibration Analysis,” *Struct. Syst. Identif.*, pp. 1–24.

[6] K. Worden and G. R. Tomlinson, “Nonlinearity in Structural Dynamics,” p. 659, 2001.

[7] S. da Silva, S. Cogan, E. Foltete, F. Buffe, S. da Silva, S. Cogan, E. Foltete, F. Buffe, S. da Silva, S. Cogan, E. Foltete, and F. Buffe, “Metrics for nonlinear model updating in structural dynamics,” *J. Braz. Soc. Mech. Sci. \& Eng.*, vol. 31, no. 1, pp. 27–34, 2009.

[8] D. Wagg and S. Neild, *Nonlinear vibration with control: for flexible and adaptive structures*. Springer, 2010.

[9] K. Worden and G. R. Tomlinson, “Nonlinearity in experimental modal analysis,” *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.*, vol. 359, no. 1778, pp. 113–130, 2001.

[10] M. N. Hamdan and T. D. Burton, “On the Steady State Response and Stability of Non-Linear Oscillators Using Harmonic Balance,” *Journal of Sound and Vibration*, vol. 166, no. 2. pp. 255–266, 1993.

[11] K. Worden, “On Jump Frequencies in the Response of the Duffing Oscillator,” vol. 198, pp. 522–525, 1996.

[12] A. Carrella, “Passive vibration isolators with high-static-low-dynamic-stiffness,” 2008.

[13] R. E. Mickens, “Comments on the method of harmonic balance,” *J. Sound Vib.*, vol. 94, no. 3, pp. 456–460, 1984.

[14] U. Filobello-Nino, H. Vazquez-Leal, K. Boubaker, Y. Khan, A. Perez-Sesma, A. Sarminto-Reyes, V. M. Jimenez-Fernandez, A. Diaz-Sanchez, A. Herrera-May, J. Sanchez-Orea, and K. Pereyra-Castro, "Perturbation Method as a Powerful Tool to Solve Highly Nonlinear Problems: The Case of Gelfand's Equation." Asian Journal of Mathematics and Statics, 2013.

[15] F. Lakrad and M. Belhaq, "Periodic Solutions of Strongly Non-Linear Oscillators By the Multiple Scales Method," *J. Sound Vib.*, vol. 258, no. 4, pp. 677–700, 2002.

[16] S. A. Neild and D. J. Wagg, "Applying the method of normal forms to second-order nonlinear vibration problems," *Proc. R. Soc.*, vol. 467, pp. 1141–1163, 2011.

[17] L. Jezequel and C. Lamarque, "Analysis of non-linear dynamical systems by the normal form theory," *J. Sound Vib.*, vol. 149, no. 3, pp. 429–459, 1991.

[18] C. Touzé, "Normal form theory and nonlinear normal modes: Theoretical settings and applications," 2012.

[19] S. A. Neild, A. R. Champneys, D. J. Wagg, T. L. Hill, and A. Cammarano, "The use of normal forms for analysing nonlinear mechanical vibrations," *Philos. Trans. R. Soc. A*, vol. 373, no. 2051, 2015.

[20] C. Touzé and O. Thomas, "Determination of Non-Linear Normal Modes for Conservative Systems."

[21] A. Vakakis, "Non-linear normal modes (nnms) and their applications in vibration theory: an overview," *Mech. Syst. Signal Process.*, vol. 11, no. 1, pp. 3–22, 1997.

[22] G. Kerschen, M. Peeters, J. C. Golinval, and A. F. Vakakis, "Nonlinear normal modes, Part I: A useful framework for the structural dynamicist," *Mech. Syst. Signal Process.*, 2009.

[23] R. M. Rosenberg, "On Nonlinear Vibrations of Systems with Many Degrees of Freedom," *Adv. Appl. Mech.*, vol. 9, pp. 155–242, 1966.

[24] K. Worden and P. L. Green, "A machine learning approach to nonlinear modal analysis," *Mech. Syst. Signal Process.*, vol. 84, pp. 34–53, 2016.

[25] C. Touzé, "A normal form approach for non-linear normal modes," *ENSTA Tech. Rep.*, 2002.

[26] C. Pierre, D. Jiang, and S. Shaw, "Nonlinear normal modes and their application in structural dynamics," *Math. Probl. Eng.*, 2006.

[27] W. Shaw, S and C. Pierre, "normal modes for nonlinear vibratory systems," *J. Sound Vib.*, vol. 164, no. 1, pp. 85–124, 1993.

[28] W. Shaw, S and C. Pierre, "normal modes of vibration for nonlinear continuous systems," *J. Sound Vib.*, vol. 169, no. 3, pp. 319–347, 1994.

[29] R. Storn and K. Price, “Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces,” *J. Glob. Optim.*, vol. 11, pp. 341–359, 1997.

