State switching in multi-stable systems: control and optimisation.

LIU, B.

2020

State Switching in Multistable Systems: Control and Optimisation

BOYING LIU

A thesis submitted in partial fulfilment of the

requirement of the

Robert Gordon University

for the degree of Doctor of Philosophy

(School of Engineering)

June 2020

State Switching in Multistable Systems: Control and Optimisation

BOYING LIU

Supervisor Team:

Principal Supervisor: Dr Wai-Keung Fung

Second Supervisor: Yang Liu

School of Engineering, Robert Gordon University,
The Sir Ian Wood Building, Riverside East,
Garthdee Road, AB10 7GJ,
Aberdeen,
United Kingdom.

DECLARATION

I declare that this report, except where otherwise state, is based on my own work. To the best of my knowledge and belief, this report contains no material previously published or written by another person, except where due reference has been made.

Acknowledgements

I am grateful to my supervisor, Dr Wai-Keung Fung, for his sincere and selfless help. He gave me lots of help and advice through all the stages of my PhD study. I am deeply grateful of his constant encouragement and guidance during the writing of this thesis. He spent much time on read each draft of the thesis and gave me inspiring suggestions. Without his help, I cannot finish this thesis.

I would like to thank my supervisor, Dr Yang Liu for his guidance and help. He provided me the opportunity of the PhD study. Thank him for his invaluable suggestions on the academic studies and writing.

I would also like to thanks Carnegie Trust for their funding. They also provide me the chance to network with other PhD students. I get lots of invaluable experience and suggestions on the PhD study from them.

I want to thank all the staff in School of Engineering and library at Robert Gordon University. They gave me lots of invaluable suggestions on how to write a PhD thesis. Moreover, they provide a comfortable environment for my PhD study.

Thank my family and friends for their help, support and encouragement. Thank them for listening to me when I encounter difficulties and helping me out of the difficulties.

Abstract

This thesis studies state switching in multistable systems so that they can switch from inefficient operating states to efficient one for performance enhancement in real life engineering systems. Multistable systems have more than one stable state under a set of parameters and the process of switching from an undesired state to a desired state is achieved by the proposed PD-like controller. It exploits the difference of the displacement and velocity between the undesired and the desired stable conditions for feedback in state switching. Three test systems are used for investigating the performance of PD-like controller namely the Duffing oscillator, which is a typical smooth multistable system; the non-smooth soft impact oscillator; and the soft impact oscillator with a drift.

A randomised triangular subdivision algorithm is proposed to reconstruct the basins of attraction of the target multistable systems in order to identify the desired state for switching. Due to the limited capacity of physical actuators, behaviours of the constrained PD-like controller are investigated using extensive simulation on the test systems. Moreover, optimisation of the controller based on multiple performance objectives can further improve system performance. Two performance objectives namely maximum peak of control input and switching duration will be adopted in optimising the proposed PD-like controller. The first objective will be minimised in order to avoid output limit and reduce energy consumption in the actuator while the second objective will be minimised to shorten the time required for state switching. These two performance objectives will be considered independently in performance optimisation using particle swarm optimisation (PSO). Since these two objectives are conflicting with each other, both objectives will be minimised simultaneously in multiobjective optimisation of the performance of the PD-like controller using Non-dominated Sorting Genetic Algorithms-II (NSGA-II). A trade-off in performance enhancement is achieved through selecting control parameters from the Pareto optimal set.

Contents

DECLARATION i

Acknowledgements ii

Abstract iii

List of Figures vii

List of Tables xii

1. Introduction 1

1.1 Project Overview 1

1.2 Aim and Objectives 7

1.2.1 Aim 7

1.2.2 Objectives 7

1.3 Three Test Systems 7

1.3.1 Duffing Oscillator 8

1.3.2 Soft impact oscillator 8

1.3.3 Soft impact oscillator with a drift 10

1.4 Contributions of the thesis 12

Published Work 13

1.5. Thesis Organisation 13

Chapter 2 Reconstruction of Basins of Attraction 14

2.1 Introduction 14

2.2 Traditional Method to Identify Basins of Attraction 14

2.3 The Proposed Method 15

2.3.1 Lawson Flip Algorithms 16

2.3.2 Incremental Construction 17

2.4 Randomised Triangular Subdivision 18

2.5 Numerical Simulation 22

2.5.1 Duffing Oscillator with Two Coexisting Stable States 22

2.5.2 Soft impact oscillator with Two Coexist Stable State 40

2.5.3 Soft impact oscillator with Three Stable State 49

2.6 Summary 57

Chapter 3 Multistable State Switching using PD-like Control 58

3.1 Introduction 58

3.1.1 Multistable State Switching Control Algorithms 58

3.1.2 The proposed PD-like Control 60

3.2 Numerical Simulation 61

3.2.1 Duffing Oscillator 62

3.2.2 Soft impact oscillator 66

3.2.3 Soft impact oscillator with a Drift 70

3.3 Summary 73
Chapter 4 Constrained PD-like Control 74
4.1 Introduction........ 74
4.1.1 Constrained Control of Multistable Systems 74
4.1.2 Constrained PD-like Control 74
4.2 Numerical Simulation 75
4.2.1 Duffing Oscillator 75
4.2.2 Soft impact oscillator 79
4.2.3 Soft impact oscillator with a Drift 81
4.3 Summary 83
Chapter 5: Sensitivity Analysis of Parameters of PD-like Controller 84
5.1 Numerical Simulation 84
5.1.1 Duffing Oscillator 84
5.1.2 Soft impact oscillator 90
5.1.3 Soft impact oscillator with a Drift 99
5.2 Summary 101
Chapter 6 Single Objective Optimisation of the PD-like Controller 102
6.1 Introduction........102
6.1.1 Single Objective Optimisation........103
6.1.2 Single Objective Optimisation Algorithms 103
6.2 Single Objectives Optimisation for PD-like Controller 106
6.2.1 Duffing Oscillator 108
6.2.2 Soft impact oscillator 113
6.2.3 Soft impact oscillator with a Drift 117
6.3 Summary 122
Chapter 7 Multi-objective Optimisation of PD-like Controller Parameters.123
7.1 Introduction........123
7.1.1 Multi-objective Optimisation........124
7.1.2 Pareto Optimal Solutions 124
7.1.3 Multi-objectives Optimisation Algorithms........125
7.2 Multi-objective PD-like Controller Optimisation 133
7.2.1 Duffing Oscillator 134
7.2.2 Soft impact oscillator 139
7.2.3 Soft impact oscillator with a Drift 145
7.3 Summary 149
Chapter 8 Conclusion and Future Work........150
8.1 Conclusion........150
8.2 Future work........152

Reference 155
Appendix A 173

List of Figures

Figure 1.1 Energy harvesting system [7] 2
Figure 1.2 Vibro-impact capsule system [8] 3
Figure 1.3 Percussive drilling system [9] 4
Figure 1.4 The complex structure of the basins of attraction with three coexist state 5
Figure 1.5 The basins of attraction of a soft impact oscillator. Small windows show the amplitudes of output responses in the two coexisting states. 6
Figure 1.6 Schematics of Duffing oscillator [21] 8
Figure 1.7 Schematics of soft impact oscillator [23] 9
Figure 1.8 Schematics of soft impact oscillator with a drift [28] 11
Figure 2.1 (a) Delaunay triangulation (b) Non- Delaunay triangulation 17
Figure 2.2 Step 2 of incremental construction 18
Figure 2.3 Step 3 of incremental construction 18
Figure 2.4 Flowchart of the randomised triangular subdivision. Red dashed block shows step 2, purple block shows step 3 and 4, blue dashed block shows step 5 and orange dashed block shows step 6. 21
Figure 2.5 Different conditions of a triangle in Delaunay Triangulation of Initial Condition Space. 22
Figure 2.6 Random samples of the initial conditions 24
Figure 2.7 Triangulating the convex by using Delaunay Triangulation based on these 50 initial conditions 24
Figure 2.8 The triangles with vertices that are all in the same state. The colour of the point shows the vertex located in which state. Needs rewriting for clarity 25
Figure 2.9 Triangles which have at least one vertex that is in a different state compared with the others. The colour of the point shows the vertex located in which state. Needs rewriting for clarity 25
Figure 2.10 Distribution of the augmented set of initial conditions. The colour of the point shows the initial condition converge to which state..... 26
Figure 2.11 Basins of attraction estimated using the random triangular subdivision with 1224 initial conditions. 26
Figure 2.12 High Resolution Basins of attraction of Duffing Oscillator (1001*1001) 27
Figure 2.13 Basins of attraction estimated by the brute force algorithm with resolution 35*35 (1225 initial conditions) 28
Figure 2.14 Triangulating the convex by using Delaunay Triangulation base on these 50 initial conditions 31
Figure 2.15 Random sample of the initial conditions. The colour of the points shows the initial condition converge to which state............................ 32
Figure 2.16 Triangles in the convex after first subdivision..................... 32
Figure 2.17 Distribution of the 102 initial conditions in the second generation. The colour of the points shows the initial condition converge to which state. 33
Figure 2.18 Basins of attraction estimated using random triangular subdivision with 102 initial conditions in the second generation 33
Figure 2.19 Basins of attraction estimated using the brute force algorithm with resolution 10*10 (100 initial conditions) 34
Figure 2.20 Triangles in the convex after second subdivision................. 34

Figure 2.21 Distribution of the 235 initial conditions in the third generation. The colour of the points shows the initial condition converge to which state. 35
Figure 2.22 Basins of attraction estimated using random triangular subdivision with 235 initial conditions in the third generation 35
Figure 2.23 Basins of attraction estimated using the brute force algorithm with resolution 15*15 (225 initial conditions) 36
Figure 2.24 Triangles in the convex after third subdivision.................... 36
Figure 2.25 Distribution of the 431 initial conditions in the fourth generation. The colour of the points shows the initial condition converge to which state.. 37
Figure 2.26 Basins of attraction estimated using the random triangular subdivision with 431 initial conditions in the fourth generation 37
Figure 2.27 Basins of attraction estimated using the brute force algorithm with resolution 21*21 (441 initial conditions) 38
Figure 2.28 Triangles in the convex after fourth subdivision.................. 38
Figure 2.29 Distribution of the 480 initial conditions in the fifth generation. The colour of the points shows the initial condition converge to which state. 39
Figure 2.30 Basins of attraction estimated using random triangular subdivision with 480 initial conditions in the fifth generation 39
Figure 2.31 Basins of attraction estimated using the brute force algorithm with resolution 22*22 (484 initial conditions) 40
Figure 2.32 Basins of attraction of the soft impact oscillator with the system parameters: $\xi = 0.01, \beta = 29, g = 1.26,\ \Gamma = 1.0385$ and $\omega = 0.686$ 43
Figure 2.33 Random sample of the initial conditions. The colour of the points shows the initial condition converge to which state............................. 44
Figure 2.34 Triangulating the convex by using Delaunay Triangulation base on these 50 initial conditions 44
Figure 2.35 Basins of attraction estimated by the randomised triangular subdivision method with 122 initial conditions 45
Figure 2.36 Basins of attraction estimated by the brute force algorithm with resolution 11*11 (121 initial conditions)... 45
Figure 2.37 Basins of attraction estimated by the randomised triangular subdivision with 298 initial conditions 46
Figure 2.38 Basins of attraction estimated by the brute force algorithm with resolution 17*17 (289 initial conditions)... 46
Figure 2.39 Basins of attraction estimated by the randomised triangular subdivision method with 618 initial conditions 47
Figure 2.40 Basins of attraction estimated by the brute force algorithm with resolution 25*25 (625 initial conditions)... 47
Figure 2.41 Basins of attraction estimated by the randomised triangular subdivision method with 893 initial conditions 48
Figure 2.42 Basins of attraction estimated by the brute force algorithm with resolution 30*30 (900 initial conditions)... 48
Figure 2.43 The basins of attraction of the soft impact oscillator with the system parameters: $\xi = 0.01, \beta = 29, g = 1.26,\ \Gamma = 0.4547$ and $\omega = 0.806$ 51
Figure 2.44 Random sample of the initial conditions. The colour of the points shows the initial condition coverage to which state............................. 52
Figure 2.45 Triangulating the convex by using Delaunay Triangulation base on these 50 initial conditions 52

Figure 2.46 Basins of attraction estimated by the random triangular subdivision with 138 initial conditions 53
Figure 2.47 Basins of attraction estimated by the brute force algorithm with resolution 12*12 (144 initial conditions)... 53
Figure 2.48 Basins of attraction estimated by the random triangular subdivision with 381 initial conditions 54
Figure 2.49 Basins of attraction estimated by the brute force algorithm with resolution 20*20 (400 initial conditions)... 54
Figure 2.50 Basins of attraction estimated by the random triangular subdivision with 860 initial conditions 55
Figure 2.51 Basins of attraction estimated by the brute force algorithm with resolution 29*29 (841 initial conditions)... 55
Figure 2.52 Basins of attraction estimated by the random triangular subdivision with 1493 initial conditions... 56
Figure 2.45 Basins of attraction estimated by the brute force algorithm with resolution 39*39 (1521 initial conditions) 56
Figure 3.1 Basin of attraction of the Duffing oscillator with system parameters: $k = 0.9, \Gamma = 1.9$ and $\omega = 1.2$ 63
Figure 3.2 State trajectory of the co-existing attractor 1 on the phase plane 63
Figure 3.3 State trajectory of the co-existing attractor 2 on the phase plane 64
Figure 3.4 Response of displacement as a function in time 64
Figure 3.5 Response of velocity as a function in time.......................... 65
Figure 3.6 The applied control force as a function in time..................... 65
Figure 3.7 The basins of attraction of the soft impact oscillator with the system parameters: $\xi = 0.01, \beta = 29, g = 1.26,\ \Gamma = 1.0385$ and $\omega = 0.686$ 67
Figure 3.8 State trajectory of the co-existing attractor 1 on the phase plane 67
Figure 3.9 State trajectory of the co-existing attractor 2 on the phase plane 68
Figure 3.10 Displacement of the mass as a function in time.................. 68
Figure 3.11 Velocity of the mass as a function in time 69
Figure 3.12 The applied control force as a function in time 69
Figure 3.13 The basins of attraction of the soft impact oscillator with a drift with the system parameters: $a = 0.3$, $b = 0.2$, $\omega = 1$, $g = 0.02$, $\xi = 0.1$ and $\varphi = \pi/2$... 71
Figure 3.14 State trajectory of the co-existing attractor 1 on the phase plane.. 71
Figure 3.15 State trajectory of the co-existing attractor 1 on the phase plane.. 72
Figure 3.16 Displacement of mass (blue line) and bottom plate (red line) as a function in time 72
Figure 3.17 The applied control force as a function of time 72
Figure 4.1 Control inputs to Duffing oscillator using constrained PD-like controller (black line) and unconstrained PD-like controller................... 76
Figure 4.2 Trajectory of the system state when the constrained PD-like controller was applied 77
Figure 4.3 The system is driven from the current state $s1$ to the desired state $s2$ directly by using the unconstrained PD-like controller............... 77

Figure 4.4 Trajectory of the system state when the unconstrained PD-like controller was applied 78
Figure 4.5 The system is driven from current state $s1$ to the desired state $s2$ through a transient state $s1'$ by using the constrained PD-like controller. . 78
Figure 4.6 Control inputs to soft impact oscillator using constrained PD-like controller (black line) and unconstrained PD-like controller (blue line) 80
Figure 4.7 Trajectory of the system state when the constrained PD-like controller was applied 80
Figure 4.8 Trajectory of the system state when the PD-like controller was applied 81
Figure 4.9 Control inputs to soft impact oscillator with a drift using constrained PD-like controller (black line) and unconstrained PD-like controller. 82
Figure 4.10 Trajectory of the system state when the constrained PD-like controller was applied 82
Figure 4.11 Trajectory of the system state when the constrained PD-like controller was applied 83
Figure 5.1 The applied control force as a function in time with the control parameters $kp = 1$, $kd = 1$ and the controller is switched on at $t = 1$ second when PD-like controller applied on the Duffing oscillator....................... 85
Figure 5.2 Displacement of the mass as a function in time with the control parameters $kp = 1$, $kd = 1$ and the controller is switched on at $t = 1$ second when PD-like controller applied on the Duffing oscillator....................... 85
Figure 5.3 The applied control force as a function in time with the control parameters $kp = 1$, $kd = 1$ and the controller is switched on at $t = 1.8$ seconds when PD-like controller applied on the Duffing oscillator........... 86
Figure 5.4 Displacement of the mass as a function in time with the control parameters $kp = 1$, $kd = 1$ and the controller is switched on at $t = 1.8$ seconds when PD-like controller applied on the Duffing oscillator........... 86
Figure 5.5 The simulation result of applying the PD-like controller on Duffing oscillator with different kp and kd. The colour bars show the magnitude of the maximum peak of control force 88
Figure 5.6 The simulation result of applying the PD-like controller on Duffing oscillator with different kp and kd. The colour bars show the magnitude of switching duration. 88
Figure 5.7 The maximum peak of control force when the PD-like controller is applied on Duffing oscillator with different switch on time..................... 89
Figure 5.8 The switching duration when the PD-like controller is applied on the Duffing oscillator with different switching time. 89
Figure 5.9 The simulation result of apply the PD controller on soft impact oscillator with different kp and kd. The colour bars show the magnitude of the maximum peak of control force 91
Figure 5.10 The simulation result of applying the PD-like controller on soft impact oscillator with different kp and kd. The colour bars show the magnitude of switching duration. 91
Figure 5.11 The maximum peak of control force when the PD-like controller is applied on soft impact oscillator with different switching time. 92
Figure 5.12 The switching duration when the PD-like controller is applied on soft impact oscillator with different switching time. 92
Figure 5.13 The applied control force as a function in time with the control parameters $kp = 3$, $kd = 3$ and the controller is switched on at $t = 3.05$ seconds when PD-like controller applied on the soft impact oscillator 94

Figure 5.14 Displacement of the mass as a function in time with the control parameters $kp = 3$, $kd = 3$ and the controller is switched on at $t = 3.05$ seconds when PD-like controller applied on the soft impact oscillator 94
Figure 5.15 The applied control force as a function in time with the control parameters $kp = 7.8$, $kd = 4.6$ and the controller is switched on at $t = 3.05$ seconds when PD-like controller applied on the soft impact oscillator 95
Figure 5.16 Displacement of the mass as a function in time with the control parameters $kp = 7.8$, $kd = 4.6$ and the controller is switched on at $t = 3.05$ seconds when PD-like controller applied on the soft impact oscillator 95
Figure 5.17 The applied control force as a function in time with the control parameters $kp = 9$, $kd = 8.8$ and the controller is switched on at $t = 3.05$ seconds when PD-like controller applied on the soft impact oscillator 96
Figure 5.18 Displacement of the mass as a function in time with the control parameters $kp = 9$, $kd = 8.8$ and the controller is switched on at $t = 3.05$ seconds when PD-like controller applied on the soft impact oscillator 96
Figure 5.19 The applied control force as a function in time with the control parameters $kp = 1$, $kd = 1$ and the controller is switched on at $t = 2.11$ seconds when PD-like controller applied on the soft impact oscillator 97
Figure 5.20 Displacement of the mass as a function in time with the control parameters $kp = 1$, $kd = 1$ and the controller is switched on at $t = 2.11$ seconds when PD-like controller applied on the soft impact oscillator 97
Figure 5.21 The applied control force as a function in time with the control parameters $kp = 1$, $kd = 1$ and the controller is switched on at $t = 4.85$ seconds when PD-like controller applied on the soft impact oscillator 98
Figure 5.22 Displacement of the mass as a function in time with the control parameters $kp = 1$, $kd = 1$ and the controller is switched on at $t = 4.85$ seconds when PD-like controller applied on the soft impact oscillator 98
Figure 5.23 The simulation result of apply the PD-like controller on soft impact oscillator with a drift with different kp and kd. The colour bars show the magnitude of the maximum peak of control force 99
Figure 5.24 The simulation result when the PD-like controller is applied on soft impact oscillator with a drift with different kp and kd. The colour bars show the magnitude of switching duration. 100
Figure 5.25 The maximum peak of control force when the PD-like controller is applied on soft impact oscillator with a drift with different switching time. 100
Figure 5.26 The switching duration of when PD-like controller is applied on soft impact oscillator with a drift with different switching time..............101
Figure 6.1 Fitness value (maximum peak of external control force) vs. iteration in a typical PSO trial. 109
Figure 6.2 Fitness value (switching duration) vs iterations in a typical trial. 111
Figure 6.3 Control input to the Duffing oscillator using the PD-like control law with the optimised control parameter $kp = 0.0984$, $kd = 0.1121$ and the controller switches on $t = 1.61$ seconds 112
Figure 6.4 Control input to the Duffing oscillator using the PD-like control law with the optimised control parameter $kp = 10$, $kd = 8.2258$ and the controller switches on at $t = 4.71$ seconds...112
Figure 6.5 Fitness value (maximum peak of external control force) vs. iteration in a typical trial...114
Figure 6.6 Fitness value (switching duration) vs. iterations in a typical trial. 115

Figure 6.7 Control input to the soft impact oscillator using the PD-like control law with the optimised control parameter $kp = 0.0003$, $kd = 0.4677$ and the controller switches on at $t = 4.44$ seconds 116
Figure 6.8 Control input to the soft impact oscillator using the PD-like control law with the optimised control parameter $kp = 7.0967$, $kd = 9.9955$ and the controller switches on at $t = 3.41$ seconds 117
Figure 6.9 Fitness value (maximum peak of external control force) vs. iterations in a typical trial. 118
Figure 6.10 Fitness value (switching duration) vs. iterations in a typical trial. 120
Figure 6.11 Control input to the soft impact oscillator with a drift using the PD-like control law with the optimised control parameter $kp = 0.0434$, $kd = 0.0714$ and the controller switches on at $t = 5.47$ seconds 121
Figure 6.12 Control input to the soft impact oscillator with a drift using the PD-like control law with the optimised control parameter $kp = 4.0264$, $kd = 9.4060$ and the controller switches on at $t = 4.36$ seconds 121
Figure 7.1 Pareto fronts for two objectives optimisation problems.......... 125
Figure 7.2 Flow chart of GA 126
Figure 7.3 Flow chart of NSGA. The dashed block shows the nondominated sorting procedure in NSGA which is different from the selection procedure in GA. 128
Figure 7.4 Pseudo code of nondominated sorting [143]....................... 129
Figure 7.5 Pseudo code of crowding distance [143] 130
Figure 7.6 Crowding distance calculation 130
Figure 7.7 Pseudo code of NSGA-II [143] 131
Figure 7.8 Sorting Processes of Individuals in NSGA-II........................ 132
Figure 7.9 Pareto Front of Duffing oscillator with system parameters: $k = 0.9, \Gamma = 1.9$ and $\omega = 1.2$ 135
Figure 7.10 The corresponding Pareto Optimal set of Pareto Front, (a) kp and kd, (b) kp and the switching time, (c) kd and the switching time 137
Figure 7.11 The Pareto optimal performance of the PD-like controller with the three chosen optimal solutions in the two objectives (a) maximum peak of external control force, (b) switching duration................................ 137
Figure 7.12 Control input u(t) to the Duffing oscillator using PD-like control law with Pareto optimal control parameters. $kp = 2.15$, $kd = 1.53$ and switching on time $t = 1.7522$ seconds.. 138
Figure 7.13 Pareto Front of soft impact oscillator with system parameters: $\xi = 0.01$, $\beta = 29$, $g = 1.26$, $\Gamma = 1.9$, and $\omega = 0.686$ 140
Figure 7.14 The corresponding Pareto Optimal set of Pareto Front for PD-like controller parameters by multi-objective optimisation, (a) kp andkd, (b) kp and the switching time, (c) kd and the switching time..................... 141
Figure 7.15 The Pareto optimal performance of the PD-like controller with the three chosen optimal solutions in the two objectives (a) maximum peak of external control force, (b) switching duration................................. 142
Figure 7.16 (a) Control input u(t) (b) response of displacement (c) state trajectory of the soft impact oscillator with Pareto optimal control parameters: $kp = 2.51$, $kd = 3.11$ and switching on time $t = 7.61$ seconds .. 144
Figure 7.17 Pareto Front of the multi-objective optimisation results in PD-like controller applied on soft impact oscillator with a drift with system parameters: a=0.3, b=0.2, ω=1, g=0.02, ξ=0.1, and φ=π/2 146
Figure 7.18 The corresponding Pareto Optimal set of Pareto Front, (a) kp andkd, (b) kp and the switching time, (c) kd and the switching time 147

Figure 7.19 The Pareto optimal performance of the PD-like controller with the three chosen optimal solutions in the two objectives (a) maximum peak of external control force, (b)..148
Figure 7.20 Control input u(t) to the soft impact oscillator with a drift using PD-like control law with Pareto optimal control parameters $kp = 7.4507$, $kd = 4.8411$ and switching on time $t = 4.0376$ second 148

List of Tables

Table 2.1 Similarity and time costs of the basins of attraction estimated by the random triangular subdivision 31
Table 2.2 Similarity and time costs of the basins of attraction estimated by the brute force algorithm 31
Table 2.3 Similarity and time costs of the basins of attraction estimated by the randomised triangular subdivision method.................................... 43
Table 2.4 Similarity and time costs of the basins of attraction estimated by the brute force algorithm 43
Table 2.5 Similarity and the time costs of the basins of attraction estimated by the random triangular subdivision 51
Table 2.6 Similarity and the time costs of the basins of attraction estimated by the brute force algorithm.. 51
Table 6.1 Optimised results of 10 trials on minimizing maximum peak of the external control force 109
Table 6.2 Optimised results of 10 trials on minimizing switching duration110
Table 6.3 Optimised results of 10 trials on minimizing maximum peak of the external control force 113
Table 6.4 Optimised results of 10 trials on switching duration 115
Table 6.5 Optimised results of 10 trials on minimizing maximum peak of the external control force 118
Table 6.6 Optimised results of 10 trials on switching duration 118

1. Introduction

1.1 Project Overview

Multistability is a common phenomenon found in areas such as biology [1], electronics [2] and mechanics [3]. It is a phenomenon of nonlinear systems showing more than one coexisting stable states, also known as attractors, given by a set of parameters. For example, in nature, the living nervous system [1], and the turbulence in liquid-He-4 [3] are examples of multistable systems. Each attractor has its own basin of attraction. The multistability of a system can be revealed by its basins of attraction, which is defined as the set of initial states converging to the same stable attractor at a steady rate [5]. Therefore, the final state of the system depends on its initial conditions [6]. From the basins of attraction, the number of the coexisting states and the structure of each basin can be revealed.

Multistability can also be observed in various engineering applications. For example, with the development of microelectronics technology, portable electronic devices and wireless Internet of Things (IoTs) sensors have become ubiquitous. Most of these devices are powered by batteries. However, due to the lifespan of the batteries, it is costly of replacing them when these devices are used in remote or hazardous environments. In order to overcome the cost of replacing batteries, the concept of energy harvesting system is proposed. Energy harvesting systems harvest energy to electrical power from ambient environment. Wu et al. have proposed an energy harvesting system which can convert the vibration energy into electrical power [7]. Figure 1.1 shows the methodical model of this systems.

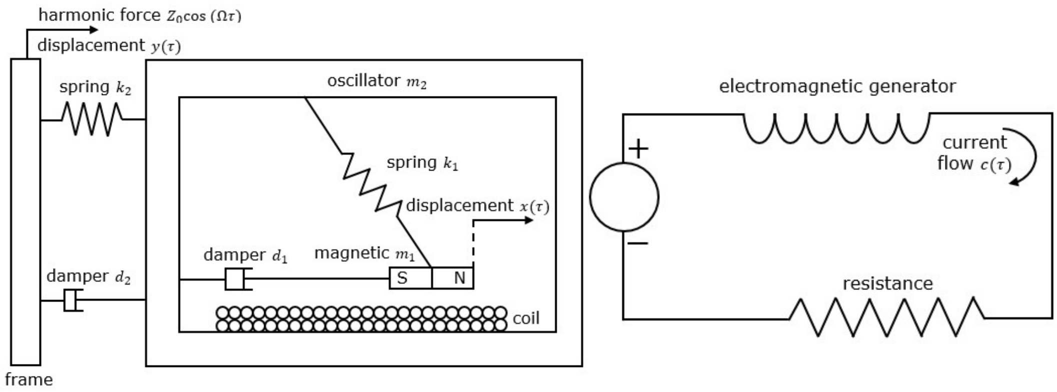


Figure 1.1 Energy harvesting system [7]

The oscillator is connected to a frame (with mass of m_2) with a linear spring (of stiffness k_2) and a viscous damper (with damping coefficient of d_2). There is a coil at the bottom of the oscillator. Above the coil there is a magnetic mass (m_1) which is connected to the oscillator with a linear spring (of stiffness k_1) and a viscous damper (with damping coefficient of d_1). The working principle of this harvesting system is as follows: when the frame is excited by a harmonic force $Z_0\cos(\Omega\tau)$, the oscillator and the magnetic mass will move with displacements $y(\tau)$ and $x(\tau)$ respectively, where τ is the non-dimensional time. Then, a current $c(\tau)$ is observed in the harvesting circuit due to electromagnetic induction. In their research, the oscillator has multistable states with some set of system parameters. Moreover, the state with large amplitude of response and high-energy dynamics can harvest more energy from ambient vibrations. Therefore, the state with larger amplitude and high-energy dynamics is desired in this energy harvesting application.

Another engineering example of multistable systems is a vibro-impact capsule system designed by Liu et al. for rectilinear motion in complex environments [8]. This system can move without external excited force i.e. it can move on its own. Figure 1.2 shows the model of the system. An internal mass m_1 is connected to a capsule with mass m_2 by a linear spring with stiffness of k_1 and a viscous damper with damping coefficient of c. The mass is harmonically driven by an external force $P_d\cos(\Omega t)$. There is a weightless plate which is connected to the capsule with a linear spring with stiffness of k_2. When the relative displacement X_1 between the mass and the capsule exceeds the gap (G) between the mass and the plate, the capsule will be subjected a force from the internal mass. If the force generated by the internal mass is larger than the friction force between the capsule and the environment, the motion of the capsule X_2 will occur. In their research, the capsule system has co-existing states with some set of system parameters. Moreover, in different states, the capsule may move forward or backward or keep still. Therefore, based on different locomotion requirement of the capsule system, the dynamics of the system can be controlled by a multistable controller.

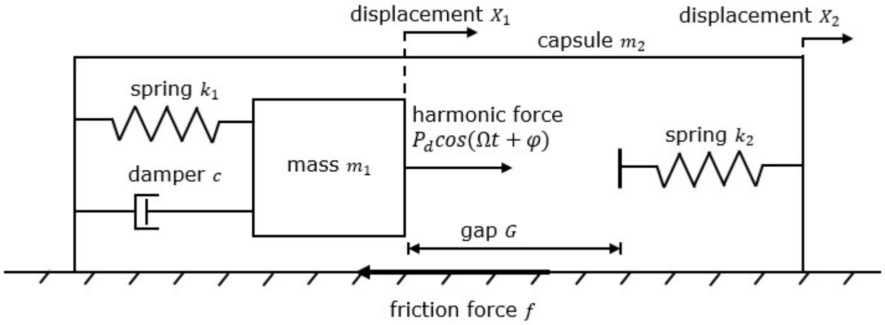


Figure 1.2 Vibro-impact capsule system [8]

In addition, percussive drilling is a drilling method using the impact of a hammering bit of heavy cutting [4]. This drilling method can be used in rock or stiff roil. The rock and stiff roil is loosened by lifting and dropping the heavy cutting or hammering bit. The schematics of percussive drilling is shown in Figure 1.3. The mass m represents the heavy cutting or hammering bit. The spring with stiffness of k, damper with damping coefficient of c and friction force P_f emulate the mechanical properties the rock or roil. The gap between the mass and the slider represent the distance between the bit and the rock. The mass is driven by an external sinusoidal force $P_d \cos(\Omega t + \varphi)$. When the displacement X_m of the mass exceeds the gap, impact occurs. If the acting force exceeds the friction force, drilling progression is observed. Wiercigroch et al. found that the percussive drilling system has co-existing states with some sets of system parameter [9]. In different states, the drilling rate are different. In order to increase the efficiency of drilling the state with high drilling rate is preferred.

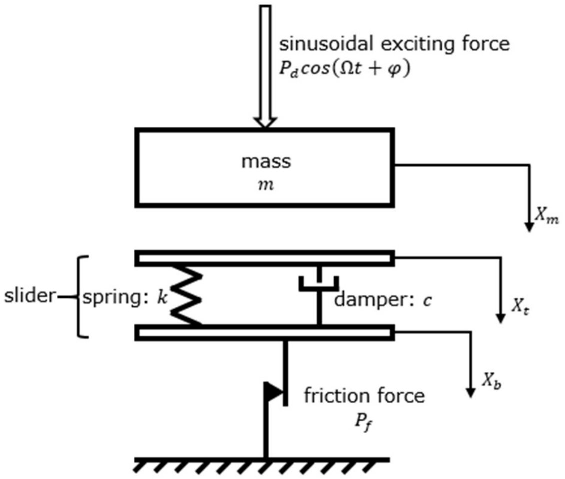


Figure 1.3 Percussive drilling system [9]

Due to the multistability nature and its high sensitivity to initial conditions as well as external noise observed in most dynamical systems, control of multistable systems is challenging. Since multiple stable states coexist in multistable systems, usually one of them is preferred for a specific application. For example, on one hand, the state with a large amplitude of oscillations is needed to gather more energy from the ocean waves in wave energy converter (WEC), which is a common type of vibrational energy harvester [7]; the larger the amplitude of the vibration, the greater the amount of energy harvested. On the other hand, the state with small amplitude of oscillations is preferred in oil well drilling in order to reduce vibrations in it. If the amplitude of vibration of an oil drill is large, its expected lifetime is reduced and the energy input for drilling is also increased [10].

Basins of attraction of a multistable system reveal coexisting states in the system. Figure 1.4 shows the basins of attraction with complex structure of a multistable system and there are three coexisting states (highlighted in red, blue and green colours respectively) in it. Under the effect of a small and unwanted external noise, the state of the system may be perturbed and the system is switched to another stable state. Various methods have been proposed for control of multistable systems. For example, switching the system using a feedforward control procedure [11-12], steering it by applying a short pulse [13-14] and destroying the undesired attractors by using a pseudo-periodic force [15-16] can switch the multistable system to a desired state. However, these methods have limitations. For instance, the destruction of attractors may change system properties and behaviours completely, including the basin of attraction’s structure, in turn making it more complex and difficult to control. Moreover, if there is an unwanted external noise injected to the system, it may switch back to the undesired state due to its high sensitivity to noise and disturbance. Therefore, in this thesis, a new multistable state switching controller, called PD-like control, is proposed. This controller switches the system to the desired state based on the difference between the current and desired trajectories as feedback. This controller can maintain the original basin of attraction and maintain the system in a desired state under the effect of external disturbances. Due to the limitations or requirements of an application, such as the limited capacity of actuators and the system designer’s priorities, multistable systems are required to switch to the desired state as promptly as possible in order to reduce the consumption of energy, the performance of the controller can be evaluated in different aspects such as peak of control input and switching duration. The PD-like controller is optimised by particle swarm optimisation (PSO) [17] with different objectives independently. However, some of them are conflicting with each other. Therefore, the PD-like controller is then optimised with multiple objectives for controller design trade-off.

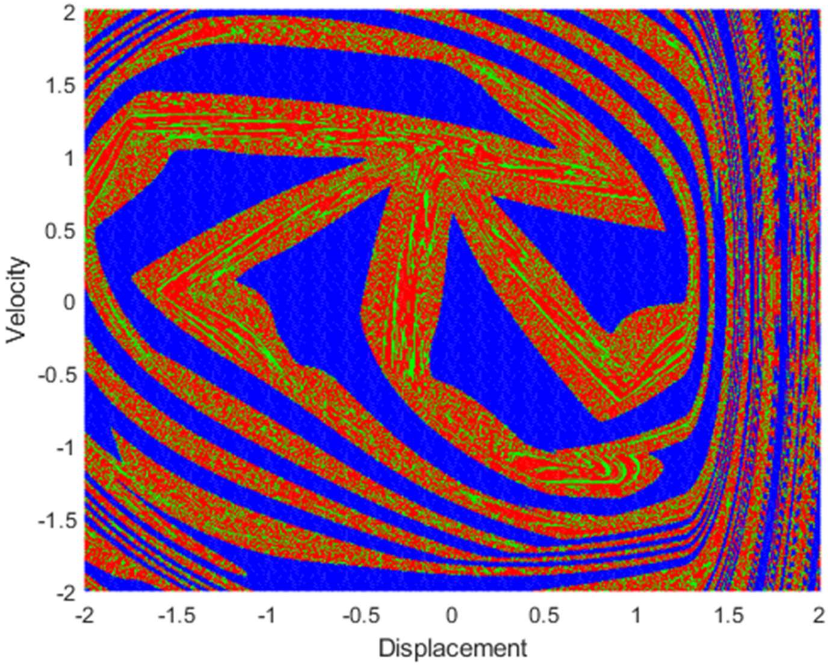


Figure 1.4 The complex structure of the basins of attraction with three coexist state

In this thesis, a novel method called Randomised Triangular Subdivision is proposed to quickly approximate basins of attraction of a multistable system. During the process of basins of attraction identification, the trajectory of each states can be observed. Moreover, with a small change of the system parameters, the structure of the basins of attraction will be changed. Figure 1.5 shows the basins of attraction of a soft impact oscillator and there are two coexisting states with different amplitudes of oscillations in the output response. In addition, due to the complex or occasionally fractal structure of its basins of attraction, a multistable system is very sensitive to external noise.

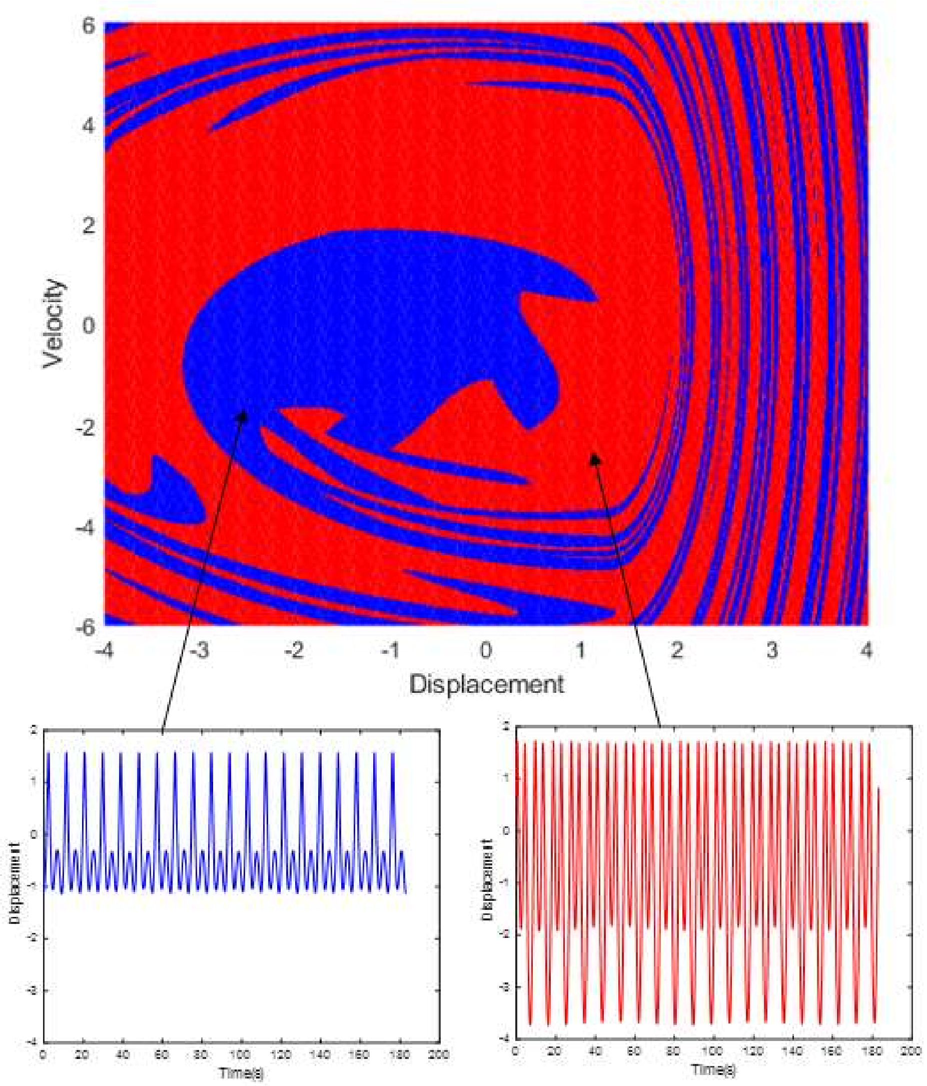


Figure 1.5 The basins of attraction of a soft impact oscillator. Small windows show the amplitudes of output responses in the two coexisting states.

1.2 Aim and Objectives

1.2.1 Aim

The aim of this thesis is to develop a novel controller which can switch the behaviour of systems from the undesired state to its desired equivalent, maintain it in the same state under the effect of unwanted external noise and enhance the performance of the controller based on different constraints (including actuator limits) and requirements of various applications.

1.2.2 Objectives

In order to achieve the above-mentioned aim, this thesis will investigate the following objectives:

1. To propose a fast method to extract knowledge of basins of attraction of a multistable system.
2. To design a simple controller to switch a multistable system from the undesired state to the desired state.
3. To enhance the performance of the controller with single objectives independently.
4. To enhance the performance of the controller with multiple objectives simultaneously.
5. To investigate the performance of the controller by numerical simulation on test systems.

1.3 Three Test Systems

Three nonlinear systems - the Duffing oscillator [18-21], the soft impact oscillator [22-27] and the soft impact oscillator with a drift [28-31], will be employed for evaluating the performance of the proposed methods in this thesis. Both the Duffing and soft impact oscillators are simple one degree of freedom systems and they are representative smooth and non-smooth systems respectively. Moreover, they are typically multistable. The soft impact oscillator with a drift can model the basic mechanism of engineering systems such as rotor, gearbox and percussive drilling system.

1.3.1 Duffing Oscillator

A Duffing oscillator is realised by an electro-magnetised vibrating beam when a harmonic driving force is applied on it [21]. Figure 1.6 shows the schematics of a Duffing oscillator which includes two magnets to prevent the beam becoming too close to one of them. The mathematical model of the oscillator is shown below [18]:

$$\ddot{x} + \delta\dot{x} + \beta x + \alpha x^3 = Fcos\omega t \tag{1.1}$$

where x is the displacement of the beam, δ is the damping ratio of the beam, β is the stiffness ratio of the beam, α is the cubic stiffness parameter of the beam, F is the amplitude of the harmonic driving force, ω is the frequency of the harmonic driving force and t is the time.

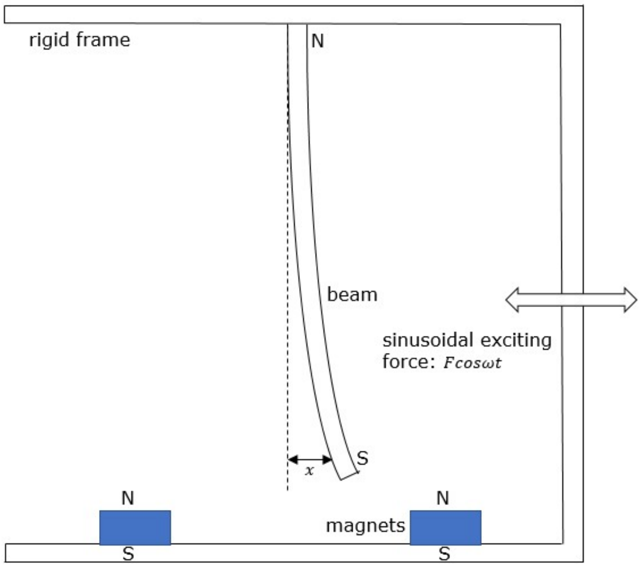


Figure 1.6 Schematics of Duffing oscillator [21]

1.3.2 Soft impact oscillator

Figure 1.7 shows the model of a soft impact oscillator and a mass [23]. They are connected via a linear spring with stiffness of k_2 and a viscous damper with damping coefficient of c. The soft impact oscillator operates when a harmonic force $a\sin\Omega t$ is applied to the exterior frame and the mass will move. Subsequently, if the displacement exceeds the gap g between the mass and the second spring k_1, an impact will occur, thus changing the movement of the mass. The mathematical model of the soft impact oscillator is shown below [23]:

$$\ddot{x} = \Gamma\ sin\ (\omega\tau)\ - 2\xi\dot{x} - x - \beta(x-e)H(x-e) \qquad (1.2)$$

This is a non-dimensional equation of the soft impact oscillator model. Moreover, the equations below show how the dimensional parameters of the soft impact oscillator are changed to non-dimensional parameters.

$$\begin{gathered} \mathrm{x} = \frac{\mathrm{y}}{\mathrm{x_0}}, \quad \tau = \omega_\mathrm{n}\mathrm{t}, \quad \beta = \frac{\mathrm{k_2}}{\mathrm{k_1}}, \quad \omega = \frac{\Omega}{\omega_\mathrm{n}} \\ \xi = \frac{\mathrm{c}}{2\mathrm{m}\omega_\mathrm{n}}, \quad \Gamma = \frac{\mathrm{a}}{\mathrm{x_0}}, \quad \mathrm{e} = \frac{\mathrm{g}}{\mathrm{x_0}} \end{gathered} \qquad (1.3)$$

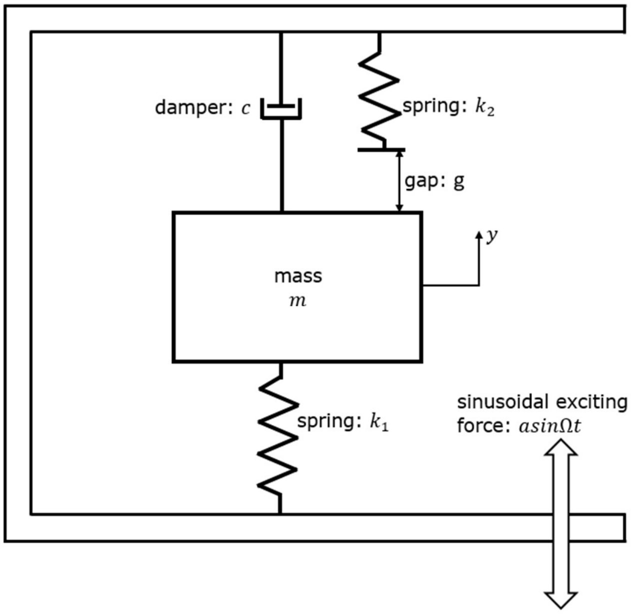


Figure 1.7 Schematics of soft impact oscillator [23]

where $\mathrm{x_0}$ and ω_n are two reference parameters for the non-dimensional formulae. $\mathrm{x_0}$ is the reference distance and $\mathrm{x_0} > 0$. ω_n is the natural frequency and $\omega_\mathrm{n} = \sqrt{\mathrm{k_1/m}}$. Moreover, x is the non-dimensional displacement of the mass, τ is the non-dimensional time, β is the stiffness ratio of the two springs, ω is the frequency ratio of the system, ξ is the damping ratio of the system, Γ is the non-dimensional amplitude of the harmonic driving force and g is the gap between the mass and the second spring with stiffness of k_2. $\mathrm{H}(.)$ is the Heaviside step function.

1.3.3 Soft impact oscillator with a drift

The soft impact oscillator with a drift has a single mass and two plates which are connected by a linear spring and a viscous damper, as shown in Figure 1.8. It provides a simplified model for percussive drilling systems. The behaviours of this system can be divided into three modes. The first mode concerns no contact between the mass and the plates. After the application of a harmonic force, the mass and top plate will not move if the displacement of mass does not exceed the gap between them. This is followed by the second mode called contact without progression. This can be observed if the displacement of mass exceeds the gap but its acting force does not exceed the critical value of friction force and thus the mass and top plate will move together but the bottom plate will not. When the displacement of mass exceeds the gap and displacement of the top plate and the acting force is larger than the friction, the mass, top plate and bottom plate will move together and the system exhibits a behaviour in the third mode called contact with progression. Based on its behaviours, the system can be described as a set of piecewise functions, with the Heaviside step function H (.) used to determine the phase of the systems. The system model is shown as follows [28]:

$$\begin{gathered}\dot{x} = y \\ \dot{y} = acos(\omega\tau + \varphi) + b - P_1P_2(1 - P_3)(2\xi y + z - v) - P_1P_3 \\ \dot{z} = P_1 y - (1 - P_1)(z - v)/2\xi \\ \dot{v} = P_1P_3P_4[y + (z - v - 1)/2\xi]\end{gathered} \tag{1.4}$$

The non-dimensional variables and parameters are as follows:

$$\begin{gathered}\tau = \Omega_0 t, \qquad x = \frac{k}{P_{max}}X_m, \qquad y = \frac{dx}{d\tau} = \frac{k}{\Omega_0 P_{max}}\dot{X}_m, \qquad z = \frac{k}{P_{max}}X_t \\ v = \frac{k}{P_{max}}X_b, \qquad \omega = \frac{\Omega}{\Omega_0}, \qquad a = \frac{P_d}{P_{max}}, \qquad b = \frac{P_s}{P_{max}}, \qquad d = \frac{P_f}{P_{max}}\end{gathered} \tag{1.5}$$

$$\Omega_0 = \sqrt{\frac{k}{m}}, \qquad \xi = \frac{c}{2m\Omega_0}, \qquad g = \frac{k}{P_{max}}G$$

where x, z, v stand for the non-dimensional displacement of the mass, top and bottom plate respectively, y is the non-dimensional velocity of mass, τ is the non-dimensional time, ω is the frequency ratio, ξ is the damping ratio, g is the non-dimensional gap between the mass and top slider, a is the non-dimensional amplitude of harmonic part of external force, b is the non-dimensional static part of external force, φ is the phase shift and H stands for the Heaviside step function.

$$\begin{aligned} P_1 &= H(x - z - g) \\ P_2 &= H(2\xi\dot{z} + z - v) \\ P_3 &= H(2\xi\dot{z} + z - v - 1) \\ P_4 &= H(\dot{v}) \end{aligned} \tag{1.6}$$

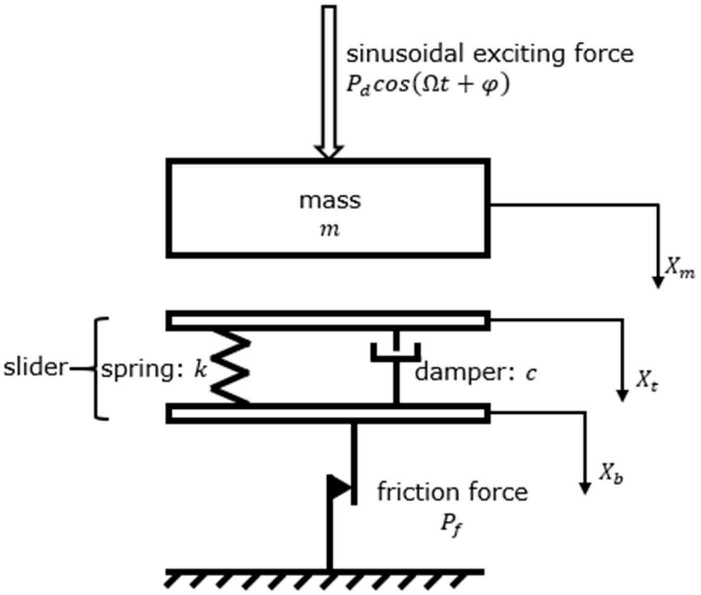


Figure 1.8 Schematics of soft impact oscillator with a drift [28]

1.4 Contributions of the thesis

The contributions of this study are shown below:

- A Randomised Triangular Subdivision method is proposed to quickly approximate the basins of attraction of multistable dynamical systems. This method's performance has been compared with the traditional brute force algorithm method. With the same number of initial conditions under consideration, the proposed method can accurately identify the basins of attraction and capture the geometrical and topological features of the basins of attraction.

- A PD-like controller is proposed for controlling state switching in multistable systems. The PD-like controller can switch the system from an undesired to the desired state without changing the parameters and destroying the original basins of attraction of the system. This feedback controller exploits the difference between the current and desired trajectories for state switching. This controller does not require knowledge of the plant's nonlinearity.

- The PD-like control is optimised by global optimisation methods to meet the requirements of the decision maker or the resource constraints. Constrained PD-like control reduces the required actuator power by limiting the strength of the control input in sacrifice of long switching duration. Moreover, the performance of the PD-like controller is respectively optimised by Particle Swarm Optimisation and Nondominated Sorting Genetic Algorithms-II (NSGA-II) based on different performance metrices independently and simultaneously. Performance metrics under consideration in this thesis are maximum peak of control input and switching duration in order to reduce the energy consumption and save the time used in state switching.

Published Work

Liu, B., Fung, W.K. and Liu, Y., 2018, February. State switching of a multistable impacting system using PD-like control. In *MATEC web of conferences* (Vol. 148). EDP Sciences.

1.5. Thesis Organisation

The rest of the thesis is organised as follows. Chapter two proposes a fast method of approximating the basins of attraction through the proposed Randomised Triangular Subdivision method. It utilises Delaunay triangulation to quickly explore the dynamic systems' basins of attraction. This method's performance is investigated with multistable systems with different numbers of coexisting stable states. Chapter three propses a new multistable PD-like controller for state switching and its performance is investigated using the three test systems outlined in section 1.3. Chapter four investigates the effect of constrained control input on the behaviours of the proposed PD-like controller. Chapter five shows the sensitivity of PD-like control through an analysis of the maximum peak of external control input and the switching durations under different control parameters. In chapter six, the performance of the PD-like controller is enhanced by particle swarm optimisation with two independent objectives. Chapter seven shows how the performance of the PD-like controller is optimised under multiple conflicting objectives using Non-dominated Sorting Genetic Algorithms-II (NSGA-II). At last, chapter eight concludes this thesis and offers recommendations for future study.

Chapter 2 Reconstruction of Basins of Attraction

This chapter proposes a novel and fast method to estimate the basins of attraction of multistable systems. From the basins of attraction of a given multistable system, the number of coexisting states and the structure of each basin can be observed and the trajectories of the states can be found during the process of basins of attraction identification.

2.1 Introduction

Dynamic responses and stable states of a system are dependent on its parameters and initial conditions. The system with different initial conditions will converge to different stable states and the stable states are called attractors [5]. In a multistable system, there is more than one stable state under the same set of system parameters and the regions which partition the initial conditions converging to the same state is called basin of attraction [32-34]. Most controllers for switching states in multistable system require knowledge of the basins of attraction for the trajectory of different states and the features of different basins. Therefore, it is important to identify the basins of attraction of multistable systems for switching to favourable state in different applications. Furthermore, each basin of attraction is associated with a state trajectory and thus the trajectories of coexisting states of the system can be recovered during the process of identification of basins of attraction.

2.2 Traditional Method to Identify Basins of Attraction

The traditional method for identification of basins of attraction of nonlinear systems is through the brute force algorithm. This method involves checking the state of each point or initial condition in the relevant regions of the initial condition space, and is computationally intensive. The pseudo code of brute force algorithm is shown below:

1. Discretise the initial condition space.
2. Raster Scan: for each possible set of initial conditions, simulate the system and identify the converged state.
3. Assign colour to each converged state for easy visualisation.

The disadvantages of this method are:

1. It is computationally intensive for basins of attraction generation at high resolution. Since the system at initial conditions which are close to each other often converge to the same state. It is arguably a waste of time to check the state of every individual point in initial condition space within the same region

2. Boundary between different basins or separatrix is usually highly nonlinear and the basins of attraction are usually highly fragmented. It is difficult to extract geometrical and topological features of basins of attraction with great details using raster scan at low resolution.

2.3 The Proposed Method

A new method, called Randomised Triangular Subdivision, is proposed which applies the Delaunay Triangulation [35-36] to quickly explore the initial condition space for approximating the basins of attraction of a multistable system. The initial condition space is then covered by triangles. Triangles are employed in initial space coverage because they are the simplex shape which can cover all 2D spaces. The proposed method adopts a randomised approach to randomly select the first generation of initial conditions which can maintain diversity of initial conditions and are as uniformly distributed as possible to cover the most of the area (more than 90%) of the region of interest. The proposed method then subdivides triangles, which have at least one vertex in a different state or at least one initial condition converged to a different state, into three smaller triangles for enhancing the level of details in the basins of attraction. This facilitates fast estimation of the boundary or separatrix between different basins of attraction. The mathematical definition of triangulation in n-dimensional space is shown below:

Definition 2.1:

A triangulation of a finite point set $P \subset \mathbb{R}^n$ is a collection $\mathcal{T}$ of triangles, such that

(1) $conv(P) = \bigcup_{T \in \mathcal{T}} T$;

(2) $P = \bigcup_{T \in \mathcal{T}} V(T)$; and

(3) For every distinct pair $T, U \in \mathcal{T}$ the intersection $T \cap U$ is either a common vertex, or a common edge, or empty. [37]

where, $V(T)$ is the set of vertices of a triangle T. $conv(P)$ is the convex hull of the set P. Convex hull of a set P is the smallest convex set that include all points in the set P [37]. The definition of a convex set is shown below:

Definition 2.2:

A set $P \subset \mathbb{R}^n$ is convex if and only if $\sum_{i=1}^{n} \lambda_i a_i \in P$, for all $n \in \mathbb{N}, a_1, \dots, a_n \in P$, and $\lambda_1, \dots, \lambda_n \geq 0 \; with \; \sum_{i=1}^{n} \lambda_i = 1$ [37]

The Delaunay Triangulation method triangulates a n-dimensional space with several discrete points and there is no point in the circumcircle of any triangle. The Delaunay Triangulation is used for estimating the basins of attraction because when a new point is inserted, it only affects the triangles which are in the vicinity of this point and it will not affect the triangles which are far from that point [38-40] and thus the triangulation is locally updated. No re-triangulation on the whole new point set is required in each subdivision step. Two efficient algorithms are commonly used for finding the Delaunay Triangulation of a given point set. One of them is called Flip Algorithm [41-47] and the other is Incremental Construction [41-42, 48-52].

2.3.1 Lawson Flip Algorithms

The process of Lawson flip is shown below:

(1) Triangulate the set of points P randomly.

(2) Check the subtriangulation by 4 points which are in the convex position. Subtriangulation refers to the triangulation of a subset of points. If the subtriangulation is not Delaunay (i.e. a point is in the circumcircle of a triangle as shown in Figure 2.1(b)), switch the subtriangulation to the Delaunay triangulation (no point is in the circumcircle of a triangle as shown in Figure 2.1(a)). This operation is called a flip operation.

(3) Repeat step (2) until the triangulation of P is a Delaunay triangulation. Moreover, this will need at most $O(n^2)$ flips, where n is the number of the points in the set P

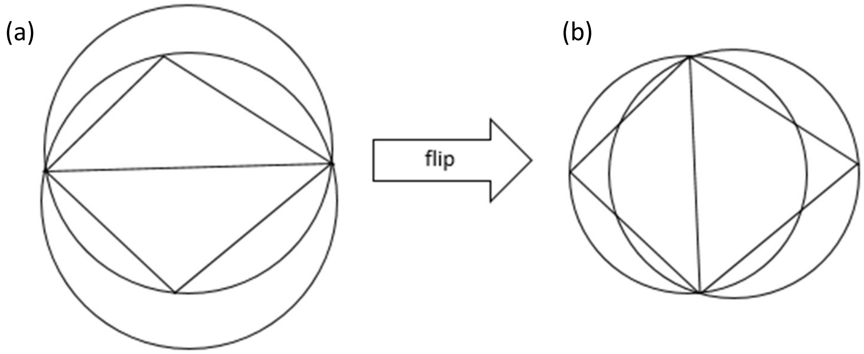


Figure 2.1 (a) Delaunay triangulation (b) Non-Delaunay triangulation

2.3.2 Incremental Construction

The process of incremental construction is shown below:

(1) Build a triangle with three artificial points and this triangle contains all the points in the set P.

(2) Inserting the points in the set P one after another. Assuming R points are inserted, and a new point s will be inserted. Find the triangle $\Delta = \Delta(p, q, r)$ contains the point s in $\mathcal{DT}(R)$, where $\mathcal{DT}(R)$ is the Delaunay Triangulation of $R \subseteq P$ and p, q, r are the vertices of the triangle Δ. This triangle is then switched to three triangles by connecting the point s with the vertices p, q, r. Figure 2.2 shows the triangulation $\mathcal{T}$ of $R \cup \{s\}$.

(3) Replace $\mathcal{T}$ by $\mathcal{DT}(R \cup \{s\})$ through a Lawson flip (Step 2 shown in Section 2.3.1) as shown in Figure 2.3.

(4) Repeat step (2) and (3) until the triangulation of P is a Delaunay Triangulation. Then delete the extra points and their incident edges. This algorithm needs at most $O(n\,log\,n)$ flips to be implemented.

Compared with Lawson Flip Algorithms, Incremental Construction has a lower computation complexity [41]. Therefore, Incremental Construction is adopted for Delaunay triangulation in this algorithm.

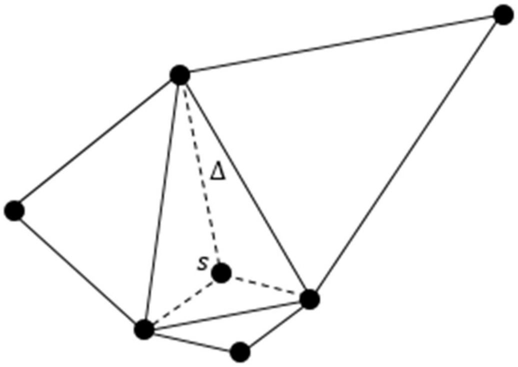


Figure 2.2 Step 2 of incremental construction

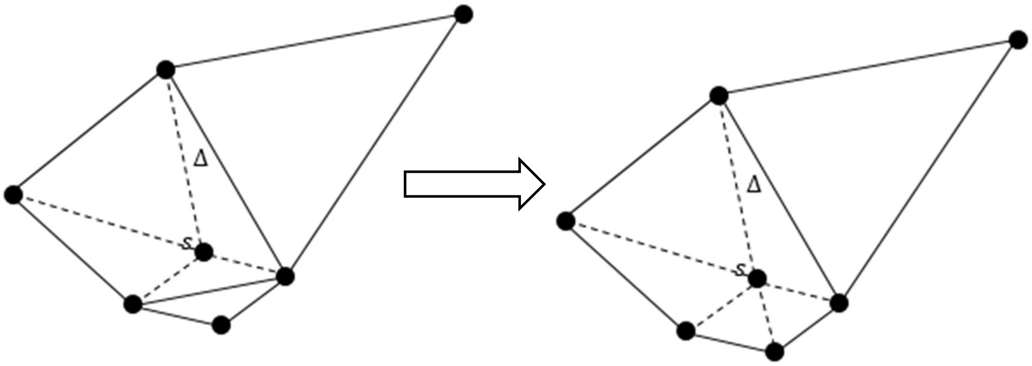


Figure 2.3 Step 3 of incremental construction

2.4 Randomised Triangular Subdivision

The first step of the randomised triangular subdivision method is to apply the Delaunay Triangulation on a set of initial conditions to divide the chosen region of interest to several triangles. Then, the vertices of each triangle are checked: If all three vertices of a triangle are in the same state, i.e. the initial conditions denoted by the three vertices converge to the same stable state, it is assumed that all the points (initial conditions) in the triangle are in the same state. If at least one of the vertices is in a different state to the other vertices then the state of the centroid point will be checked. The following paragraphs describe in details the process by which this method approximates the basins of attraction of a multistable system.

The proposed Randomised Triangular Subdivision algorithm is presented as follows:

1. A set of points (initial conditions) in a given region of interest in the initial condition space of the multistable system are randomly selected. The distribution of these initial conditions is checked: if they are too crowded in small regions in the initial condition space, they should be re-selected until the distribution of the initial conditions can cover the region as thoroughly as possible (over 90% of the region of interest in initial condition space).

2. A convex hull based on these selected initial conditions is constructed, and triangulated using Delaunay Triangulation. Then the states of the vertices of each triangle are checked. For each triangle, if the vertices of the triangle are in the same state, the method will process according to step 3. If at least one vertex of the triangle is in a different state compared to the other two vertices, the method will process according to step 5. Two different threshold values p and q of triangular area are selected based on the area of the convex hull. p and q are measured on the percentage of area of the whole initial condition space. p is the threshold value of the area of the triangles whose vertices are in the same state. q is the threshold value of the area of the triangles who has at least one vertex is in a different state compared to the other two vertices. With smaller p and q, the boundaries between different states are smoother and have higher level of details. For a triangle with three vertices in the same state, if the area of this triangle is smaller than p the process will stop. For a triangle with at least one vertex in a different state compared to the other two vertices and if the area of this triangle is smaller than q the process will stop. This step is depicted in the red dashed box in Figure 2.4.

