SMOOTHING AND DIFFERENTIATION OF DYNAMIC DATA

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

FATİH TİTREK

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

MAY 2010

Approval of the thesis:

SMOOTHING AND DIFFERENTIATION OF DYNAMIC DATA

submitted by **FATİH TİTREK** in partial fullfillment of the requirements for the degree of **Master of Science in Computer Engineering, Middle East Technical University** by,

Prof. Dr. Canan Özgen
Dean, Graduate School of **Natural and Applied Sciences** _____________________

Prof. Dr. Adnan Yazıcı
Head of Department, **Computer Engineering** _____________________

Prof. Dr. Sibel Tari
Supervisor, **Computer Engineering Dept., METU** _____________________

Examining Committee Members:

Prof. Dr. Müslim Bozyiğit
Computer Engineering Dept., METU _____________________

Prof. Dr. Sibel Tari
Computer Engineering Dept., METU _____________________

Prof. Dr. Volkan Atalay
Computer Engineering Dept., METU _____________________

Assoc. Prof. Dr. M. S. Halit Oğuztüzün
Computer Engineering Dept., METU _____________________

Prof. Dr. Kemal Leblebicioğlu
Electrical and Electronics Engineering Dept., METU _____________________

Date: _____05.05.2010______

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: Fatih Titrek

Signature :

ABSTRACT

SMOOTHING AND DIFFERENTIATION OF DYNAMIC DATA

Titrek, Fatih

M.S., Department of Computer Engineering
Supervisor: Prof. Dr. Sibel Tari

May 2010, 113 pages

Smoothing is an important part of the pre-processing step in Signal Processing. A signal, which is purified from noise as much as possible, is necessary to achieve our aim. There are many smoothing algorithms which give good result on a stationary data, but these smoothing algorithms don't give expected result in a non-stationary data. Studying Acceleration data is an effective method to see whether the smoothing is successful or not. The small part of the noise that takes place in the Displacement data will affect our Acceleration data, which are obtained by taking the second derivative of the Displacement data, severely. In this thesis, some linear and non-linear smoothing algorithms will be analyzed in a non-stationary dataset.

Keywords: One dimensional filters, Linear smoothing filters, Non-linear smoothing filters, Diffusion, Non-stationary signals.

ÖZ

DİNAMİK VERİLERİN DÜZGÜNLEŞTİRİLMESİ VE AYRIMLAŞTIRILMASI

Titrek, Fatih

Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Sibel Tari

Mayıs 2010, 113 sayfa

Düzgünleştirme işlemi, sinyal işlemede önemli bir ön işlemdir. Gürültüden temizlenebildiği kadar temizlenmiş bir sinyal, amacımıza ulaşmamız için gereklidir. Durağan sinyaller üzerinde iyi sonuçlar veren birçok düzgünleştirme algoritması vardır. Fakat bu algoritmalar durağan olmayan sinyaller üzerinde beklenilen sonuçları vermezler. İvme verisinin incelenmesi, düzgünleştirme işleminin başarılı olarak yapılıp yapılmadığını görmede etkili bir yöntemdir. Çünkü yer değiştirme verisinin içinde yer alan ufak gürültüler, yer değiştirme verisinin ikinci türevini alarak elde edilen ivme verisini büyük ölçüde etkileyecektir. Bu tezde, bazı doğrusal ve doğrusal olmayan düzgünleştirme algoritmaları durağan olmayan bir veri kümesi üzerinde incelenecektir.

Anahtar Kelimeler: Tek boyutlu filtreleme, Doğrusal düzgünleştirme filtreleri, Doğrusal olmayan düzgünleştirme filtreleri, Difüzyon, Durağan olmayan sinyaller.

To my parents, my sister and to my brother.

For being with me all the time.

ACKNOWLEDGEMENTS

I would like to thank Prof. Dr. Sibel TARI for her supervision, guidance, invaluable suggestions and especially motivating me throughout the development of the thesis. Her patience, vision and friendly approach is the reason to vitalize this work.