[30] R. Mallipeddi and P. N. Suganthan, “Differential evolution algorithm with ensemble of parameters and mutation and crossover strategies,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6466 LNCS, pp. 71–78, 2010.

[31] S. Das and P. N. Suganthan, “Differential evolution: A survey of the state-of-the-art,” *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 4–31, 2011.

[32] K. Worden and G. Manson, “On the identification of hysteretic systems. Part I: Fitness landscapes and evolutionary identification,” *Mech. Syst. Signal Process.*, vol. 29, pp. 201–212, 2012.

[33] T. Rogalsky, R. W. Derksen, and S. Kocabiyik, “Differential Evolution in Aerodynamic Optimization,” *Proc. 46th Annu. Conf. Can. Aeronaut. Sp. Inst.*, pp. 29–36, 1999.

[34] J. Ilonen, J. K. Kamarainen, and J. Lampinen, “Differential evolution training algorithm for feed-forward neural networks,” *Neural Process. Lett.*, vol. 17, no. 1, pp. 93–105, 2003.

[35] R. Joshi and A. C. Sanderson, “Minimal representation multisensor fusion using differential evolution,” *IEEE Trans. Syst. Man, Cybern. - Part A Syst. Humans*, vol. 29, no. 1, pp. 63–76, 1999.

[36] A. K. Qin and P. N. Suganthan, “Self-adaptive differential evolution algorithm for numerical optimization,” *IEEE Congr. Evol. Comput.*, pp. 1785–1791, 2005.

[37] A. K. Qin, V. L. Huang, and P. N. Suganthan, “Differential evolution algorithm with strategy adaptation for global numerical optimization,” *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 398–417, 2009.

[38] M. G. H. Omran, A. Salman, and A. P. Engelbrecht, “Self-adaptive differential evolution,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3801 LNAI, pp. 192–199, 2005.

[39] J. Zhang, S. Member, and A. C. Sanderson, “JADE : Adaptive Differential Evolution with Optional External Archive,” vol. 13, no. 5, pp. 945–958, 2009.

[40] F. Neri and V. Tirronen, *Recent advances in differential evolution: A survey and experimental analysis*, vol. 33, no. 1–2. 2010.

[41] K. Zielinski, P. Weitkemper, R. Laur, and K.-D. Kammeyer, “Parameter Study for Differential Evolution Using a Power Allocation Problem Including Interference Cancellation,” *2006 IEEE Int. Conf. Evol. Comput.*, no. 2, pp. 1857–1864, 2006.

[42] K. K. Mandal and N. Chakraborty, “Parameter study of differential evolution based optimal scheduling of hydrothermal systems,” *J. Hydro-Environment Res.*, vol. 7, no. 1, pp. 72–80, 2013.

[43] V. Gonuguntla, R. Mallipeddi, and K. C. Veluvolu, “Differential evolution with population and strategy parameter adaptation,” *Math. Probl. Eng.*, vol. 2015, 2015.

[44] F. Peñuñuri, C. Cab, O. Carvente, M. A. Zambrano-Arjona, and J. A. Tapia, “A study of the Classical Differential Evolution control parameters,” *Swarm Evol. Comput.*, vol. 26, pp. 86–96, 2016.

[45] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning.*, vol. 14, no. 2. 2004.

[46] C. K. I. Williams, “Prediction with Gaussian Processes: From Linear Regression to Linear Prediction and Beyond,” *Learn. Graph. Model.*, no. 89, pp. 599–621, 1998.

[47] D. J. C. Mackay, *Introduction to Gaussian Processes*, Volume 168. Berlin: Springer, 1998.

[48] J. Quiñonero-candela, C. E. Rasmussen, and R. Herbrich, “A unifying view of sparse approximate Gaussian process regression,” *J. Mach. Learn. Res.*, vol. 6, pp. 1935–1959, 2005.

[49] A. J. Smola and P. Bartlett, “Sparse Greedy Gaussian Process Regression,” *Adv. Neural Inf. Process. Syst. 13*, vol. 13, pp. 619–625, 2001.

[50] N. D. Lawrance, “Introduction to Gaussian processes,” Sydney, 2015. [Online]. Available: http://rp-www.cs.usyd.edu.au/~mlss/content/mlss2015-lawrence.pdf [Accessed: 24 June 2017].

[51] MathWorks, “ode45,” *MathWorks Documentation*, 2017. [Online]. Available: https://uk.mathworks.com/help/matlab/ref/ode45.html [Accessed: 19 February 2017].

[52] C. E. Rasmussen and H. Nickisch, “Documentation for GMPL Matlab Code version 4.0,” 2006. [Online]. Available: http://www.gaussianprocess.org/gpml/code/matlab/doc/ [Accessed: 1 July 2017]