3. For each triangle, check its area. If the area of the triangle is smaller than p the vertices of the triangle are recorded. If the area of the triangle is larger than p the state of its centroid point is checked. If the state of the centroid point of the triangle is in the same state as its vertices, the method will process according to step 4. If the centroid point of the triangle is in a different state compared to its vertices then the method will process according to step 5. This step is depicted in the purple dashed box in Figure 2.4.

4. If the area of the triangle is smaller than or equal to p then record the vertices of the triangle. Otherwise, check the state of its centroid point. If the state of the centroid point of the triangle is in the same state as its vertices then record this centroid point. If the centroid point of the triangle is in a different state compared with its vertices then the method will process according to step 5. This step is depicted in the purple dashed box in Figure 2.4.

5. Triangulate the triangle based on the vertices and centroid of the triangle. Check the area of the new triangle. If it is smaller than or equal to q then record the vertices of the new triangle. Otherwise, check the state of the triangle's vertices. If all three vertices of the triangle are in the same state, the method will process according to step 3. If at least one vertex of the triangle is in a different state compared to the other two vertices repeat step 5. If the areas of all the triangles, which have at least one vertex is in a different state, are less than q then the method will process according to step 6. This step is depicted in the blue dashed box in Figure 2.4.

6. Check the state of each initial condition in the set that consists of randomly selected initial conditions in step 1 and the newly added ones by subdivision in steps 3, 4 and 5. Then triangulate the convex hull based on all the augmented initial conditions set. For each triangle, if all the three vertices are in the same state assume that all the points in this triangle are in the same state. If one vertex is in a different state compared with the other two vertices then assume all the points

in this triangle are in the same state as the two vertices that are in the same state. If all three points are in different states then check the state of the centroid point of the triangle and assume that all the points in this triangle are in the same state as the centroid point. Figure 2.5 shows the triangle with different conditions of its vertices. This step is depicted in the orange dashed box in Figure 2.4.

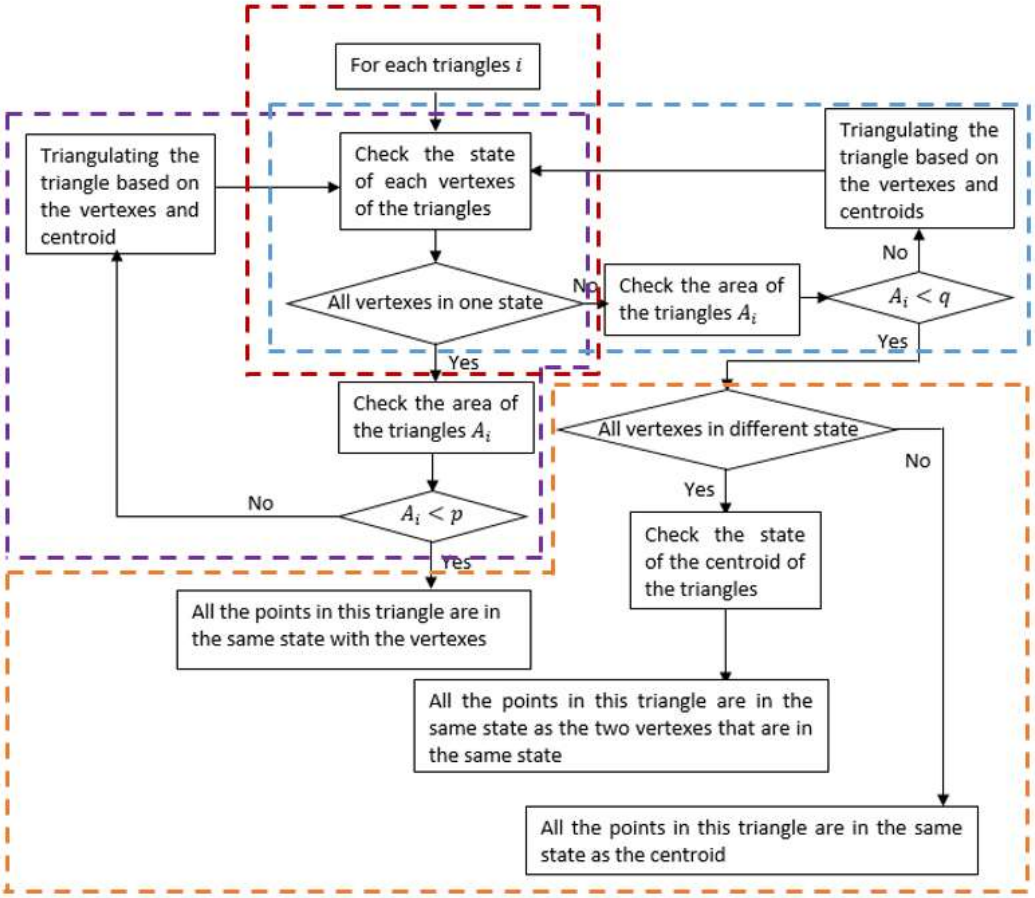
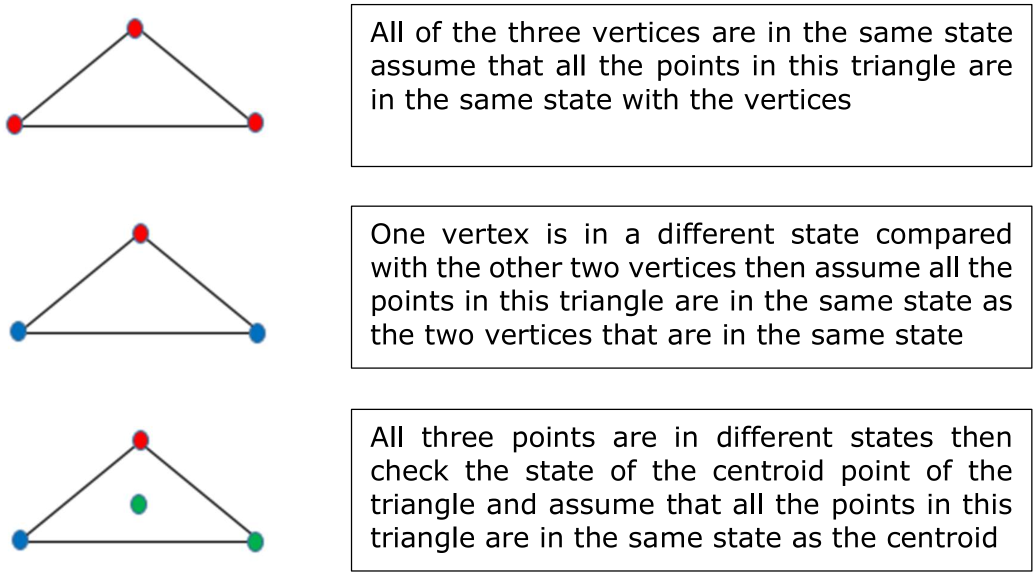

For each triangles i
Check the state of each vertexes of the triangles
Triangulating the triangle based on the vertexes and centroid
Triangulating the triangle based on the vertexes and centroids
All vertexes in one state
No
Yes
Check the area of the triangles A_i
$A_i < q$
No
Yes
Check the area of the triangles A_i
$A_i < p$
No
Yes
All vertexes in different state
Yes
No
Check the state of the centroid of the triangles
All the points in this triangle are in the same state with the vertexes
All the points in this triangle are in the same state as the two vertexes that are in the same state
All the points in this triangle are in the same state as the centroid

Figure 2.4 Flowchart of the randomised triangular subdivision. Red dashed block shows step 2, purple block shows step 3 and 4, blue dashed block shows step 5 and orange dashed block shows step 6.



All of the three vertices are in the same state assume that all the points in this triangle are in the same state with the vertices

One vertex is in a different state compared with the other two vertices then assume all the points in this triangle are in the same state as the two vertices that are in the same state

All three points are in different states then check the state of the centroid point of the triangle and assume that all the points in this triangle are in the same state as the centroid

Figure 2.5 Different conditions of a triangle in Delaunay Triangulation of Initial Condition Space.

2.5 Numerical Simulation

In order to demonstrate the effectiveness of the proposed method, two multistable systems are studied, namely: Duffing oscillator and soft impact oscillator, in simulation.

2.5.1 Duffing Oscillator with Two Coexisting Stable States

The section below provides a detailed description of the process of estimating the basins of attraction of the Duffing oscillator.

Firstly, the parameters of the Duffing oscillator are set as: $k = 0.9$, $\Gamma = 1.9$ and $\omega = 1.2$. The Duffing oscillator has two co-existing states under this set of system parameters [62]. Moreover, the region of interest in the initial condition space is selected from -5 to 5 in the x-axis (displacement) and y-axis (velocity). 50 initial conditions are randomly selected in this region. Several tests have been conducted for finding the suitable number of initial conditions. If the number of the initial condition is too small, the detail of the basins of attraction may be lost. If the number of the initial conditions is too large, the computing complexity will increase drastically. The area of the convex hull which is created by these 50 initial conditions is 95.1207, which is 95.1207% of the selected region. Moreover, the area threshold values p and q are 0.095 and 0.0095 respectively, which are 0.1% and 0.01% of the area of the convex hull respectively. Therefore, these 50 initial conditions are used for the first iteration of the simulation. The distribution of these initial conditions is shown in Figure 2.6.

Secondly, the convex hull of the 50 selected initial conditions is triangulated using the Delaunay Triangulation, as illustrated in Figure 2.7 and 90 triangles are generated as counted by the developed MATLAB code. Next, the states of the vertices of each triangle are checked. There are 40 triangles with all three vertices in the same states (as shown in Figure 2.8) while the remaining 50 triangles have at least one vertex that is in a different state compared to the others (as shown in Figure 2.9). Steps 4 and 5 were repeated until the area of the triangle is smaller than 0.01 times of the initial convex hull area or all three vertices of the triangle are in the same state. The subdivision process ends with a total of 1224 points, which provides the new set of initial conditions for resultant Delaunay Triangulation.

Thirdly, the state of the new set of initial conditions is checked and the convex hull is triangulated using the Delaunay Triangulation based on the new set of initial conditions. The convex hull is subdivided into 2438 triangles and there are two stable states under this set of system parameters. Moreover, as shown is Figure 2.10, 607 points (initial conditions) converge to the first stable state (red dots) and 617 points converge to the other state (blue dots). In order to identify the basins of attraction, for each triangle, if all three vertices are in the same state, fill the triangle with the same colour as the vertices. If two vertices are in the same state and the remaining vertex is in a different state, fill the triangle with the same colour as the two points which are in the same state. Therefore, there are 1216 triangles that belong to the first state (filled in red colour) and 1222 triangles that belong to the second state (filled in blue colour). The basins of attraction estimated by this method are shown in Figure 2.11 below.

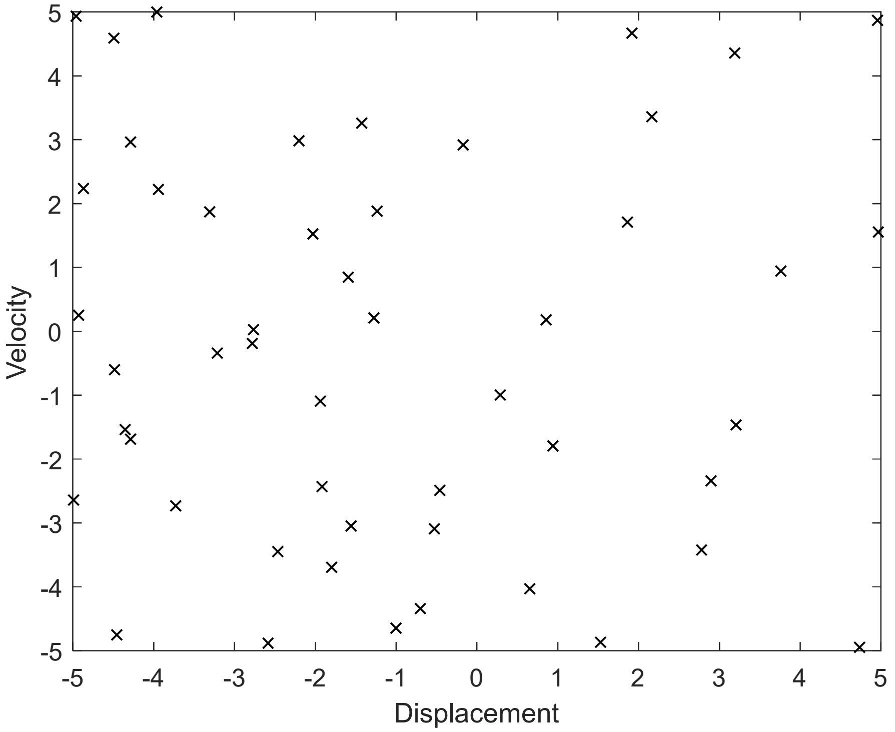


Figure 2.6 Random samples of the initial conditions

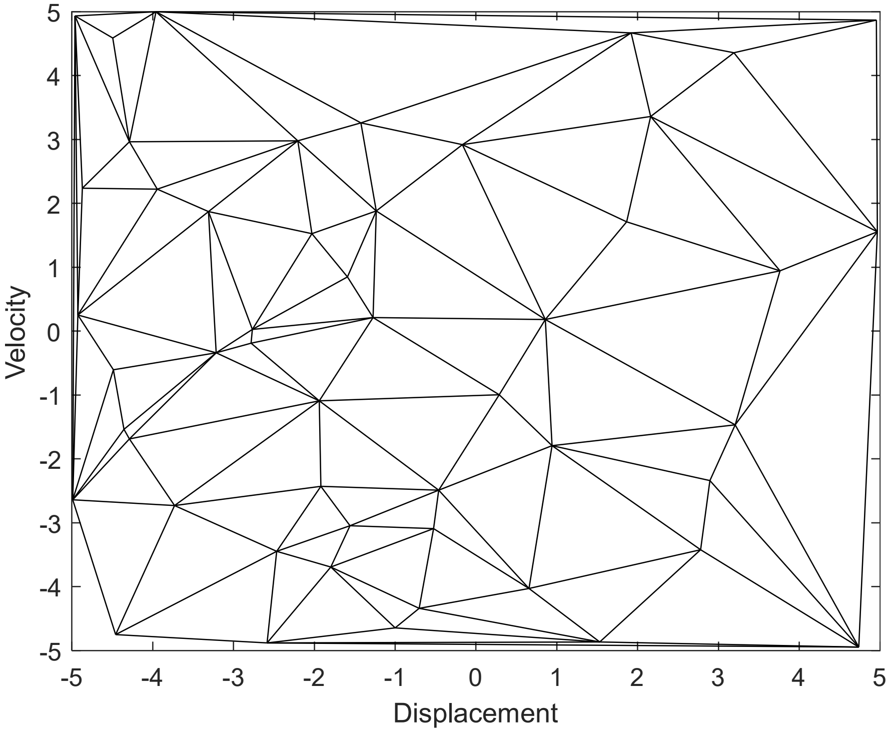


Figure 2.7 Triangulating the convex by using Delaunay Triangulation based on these 50 initial conditions

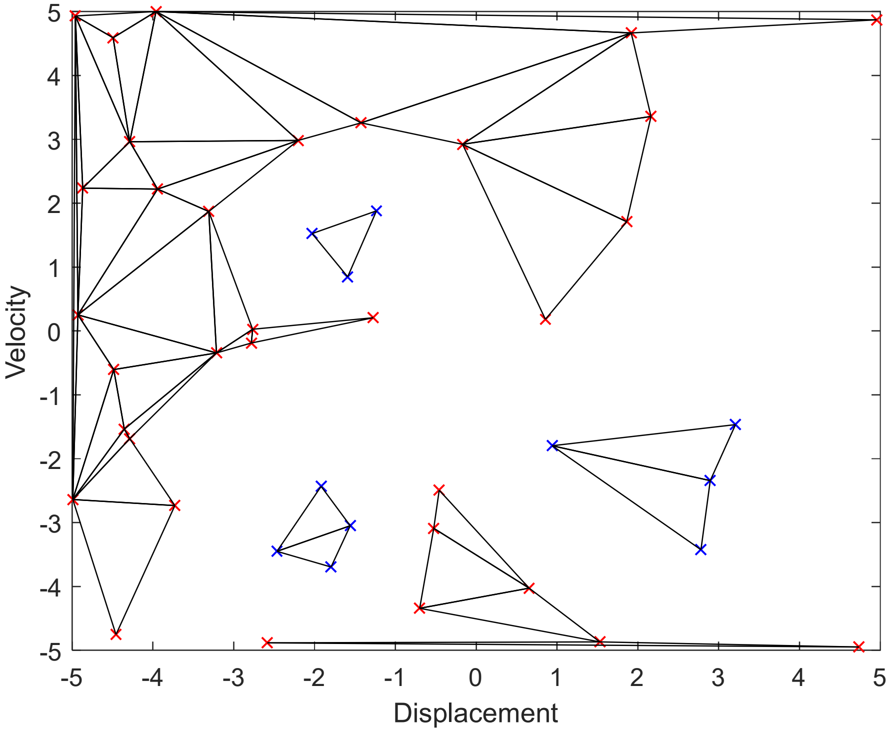


Figure 2.8 The triangles with vertices that are all in the same state. The colour of the point shows the vertex located in which state. Needs rewriting for clarity

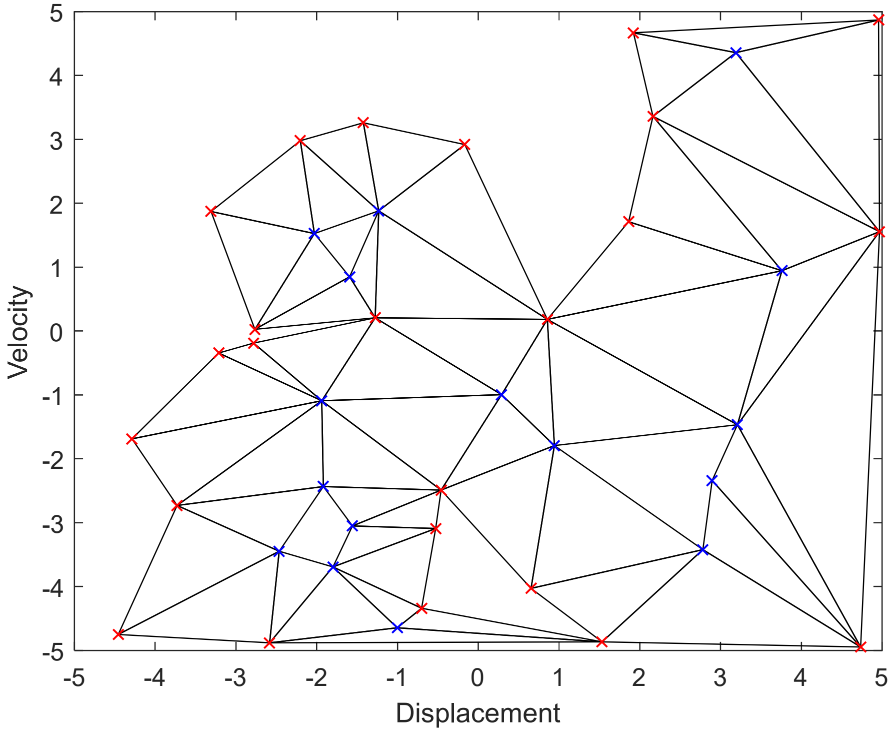


Figure 2.9 Triangles which have at least one vertex that is in a different state compared with the others. The colour of the point shows the vertex located in which state. Needs rewriting for clarity

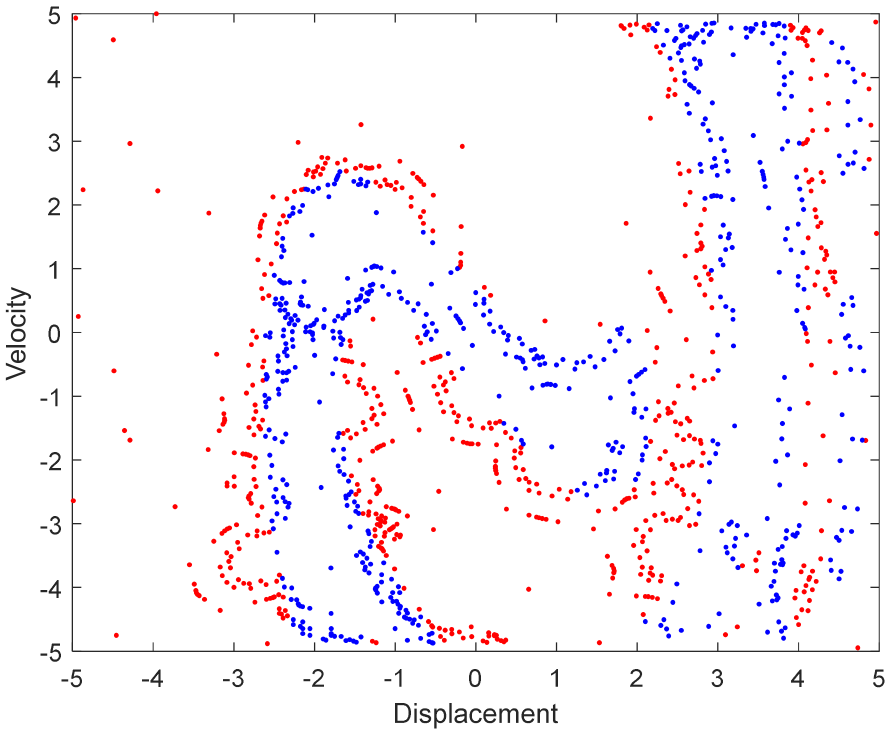


Figure 2.10 Distribution of the augmented set of initial conditions. The colour of the point shows the initial condition converge to which state.

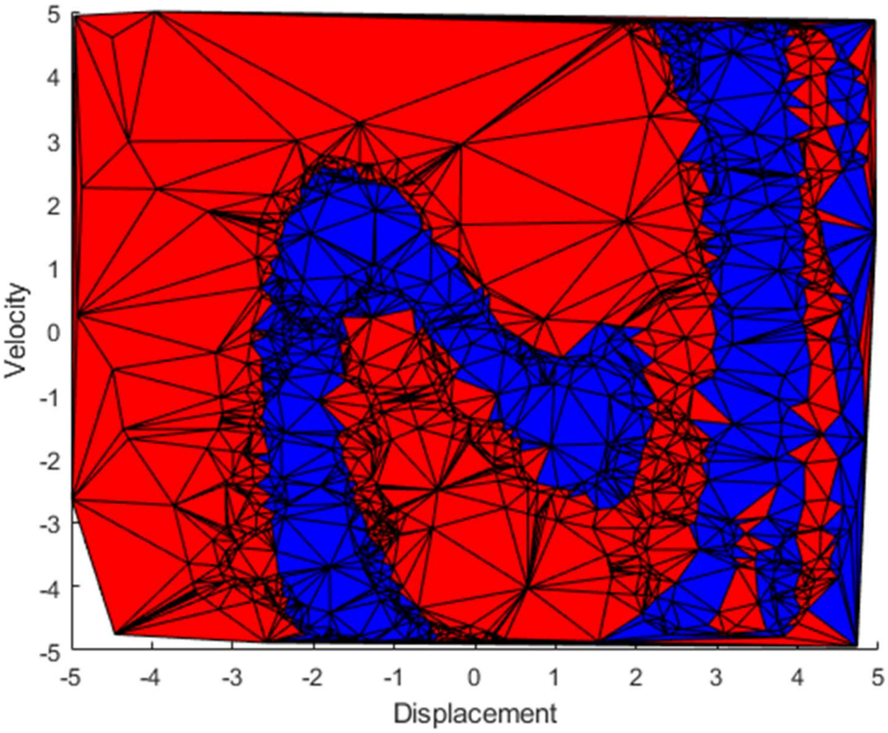


Figure 2.11 Basins of attraction estimated using the random triangular subdivision with 1224 initial conditions.

In order to check the accuracy of the basins of attraction estimated by the proposed randomised triangular subdivision method, a high resolution basins of attraction of the Duffing oscillation under the same set of parameters is identified by traditional brute force algorithm. High resolution basins of attraction are shown in Figure 2.12. The resolution of these basins of attraction is 1001*1001. These basins of attraction estimated by the brute force algorithm, which act as a ground truth, is compared with that estimated by the randomised triangular subdivision method. The features of both basins of attraction are similar by observation. Then, the similarity between these two basins of attractions is calculated. The main method for doing so involves checking each point in the high-resolution basin of attraction to see if they are in the same state in the basins of attraction which were estimated using the proposed randomised triangular subdivision. A comparison shows that the similarity between these two basins of attraction is 97.04%. Figure 2.13 shows the basins of attraction with the resolution of 35*35 using the brute force method with a similarity of 94.95% with the ground truth. Figure 2.16-2.23 show a comparison of the basins of attraction estimated by the randomised triangular subdivision method and the brute force algorithm.

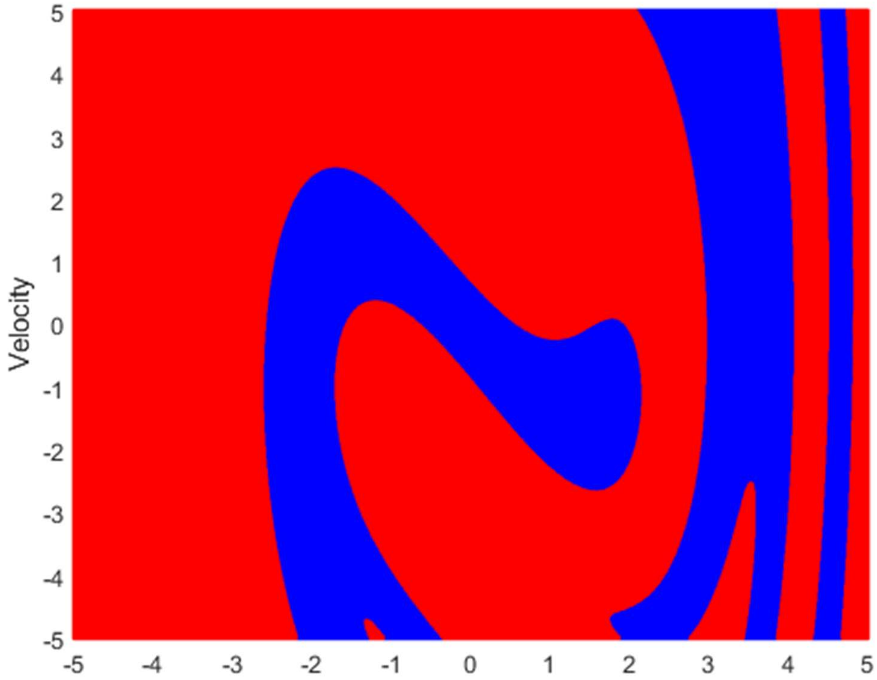


*Figure 2.12 High Resolution Basins of attraction of Duffing Oscillator (1001*1001)*

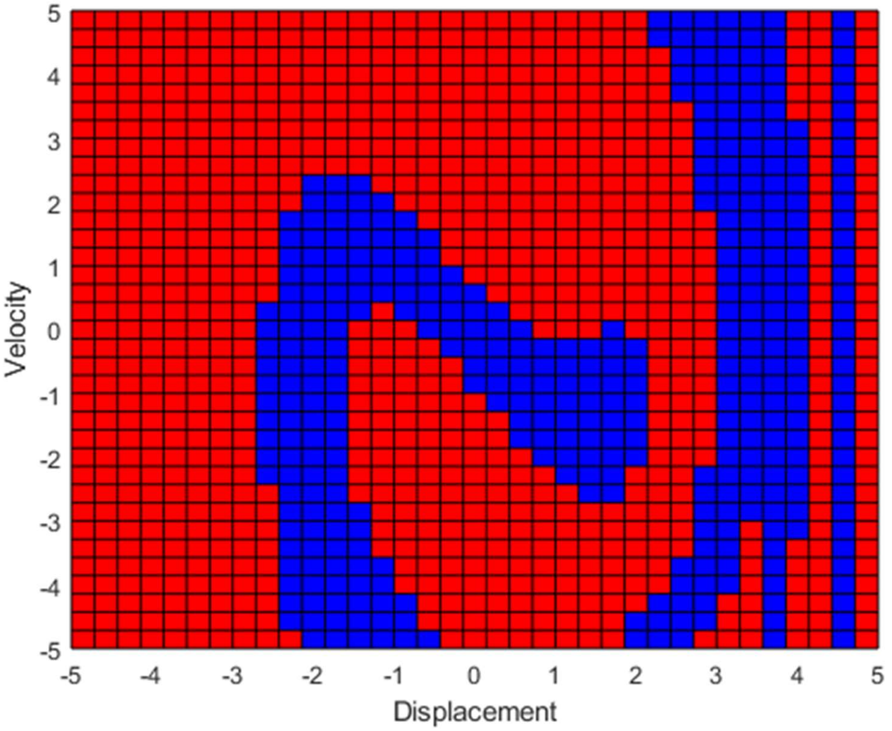


*Figure 2.13 Basins of attraction estimated by the brute force algorithm with resolution 35*35 (1225 initial conditions)*

Another trial of identification the basins of attraction using the randomised triangular subdivision method is conducted in order to investigate the evolution of basins of attraction through subdivision of the proposed method. The distribution of these 50 initial conditions and the state of each initial condition are shown in Figure 2.15. 36 initial conditions converge to the first state (red points) and the other 14 initial conditions converge to the second state (blue points). The area of the initial convex hull is 95.0287 and the threshold values of the area p and q are 0.95 and 0.095 respectively, which correspond to 1% and 0.1% of the area of the convex hull respectively.

50 points are randomly selected for the first generation and the convex hull is divided into 91 triangles which is shown in Figure 2.14. After checking the state of the vertices of each triangle, 52 triangles are passed to the next step and 52 centroid points of triangles are selected for the second generation. Each of these 52 triangles is subdivided into smaller triangles based on its vertices and centroid points. Figure 2.16 shows the convex hull after the first subdivision. Figure 2.17 shows the distribution of the points in the second

generation which includes the first 50 initial conditions and the 52 centroid points (102 initial conditions). The colour of the point shows which state that point (initial condition) converge to. Figure 2.18 shows the basins of attraction estimates based on these points in the second generation. Additional 3 iterations are required before the area of the triangle is smaller than 0.095 or all three vertices of the triangle reach the same state. Moreover, 133, 196 and 49 centroid points are selected in the next 3 generations respectively. Figures 2.20, 2.24 and 2.28 show the changing of the convex hull after each subdivision. Figures 2.21, 2.25 and 2.29 show the distribution of the points in each generation. Figures 2.18, 2.22, 2.26, 2.30 show the evolution of the basins of attraction during the subdivision process.

In Figure 2.18 there are 102 points and the convex hull was divided into 195 triangles. Moreover, 122 triangles belong to the first state (filled in red colour), while 73 triangles belong to the second state (filled in blue colour). As compared with Figure 2.12, the major geometrical and topological features of the basins of attraction in Figure 2.18 are captured. It should be noted that the boundary or separatrix between the two basins of attraction is not smooth. In Figure 2.22, there are 235 points and the convex hull is divided into 461 triangles and 282 triangles belong to the first state (filled in red colour) while 179 triangles belong to the second state (filled in blue colour). In Figure 2.26, there are 431 points and the convex hull is divided into 853 triangles. 480 triangles belong to the first state (filled in red colour) and 373 triangles belong to the second state (filled in blue colour). In Figure 2.30, there are 480 points and the convex hull is divided into 951 triangles in which 532 triangles belong to the first state (filled in red colour) and 419 triangles belong to the second state (filled in blue colour). By comparing Figures 2.18, 2.22, 2.26, 2.30, major geometrical and topological features of the basins of attraction are easily observed across the evolution of the basins of attraction identification. The number of points near the boundary between the two states increases. The triangles which are farther away from the boundary do not change much but the triangles which are near to the boundary are divided into small triangles. Therefore, from Figures 2.18, 2.22, 2.26, 2.30, the boundary between two states becomes smoother and smoother through the subdivision process.

Figures 2.19, 2.23, 2.27, 2.31 show the basins of attraction estimated by the brute force algorithm with the resolutions 10*10, 15*15, 21*21 and 22*22, which are in almost the same resolution as those in Figures 2.18, 2.22, 2.26, 2.30 respectively. The proposed method is able to identify the major geometrical and topological features of basins of attraction at low resolution.

Moreover, a comparison will be made between the basins of attraction estimated by the proposed randomised triangular subdivision method and the brute force algorithm at different resolutions. Firstly, the basins of attraction estimated by the randomised triangular subdivision method provide more details about the features of the basins of attraction. Comparison of Figure 2.18 with Figure 2.19 and Figure 2.22 with Figure 2.23 show that they have a similar number of initial conditions. However, in the right part of Figures 2.18 and 2.19 the blue region is divided into two parts by the red region but in Figures 2.22 and 2.23 the red region is ignored.

Secondly, comparisons of Figure 2.26 with Figure 2.27 and Figure 2.30 with Figure 2.31 respectively show that they have a similar number of initial conditions but the basins of attraction estimated by the randomised triangular subdivision gives a smoother boundary between the two states. This is because the randomised triangular subdivision method can be adaptive to the geometrical and topological features of the basins of attraction so that more initial conditions are used to estimate the boundary between the two states. The yellow dashed block shows in Figures 2.30 and 2.31 show that, the basins of attraction estimated by the randomised triangular subdivision gives more details than the traditional brute force method.

Thirdly, Tables 2.1 and 2.2 show the similarity between Figures 2.18, 2.19, 2.22, 2.23, 2.26, 2.27, 2.30, 2.31 and Figure 2.12 and computation time for basins of attraction estimated using MATLAB running on a PC with quad core 3.60 GHz CPU and 32 G RAM. According to the tables, the similarity of the basins of attraction estimated by the randomised triangular subdivision method is, in general, higher than that given by the brute force algorithm. Moreover, when the number of initial conditions increases, the similarity of the basins of attraction estimated by the two methods also increases, but the increase speed is reduced. Moreover, there are fewer differences between the basins of attraction produced by these two methods. In addition, the computation time of the proposed method is slightly higher than that of the traditional brute force method at similar resolution. The proposed method is able to produce estimates of basins of attraction with higher accuracy and smoothness in sacrifice of slightly longer computation time.

Table 2.1 Similarity and time costs of the basins of attraction estimated by the random triangular subdivision

Number of initial conditions	Similarity	Time Costs (seconds)
102	90.07%	40.70
235	91.59%	105.23
431	93.73%	169.28
480	93.92%	180.45

Table 2.2 Similarity and time costs of the basins of attraction estimated by the brute force algorithm

Number of initial conditions	Similarity	Time Costs (seconds)
10*10 (100)	83.76%	35.26
15*15 (225)	89.14%	91.39
21*21 (441)	91.11%	133.78
22*22 (484)	91.14%	139.97

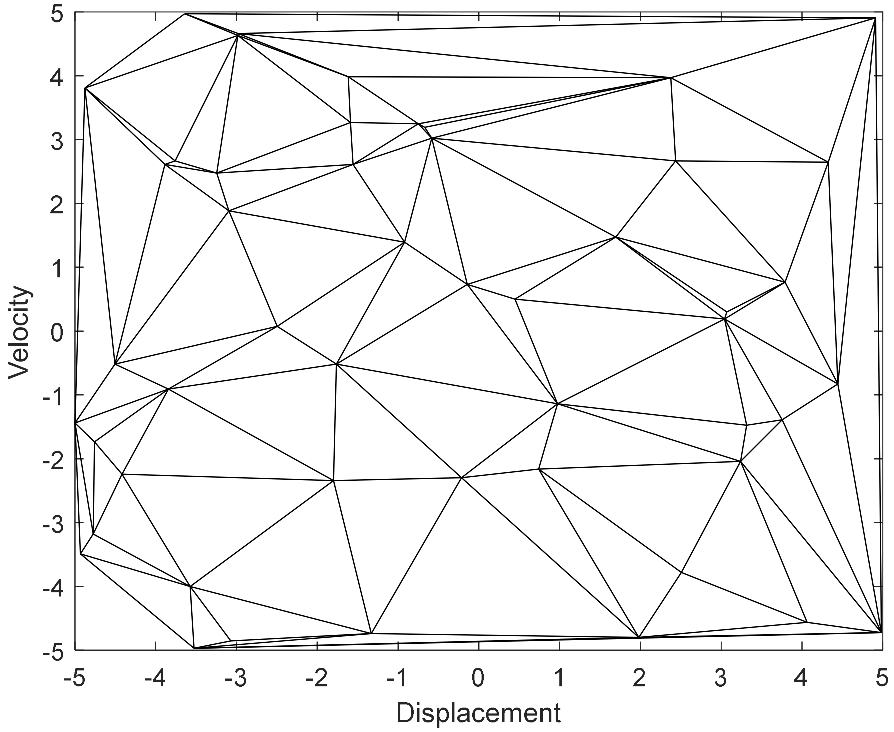


Figure 2.14 Triangulating the convex by using Delaunay Triangulation based on 50 initial conditions

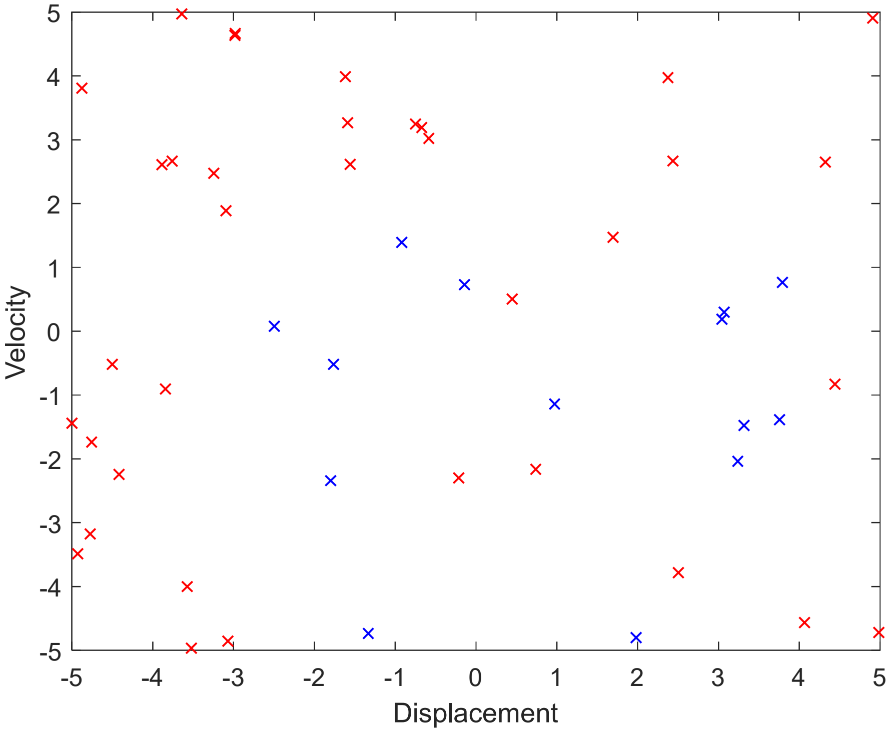


Figure 2.15 Random sample of the initial conditions. The colour of the points shows the initial condition converge to which state.

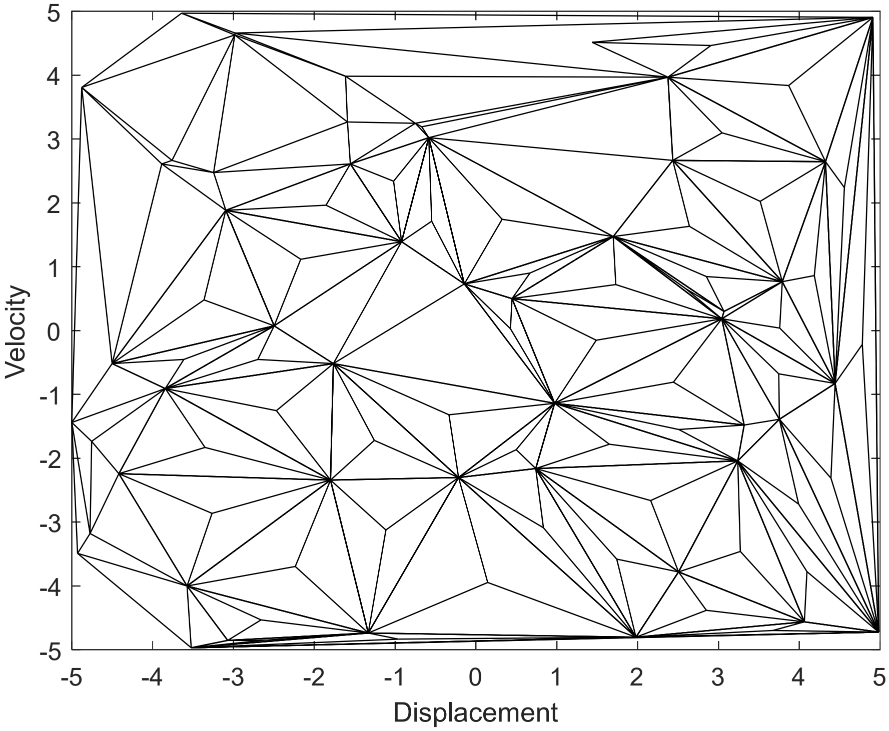


Figure 2.16 Triangles in the convex after first subdivision

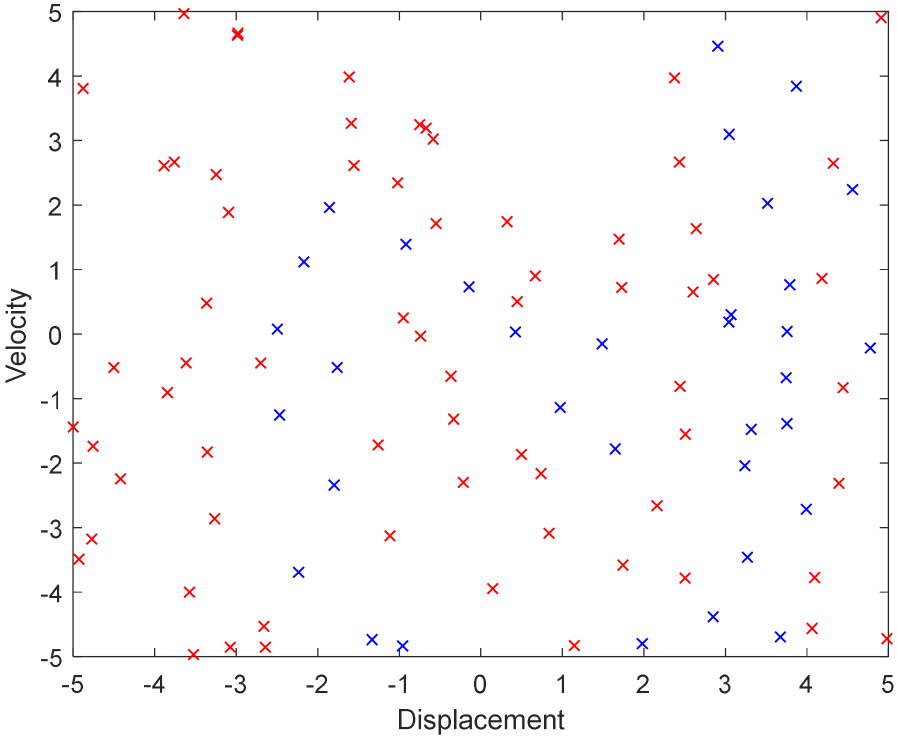


Figure 2.17 Distribution of the 102 initial conditions in the second generation. The colour of the points shows the initial condition converge to which state.

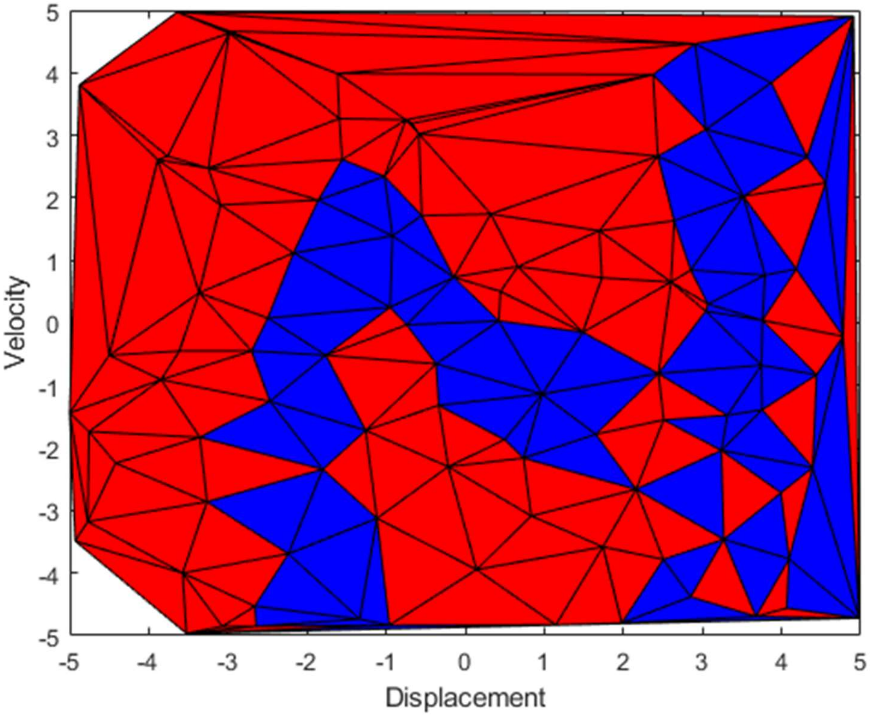


Figure 2.18 Basins of attraction estimated using random triangular subdivision with 102 initial conditions in the second generation

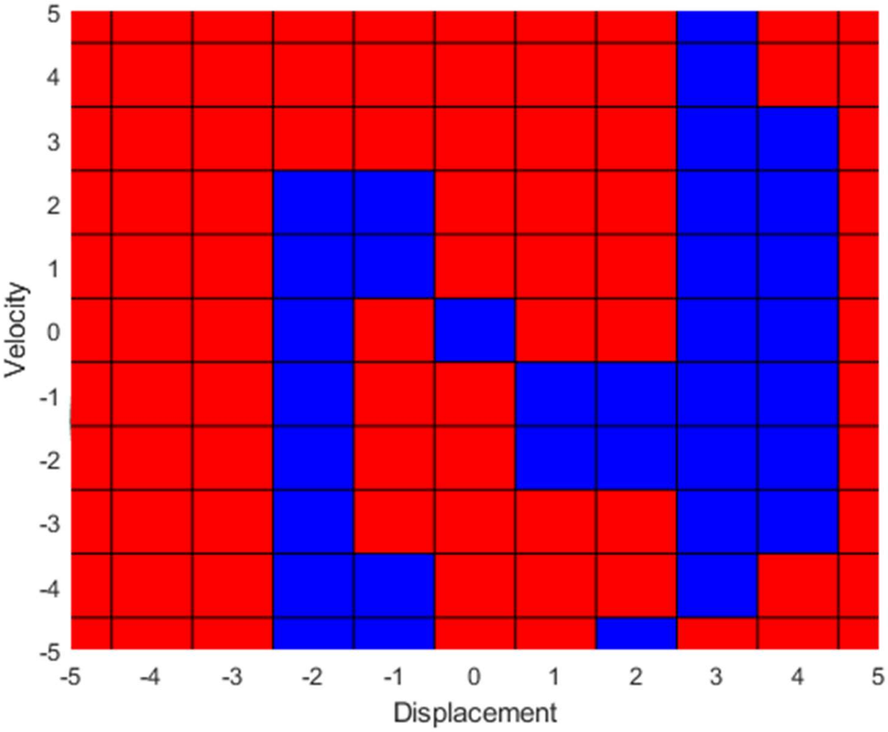


*Figure 2.19 Basins of attraction estimated using the brute force algorithm with resolution 10*10 (100 initial conditions)*

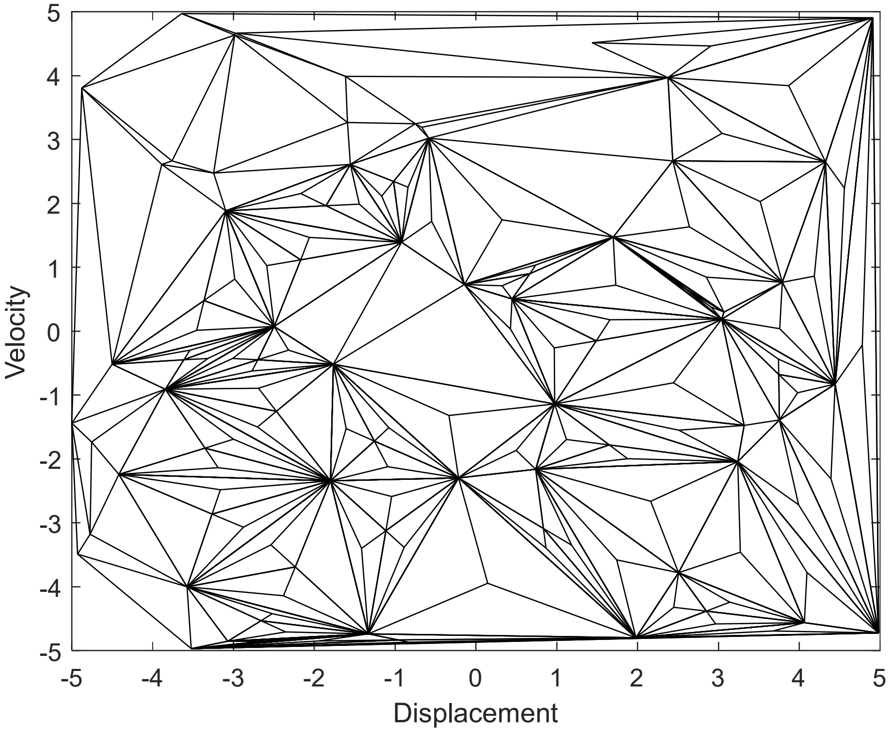


Figure 2.20 Triangles in the convex after second subdivision

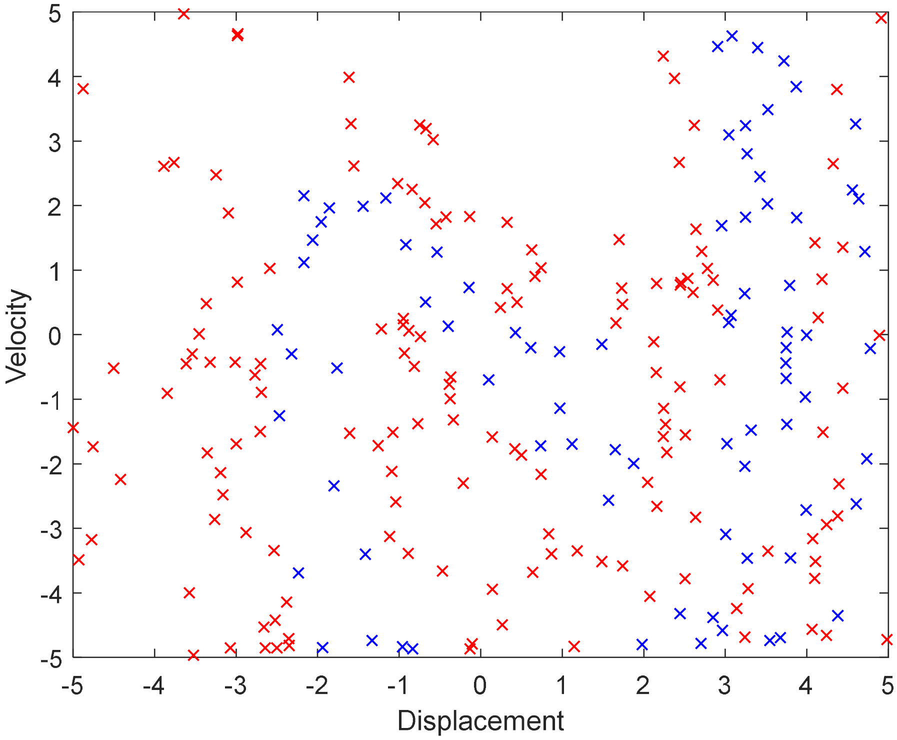


Figure 2.21 Distribution of the 235 initial conditions in the third generation. The colour of the points shows the initial condition converge to which state.

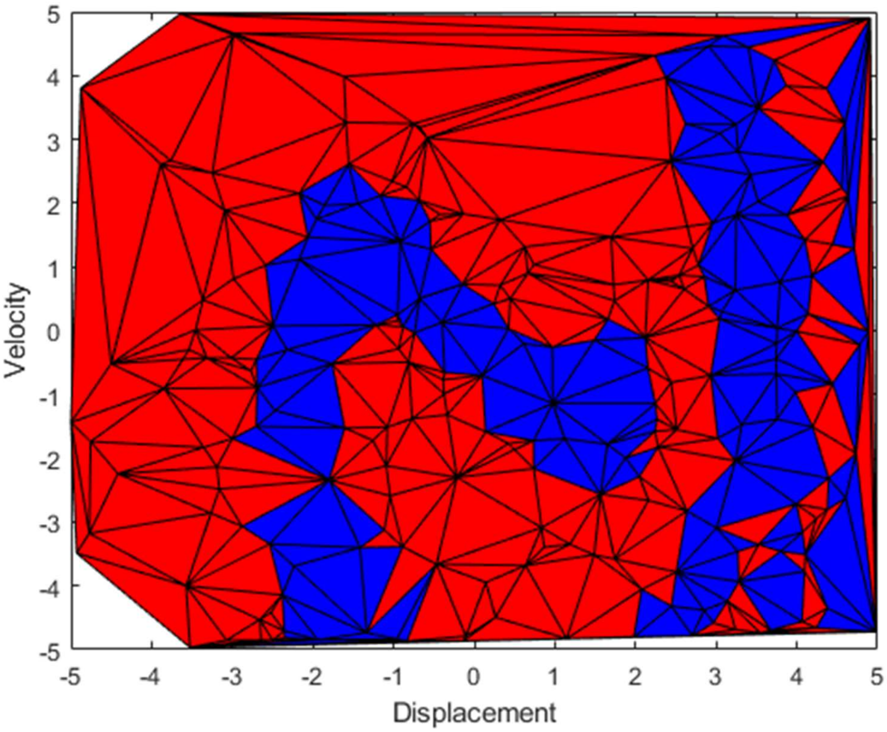


Figure 2.22 Basins of attraction estimated using random triangular subdivision with 235 initial conditions in the third generation

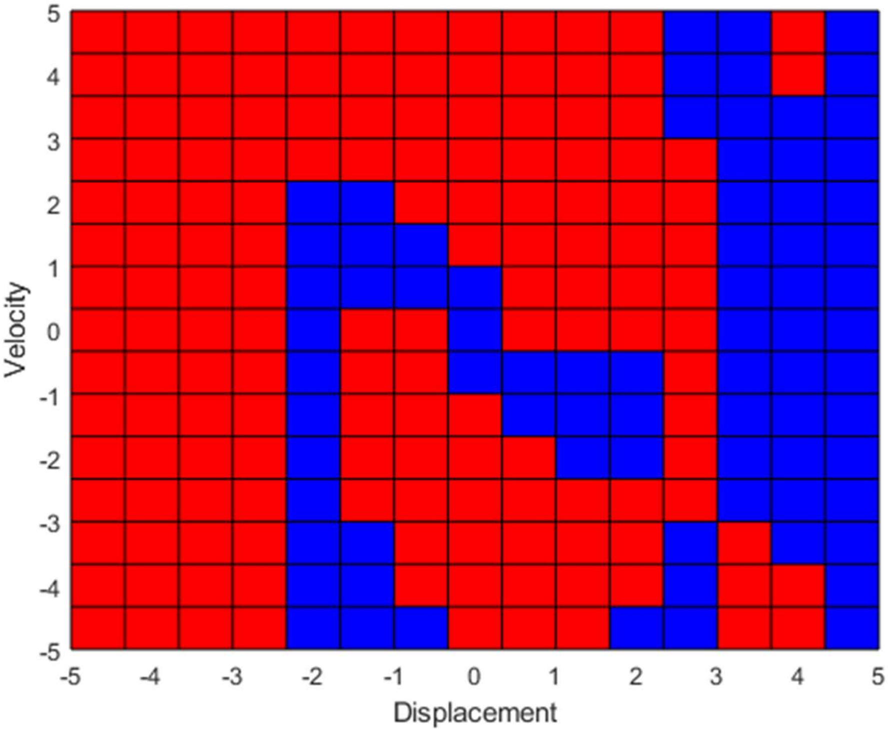


*Figure 2.23 Basins of attraction estimated using the brute force algorithm with resolution 15*15 (225 initial conditions)*

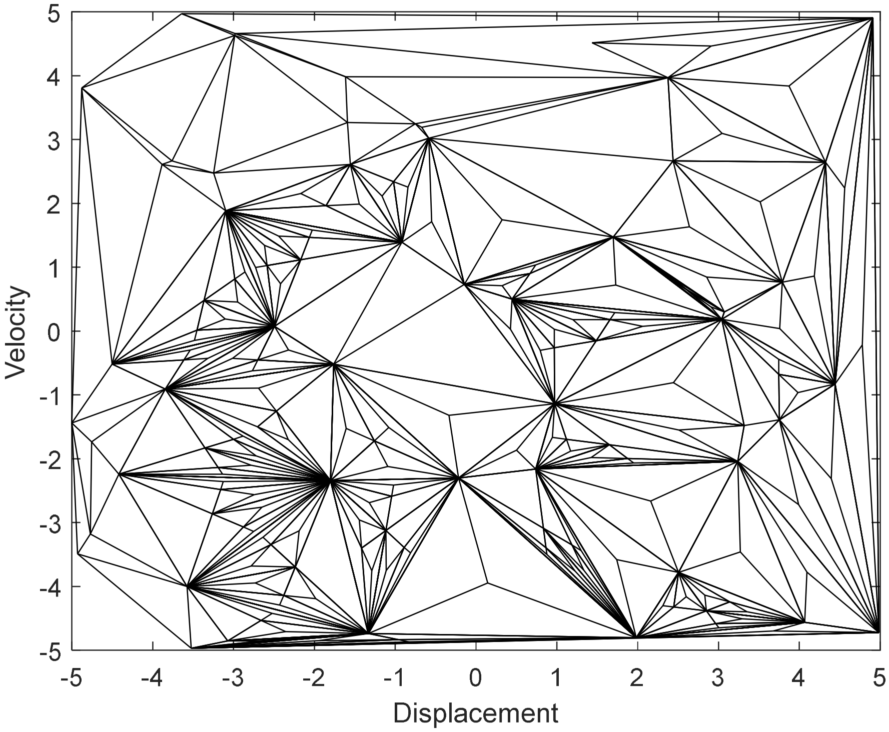


Figure 2.24 Triangles in the convex after third subdivision

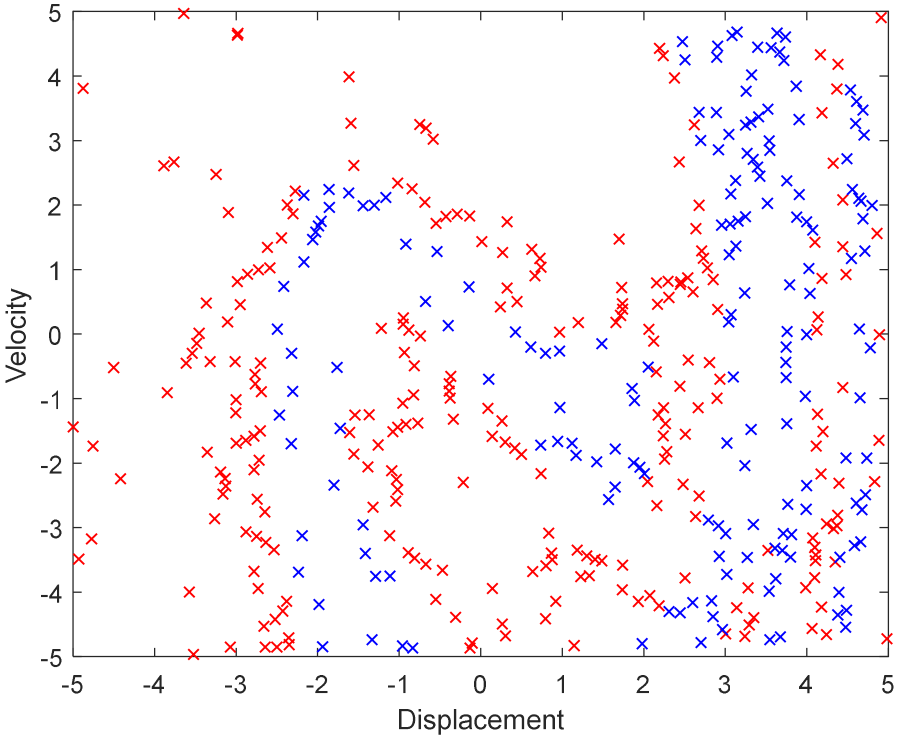


Figure 2.25 Distribution of the 431 initial conditions in the fourth generation. The colour of the points shows the initial condition converge to which state.