It is my pleasure to express my deepest appreciation to members of Image Processing and Pattern Recognition Laboratory at Department of Computer Engineering. I always felt their helpfulness during all my study.

I would like to express my gratitude to my roommates and all staff at the Department of Computer Engineering.

It is my pleasure to express my appreciation to her for sharing hard times of my work.

I would also acknowledge to all those who contribute the completion of this work in every respect.

Finally, I am grateful to my parents, sister and brother for their encouragement to finish my study, complimentary love and support…

TABLE OF CONTENTS

ABSTRACT iv

ÖZ v

ACKNOWLEDGEMENTS vii

TABLE OF CONTENTS viii

LIST OF TABLES xi

LIST OF FIGURES xiii

CHAPTERS

1. INTRODUCTION 1

1.1 EXPERIMENTAL DATA SETS 8

1.2 ORGANIZATION 9

2. LINEAR SMOOTHING FILTERS 10

2.1 MEAN FILTER 12

2.2 ITERATED MEAN FILTER 14

2.2.1 Golf Ball Data Results for Mean Filtering and Iterated Mean Filtering 15

2.2.2 Displacement Data Results for Mean Filtering 18

2.2.3 Acceleration Data Results for Mean Filtering 21

2.2.4 Comparing Mean and Iterated Mean Filters 22

2.3 GAUSSIAN FILTER 24

2.3.1 Displacement Data Results for Gaussian Filtering 25

2.3.2 Second Derivative Results for Gaussian Filtering 26

2.3.3 Acceleration Data Results for Gaussian Filtering 27

2.3.4 Comparing Gaussian Filters of Varying Standart Deviation 29

2.4 IDEAL LOW PASS FILTER 30

2.4.1 Golf Ball Data Results for Ideal Low Pass Filtering 31

2.4.2 Displacement Data Results for Ideal Low Pass Filtering 34

2.4.3 Acceleration Data Results for Ideal Low Pass Filtering 35
2.4.4 Second Derivative Results for Ideal Low Pass Filtering....38
2.4.5 Comparing Ideal Low Pass Filters of Varying Cut off Frequency 39

2.5 BUTTERWORTH LOW PASS FILTER....40
2.5.1 Displacement Data Results for Butterworth Low Pass Filtering 41
2.5.2 Acceleration Data Results for Butterworth Low Pass Filtering....43
2.5.3 Second Derivative Results for Butterworth Low Pass Filtering 46
2.5.4 Comparing Butterworth Low Pass Filters of Varying Cut off and Order Parameters....47

2.6 LINEAR DIFFUSION FILTER....48
2.6.1 Acceleration Data Results for Linear Diffusion Filtering 49
2.6.2 Displacement Data Results for Linear Diffusion Filtering 52
2.6.3 Second Derivative Results for Linear Diffusion Filtering 57
2.6.4 Comparing Linear Diffusion Filters of Varying Parameters....57

2.7 GENERAL REVIEW OF LINEAR SMOOTHING FILTERS....59

3. NON-LINEAR SMOOTHING FILTERS....62

3.1 KUWAHARA FILTER....64
3.1.1 Displacement Data Results for Kuwahara Filtering....64
3.1.2 Acceleration Data Results for Kuwahara Filtering....66
3.1.3 Second Derivative Results for Kuwahara Filtering....68
3.1.4 Comparing Kuwahara Filters of Varying Mask Size and Iteration 69

3.2 SIGMA FILTER....70
3.2.1 Displacement Data Results for Sigma Filtering 71
3.2.2 Acceleration Data Results for Sigma Filtering....73
3.2.3 Second Derivative Results for Sigma Filtering 75
3.2.4 Comparing Sigma Filters of Varying Sigma and Mask Size Parameters...75

3.3 MEDIAN FILTER....77
3.3.1 Golf Ball Data Results for Median Filtering 78
3.3.2 Displacement Data Results for Median Filtering 79
3.3.3 Acceleration Data Results for Median Filtering....80
3.3.4 Second Derivative Results for Median Filtering 82
3.3.5 Comparing Median Filters of Varying Filter Size and Iteration Parameters....83