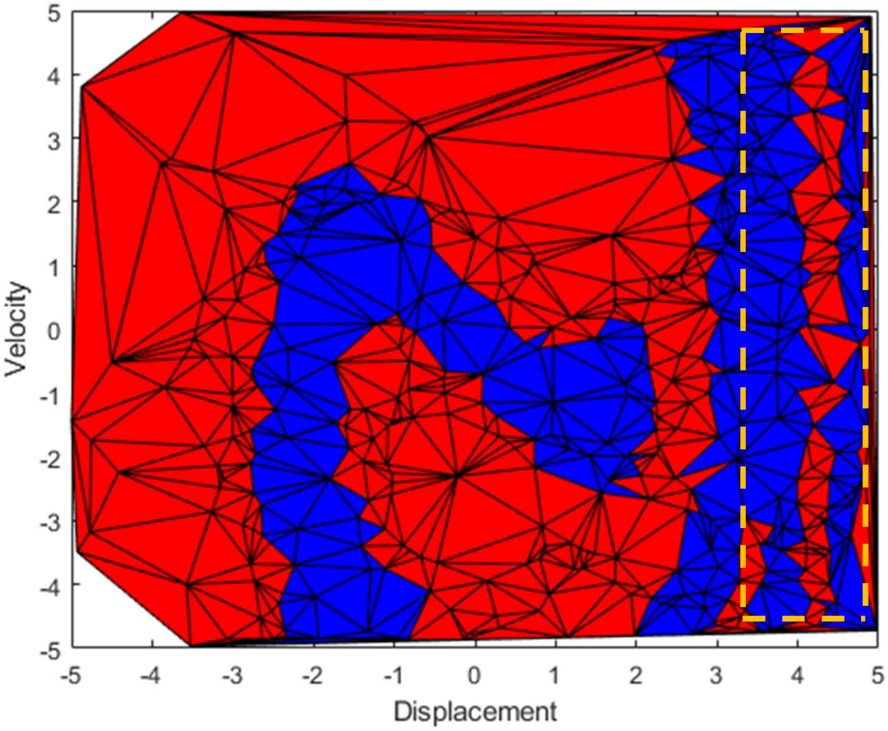


Figure 2.26 Basins of attraction estimated using the random triangular subdivision with 431 initial conditions in the fourth generation

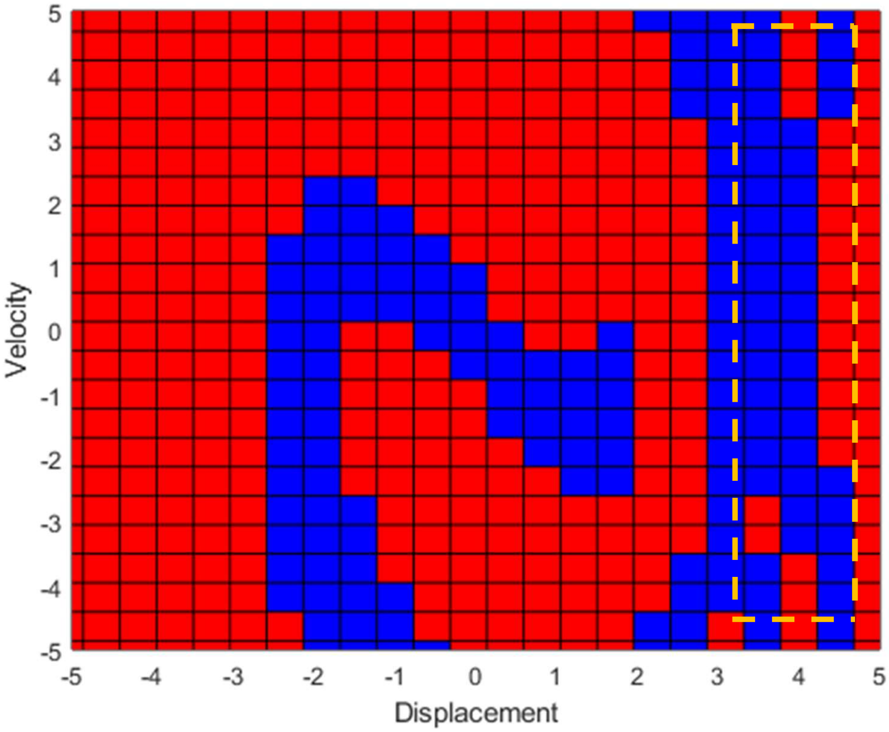


*Figure 2.27 Basins of attraction estimated using the brute force algorithm with resolution 21*21 (441 initial conditions)*

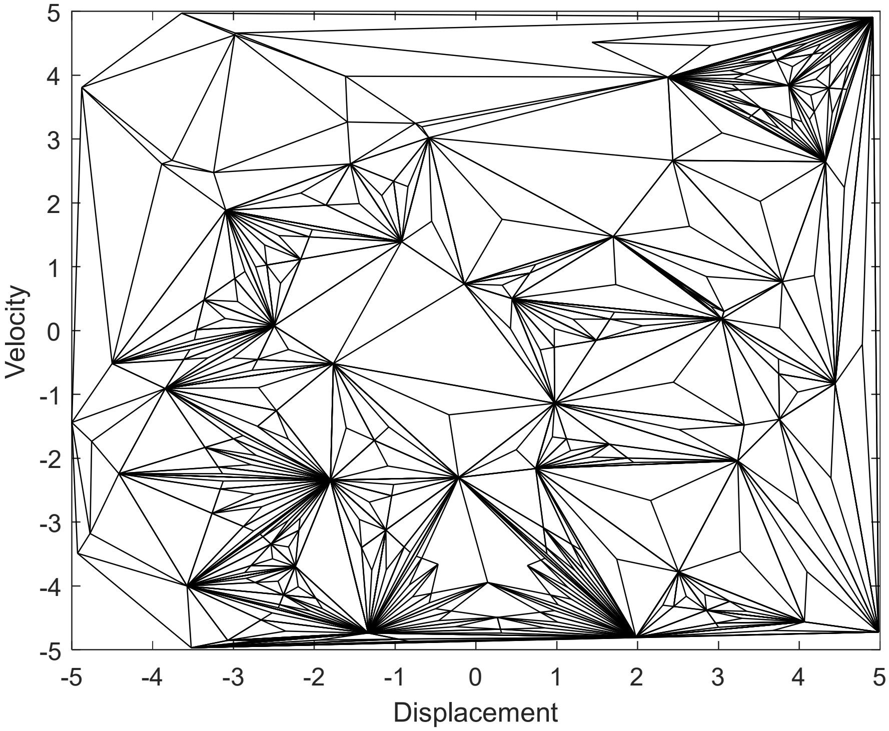


Figure 2.28 Triangles in the convex after fourth subdivision

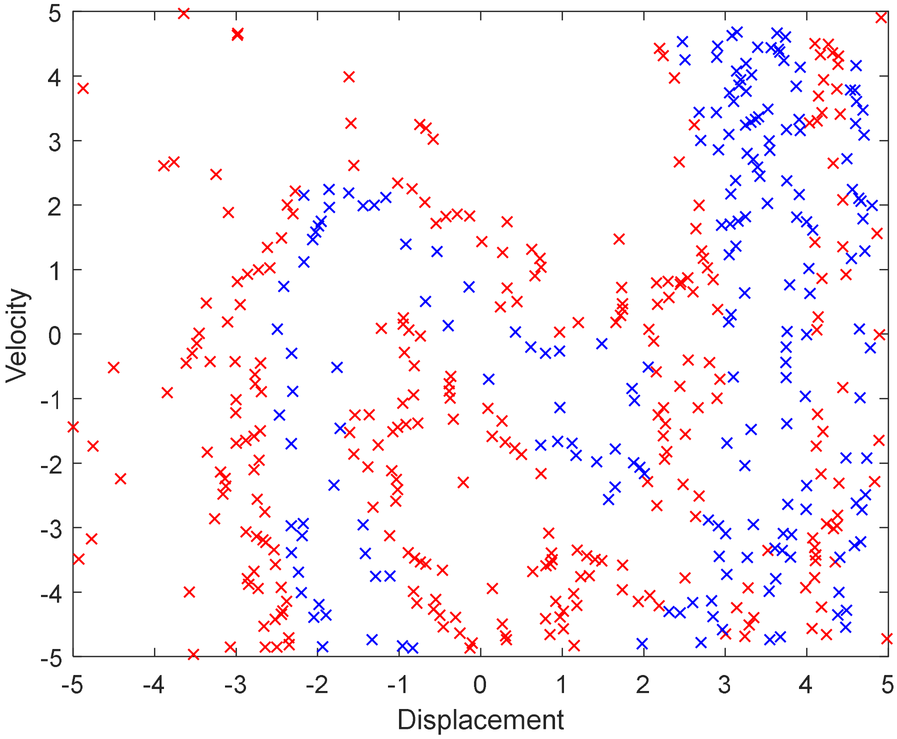


Figure 2.29 Distribution of the 480 initial conditions in the fifth generation. The colour of the points shows the initial condition converge to which state.

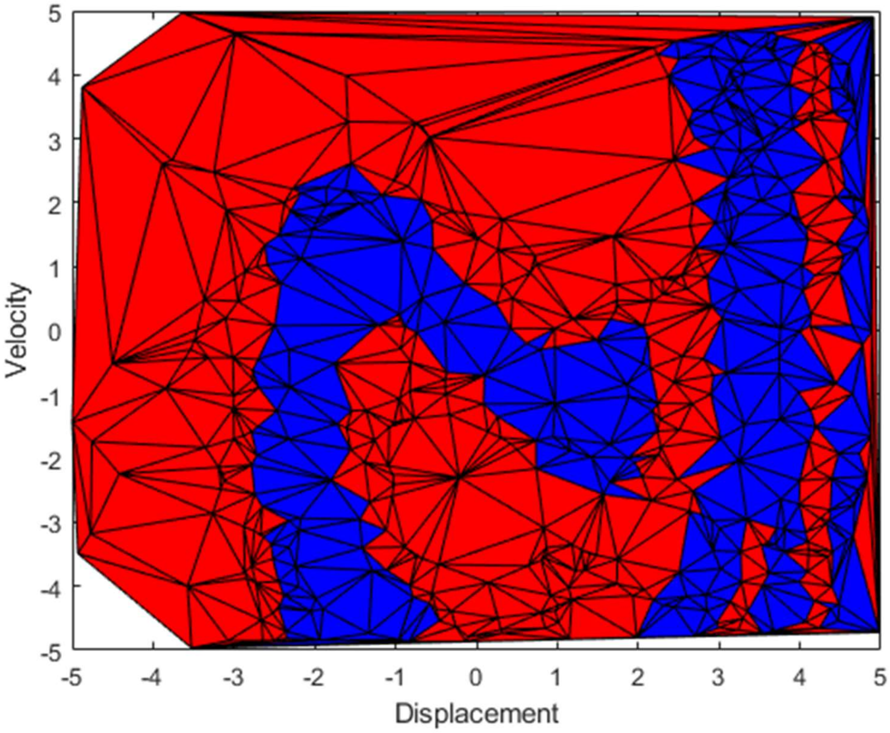


Figure 2.30 Basins of attraction estimated using random triangular subdivision with 480 initial conditions in the fifth generation

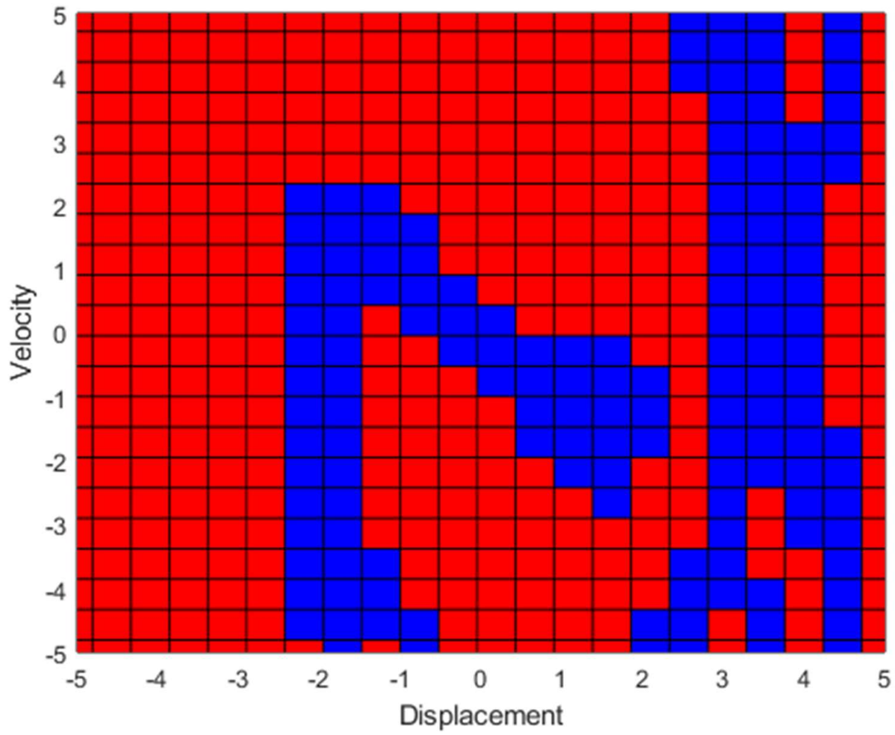


*Figure 2.31 Basins of attraction estimated using the brute force algorithm with resolution 22*22 (484 initial conditions)*

2.5.2 Soft impact oscillator with Two Coexist Stable State

In order to investigate the robustness of the proposed method for different systems, the basins of attraction of the soft impact oscillator is estimated. The system parameters are selected as follows, $\xi = 0.01,\ \beta = 29,\ g = 1.26,\ \Gamma = 1.0385$ and $\omega = 0.686$. The region of interest in the initial condition space is selected from -4 to 4 for the x-axis (displacement) and from -6 to 6 for the y-axis (velocity). The basins of attractions at high resolution (501*501) is provided in Figure 2.32. There are two co-existing stable states in the system with this given set of parameters [66]. Comparing with the investigation in Section 2.5.1, these basins of attraction is more complex than those of the Duffing oscillator.

In order to estimate the basins of attraction of the soft impact oscillator under this set of parameters by the proposed method, 50 initial conditions are randomly selected as the first generation in this study. The distribution of these 50 initial conditions and the state of each initial condition are shown in

Figure 2.33. 34 initial conditions converge to the first state (red points) and the other 16 initial conditions converge to the second state (blue points). The area of the convex hull which is built based on these initial conditions is 91.3936 which is 95.20% of the selected region. Therefore, the threshold values of the triangular area p and q are 0.0914 and 0.0457 which are 0.1% and 0.05% of the area of the convex hull respectively.

At first the convex hull is divided into 88 triangles based on these 50 initial conditions which is shown in 2.34. After one iteration, 72 new points are added to the set of the initial conditions and the convex hull is divided into 232 triangles. 158 triangles belong to the first state (filled in red colour) and 74 triangles belong to the second state (filled in blue colour). Figure 2.35 shows the basins of attraction estimated based on these 122 initial conditions. Comparing with Figure 2.32, the geometrical and topological features of the boundary between the two stable states are observed. However, the boundary is not smooth. Moreover, for the complex part of the basin of attraction (in dashed box in Figure 2.35), Figure 2.35 does not show much details between these two states. There are three more iterations before the stopping conditions are met. During the second iteration, 176 points are added to the set of initial conditions and the convex hull is divided into 584 triangles. 373 triangles belong to the first state and 211 triangles belong to the second state. During the third iteration, 320 points are added to the set of initial conditions and the convex hull is divided into 1224 triangles. 733 triangles belong to the first state and 491 triangles belong to the second state. In the last iteration, 275 points are added to the set of initial conditions and the convex hull is divided into 1774 triangles. 1069 triangles belong to the first state and 705 triangles belong to the second state. The evolution of the basins of attraction is shown from Figures 2.35, 2.37, 2.39, to 2.41. The geometrical and topological features of the basins of attraction of the soft impact oscillator can be identified easily with small number of initial conditions. When the number of initial conditions increases, more detail in the basins of attraction is revealed and the boundary between the two states becomes smoother.

To compare with the basins of attraction estimated by the traditional brute force method, four levels of resolution were studied namely 11*11, 17*17, 25*25 and 30*30 which are similar to the numbers of initial conditions in Figures 2.35, 2.37, 2.39, 2.41 respectively. These basins of attraction are shown in Figures 2.36, 2.38, 2.40, 2.42 respectively. Comparing the basins of attraction estimated by the two methods, at low resolution such as Figure 2.35 and Figure 2.36, the basins of attraction estimated by the proposed method show more details than that by the traditional method. At high resolution such as Figures 2.41 and 2.42, similar to the low resolution cases, the basins of attraction estimated by the proposed state shows more detail than that by the traditional method. Moreover, the boundary or separatrix between the two states in Figure 2.41 is smoother than that in Figure 2.42. The similarities between Figures 2.35-2.42 and Figure 2.32 and time costs for basins of attraction estimated are shown in Tables 2.3 and 2.4. From Tables 2.3 and 2.4, similar to the Duffing oscillator case in Section 2.5.1, the similarities of the basin of attraction estimated by the proposed method are higher than that by the traditional method under similar numbers of the initial conditions. The higher the number of initial conditions, the higher the similarity but the increase rate of the similarity decreases. Furthermore, the difference of the similarity between these two methods is reduced. Since the basins of attraction of soft impact oscillator is more complex than that of Duffing Oscillator, the similarity in this study is lower than those in Duffing Oscillator with similar number of initial conditions. In addition, the proposed method is able to produce estimates of basins of attraction with higher accuracy and smoothness in sacrifice of slightly longer computation time.

Table 2.3 Similarity and time costs of the basins of attraction estimated by the randomised triangular subdivision method

Number of initial conditions	Similarity	Time Costs (seconds)
122	74.50%	134.19
298	77.11%	319.46
618	79.59%	696.91
893	80.55%	996.17

Table 2.4 Similarity and time costs of the basins of attraction estimated by the brute force algorithm

Number of initial conditions	Similarity	Time Costs(seconds)
11*11 (121)	70.78%	116.66
17*17 (289)	73.78%	271.84
25*25 (625)	76.50%	601.135
30*30 (900)	79.16%	847.366

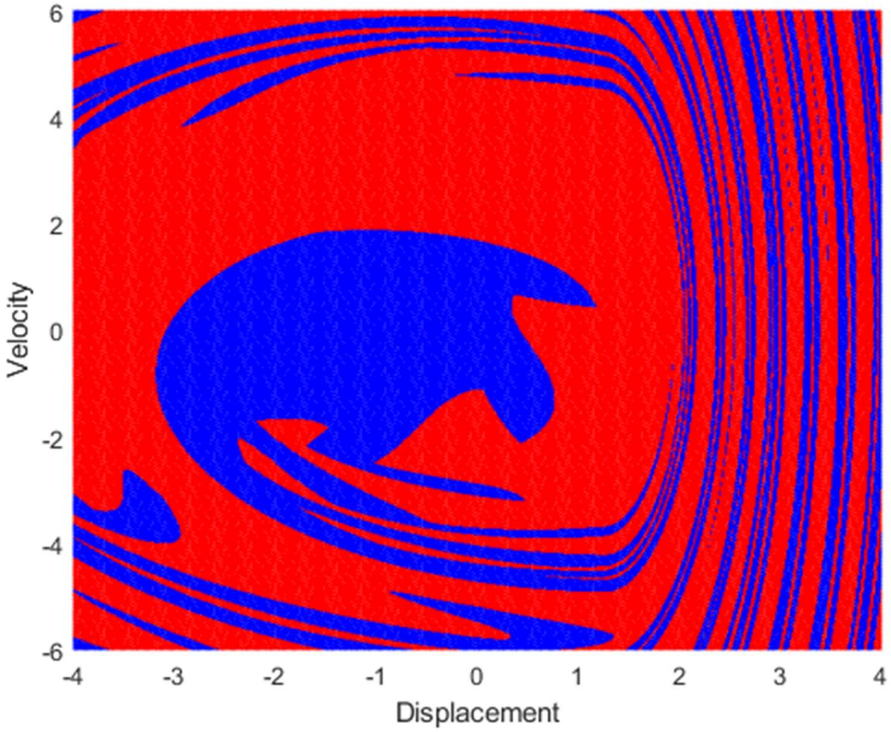


Figure 2.32 Basins of attraction of the soft impact oscillator with the system parameters: $\xi = 0.01, \beta = 29, g = 1.26,\ \Gamma = 1.0385$ *and* $\omega = 0.686$

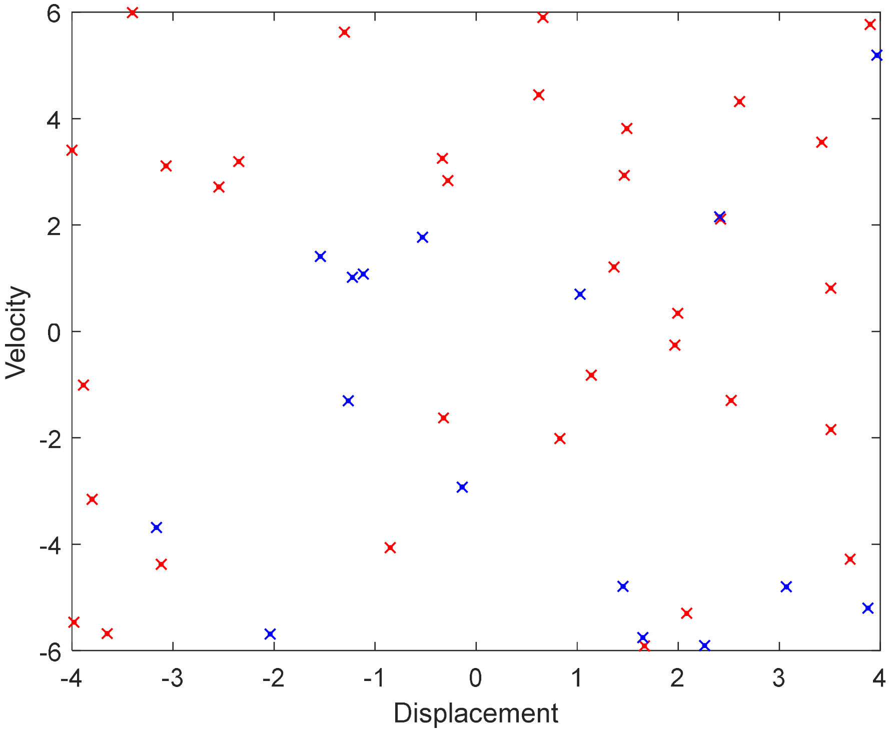


Figure 2.33 Random sample of the initial conditions. The colour of the points shows the initial condition converge to which state.

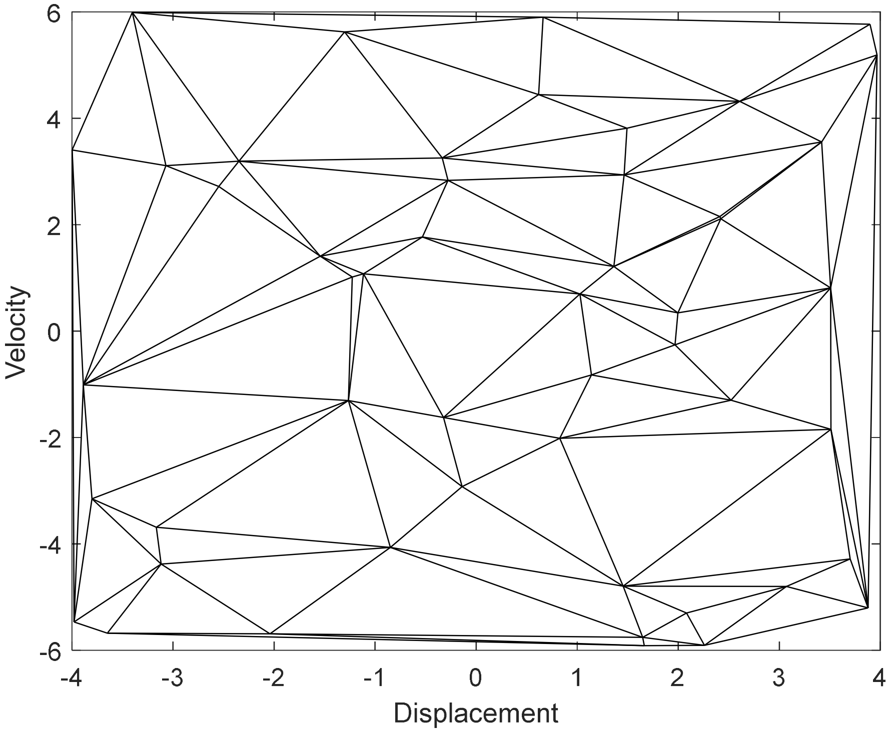


Figure 2.34 Triangulating the convex by using Delaunay Triangulation base on these 50 initial conditions

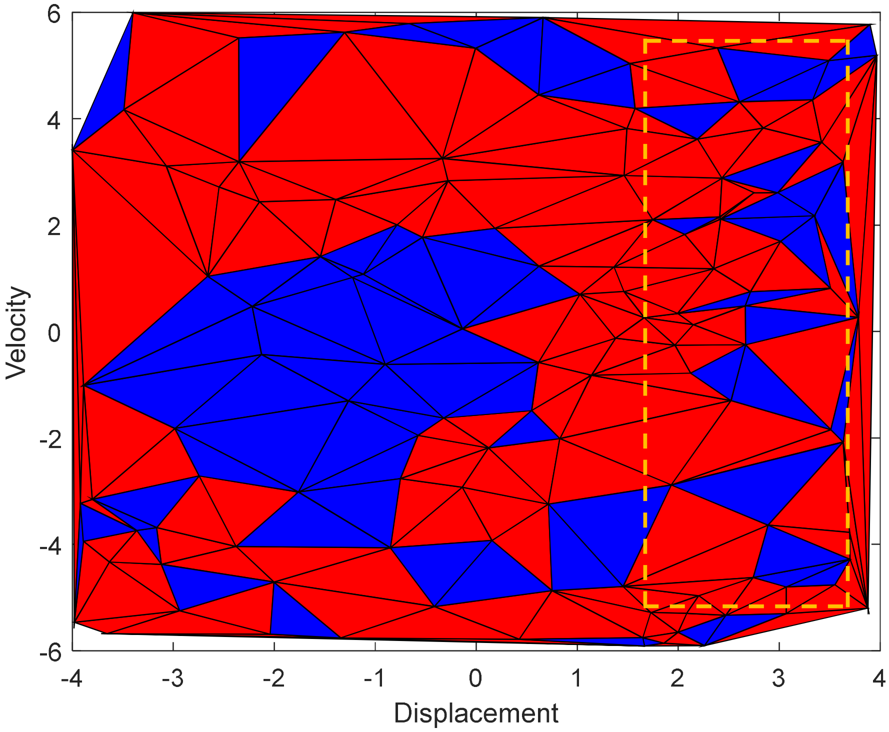


Figure 2.35 Basins of attraction estimated by the randomised triangular subdivision method with 122 initial conditions

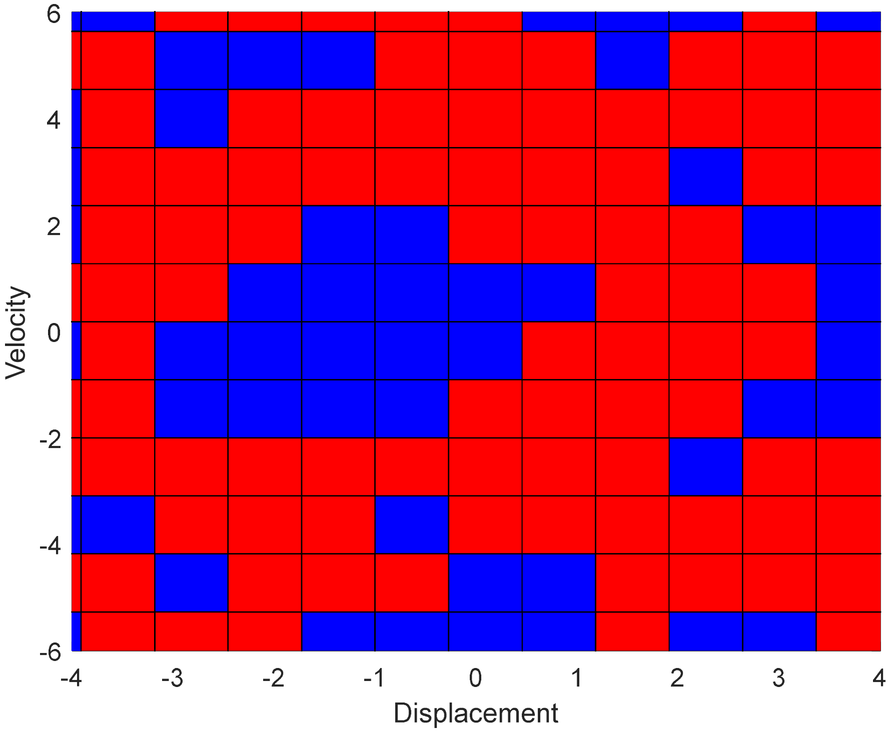


*Figure 2.36 Basins of attraction estimated by the brute force algorithm with resolution 11*11 (121 initial conditions)*

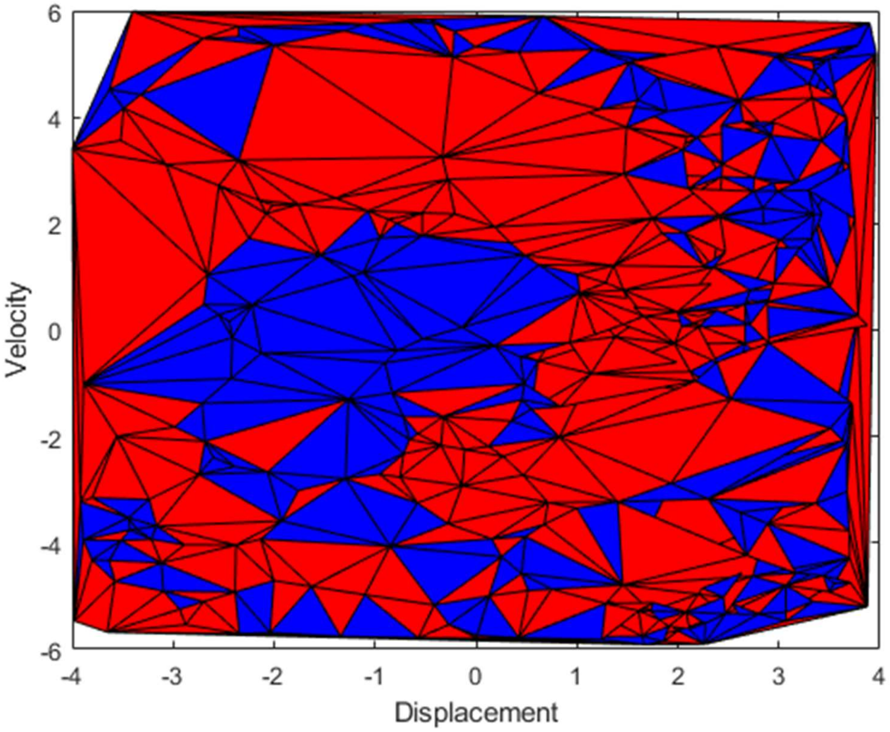


Figure 2.37 Basins of attraction estimated by the randomised triangular subdivision with 298 initial conditions

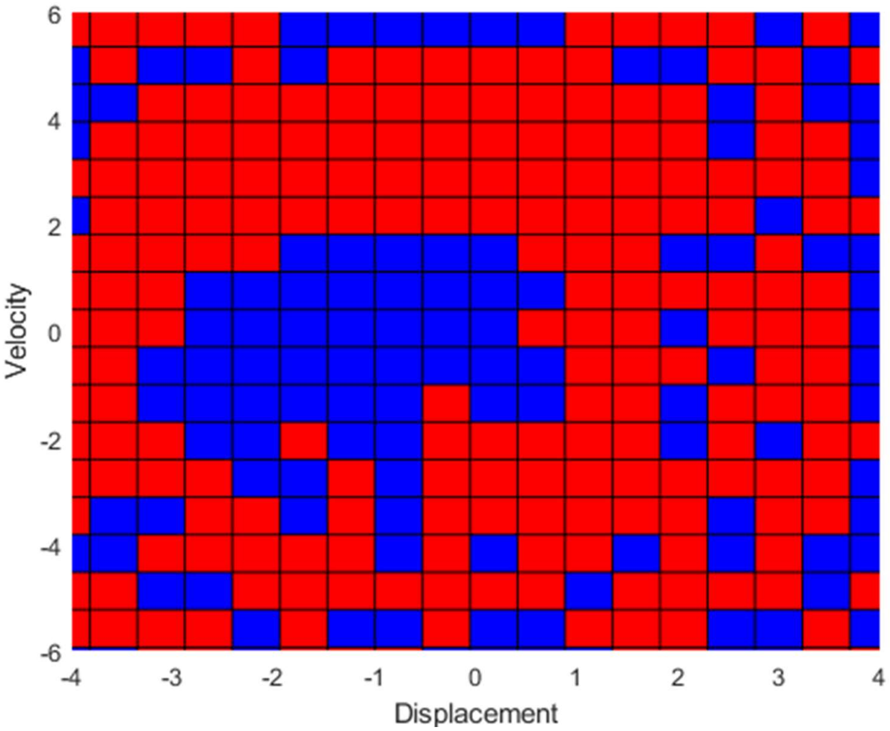


*Figure 2.38 Basins of attraction estimated by the brute force algorithm with resolution 17*17 (289 initial conditions)*

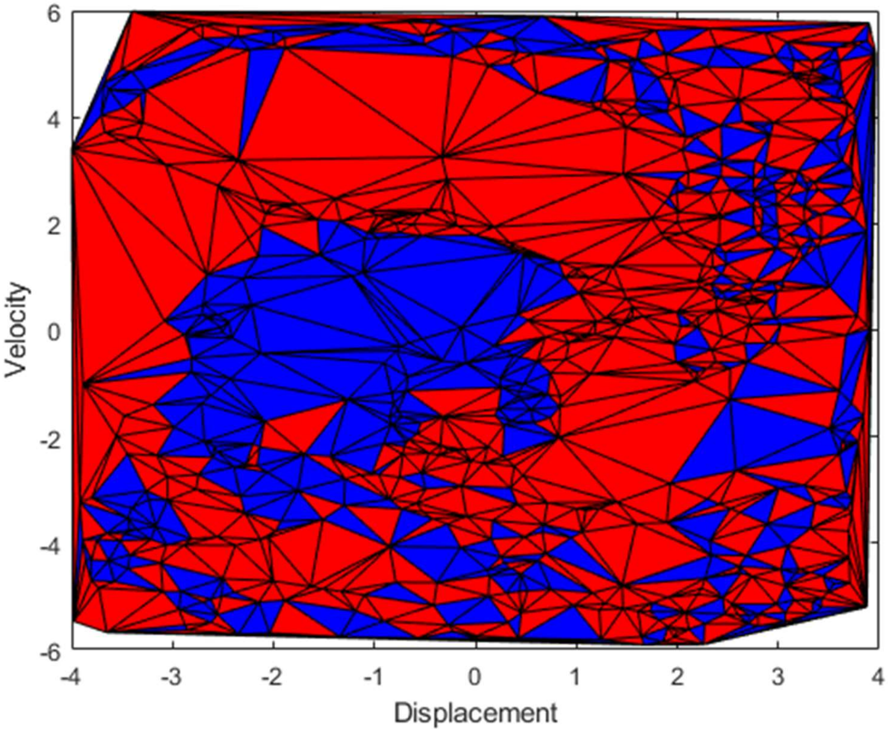


Figure 2.39 Basins of attraction estimated by the randomised triangular subdivision method with 618 initial conditions

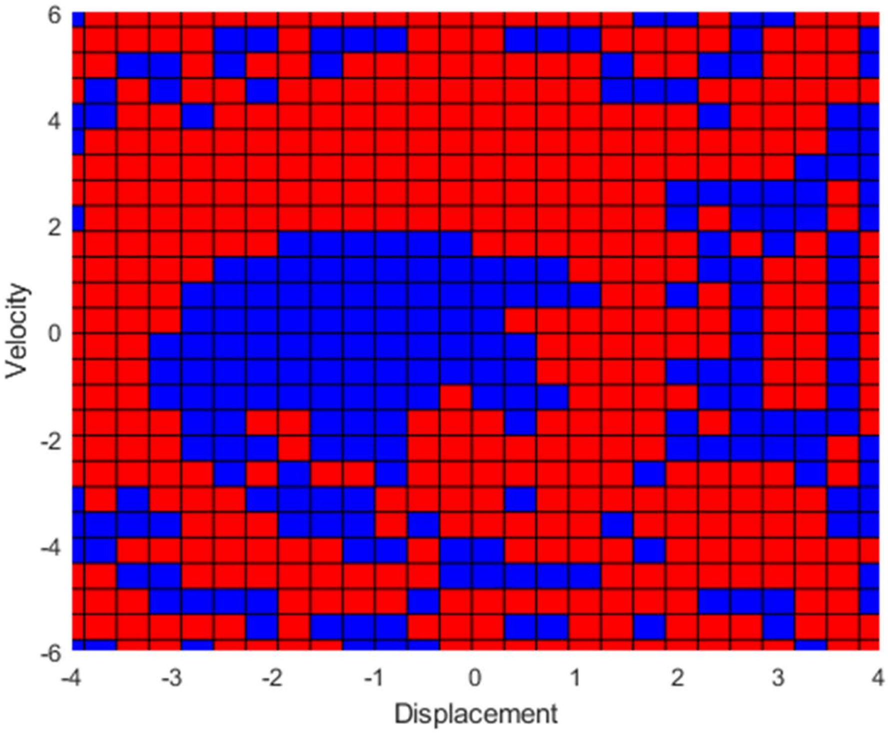


*Figure 2.40 Basins of attraction estimated by the brute force algorithm with resolution 25*25 (625 initial conditions)*

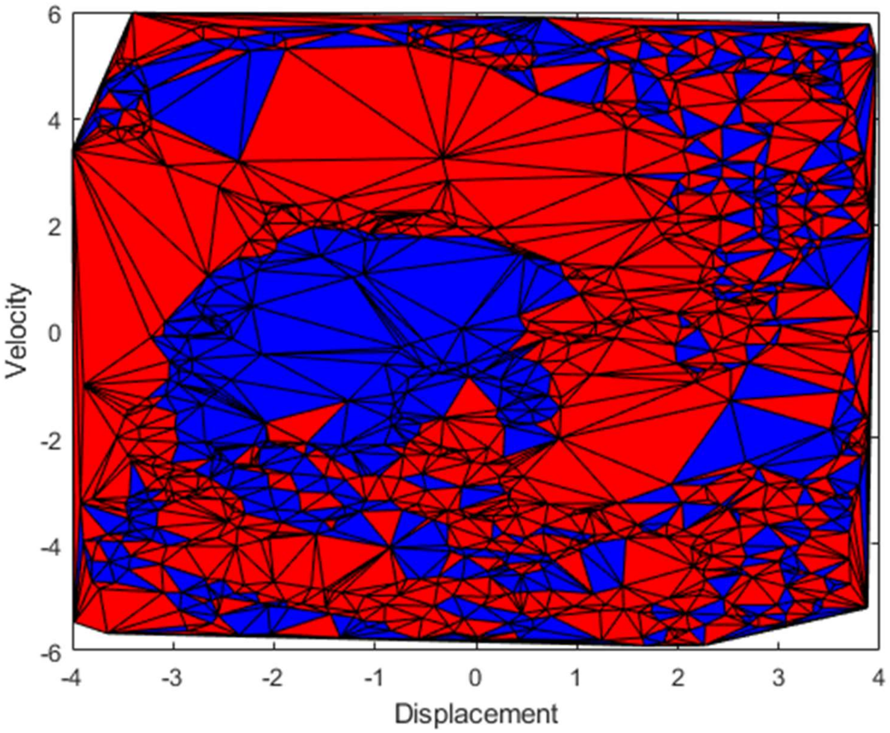


Figure 2.41 Basins of attraction estimated by the randomised triangular subdivision method with 893 initial conditions

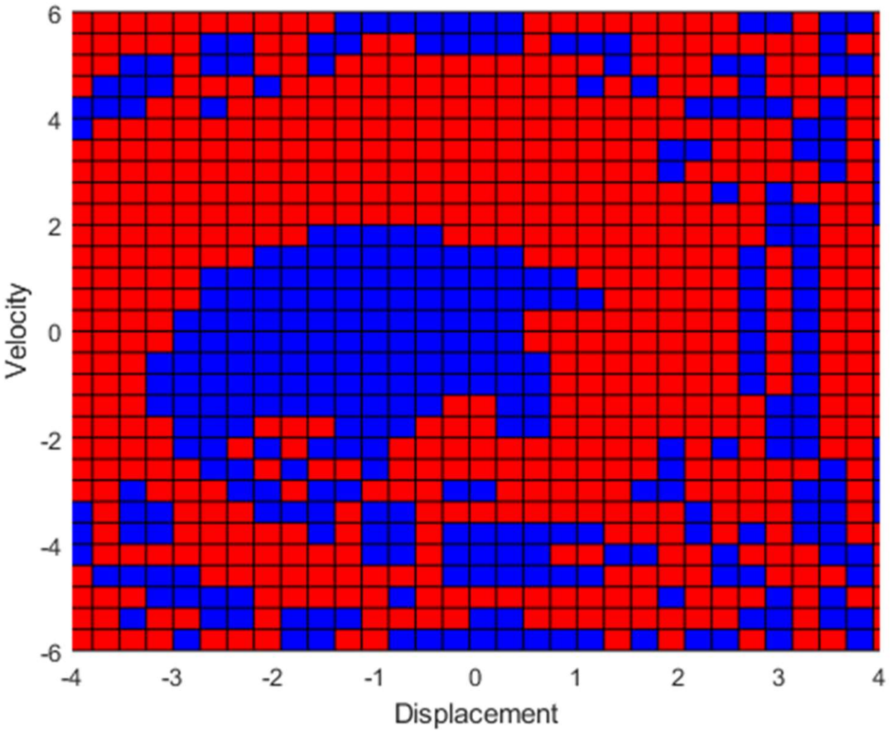


*Figure 2.42 Basins of attraction estimated by the brute force algorithm with resolution 30*30 (900 initial conditions)*

2.5.3 Soft impact oscillator with Three Stable State

Another set of system parameters of soft impact oscillator is selected for testing with $\xi = 0.01,\ \beta = 29,\ g\ = 1.26,\ \Gamma = 0.4547$ and $\omega = 0.806$. The region of interest in the initial condition space is selected from -2 to 2 for both the x-axis (displacement) and y-axis (velocity). The high resolution (501*501) basins of attraction are shown in Figure 2.43. Under this set of parameters, there are 3 co-existing stable states in the system [66]. Moreover, the basins of attraction are more complex than those in the last two cases. The basins of attraction of these three states are highly scattered and intermeshed.

50 initial conditions are randomly selected as the first generation in this region using the proposed method. Then the state of these 50 initial conditions are checked. 20 points converge to the first state (red points), 24 points converge to the second state (blue points) and 6 points converge to the third state (green points). Figure 2.44 shows the distribution of these points and the state of each initial condition. The area of the convex hull built by these 50 initial conditions is 15.2 which is 95% of the selected region. Therefore, the threshold values p and q of the area are 0.076 and 0.0076 which are 0.5% and 0.05% of the area of the convex hull respectively. Using the Delaunay Triangulation, the convex hull is divided into 90 triangles based on these 50 initial conditions. There are 4 iterations before the area of the triangle is smaller than 0.0076 or all three vertices of the triangle are in the same state. In the second generation, 88 points are added to the set of initial conditions. There are 138 initial conditions and the convex hull is divided into 266 triangles. 113 triangles belong to the first state (filled in red colour), 109 triangles belong to the second state (filled in blue colour) and 44 triangles belong to the third state (filled in green colour). Comparing with the high resolution basins of attraction cases, the geometrical and topological features of the first state (red region) and the second state (blue region) can be observed but the boundary between these two state is not smooth. Moreover, since the first state and the third state (green region) are intermeshed, these basins of attraction do not show the detail of boundary between the first state and third state.

Figure 2.46 shows the basins of attractions of the second generation. In total, five iterations were run before stopping conditions were met. Figures 2.48, 2.50, 2.55 show the basins of attraction of the last three generations. In the third iteration, there are 381 initial conditions and the convex hull is divided into 752 triangles. 317 triangles belong to the first state, 308 triangles belong to the second state and 127 triangles belong to the third state. In the fourth iteration 860 initial conditions are added and the convex hull is divided into 1710 triangles. 776 triangles belong to first state, 690 triangles belong to the second state and 244 triangles belong to the third state. In the fifth iteration, there are 1493 initial conditions and the convex hull is divided into 2976 triangles. 1359 triangles belong to the first state, 1178 triangles belong to the second state and 439 triangles belong to the third state. By comparing Figures 2.46, 2.48, 2.50, and 2.52, the boundary between the first state and the second state become smoother and there are more details between the first state and the third state. Moreover, the geometrical and topological features of each state become more similar to the features of each state in Figure 2.43.

Figure 2.47, 2.49, 2.51, and 2.53 show the basins of attraction estimated by the traditional method with the resolutions of 12*12, 20*20, 29*29 and 39*39 which are similar to the number of initial conditions in Figures 2.46, 2.48, 2.50, and 2.52 respectively. Comparing the basins of attraction estimated by both methods, at low resolutions such as Figure 2.48 and Figure 2.49, the basins of attraction estimated by the proposed method show more detail than the one estimated by the traditional method. When the resolution increases, the basins of attractions estimated by both methods provide more details. Tables 2.5 and 2.6 show time costs for basins of attraction estimated and the similarity between the basins of attraction built by the two methods and Figure 2.43. According to these two tables, like the last two cases presented in Sections 2.5.1 and 2.5.2, the basins of attraction estimated by the proposed mothed have a higher similarity than that by the traditional method. However, by comparing with the results in the last two subsections, the similarity is lower even more initial conditions are used in every iteration because the basins of attraction in this case is more complex than those of the other two systems. Similarly, the computation time of the proposed method is slightly higher than that of the traditional brute force method at similar resolution.

Table 2.5 Similarity and the time costs of the basins of attraction estimated by the random triangular subdivision

Number of initial conditions	Similarity	Time Costs (seconds)
138	50.65%	161.68
381	57.59%	424.86
860	61.99%	876.01
1493	65.02%	1578.10

Table 2.6 Similarity and the time costs of the basins of attraction estimated by the brute force algorithm

Number of initial conditions	Similarity	Time Costs (seconds)
12*12 (144)	47.08%	150.12
20*20 (400)	55.63%	387.54
29*29 (841)	60.80%	774.95
39*39 (1521)	64.09%	1407.07

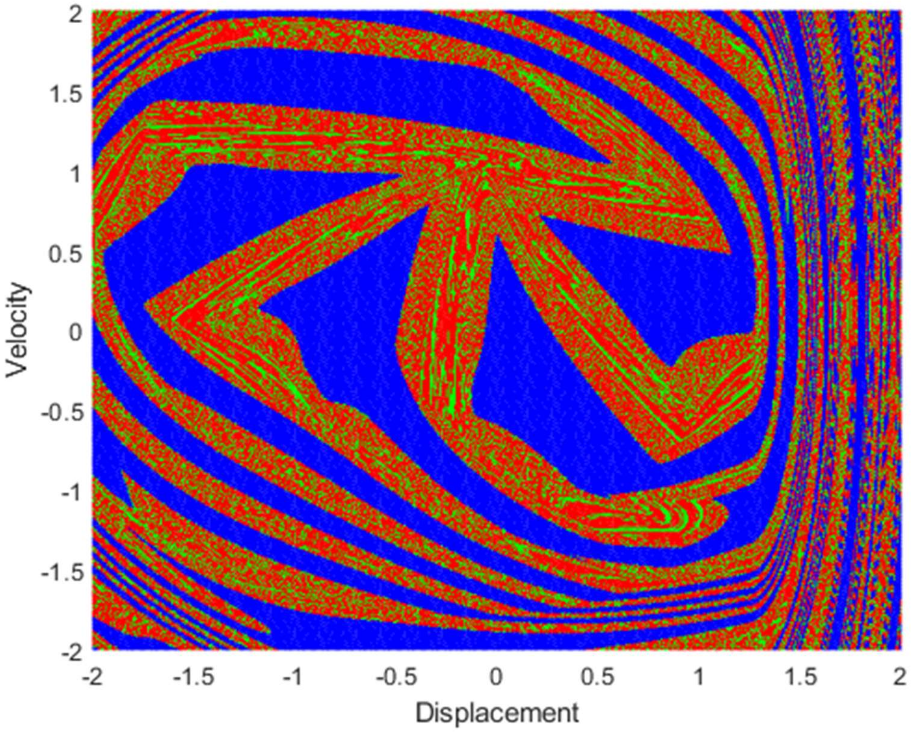


Figure 2.43 The basins of attraction of the soft impact oscillator with the system parameters: $\xi = 0.01, \beta = 29, g = 1.26, \ \Gamma = 0.4547$ *and* $\omega = 0.806$

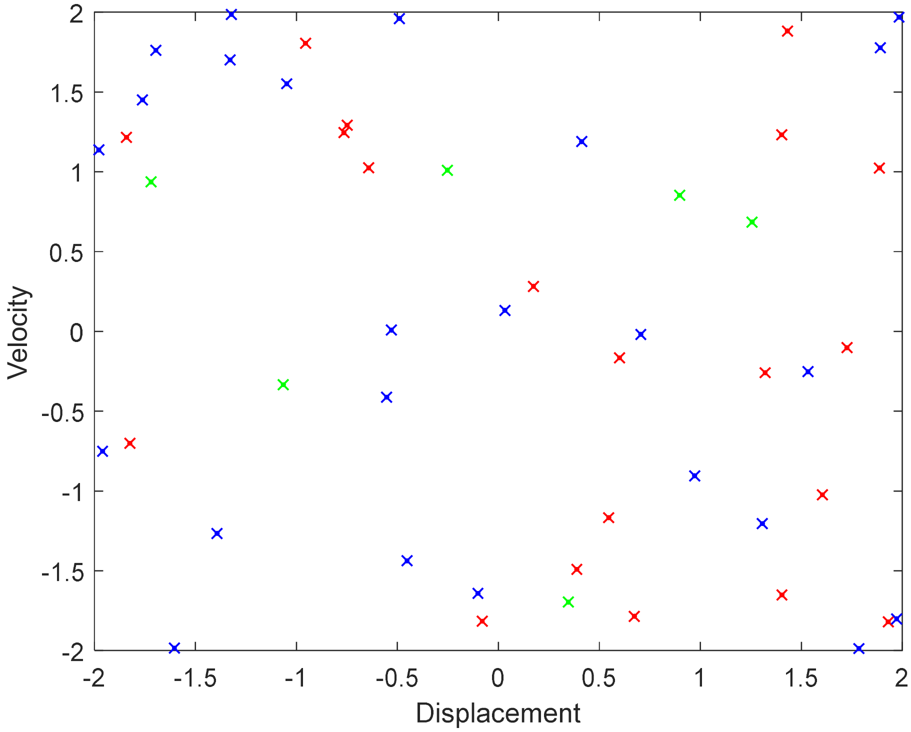


Figure 2.44 Random sample of the initial conditions. The colour of the points shows the initial condition coverage to which state.

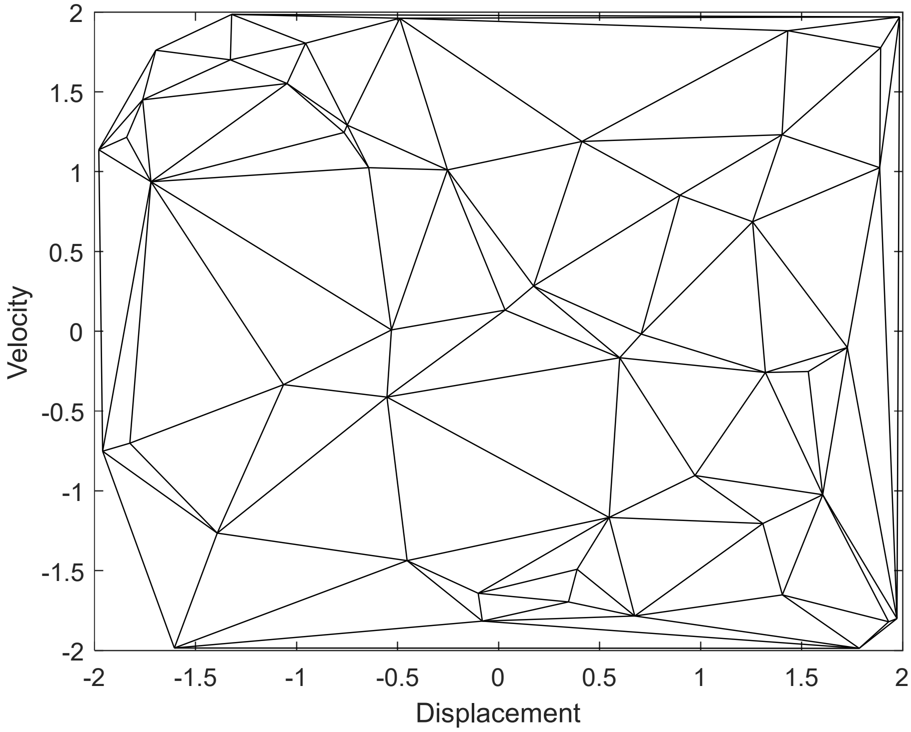


Figure 2.45 Triangulating the convex by using Delaunay Triangulation base on these 50 initial conditions

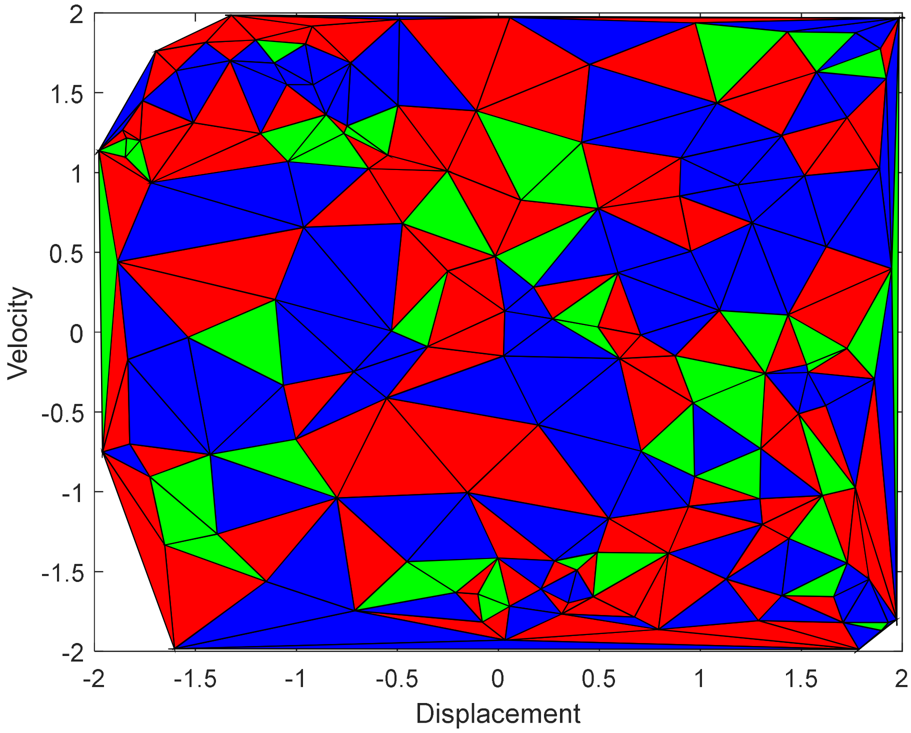


Figure 2.46 Basins of attraction estimated by the random triangular subdivision with 138 initial conditions

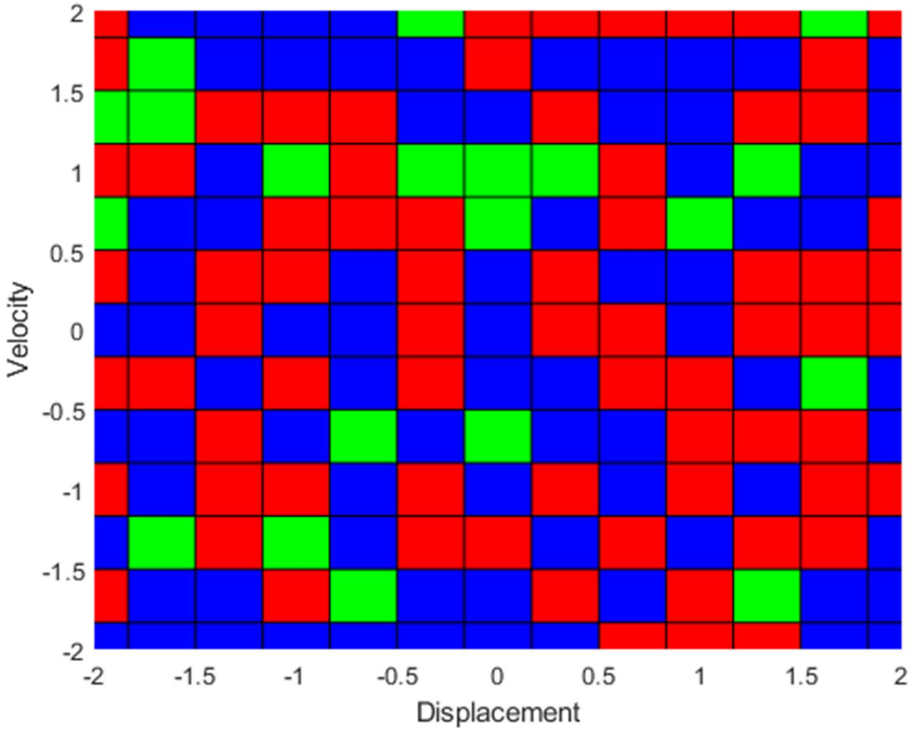


*Figure 2.47 Basins of attraction estimated by the brute force algorithm with resolution 12*12 (144 initial conditions)*

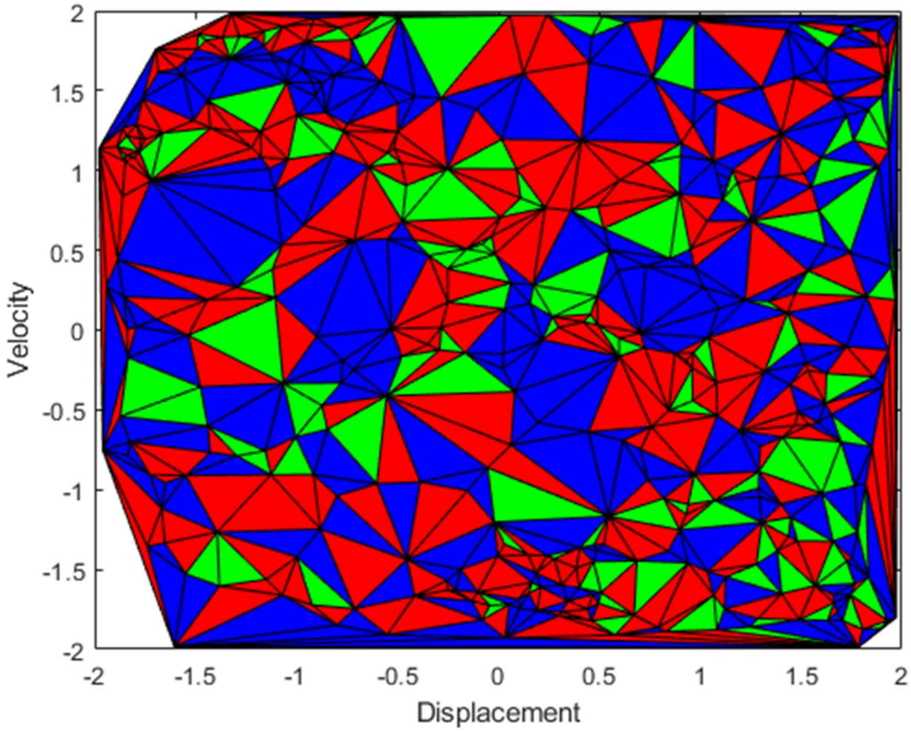


Figure 2.48 Basins of attraction estimated by the random triangular subdivision with 381 initial conditions

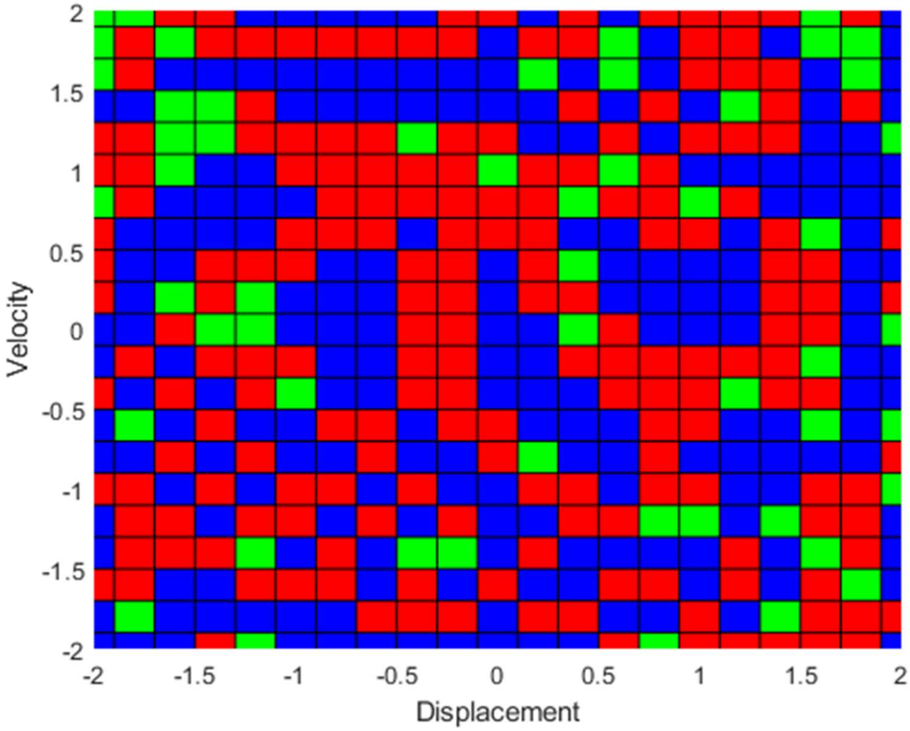


*Figure 2.49 Basins of attraction estimated by the brute force algorithm with resolution 20*20 (400 initial conditions)*

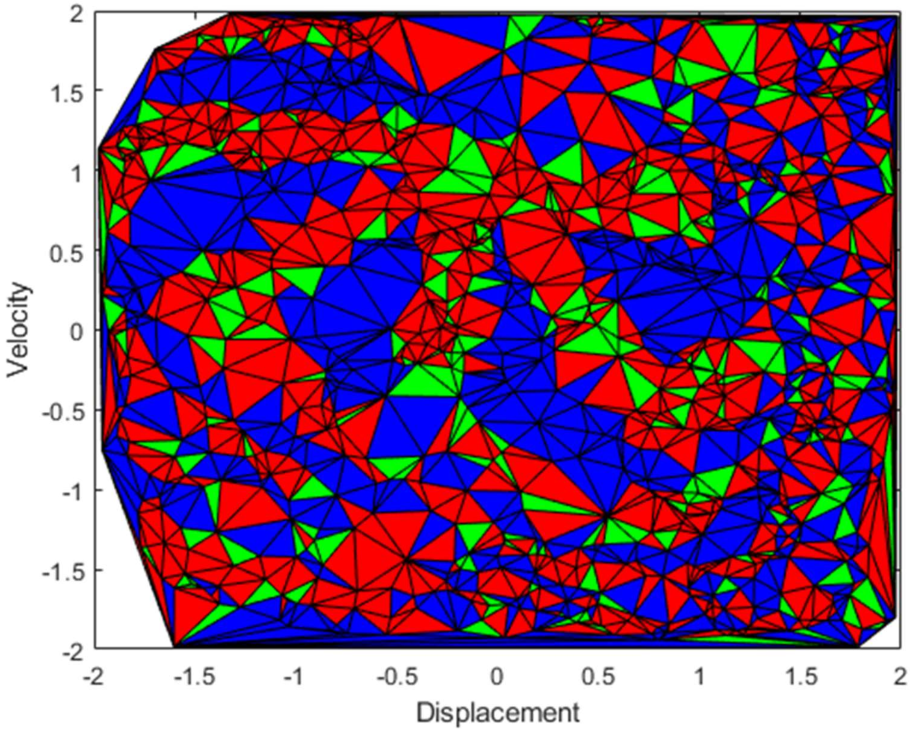


Figure 2.50 Basins of attraction estimated by the random triangular subdivision with 860 initial conditions

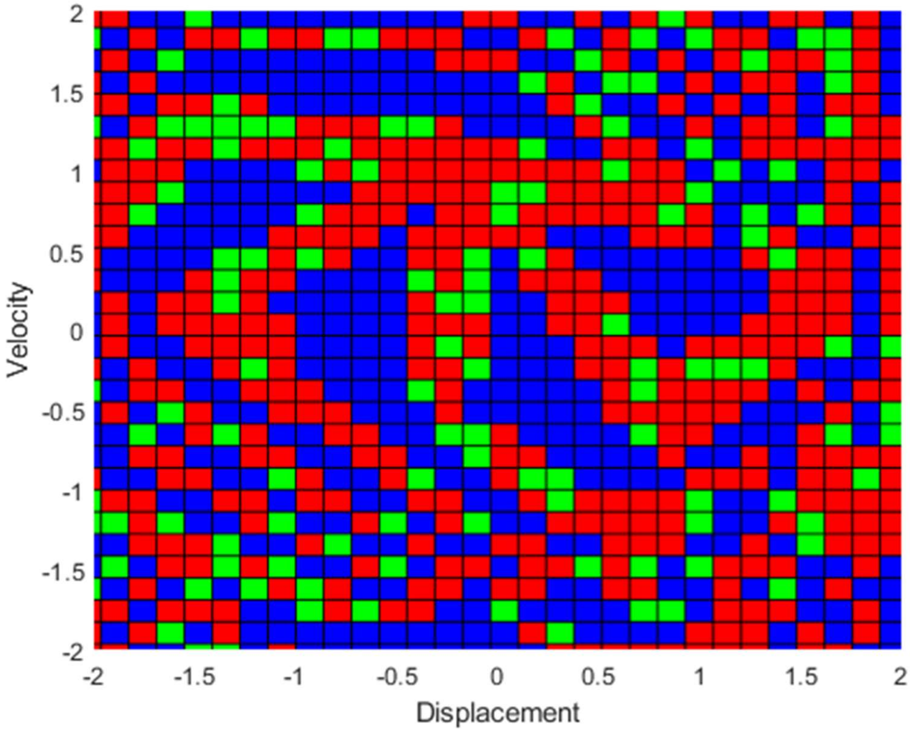


*Figure 2.51 Basins of attraction estimated by the brute force algorithm with resolution 29*29 (841 initial conditions)*

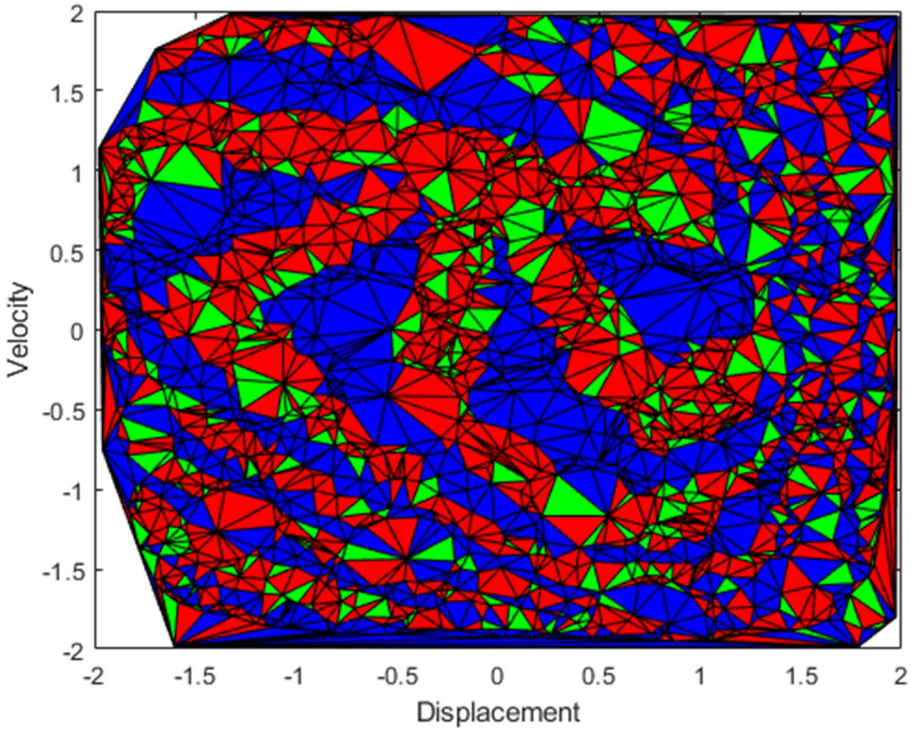


Figure 2.52 Basins of attraction estimated by the random triangular subdivision with 1493 initial conditions

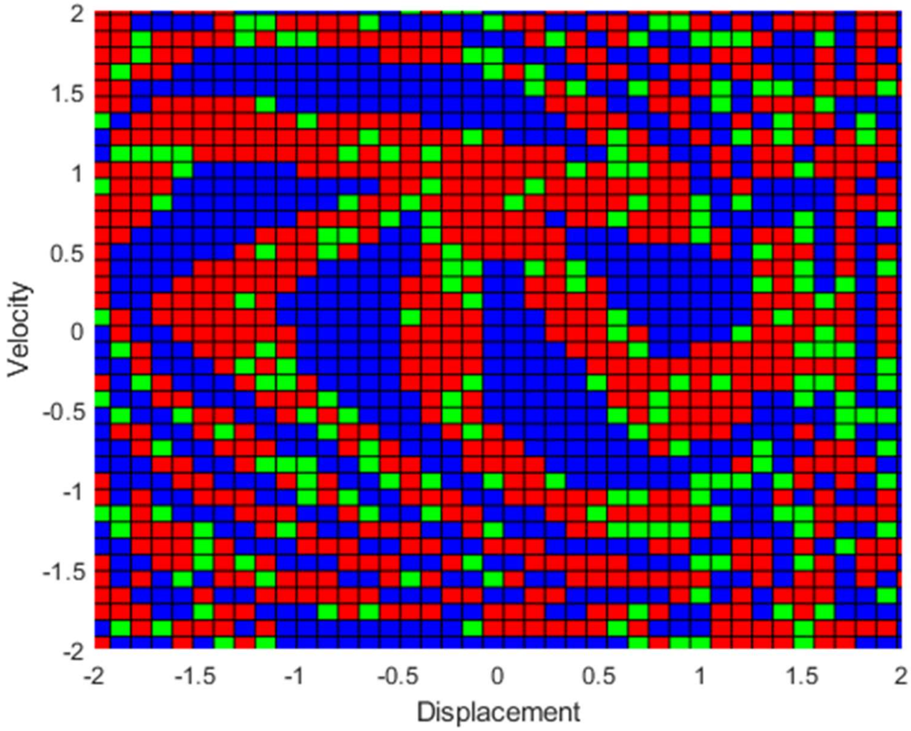


*Figure 2.53 Basins of attraction estimated by the brute force algorithm with resolution 39*39 (1521 initial conditions)*

2.6 Summary

In this chapter, a method for estimating the basin of attraction of multistable system called Randomised Triangular Subdivision was proposed and it is based on Delaunay Triangulation. Moreover, performance of the randomised triangular subdivision was investigated on two test systems namely Duffing oscillator and soft impact oscillator with different number of coexisting states respectively. Comparing with the traditional brute force algorithm, with the same number of initial conditions, the random triangular subdivision method can estimate the basins of attraction with better accuracy and can capture geometrical and topological features at low resolution in sacrifice of slightly longer computation time.

Chapter 3 Multistable State Switching using PD-like Control

This chapter proposes a PD-like controller to switch multistable systems from undesired state to desired state. The proposed method has been applied on three multistable systems namely, the Duffing oscillator, the soft impact oscillator and the soft impact oscillator with a drift, for performance evaluation.

3.1 Introduction

Multistable control switches a system from an undesired state to its desired state. This in turn can increases efficiency of the system. For example, a state with small amplitude oscillation can increase the lifespan of drill strings due to vibration reduction, which decreases maintenance costs and enhances productivity [8]. On the contrary, a large amplitude motion can generate more power, and thus improve the efficiency in a vibrational energy harvesting system [7].

3.1.1 Multistable State Switching Control Algorithms

In order to switch states in multistable systems, several methods have been proposed in literature. For example, state switching in the system using a feedforward control procedure [11-12] and steering it by applying a short pulse [13-14, 53-60] were reported in literature. These two methods are easy to apply and the knowledge of the dynamic system are not required. However, these methods have limitations. For instance, if there is an unwanted external noise injected to the system, it may switch back to the undesired state due to its high sensitivity to noise and disturbance.

A method was proposed for multistable control by applying a short pulse [53-60]. The state of the system is dependent on the initial conditions of the system. The principle of the method is as follow: when a short pulse is applied to a multistable system and the system can be seen as switch off for a short period of time. Then the system will switch on again from another initial conditions which belong to the basins of attraction of another attractor, i.e. the effect of the short pulse is equivalent to changing of initial conditions. Due to the short duration of the pulse, the parameters of the system will not change, i.e. the structure of the basins of attractor will remain unaltered. When the controller is applied on the system, it only effects the displacement of the system. If the variation of the displacement is large, the system is switched from one state to another. Otherwise, the system returns to the original state. Therefore, if the structure of the basins of attraction is complex, it is difficult to determine the level of the control input. Moreover, if there is external noise after control, the system may switch back to the undesired state due to its high sensitivity to its initial condition.

More methods were proposed for state switching by changing overall system dynamics. For example, destroying the undesired attractors by using a pseudo-periodic force can switch the multistable system to a desired state [15-16]; destroying the undesired attractor by adding a harmonic perturbation can also drive the system to the neighbouring desired attractor [61-63] and using a harmonic modulation to switch the system from multistable system to a monostable system [64-65]. These methods control the system by destroying the undesired attractor and the basin of the desired attractor is usually enlarged. However, the destruction of attractors may change system properties, including the basin of attraction's structure, in turn making it more complex and difficult to control. Another method for multistable control is through destroying the undesired attractor. The coexisting attractor can be destroyed by a parameter modulation with the suitable frequency and amplitude. Pisarchik found that, with different control amplitude of the modulation, the number of the coexisting attractor is different [62]. Moreover, for each coexisting attractor, it has its own relaxation oscillation frequency. If the control frequency of the modulation is close to the relaxation oscillation frequency of the undesired attractor, the attractor will lose its stability because the modulation and relaxation oscillation get in resonance and the response of the system become stronger. This control method can destroy the undesired attractors without changing the observed phenomenon of the desired attractor. However, this method is not suitable for all dynamic systems. It is because for some systems, with a small changing in system parameter, the structure of the basins of attraction will become more complicated. Therefore, it is important for a control law to maintain the original basin of attraction and maintain the system in a desired state under the effect of external disturbances.

A new control law called the intermittent control of coexisting attractors, was proposed by Liu *et al.* [66]. This is based on the 'act-and-wait' concept and the feedback controller switches on during the acting period and switches off when waiting [67-73]. The principle of this controller is to perturb the undesired attractor by an impulsive force and drive the system to the desired trajectory. Similarly, a displacement error between the current and desired trajectories will be set. If it is smaller than the threshold error the controller will switch on, and, otherwise the controller will turn off. The intermittent control law can switch a multistable system from the current state to a desired state without changing the system's parameters, although this requires knowledge of the plant's nonlinear model, which is sometimes difficult to obtain accurately. Therefore, a new PD-like control law is proposed in this thesis to control the system without prior knowledge of the plant's nonlinear model and without changing the system's original basin of attraction.

3.1.2 The proposed PD-like Control

Consider a multistable system with the following dynamic model,

$$\begin{aligned} \ddot{x} &= f(x,\dot{x}) + p(t) + u(t) \\ \ddot{x}_d &= f\ (x_d,\dot{x}_d) + p(t) \end{aligned} \tag{3.1}$$

where $x, \dot{x}$ and $x_d, \dot{x}_d$ are the displacement and velocity of the current and desired states of the system respectively. $f(x,\dot{x})$ and $f\ (x_d,\dot{x}_d)$ are the nonlinear functions of the current and desired states respectively. For the proposed PD-like controller, the exact nonlinear models of the plants $f(x,\dot{x})$ and $f\ (x_d,\dot{x}_d)$ are not required. $p(t)$ is the harmonic excitation applied to the model and $u(t)$ is the function for external control input.

The proposed PD-like control law is defined as follows:

$$u(t) = -k_p e_1 - k_d e_2 \tag{3.2}$$

where k_p and k_d are the positive gains of the controller. e_1 and e_2 are the displacement and velocity differences between the current and desired trajectories which are defined as follows:

$$\begin{aligned} e_1 &= x - x_d \\ e_2 &= \dot{x} - \dot{x}_d \end{aligned} \tag{3.3}$$

According to the principle of the proposed control law, the system will be driven from the current state to the desired state by exploiting the difference between the current and desired trajectories. The trajectory of the system will subsequently keep to the desired path, i.e. e_1 and e_2 converge to zeros and the external control force $u(t)$ is equal to zero. The system's basin of attraction will not be affected by the applied control because the system parameters will not be changed after control. If the system is under the effect of external noise and the current state does not follow the trajectory of the desired state, the controller will switch on automatically and drive the system back to the desired state. Furthermore, only the current and desired trajectories are used to steer the dynamical system and thus knowledge of the plant's nonlinearity is not required. However, this controller has some limitations. For example, sometimes, the trajectories of the state cannot be observed. This limitation can be overcome by estimating the trajectory of the state with the output response of the system and a nonlinear observer [74]. Moreover, the controller will switch on automatically when the system meets noise. However, the system can be driven back to the desired state by itself if the noise is small and the controller does not need to switch on. In order to overcome this limitation, the critical value of the noise level should be set for automatic activation of the proposed controller. If the noise level is larger than that critical value, the controller will switch on. Otherwise, the controller will not switch on and the system will follow the current trajectory on its own.

3.2 Numerical Simulation

The basins of attraction are estimated by the randomised triangular subdivision method described in Chapter 2 and which the system's trajectories can be observed during the process. The current state can be switched to the desired state by applying the proposed PD-like controller. The state switching function of the proposed controller is testedin the three test systems, namely Duffing oscillator, soft impact oscillator and soft impact oscillator with a drift.

3.2.1 Duffing Oscillator

Figure 3.1 shows the Duffing oscillator's basin of attraction with parameters $k = 0.9,\ \Gamma = 1.9$ and $\omega = 1.2$. There are two coexisting attractors under this set of system parameters. Figures 3.2 and 3.3 show the trajectories of the two coexisting attractors respectively. Both are period-1 response with small (red region in Figures 3.1 and 3.2) and large (blue region in Figures 3.1 and 3.3) amplitude respectively of the system. Figures 3.4 and 3.5 show respectively the displacement and velocity responses of the Duffing oscillator when the PD-like controller is applied to the system at 105.77 seconds with the arbitrarily chosen control parameters $k_p = 1$ and $k_d = 1$. The controller with this set of control parameters can switch the system from current state to desired state successfully. The maximum peak of the external control force is equal to 1.6730 and after approximately 15 seconds the system is switched to the desired state. When the system converges to the other state, the control input $u(t)$ becomes 0, as shown in Figure 3.6.

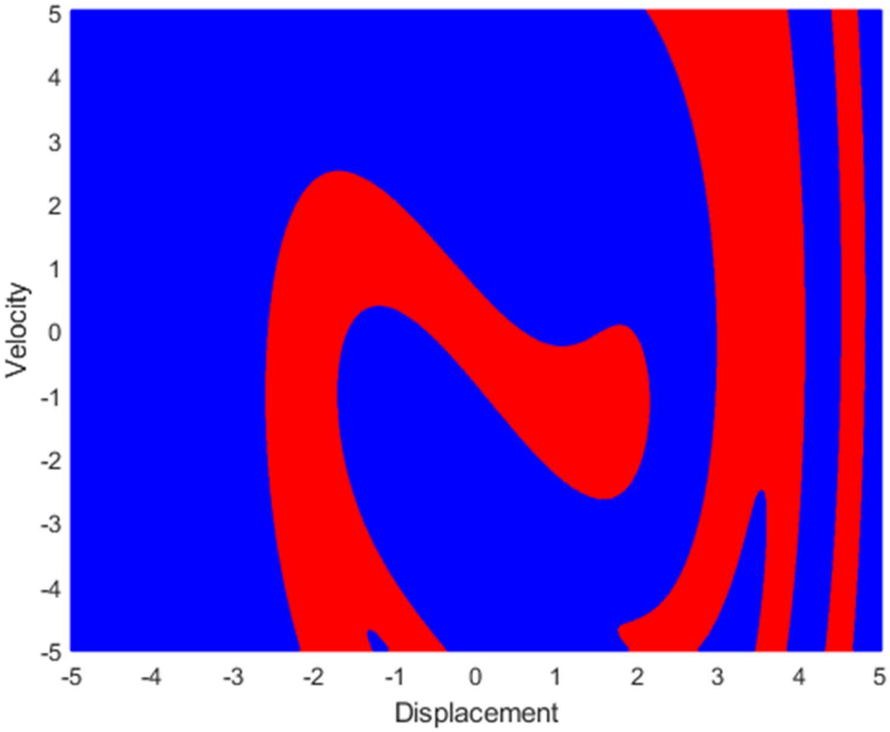


Figure 3.1 Basin of attraction of the Duffing oscillator with system parameters: $k = 0.9, \Gamma = 1.9$ *and* $\omega = 1.2$

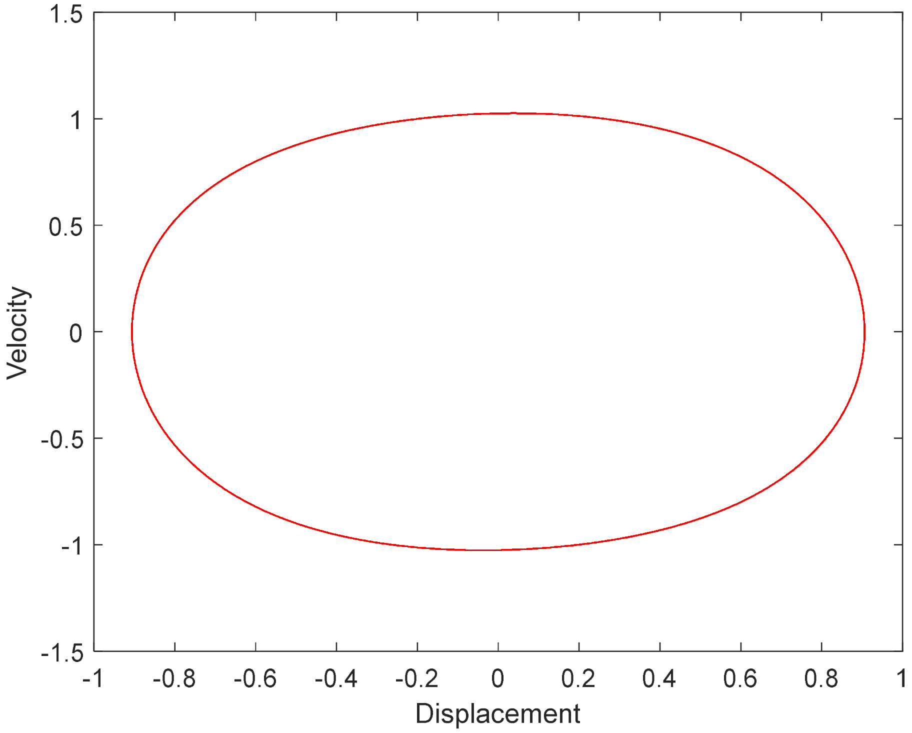


Figure 3.2 State trajectory of the co-existing attractor 1 on the phase plane

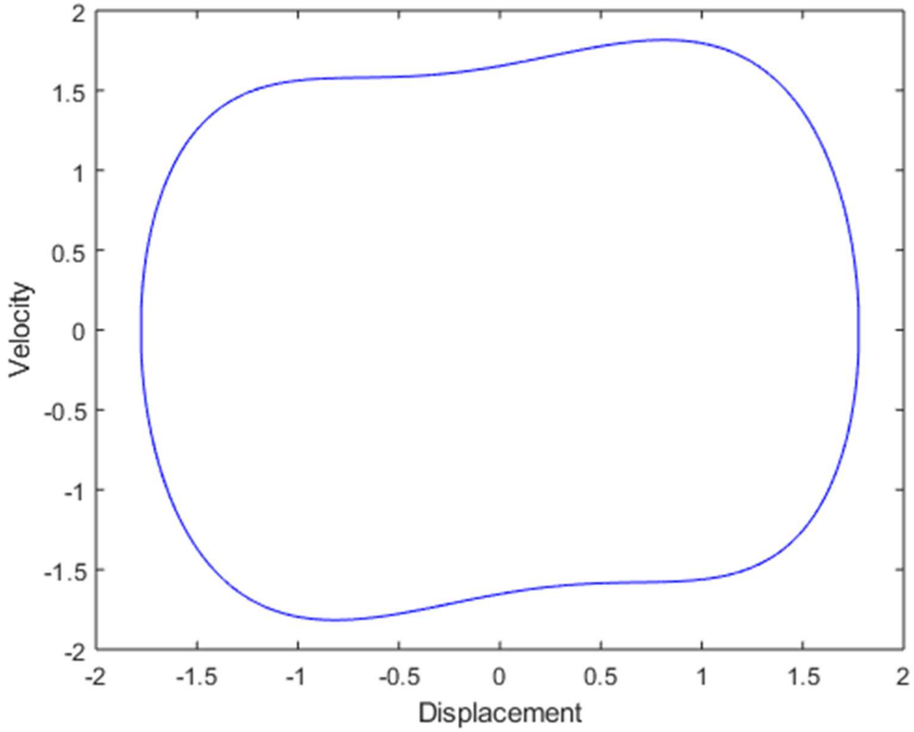


Figure 3.3 State trajectory of the co-existing attractor 2 on the phase plane

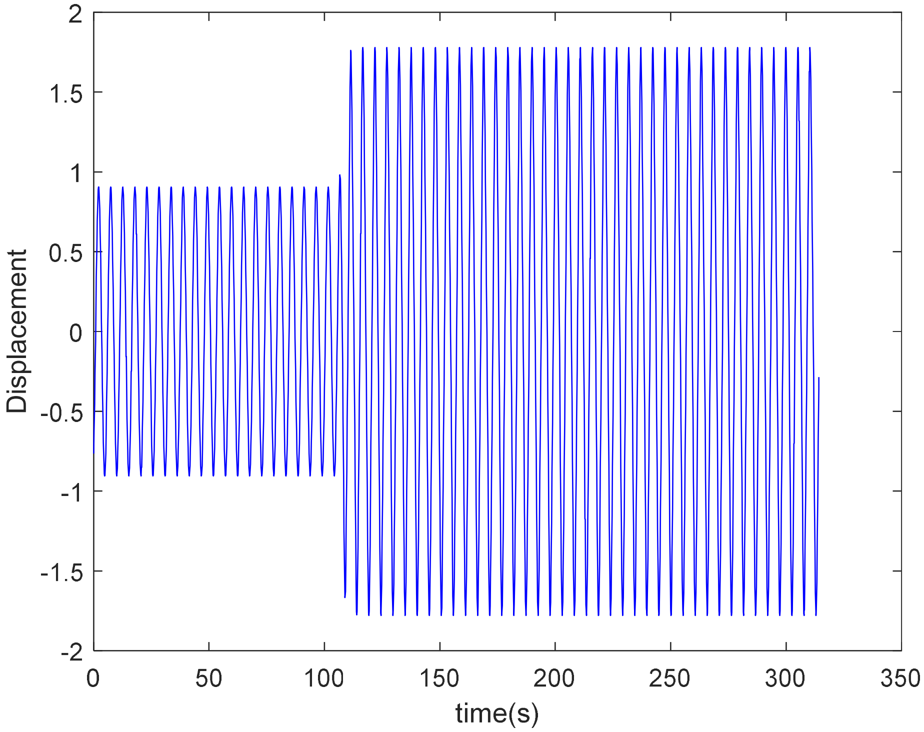


Figure 3.4 Response of displacement as a function in time

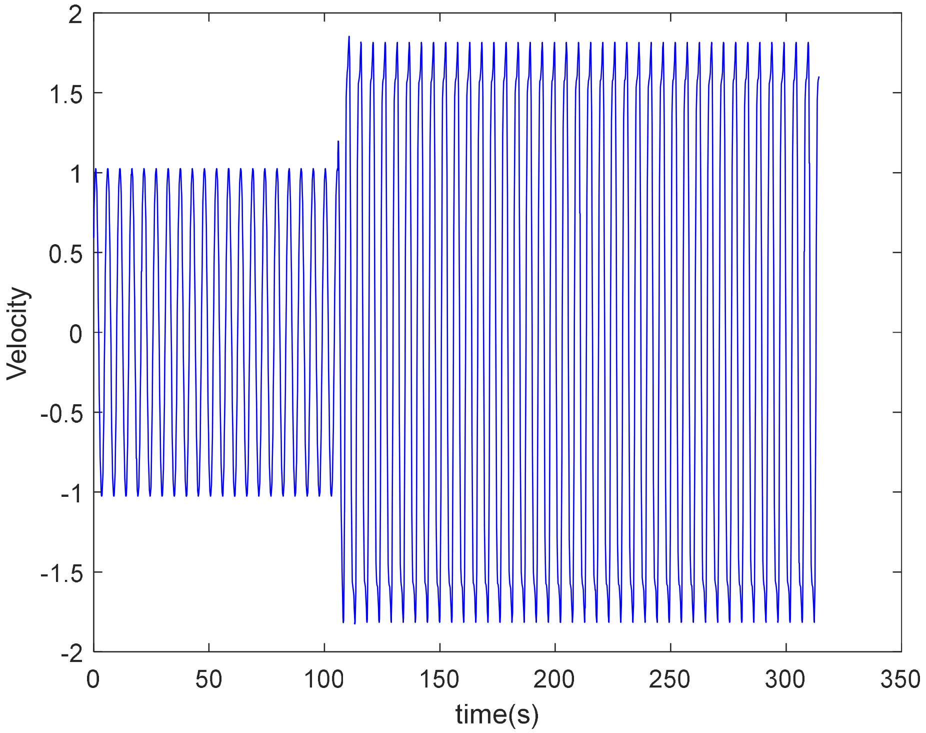


Figure 3.5 Response of velocity as a function in time

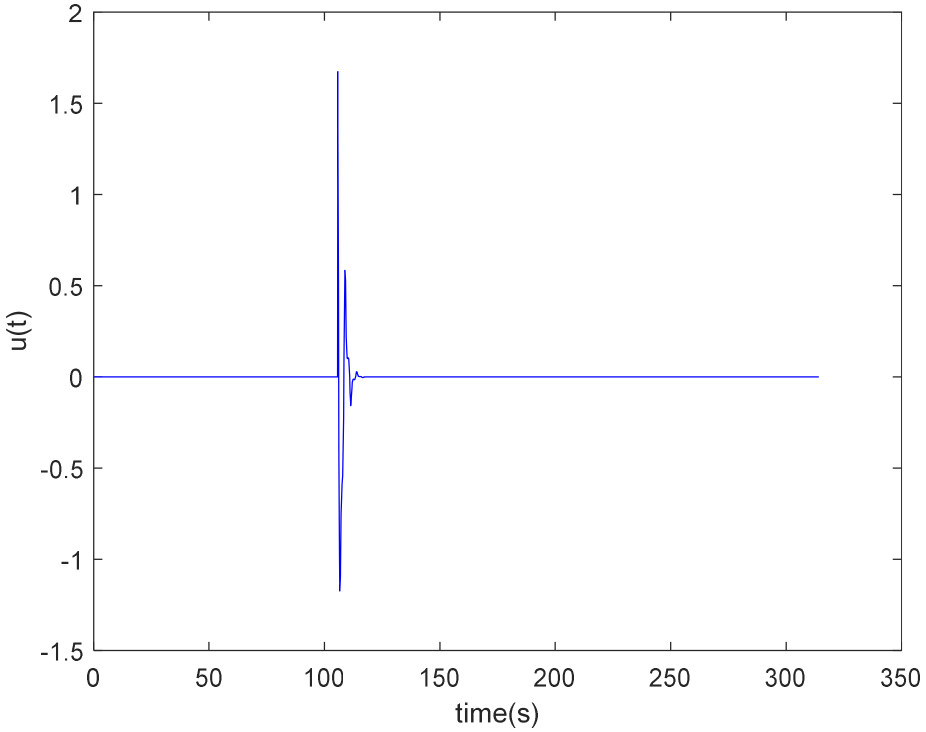


Figure 3.6 The applied control force as a function in time

3.2.2 Soft impact oscillator

There are two coexisting stable states in the soft impact oscillator under the set of chosen parameters: $\xi = 0.01$, $\beta = 29$,$g = 1.26$, $\Gamma = 1.0385$ and $\omega = 0.686$. Figure 3.7 shows the system's basin of attraction and Figures 3.8 and 3.9 show the two states' trajectories respectively. One stable state is a period-1 response with one impact response (blue region in Figures 3.7 and 3.8) while the other is a period-2 with two impact responses (red region in Figures 3.7 and 3.9). Moreover, the first and second states have small and large amplitudes respectively. The x axis indicates the displacement of the mass and the y axis shows its velocity in Figures 3.7 - 3.9. Moreover, Figures 3.10 and 3.11 show the two states' performances as functions in time. The controller's objective is to switch the system from the period-1 with one impact response to the period-2 with two impacts response. The arbitrarily chosen control parameters are $k_p = 1$ and $k_d = 1$ which can switch the system from current state to the desired state successfully. Figure 3.12 shows the simulation result of applying the controller on the soft impact oscillator at 94.64 seconds. The system then completed switching from period-1 with one impact response to the period-2 with two impact responses at 116.83 seconds. The external control force's maximum peak is 3.1575. Moreover, the external control input $u(t)$ becomes 0 when the system is driven to the other state.

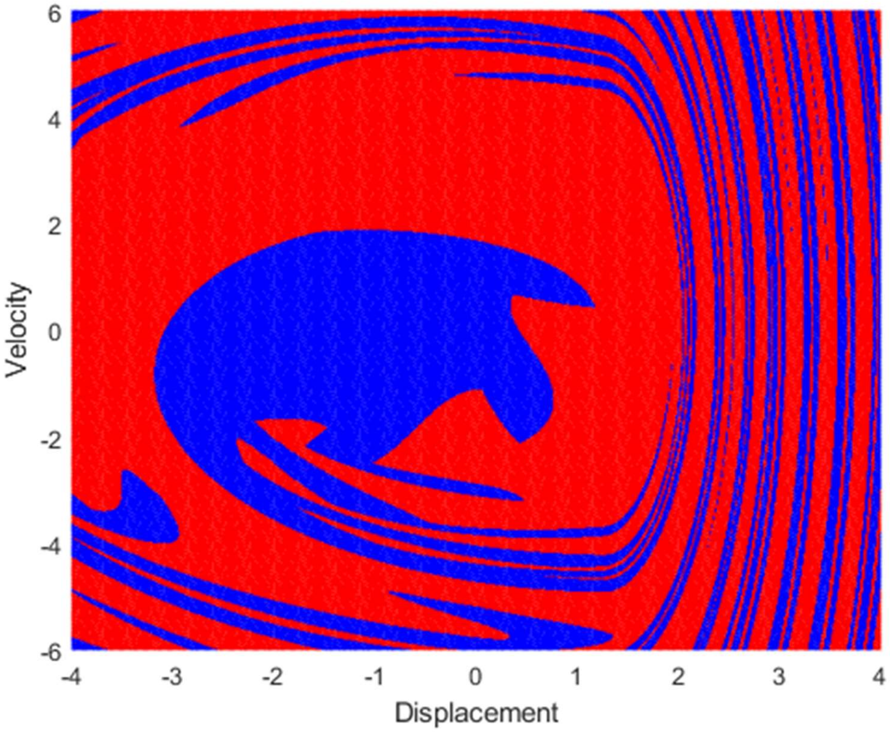


Figure 3.7 The basins of attraction of the soft impact oscillator with the system parameters: $\xi = 0.01, \beta = 29, g = 1.26,\ \Gamma = 1.0385$ *and* $\omega = 0.686$

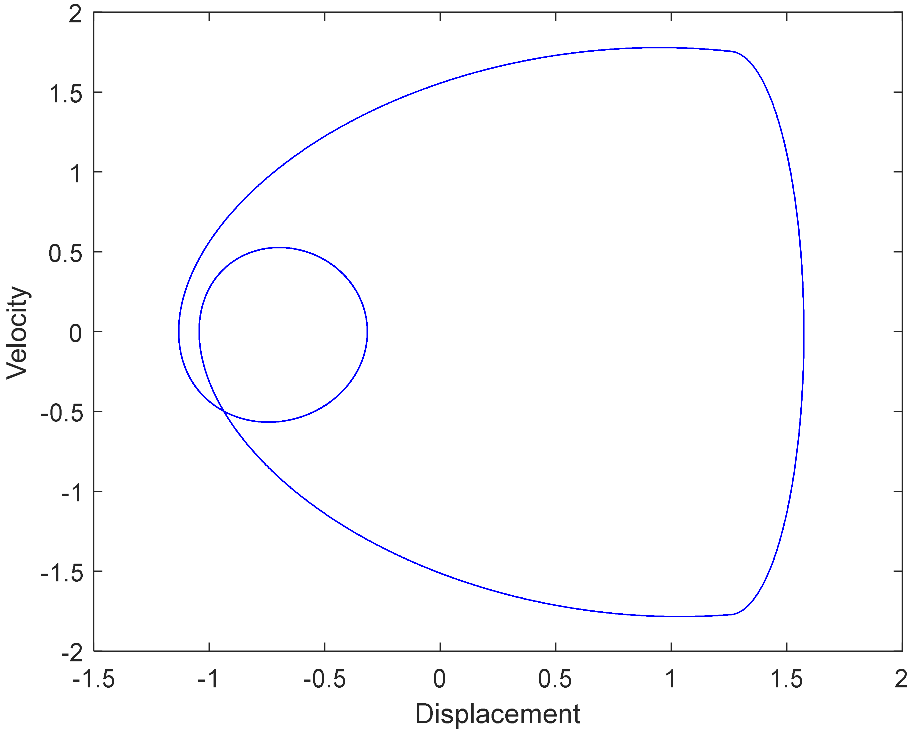


Figure 3.8 State trajectory of the co-existing attractor 1 on the phase plane

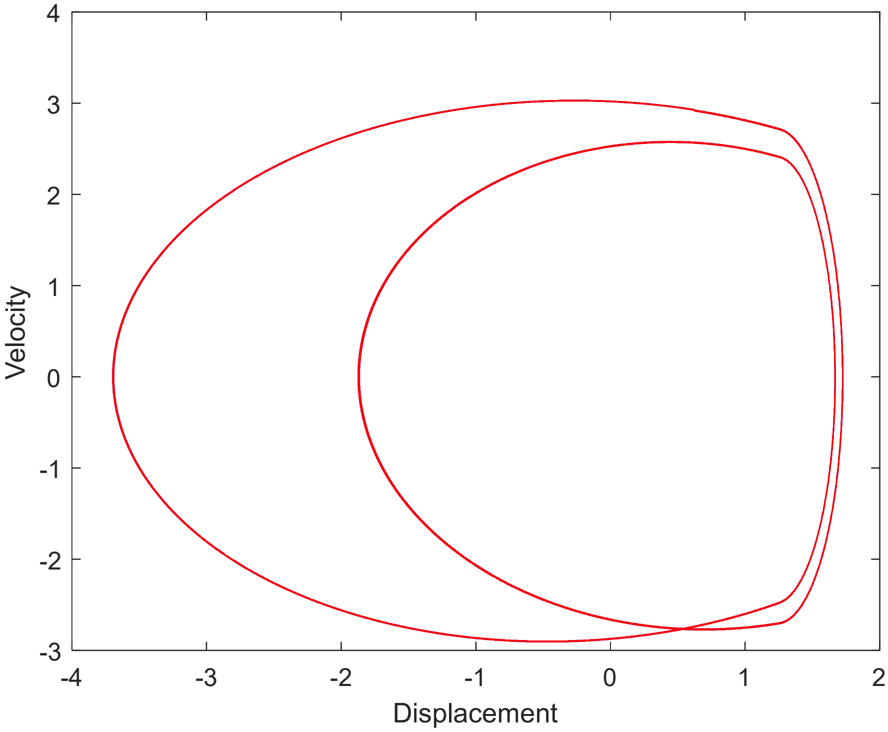


Figure 3.9 State trajectory of the co-existing attractor 2 on the phase plane

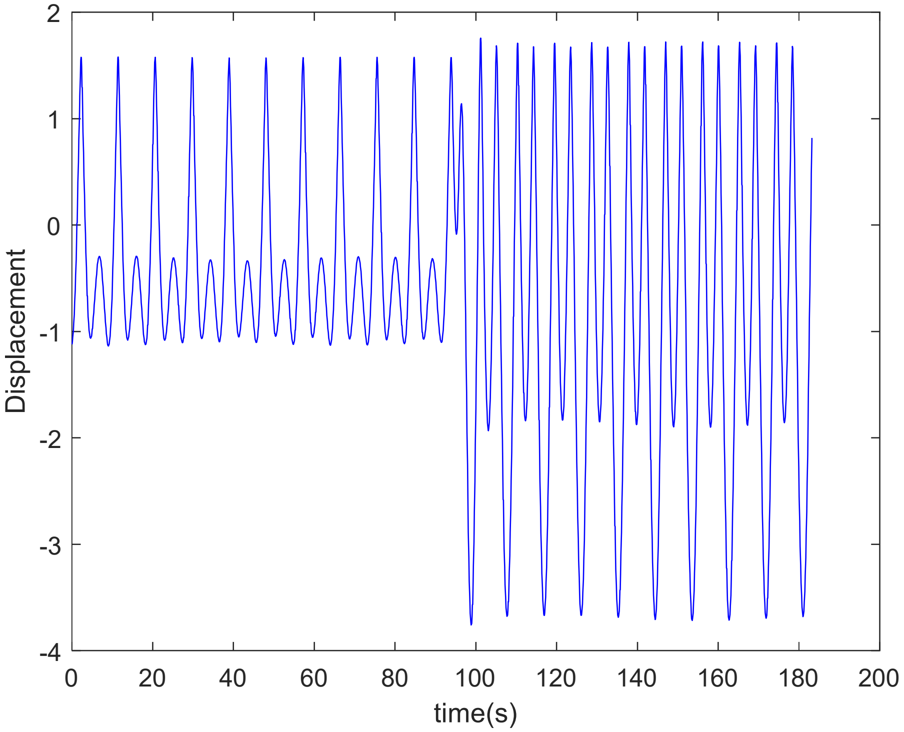


Figure 3.10 Displacement of the mass as a function in time

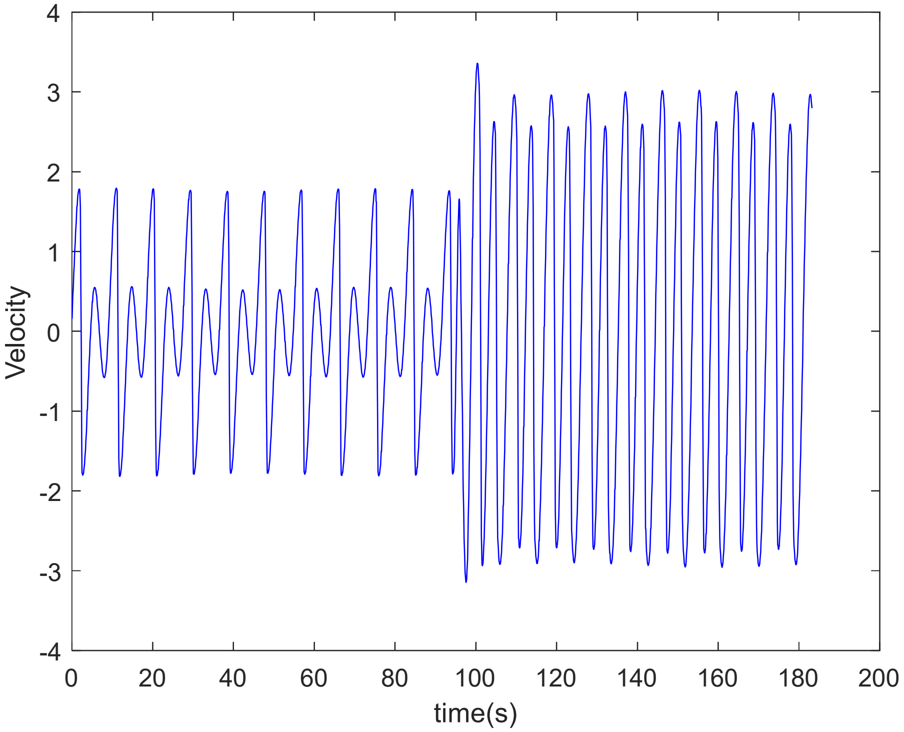


Figure 3.11 Velocity of the mass as a function in time

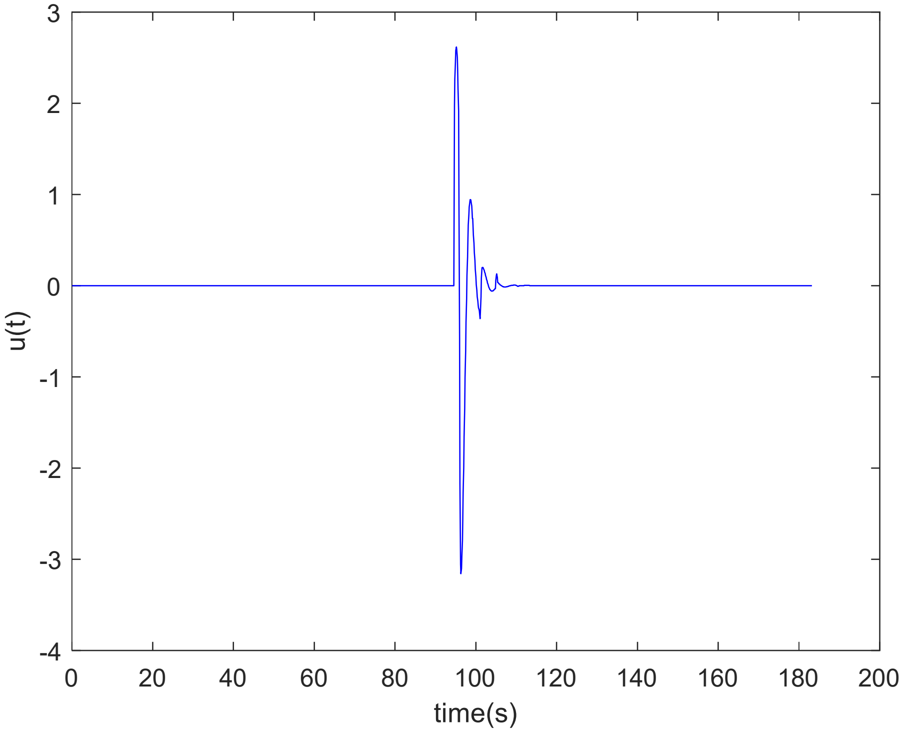


Figure 3.12 The applied control force as a function in time

3.2.3 Soft impact oscillator with a Drift

When the soft impact oscillator with a drift is under this set of parameters of $a = 0.3$, $b = 0.2$, $\omega = 1$, $g = 0.02$, $\xi = 0.1$ and $\varphi = \pi/2$, it has two coexisting states [28]. Figure 3.13 shows the basin of attraction and two coexisting attractors are observed. The trajectories of the multistable states are illustrated in Figures 3.14 and 3.15 with the x-axis showing the relative displacement between the mass and bottom plates and the y-axis showing the velocity of the mass. One trajectory exhibits a period-2 response (blue area) and the other exhibits a period-1 response (red area) as shown in Figure 3.13. Figure 3.16 illustrates the trajectory as a result of the displacement of mass (blue line) and bottom plates (red line) as a function of time under the effect of the proposed controller. At first, the mass moves but keeping the bottom plate static. Then, after approximately 252.37 seconds, the controller is applied. The control parameters are arbitrarily chosen as $k_p = 1$ and $k_d = 1$. The controller can switch the system from current state to the desired state under this set of control parameters successfully. Within 16 seconds the system switches to the desired state in which the mass moves with the bottom plate. Moreover, this soft impact oscillator with a drift is a simplified model of percussive drilling. No drilling progression is observed before applying the proposed controller as the mass moves but the bottom slider does not. This simulates the case when the drill bit stays in the same position. On the other hand, drill bit progresses as the mass moves with the bottom slider after the proposed controller is applied to switch to the desired state. The maximum peak of the external control force is 2.0829. Moreover, the external control force becomes 0 after the system switches to the desired state as shown in Figure 3.17.

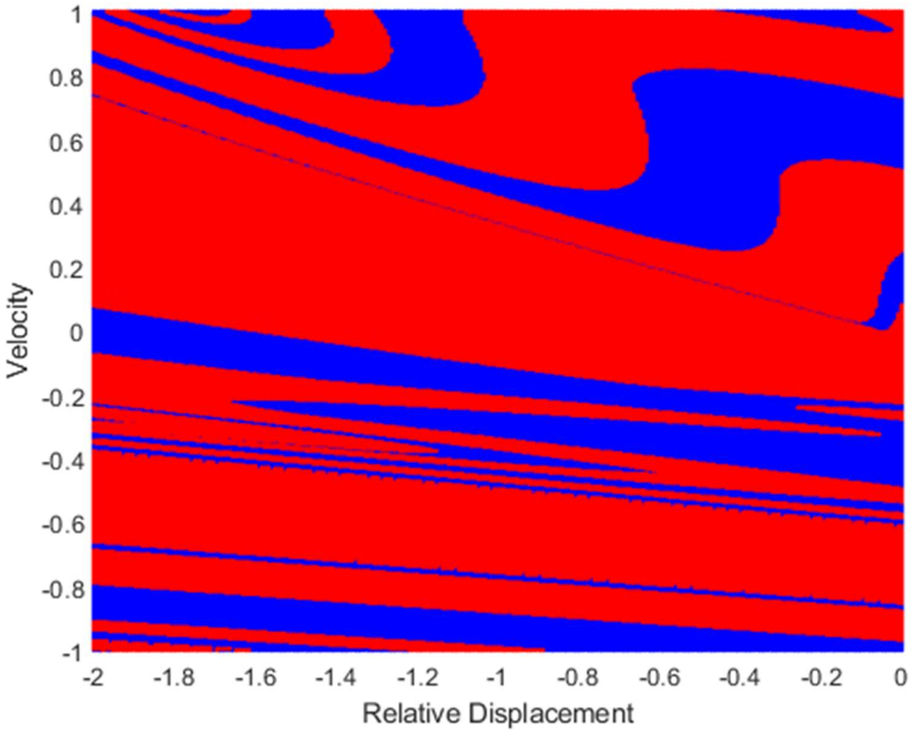


Figure 3.13 The basins of attraction of the soft impact oscillator with a drift with the system parameters: $a = 0.3$, $b = 0.2$, $\omega = 1$, $g = 0.02$, $\xi = 0.1$ *and* $\varphi = \pi/2$

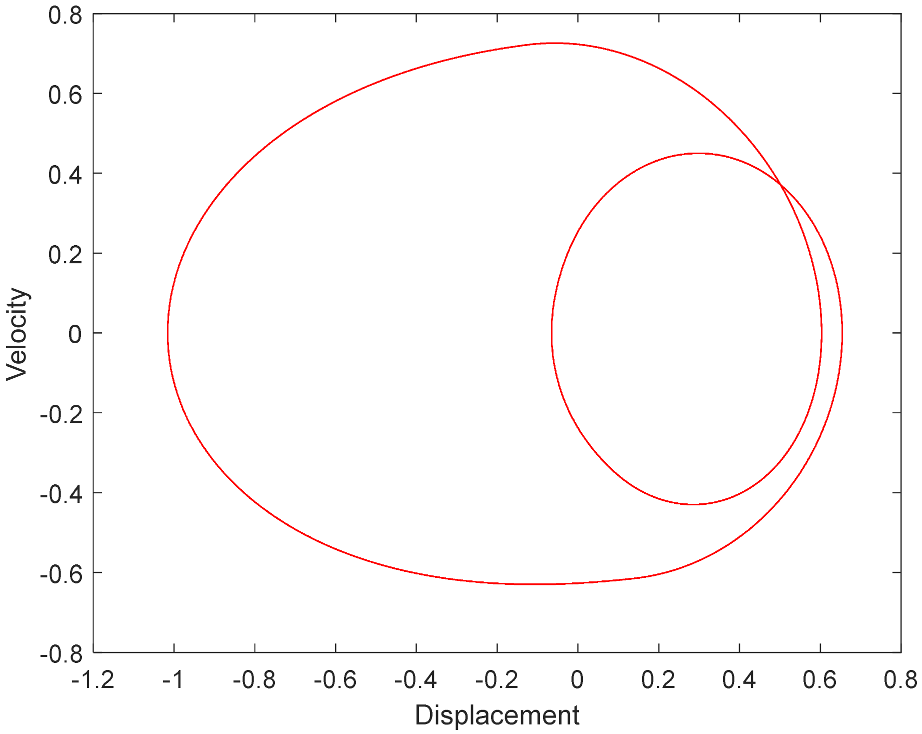


Figure 3.14 State trajectory of the co-existing attractor 1 on the phase plane

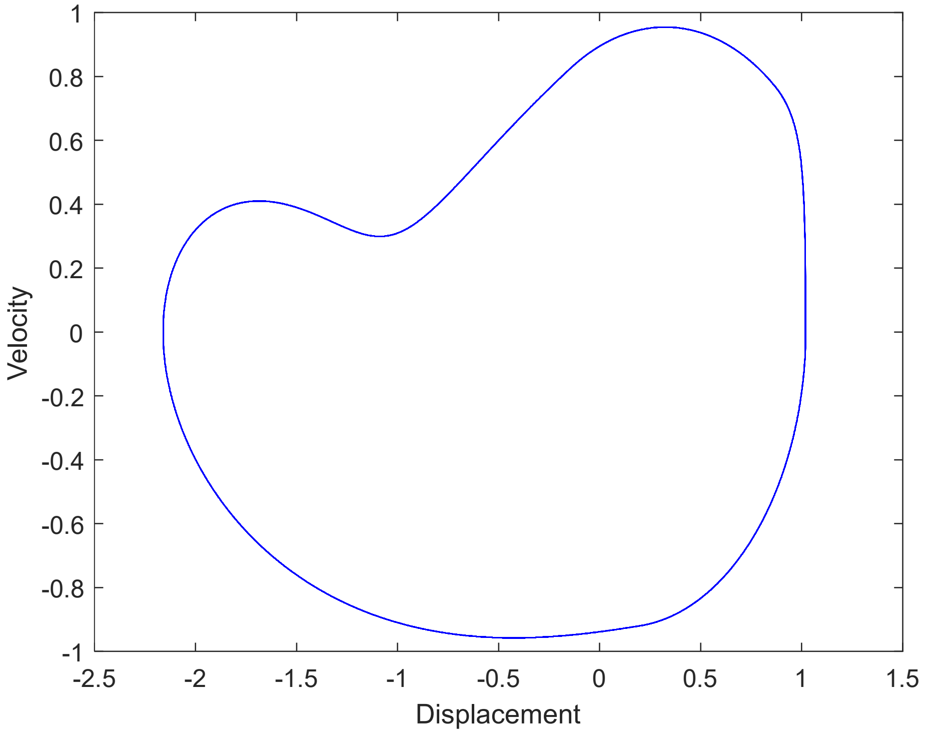


Figure 3.15 State trajectory of the co-existing attractor 1 on the phase plane

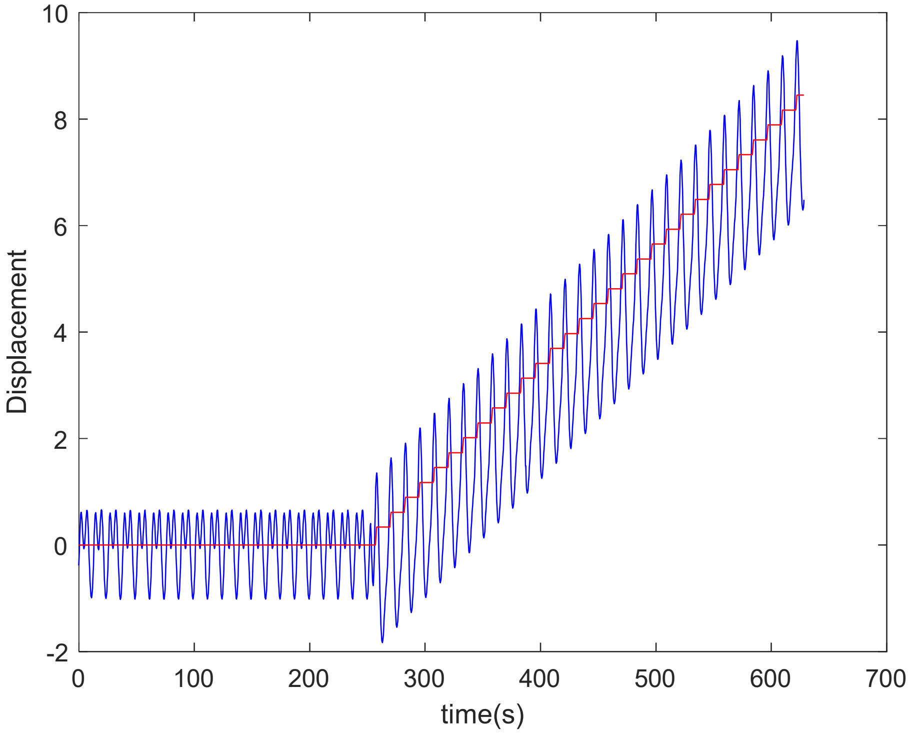


Figure 3.16 Displacement of mass (blue line) and bottom plate (red line) as a function in time

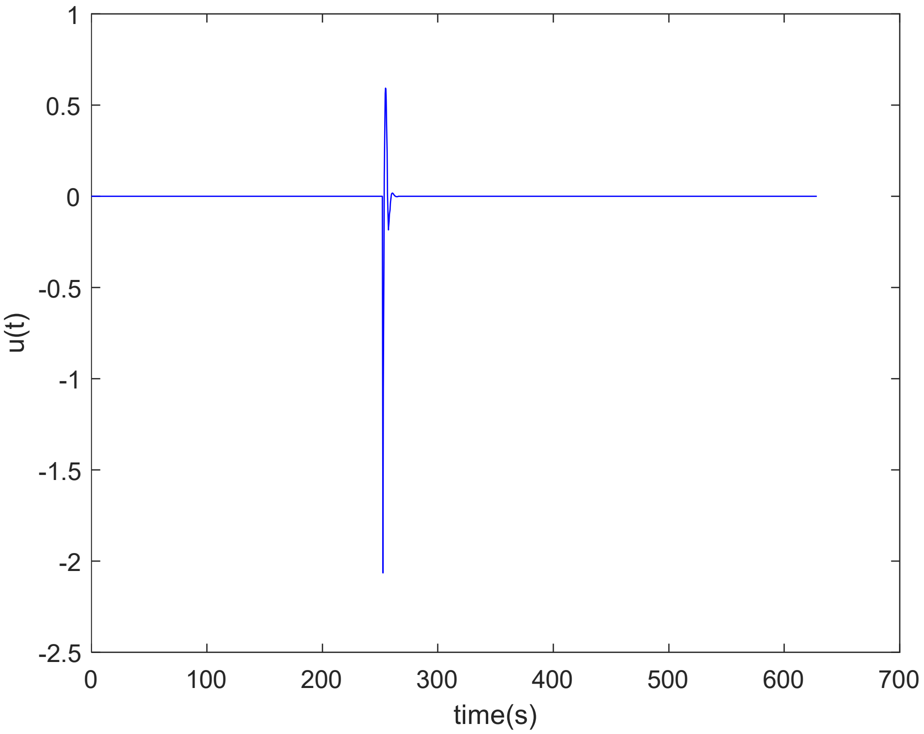


Figure 3.17 The applied control force as a function of time

3.3 Summary

In this section, a new multistable state switching controller called the PD-like controller, was proposed. It is an extension of the intermittent control and exploits the difference of the displacements between current and desired states for state switching. This method was tested on the Duffing and soft impact oscillators, which are smooth and non-smooth systems respectively. Moreover, the controller was also applied on the soft impact oscillator with a drift to demonstrate the effectiveness of the proposed controller in complicated systems. According to the results presented in this chapter, the PD-like controller is capable of switching a multistable system from its current, undesired state to the desired state. However, occasionally the controller requires a sequence of large spikes in actuator output (control input) for state switching, which is sometimes impossible for an actuator to provide and it is harmful to the actuator for prolonged operation. Therefore, in the next chapter, the constrained PD-like controller will be proposed to limit the control input's strength.

Chapter 4 Constrained PD-like Control

This chapter presents a constrained PD-like controller for limited actuator output. It is applied on the three test systems namely: the Duffing oscillator, the soft impact oscillator and the soft impact oscillator with a drift (described in Section 1.3) for performance evaluation. In addition, the performance of the constrained PD-like control will be compared with that of the unconstrained counterpart presented in Chapter 3.

4.1 Introduction

4.1.1 Constrained Control of Multistable Systems

In practice, control systems are subjected to various constraints. For example, physical actuators have limited output capacity and they may not be able to generate the required control input to switch states in multistable systems. Sometimes, controllers suggest a series of huge spiky control input for state switching which cannot be achieved by existing actuators. It is also harmful to the actuators for continuous and prolonged generation of output going back and forth between two extremes according to the spiky control input determined by the proposed PD-like controller. Therefore, it is useful to bound the control input based on the capacity of the actuator and this chapter will compare the performance of both the unconstrained (Chapter 3) and constrained PD-like controllers in state switching of multistable systems [66].

4.1.2 Constrained PD-like Control

The PD-like controller is capable of switching a multistable system to the desired state without any constraint on the magnitude of the control input. Sometimes, actuators cannot provide unlimited power to generate the control input required by the controller to switch state in the multistable plant. The constrained PD-like controller is proposed to limit the control input magnitude as shown below:

$$u(t) = \begin{cases} -k_p e_1 - k_d e_2 & |u| < u_{max} \\ sign(u) \cdot u_{max} & |u| \geq u_{max} \end{cases} \tag{4.1}$$

where $sign(u)$ is the sign of the control input u and u_{max} is the maximum control input supported by the actuator.

According to the principle of constrained PD-like control law, if the external control force is smaller than u_{max} the system can be immediately driven from the current state to the desired state using the proposed PD-like controller described in Chapter 3. If the external control force magnitude exceeds u_{max}, it will be constrained to $\pm u_{max}$ until the control force is smaller than u_{max}.

4.2 Numerical Simulation

In this section, the constrained PD-like controller will be applied to the three test systems namely Duffing oscillator, soft impact oscillator and soft impact oscillator with a drift, in order to compare its performance with that of the unconstrained PD-like controller.

4.2.1 Duffing Oscillator

The performance of the constrained PD-like control on the Duffing oscillator is shown in Figures 4.1 and 4.2. The blue line in Figure 4.1 depicts the control input generated by the unconstrained PD-like controller. Assume the actuator can provide limited power to drive the system and the bound of the external control force is set as $u_{max} = 0.3$, which is about 20% of the maximum peak of the PD-like control's external force (1.67). Both the unconstrained and constrained controllers adopt the same controller gains k_p, k_d and both controllers are switched at the same time.

According to the simulation results illustrated in Figures 4.1 and 4.2, the constrained PD-like control is capable of switching the Duffing oscillator to the desired state by limiting the control force to 0.3. The switching duration of the constrained PD-like controller is about 22 seconds, while its unconstrained counterpart only takes 15 seconds for successful state switching. Moreover, the control input has two more peaks than that generated by the unconstrained PD-like controller. In Figures 4.2 and 4.3, when the system is subject to an unconstrained input force, the velocity experiences a large jump and the system can be immediately switched to the desired state. Figures 4.4 and 4.5 show when the system is subject to a constrained input force, the process of driving the system to the desired state is divided into several stages, with incremental changes in the velocity across consecutive stages. The system is driven from current state to an intermediate state then to the desired state. Moreover, the external control force generated by the constrained PD-like controller is smoother than that by the unconstrained PD-like controller.

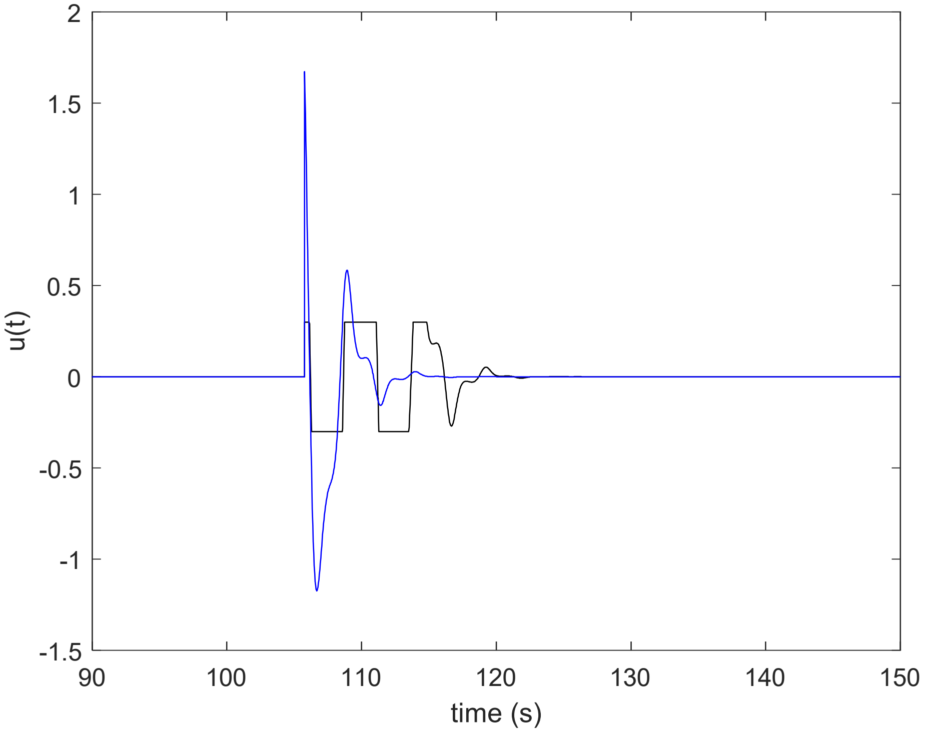


Figure 4.1 Control inputs to Duffing oscillator using constrained PD-like controller (black line) and unconstrained PD-like controller

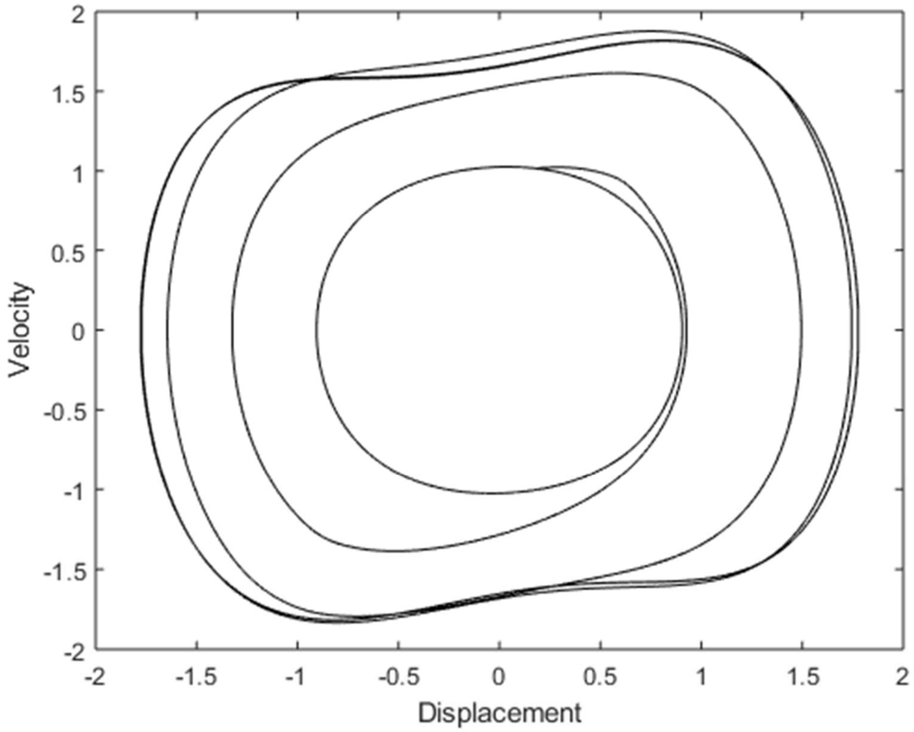


Figure 4.2 Trajectory of the system state when the constrained PD-like controller was applied

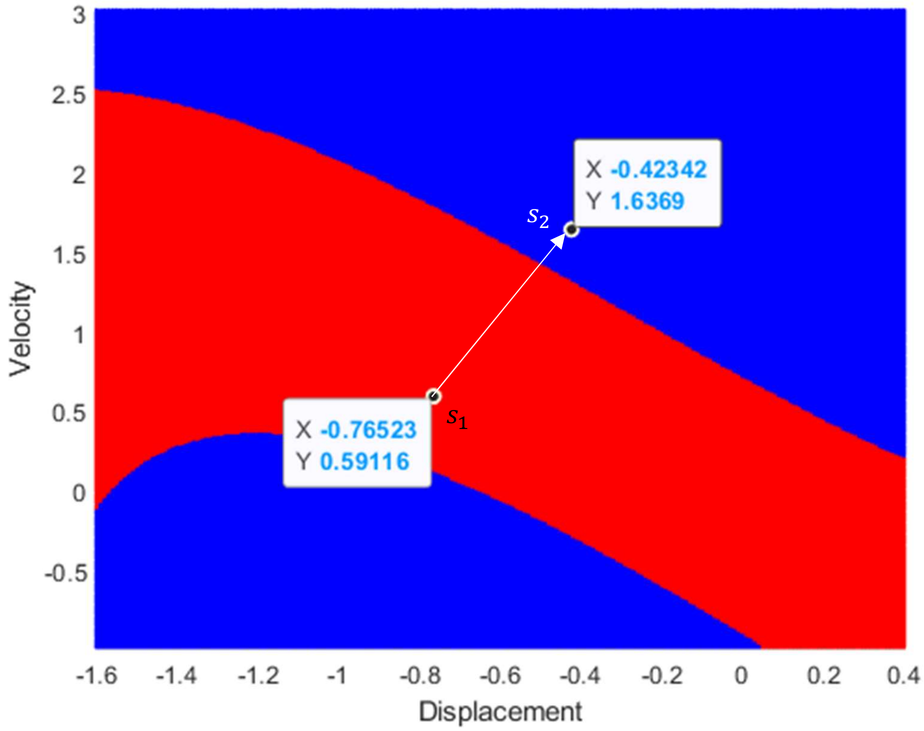


Figure 4.3 The system is driven from the current state s_1 to the desired state s_2 directly by using the unconstrained PD-like controller

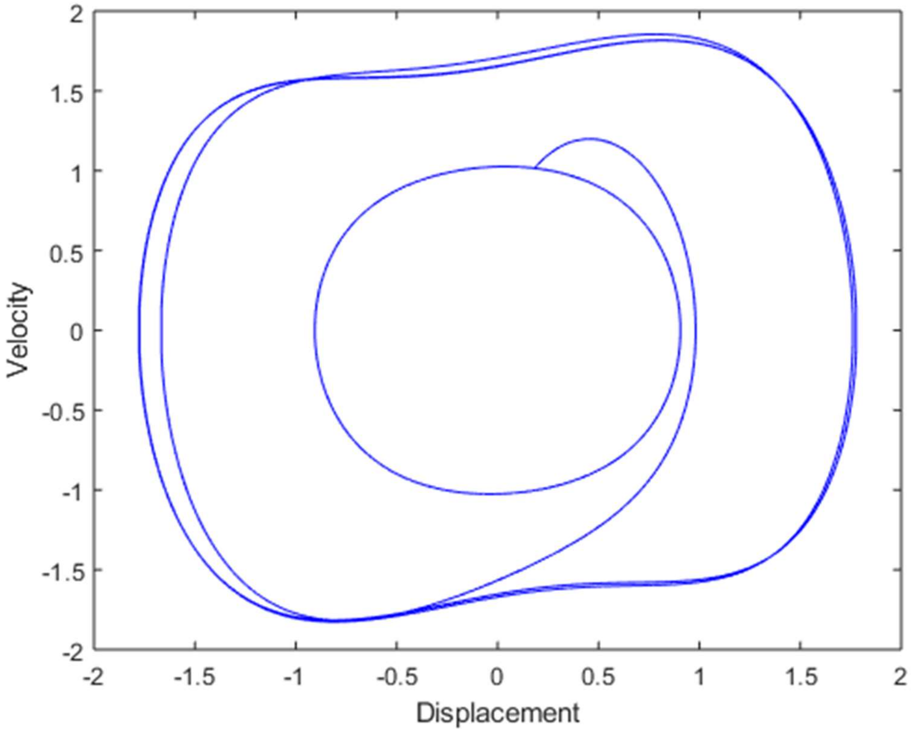


Figure 4.4 Trajectory of the system state when the unconstrained PD-like controller was

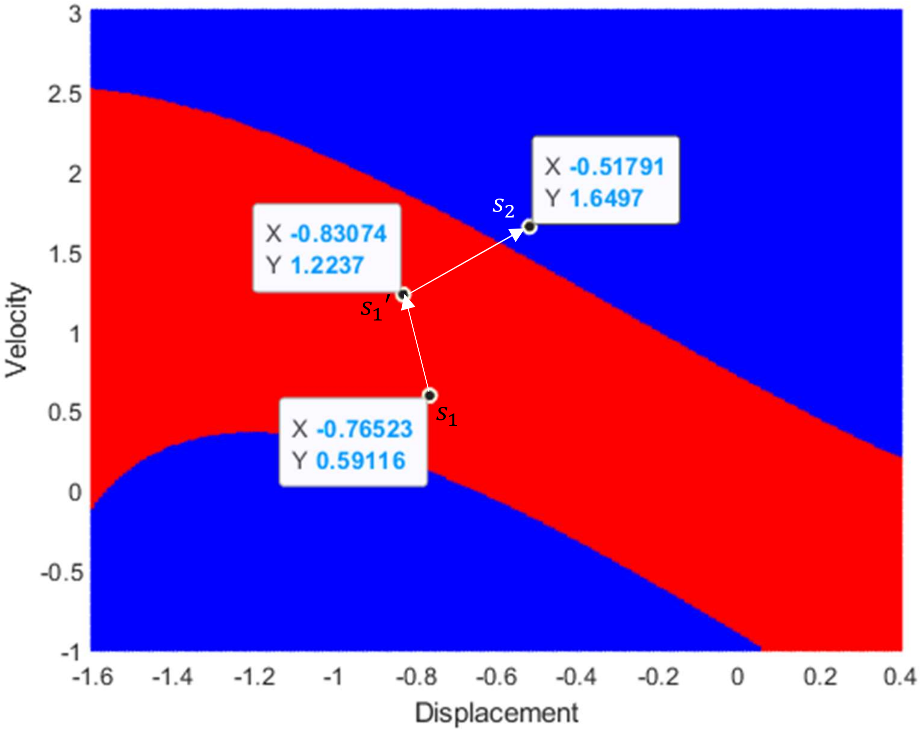


Figure 4.5 The system is driven from current state s_1 to the desired state s_2 through a transient state $s_1{}'$ by using the constrained PD-like controller.