3.4 PERONA MALIK FILTER 84

3.4.1 Displacement Data Results for Perona Malik Filtering.....85

3.4.2 Acceleration Data Results for Perona Malik Filtering.....86

3.4.3 Second Derivative Results for Perona Malik Filtering.....89

3.4.4 Comparing Perona Malik Filters of Varying Parameters.....90

3.5 THE AMBROSIO-TORTORELLI APPROXIMATION OF THE MUMFORD-SHAH MODEL 91

3.5.1 Acceleration Data Results for Ambrosio Tortorelli Approximation of the Mumford Shah.....92

3.5.2 Displacement Data Results for Ambrosio Tortorelli Approximation of the Mumford Shah.....96

3.5.3 Second Derivative Results for Ambrosio Tortorelli Approximation of the Mumford Shah.....97

3.5.4 Comparing Ambrosio Tortorelli Approximation of Mumford Shah of Varying Parameters 97

3.6 GENERAL REVIEW OF THE NON-LINEAR SMOOTHING FILTERS.....99

3.7 BAKLAVA FILTER 103

3.7.1 Golf Ball Drop Data Results for Baklava Filtering 104

3.7.2 Displacement Data Results for Baklava Filtering 105

3.7.3 Acceleration Data Results for Baklava Filtering.....107

3.7.4 Second Derivative Results for Baklava Filtering: 108

4. SUMMARY AND CONCLUSION 109

REFERENCES.....112

LIST OF TABLES

TABLES

Table 2.2.1.1 Original Golf Ball Data Values....15

Table 2.2.1.2 Smoothed Golf Data after applying Mean filter by using mask 1....15

Table 2.2.1.3 Smoothed Golf Data after applying Mean filter by using mask 1 for 20 iterations....15

Table 2.2.1.4 Smoothed Golf Data after applying Mean filter by using mask 1 for 30 iterations....15

Table 2.2.1.5 Smoothed Golf Data after applying Mean filter by using mask 2....16

Table 2.2.1.6 Smoothed Golf Data after applying Mean filter by using mask 2 for 20 iterations....16

Table 2.2.1.7 Smoothed Golf Data after applying Mean filter by using mask 2 for 30 iterations....16

Table 2.4.1 Ideal Low Pass Filter....30

Table 2.4.1.1 Smoothed Golf Data after applying ILP filter by using Cut off Frequency=10....32

Table 2.4.1.2 Smoothed Golf Data after applying ILP filter by using Cut off Frequency=24....32

Table 2.4.1.3 Smoothed Golf Data after applying ILP filter by using Cut off Frequency=25....33

Table 2.4.1.4 Smoothed Golf Data after applying ILP filter by using Cut off Frequency=40....33

Table 2.4.2.1 Original Displacement Data Values....34

Table 2.4.2.2 Smoothed Displacement Data after applying ILP filter by using Cut off Freq.=10....35

Table 2.4.2.3 Smoothed Displacement Data after applying ILP filter by using Cut off Freq.=40....35

Table 2.4.2.4 Smoothed Displacement Data after applying ILP filter by using Cut off Freq.=100....35

Table 2.4.3.1 Original Acceleration Data Values....35

Table 2.4.3.2 Smoothed Acceleration Data after applying ILP filter by using Cut off Freq.=10 37

Table 2.4.3.3 Smoothed Acceleration Data after applying ILP filter by using Cut off Freq.=40 37

Table 2.4.3.4 Smoothed Acceleration Data after applying ILP filter by using Cut off Freq.=100 37

Table 2.5.1.1 Smoothed Displacement Data after applying Butterworth Low Pass filter by using Cut off Frequency=10 and Order=3 43

Table 2.5.1.2 Smoothed Displacement Data after applying Butterworth Low Pass filter by using Cut off Frequency=100 and Order=3 43

Table 2.5.2.1 Smoothed Acceleration Data after applying Butterworth Low Pass filter by using Cut off Frequency=10 and Order=3 45

Table 2.5.2.2 Smoothed Acceleration Data after applying Butterworth Low Pass filter by using Cut off Frequency=40 and Order=3 45

Table 2.5.2.3 Smoothed Acceleration Data after applying Butterworth Low Pass filter by using Cut off Frequency=100 and Order=3 45

Table 3.1.1.1 Kuwahara Filter Results by using Mask Size=5 65

Table 3.1.1.2 Kuwahara Filter Results by using Mask Size=5 for 3 iterations 65

Table 3.1.1.3 Kuwahara Filter Results by using Mask Size=9 66

Table 3.1.1.4 Kuwahara Filter Results by using Mask Size=9 for 3 iterations 66

Table 3.1.2.1 Kuwahara Filter Results by using Mask Size=5 67

Table 3.1.2.2 Kuwahara Filter Results by using Mask Size=5 for 3 iterations 67

Table 3.1.2.3 Kuwahara Filter Results by using Mask Size=9 67

Table 3.1.2.4 Kuwahara Filter Results by using Mask Size=9 for 3 iterations 67

LIST OF FIGURES

FIGURES

Figure 1.1 Original Data 3

Figure 1.2 First Derivative Result Of Figure 1.1 3

Figure 1.3 Second Derivative Result Of Figure 1.1 4

Figure 1.4 Noisy Data 4

Figure 1.5 First Derivative Result after Smoothing in Figure 1.4 5

Figure 1.6 Second Derivative Result after Smoothing less in Figure 1.4 6

Figure 1.7 Second Derivative Result after Smoothing in Figure 1.4 6

Figure 1.1.1 Original Golf Ball Drop Data 8

Figure 1.1.2.a Original Angular Displacement Data 9

Figure 1.1.2.b Original Acceleration Data 9

Figure 2.1.1 Measurements 12

Figure 2.2.1 (Mean Filter vs. Iterated Mean Filter). (a) Golf Ball Data Result using the mask 1, (b) The samples, between 16 and 22, of Figure 2.2.1.a. 15

Figure 2.2.2 (Mean Filter vs. Iterated Mean Filter). (a) Golf Ball Data Result using the mask 2, (b) The samples, between 16 and 22, of Figure 2.2.2.a. 16

Figure 2.2.3 (Mean Filter vs. Iterated Mean Filter). (a) Second Derivative Result of Smoothed Golf Ball Data using the mask 1, (b) The samples, between 10 and 22, of Figure 2.2.3.a 17

Figure 2.2.4 (Mean Filter vs. Iterated Mean Filter). (a) Second Derivative Result of Smoothed Golf Ball Data using the mask 2, (b) The samples, between 10 and 22, of Figure 2.2.4.a 17

Figure 2.2.5 (Iterated Mean Filter). (a) Second Derivative Result of Smoothed Displacement Data using the mask 1 (1x3), (b) Second Derivative Result of Smoothed Displacement Data using the mask 2 (1x5) 18

Figure 2.2.6 (Mean Filter vs. Iterated Mean Filter). (a) Displacement Data Result using the mask 1, (b) The samples, between 333 and 364, of Figure 2.2.6.a 18

Figure 2.2.7 (Mean Filter vs. Iterated Mean Filter). (a) Displacement Data Result using the mask 2, (b) The samples, between 333 and 364, of Figure 2.2.7.a 19

Figure 2.2.8 (Mean Filter vs. Iterated Mean Filter). (a) Second Derivative Result of Smoothed Displacement Data using the mask 1, (b) The samples, between 333 and 364, of Figure 2.2.8.a 19

Figure 2.2.9 (Mean Filter vs. Iterated Mean Filter). (a) Second Derivative Result of Smoothed Displacement Data using the mask 2, (b) The samples, between 333 and 364, of Figure 2.2.9.a 20

Figure 2.2.10 (Iterated Mean Filter). (a) Second Derivative Result of Smoothed Displacement Data using the mask 1 (1x3), (b) Second Derivative Result of Smoothed Displacement Data using the mask 2 (1x5)............20

Figure 2.2.11 (Mean Filter vs. Iterated Mean Filter). (a) Acceleration Data Result using the mask 1, (b) The samples, between 207 and 242, of Figure 2.2.11.a 21