4.2.2 Soft impact oscillator

Assume the controller is constrained by the maximum output of the actuator and the constrained PD-like controller is applied on the soft impact oscillator with a bounded control input $u_{max} = 0.6$, which is approximately 20% of the maximum peak of the PD-like control's external force (3.16). The constrained controller has the same control parameters as the unconstrained PD-like controller and both are applied to the system at the same time.

The black line in Figure 4.6 shows the constrained PD-like controller's external control input, while Figure 4.7 shows the system's trajectory following the application of the constrained PD-like control which is capable of switching the soft impact oscillator to the desired state with the bounded control input of $u_{max} = 0.6$. The blue line in Figure 4.6 depicts the control input generated by the unconstrained controller. The switching duration of the constrained controller is about 31 seconds while the switching duration of the unconstrained PD-like controller is about 22 seconds. There are eleven peaks and troughs in the control input, compared to seven peaks in the unconstrained PD-like controller. The results in Figures 4.7 and 4.8 reveal that the velocity changes of both the soft impact oscillator and unconstrained PD-like controller are faster than that of the constrained PD-like controller.

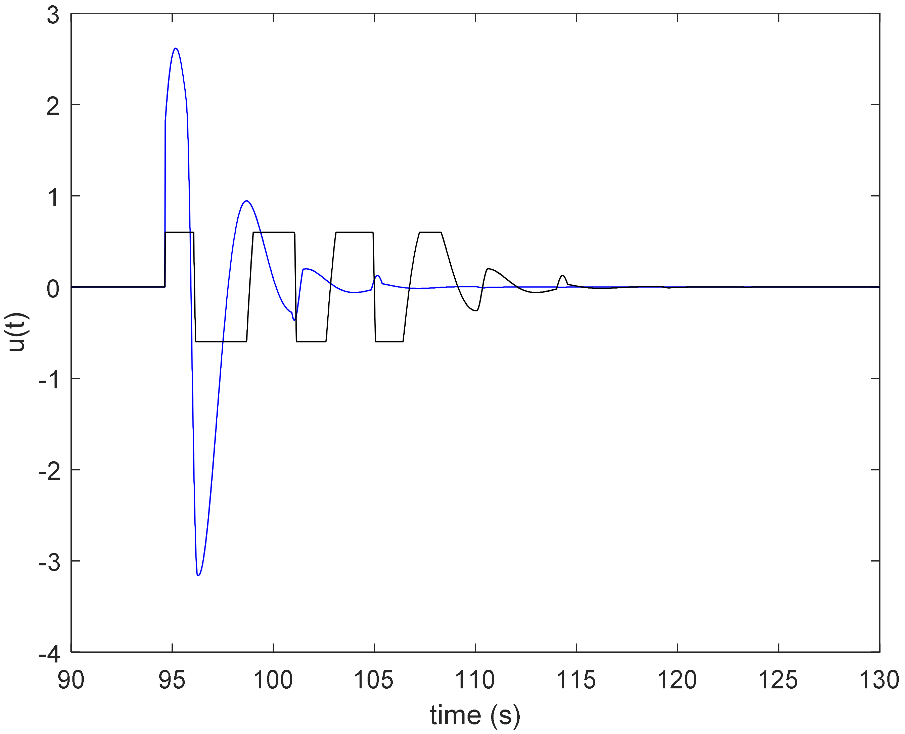


Figure 4.6 Control inputs to soft impact oscillator using constrained PD-like controller (black line) and unconstrained PD-like controller (blue line)

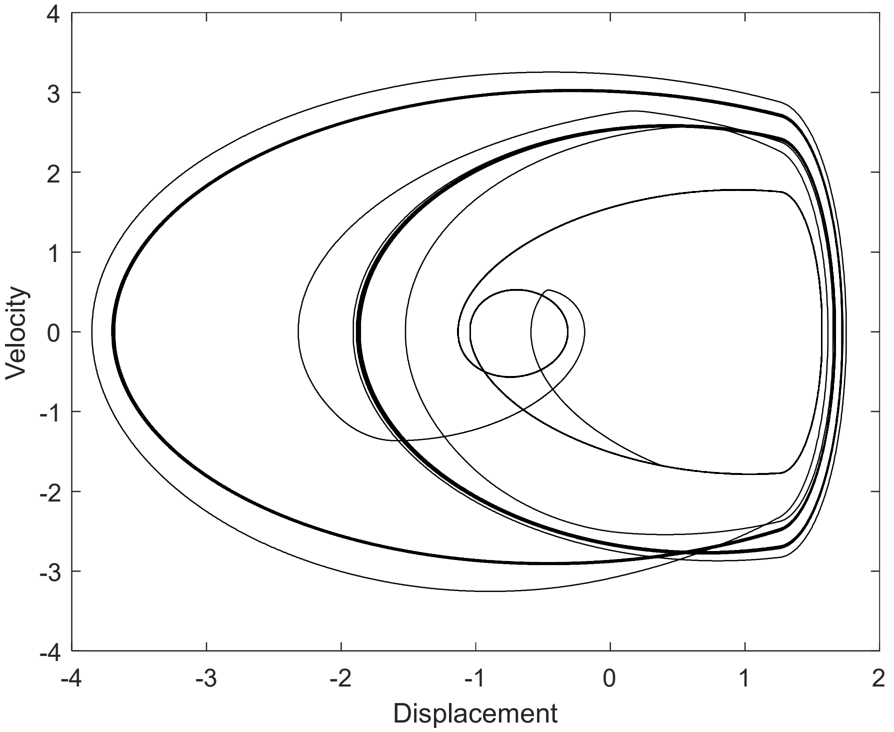


Figure 4.7 Trajectory of the system state when the constrained PD-like controller was applied

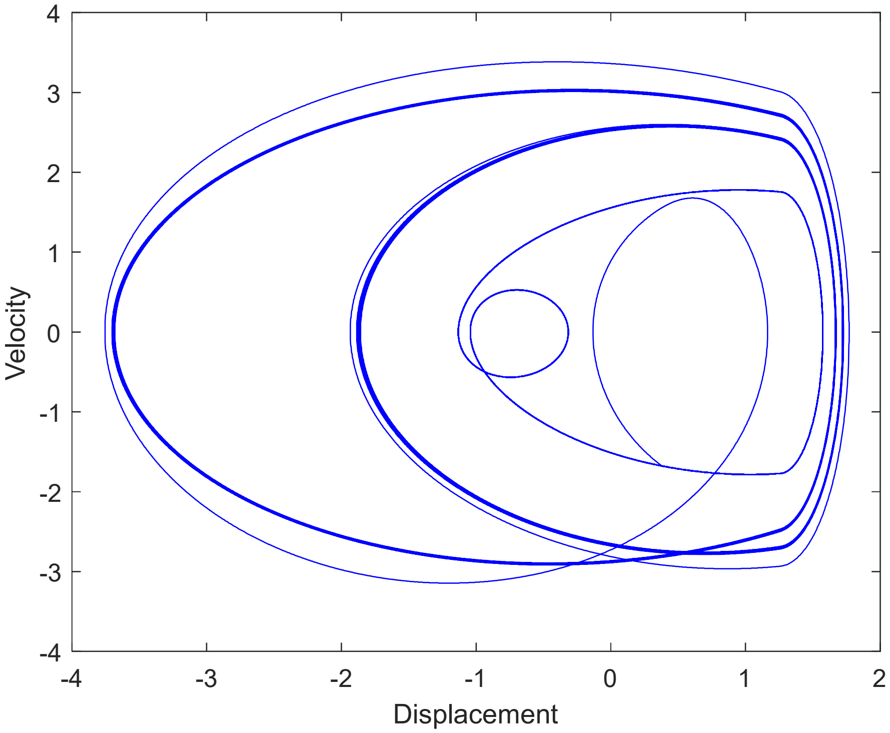


Figure 4.8 Trajectory of the system state when the PD-like controller was applied

4.2.3 Soft impact oscillator with a Drift

The constrained PD-like controller is applied on the soft impach oscillator with a drift with the performance shown below with Figures 4.9 and 4.10 illustrating the control input and state trajectory respectively. The blue line in Figure 4.9 depicts the control input generated by the unconstrained PD-like controller. Assume the actuator can only provide limited output and the maximum control input is set as $u_{max} = 0.4$, which is 20% of the PD-like control's peak external force (2.08). According to Figures 4.9 and 4.11, the constrained PD-like controller is capable of switching the system to the desired state within the bound of $u_{max} = 0.4$. The constrained PD-like controller and its unconstrained counterpart have the same control parameters and were applied at the same time instants. There are five peaks in control input and the switching duration is approximately 16 seconds. The blue line in Figure 4.9 reveals that the unconstrained PD-like controller's input also has five peaks, the maximum of which is 2.08, and the switching duration is about 15 seconds. In addition, simulation results show that the control input's maximum peak is limited while the switching duration increases. Moreover, the unconstrained controller's system was subject to a larger impulsive force. Figures 4.10 and 4.11 show that the respective trajectories of the unconstrained PD-like and constrained PD-like controllers are similar but the resulting velocity of the former is larger.

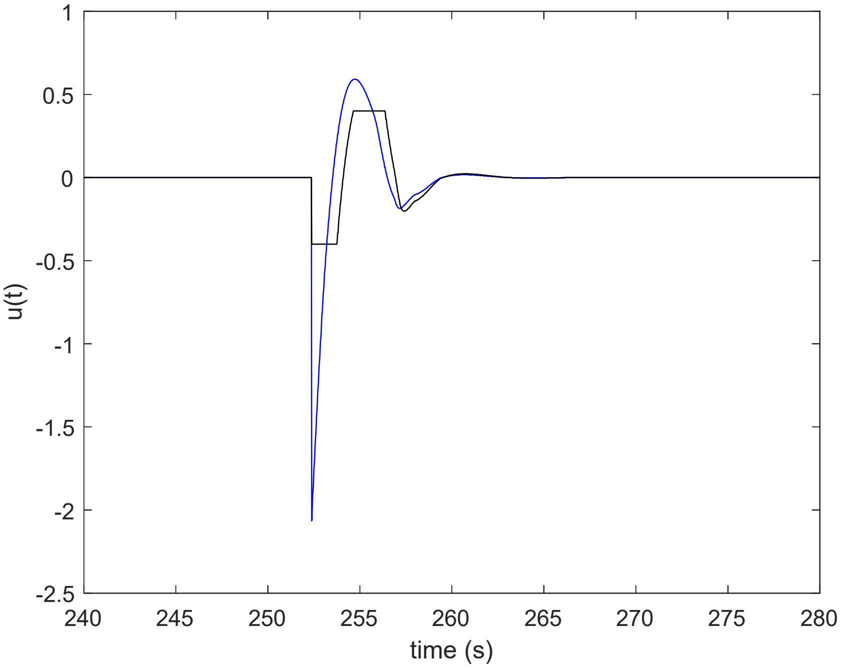


Figure 4.9 Control inputs to soft impact oscillator with a drift using constrained PD-like controller (black line) and unconstrained PD-like controller.

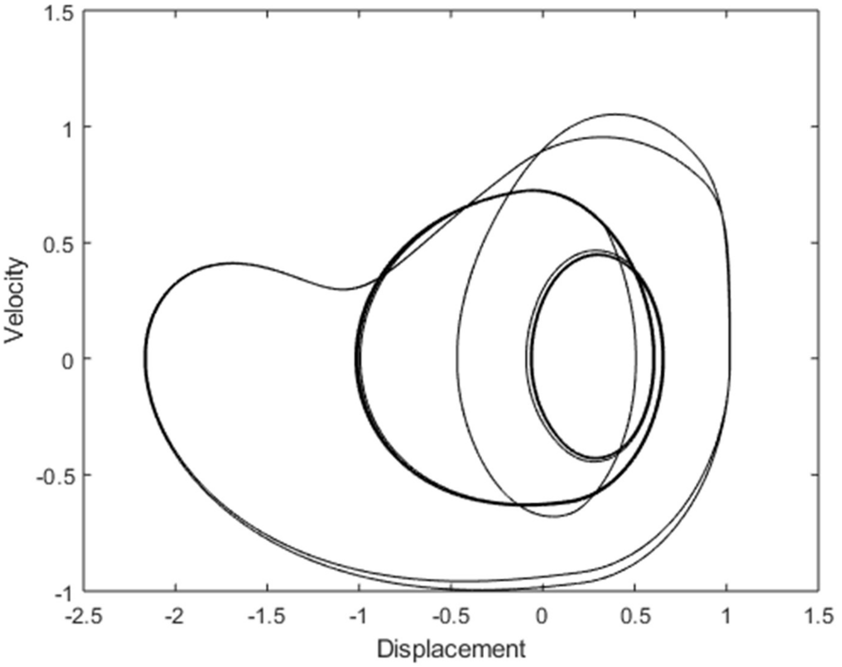


Figure 4.10 Trajectory of the system state when the constrained PD-like controller was applied

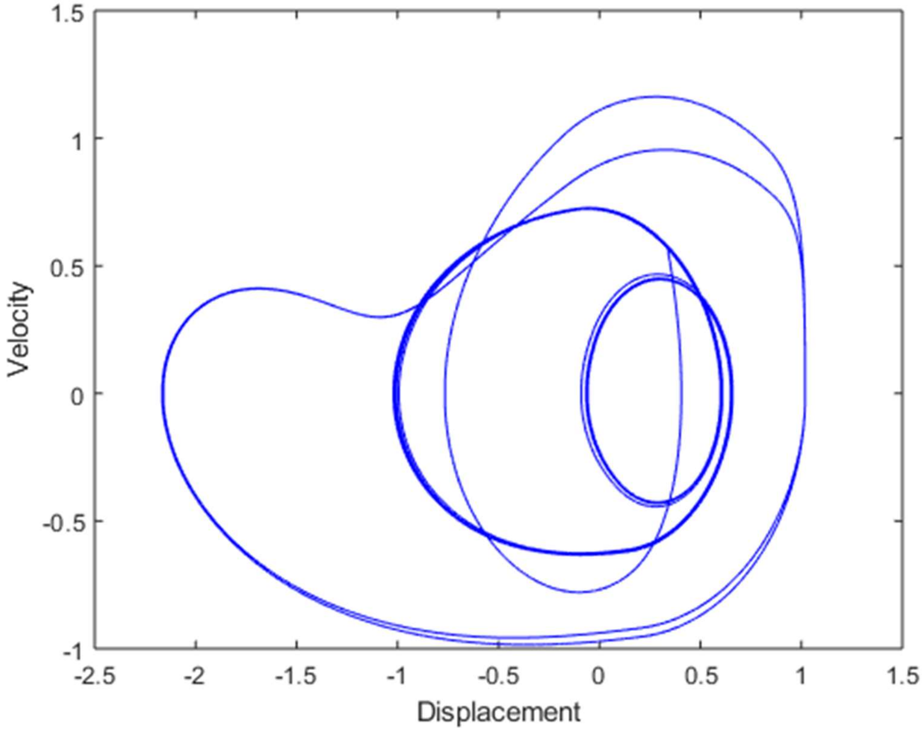


Figure 4.11 Trajectory of the system state when the constrained PD-like controller was applied

4.3 Summary

In this chapter the constrained PD-like control was proposed to limit the strength of the control input. Numerical simulation results on the systems namely Duffing oscillator, soft impact oscillator and soft impact oscillator with a drift demonstrated that the constrained PD-like control can switch the systems from current, undesired state to the desired state successfully. According to the results shown in this chapter, the strength of the control input is within the defined bounds while the switching duration of the controller increases. Moreover, if the control input is large enough the system will be driven to the desired state immediately. Otherwise, the system will be switched to desired state through a few intermediate stages. Performance of the proposed PD-like controller will be optimised in Chapters 6 and 7.

Chapter 5: Sensitivity Analysis of Parameters of PD-like Controller

In the previous chapter, the constrained PD-like controller is proposed to limit the strength of the control input. Its performance is dependent on the control parameters such as positive control gains k_p, k_d and the switching time. Hence, identifying the most appropriate set of control parameters can help to optimise the PD-like controllers' performance. In this chapter, more sets of control parameters are employed to investigate the performances of the controller using the three test systems, namely Duffing oscillator, soft impact oscillator and soft impact oscillator with a drift, described in Section 1.3.

5.1 Numerical Simulation

5.1.1 Duffing Oscillator

Two different sets of control parameters are used to investigate the performance of the controller applied on the Duffing oscillator. The performance of the controller is described by two indices namely the maximum peak of external control force and the switching duration and these two indices can be calculated by the developed MATLAB code. The maximum peak of external control force is the absolute maximum value of the external control input. Assume the controller is switched off when the distance between the current trajectory and the desired trajectory is small than 10^{-4}. The switching duration is defined as the time instant when the controller is switched off minus the time instant when the controller is switched on. Figures 5.1 and 5.2 show respectively the displacement response of Duffing oscillator and the control input of the controller with the first set of control parameters $k_p = 1$, $k_d = 1$ and the controller is switched on at $t = 1$ second. The maximum peak of the external control force is 1.76 and the switching duration is 15.15 seconds. Figures 5.3 and 5.4 show respectively the controller with the second set of control parameters $k_p = 1$, $k_d = 1$ and the controller is switched on at $t = 1.8$ seconds. The maximum peak of the external control force is 1.13 and the switching duration is 14.24 seconds. From these two tests, with a small change of one control parameter namely switched on time, the performance of the controller can be changed significantly. Therefore, in this chapter the performance of the controller is investigated under different set of the control parameters. Moreover, the performance of the controller in terms of maximum peak of external control force and switching duration of each test are recorded.

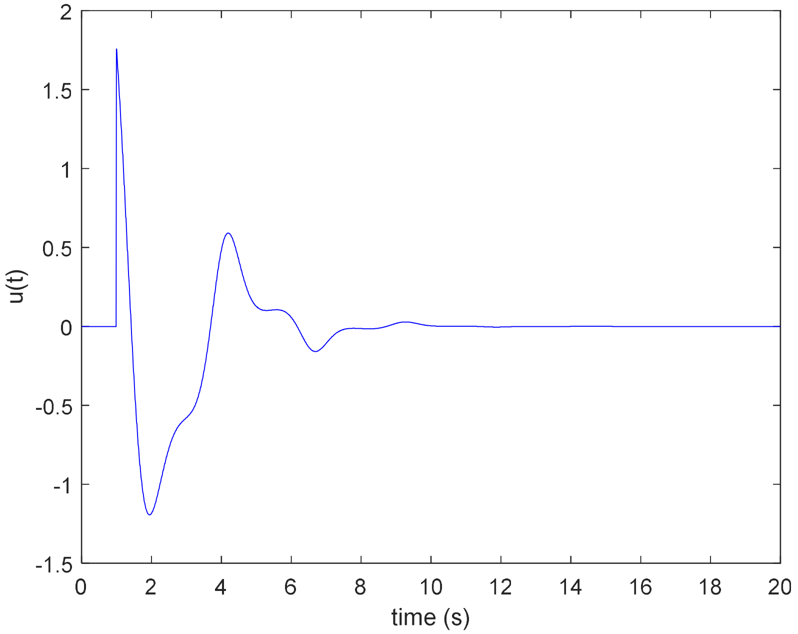


Figure 5.1 The applied control force as a function in time with the control parameters $k_p = 1$, $k_d = 1$ and the controller is switched on at $t = 1$ second when PD-like controller applied on the Duffing oscillator

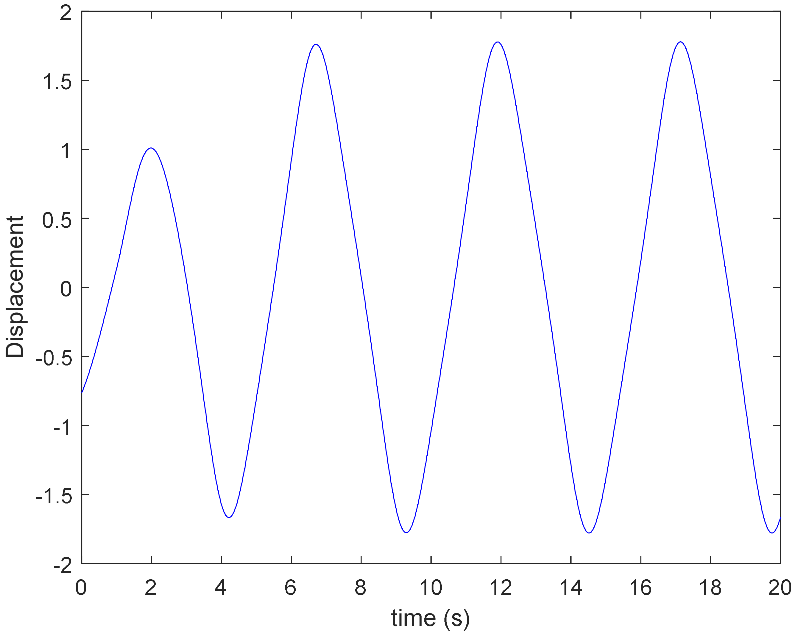


Figure 5.2 Displacement of the mass as a function in time with the control parameters $k_p = 1$, $k_d = 1$ and the controller is switched on at $t = 1$ second when PD-like controller applied on the Duffing oscillator

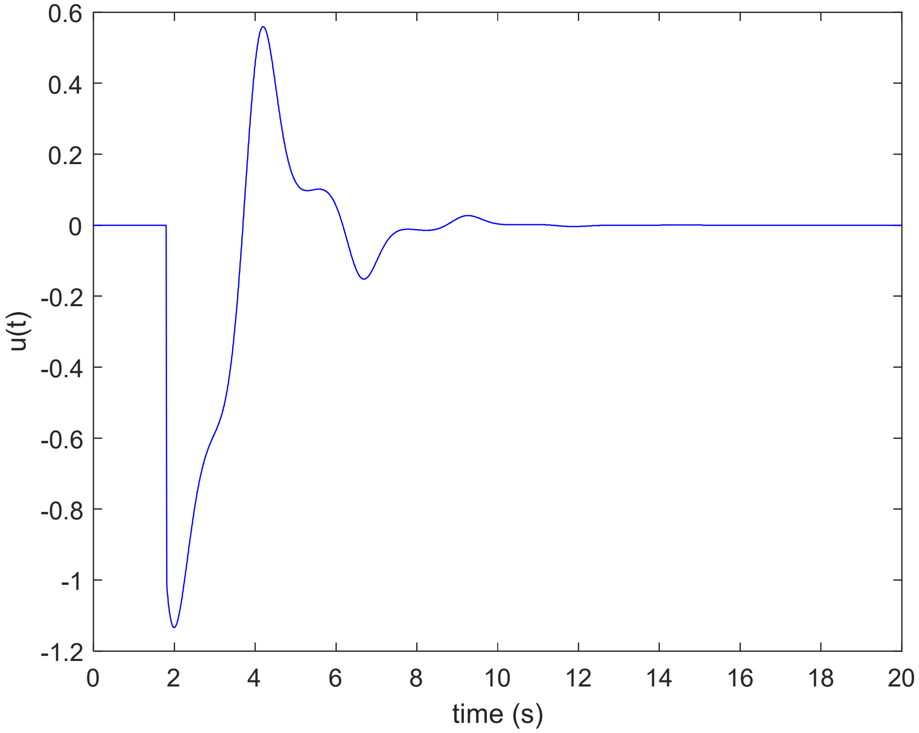


Figure 5.3 The applied control force as a function in time with the control parameters $k_p = 1$, $k_d = 1$ and the controller is switched on at $t = 1.8$ seconds when PD-like controller applied on the Duffing oscillator

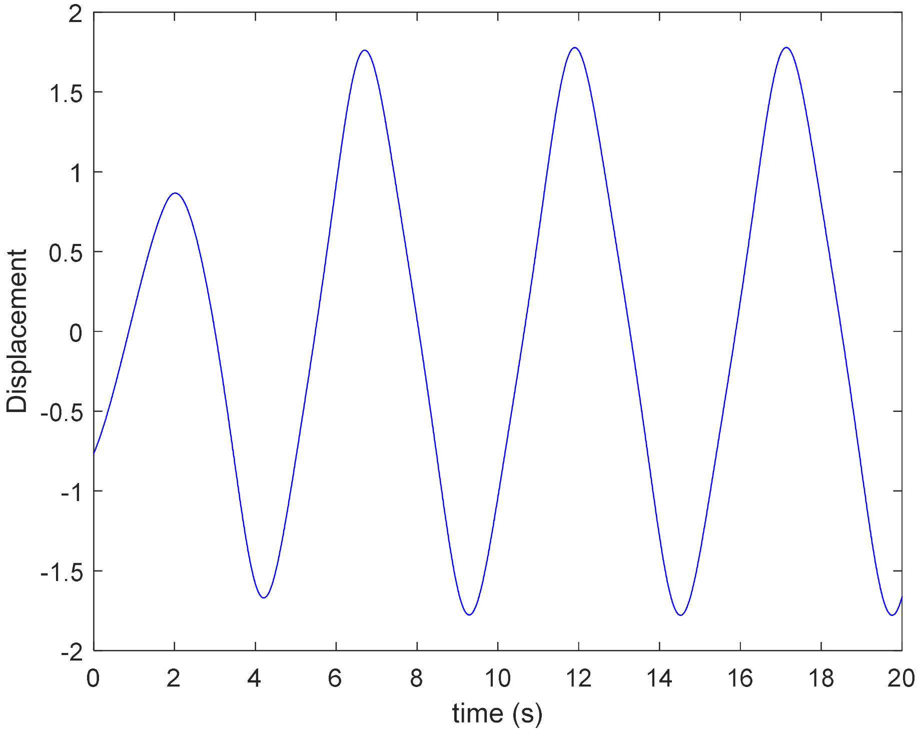


Figure 5.4 Displacement of the mass as a function in time with the control parameters $k_p = 1$, $k_d = 1$ and the controller is switched on at $t = 1.8$ seconds when PD-like controller applied on the Duffing oscillator

Figures 5.5 and 5.6 show the maximum peak of nondimensional control force and switching duration of the Duffing oscillator with different k_p, k_d, and the controller is switched on at 1.05 seconds. In Figure 5.5, the maximum peak of control force ranges from 0.27 to 16.73 with k_p and k_d varying from 0 to 10. With the increasing k_p and k_d, the maximum peak of control force increases. In Figure 5.6, the switching duration ranges from 2.87 seconds to 139.88 seconds. The switching duration decreases with increasing k_d. The switching duration is long when both k_p and k_d are less than 1 or k_p is small and k_d is large. The switching duration is short when k_p is approximately 5.5 and k_d is around 8.5.

Figures 5.7 and 5.8 show the maximum peak of control force and switching duration with different switching times, and the control parameters are set as $k_p = 1$ and $k_d = 1$. In Figure 5.7, the maximum peak of control force varies from 1.01 to 1.91 with the switch on time ranging from 0 seconds to 5.24 seconds (one period of the system response). Figure 5.7 illustrates that there are two periods of the maximum peak of control force when the switching time of the controller varies from 0 seconds to 5.24 seconds. In one of the periods, the maximum peak of control force varies as the switching time is increased. Similarly, the switching duration periodically varies with increasing the switch on time. Moreover, the switching duration varies from 14.24 seconds to 15.35 seconds only.

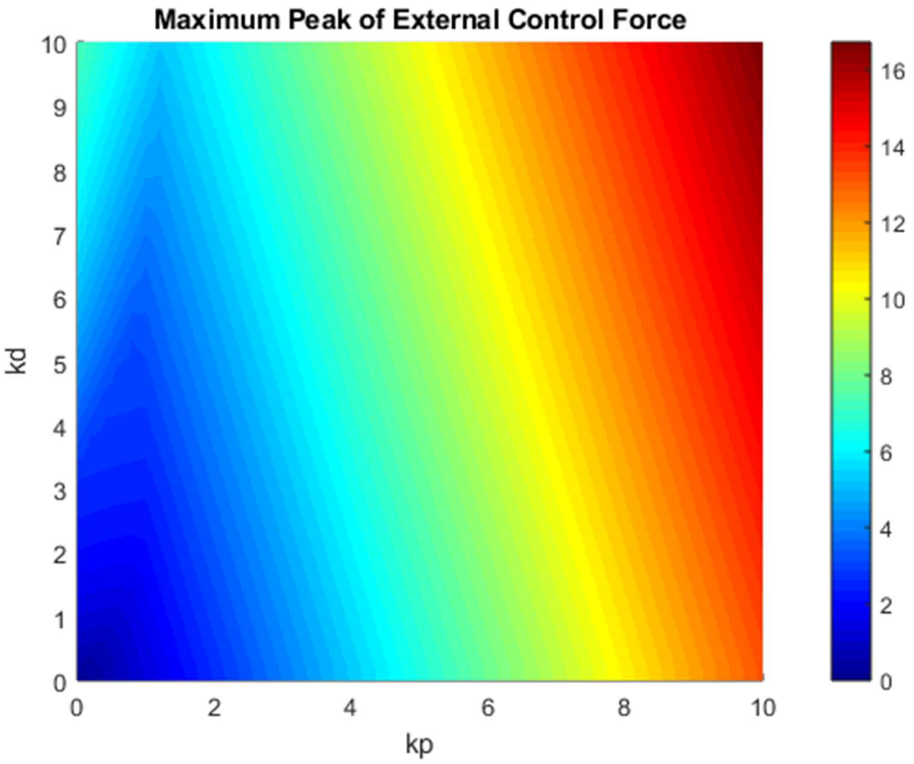


Figure 5.5 The simulation result of applying the PD-like controller on Duffing oscillator with different k_p and k_d. The colour bars show the magnitude of the maximum peak of control force

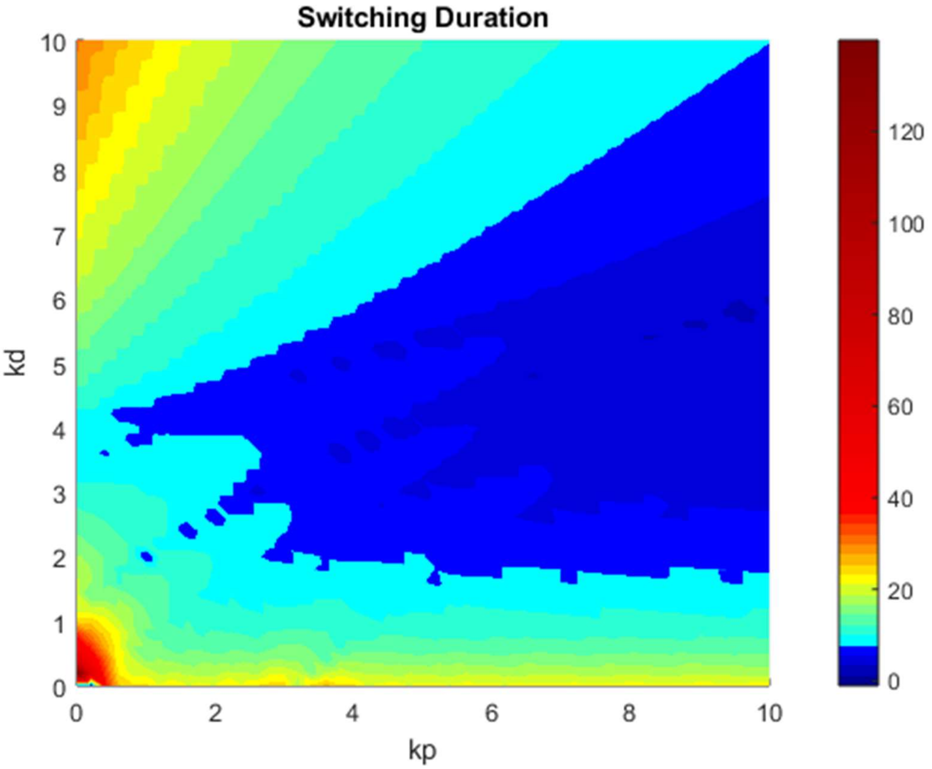


Figure 5.6 The simulation result of applying the PD-like controller on Duffing oscillator with different k_p and k_d. The colour bars show the magnitude of switching duration.

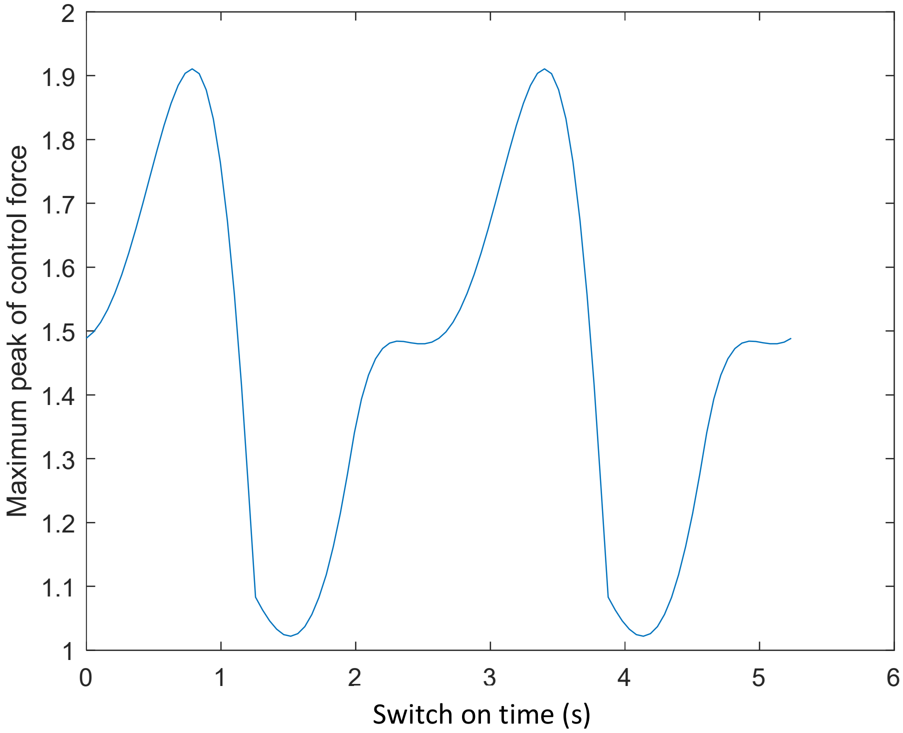


Figure 5.7 The maximum peak of control force when the PD-like controller is applied on Duffing oscillator with different switch on time.

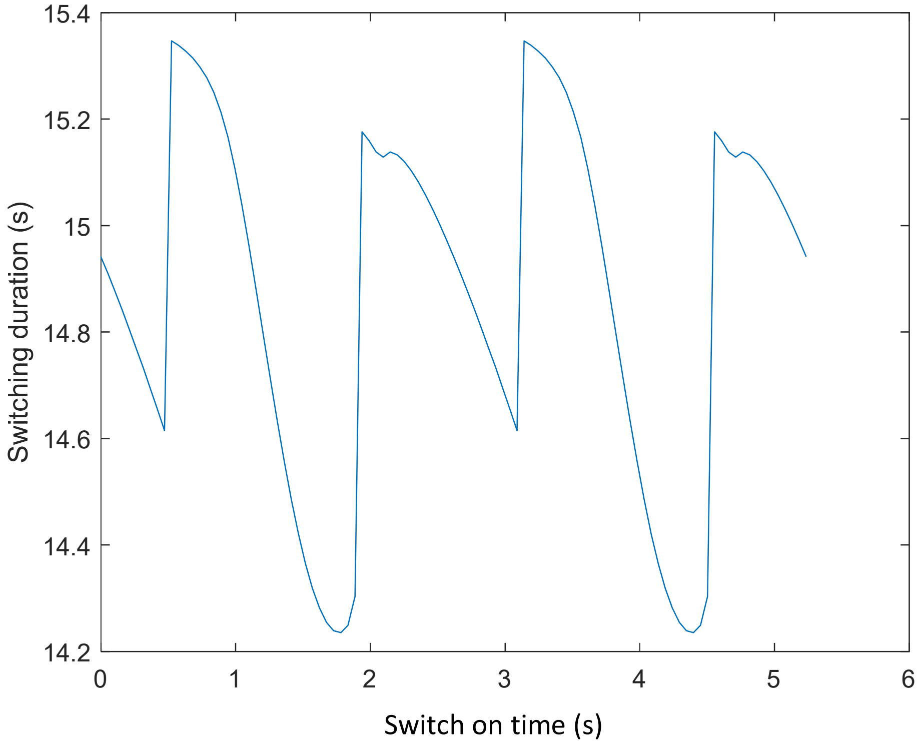


Figure 5.8 The switching duration when the PD-like controller is applied on the Duffing oscillator with different switching time.

5.1.2 Soft impact oscillator

Figures 5.9 and 5.10 show the maximum peak of nondimensional control force and switching duration of the soft impact oscillator with different k_p and k_d. The controller is switched on at 3.05 seconds, and both k_p and k_d increase from 0 to 10. The maximum peak of control force and switching duration varies from 1.12 to 32.08, and from 3.01 seconds to 143.35 seconds, respectively. The maximum peak of the control force is defined as 0 if the controller cannot drive the system to the desired state. As depicted in Figure 5.9, the controller cannot drive the system to the desired state when k_d is small. Moreover, the maximum peak of control force is large when k_p is small and k_d is large. When both k_p and k_d are small, the maximum peak of control force decreases. The switching duration is the smallest when k_p is around 5.8 and k_d is around 8.5. The switching duration is long when k_p is small. Figures 5.11 and 5.12 depict the maximum peak of control force and switching duration with increasing switch on time respectively. The switch on time varies from 0 seconds to 9.16 seconds and the control parameters are set at $k_p = 1$ and $k_d = 1$. The maximum peak of control force varies from 2.18 to 5.85, and the switching duration ranges from 17.65 seconds to 24.36 seconds. Moreover, the fluctuations of both maximum peak of control force and switching duration are irregular.

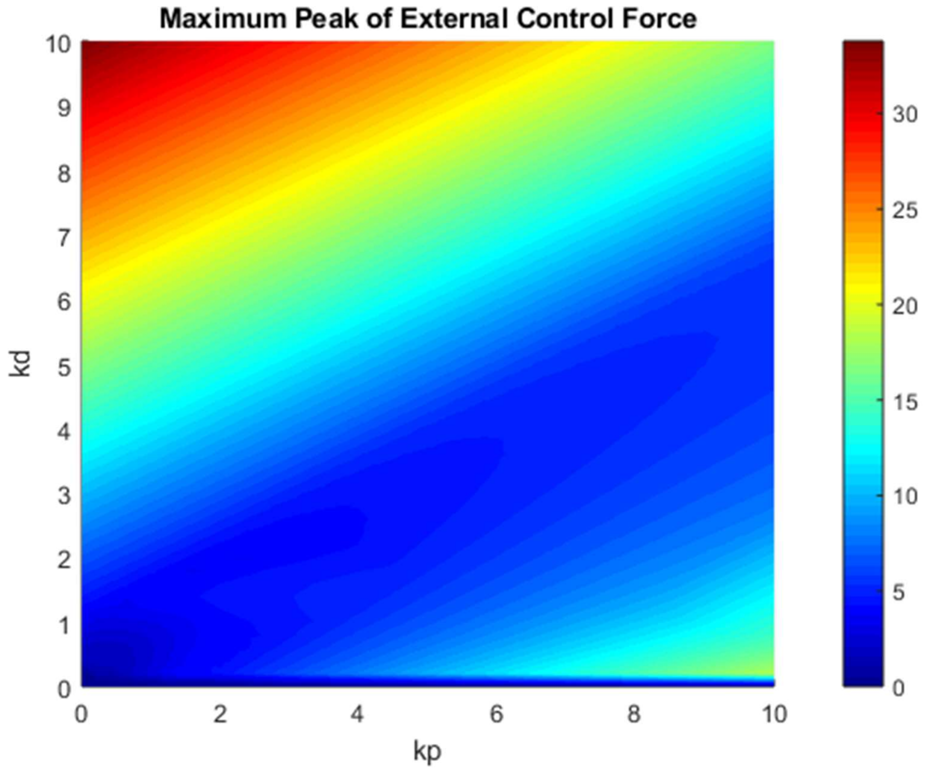


Figure 5.9 The simulation result of apply the PD controller on soft impact oscillator with different k_p and k_d. The colour bars show the magnitude of the maximum peak of control force

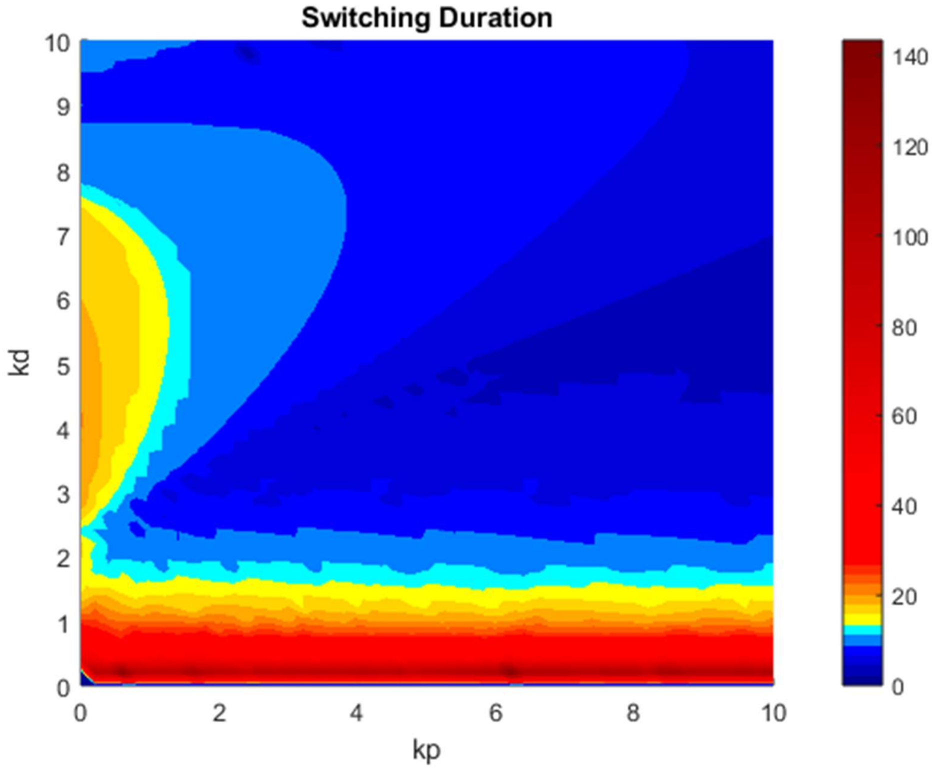


Figure 5.10 The simulation result of applying the PD-like controller on soft impact oscillator with different k_p and k_d. The colour bars show the magnitude of switching duration.

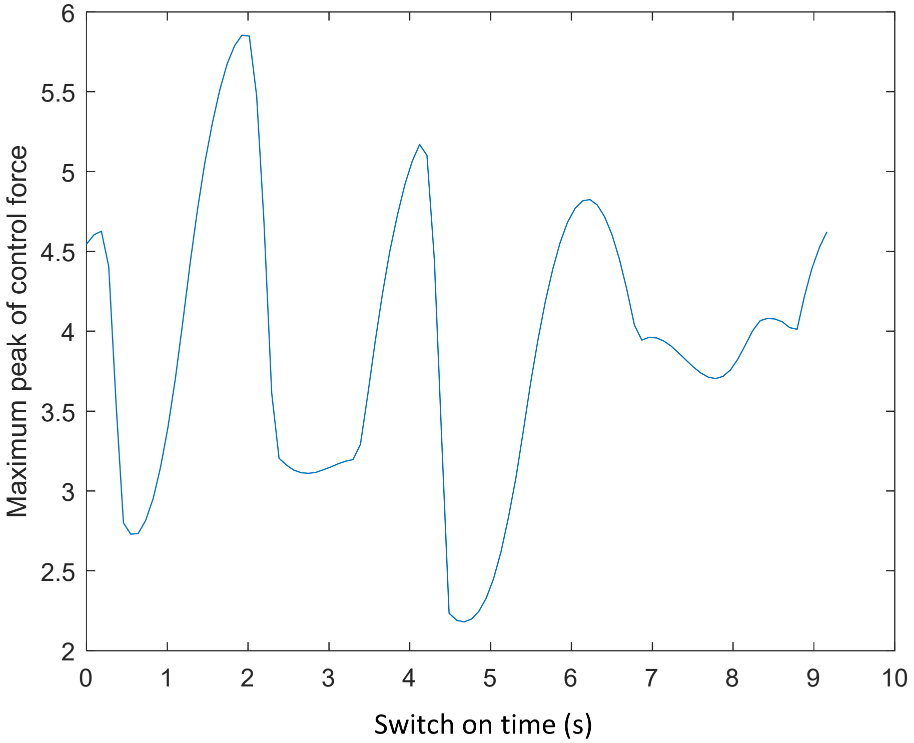


Figure 5.11 The maximum peak of control force when the PD-like controller is applied on soft impact oscillator with different switching time.

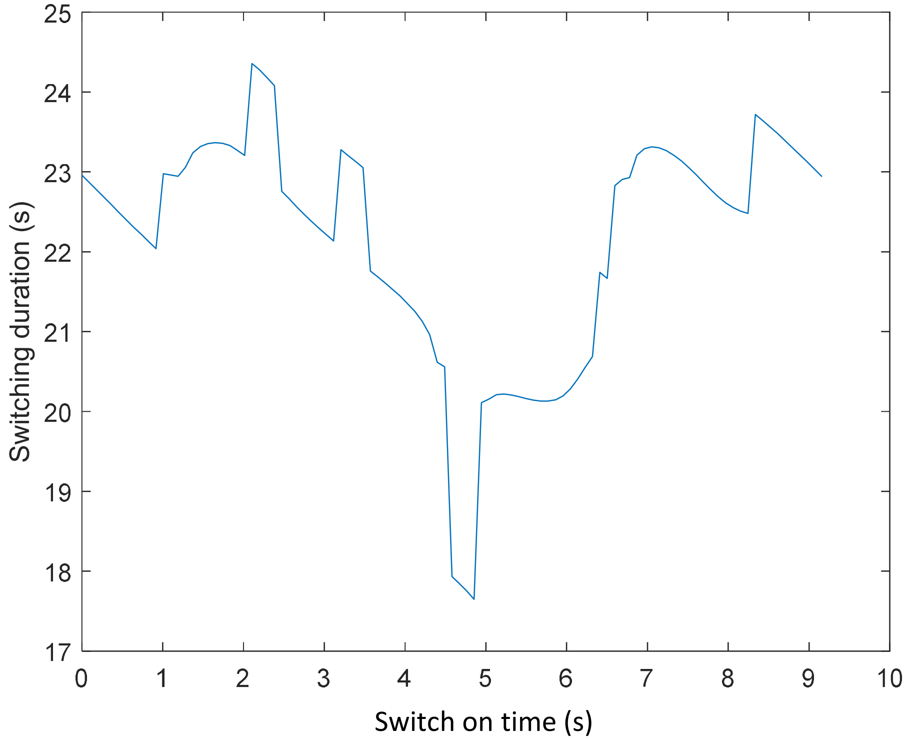


Figure 5.12 The switching duration when the PD-like controller is applied on soft impact oscillator with different switching time.

Based on Figures 5.9 and 5.10, three sets of control parameters are selected. In all of the three tests, the controller is switched on at $t = 3.05$ seconds. Figures 5.13 and 14 show the displacement response of the soft impact oscillator and the external control force of the controller with the first set of controller parameters: $k_p = 3$ and $k_d = 3$. The maximum peak of the external control force is 5.02 and the switching duration is 6.45 seconds. Figures 5.15 and 16 show respectively the displacement response of the soft impact oscillator and the external control force of the controller with the first set of controller parameters: $k_p = 7.8$ and $k_d = 4.6$. The maximum peak of the external control force is 5.07 and the switching duration is 4.69 seconds. Figures 5.17 and 18 show the displacement response of the soft impact oscillator and the external control force of the controller with the first set of controller parameters: $k_p = 9$ and $k_d = 8.8$. The maximum peak of the external control force is 14.39 and the switching duration is 6.63 seconds. The performance of the controller can be described by the maximum peak of the external control force and the switching duration. Therefore, by comparing these three tests, with different sets of the control parameters, the performance of the controller may be similar in both aspect of the performance or it may be similar in only one aspect of the performance. Then based on Figures 5.11 and 12, another two sets of control parameters are selected. For both sets, $k_p = 1$ and $k_d = 1$. Figures 5.19 and 20 show the displacement response of the soft impact oscillator and the external control force of the controller and the controlled is switched at 2.11 seconds. The maximum peak of the external control force is 5.47 and the switching duration is 24.36 seconds. Figures 5.21 and 22 show the displacement response of the soft impact oscillator and the external control force of the controller and the controller is switched on at 4.85 seconds. The maximum peak of the external control force is 2.25 and the switching duration is 17.65 seconds. Comparing these two tests, with the same k_p and k_d, when the switch on time changes, the performance of the controller also changes. Therefore, according to these five tests, the performance of the controller is complicated based on variations of k_p, k_d and the switch on time.

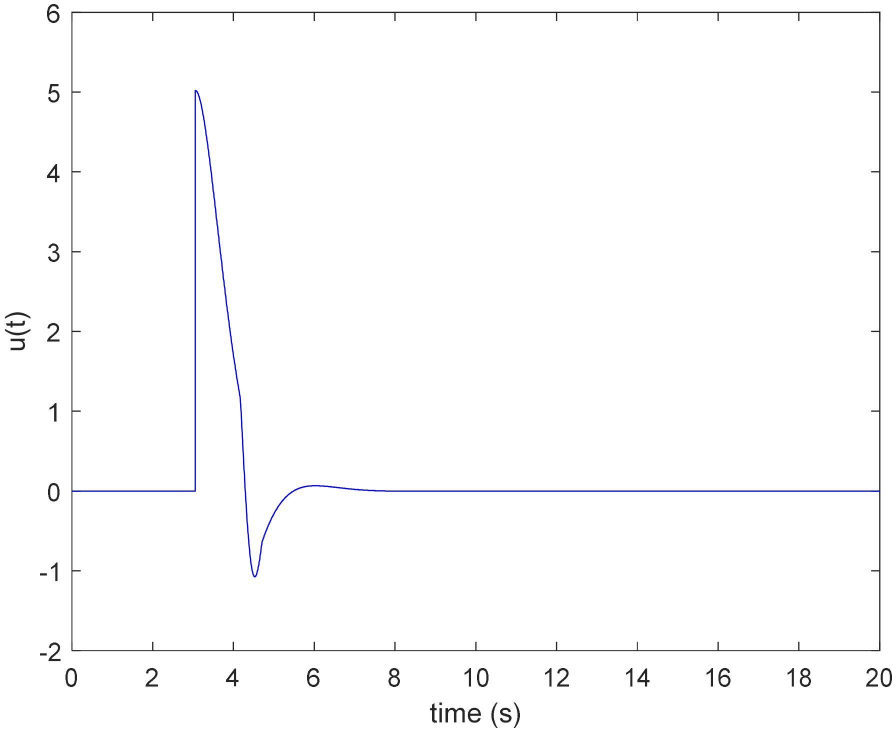


Figure 5.13 The applied control force as a function in time with the control parameters $k_p = 3$, $k_d = 3$ *and the controller is switched on at* $t = 3.05$ *seconds when PD-like controller applied on the soft impact oscillator*

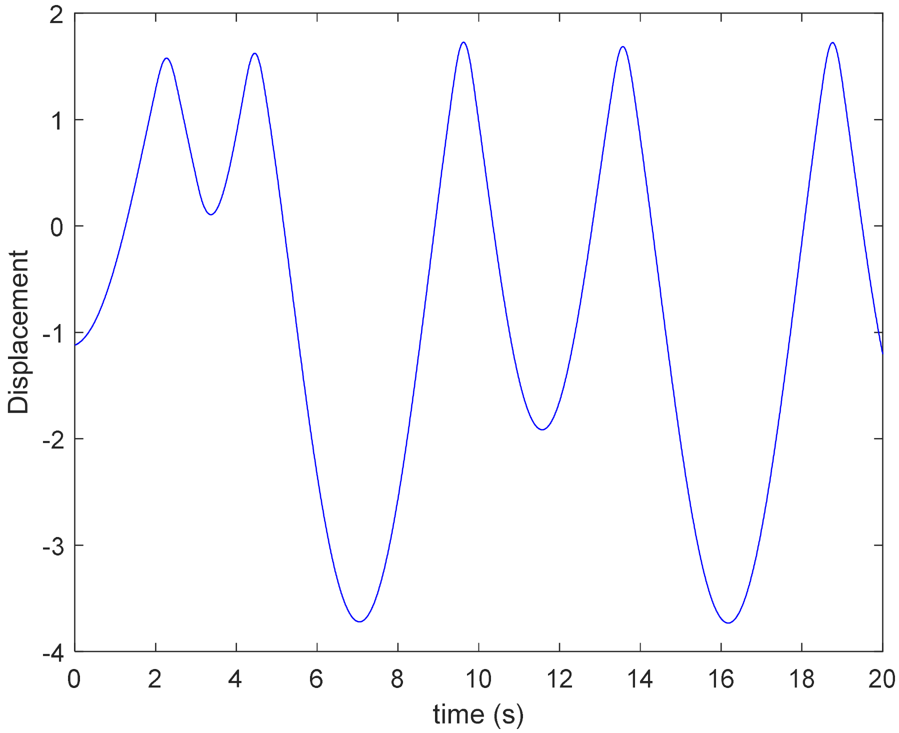


Figure 5.14 Displacement of the mass as a function in time with the control parameters $k_p = 3$, $k_d = 3$ *and the controller is switched on at* $t = 3.05$ *seconds when PD-like controller applied on the soft impact oscillator*

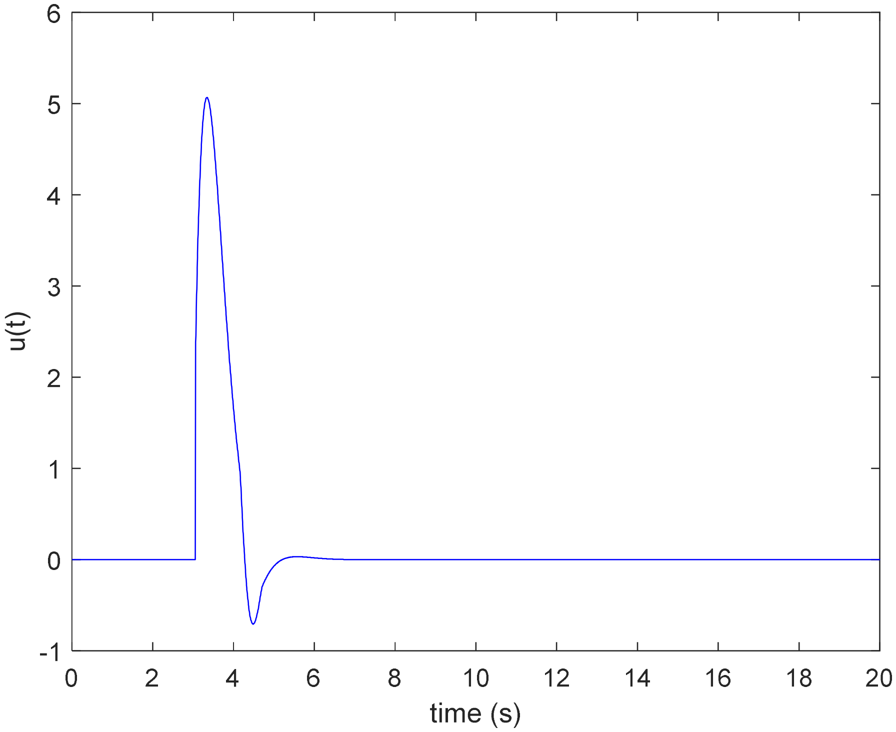


Figure 5.15 The applied control force as a function in time with the control parameters $k_p = 7.8$, $k_d = 4.6$ *and the controller is switched on at* $t = 3.05$ *seconds when PD-like controller applied on the soft impact oscillator*

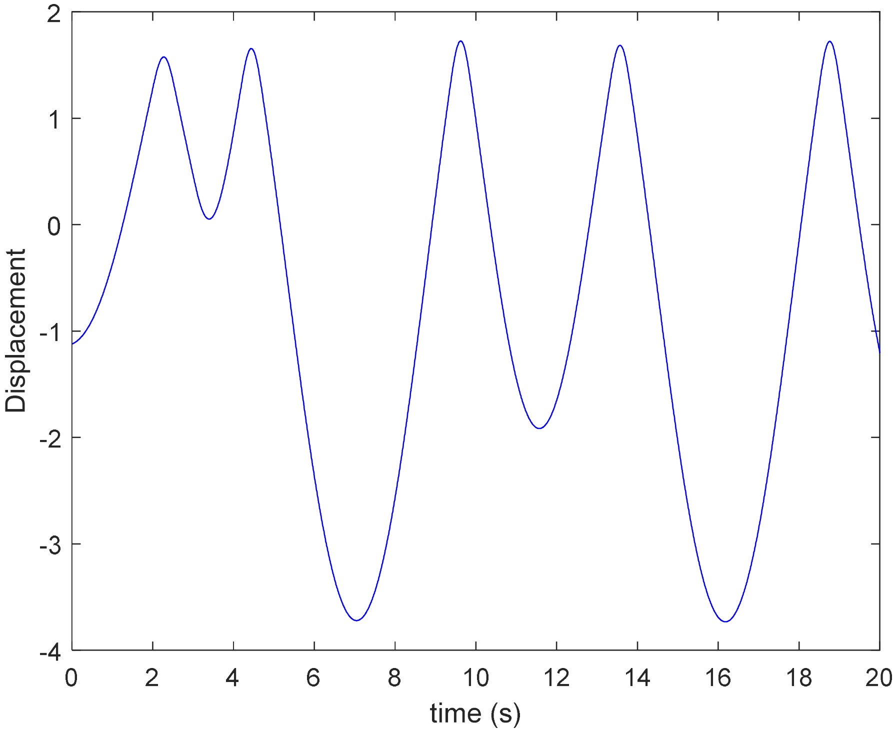


Figure 5.16 Displacement of the mass as a function in time with the control parameters $k_p = 7.8$, $k_d = 4.6$ *and the controller is switched on at* $t = 3.05$ *seconds when PD-like controller applied on the soft impact oscillator*

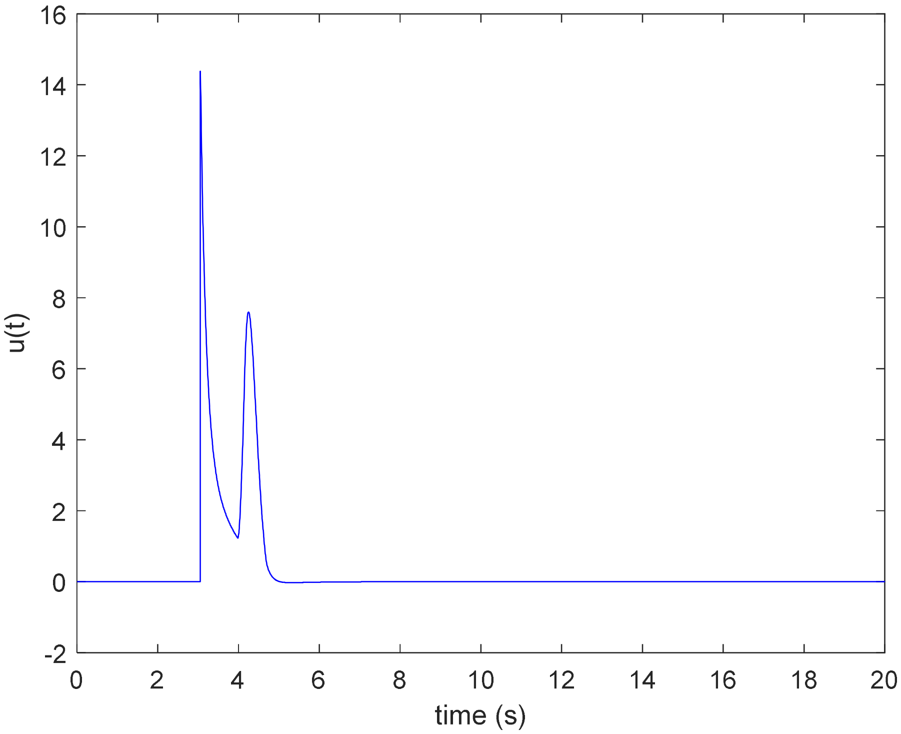


Figure 5.17 The applied control force as a function in time with the control parameters $k_p = 9$*,* $k_d = 8.8$ *and the controller is switched on at* $t = 3.05$ *seconds when PD-like controller applied on the soft impact oscillator*

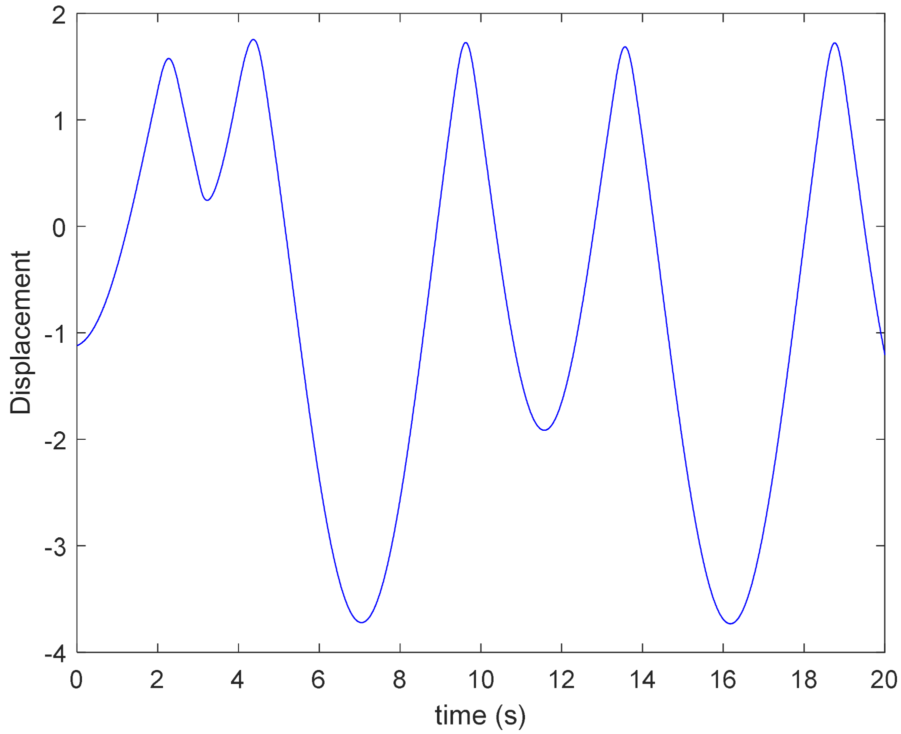


Figure 5.18 Displacement of the mass as a function in time with the control parameters $k_p = 9$*,* $k_d = 8.8$ *and the controller is switched on at* $t = 3.05$ *seconds when PD-like controller applied on the soft impact oscillator*