Figure 2.2.12 (Mean Filter vs. Iterated Mean Filter). (a) Acceleration Data Result using the mask 2, (b) The samples, between 207 and 242, of Figure 2.2.12.a 21

Figure 2.2.13 (Mean Filtering). The Comparative Second Derivative Results of Smoothed Displacement Data and Original Acceleration Data 22

Figure 2.2.14 (Iterated Mean Filtering). The Comparative Second Derivative Results of Smoothed Displacement Data and Original Acceleration Data 23

Figure 2.2.15 (Iterated Mean Filtering). The Comparative Second Derivative Results of Smoothed Displacement Data and Original Acceleration Data............23

Figure 2.3.1 Gaussian distribution where σ =3, Size=6*σ+1............24

Figure 2.3.2 (Gaussian Filtering). (a) Displacement Data Results for σ=1, 2, 4, 6, size=6*σ+1, (b) The samples, between 333 and 364, of Figure 2.3.2.a...........25

Figure 2.3.3 (Gaussian Filtering). (a) Displacement Data Results between 180 and 370 for σ=1, 6, size=6*σ+1, (b) Displacement Data Results between 460 and 600 25

Figure 2.3.4 (Gaussian Filtering). (a) Second Derivative Result of Smoothed Displacement Data for σ=1, 2, 4, 6, size=6*σ+1, (b) The samples, between 333 and 364, of Figure 2.3.4.a 26

Figure 2.3.5 (Gaussian Filtering). (a) Second Derivative Result of Smoothed Displacement Data for σ=4, 6, size=6*σ+1, (b) Second Derivative Result of Smoothed Displacement Data for σ=6, 10 size=6*σ+1............26

Figure 2.3.6 (Gaussian Filtering). Kernel Size is calculated by using (2.14). (a) σ=1, (b) σ=2, (c) σ=4, (d) σ=6............27

Figure 2.3.7 (Gaussian Filtering). Results between 186 and 215 for σ=1, 2, 3, 4, size=6*σ+1............28

Figure 2.3.8 (Gaussian Filtering). (a) Acceleration Data Results for σ=1, 2, 4, 6, size=6*σ+1, (b) The samples, between 185 and 230, of Figure 2.3.8.a...........28

Figure 2.3.9 (Gaussian Filtering). The Comparative Second Derivative Results of Smoothed Displacement Data and Original Acceleration Data 29

Figure 2.4.1 Ideal Low Pass Filter Cross Section 30

Figure 2.4.2 (Ideal Low Pass Filtering). (a) Smoothed Golf Data by using the Cut off Frequency=10, (b) Smoothed Golf Data by using the Cut off Freq.=24......... 31

Figure 2.4.3 (Ideal Low Pass Filtering). (a) Smoothed Golf Data by using the Cut off Frequency=25, (b) Smoothed Golf Data by using the Cut off Freq.=40......... 33

Figure 2.4.4 (Ideal Low Pass Filtering). (a) Smoothed Displacement Data by using the Cut off Frequency=10, (b) Smoothed Displacement Data by using the Cut off Frequency=40, (c) Smoothed Displacement Data by using the Cut off Frequency=100...34

Figure 2.4.5 (Ideal Low Pass Filtering). (a) Smoothed Acceleration Data by using the Cut off Frequency=10, (b) Smoothed Acceleration Data by using the Cut off Frequency=40...36

Figure 2.4.6 (Ideal Low Pass Filtering). Smoothed Acceleration Data by using the Cut off Frequency=100..37

Figure 2.4.7 (Ideal Low Pass Filtering). Second Derivative Result of Smoothed Golf Ball Data using Cut off Frequency=10, 24 and 25 38

Figure 2.4.8 (Ideal Low Pass Filtering). (a) Second Derivative Result of Smoothed Displacement Data using Cut off Frequency=10, 24 and 25, (b) Same as Figure 2.4.8.a except Original Displacement Data (Black Line).....................38

Figure 2.4.9 (Ideal Low Pass Filtering). The Comparative Second Derivative Results of Smoothed Displacement Data and Original Acceleration Data.......................39

Figure 2.5.1 Butterworth Low Pass Filter Cross Section 41

Figure 2.5.2 (Butterworth Low Pass Filtering). (a) Smoothed Displacement Data by using the Cut off Frequency=10 and Order=3, (b) Smoothed Displacement Data by using the Cut off Frequency=40 and Order=3, (c) Smoothed Displacement Data by using the Cut off Frequency=100 and Order=3 42

Figure 2.5.3 (Butterworth Low Pass Filtering). Blue Line is the Original input data. Green Line, Red Line and Black Line are the Smoothed Displacement Data by using the Cut off Frequency is 40 and Orders are 1, 2 and 50. Magenta Line is the Ideal Low Pass Filter Result where Cut off Frequency=40...........43

Figure 2.5.4 (Butterworth Low Pass Filtering). (a) Smoothed Acceleration Data by using the Cut off Frequency=10 and Order=3, (b) Smoothed Acceleration Data by using the Cut off Frequency=40 and Order=3, (c) Smoothed Acceleration Data by using the Cut off Frequency=100 and Order=3 44

Figure 2.5.5 (Butterworth Low Pass Filtering). Zoom in Smoothed Acceleration Data by using Cut off Frequency=40 and Order=1,2 and 10...45

Figure 2.5.6 (Butterworth Low Pass Filtering). (a) Second Derivative Result of Smoothed Displacement Data using Cut off Frequency=10, 40, 100, and Order=3 (b) Same as Figure 2.5.6.a except Original Displacement Data........46

Figure 2.5.7 (Butterworth Low Pass Filtering). (a) Second Derivative Result of Smoothed Displacement Data using Cut off Frequency=40, and Order=1, 2, 50 (b) Same as Figure 2.5.7.a except Original Displacement Data (Black Line)........46

Figure 2.5.8 (Butterworth Low Pass Filtering). The Comparative Second Derivative Results of Smoothed Displacement Data and Original Acceleration Data......47

Figure 2.6.1 (Linear Diffusion Filtering). T=1 & Δt=0.01, 0.1, 0.2, 0.5........49

Figure 2.6.2 (Linear Diffusion Filtering). T=5 & Δt=0.01, 0.1, 0.2, 0.5........49

Figure 2.6.3 (Linear Diffusion Filtering). T=20 & Δt=0.01, 0.1, 0.2, 0.5........50

Figure 2.6.4 (Linear Diffusion Filtering). Δt=0.1 & T=1, 5, 20, 30........50

Figure 2.6.5 (Linear Diffusion Filtering). T=1, Δt=-0.05........51

Figure 2.6.6 (Linear Diffusion Filtering). T=20, Δt=0.53........51

Figure 2.6.7 (Linear Diffusion Filtering). T=20 & Δt=0.53, 0.55........52

Figure 2.6.8 (Linear Diffusion Filtering)........52

Figure 2.6.9 (Linear Diffusion Filtering) The samples, between 190 and 380, of Figure 2.6.8........53

Figure 2.6.10 (Linear Diffusion Filtering). T=10, Δt=0.1........54

Figure 2.6.11 (Linear Diffusion Filtering). T=10, Δt=0.1 (a) Mean Value, (b) Variance, (c) Total Gradient, (d) Standard Deviation, (e) Entropy for the values T=200 and Δt=0.01........54

Figure 2.6.12 (Linear Diffusion Filtering). T=10, Δt=0.01, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5 (a)Mean Value, (b)Variance, (c) Total Gradient, (d) Standard Deviation, (e) Entropy........56

Figure 2.6.13 (Linear Diffusion Filtering). (a) Second Derivative Result of Smoothed Displacement Data (b) Same as Figure 2.6.13.a except Original Displacement Data (Black Line) and Green Line........57

Figure 2.6.14 (Linear Diffusion Filtering). The Comparative Second Derivative Results of Smoothed Displacement Data and Original Acceleration Data........58