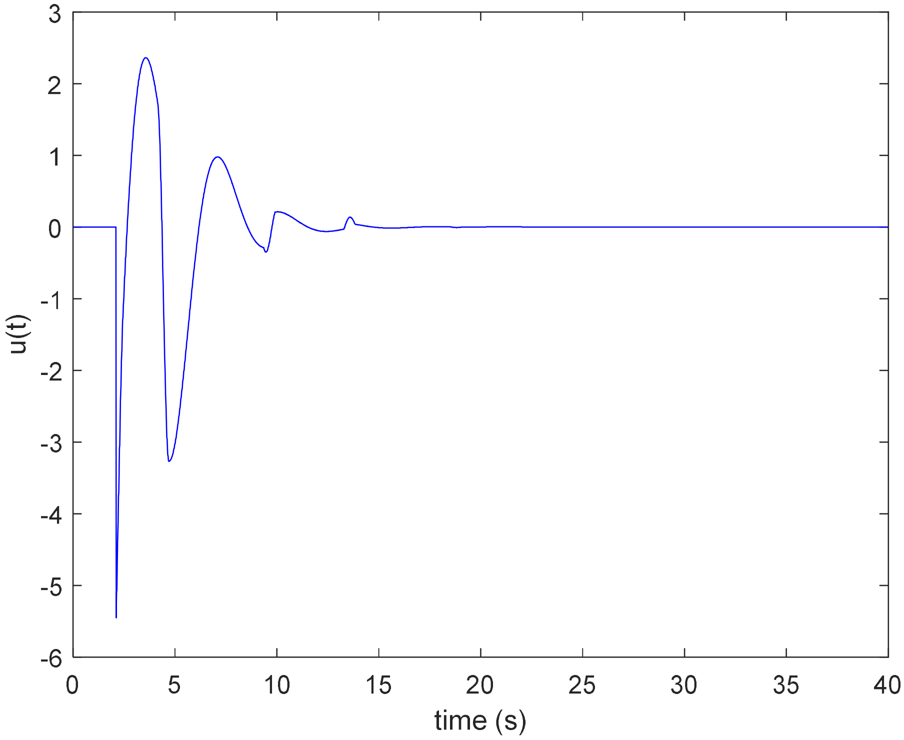


Figure 5.19 The applied control force as a function in time with the control parameters $k_p = 1$, $k_d = 1$ and the controller is switched on at $t = 2.11$ seconds when PD-like controller applied on the soft impact oscillator

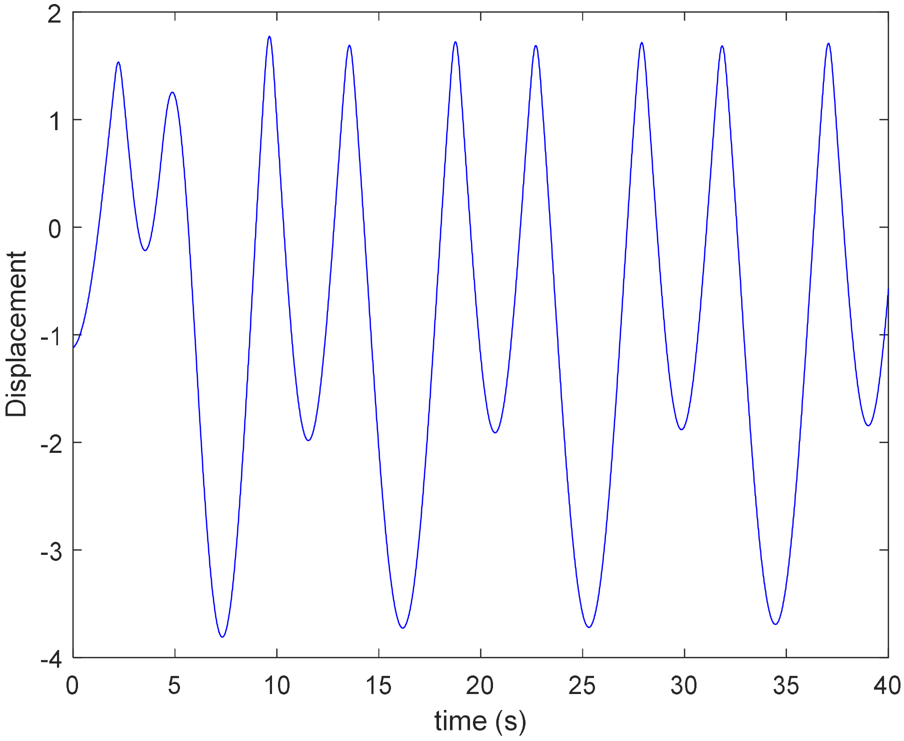


Figure 5.20 Displacement of the mass as a function in time with the control parameters $k_p = 1$, $k_d = 1$ and the controller is switched on at $t = 2.11$ seconds when PD-like controller applied on the soft impact oscillator

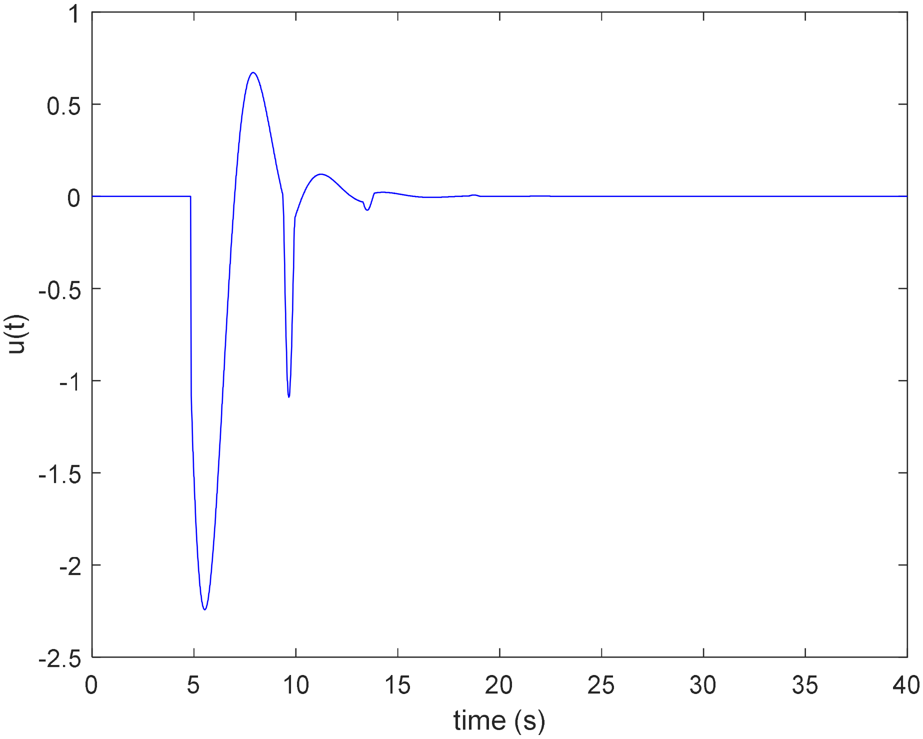


Figure 5.21 The applied control force as a function in time with the control parameters $k_p = 1$, $k_d = 1$ and the controller is switched on at $t = 4.85$ seconds when PD-like controller applied on the soft impact oscillator

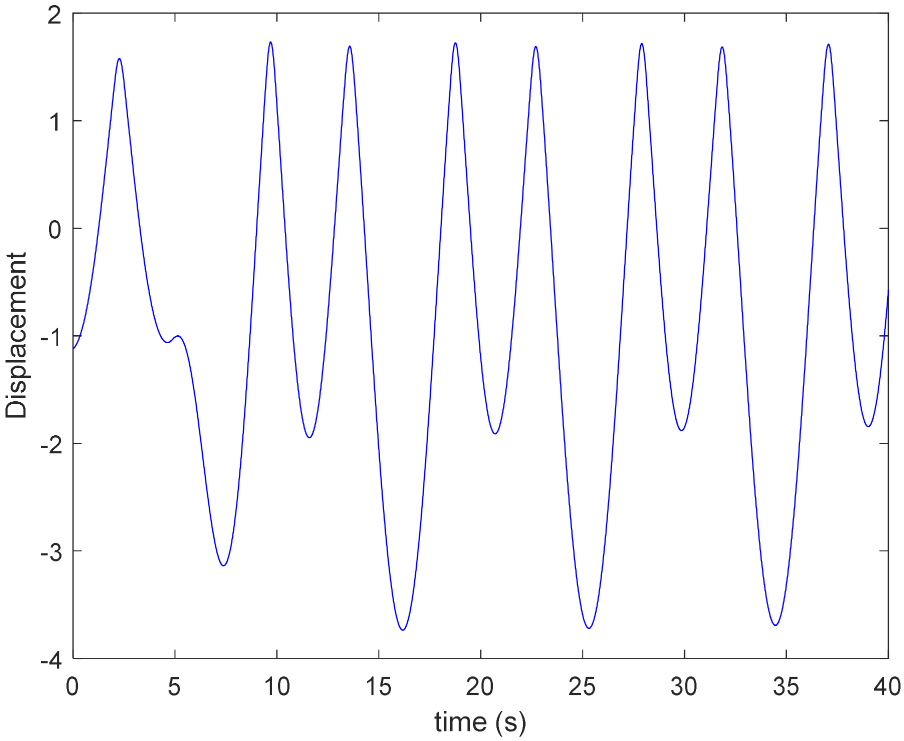


Figure 5.22 Displacement of the mass as a function in time with the control parameters $k_p = 1$, $k_d = 1$ and the controller is switched on at $t = 4.85$ seconds when PD-like controller applied on the soft impact oscillator

5.1.3 Soft impact oscillator with a Drift

Figures 5.23 and 5.24 present respectively the maximum peak of nondimensional control force and switching duration of the soft impact oscillator with a drift, with different sets of k_p and k_d. k_p and k_d range from 0 to 10, and the controller is applied at 1.05 seconds. The maximum peak of control force varies from 1.04 to 20.83. The larger the k_p and k_d, the higher the maximum peak of control force. The switching duration ranges from 2.04 seconds to 157.11 seconds. The switching duration appears minimum when k_p is around 9.4 and k_d is round 5.2. The switching duration is long when k_p is large or k_d is small. Figures 5.25 and 5.26 show the maximum peak of control force and switching duration, with increasing switch on times. The switch on time varies from 0 seconds to 6.28 seconds (for one period), and the control parameters are set as $k_p = 1$ and $k_d = 1$. The maximum peak of control force and switching duration varies from 0.39 to 2.11 and from 10.45 seconds to 16.14 seconds respectively. The maximum peak of control force varies smoothly with the switch on time. However, the switching duration varies irregularly with the switching time.

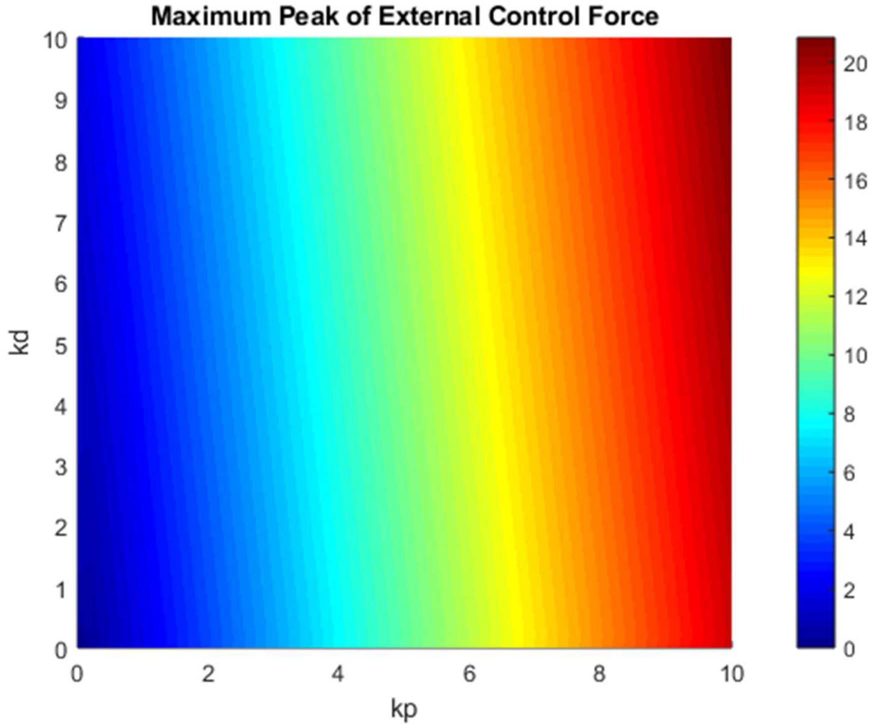


Figure 5.23 The simulation result of apply the PD-like controller on soft impact oscillator with a drift with different k_p and k_d. The colour bars show the magnitude of the maximum peak of control force

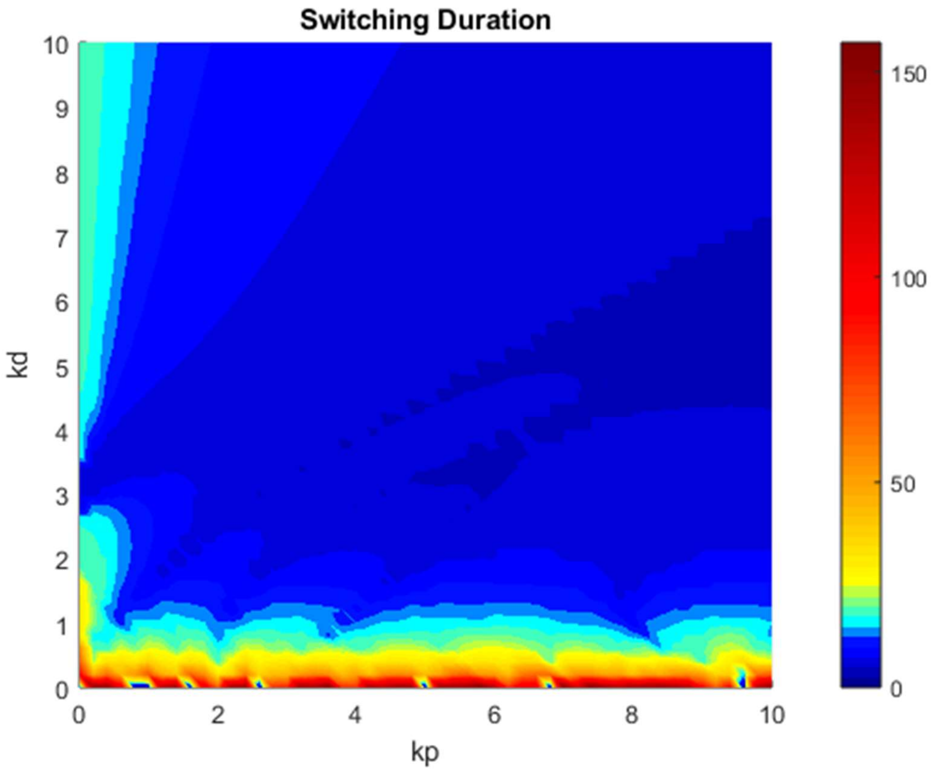


Figure 5.24 The simulation result when the PD-like controller is applied on soft impact oscillator with a drift with different k_p and k_d. The colour bars show the magnitude of switching duration.

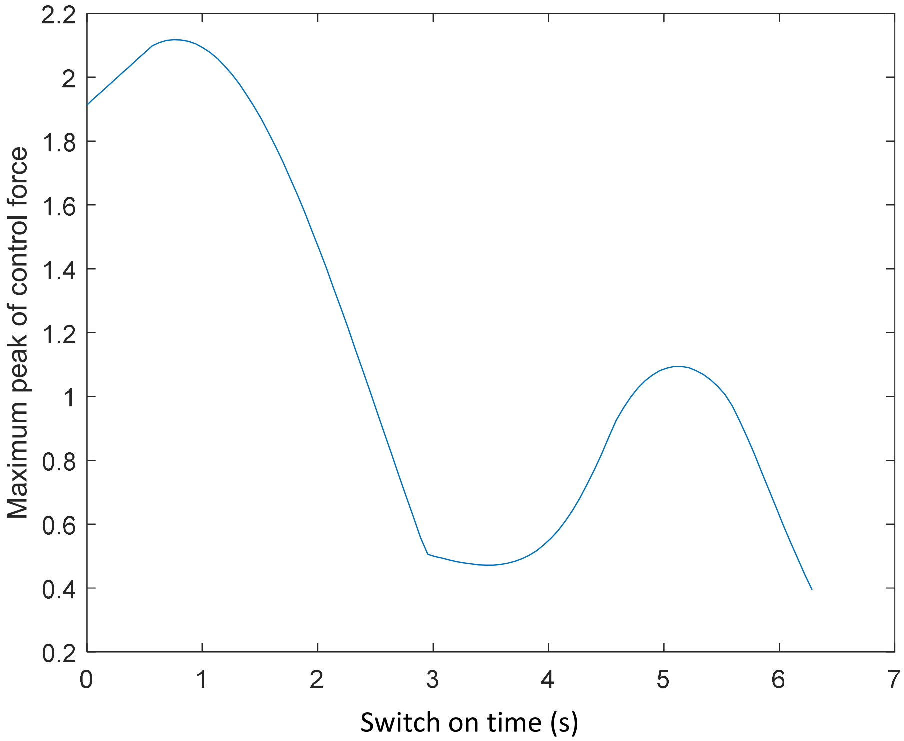


Figure 5.25 The maximum peak of control force when the PD-like controller is applied on soft impact oscillator with a drift with different switching time.

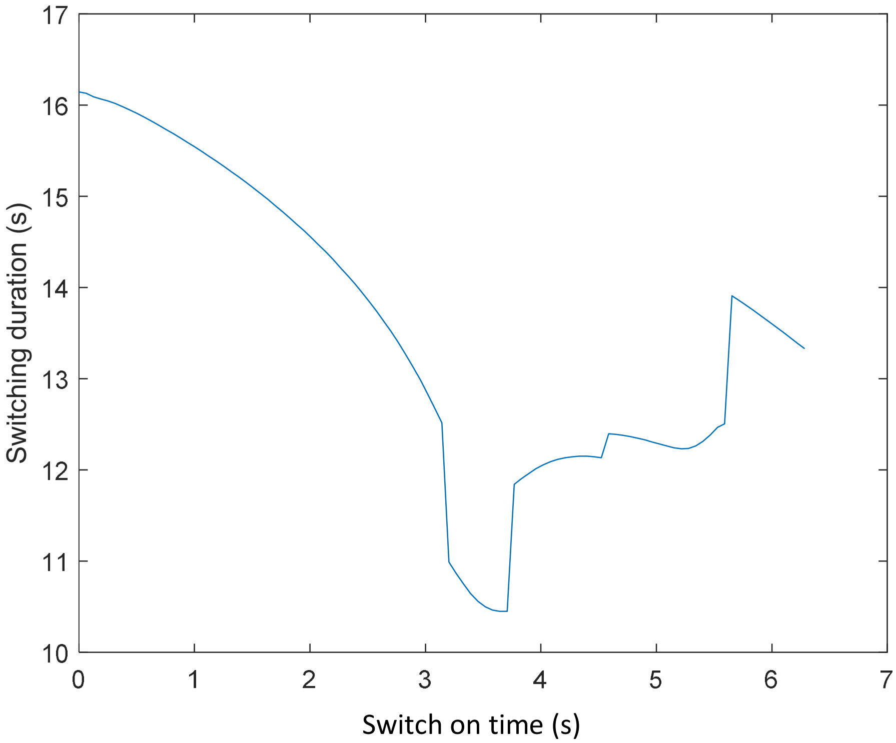


Figure 5.26 The switching duration of when PD-like controller is applied on soft impact oscillator with a drift with different switching time.

5.2 Summary

Overall, both the maximum peak of control force and the switching duration in multistable state switching control are sensitive to the positive control gains k_p, k_d and switch on time. Therefore, the choice of appropriate control parameters can minimise the maximum peak of control force and switching duration through optimisation algorithms.

Chapter 6 Single Objective Optimisation of the PD-like Controller

The performance of the proposed PD-like controller is sensitive to the parameters, including the gains and switch on time of the controller. Therefore, its performance can be improved by carefully selecting its parameters using optimisation methods. In this chapter, two objectives namely the maximum peak of the external control force and switching duration are minimised independently for the proposed PD-like controller. Performance optimisation of the proposed PD-like controller is conducted on the three test systems namely the Duffing oscillator, the soft impact oscillator and the soft impact oscillator with a drift.

6.1 Introduction

In real-life situations, designers not only want to achieve the goal of the work but also want to achieve the goal without unnecessary waste of power and time [75]. The discipline of adjusting a process in order to achieve the best design is known as optimisation. For example, the PD-like controller can switch the multistable system from undesired to desired state. However, the designer wants to switch the system to the desired state as soon as possible (with minimum switching duration). Different performance metrics are defined to quantify the system performance in various aspects for performance evaluation of the design.

They can then be transformed into objectives in optimisation problems and they can either be minimised or maximised. For example, in order to minimise the switching duration of the PD-like controller, the method for determining switching duration should be empirically defined or rigorously defined in mathematical functions in terms of optimising variables. Similarly, maximum peak of control input $u(t)$ of the PD-like controller is another performance metric used to evaluate the energy consumption of the system. Secondly, the system variables which affects the system responses of interest should be defined. For the PD-like controller, the control gains and the switch on time will affect the performance (switching duration and maximum peak of control input) of the controller. Thirdly, constraints involved with the optimising variables in the optimisation should also be identified. Physical constraints in system design are modelled as equality or inequality constraints in optimisation problems. In the case of optimising the PD-like controller, the constraint is related to the output capacity limit of the actuator for state switching in a multistable system.

6.1.1 Single Objective Optimisation

In general, a single optimisation problem is formulated as follows [76]:

Definition 6.1:

$$\begin{aligned} \text{Minimise:} \quad & obj(x) \\ \text{Subject to:} \quad & g_i(x) \leq 0, \quad i = 1, \dots, m \\ & h_j(x) = 0, \quad j = 1, \dots, w \\ & x = [x_1, x_2, \dots, x_n]^T \\ & x_k^l \leq x_k \leq x_k^u, \quad k = 1, n \end{aligned} \tag{6.1}$$

Where $obj(x)$ is the objective function to optimise, $g_i(x)$ is the $i-$th inequality constraint function, m is the number of inequality constraint functions, $h_j(x)$ is the $j-th$ equality constraint function, w is the number of equality constraint functions, x is the vector of independent optimising variables, x_k is the k-th optimising variable, n is the number of the optimising variables, x_k^l and x_k^u are the lower bound and upper bound of the k-th optimising variable. A maximisation problem can be easily converted into a minimisation problem as follows:

$$Max\ obj(x) \equiv Min\ -obj(x)\ or\ \frac{1}{obj(x)} \tag{6.2}$$

6.1.2 Single Objective Optimisation Algorithms

There are several methods of solving single objective optimisation problems. For example, dynamic programming (DP) solves deterministic optimisation problems by dividing them into simple sub-problems or stages [77-83]. However, it is not feasible in all instances because some problems which cannot be divided into multiple stages. Another kind of methods can be used for optimisation is called evolutionary algorithms (EAs). EAs are proposed based on the biological evolution and they emulate different natural operations such as selection, crossover, mutation and recombination in optimal solution searching. Compared with DP, EAs have some advantages like self-learning, self-adaptation and they can be applied to solve optimisation problems formulated as Definition 6.1. One common method of EAs is called Genetic Algorithm (GA) for global optimal solutions and it is based on survival of the fittest principle of Darwin’s evolution theory [84]. It solves the optimisation problems by using the selection, crossover and mutation operators [85-86].

Particle Swarm Optimisation (PSO) algorithm is another popular optimisation method and it was proposed by Kennedy and Eberhart based on the behaviour found within flocks of birds or schools of fish [87]. The main idea of PSO is information sharing between each particle in a population. Each particle in PSO represents a candidate solution to a given optimisation problem. Similar to Genetic Algorithm (GA), PSO is also an evolutionary algorithm which uses the position and movement of each particle to solve optimisation problems [88-89]. However, unlike GA, PSO does not require genetic operators like crossover and mutation to produce offspring solutions for next generation. It solves problems based on the best value of itself and its neighbourhood. PSO holds a range of advantages over the GA: firstly, PSO is easier to apply with fewer parameters introduced in the algorithm; secondly, the memory requirement of PSO is lower than that of GA; and thirdly, PSO has greater diversity than GA. With the help of the inertia of the particle movement, PSO can keep the diversity of the population. However, GA keeps the diversity by applying different mutation probability which is hard to determine. If the mutation probability is too small, GA may be easily trapped in a local best solution rather than converging to the global best solution. If the mutation probability is too large, the candidate solutions in each generation can be said as randomly selected [90].

Ant colony optimisation (ACO) is another common method of optimisation. This method is proposed by Dorigo in 1990 [107-108]. The principle of this method is based on the behaviour of ant colony. Ants can find the best path between their nest and the food. At first, the ants move from their nest to the food by different random paths. Before the ants return to the nest, they will leave pheromone on the paths which they have passed. The pheromone information on the path will be updated according to the evaporation rate after ants return to the nest. The pheromone information updated can be calculated as follow [107]:

$$\tau_{i,j} = (1-\rho)\tau_{i,j} + \sum_{k=1}^{m} \Delta\tau_{i,j}^{k}$$

$$\Delta\tau_{i,j}^{k} = \begin{cases} \frac{Q}{L_k} & k^{th}\ ant\ travels\ on\ the\ edge\ (i,j) \\ 0 & otherwise \end{cases} \tag{6.3}$$

where, $\tau_{i,j}$ is the amount of the pheromone laid on the edge (i,j), edge (i,j) means the curve between elements i and j, ρ is the evaporation rate, m is the number of ants, $\Delta\tau_{i,j}^{k}$ is the is the amount of the pheromone that the k^{th} ant laid on the edge (i,j), Q is the constant of the quantity of pheromone and L_k is the length of the path. Based on these updated pheromone level, the probability of an ant follows edge (i,j) can be calculated as follow:

$$P_{i,j} = \frac{\left(\tau_{i,j}\right)^{\alpha}(\eta_{i,j})^{\beta}}{\sum_{ih\in allowed_{ih}}(\left(\tau_{i,h}\right)^{\alpha}\left(\eta_{i,h}\right)^{\beta})} \quad i,j \in allowed_{ih} \tag{6.4}$$

where $\eta_{i,j}$ is the desirability of the edge (i,j), α and β are the parameters that reflects the importance of $\tau_{i,j}$ and $\eta_{i,j}$ respectively. $allowed_{ih}$ is the set of all the possible edges start from the element i. h is the element not yet visited by the k^{th} ant. $\tau_{i,h}$ and $\eta_{i,h}$ shows the pheromone level and desirability of other possible edges. Then, the pheromone updates will not stop unless all ants converge to the same path as the best path. Comparing with the PSO, the PSO is faster and need fewer algorithm parameters to handle than the ACO in general.

In this chapter, the proposed PD-like controller will be optimised by the Particle Swarm Optimisation algorithm. PSO sees a swarm of particles, which are candidate solutions to a given optimisation problem, move iteratively through solution space to find an optimised solution through a fitness equation [91-106]. The fitness equation is defined based on the objectives and constraints formulated as penalty functions of the optimisation problem. The equations for solution search in PSO are shown below [90]:

$$v_i(t) = wv_i(t-1) + c_1r_1(p_{lb}(t-1) - p_i(t-1)) + c_2r_2\big(p_{gb}(t-1) - p_i(t-1)\big)$$
$$p_i(t) = p_i(t-1) + v_i(t) \qquad (6.5)$$

where $p_i(t)$ and $p_i(t-1)$ are the current and previous positions of the i -th particle. p_{lb} is the local best position of this particle. p_{gb} is the global best position of the PSO population. $v_i(t)$ and $v_i(t-1)$ are the current and previous velocities of the i -th particle. r_1 and r_2 are random numbers between 0 and 1 at each iteration. c_1 and c_2 are two positive acceleration constants acting as coefficients of the self-recognition and social components. w is the inertia factor. The process of PSO can be divided into six steps.

1. The initial position and velocity of each particle is randomly selected.
2. The particles are evaluated by the fitness function.
3. The fitness value of each particle's current position is compared with the local best cost (l_{best}). If the value is better than l_{best}, l_{best} will be replaced by this value and p_{lb} will be supplanted by the current position.
4. The local best position is compared to the fitness value of the position with the best global cost (g_{best}). If the value is better than g_{best} then g_{best} and p_{gb} will be replaced by its fitness value and position.
5. The updated positions and velocities of all particles are generated using equation (6.3).
6. Steps two to five are repeated until stopping criterion, such as a suitable fitness value and maximum number of iterations, is achieved.

6.2 Single Objectives Optimisation for PD-like Controller

In Chapter three, the PD-like controller is proposed to control state switching in multistable systems and the strength of the control input is limited by using the constrained PD-like controller in Chapter four. Moreover, the performance of the PD-like controller is sensitive to the control parameters, and thus the controller performance can be optimised in the design process. In this chapter, the performance of the proposed PD-like controller is optimised with the help of PSO with two single objectives namely: minimising the external control force's maximum peak and switching duration. The external control force is provided by an actuator and the peak of control input in multistable state switching should be minimised so that it does not exceed the output capacity limit of the actuator involved in the system. Thus, the actuator will not be easily damaged and its lifespan is prolonged with reduced maintenance cost. Moreover, if the maximum peak of control input is too large, the whole system may be damaged. For example, in percussive drilling system, if the system is subjected to a large external force, the drill string may be easily damaged and down time occurs. Minimising the switching duration can quickly switch the system from its current, undesired state to the desired state. In the case of percussive drilling system, if the switching duration is too long, too much time and energy will be consumed not on drilling progression but on state switching. Three control parameters are considered in state switching controller optimisation, namely the controller gains k_p, k_d and switch on time. The formulation of the optimisation problem is shown below:

Minimise: $obj(x)\text{: } maximum\ peak\ of\ external\ control\ force\ \ max|u(t)|$ or

$obj(x)\text{: } switching\ duration\ \ \Delta t = t_2 - t_1$

Subject to: $h_j(x)\text{: } the\ non-linear\ dynamic\ model\ of\ a\ \ test\ system$

$x_k\text{: } k_p,\ k_d\ and\ switch\ on\ time\ t_1\ t_1$

where t_1 is the time instant when the controller is switched on and t_2 is the time instant when the controller is switched off.

The performance of the proposed PD-like controller is optimised with the help of the PSO on three test systems namely Duffing oscillator, soft impact oscillator and soft impact oscillator with a drift. In these tests, the position of the particles represents the three parameters of the controller namely $k_p,\ k_d$ and switch on time t_1. The velocity of the particles represents the variable speed of the parameters in optimal solution search. Each test has 10 trials and each trial has 50 random particles in initial population and 100 iterations are run in optimisation for converging to optimal solution. Several tests have been conducted for finding a suitable number of particles in the population and number of iterations required for convergence. The numbers of the particles and iterations were chosen as 50 and 100 respectively because if the number of the particles and iterations are too small, the fitness value may trap in a local best position easily. If the number of the particles and iterations are too large, the fitness value will converge to the same global best position way before the specified number of iterations. In each test, after 10 trials, the fitness values of all 10 trials converge to a similar result and the standard deviation of optimised results of 10 trials is small, it is safe to conclude that the fitness value has converged to the global best position.

6.2.1 Duffing Oscillator

The performance of the PD-like controller is optimised by PSO when it is applied on the Duffing oscillator. The parameters of the system are $k = 0.9$, $\Gamma = 1.9$ and $\omega = 1.2$. The first test, which comprises ten trials, minimises the external control force's maximum peak. Furthermore, each trial has 50 particles in the population and 100 iterations were run in optimisation. The best results are recorded in Table 6.1. The mean value and standard deviation of the fitness value, i.e. maximum peak of control input, are calculated to observe the result differences between each trial. Figure 6.1 shows the change of fitness values against different iterations in a typical trial. The fitness value converges to the optimal result in the 44th iteration in this trial. From the table, the fitness value of the control input's maximum peak is 0.1710 with $k_p = 0.0984$, k_d=0.1121 and the controller switches at t=1.61 seconds. The optimal results obtained in all 10 trials are consistent.

Table 6.1 Optimised results of 10 trials on minimizing maximum peak of the external control force

k_p	k_d	t_1	Maximum Peak of the External Control Force
0.0988	0.1127	1.6112	0.1708
0.0980	0.1126	1.6117	0.1707
0.0987	0.1128	1.6112	0.1701
0.1003	0.1121	1.6116	0.1708
0.1001	0.1128	1.6118	0.1710
0.0984	0.1121	1.6111	0.1700
0.1004	0.1128	1.6118	0.1713
0.0987	0.1129	1.6114	0.1711
0.1005	0.1124	1.6114	0.1712
0.0998	0.1128	1.6118	0.1710

$$mean\ value = 0.1709$$

$$standard\ deviation = 3.44 \times 10^{-4}$$

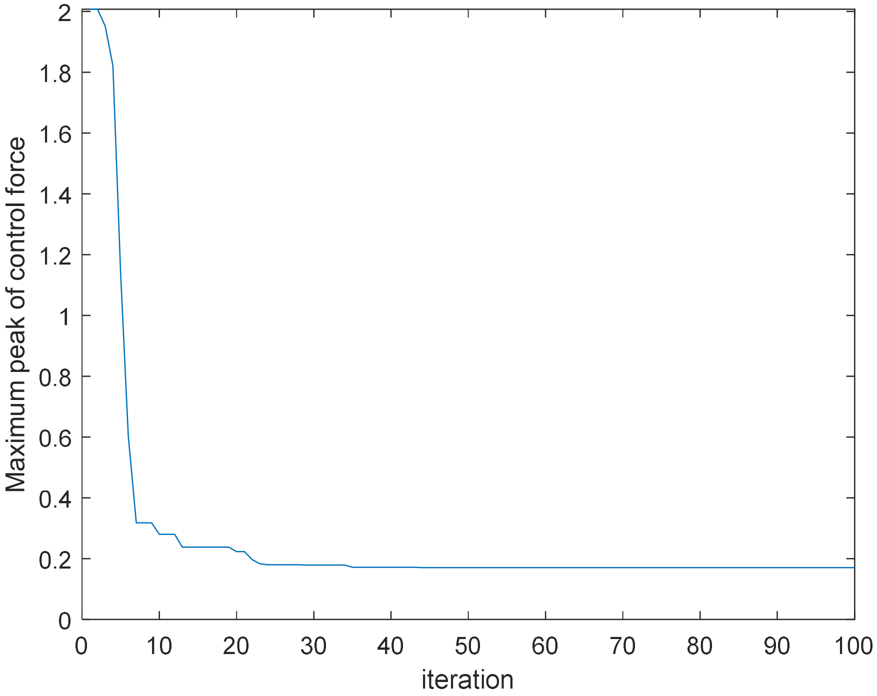


Figure 6.1 Fitness value (maximum peak of external control force) vs. iteration in a typical PSO trial.

Next, the controller's switching duration is minimised. Similar to the previous case, 10 trials have been run for 100 iterations and each has 50 particles in the population. In each trial, the mean value and standard deviations of the minimum switching duration are calculated, with the results recorded in Table 6.2. The difference between the best solutions are small as indicated by the small standard deviation. Figure 6.2 shows the fitness value (switching duration) in each iteration in a typical trial. In this trial, the fitness value decreases in the first seven iterations and is trapped in a local best position from the 8th to 25th iterations. Subsequently, it is trapped in another local best position from the 26th to 31st iterations. After the 34th iteration, it converges to the global best position. From the table, the minimum switching duration is 1.1765 seconds with the control parameters k_p=10, k_d=8.2258 and the controller switches at t=4.71 seconds.

Table 6.2 Optimised results of 10 trials on minimizing switching duration

k_p	k_d	t_1	Switching Duration (s)
10	8.2259	4.7108	1.1765
10	8.2258	4.7108	1.1765
10	8.2258	4.7108	1.1765
9.9577	8.2260	4.7087	1.1778
10	8.2259	4.7108	1.1765
10	8.2260	4.7196	1.1777
10	8.2262	4.7105	1.1781
10	8.2261	4.7112	1.1781
10	8.2259	4.7108	1.1765
10	8.2261	4.7112	1.1781

$$mean\ value = 1.1772$$

$$standard\ deviation = 7.40 \times 10^{-4}$$

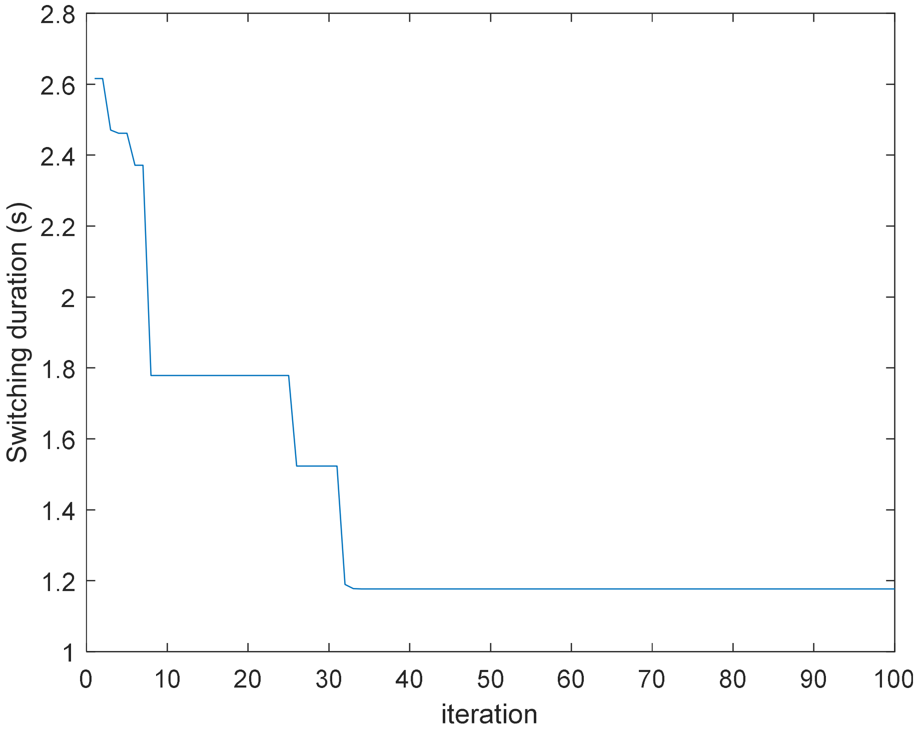


Figure 6.2 Fitness value (switching duration) vs iterations in a typical trial.

According to the first test, the optimisation parameters for minimising the control input's maximum peak are $k_p = 0.0984$, k_d=0.1121 and the controller switches on at t=1.61 seconds. As depicted in Figure 6.3, the maximum peak of the control input is 0.17 and switching duration is about 134 seconds. In the first test on the Duffing oscillator (as shown in Figure 3.6), the maximum peak of the control input was decreased by 89.94% while the switching duration was increased by 88.81%. Figure 6.4 shows the control input with minimum switching duration and the control parameters are k_p =10, k_d=8.2258 with the controller switching on at t=4.71 seconds. In addition, the maximum peak of the control input is 11.38 and switching duration is about 1 second. Compared with the first test on the Duffing oscillator, the maximum peak of the control input is increased by 85.30% and the switching duration is decreased by 93.33%. According to the optimisation results, the two objectives, namely maximum peak of control input and switching duration are conflicting with each other and no single set of control parameters can optimise them at the same time. Hence, trade-off on performance of the proposed controller must be considered in optimisation.

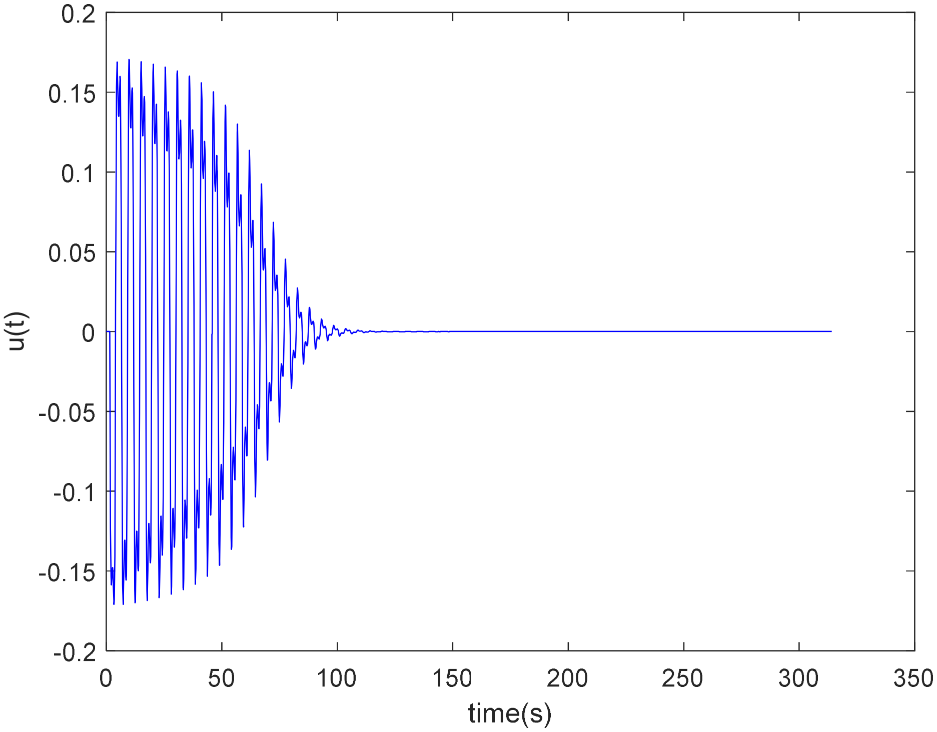


Figure 6.3 Control input to the Duffing oscillator using the PD-like control law with the optimised control parameter $k_p = 0.0984$, $k_d = 0.1121$ and the controller switches on $t = 1.61$ seconds

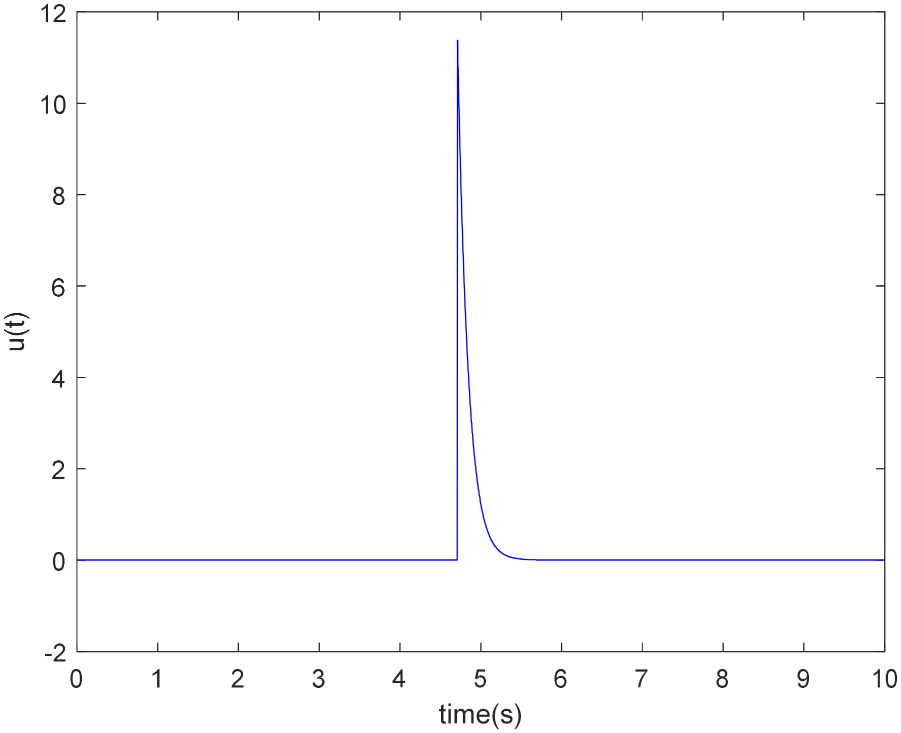


Figure 6.4 Control input to the Duffing oscillator using the PD-like control law with the optimised control parameter $k_p = 10$, $k_d = 8.2258$ and the controller switches on at $t = 4.71$ seconds

6.2.2 Soft impact oscillator

In this section, the PD-like controller is optimised by PSO when it is applied to the soft impact oscillator. The system parameters of the soft impact oscillator are $\xi = 0.01$, $\beta = 29$, $g = 1.26$, $\Gamma = 1.9$ and $\omega = 0.686$. The first test, involving 10 trials, that have 50 particles and are run in 100 iterations in each trial, is to minimise the maximum peak of the control force. The best results of each trial are recorded in Table 6.3. Moreover, in order to examine the different results in each trial the mean value and standard deviation are calculated. It can be observed that all trials perform similarly. Figure 6.5 shows the change of fitness values with iterations run. Figure 6.5 shows that, in a typical trial, the fitness value decreases quickly in the first 17 iterations, and after 58 iterations it converges to a steady value of the fitness. Furthermore, as can be seen in Table 6.3, the fitness value of the maximum peak of the control input is 1.2435 with $k_p = 0.0003$, k_d=0.4677 and the controller switches on at t=4.44 seconds. According to Table 6.3, all 10 trials give consistent optimal parameters of the proposed PD-like controller.

Table 6.3 Optimised results of 10 trials on minimizing maximum peak of the external control force

k_p	k_d	t_1	Peak of the External Control Force
0.0003	0.4677	4.4407	1.2435
0.0001	0.4677	4.4306	1.2437
0.0001	0.4678	4.3628	1.2438
0.0002	0.4678	4.4346	1.2437
0.0002	0.4677	4.4544	1.2437
0.0001	0.4677	4.4474	1.2436
0.0002	0.4678	4.4462	1.2437
0.0002	0.4677	4.4410	1.2435
0.0002	0.4677	4.4480	1.2436
0.0002	0.4678	4.3865	1.2438

$$mean\ value = 1.2437$$

$$standard\ deviation = 1.18 \times 10^{-4}$$

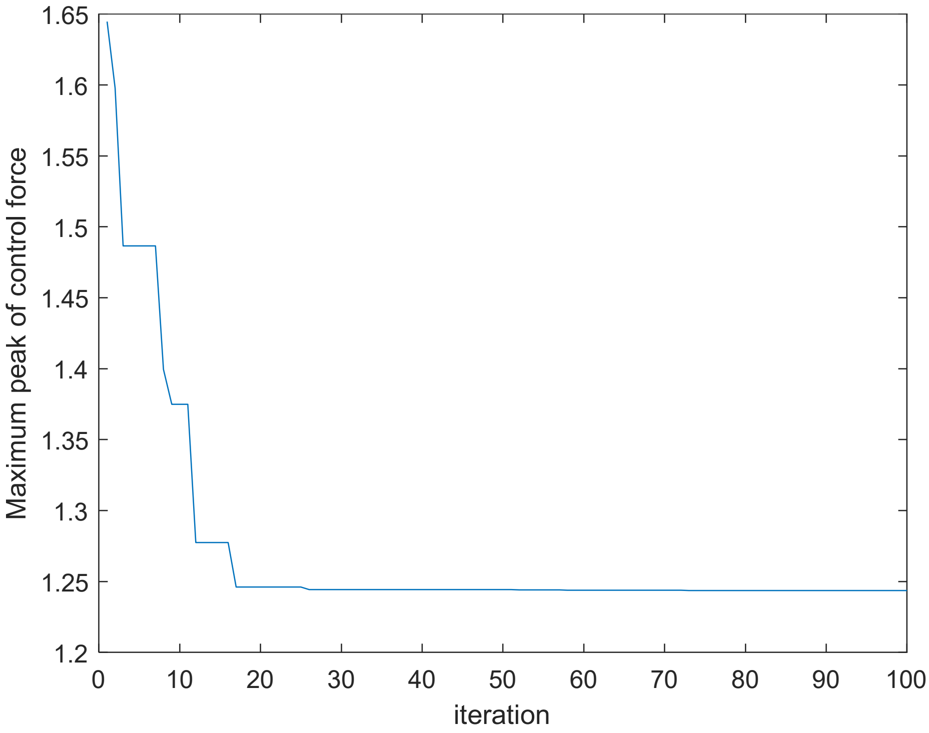


Figure 6.5 Fitness value (maximum peak of external control force) vs. iteration in a typical trial.

The goal of the second test, which in this instance incorporates 10 trials each with 50 particles and 100 iterations, is to optimise the controller's switching duration. Each trial's switching duration fitness value is recorded in Table 6.4, while the mean value and standard deviation in each test are also calculated. All 10 trials are able to produce consistent optimal design of the PD-like controller. Figure 6.6 depicts the switching duration fitness value of each iteration in a typical trial. In this trial, within the first five iterations, the fitness value quickly decreases before stabilising. It then remains in a local best position. At the 24th iteration the value decreases again, and after the 25th iteration it converges to the optimal value. The table shows that the minimum switching duration is 1.038 seconds with control parameters k_p=7.0967, k_d=9.9955 and the controller switches on at t=3.41 seconds.

Table 6.4 Optimised results of 10 trials on switching duration

k_p	k_d	t_1	Switching Duration (s)
7.0967	9.9955	3.4133	1.0380
7.0969	9.9954	3.4133	1.0380
7.0981	9.9976	3.4133	1.0625
7.0967	9.9954	3.4133	1.0380
7.0970	9.9968	3.4133	1.0441
7.0969	9.9954	3.4133	1.0380
7.0948	9.9954	3.4133	1.0380
7.0965	9.9960	3.4133	1.0472
7.0956	9.9955	3.4133	1.0380
7.0957	9.9963	3.4133	1.0503

$$mean\ value = 1.04321$$

$$standard\ deviation = 7.76 \times 10^{-3}$$

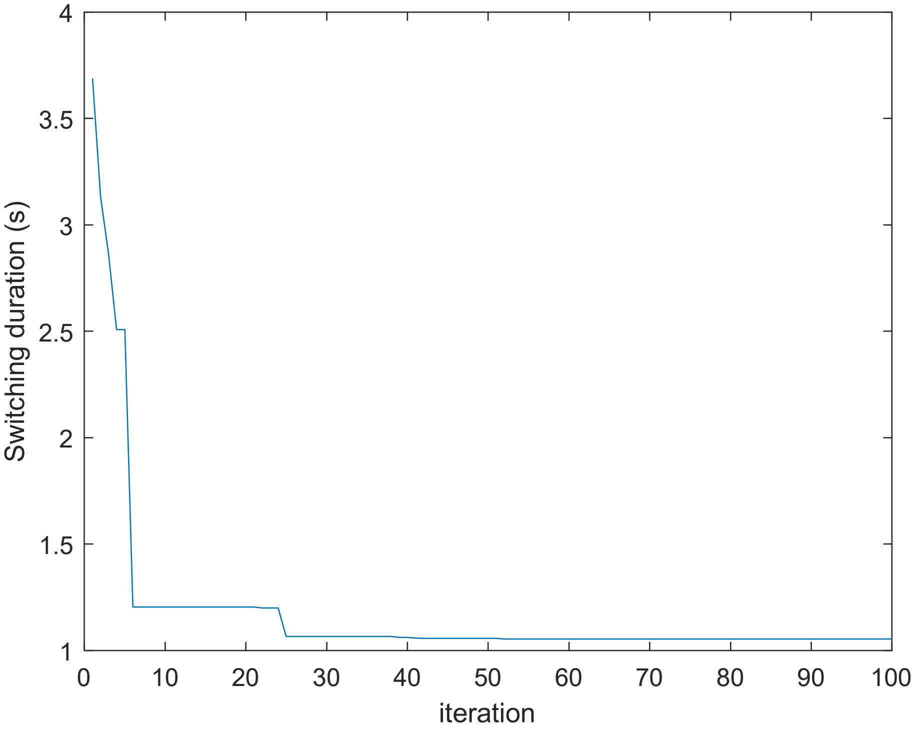


Figure 6.6 Fitness value (switching duration) vs. iterations in a typical trial.

According to the previous two tests, the control parameters for minimising the maximum peak of the external control force are obtained as $k_p = 0.0003$, k_d=0.4677 and the controller switches on at t=4.44 seconds. Figure 6.7 shows the maximum peak of control force is 1.2435 and the switching duration is about 63 seconds. Compared with the non-optimal control parameters (shown in Figure 3.12), the maximum control force is reduced by approximately 60.62% but the control duration is increased by 65.08%. The optimal control parameters for minimum switching duration are k_p=7.0967, k_d=9.9955 and the controller switches on at t=3.41 seconds. Compared with the non-optimal control parameters as shown in Figure 3.12, the switching duration in figure 6.8 is only around 1 second which is reduced by 95.45%. However, the maximum peak of the external control force in Figure 6.8 is 34.47 which is an increase of 90.84%.

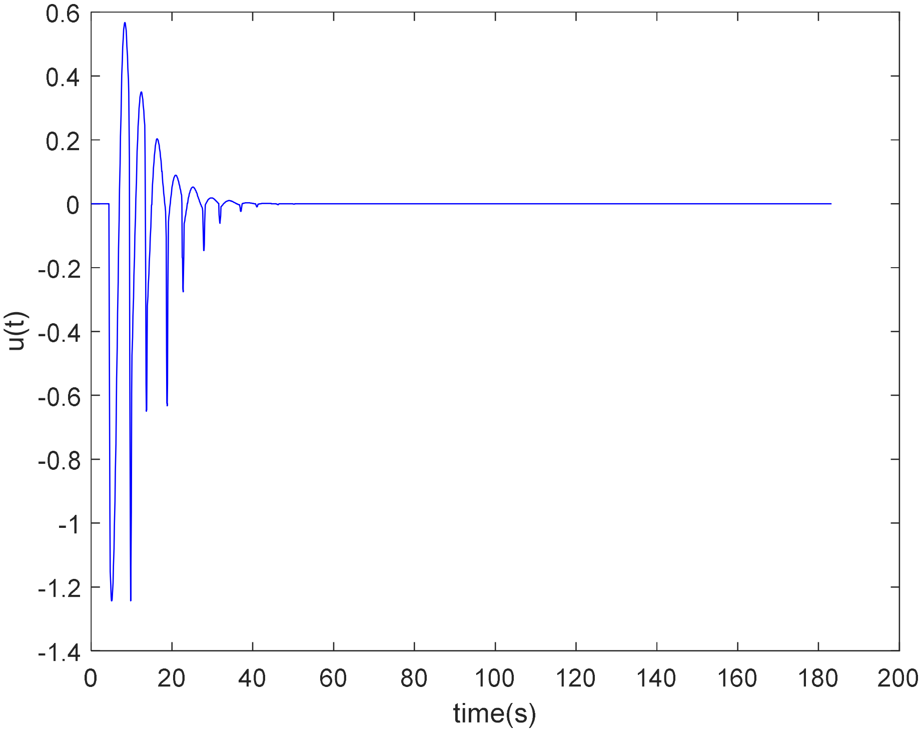


Figure 6.7 Control input to the soft impact oscillator using the PD-like control law with the optimised control parameter $k_p = 0.0003$, $k_d = 0.4677$ and the controller switches on at $t = 4.44$ seconds

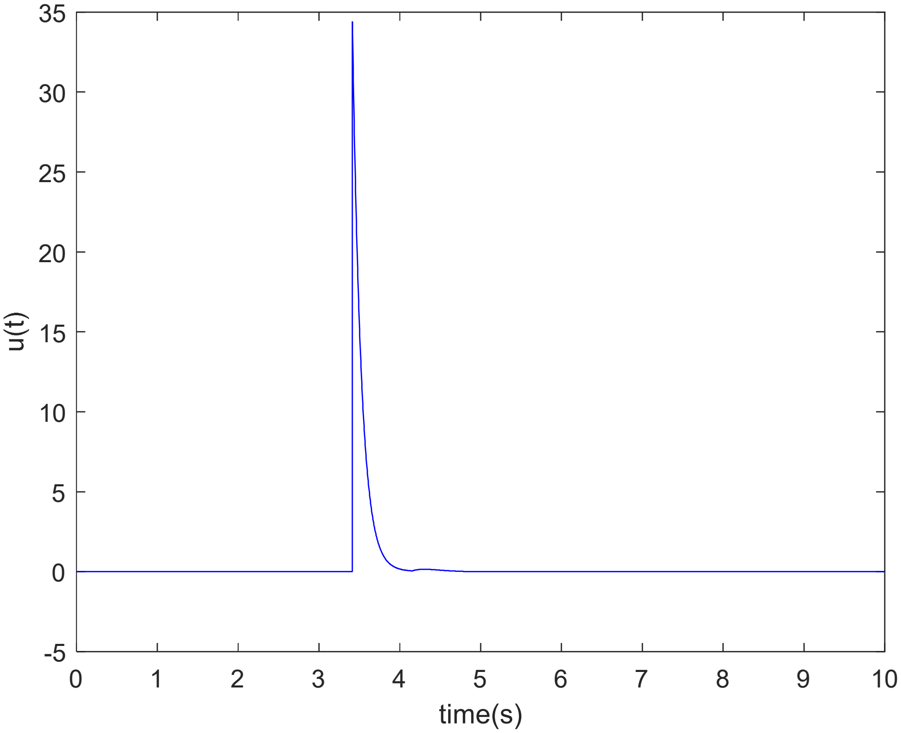


Figure 6.8 Control input to the soft impact oscillator using the PD-like control law with the optimised control parameter $k_p = 7.0967$, $k_d = 9.9955$ and the controller switches on at $t = 3.41$ seconds

6.2.3 Soft impact oscillator with a Drift

In this test, the PD-like control is optimised by PSO when applied on the soft impact oscillator with a drift. The system parameters of the oscillator are $a = 0.3$, $b = 0.2$, $\omega = 1$, $g = 0.02$, $\xi = 0.1$ and $\varphi = \pi/2$. The objective of this test, which features 10 trials with 50 particles and in 100 iterations each, is to reduce the maximum peak of the controller's external control force. The results for each trial are shown in Table 6.5, as well as the mean value and standard deviation across all 10 trials. In addition, Figure 6.9 shows the variation of the fitness value with the number of iterations increases. In a typical trial, it takes 55 iterations to converge to an optimum. From Table 6.5 the minimum peak of external control force is 0.0555 with k_p=0.0434, k_d=0.0714 and the controller is switched on at t=5.4727 second.

Table 6.5 Optimised results of 10 trials on minimizing maximum peak of the external control force

k_p	k_d	t_1	Maximum Peak of the External Control Force
0.0434	0.0714	5.4727	0.0555
0.0425	0.0710	5.4768	0.0557
0.0434	0.0714	5.4727	0.0555
0.0434	0.0714	6.2832	0.0562
0.0445	0.0723	5.5501	0.0556
0.0432	0.0714	5.4727	0.0555
0.0434	0.0714	5.6004	0.0557
0.0434	0.0714	5.4978	0.0556
0.0434	0.0714	5.4727	0.0555
0.0434	0.0714	5.7271	0.0560

$$mean\ value = 0.0557$$

$$standard\ deviation = 2.27 \times 10^{-4}$$

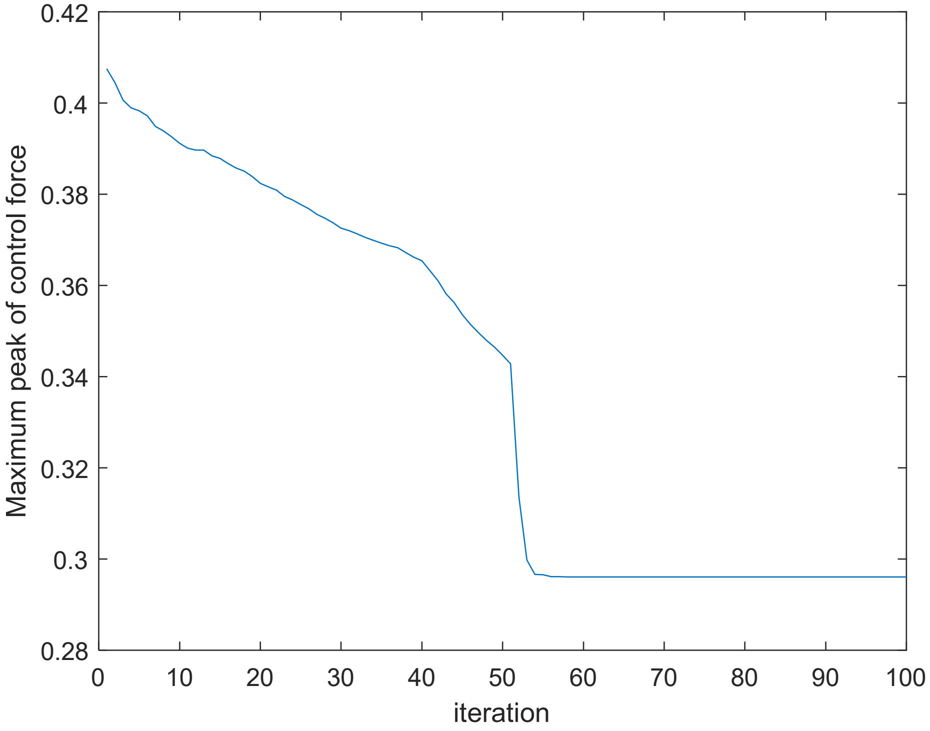


Figure 6.9 Fitness value (maximum peak of external control force) vs. iterations in a typical trial.

10 trials have been conducted to optimise the switching duration when the proposed PD-like controller is applied to a soft impact oscillator with a drift. The initial population has 50 particles and a PSO is run for 100 iterations. The optimal results are listed in Table 6.6. The mean value and standard deviation of the results are also calculated and it is noted that all 10 trials are resulted in consistent optimal solution. Figure 6.10 shows the performance of a typical trial and indicates that the PSO took 12 iterations to converge to the optimal result in this trial. From Table 6.6 the minimum switching duration is 0.8126 seconds with the control parameters of k_p=4.0264, k_d=9.4060 and the controller is switched on at t=4.3668 second.

Table 6.6 Optimised results of 10 trials on switching duration

k_p	k_d	t_1	Switching Duration (s)
4.0264	9.4060	4.3668	0.8126
4.0264	9.4060	4.3668	0.8126
4.1354	9.4050	4.3762	0.8221
4.2358	9.4055	4.3616	0.8137
4.5067	9.4060	4.3647	0.8158
4.0264	9.4060	4.3668	0.8126
4.0264	9.4051	4.3720	0.8137
4.0264	9.4060	4.3668	0.8126
4.0264	9.4060	4.3668	0.8126
4.2352	9.4065	4.3689	0.8294

$$mean\ value = 0.8158$$

$$standard\ deviation = 5.34 \times 10^{-3}$$

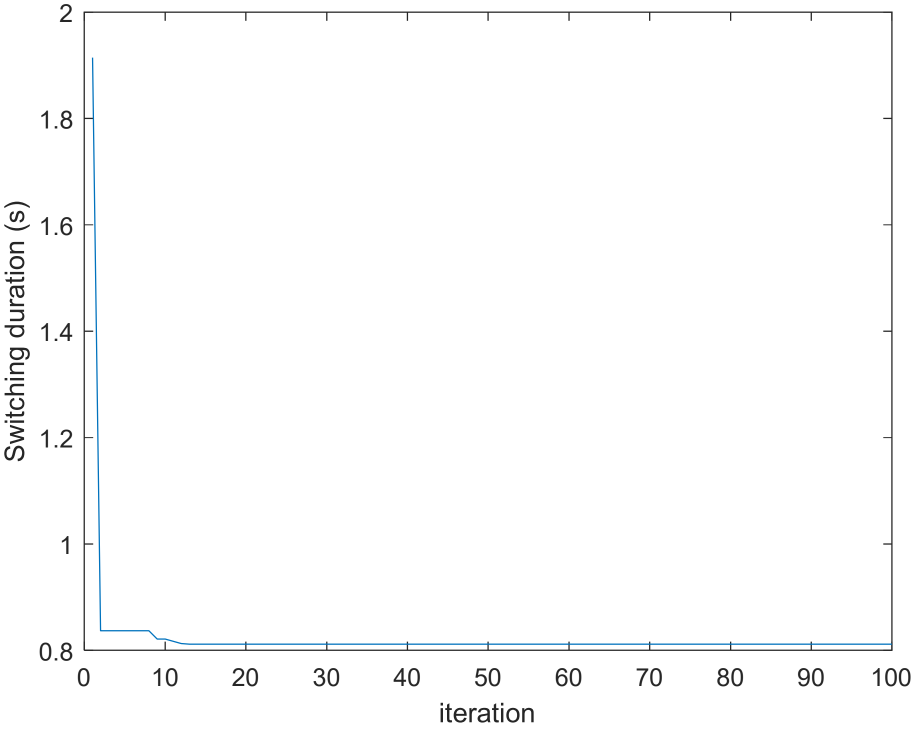


Figure 6.10 Fitness value (switching duration) vs. iterations in a typical trial.

When analysing state switching in the soft impact oscillator with a drift, the controller parameters for minimising the external control force's maximum peak are $k_p = 0.0434$, $k_d = 0.0714$ and the controller switches on at $t = 5.47$ seconds. Figure 6.11 shows the control input of the controller using this set of parameters. The maximum peak of the control input is only 0.055, which is 97.36% smaller than the control input using the non-optimal set of control parameters (as shown in Figure 3.17). However, it takes approximately 140 seconds to switch the system to its desired state, which is 88.57% longer than the case shown in Figure 3.17. Figure 6.12 illustrates the control input of the controller with the minimum switching duration. The set of control parameters are $k_p = 4.0264$, $k_d = 9.4060$ and the controller switches on at $t = 4.36$. The switching duration with this set of control parameters is about 1 second, which is 93.75% shorter than the case with non-optimal PD-like control parameters as shown in Figure 3.17. However, the maximum control input is 7.49, which is 72.19% higher compared to the non-optimal case.

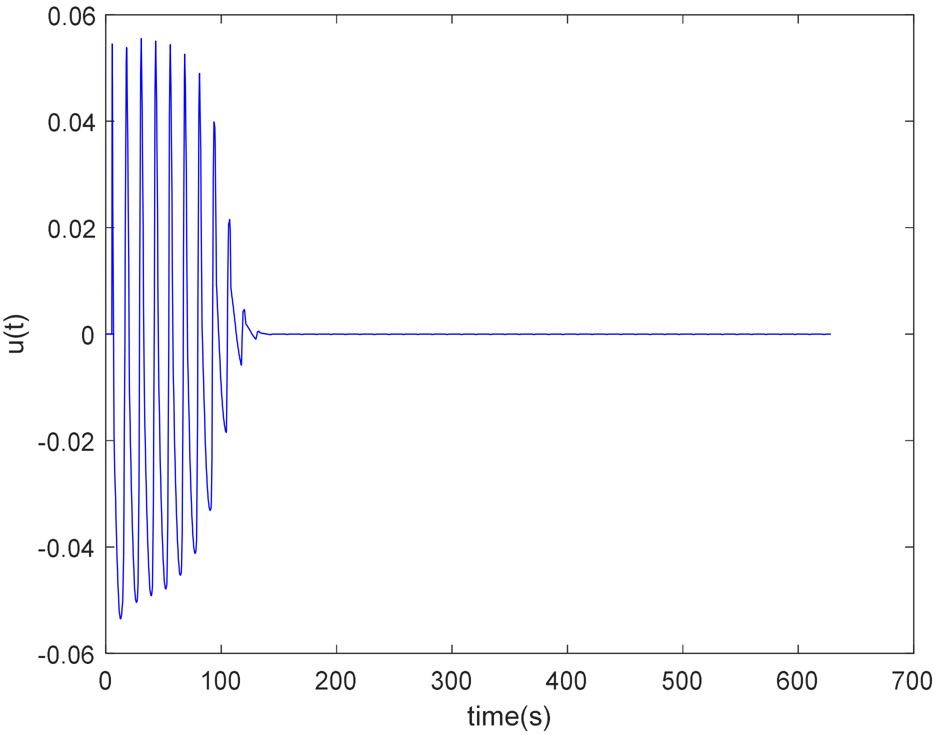


Figure 6.11 Control input to the soft impact oscillator with a drift using the PD-like control law with the optimised control parameter $k_p = 0.0434$, $k_d = 0.0714$ and the controller switches on at $t = 5.47$ seconds

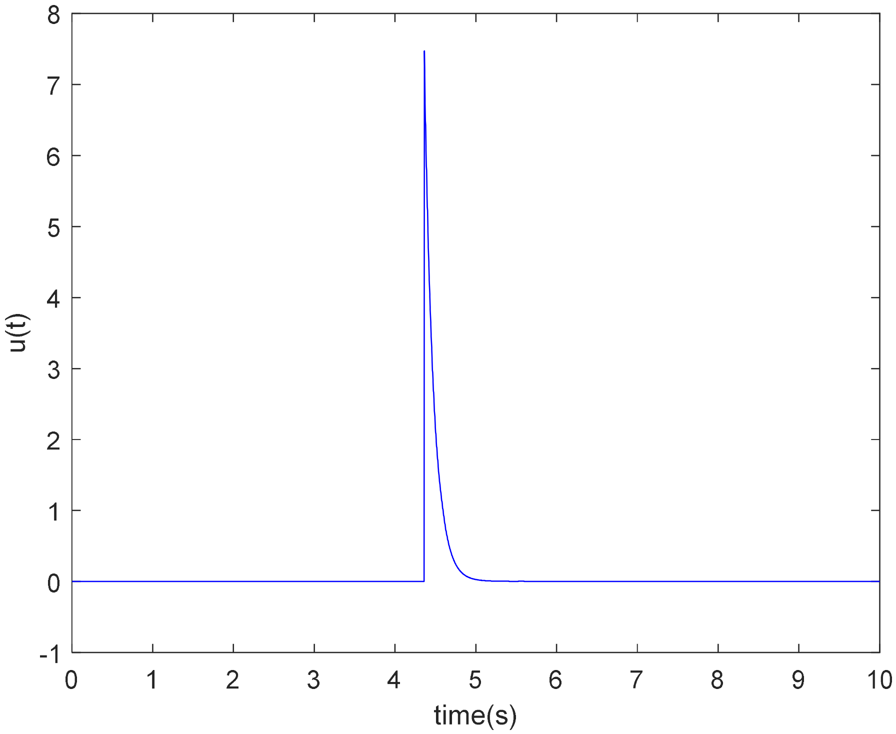


Figure 6.12 Control input to the soft impact oscillator with a drift using the PD-like control law with the optimised control parameter $k_p = 4.0264$, $k_d = 9.4060$ and the controller switches on at $t = 4.36$ seconds

According to the analysis on the three test systems, these two objectives are conflicting with each other. On one hand, when the maximum peak of the external control force is minimised, the switching duration will increase significantly. The small external control force can reduce the energy and maintenance costs, but it requires more time to switch the system to the desired state. On the other hand, when the switching duration is minimised, the maximum peak of the external control force will increase. The small switching duration implies that the system can be switched to the desired stable state quickly although sometimes the actuator cannot provide the power required to shorten the switching duration.

6.3 Summary

PSO was applied for optimising the PD-like controller parameters with two single objectives namely: minimising the maximum peak of external control force and switching duration. The performance of PSO was investigated on three test systems namely: the Duffing oscillator, the soft impact oscillator and the soft impact oscillator with a drift. According to the analysis, the two single objectives are conflicting with other. Therefore, it is important to find a trade-off between the external control force and the switching duration by optimising the controller with multiple objectives at the same time.

Chapter 7 Multi-objective Optimisation of PD-like Controller Parameters

This chapter further optimises the proposed PD-like controller with multiple conflicting objectives. Specifically, the two objectives optimised independently in previous chapter are optimised at the same time using multi-objective optimisation. The objectives under consideration are minimising the maximum peak of external control force and the switching duration. The PD-like controller parameters are optimised for the three test systems namely, Duffing oscillator, soft impact oscillator and soft impact oscillator with a drift, in this chapter for improved performance.

7.1 Introduction

In many real-life problems, designers consider multiple objectives in system design and optimisation. However, in many cases, some objectives are in conflict with other and they cannot be optimised simultaneously with the same set of parameters. For example, in last chapter, the PD-like controller was optimised independently by PSO with two objectives namely, minimising the maximum peak of external control force and switching duration. The results showed that, these two objectives are contrary to each other. Moreover, in engineering design and optimisation, decision makers usually consider conflicting objectives such as improving the performance of the system while reducing the consumption of energy and emission pollutants. He/she usually faces trade-off in enhancing one performance objective in sacrifice of the other based on multi-objective optimisation results.

7.1.1 Multi-objective Optimisation

In general, a multi-objective optimisation problem is formulated as follows [78]:

$$\begin{aligned}
&\text{Minimise:} && OBJ(x) = [obj_1(x), obj_2(x), \dots, obj_q(x)]^T \\
&\text{Subject to:} && g_i(x) \leq 0, \qquad i = 1, \dots, m \\
& && h_j(x) = 0, \qquad j = 1, \dots, p \\
& && x = [x_1, x_2, \dots, x_n]^T \\
& && x_k^l \leq x_k \leq x_k^u, \qquad k = 1, \dots, n
\end{aligned} \tag{7.1}$$

where $OBJ(x)$ is a vector of multiple objective functions, q is the number of the objectives under consideration, $g_i(x)$ is the $i-$th inequality constraint function, m is the number of inequality constraint functions, $h_j(x)$ is the $j-$th equality constraint function, p is the number of equality constraint functions, x is the vector of the independent optimising variables, x_k is the $k-$th independent variable, n is the number of the independent variables, x_k^l and x_k^u are the lower bound and upper bound of the $k-$th independent variable respectively. Maximization of an objective can easily be converted into a minimisation by defining the minimisation objective as: $l_n(x) = \frac{1}{f_n(x)}$, where $f_n(x)$ is the function of $n-$th objective for maximization [78,109].

7.1.2 Pareto Optimal Solutions

As compared with single objective optimisation, multi-objective optimisation usually has multiple optimal solutions, known as Pareto optimal solutions [110]. Without any requirement, there is no best solution in the set of optimal solutions that optimise all objectives at the same time. A Pareto optimal solution is a solution that is better in one or more objectives and not worse in all objectives and it also is known as the nondominated solution [78,111]. For example, the proposed PD-like control is applied on a multistable system with two sets of control parameters. The corresponding objective values (maximum peak of external control force and switching duration) of two different sets of control parameters are observed and they are named by solution A and solution B. If solution A is not worse than solution B in both objectives (the maximum peak of external control force and switching duration of solution A are not larger than those of solution B) and solution A is better than solution B in at least one objective (the maximum peak of external control force of solution A is smaller than that in solution B and/or the switching duration of solution A is smaller than that in solution B), solution A is a nondominated solution and solution B is a dominated solution. Mathematically, one solution x_1 is said to dominate (in the Pareto's sense) another solution x_2, if

$$obj_i(x_1) \leq obj_i(x_2) \quad \forall i \in \{1, \dots, m\} \text{ and } \exists j \in \{1, \dots, m\} \mid obj_j(x_1) < obj_j(x_2) \quad (7.2)$$

Therefore, the nondominated solution or Pareto optimal solution is the solution which have no other solution dominated itself. Moreover, the set of Pareto optimal solutions is called the Pareto Front. Figure 7.1 shows an example of Pareto Front with two objectives f_1 and f_2. The bold line shows the Pareto Front and the yellow point shows one Pareto optimal solution (nondominated solution).

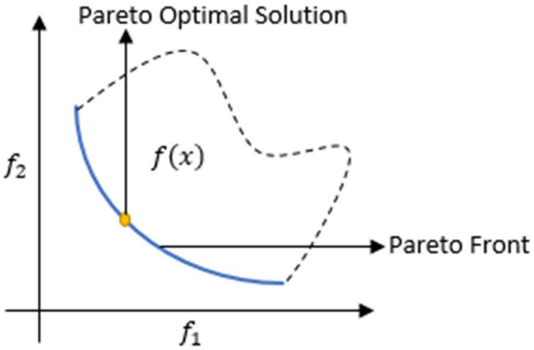


Figure 7.1 Pareto fronts for two objectives optimisation problems

7.1.3 Multi-objectives Optimisation Algorithms

There are several methods that can be applied for solving multi-objective optimisation problem. The traditional method involves transforming the multi-objective problem into a single objective problem by adding a weight factor to every objective. Although this method is simple and has low computation complexity, the optimal solution is mainly affected by each objective's weight factor which depends on the applications and/or user preference. Sometimes, it is difficult to determine a suitable combination of weights for multiple objectives in real life applications.

For attaining a set of Pareto-optimal solution through one simulation, a Nondominated Sorting Genetic Algorithm (NSGA) was proposed in 1995 [112-129]. This method is based on the genetic algorithm (GA) which is an evolutionary algorithm (EA) and follows the principle of survival of the fittest [130-135]. Firstly, a population of candidate solutions is randomly selected and the fitness value of every solution is calculated based on the objective functions [136]. Secondly, according to the fitness values of every solution in the population, the solutions with high fitness value are selected from the parent population. Thirdly, offspring candidate solution population is generated through crossover and mutation operations based on the parent population. Fourthly, a new population, which is combined by the parent and offspring populations, is evaluated by the fitness function. Finally, when certain stopping conditions such as maximum number of generations or maximum running time is achieved, optimal solution is obtained. Figure 7.2 shows the flowchart of GA.

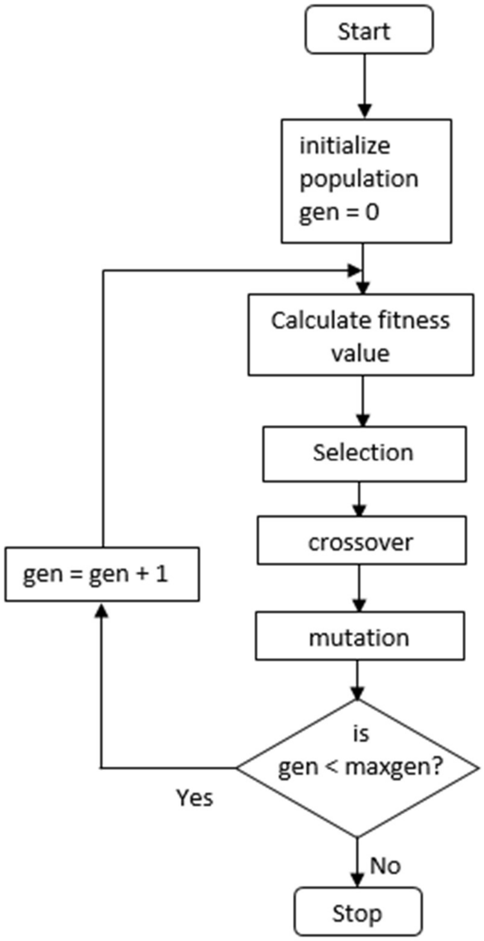


Figure 7.2 Flow chart of GA

NSGA and GA adopt similar principles. However, there is a little difference between the two methods in terms of the selection process. The NSGA requires the fitness values of each individual. The individuals are divided into different levels according to the nondominated sorting procedure. Every individual in *p* nondominated front is provided with a fitness number f_p, with the individuals in the first nondominated level given a larger fitness number. Furthermore, a sharing method is suggested for ensuring the diversity of the population. The following equation is used for calculating the sharing value between two individuals in the same front [112].

$$sh(d_{ij}) = \begin{cases} 1 - \left(\frac{d_{ij}}{\sigma_{share}}\right)^2, & if\ d_{ij} < \sigma_{share} \\ 0, & otherwise \end{cases} \tag{7.3}$$

where d_{ij} is the Euclidean distance between the two individual solutions x_i and x_j; σ_{share} is a sharing parameter and is the maximum phenotypic distance allowed between any two individuals (candidate solutions), indicating set value. If d_{ij} is less than σ_{share}, the $sh(d_{ij})$ is larger than 0 and smaller than 1 and large $sh(d_{ij})$ implies the two individuals i and j have a higher similarity. If d_{ij} is greater than σ_{share}, $sh(d_{ij})$ equals to zero. Then, the niche count of the individuals is calculated in the current front by the following equation [112],

$$c_i = \sum_{j=1}^{n_p} sh(d_{ij}), i = 1, 2, \dots, n_p \tag{7.4}$$

where n_p is the number of individuals in the same front. The niche count is dependent on the chosen σ_{share}. Large σ_{share} implies that more other individuals are in the niche of the individual and the niche count of the individual will increase. Moreover, the shared fitness value of the individuals in the *p* nondominated front is computed as follows [112].

$$f_p'(x_i) = \frac{f_p(x_i)}{c_i} \tag{7.5}$$

Regarding two randomly selected individuals, if one is nondominated and the other is dominated, then the nondominated one is considered to be a better solution. In case of both being in the same front, the individual with small niche count is selected.

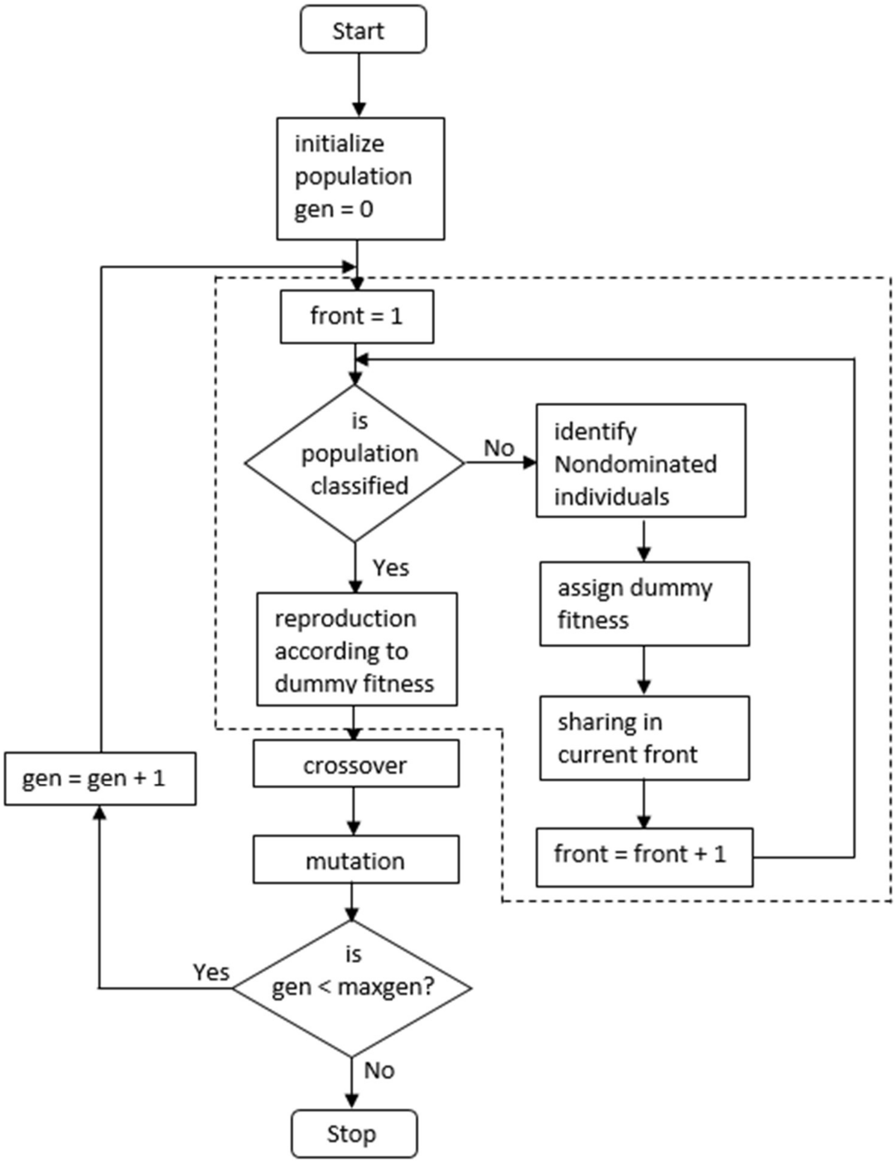
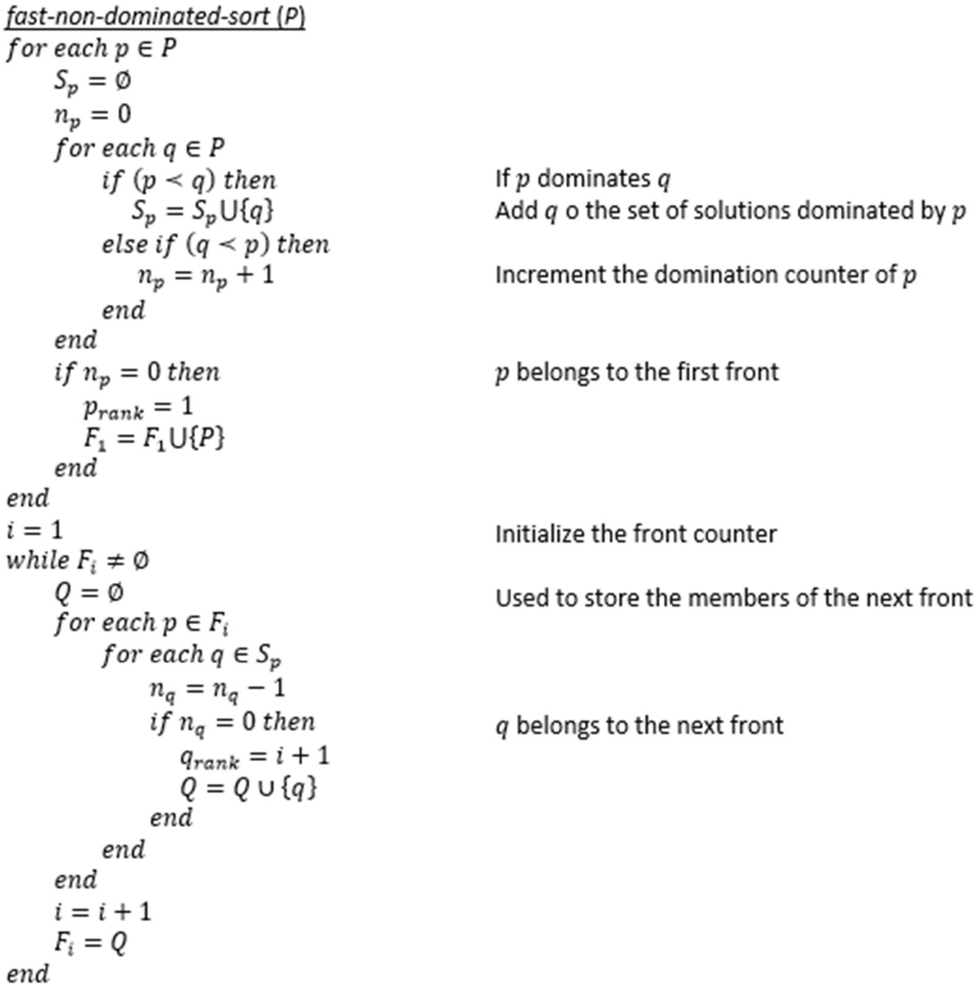


Figure 7.3 Flow chart of NSGA. The dashed block shows the nondominated sorting procedure in NSGA which is different from the selection procedure in GA.

However, the NSGA has three major disadvantages [137-142]. Firstly, it has high computational complexity. Secondly, it has the property of absence of elitism which means it may lose the fittest solutions they found before and cannot pass them to next generations. Thirdly, the diversity of the population is dependent on the sharing function and the sharing function need a sharing parameter σ_{share} which is a user-defined parameter. For overcoming these limitations, a fast and elitist multi-objective genetic algorithm (NSGA-II) was proposed in 2002 [141]. Moreover, NSGA-II adopts the nondominated sorting and crowding distance for determining the Pareto front [143-147].

For each solution p, two parameters are calculated. The first parameter n_p is the number of solutions that dominate solution p, while the second parameter S_p is the set of the solutions dominated by the solution p,. Next, all solutions whose $n_p = 0$ are identified and these solutions belong to the first nondominated front F_1. For every solution j in F_1, its set S_q is verified. Furthermore, for every solutions q in S_q, the number of domination count, n_q, is determined. Then, all solutions who are only dominated by one solution $(n_q = 1)$ are established which all belong to the second nondominated front F_2. These processes stop only after identifying all solutions.



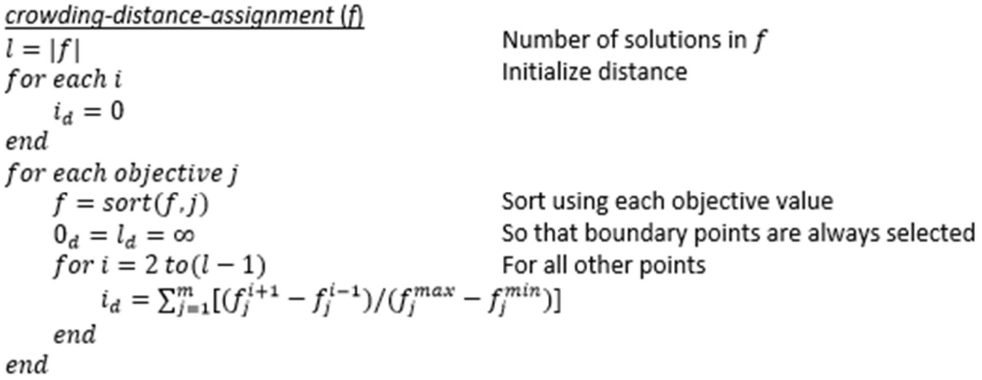
```
fast-non-dominated-sort (P)
for each $p \in P$
    $S_p = \emptyset$
    $n_p = 0$
    for each $q \in P$
        if $(p \prec q)$ then                    If $p$ dominates $q$
            $S_p = S_p \cup \{q\}$               Add $q$ o the set of solutions dominated by $p$
        else if $(q \prec p)$ then
            $n_p = n_p + 1$                      Increment the domination counter of $p$
        end
    end
    if $n_p = 0$ then                            $p$ belongs to the first front
        $p_{rank} = 1$
        $F_1 = F_1 \cup \{P\}$
    end
end
$i = 1$                                          Initialize the front counter
while $F_i \neq \emptyset$
    $Q = \emptyset$                              Used to store the members of the next front
    for each $p \in F_i$
        for each $q \in S_p$
            $n_q = n_q - 1$
            if $n_q = 0$ then                    $q$ belongs to the next front
                $q_{rank} = i + 1$
                $Q = Q \cup \{q\}$
            end
        end
    end
    $i = i + 1$
    $F_i = Q$
end
```

Figure 7.4 Pseudo code of nondominated sorting [143]

The NSGA uses the sharing function method to preserve diversity in population. The performance of the sharing function depends on the sharing parameter which is a user-defined parameter. Sometimes, it is difficult to determine the suitable sharing parameter for the sharing function. Moreover, each solution is compared to all other solutions, leading to high computational complexity. The crowded comparison method is recommended to overcome these limitations. (the pseudo code of crowding distance assignment is shown in Figure 7.5). In this method, the crowding-distance of every solution is first set to zero. Then, the crowding-distance of the boundary solutions for every objective is set to infinity. Next, every solution's crowding distance is calculated using the following equation:

$$i_d = \sum_{j=1}^{m} [(f_j^{i+1} - f_j^{i-1})/(f_j^{ma} \ \ - f_j^{min})] \qquad (7.6)$$

where i_d is the crowding distance of solution i, m is the total number of objectives, f_j^{i+1} and f_j^{i-1} are the function value of solutions $i+1$ and $i-1$ of the j^{th} objective, and f_j^{max} and f_j^{min} are the maximum and minimum function value of the j^{th} objective, respectively.



```
crowding-distance-assignment (f)
$l = |f|$                                   Number of solutions in $f$
$for\ each\ i$                              Initialize distance
    $i_d = 0$
$end$
$for\ each\ objective\ j$
    $f = sort(f, j)$                        Sort using each objective value
    $0_d = l_d = \infty$                    So that boundary points are always selected
    $for\ i = 2\ to(l-1)$                   For all other points
        $i_d = \sum_{j=1}^{m}[(f_j^{i+1} - f_j^{i-1})/(f_j^{max} - f_j^{min})]$
    $end$
$end$
```

Figure 7.5 Pseudo code of crowding distance [143]

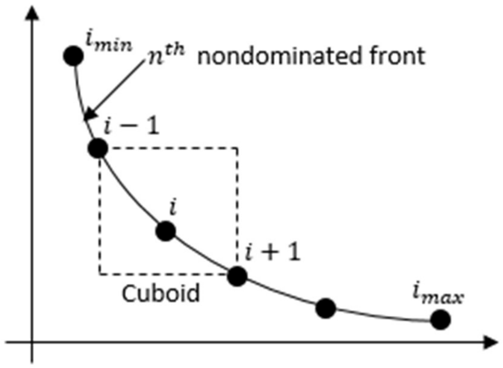
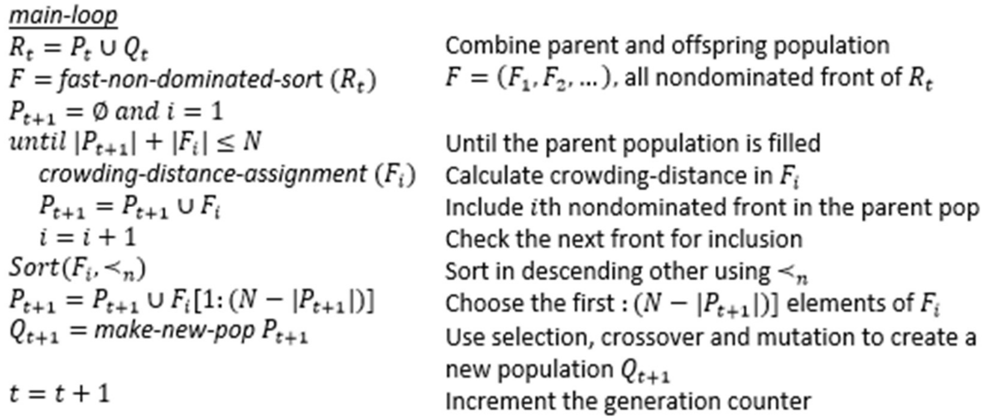


Figure 7.6 Crowding distance calculation

Next, the crowded-comparison operator ($\prec_n$) is taken into consideration for Pareto front building. For every solution, i has two attributes, one is nondomination rank (i_{rank}) and the other is the crowding distance (i_d). For two solutions, i and j, the following equation is implemented:

$$i \prec_n j \ \ if \ (i_{rank} \leq j_{rank}) \ and \ (i_d > j_d) \qquad (7.7)$$

While the Pareto front is building, if solution i has a lower nondomination rank than that of solution j, then solution i is preferred. If i and j are in the same front and if the crowding distance of i is larger than that of j, then solution i is preferred.



```
main-loop
$R_t = P_t \cup Q_t$                            Combine parent and offspring population
$F =$ fast-non-dominated-sort $(R_t)$           $F = (F_1, F_2, \dots)$, all nondominated front of $R_t$
$P_{t+1} = \emptyset$ and $i = 1$
until $|P_{t+1}| + |F_i| \leq N$                Until the parent population is filled
    crowding-distance-assignment $(F_i)$        Calculate crowding-distance in $F_i$
    $P_{t+1} = P_{t+1} \cup F_i$                Include $i$th nondominated front in the parent pop
    $i = i + 1$                                 Check the next front for inclusion
Sort$(F_i, \prec_n)$                            Sort in descending other using $\prec_n$
$P_{t+1} = P_{t+1} \cup F_i[1: (N - |P_{t+1}|)]$    Choose the first : $(N - |P_{t+1}|)]$ elements of $F_i$
$Q_{t+1} =$ make-new-pop $P_{t+1}$              Use selection, crossover and mutation to create a
                                                new population $Q_{t+1}$
$t = t + 1$                                     Increment the generation counter
```

Figure 7.7 Pseudo code of NSGA-II [143]

In the elitism approach of NSGA-II, the first generation is randomly selected and classified as nondominated. Based on the selection, crossover, and mutation operators in GA, their offspring are generated with size of *N*. The current population is then classified as per the nondomination, following which the elitism approach is introduced. The elitism approach of the t^{th} generation (shown in Figure 7.7) begins with introducing a combined population R_t with size 2*N* that combines the parent population P_t with the offspring population Q_t. Then, the population is sorted according to the nondomination rank while the solutions in R_t is divided into different sets, F_1 (the set of all the solutions that belong to the first nondominated front), F_2, …, F_n. Furthermore, all solutions in F_1 are the best individuals in the population R_t. If the size of F_1 is smaller than *N*, all solutions in F_1 are selected to be the parent population of the next generation P_{t+1}. The next nondominated front F_2 and subsequent nondominated F_3, and so on, are then considered until the size of population P_{t+1} is equal to *N*. F_l is assumed to be the last set chosen for P_{t+1} . It should be noted that the size of the solutions from the set F_1 to F_l is typically larger than N. Therefore, regarding set F_l, the solutions are classified using the crowded comparison operator while the better individuals are selected for the population P_{t+1} first. The new population P_{t+1} is then used for developing the next offspring population Q_{t+1} with size N.

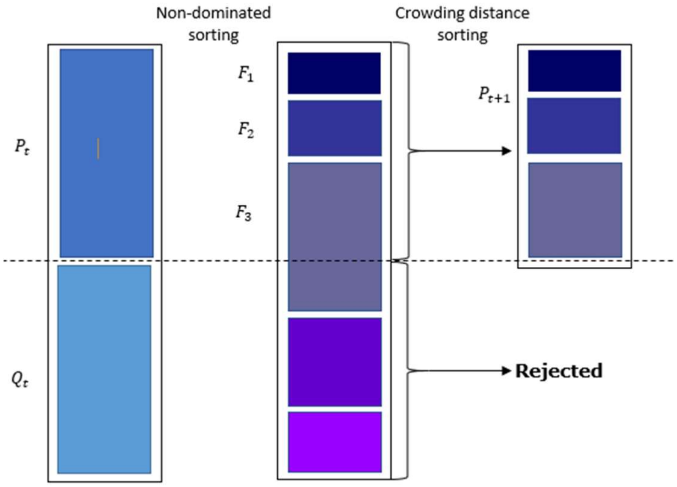


Figure 7.8 Sorting Processes of Individuals in NSGA-II

Another common multiobjective optimisation method is called Strength Pareto Evolutionary Algorithm (SPEA) [148-149]. The process of this algorithm is, firstly, a set P of initial population and an archive P' (an external empty set) are set. Secondly, all nondominated solutions in P are copied to the archive P' and all the dominated solutions in the archive P' are deleted. If the size of the solution set is more than the limit of the size of P', the solutions in the archive will be selected by a clustering technique. Thirdly, the fitness value of the solutions in P and P' are calculated. Fourthly, the solutions in P and P' are randomly selected for mutation. The solutions in the archive P' has higher chance to be selected because these solutions have the better fitness value. Finally, after all of these processes the set P are replaced by the offspring population. These processes will then be repeated until the ending condition is achieved such as the maximum working time. The method of calculating the fitness value of each solutions is shown below:

Step one: for each solution p_i^* in the archive P', a strength value S_i is calculated as

$$S_i = \frac{n}{N+1} \tag{7.8}$$

where n is the number of solutions dominated by the solution p^* and N is the number of the solutions in set P. The strength value of the solutions in P' are their fitness value.

Step two: the fitness value of each solution p_j in set P is equal to the sum of the strength value of the solutions that dominate solution p_j.

$$S_j = 1 + \sum_{i,i \geq j} S_i \qquad where\ S_j \in [1, N) \tag{7.9}$$

Comparing with NSGA-II, SPEA can find the pareto front faster with a small population size in general. However, the small population cannot guarantee the diversity of the population in searching optimal solutions. Moreover, SPEA is more computationally expensive as compared with NSGA-II. Therefore, NSGA-II is adopted in optimising the proposed PD-like controller with multiple objectives simultaneously.

7.2 Multi-objective PD-like Controller Optimisation

In Chapter 6, the PD-like control was optimised by PSO with two single objectives, namely minimising the maximum peak of external control input and switching duration, independently. However, these two objectives are conflicting with each other. The maximum peak of the external control input decreases while the switching duration increases or vice versa as illustrated in the last chapter. Compromised choice of optimal controller parameters is thus selected based on multi-objective optimisation.

Large external control force may damage the actuator that drive the plant because the required control input may exceed the output actuator capacity. Long switching duration implies waste of time in state switching. Therefore, the system is expected to switch from current, undesired state to the desired state with small external control force and short switching time in order to prolong lifespan of the system and reduce energy cost. The proposed PD-like control is optimised by the NSGA-II algorithm with two objectives simultaneously, namely minimizing the maximum peak of external force and the switching duration. Three control parameters namely the positive control gains k_p, k_d and the switch on time are considered in the multi-objective optimisation problem formulated as follows:

Minimise: $obj_1(x)$: $maximum\ peak\ of\ external\ control\ force\ \ max|u(t)|$ and
$obj_2(x)$: $switching\ duration\ \ \Delta t = t_2 - t_1$
Subject to: $h_j(x)$: $the\ nonlinear\ dynamic\ model\ of\ the\ test\ system$
x_k: $k_p,\ k_d\ and\ switching\ time\ t_1$

where t_1 is the time instant when the controller switches on and t_2 is the time instant when the controller switches off.

The following subsections present the multi-objective optimisation of the proposed PD-like controller on the three test systems.

7.2.1 Duffing Oscillator

NSGA-II algorithm is applied to optimise the PD-like controller applied on the Duffing oscillator with the two objectives simultaneously. The parameters of the Duffing oscillator are selected as $k = 0.9, \Gamma = 1.9$ and $\omega = 1.2$. Figure 7.9 shows the Pareto Front of the optimisation and Figure 7.10 shows the corresponding Pareto optimal set. All point in the Pareto optimal set can be chosen as an optimal solution based on different requirements or objectives priorities and user preferences. The solutions in Pareto optimal set can be divided into 3 groups and solutions within a group exhibit similar performance. Solution groupings are identified by visual inspection for small cases like this example. Clustering algorithms can be adopted for solution groupings in complicated cases with more than 3 objectives. Three solutions are arbitrarily selected from each group for system performance investigation and they are solutions #10, #19 and #31 (blue-coloured in Figure 7.10).

Figure 7.11 shows the results of the three chosen optimal solutions. Solution #31 minimises the switching duration while maximising the maximum peak of external control force. Solution #10 minimises the maximum peak of external control force and maximises the switching duration. Solution #19 outperforms solution #10 in reducing the switching duration by 79.28% but solution #10 outperforms solution #19 in decreasing the maximum peak of external control force by 65.05%. Moreover, solution #19 outperforms solution #31 in reducing the maximum peak of external control force by 83.98% but solution #31 outperforms solution #19 in shortening the switching duration by 88.53%. Solutions are adopted based on application, resource availability and user preference. For example, the set of parameters represented by solution #10 is selected for reducing the switching duration when an actuator is available for providing adequate power. The set of parameters represented by solution #31 is selected when the decision maker wants to save energy consumed by the system even though the switching duration may increase. The set of parameters represented by solution #19 is selected when the decision maker wants to have a balance between the maximum peak of external control force and the switching duration.

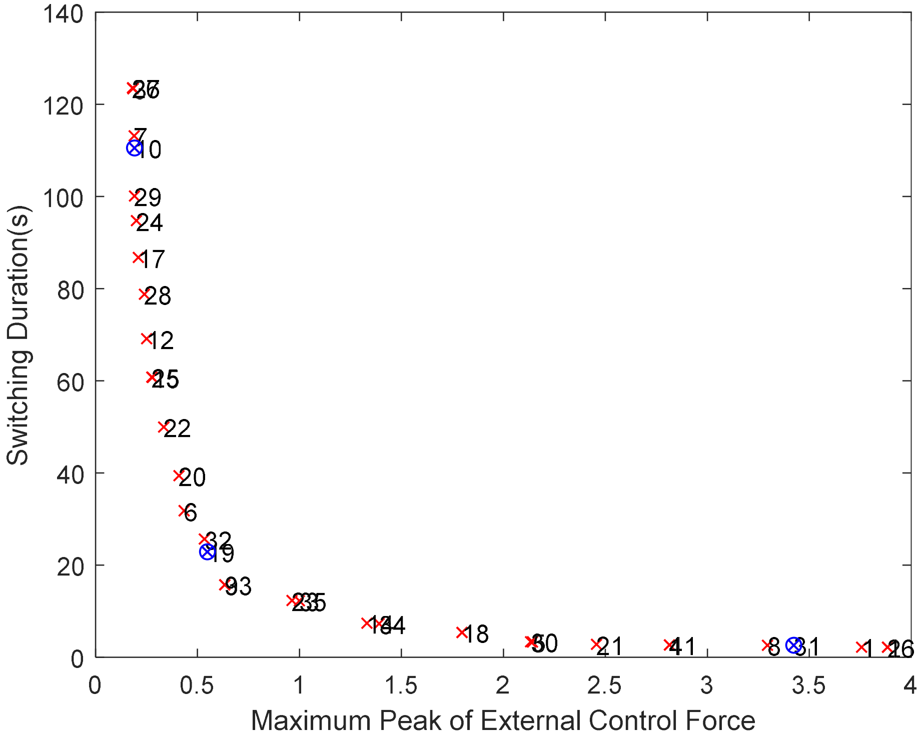
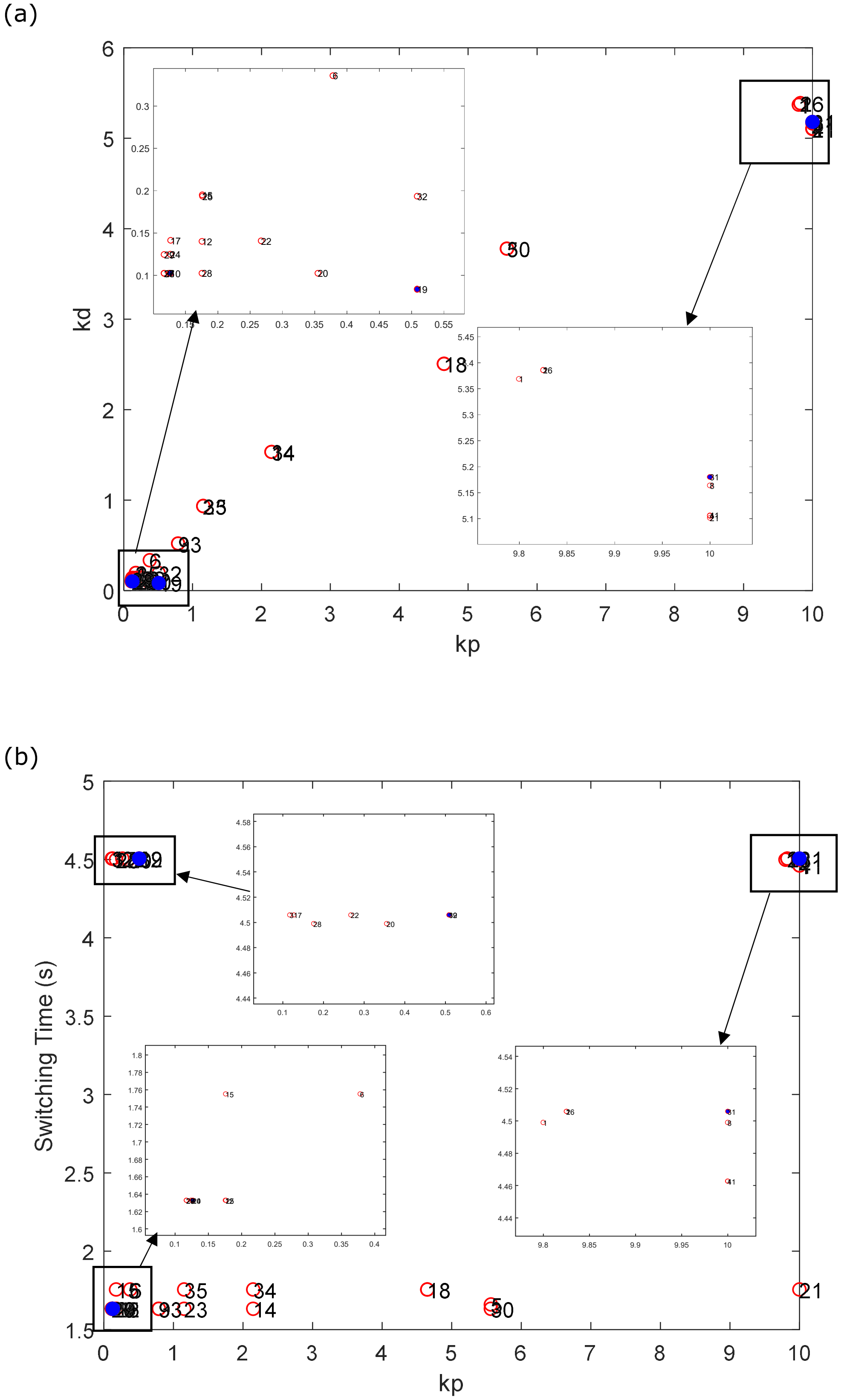


Figure 7.9 Pareto Front of Duffing oscillator with system parameters: $k = 0.9, \Gamma = 1.9$ *and* $\omega = 1.2$


(a)
kd
kp
0
1
2
3
4
5
6
7
8
9
10
30
18
34
35
93
6
32
16
41
(b)
Switching Time (s)
kp
1.5
2
2.5
3
3.5
4
4.5
5
35
34
18
30
21
14
23
93
15
6
41

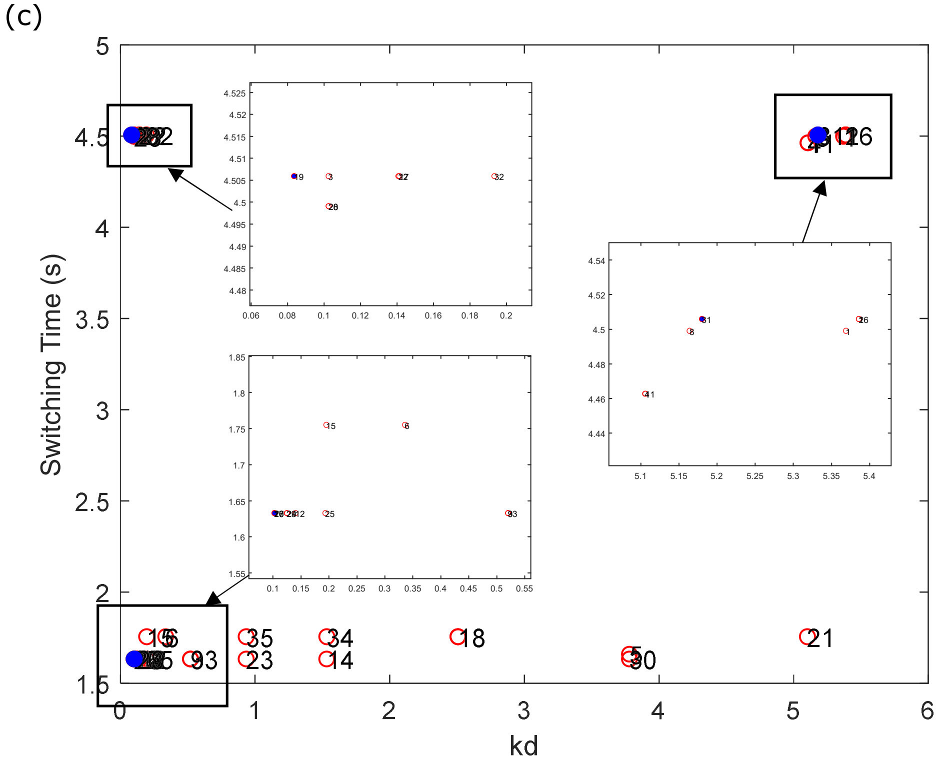


Figure 7.10 The corresponding Pareto Optimal set of Pareto Front, (a) k_p and k_d, (b) k_p and the switching time, (c) k_d and the switching time

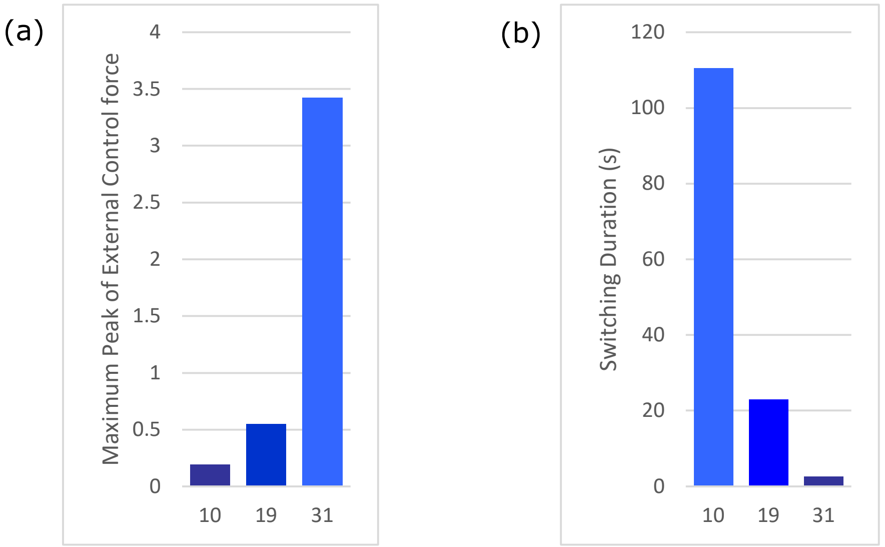


Figure 7.11 The Pareto optimal performance of the PD-like controller with the three chosen optimal solutions in the two objectives (a) maximum peak of external control force, (b) switching duration

The 14th set of control parameters are chosen for testing which has a better performance than the non-optimal controller (in the previous test on Duffing oscillator as shown in Figure 3.6) in both objectives. The chosen set of control parameters are: $k_p = 2.15$, $k_d = 1.53$ and the controller switches on at $t = 1.75$ second. The maximum peak of external control force and switching duration of the optimised control parameters are 1.39 and 7.30 seconds respectively. Figure 7.12 shows the control input to the Duffing oscillator with the Pareto optimal control parameters. Comparing with non-optimal controller (shown in Figure 3.6), the control input with the optimised control parameters decreases the maximum peak of external control input and switching duration by 16.61% and 51.50% respectively. Comparing with the controller optimised with the objective minimizing the maximum peak of external control force (shown in Figure 6.3), the switching duration decreases by 94.55% but the maximum peak of external control force increases by 87.77%. Moreover, comparing with the controller optimised with the objective minimizing the switching duration (shown Figure 6.4), the maximum peak of external control force decreases by 87.79% but the switching duration increases by 83.84%. Therefore, the PD-like control with this Pareto optimal control parameters can switch the Duffing oscillator to the desired state with a small external control force and short switching duration.

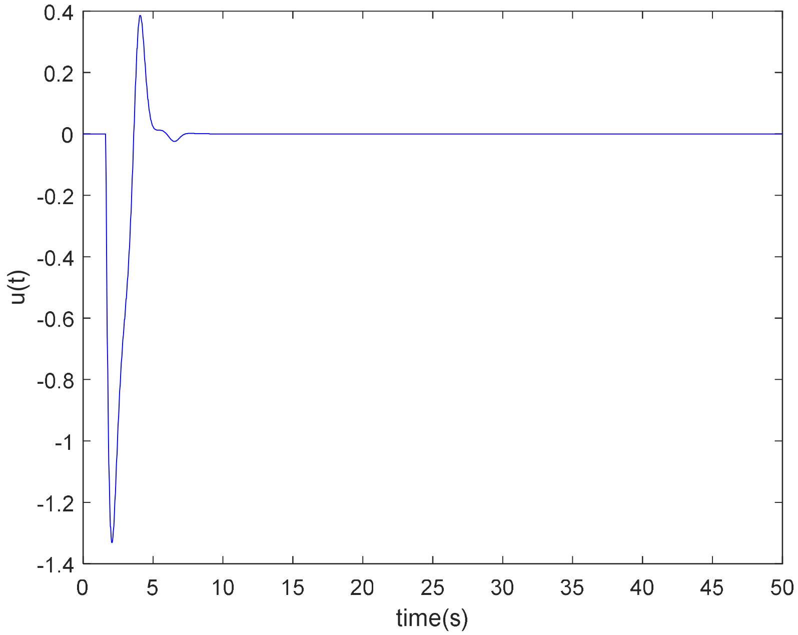


Figure 7.12 Control input u(t) to the Duffing oscillator using PD-like control law with Pareto optimal control parameters. $k_p = 2.15$, $k_d = 1.53$ *and switching on time* $t = 1.7522$ seconds

7.2.2 Soft impact oscillator

The NSGA-II algorithm is utilised for optimising the PD-like controller applied on the soft impact oscillator with the two objectives simultaneously. The parameters of soft impact oscillator under consideration are selected as $\xi = 0.01$, $\beta = 29$, $g = 1.26$, $\Gamma = 1.9$, and $\omega = 0.686$. Figure 7.13 illustrates the Pareto Front of the multi-objective optimisation result while Figure 7.14 presents the corresponding Pareto optimal set. All points in the Pareto optimal set can be selected as an optimal solution depending on various requirement or objective priorities. In addition, as the solutions in Pareto optimal set can be classified into distinct five groups, five representative solutions are selected from each group and they are #7, #9, #17, #18, and #27 (blue-coloured in Figure 7.14).

Figure 7.15 presents the results of the five chosen optimal solutions. It can be seen that solution #27 minimises the switching duration but maximises the maximum peak of external control force. While solution #9 minimises the external force's maximum peak, it increases the switching duration. Solutions #7, #17, and #18, on the other hand, outperform solution #27 in reducing the external force's maximum peak by 92.12%, 51.31%, and 87.75%, respectively; they also outperform solution #9 in shortening the switching duration by 39.51%, 94.04%, and 84.51%, respectively. However, solutions #7, #17, and #18 are outperformed by solution #27 in terms of the objective that reduces the switching duration by 94.57%, 44.9%, and 78.79%, respectively, and they are outperformed by solution #9 in decreasing the external force's maximum peak by 13.38%, 85.99%, and 44.29%, respectively. All five solutions can be considered as the optimal solution based on different requirements concerning switching duration and actuator limit. Moreover, the set of parameters represented by solutions #7, #9, and #18 can be selected for saving energy consumption when short switching duration is not a priority in performance. The set of parameters represented by solutions #17, #18, and #27 can also be selected for reducing the switching duration when an actuator is available for providing adequate power. Therefore, in these five solutions, solution #18 can be regarded as a good trade-off solution that both saves energy consumption and reduces the switching duration.

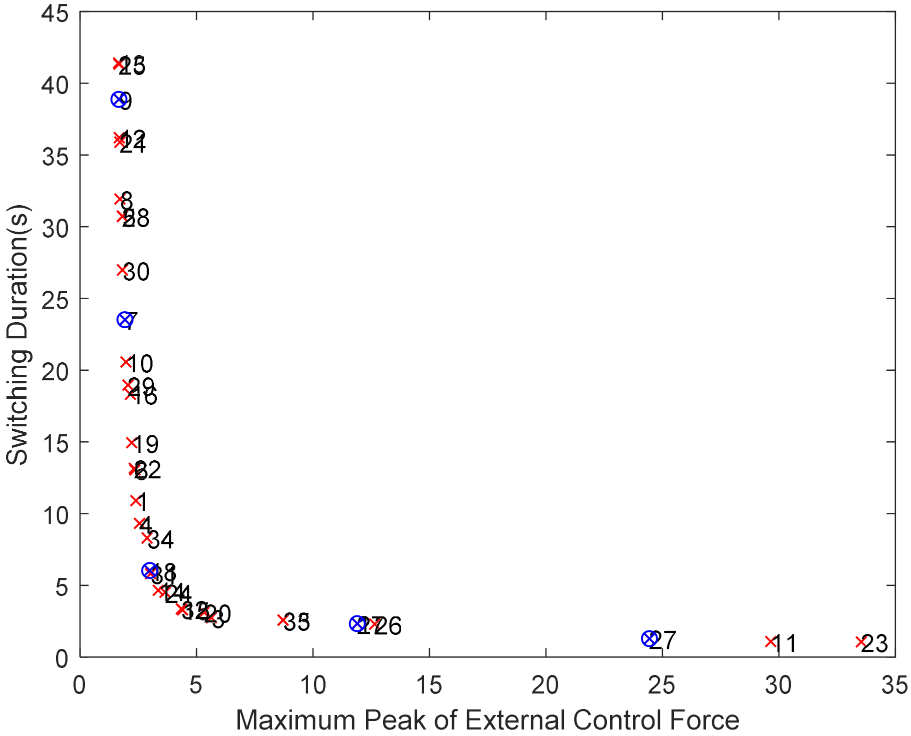
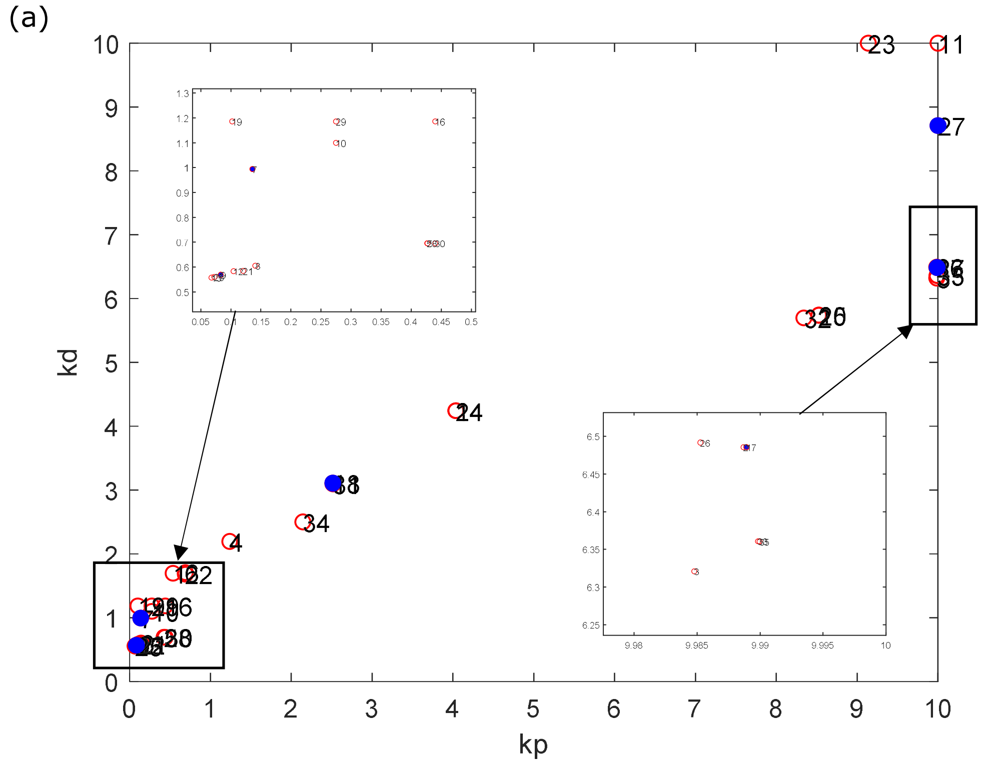


Figure 7.13 Pareto Front of soft impact oscillator with system parameters: ξ = 0.01, β = 29, g = 1.26, Γ = 1.9, and ω = 0.686



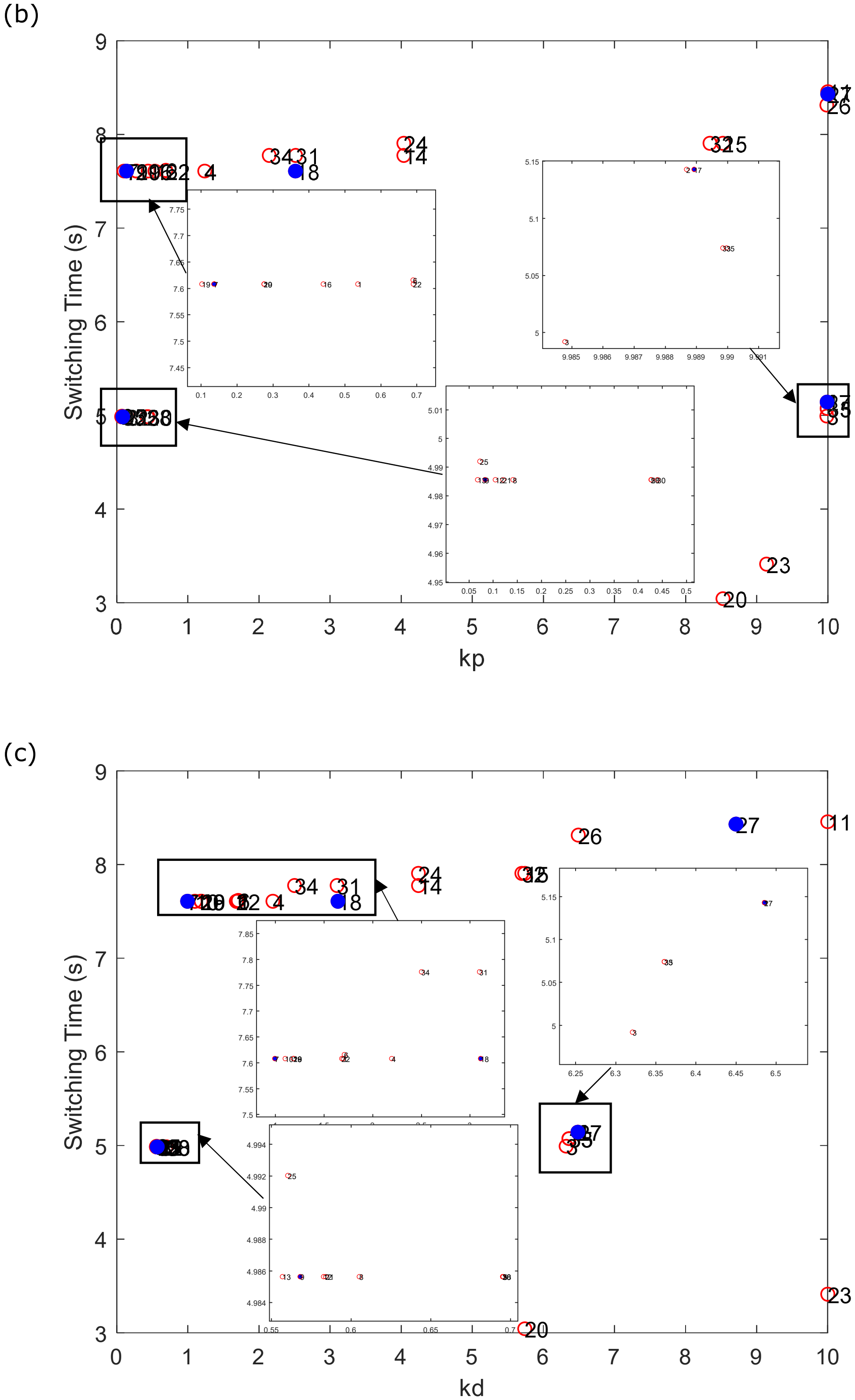


Figure 7.14 The corresponding Pareto Optimal set of Pareto Front for PD-like controller parameters by multi-objective optimisation, (a) k_p andk_d, (b) k_p and the switching time, (c) k_d and the switching time

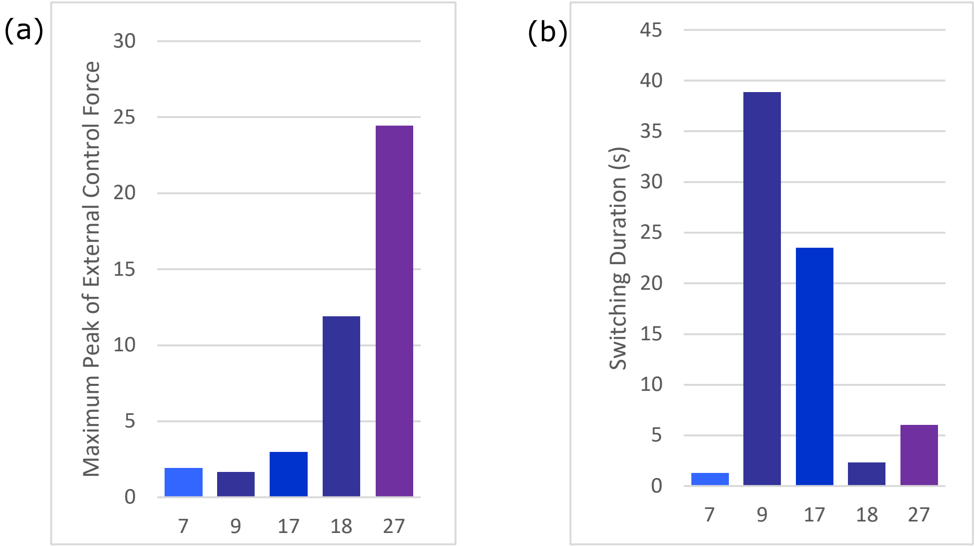
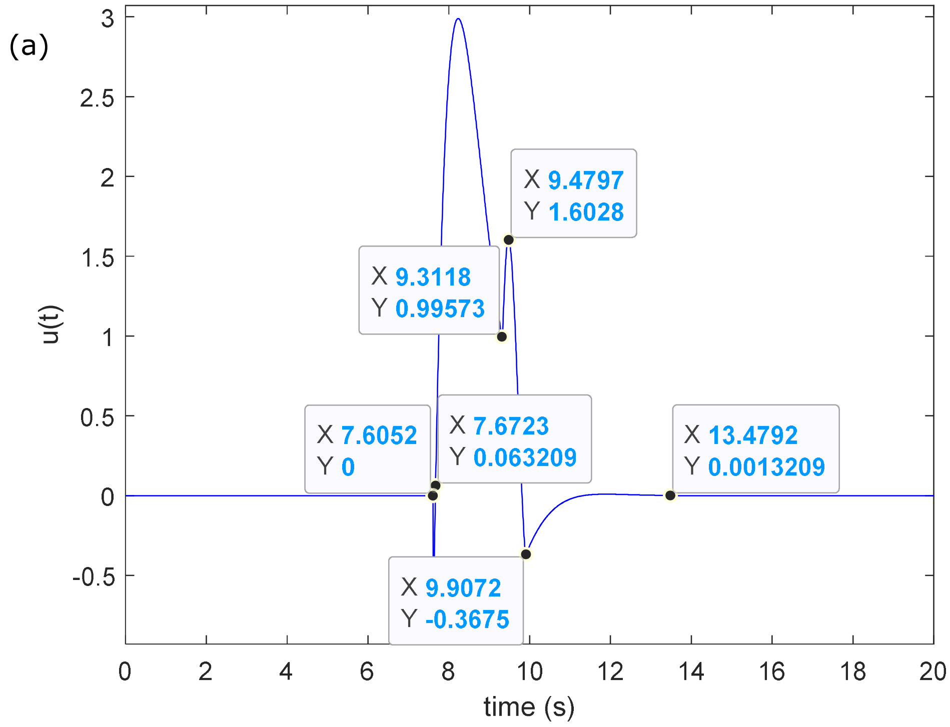


Figure 7.15 The Pareto optimal performance of the PD-like controller with the three chosen optimal solutions in the two objectives (a) maximum peak of external control force, (b) switching duration

The set of Pareto-optimal control parameters of solution #18 is chosen for testing which has a better performance than non-optimal controller (the previous test on soft impact oscillator shown in Figure 3.12). The chosen set of control parameters are: $k_p = 2.51$, $k_d = 3.11$ and the controller switches on at $t = 7.61$ second. Figures 7.16 (a), (b) and (c) show the control input response of the displacement and the trajectory of the system in the first 20 seconds with the chosen set of control parameters respectively. The process of switching the soft impact oscillator from the current state to the desired state is shown below. At about 7.6 seconds, the controller is switched on. The external control force is negative which means the direction of the force is opposite to the direction of the displacement of the mass. The control input is used to change the direction of the mass. At about 7.67 seconds, the external control force become positive, which means the direction of the external control force is the same as the direction of the force. The external control force is used to increase the velocity of the mass. At about 9.3 seconds, the displacement of the mass exceeds the gap between the mass and the second spring and the impact occurs. At about 9.48 seconds, the displacement of the mass achieves the maximum amplitude and at about 9.91 seconds, the displacement of the mass is less than the gap and the impact finishes. Within about 3.5 seconds, the trajectory of the system converges to the desired trajectory. In comparison of the performance of this

Pareto optimal controller with the non-optimal controller (shown Figure 3.12), the control input with the optimised control parameters decreases the maximum peak of external control input and switching duration by 5.21% and 72.84% respectively. Moreover, this Pareto optimised controller is compared with the single-objective optimised controllers. When the controller minimised maximum peak of external control force (shown in Figure 6.7) is compared with the controller with parameters from solution #18, the switching duration decreases by 90.41% but the maximum peak of external control force increases by 57.41%. Similarly, the multi-objective optimised PD-like controller (solution #18) decreases the switching duration by 82.75% and the maximum peak of the external control force by 91.33% compared to the controller with minimised switching duration alone (as shown Figure 6.8). Therefore, the PD-like control with this Pareto optimal control parameters can switch the soft impact oscillator to the desired state with a small external control force and short switching duration.