Figure 2.7.1 (Linear Diffusion Filtering vs. Gauss Convolution). T=10, Δt=0.25, (a) Original Acceleration Data, (b) Linear Diffusion for T=8 vs. Gauss Convolution for σ=4, (c) Linear Diffusion for T=18 vs. Gauss Convolution for σ=6, (d) Linear Diffusion for T=32 vs. Gauss Convolution for σ=8........60

Figure 2.7.2 (The Comparative Second Derivative Results of Displacement Data by using All Experimented Linear Smoothing Filters). Black Line: Original Acceleration Data, (a) Ideal Low Pass Filter Result by using the Cut off Frequency=8, (b) Iterated Mean Filter Result by using Filter Size=11 for 3 iterations, (c) Linear Diffusion Filter Result by using the T=20, Δt=0.1, (d) Gaussian Filter Result by using σ=6, (e) Butterworth Low Pass Filter Result by using the Cut off Frequency=24 and Order=2, (f) Comparison of all results from Figure 2.7.2.a to Figure 2.7.2.e..61

Figure 3.1.1 (Kuwahara Filtering). (a) Smoothed Displacement Data Result, (b) The samples, between 200 and 360, of Figure 3.1.1.a...65

Figure 3.1.2 (Kuwahara Filtering). (a) Smoothed Displacement Data Result, (b) The samples, between 200 and 360, of Figure 3.1.2.a...65

Figure 3.1.3 (Kuwahara Filtering). (a) Smoothed Acceleration Data Result, (b) The samples, between 190 and 220, of Figure 3.1.3.a...66

Figure 3.1.4 (Kuwahara Filtering). (a) Smoothed Acceleration Data Result, (b) The samples, between 190 and 220, of Figure 3.1.4.a...67

Figure 3.1.5 (Kuwahara Filtering). (a) Second Derivative Result of Smoothed Displacement Data using Mask Size= 5 and 9 (b) Same as Figure 3.1.5.a except Original Displacement Data (Black Line)...68

Figure 3.1.6 (Kuwahara Filtering). (a) Second Derivative Result of Smoothed Displacement Data using Mask Size= 5 and 9 for 3 iteration (b) Same as Figure 3.1.6.a except Original Displacement Data (Black Line).....................68

Figure 3.1.7 (Kuwahara Filtering). The Comparative Second Derivative Results of Smoothed Displacement Data and Original Acceleration Data 69

Figure 3.1.8 (Kuwahara Filtering). The Comparative Second Derivative Results of Smoothed Displacement Data and Original Acceleration Data 69

Figure 3.2.1 (Gaussian (Normal) Distribution Curve)..70

Figure 3.2.2 (Sigma Filtering), Mask Size is constant. (a) Smoothed Displacement Data by using the values: Black Line: The Original Data, Blue Line: σ=0.015, Mask Size= 5, Red Line: σ=0.005, Mask Size= 5. (b) The samples, between 333 and 364, Of Figure 3.2.2.a. 71

Figure 3.2.3 (Sigma Filtering), Sigma is constant. (a) Smoothed Displacement Data by using the values: Black Line: The Original Data, Blue Line: σ=0.01, Mask Size= 3, Red Line: σ=0.01, Mask Size=7, Green Line: σ=0.01, Mask Size= 15. (b) The samples, between 333 and 364, of Figure 3.2.3.a.. 72

Figure 3.2.4 (Sigma Filtering), Mask Size is constant. (a) Smoothed Acceleration Data by using the values: Black Line: The Original Data, Blue Line: σ=5, Mask Size= 5, Red Line: σ=10, Mask Size= 5, Green Line: σ=15, Mask Size= 5. (b) The samples, between 209 and 218, of Figure 3.2.4.a 73

Figure 3.2.5 (Sigma Filtering), Sigma is constant. (a) Smoothed Acceleration Data by using the values: Black Line: The Original Data, Blue Line: σ=3, Mask Size= 21, Red Line: σ=3, Mask Size= 51, Green Line: σ=3, Mask Size= 101. (b) The samples, between 209 and 218, of Figure 3.2.5.a 74