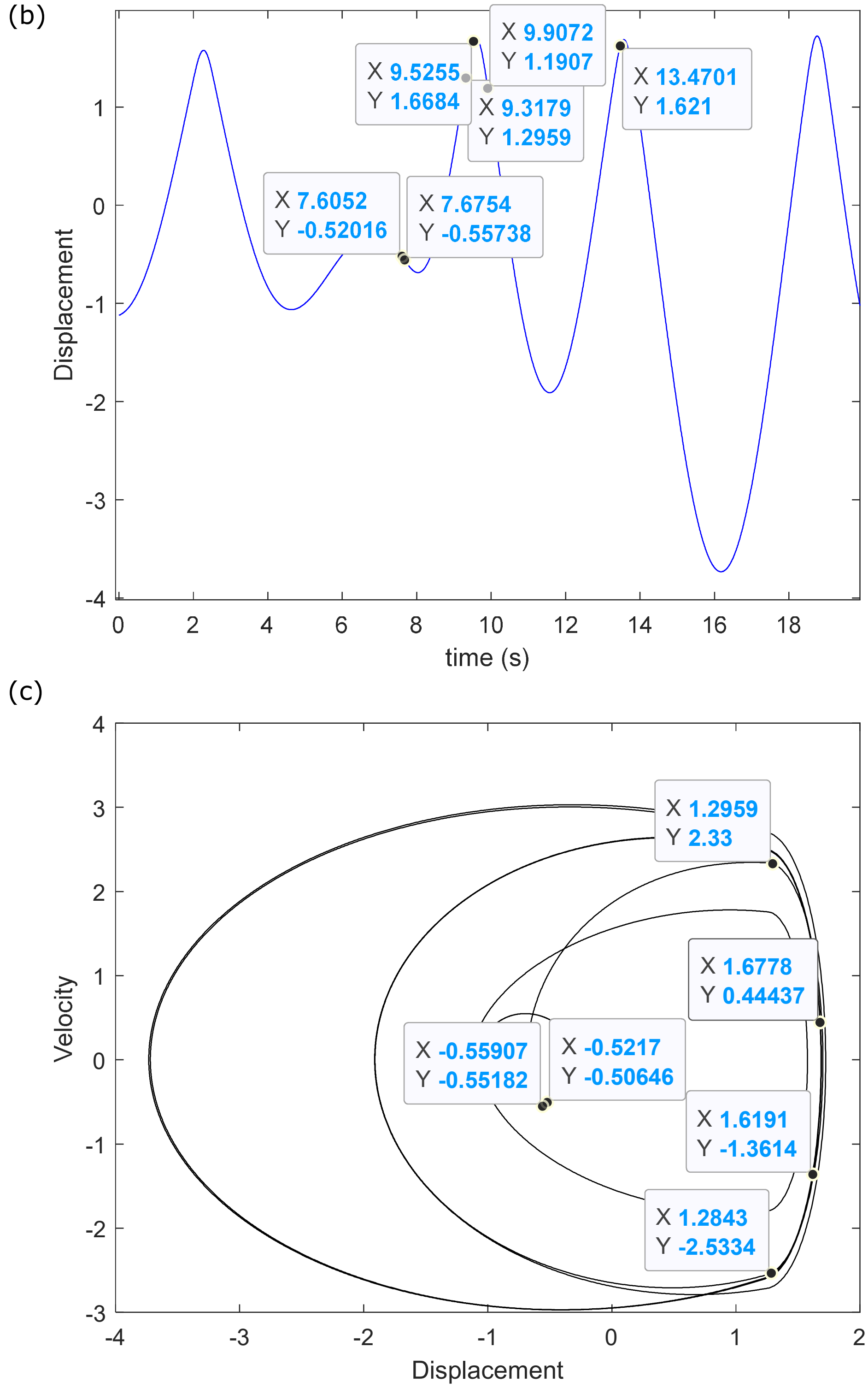


Figure 7.16 (a) Control input u(t) (b) response of displacement (c) state trajectory of the soft impact oscillator with Pareto optimal control parameters: $k_p = 2.51$, $k_d = 3.11$ *and switching on time* $t = 7.61$ *seconds*

7.2.3 Soft impact oscillator with a Drift

This section demonstrates how NSGA-II can be used for optimising the PD-like controller applied to the soft impact oscillator with a drift concerning the two objectives. The parameters of the soft impact oscillator with a drift include $a = 0.3$, $b = 0.2$, $\omega = 1$, $g = 0.02$, $\xi = 0.1$, and $\varphi = \pi/2$. Figures 7.17 and 7.18 illustrate the Pareto Front of the multi-objective optimisation results and the corresponding Pareto optimal set, respectively. Depending on different requirement or limitation of the applications, any points in the Pareto optimal set can be chosen by decision makers. The solutions in the Pareto optimal set can be divided into 4 groups. One representative solution is selected from each group. The four selected solutions (blue-coloured in Figure 7.18) are #5, #22, #25 and #27.

Figure 7.19 shows the result of the four chosen optimal solutions. Solution #5 minimises the switching duration but maximises the maximum peak of external control force. On the contrary, solution #27 minimises the maximum peak of external control force but maximise the switching duration. Solutions #22 and #25 outperform solution #27 in reducing the switching duration by 97.64% and 98.53% respectively but solution #27 outperforms solutions #22 and #25 in reducing the maximum peak of external force by 95.34% and 98.83% respectively. Moreover, solutions #22 and #25 outperform solution #5 in decreasing the maximum peak of external force by 82.30% and 29.49% respectively. In addition, solution #5 outperforms solutions #22 and #25 in shortening the switching duration by 55.51% and 28.73% respectively. Depending on different requirements or user preference, the sets of parameters represented by the four solutions are chosen. The set of parameters represented by solution #27 can save energy consumption while it spends lots of time in state switching. Both solutions #5 and #25 can shorten the switching duration while consuming a large amount of energy in state switching. The set of parameters represented by solution #22 is chosen if the decision maker wants to reduce both the switching duration and energy consumption.

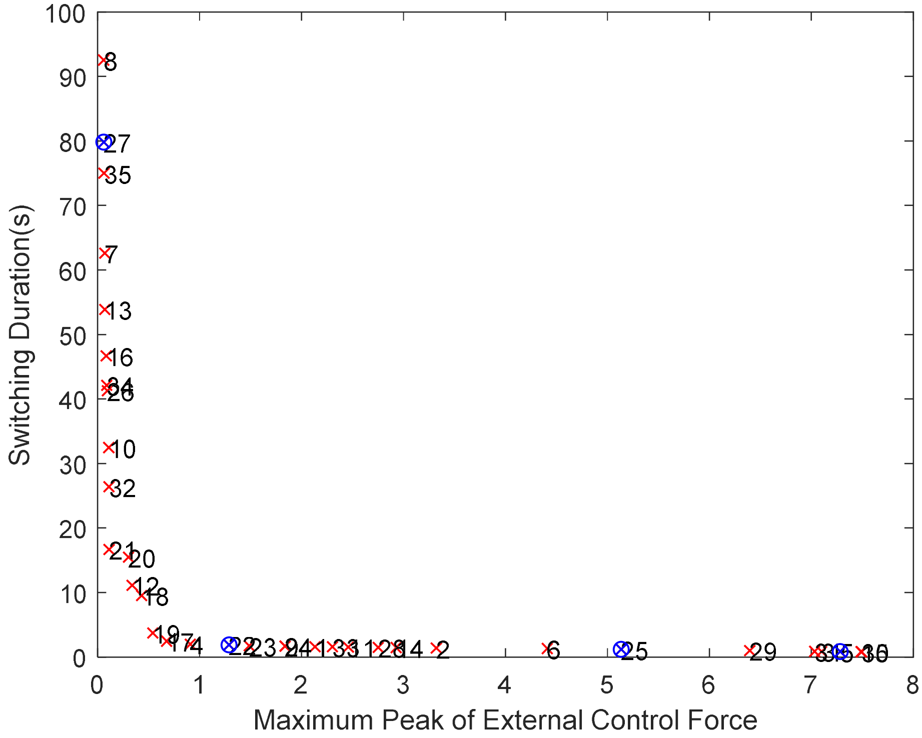
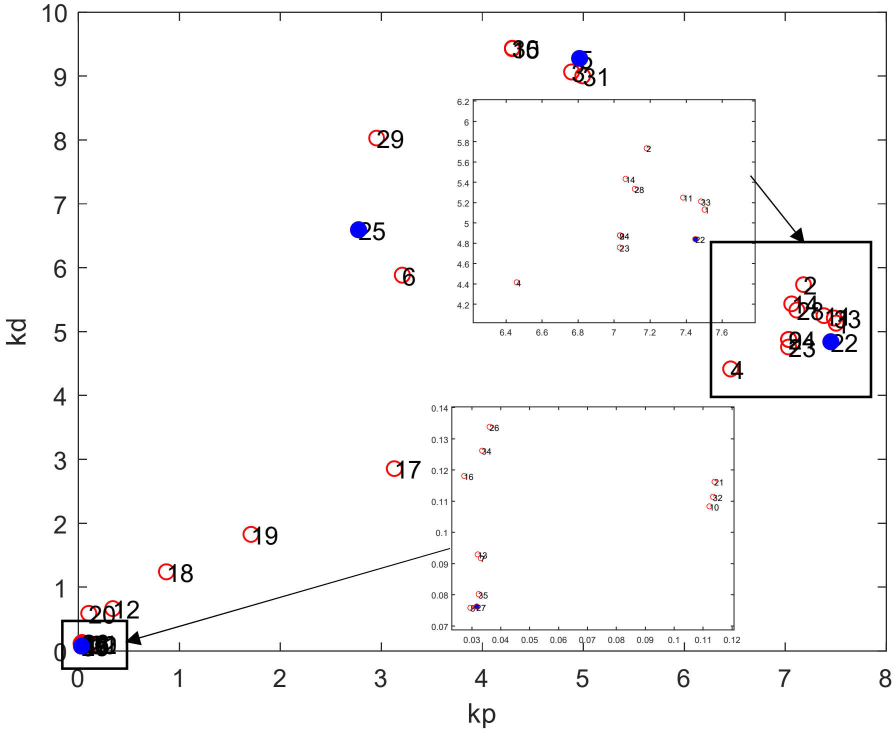


Figure 7.17 Pareto Front of the multi-objective optimisation results in PD-like controller applied on soft impact oscillator with a drift with system parameters: a=0.3, b=0.2, ω=1, g=0.02, ξ=0.1, and φ=π/2

(a)



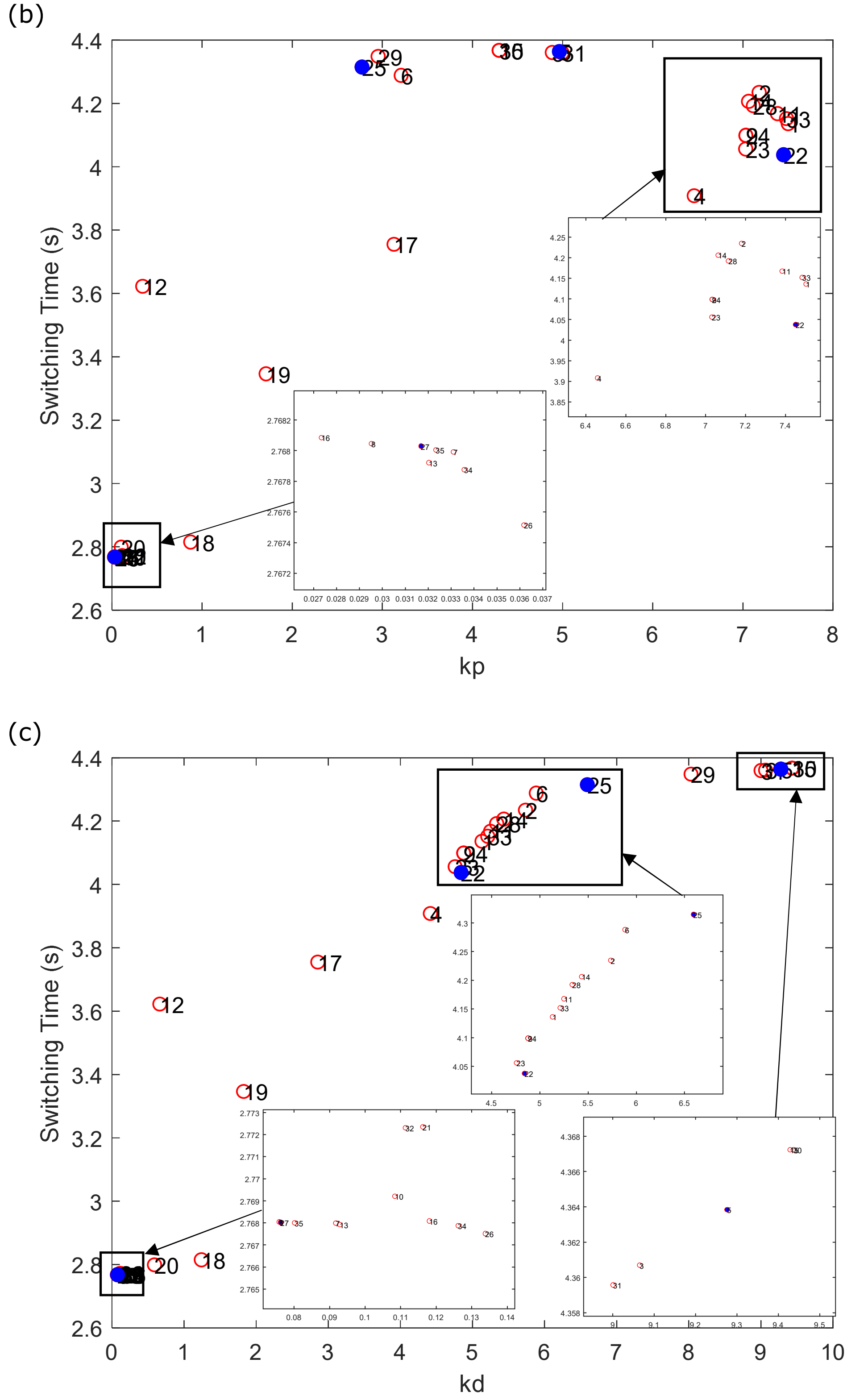


Figure 7.18 The corresponding Pareto Optimal set of Pareto Front, (a) k_p andk_d, (b) k_p and the switching time, (c) k_d and the switching time

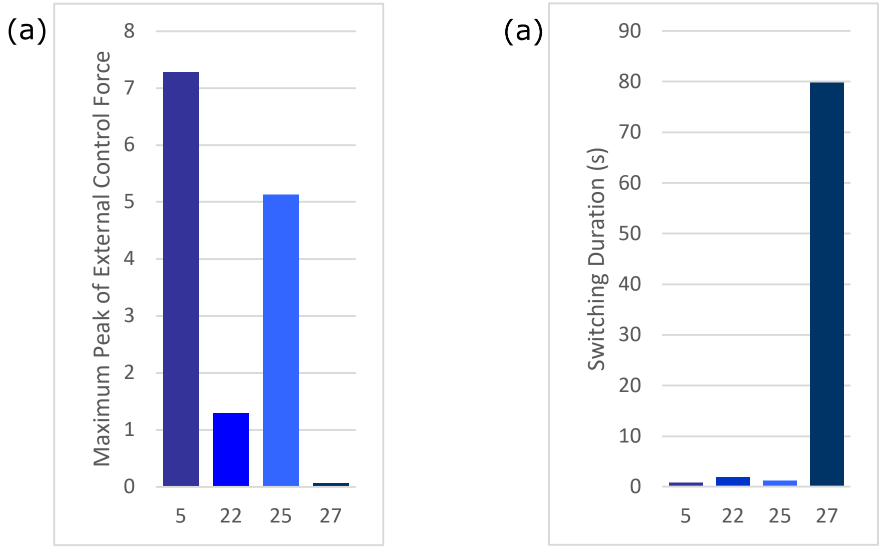


Figure 7.19 The Pareto optimal performance of the PD-like controller with the three chosen optimal solutions in the two objectives (a) maximum peak of external control force, (b) switching duration

The control parameters of solution #22 is chosen for testing which has a better performance than the non-optimal controller (the first test on soft impact oscillator with a drift, shown in Figure 3.12) in both objectives. The chosen Pareto-optimal set of control parameters are: $k_p = 7.45$, $k_d = 4.84$ and the controller switches on at $t = 4.04$ second. Figure 7.20 shows the control input to the soft impact oscillator with a drift with the Pareto-optimal control parameters. The maximum peak of external control force and switching duration of the optimised control parameters are 1.29 and 1.89 seconds respectively. This Pareto optimal controller is compared with three cases. Firstly, comparing with the non-optimal controller (shown in Figure 3.12), the control input with the Pareto optimised control parameters decreases the maximum peak of external control input and switching duration by 38.10% and 87.84% respectively. Secondly, comparing with the controller with minimised maximum peak of external control force (shown in Figure 6.11), the control input with the Pareto optimised control parameters decreases the switching duration by 98.65% but it increases the maximum peak of external control force by 95.74%. Thirdly, comparing with the controller with minimised switching duration (shown in Figure 6.12), the control input with the Pareto optimised control parameters decreases the maximum peak of external control input by 82.77% but it increases the switching duration by 57.14%. Therefore, the PD-like controller can switch the soft impact oscillator with a drift to the desired state with this set of Pareto optimal control parameters.

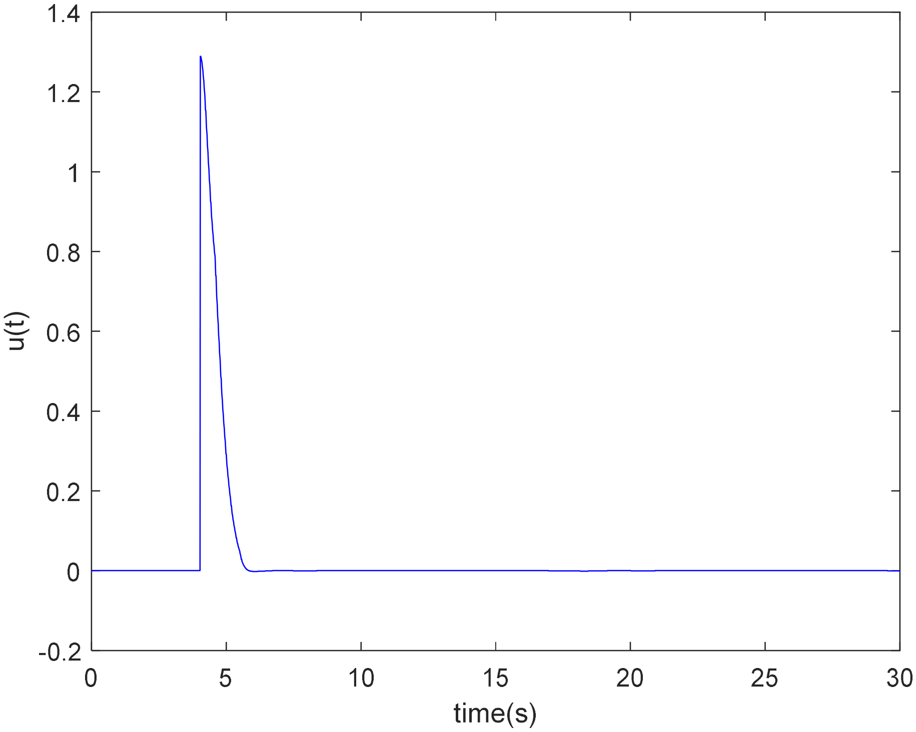


Figure 7.20 Control input u(t) to the soft impact oscillator with a drift using PD-like control law with Pareto optimal control parameters $k_p = 7.4507$, $k_d = 4.8411$ *and switching on time* $t = 4.0376$ *second*

7.3 Summary

In conclusion, the PD-like controller is optimised based on multiple conflicting objectives: minimising the maximum peak of external force and the switching duration. NSGA-II algorithm was applied for multi-objective optimisation of the proposed PD-like controller for state switching in multistable systems. Moreover, performance of the Pareto-optimal controllers was investigated on the three test systems namely Duffing oscillator, soft impact oscillator, and soft impact oscillator with a drift. All results in the Pareto optimal set can be chosen by the decision maker based on the requirements, limitations of application and the priorities of objectives.

Chapter 8 Conclusion and Future Work

This thesis has investigated state switching control of multistable systems so that they operate in the preferred mode for various applications. A fast method for extracting knowledge of the basins of attraction of the multistable systems was proposed in this work. A novel PD-like controller was proposed to switch a multistable system from undesired state to desired state. The proposed PD-like controller was then optimised based on various performance objectives independently and simultaneously using single objective optimisation and multi-objective optimisation respectively.

8.1 Conclusion

In this thesis, a fast method for approximation of basins of attraction of multistable systems was proposed. The proposed method is called Randomised Triangular Subdivision and is based on Delaunay Triangulation. The performance of this method is investigated by a Duffing oscillator and a soft impact oscillator, which are typical multistable systems, with different number of co-existing states. Comparing with the traditional brute force algorithm, with the same number of initial conditions, the proposed randomised triangular subdivision method can estimate the basins of attraction with better accuracy and can capture geometrical and topological features at low resolution.

A PD-like control was proposed in this thesis to switch the system from current undesired state to the desired state without changing the system parameters and destroying the original basins of attraction. The knowledge of nonlinearity of the systems is not required in synthesising the proposed PD-like controller. Moreover, it can maintain the systems in the desired state when the systems are subjected to external disturbance or noise. The principle of the controller is to apply an external control force which is dependent on the difference between the trajectories of the current state and desired state. This controller is investigated on two common multistable systems, namely Duffing oscillator and soft impact oscillator, which are simple one degree of freedom examples of smooth and non-smooth systems respectively. Moreover, for further study of the capability of the proposed controller, the proposed controller is also tested on the soft impact oscillator with a drift system which is a complex multistable system for modelling percussive drilling. As demonstrated by simulations, the proposed PD-like controller was able to switch states in the three test systems successfully.

This thesis also considered practical application of the proposed PD-like controller. In real life, actuators that are used to drive the plant/process cannot provide unlimited strength. The constrained PD-like controller was then proposed in order to investigate the behaviours of the proposed controller under the consideration of the limited capacity of actuators in the systems. This method limits the strength of the control input by setting a bound for the control input based on the technical specification of the actuators. Compared with the performance of the unconstrained PD-like controller, the maximum peak of the control input is decreased and within actuator limits. However, the constrained PD-like control takes more time to switch the system to the desired state. If the actuator can provide enough power, the system will be switched to the desired state immediately. If the actuator provides small power, the system will be switched to the desired state through several intermediate stages.

The performance of the PD-like controller is dependent on the controller parameters. Therefore, the performance of the PD-like controller can be optimised by finding a suitable set of control parameters. In this thesis, Particle Swarm Optimisation (PSO) was used to minimise the controller based on two single performance metrics or objectives, namely the maximum peak of the control input and the switching duration. The result of PSO was obtained through extensive numerical simulation. According to the simulation results, these two objectives are conflicting with each other. When the maximum peak of external control force is minimised, the switching duration will increase and vice versa. Therefore, the controller should be optimised based on multiple objectives simultaneously. NSGA-II algorithm was applied to optimise the controller with the two objectives mentioned above at the same time. The Pareto optimal results were obtained through extensive numerical simulation. All results in the Pareto optimal set can be chosen by the decision maker based on the requirements, limitations of application and the priorities of objectives or user's preference.

8.2 Future work

The research study can be further extended in the following directions:

1. In chapter 2, the Randomised Triangular Subdivision was proposed to estimate the basins of attraction of multistable systems. The area of the convex hull which can cover the most part of the region of interest in the initial condition space was considered. In order to ensure the diversity of the initial conditions selected in the first Delaunay Triangulation of the space, the distribution of the random sample of the initial condition is expected to be as uniform as possible. Latin Hypercube Sampling will be adopted to ensure thorough coverage of the initial condition space [150]. Moreover, smooth boundary between different states should be reconstructed by the introduction of adaptive subdivision methods. Furthermore, sometimes, there are lots of small triangles which are close to each other are in the same state in the estimated basins of attraction. Intelligent merging algorithm will be devised to combine these small triangles so as to reduce memory storage for high dimensional multistable systems. Adaptive subdivision and merging algorithm will then be developed to intelligently identify smooth separatrix at high level of details while covering homogeneous regions of initial conditions in the same basin of attraction with fewer triangles as they only require low level of details in the future.

2. In chapter 3, the PD-like controller was proposed for state switching in multistable systems. However, this controller requires the trajectory of the desired state for control feedback which is sometimes difficult to obtain. Moreover, sometimes no dynamic model of the system is available and the physical systems cannot be excited in many trials with different initial conditions for identification of the basins of attraction. The PD-like controller can be extended so that it can switch

the system with less information about the desired state and for instance only an arbitrary initial condition which is located in the basin of the desired state is needed. This method switches the system to the desired state by driving the system to the neighbourhood of that condition. This new method does not require complete information about the desired trajectory and is easy to apply. However, it cannot maintain the system in the desired state under the effect of external noise. This limitation can be easily overcome by recording the desired trajectory when the system is switched to the desired state and then applying the proposed PD-like controller to maintain the system in the desired state.

3. In chapter 7, the PD-like controller was optimised by the NSGA-II algorithm with two objectives simultaneously. More objectives can be considered in enhancing the performance of the proposed PD-like controller. For example, smoothness of the external control input is critical to the health of the actuator involved so that the actuator does not need to generate spurious output between two extremes continuously. Smoothness of the external control input can be estimated by the number of zero-crossing in the function $u(t)$. Therefore, the direction of the external control force will not change too many times in a short switching duration.

4. In this thesis, the performance of the proposed PD-like controller was investigated by extensive numerical simulation. In future work, the performance of the controller will be tested using experimental drilling test rig. Since, in real life experiment, some factors that affect the performance of the controller are not considered and are difficult to model in simulation. For instance, the controller cannot switch on exactly at the same time as the setting in numerical simulation. Moreover, for soft impact oscillator and soft impact oscillator with a drift, in the experiment, the spring and damper may be easily damaged because of the large external control force suggested by the controller. The physical parts in the test rig become fatigue easily over prolonged and heavy duty operation.

5. The stability analysis of the proposed PD-like controller will be studied in the future. One method to prove the stability of the controller is to use Lyapunov Stability Theorem. Moreover, Lyapunov function can also be constructed numerically [151-153].

Reference

[1] MAURER, J. and LIBCHABER, A., 1980. Effect of the Prandtl number on the onset of turbulence in liquid 4He. *Journal de Physique lettres*, 41(21), pp.515-518.

[2] THOMPSON, J.M.T. and STEWART, H.B., 2002. *Nonlinear dynamics and chaos*. John Wiley & Sons.

[3] FOSS, J., LONGTIN, A., MENSOUR, B. and MILTON, J., 1996. Multistability and delayed recurrent loops. *Physical Review Letters*, 76(4), p.708.

[4] DEPOUHON, A., DENOEL, V. and DETOURNAY, E., 2013. A drifting impact oscillator with periodic impulsive loading: Application to percussive drilling. *Physica D: Nonlinear Phenomena*, 258, pp.1-10.

[5] PISARCHIK, A.N. and Feudel, U., 2014. Control of multistability. *Physics Reports*, 540(4), pp.167-218.

[6] KUZMA, P., KAPITANIAK, M. and KAPITANIAK, T., 2014. Coupling multistable systems: uncertainty due to the initial positions on the attractors. *JOURNAL OF THEORETICAL AND APPLIED MECHANICS,* 52(1), PP.281-284.

[7] WU, Z., HARNE, R.L. and WANG, K.W., 2014. Energy harvester synthesis via coupled linear-bistable system with multistable dynamics. *Journal of Applied Mechanics*, 81(6), p.061005.

[8] LIU, Y., WIERCIGROCH, M., PAVLOVSKAIA, E. and Yu, H., 2013. Modelling of a vibro-impact capsule system. *International Journal of Mechanical Sciences*, *66*, pp.2-11.

[9] WIERCIGROCH, M., NEILSON, R.D. and Player, M.A., 1999. Material removal rate prediction for ultrasonic drilling of hard materials using an impact oscillator approach. *Physics Letters A*, *259*(2), pp.91-96.

[10] PAVLOVSKAIA, E., HENDRY, D.C. and WIERCIGROCH, M., 2015. Modelling of high frequency vibro-impact drilling. *International Journal of Mechanical Sciences*, *91*, pp.110-119.

[11] KANEKO, K., 1990. Clustering, coding, switching, hierarchical ordering, and control in a network of chaotic elements. *Physica D: Nonlinear Phenomena*, 41(2), pp.137-172.

[12] KANEKO, K., 1989. Chaotic but regular posi-nega switch among coded attractors by cluster-size variation. *Physical Review Letters*, 63(3), p.219.

[13] CHIZHEVSKY, V.N., CORBALAN, R. and PISARCHIK, A.N., 1997. Attractor splitting induced by resonant perturbations. *Physical Review E*, *56*(2), p.1580.

[14] CHIZHEVSKY, V.N., VILASECA, R. and CORBALAN, R., 1998. Experimental switchings in bistability domains induced by resonant perturbations. *International Journal of Bifurcation and Chaos*, 8(09), pp.1777-1782.

[15] CARROLL, T.L. and PECORA, L.M., 1993. Using chaos to keep period-multiplied systems in phase. *Physical Review E*, 48(4), p.2426-2436.

[16] PECORA, L.M. and CARROLL, T.L., 1991. Pseudoperiodic driving: eliminating multiple domains of attraction using chaos. *Physical review letters*, 67(8), p.945-948.

[17] KENNEDY, J. and EBERHART, R, 1995. Particle swarm optimization. *Proc. IEEE Int. Conf. Neural Netw. (ICNN)*, 4, pp. 1942-1948

[18] KOVACIC, I. and BRENNAN, M.J., 2011. *The Duffing equation: nonlinear oscillators and their behaviour.* John Wiley & Sons.

[19] NIJMEIJER, H. and BERGHUIS, H., 1995. On Lyapunov control of the Duffing equation. *IEEE Transactions on Circuits and Systems I: Fundamental theory and applications*, 42(8), pp.473-477.

[20] QAISI, M.I., 1996. Analytical solution of the forced Duffing's oscillator. *Journal of sound and vibration*, 194(4), pp.513-520.

[21] KANAMARU, T., 2007. Van der Pol oscillator. *Scholarpedia*, *2*(1), p.2202.

[22] ING, J., PAVLOVSKAIA, E., WIERCIGROCH, M. and BANERJEE, S., 2010. Bifurcation analysis of an impact oscillator with a one-sided elastic constraint near grazing. *Physica D: Nonlinear Phenomena*, 239(6), pp.312-321.

[23] ING, J., 2008. *Near grazing dynamics of piecewise linear oscillators* (Doctoral dissertation, University of Aberdeen).

[24] ING, J., PAVLOVSKAIA, E., WIERCIGROCH, M. and BANERJEE, S., 2007. Experimental study of impact oscillator with one-sided elastic constraint. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1866), pp.679-705.

[25] HINRICHS, N., OESTREICH, M. and POPP, K., 1997. Dynamics of oscillators with impact and friction. *Chaos, Solitons & Fractals*, 8(4), pp.535-558.

[26] SHAW, S.W., 1986. On the dynamic response of a system with dry friction. *Journal of Sound and Vibration*, 108(2), pp.305-325.

[27] MCMILLAN, A.J., 1997. A non-linear friction model for self-excited vibrations. *Journal of sound and vibration*, 205(3), pp.323-335.

[28] AJIBOSE, O.K., WIERCIGROCH, M., PAVLOVSKAIA, E. and AKISANYA, A.R., 2010. Global and local dynamics of drifting oscillator for different

contact force models. *International Journal of Non-Linear Mechanics*, 45(9), pp.850-858.

[29] PAVLOVSKAIA, E., WIERCIGROCH, M. and GREBOGI, C., 2001. Modelling of an impact system with a drift. *Physical Review E*, 64(5), pp.056224-1-0560224-9

[30] PAVLOVSKAIA, E. and WIERCIGROCH, M., 2003. Periodic solution finder for an impact oscillator with a drift. *Journal of Sound and Vibration*, 267(4), pp.893-911.

[31] LUO, G.W. and XIE, J.H., 1998. Hopf bifurcation of a two-degree-of-freedom vibro-impact system. *Journal of Sound and Vibration*, 213(3), pp.391-408.

[32] NUSSE, H.E. and YORKE, J.A., 1996. Basins of attraction. *Science,* 271(5254), pp. 1376-1380

[33] DE ROSSI, A., PERRRACCHIONE, E. and VENTURINO, E., 2018. Meshless partition of unity method for attraction basins of periodic orbits: Fast detection of separatrix points. *Dolomites Research Notes on Approximation,* 11, pp.15-22

[34] CAVORETTO, R., DE ROSSI, A., PERRACCHIONE, E. and VENTURINO, E., 2016. Robust approximation algorithms for the detection of attraction basins in dynamical systems. *Journal of Scientific Computing*, *68*(1), pp.395-415.

[35] FORTUNE, S., 1995. Voronoi diagrams and Delaunay triangulations. In *Computing in Euclidean geometry,* pp. 225-265.

[36] AURENHAMMER, F., KLEIN, R. and LEE, D.T., 2013. *Voronoi diagrams and Delaunay triangulations*. World Scientific Publishing Company.

[37] GARTNER, B. and HOFFMANN, M., 2014. Computational Geometry Lecture Notes1 HS 2012.

[38] LEE, D.T. and SCHACHTER, B.J., 1980. Two algorithms for constructing a Delaunay triangulation. *International Journal of Computer & Information Sciences*, *9*(3), pp.219-242.

[39] GUIBAS, L.J., KNUTH, D.E. and SHARIR, M., 1990, July. Randomized incremental construction of Delaunay and Voronoi diagrams. In *International Colloquium on Automata, Languages, and Programming,* pp. 414-431.

[40] CHEW, L.P., 1989. Constrained delaunay triangulations. *Algorithmica*, 4(1-4), pp.97-108.

[41] SLOAN, S.W., 1987. A fast algorithm for constructing Delaunay triangulations in the plane. *Advances in Engineering Software (1978)*, 9(1), pp.34-55.

[42] SU, P. and DRYSDALE, R.L.S., 1997. A comparison of sequential Delaunay triangulation algorithms. *Computational Geometry*, 7(5-6), pp.361-385.

[43] LAWSON, C.L., 1986. Properties of n-dimensional triangulations. *Computer Aided Geometric Design*, 3(4), pp.231-246.

[44] SHEWCHUK, R., 2005. Star splaying: an algorithm for repairing Delaunay triangulations and convex hulls. *Proceedings of the twenty-first annual symposium on Computational geometry,* pp. 237-246

[45] EDELSBRUNNERS, H., 1993. Geometric algorithms. *Handbook of Convex Geometry,* pp. 699-735

[46] GAO, M., CAO, T.T., TAN, T.S. and HUANG, Z., 2013, March. Flip-flop: convex hull construction via star-shaped polyhedron in 3D. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games,* pp. 45-54.

[47] BRONNIMANN, H., KETTNER, L., POCCHIOLA, M. and SNOEYINK, J., 2001. Counting and enumerating pseudo-triangulations with the greedy flip algorithm. *Manuscript,* PP. 1-19.

[48] ŽALIK, B. and KOLINGEROVA, I., 2003. An incremental construction algorithm for Delaunay triangulation using the nearest-point paradigm. *International Journal of Geographical Information Science*, 17(2), pp.119-138.

[49] BARBER, C.B., DOBKIN, D.P. and HUHDANPAA, H., 1996. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software (TOMS)*, 22(4), pp.469-483.

[50] ANGLADA, M.V., 1997. An improved incremental algorithm for constructing restricted Delaunay triangulations. *Computers & Graphics*, 21(2), pp.215-223.

[51] CIGNONI, P., MONTANI, C. and SCOPIGNO, R., 1998. DeWall: A fast divide and conquer Delaunay triangulation algorithm in Ed. *Computer-Aided Design*, 30(5), pp.333-341.

[52] LISCHINSKI, D., 1994. Incremental delaunay triangulation. *Graphics gems IV*, pp.47-59.

[53] KUNTSEVICH, B.F., PISARCHIK, A.N., CHIZHEVSKII, V.N. and CHURAKOV, V.V., 1983. Amplitude modulation of the radiation of a CO 2-laser by optically controlled absorption in semiconductors. *Journal of Applied Spectroscopy*, 38(1), pp.107-112.

[54] CHIZHEVSKY, V.N. and TUROVETS, S.I., 1993. Small signal amplification and classical squeezing near period-doubling bifurcations in a modulated CO2-laser. *Optics communications*, 102(1-2), pp.175-182.

[55] SAMSON, A.M., TUROVETS, S.I., CHIZHEVSKY, V.N. and CHURAKOV, V.V., 1992. Nonlinear dynamics of a loss-switched CO2 laser. *Sov. Phys. JETP*, 74(4), pp.628-639.

[56] CHIZHEVSKY, V.N. and TUROVETS, S.I., 1994. Periodically loss-modulated CO 2 laser as an optical amplitude and phase multitrigger. *Physical Review A*, 50(2), p.1840-1843.

[57] CHIZHEVSKY, V.N., GRIGORIEVA, E.V. and KASHCHENKO, S.A., 1997. Optimal timing for targeting periodic orbits in a loss-driven CO2 laser. *Optics communications*, 133(1-6), pp.189-195.

[58] CHIZHEVSKY, V.N., 2000. Coexisting attractors in a CO2 laser with modulated losses. *Journal of Optics B: Quantum and Semiclassical Optics*, 2(6), p.711-717.

[59] VELDMAN, D.W., FEY, R.H. and ZWART, H., 2017. Impulsive steering between coexisting stable periodic solutions with an application to vibrating plates. *Journal of computational and nonlinear dynamics*, 12(1), p.1-27.

[60] LI, C., XU, W., WANG, L. and WANG, Z., 2014. Impulsive control of sticking motion in van der Pol one-sided constraint system. *Applied Mathematics and Computation*, 248, pp.363-370.

[61] OTT, E. and SPANO, M., 1996. Controlling chaos. In *AIP Conference Proceedings*, 375(1), pp. 92-103.

[62] PISARCHIK, A.N., 2001. Controlling the multistability of nonlinear systems with coexisting attractors. *Physical Review E*, 64(4), pp .046203-1-046203-5.

[63] GOSWAMI, B.K. and PISARCHIK, A.N., 2008. Controlling multistability by small periodic perturbation. *International Journal of Bifurcation and Chaos*, 18(06), pp.1645-1673.

[64] PYRAGAS, K., 2006. Delayed feedback control of chaos. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 364(1846), pp.2309-2334.

[65] MARTINEZ-ZEREGA, B.E., PISARCHIK, A.N. and TSIMRING, L.S., 2003. Using periodic modulation to control coexisting attractors induced by delayed feedback. *Physics Letters A*, 318(1-2), pp.102-111.

[66] LIU, Y., WIERCIGROCH, M., ING, J. and PAVLOVSKAIA, E., 2013. Intermittent control of coexisting attractors. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1993), PP.1-15

[67] RONCO, E., ARSAN, T. and GAWTHROP, P.J., 1999. Open-loop intermittent feedback control: practical continuous-time GPC. *IEE Proceedings-Control Theory and Applications*, 146(5), pp.426-434.

[68] WANG, Y., HAO, J. and ZUO, Z., 2010. A new method for exponential synchronization of chaotic delayed systems via intermittent control. *Physics Letters A*, 374(19-20), pp.2024-2029.

[69] LI, C., LIAO, X. and HUANG, T., 2007. Exponential stabilization of chaotic systems with delay by periodically intermittent control. *Chaos: An interdisciplinary journal of nonlinear science*, 17(1), p.013103.

[70] GAWTHROP, P.J. and WANG, L., 2009. Event-driven intermittent control. *International Journal of Control*, 82(12), pp.2235-2248.

[71] INSPERGER, T., 2006. Act-and-wait concept for continuous-time control systems with feedback delay. *IEEE Transactions on Control Systems Technology*, 14(5), pp.974-977.

[72] GAWTHROP, P., 2009. Act-and-wait and intermittent control: some comments. *IEEE transactions on control systems technology*, 18(5), pp.1195-1198.

[73] LI, C., FENG, G. and LIAO, X., 2007. Stabilization of nonlinear systems via periodically intermittent control. *IEEE Transactions on Circuits and Systems II: Express Briefs*, *54*(11), pp.1019-1023.

[74] BESANCON, G., 2007. *Nonlinear observers and applications* (Vol. 363). Springer-Verlag Berlin Heidelberg.

[75] PARKINSON, A.R., BALLING, R., and HEDENGREN, J.D., 2018. *Optimization Methods for Engineering Design*, Brigham Young University.

[76] CHANG, K.H., 2014. *Design theory and methods using CAD/CAE: The computer aided engineering design series*. Academic Press.

[77] BOULAXIS, N.G. and PAPADOPOULOS, M.P., 2002. Optimal feeder routing in distribution system planning using dynamic programming technique and GIS facilities. *IEEE Transactions on Power Delivery*, 17(1), pp.242-247.

[78] HUANG, K.Y., 2003. Demand subscription services-an iterative dynamic programming for the substation suffering from capacity shortage. *IEEE Transactions on Power Systems*, 18(2), pp.947-953.

[79] DIAZ-DORADO, E., MIGUEZ, E. and CIDRAS, J., 2001. Design of large rural low-voltage networks using dynamic programming optimization. *IEEE Transactions on Power Systems*, 16(4), pp.898-903.

[80] BERTSEKAS, D.P., 1995. *Dynamic programming and optimal control*, Belmont, MA: Athena scientific.

[81] DIAZ-DORADO, E. and PIDRE, J.C., 2004. Optimal planning of unbalanced networks using dynamic programming optimization. *IEEE Transactions on Power Systems*, 19(4), pp.2077-2085.

[82] OUYANG, Z. and SHAHIDEHPOUR, S.M., 1991. An intelligent dynamic programming for unit commitment application. *IEEE Transactions on Power Systems*, 6(3), pp.1203-1209.

[83] HOBBS, W.J., HERMON, G., WARNER, S. and SHELBE, G.B., 1988. An enhanced dynamic programming approach for unit commitment. *IEEE Transactions on Power systems*, 3(3), pp.1201-1205.

[84] JONES, K.O., 2005, June. Comparison of genetic algorithm and particle swarm optimization. In *Proceedings of the International Conference on Computer Systems and Technologies,* pp. 1-6

[85] GOLDBERG. D.R., 1989. *Genetic Algorithm in Search, Optimization and Machine Learning,* Norwell, MA: Kluwer.

[86] GOLDBERG, D.E. and HOLLAND, J.H., 1988. Genetic algorithms and machine learning. *Machine learning*, 3(2), pp.95-99.

[87] EBERHART, R. and KENNEDY, J., 1995, October. A new optimizer using particle swarm theory. In *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science,* pp. 39-43.

[88] YANG, S. and WANG, M., 2004, June. A quantum particle swarm optimization. In *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No. 04TH8753),* 1, pp. 320-324.

[89] KENNEDY, J., 2006. Swarm intelligence. In *Handbook of nature-inspired and innovative computing* pp. 187-219. Springer, Boston, MA.

[90] KENNEDY, J., 2010. Particle swarm optimization. *Encyclopedia of machine learning*, pp.760-766.

[91] PARSOPOULOS, K.E. and VRAHATIS, M.N., 2002. Particle swarm optimization method for constrained optimization problems. *Intelligent Technologies–Theory and Application: New Trends in Intelligent Technologies*, 76(1), pp.214-220.

[92] SAHARAN, L.K. and MITTAL, M.L., 2013. Comparison of Various PSO Variants with Different Mathematical Function. *International Journal on Theoretical and Applied Research in Mechanical Engineering,* 2(3), pp. 135-142

[93] LI, C. and YANG, S., 2009, May. An adaptive learning particle swarm optimizer for function optimization. In *2009 IEEE Congress on Evolutionary Computation,* pp. 381-388

[94] JANSON, S. and MIDDENDORF, M., 2005. A hierarchical particle swarm optimizer and its adaptive variant. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 35(6), pp.1272-1282.

[95] IMRAN, M., HASHIM, R. and KHALID, N.E.A., 2013. An overview of particle swarm optimization variants. *Procedia Engineering*, 53, pp.491-496.

[96] CABRERA, J.C.F. and COELLO, C.A.C., 2007, November. Handling constraints in particle swarm optimization using a small population size. In *Mexican International Conference on Artificial Intelligence,* pp. 41-51

[97] ABRAHAM, A. and LIU, H., 2009. Turbulent particle swarm optimization using fuzzy parameter tuning. In *Foundations of Computational Intelligence,*3, pp.291-312

[98] PULIDO, G.T. and COELLO, C.A.C., 2004, June. A constraint-handling mechanism for particle swarm optimization. In *Proceedings of the 2004 Congress on Evolutionary Computation,* 2, pp.1396-1403

[99] SOUSA-FERREIRA, I. and SOUSA, D., 2017. A review of velocity-type PSO variants. *Journal of Algorithms & Computational Technology,* 11(1), pp. 23-30

[100] XIANG, T., WANG, J. and LIAO, X., 2007, September. An improved particle swarm optimizer with momentum. In *2007 IEEE Congress on Evolutionary Computation* pp. 3341-3345

[101] DEL VALLE, Y., VENAYAGAMOORTHY, G.K., MOHAGHEGHI, S., HERNANDEZ, J.C. and HARLEY, R.G., 2008. Particle swarm optimization: basic concepts, variants and applications in power systems. *IEEE Transactions on evolutionary computation*, 12(2), pp.171-195.

[102] THANGARAJ, R., PANT, M., ABRAHAM, A. and SNASEL, V., 2012. Modified particle swarm optimization with time varying velocity vector. *International Journal of Innovative Computing, Information and Control*, 8(1), pp.201-218.

[103] FUKUYAMA, Y., 2008. Fundamentals of particle swarm optimization techniques. *Modern heuristic optimization techniques: theory and applications to power systems*, pp.71-87.

[104] POLI, R., KENNEDY, J. and BLACKWELL, T., 2007. Particle swarm optimization. *Swarm intelligence*, 1(1), pp.33-57.

[105] HASHIM, R., IMRAN, M. and KHALID, N.E.A., 2013. Particle swarm optimization (PSO) variants with Triangular Mutation. *Journal of Engineering and Technology (JET)*, 4(1), pp.95-108.

[106] KUMAR, A., SINGH, B.K. and PATRO, B.D.K., 2016. Particle Swarm Optimization: A Study of Variants and Their Applications. *International Journal of Computer Applications*, 135(5), pp.24-30.

[107] DORIGO, M., MANIEZZO, V. and COLORNI, A., 1996. Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *26*(1), pp.29-41.

[108] DORIGO, M. and DI CARO, G., 1999, July. Ant colony optimization: a new meta-heuristic. In *Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406)* (Vol. 2, pp. 1470-1477). IEEE.

[109] RAO, S.S., 1991. *Optimization theory and applications*. New Delhi: Wiley Eastern Limited.

[110] ISHIBUCHI, H., NOJIMA, Y. and Doi, T., 2006, July. Comparison between single-objective and multi-objective genetic algorithms: Performance comparison and performance measures. *2006 IEEE International Conference on Evolutionary Computation,* pp. 1143-1150.

[111] CHANKONG, V. and HAIMES, Y.Y., 2008. *Multiobjective decision making: theory and methodology*. Courier Dover Publications.

[112] SRINIVAS, N. and DEB, K., 1994. Muiltiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary computation*, 2(3), pp.221-248.

[113] DEB, K., 2001. *Multi-objective optimization using evolutionary algorithms,* 16, John Wiley & Sons.

[114] GOLDBERG, D.E. and RICHARDSON, J., 1987, Genetic algorithms with sharing for multimodal function optimization. *Genetic algorithms and their applications: Proceedings of the Second International Conference on Genetic Algorithms* pp. 41-49.

[115] ZHANG, G. and ZUO, H., 2013. Pareto optimal solution analysis of convex multi-objective programming problem. *Journal of Networks*, 8(2), p.437-444

[116] GORISSEN, B.L. and Den HERTOG, D., 2012. Approximating the Pareto set of multiobjective linear programs via robust optimization. *Operations Research Letters*, *40*(5), pp.319-324.

[117] FIELDSEND, J.E. and SINGH, S., 2002. A multi-objective algorithm based upon particle swarm optimisation, an efficient data structure and turbulence.

[118] YANG, Z., WU, A. and MIN, H., 2014, August. A multi-objective discrete pso algorithm based on enhanced search. In *2014 Sixth International Conference on Intelligent Human-Machine Systems and Cybernetics,* 2, pp. 198-201

[119] TIAN, Y., CHENG, R., ZHANG, X. and JIN, Y., 2017. PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [educational forum]. *IEEE Computational Intelligence Magazine*, 12(4), pp.73-87.

[120] ISHIBUCHI, H., TSUKAMOTO, N. and NOJIMA, Y., 2008, June. Evolutionary many-objective optimization: A short review. In *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)* pp. 2419-2426.

[121] LI, M., ZHEN, L. and YAO, X., 2017. How to read many-objective solution sets in parallel coordinates [educational forum]. *IEEE Computational Intelligence Magazine*, 12(4), pp.88-100.

[122] ISHIBUCHI, H., MASUDA, H. and NOJIMA, Y., 2015. Pareto fronts of many-objective degenerate test problems. *IEEE Transactions on Evolutionary Computation*, 20(5), pp.807-813.

[123] ZITZLER, E., 1999. *Evolutionary algorithms for multiobjective optimization: Methods and applications,* 63, Ithaca: Shaker.

[124] KNOWLES, J. and CORNE, D., 1999, July. The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation. In *Congress on Evolutionary Computation (CEC99),* 1, pp. 98-105.

[125] HANS, A.E., 1988. Multicriteria optimization for highly accurate systems. *Multicriteria Optimization in Engineering and Sciences*, 19, pp.309-352.

[126] SCHAFFER, J.D., 1986. *SOME EXPERIMENTS IN MACHINE LEARNING USING VECTOR EVALUATED GENETIC ALGORITHMS* (Doctoral dissertation, Vanderbilt University)

[127] TAMURA, K. and MIURA, S., 1979. Necessary and sufficient conditions for local and global nondominated solutions in decision problems with multi-objectives. *Journal of Optimization Theory and Applications*, 28(4), pp.501-523.

[128] VAN VELDHUIZEN, D.A., 1999. *Multiobjective evolutionary algorithms: classifications, analyses, and new innovations* (No. AFIT/DS/ENG/99-01). AIR FORCE INST OF TECH WRIGHT-PATTERSONAFB OH SCHOOL OF ENGINEERING.

[129] YILMAZ, Ö.F. and DURMUSOGLU, M.B., 2018. Evolutionary Algorithms for Multi-Objective Scheduling in a Hybrid Manufacturing System. *Handbook of Research on Applied Optimization Methodologies in Manufacturing Systems,* pp. 162-187. IGI Global.

[130] FONSECA, C.M. and FLEMING, P.J., 1993, June. Genetic Algorithms for Multiobjective Optimization: Formulation Discussion and Generalization. *Icga,* 93, pp. 416-423.

[131] KONAK, A., COIT, D.W. and SMITH, A.E., 2006. Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering & System Safety*, 91(9), pp.992-1007.

[132] FONSECA, C.M. and FLEMING, P.J., 1998. Multiobjective optimization and multiple constraint handling with evolutionary algorithms. I. A unified formulation. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 28(1), pp.26-37.

[133] FONSECA, C.M. and FLEMING, P.J., 1998. Multiobjective optimization and multiple constraint handling with evolutionary algorithms. II. Application

example. *IEEE Transactions on systems, man, and cybernetics-Part A: Systems and humans*, 28(1), pp.38-47.

[134] ZITZLER, E. and THIELE, L., 1998, September. Multiobjective optimization using evolutionary algorithms—a comparative case study. *International conference on parallel problem solving from nature,* pp. 292-301

[135] TANAKA, M., WATANABE, H., FURUKAWA, Y. and TANINO, T., 1995, October. GA-based decision support system for multicriteria optimization. In *IEEE International Conference on Systems Man and Cybernetics,* 2, pp. 1556-1561.

[136] BLICKLE, T. and THIELE, L., 1996. A comparison of selection schemes used in evolutionary algorithms. *Evolutionary Computation*, 4(4), pp.361-394.

[137] MILLER, B.L. and SHAW, M.J., 1996, May. Genetic algorithms with dynamic niche sharing for multimodal function optimization. In *Proceedings of IEEE international conference on evolutionary computation,* pp. 786-791

[138] DEB, K., and GOLDBERG, D.E., 1991 An Investigation of niches and species formation in genetic function optimization. *Proceedings of the Third International Conference on Genetic Algorithms,* pp. 42-50

[139] GOLDBERG, D.E. and RICHARDSON, J., 1987, July. Genetic algorithms with sharing for multimodal function optimization. *Genetic algorithms and their applications: Proceedings of the Second International Conference on Genetic Algorithms,* pp. 41-49.

[140] HORN, J., NAFPLIOTIS, N. and GOLDBERG, D.E., 1994, June. A niched Pareto genetic algorithm for multiobjective optimization. *Proceedings of the first IEEE conference on evolutionary computation, IEEE world congress on computational intelligence,* 1, pp. 82-87

[141] RUDOLPH, G., 1999. Evolutionary search under partially ordered sets. *Dept. Comput. Sci./LS11, Univ. Dortmund, Dortmund, Germany, Tech. Rep. CI-67/99*.

[142] ZITZLER, E., DEB, K. and THIELE, L., 2000. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary computation*, 8(2), pp.173-195.

[143] DEB, K., PRATAP, A., AGARWAL, S. and MEYARIVAN, T.A.M.T., 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, 6(2), pp.182-197.

[144] SABRA, A., FUNG, W.K. and CHURN, P., 2019. Multi-objective Optimization of Confidence-Based Localization in Large-Scale Underwater Robotic Swarms. In *Distributed Autonomous Robotic Systems,* pp.109-123.

[145] JENSEN, M.T., 2003. Reducing the run-time complexity of multiobjective EAs: The NSGA-II and other algorithms. *IEEE Transactions on Evolutionary Computation*, 7(5), pp.503-515.

[146] BEKELE, E.G. and NICKLOW, J.W., 2007. Multi-objective automatic calibration of SWAT using NSGA-II. *Journal of Hydrology*, 341(3-4), pp.165-176.

[147] LI, H. and ZHANG, Q., 2008. Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. *IEEE transactions on evolutionary computation*, 13(2), pp.284-302.

[148] ZITZLER, E. and THIELE, L., 1998. An evolutionary algorithm for multiobjective optimization: The strength pareto approach. *TIK-report*, *43*.

[149] JIANG, S. and YANG, S., 2017. A strength Pareto evolutionary algorithm based on reference direction for multiobjective and many-objective optimization. *IEEE Transactions on Evolutionary Computation*, *21*(3), pp.329-346.

[150] GARUD, S.S., KARIMI, I. A. and KRAFT, M., 2017. Design of computer experiments: A review. *Computers & Chemical Engineering,* 106**,** 71-95.

[151] GIESL, P. and HAFSTEIN, S., 2015. Review on computational methods for Lyapunov functions. *Discrete and Continuous Dynamical Systems-Series B*, 20(8), pp.2291-2331.

[152] PANIKHOM, S. and SUJITIORN, S., 2010, July. Numerical approach to Lyapunov's stability analysis of nonlinear systems using threshold accepting algorithms. In *The 25th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC),* pp. 811-814.

[153] JI, Z., WU, W., FENG, Y. and ZHANG, G., 2013. Constructing the Lyapunov function through solving positive dimensional polynomial system. *Journal of Applied Mathematics*, *2013*.

Appendix A

Appendix A shows the Matlab code of how to apply the Randomised Triangular Subdivision on Duffing oscillator

```
% random select 50 initial conditions in the selected region
for i = 1:1:50000
nPop = 50;
Vx1 = -10 * rand(nPop,1) + 5;
Vx2 = -10 * rand(nPop,1) + 5;
vi = convhull(Vx1,Vx2);
A(i) = polyarea(Vx1(vi),Vx2(vi));
if A(i)>95
save('Myfile','Vx1','Vx2');
end
end
```

```
% Delaunay Triangulate the selected region using the selected initial
conditions
load Myfile
vi = convhull(Vx1,Vx2);
polyarea(Vx1(vi),Vx2(vi))
Vx = [Vx1,Vx2];
dt = delaunayTriangulation(Vx1,Vx2);
```

```
% the mathematical model of the Duffing oscillator
function x=osmodel(omega,F,k,xpre,i,Ts);

x_pos = xpre(1);
x_vel = xpre(2);
x_acc = xpre(3);

temp = F*cos(omega*i*Ts) - k*x_vel + x_pos - x_pos^3;

x(3) = temp;
x(2) = x_vel + temp*Ts;
x(1) = x_pos + x_vel*Ts;
```

```
% check whether all three vertices of the triangle are in the same
state or not
% number of iterations
np = 60;

% number of steps in each iteration
n = 5000;
iter = np*n;

% parameters of the system
k = 0.9;
F = 1.9;
omega = 1.2;
Ts = 2*pi/(omega*n);

% states of the system
r1 = [-0.2868;1.6029];
r2 = [0.6964;2.8861];

% number of the triangles
```

```
nPop = 91;

x_pos = zeros(1,iter+1);
x_vel = zeros(1,iter+1);
x_acc = zeros(1,iter+1);

sa = 0;
di = 0;
SAx = [];
SAy = [];
DIx = [];
DIy = [];
Q = [];

% vertices of each triangle
xx = [dt.Points(dt.ConnectivityList(:,1),1),
dt.Points(dt.ConnectivityList(:,1),2)];
yy = [dt.Points(dt.ConnectivityList(:,2),1),
dt.Points(dt.ConnectivityList(:,2),2)];
zz = [dt.Points(dt.ConnectivityList(:,3),1),
dt.Points(dt.ConnectivityList(:,3),2)];

%check the state of each vertex of each triangle
for ii = 1:1:nPop
    omega = 1.2;
    Vx = [xx(ii,:);yy(ii,:);zz(ii,:)];
for it = 1:1:3
    x_pos(1) = Vx(it,1);
    x_vel(1) = Vx(it,2);
    X0 = [Vx(it,1);Vx(it,2)];
for i = 1:iter
    x = [x_pos(i),x_vel(i),x_acc(i)];

    x_later = osmodel(omega,F,k,x,i,Ts);
    x_pos(i+1)=x_later(1);
    x_vel(i+1)=x_later(2);
    x_acc(i+1)=x_later(3);
end
    X = [x_pos(end);x_vel(end)];
    if norm(X-r1)<1e-2
       Q(ii,it) = 1;
    else
       Q(ii,it) = 2;
    end
end

% check whether all three vertices of the triangle are in the same
state or not
    if Q(ii,1)==Q(ii,2) && Q(ii,2)==Q(ii,3) && Q(ii,3)==Q(ii,1)
        sa = sa+1;
        SAx(sa,:) = (Vx(:,1)');
        SAy(sa,:) = (Vx(:,2)');
        A1(sa) = A(ii);
    else
        di = di+1;
        DIx(di,:) = (Vx(:,1)');
        DIy(di,:) = (Vx(:,2)');
        A2(di) = A(ii);
    end
end
```

```
% Calculate the area of each triangle
% xx, yy, zz are the vertices of each triangle
xx = [dt.Points(dt.ConnectivityList(:,1),1),
dt.Points(dt.ConnectivityList(:,1),2)];
yy = [dt.Points(dt.ConnectivityList(:,2),1),
dt.Points(dt.ConnectivityList(:,2),2)];
zz = [dt.Points(dt.ConnectivityList(:,3),1),
dt.Points(dt.ConnectivityList(:,3),2)];
A = [];
for i = 1:1:91
   A(i,1) = 0.5*abs(x(i,1)*(y(i,2)-z(i,2))+y(i,1)* (z(i,2)-
x(i,2))+z(i,1)*(x(i,2)-y(i,2)));
end
```

```
% calculate the centroid of each triangle
% x coordinates of the centroid of triangles
ctx = (dt.Points(dt.ConnectivityList(:,1),1) +
dt.Points(dt.ConnectivityList(:,2),1) +
dt.Points(dt.ConnectivityList(:,3),1))/3;

% y coordinates of the centroid of triangles
cty = (dt.Points(dt.ConnectivityList(:,1),2) +
dt.Points(dt.ConnectivityList(:,2),2) +
dt.Points(dt.ConnectivityList(:,3),2))/3;

% centroid of triangles
ct = [ctx,cty];
```