Figure 3.2.6 (Sigma Filtering). (a) Second Derivative Result of Smoothed Displacement Data using σ= 0.015, 0.01 and Mask Size = 7 (b) Same as Figure 3.2.6.a except Original Displacement Data (Black Line)...75

Figure 3.2.7 (Sigma Filtering). The Comparative Second Derivative Results of Smoothed Displacement Data and Original Acceleration Data 76

Figure 3.3.1 (Median Filtering). (a) Filter Size=5, (b) Filter Size=7....................................78

Figure 3.3.2 (Median Filtering). (a) Filter Size=5, (b) Filter Size=5 & Iteration number=3, (c) Filter Size=7, (d) Filter Size=7 & Iteration number=3............79

Figure 3.3.3 (Median Filtering). (a) Filter Size=5, (b) Filter Size=5 & Iteration number=3, (c) Filter Size=7, (d) Filter Size=7 & Iteration number=3.............80

Figure 3.3.4 (Median Filtering). (a) Smoothed Displacement Data by using Filter Size=21 and Iteration number=7, (b) Smoothed Acceleration Data by using Filter Size=21 and Iteration number=7..81

Figure 3.3.5 (Median Filtering). (a) Second Derivative Result of Smoothed Displacement Data using Filter Size = 5, 7, 21 (b) Same as Figure 3.3.5.a except Original Displacement Data (Black Line)...82

Figure 3.3.6 (Median Filtering). (a) Second Derivative Result of Smoothed Displacement Data using Filter Size = 5, 7, 21 for 3 iteration (b) Same as Figure 3.3.6.a except Original Displacement Data (Black Line)......................82

Figure 3.3.7 (Median Filtering). The Comparative Second Derivative Results of Smoothed Displacement Data and Original Acceleration Data 83

Figure 3.3.8 (Iterative Median Filtering). The Comparative Second Derivative Results of Smoothed Displacement Data and Original Acceleration Data.................. 83

Figure 3.4.1 (Perona Malik Filtering), (a) Smoothed Displacement Data Result, (b) The samples, between 200 and 360, of Figure 3.4.1.a...85

Figure 3.4.2 (Perona Malik Filtering), (a) Smoothed Displacement Data Result, (b) The samples, between 200 and 360, of Figure 3.4.2.a...85

Figure 3.4.3 (Original Acceleration Data between the samples of 176 - 284)......................86

Figure 3.4.4 (Perona Malik Filtering). Results between the samples of 176 and 284 for λ=10, T=10, Δt=0.25 and various σ=1, 3, 7, 10, 17, 35 86

Figure 3.4.5 (Perona Malik Filtering). Results between the samples of 176 and 284 for σ= 3, T=10, Δt=0.25 and various λ=5, 10, 20, 40, 80, 120.........................87

Figure 3.4.6 (Perona Malik Filtering).) Smoothed Acceleration Data by using σ=3, λ=10, T=50 and Δt=0.05..88

Figure 3.4.7 (Perona Malik Filtering). (a) Mean Value vs. Iteration for σ=3, λ=10, T=50 and Δt=0.05 (b) Variance vs. Iteration for σ=3, λ=10, T=50 and Δt=0.05 (c) Total Gradient vs. Iteration for σ=3, λ=10, T=50 and Δt=0.05 (d) Standard Deviation vs. Iteration for σ=3, λ=10, T=50 and Δt=0.05 (e) Entropy vs. Iteration for σ=3, λ=10, T=50 and Δt=0.05 88

Figure 3.4.8 (Perona Malik Filtering). (a) Second Derivative Result of Smoothed Displacement Data (b) Same as Figure 3.4.8.a except Original Displacement Data (Black Line) 89

Figure 3.4.9 (Perona Malik Filtering). The Comparative Second Derivative Results of Smoothed Displacement Data and Original Acceleration Data 90

Figure 3.5.1 (Result for the Ambrosio Tortorelli Approximation of the Mumford Shah) The results for T=100, Δt=0.4, α=8, β=0.1, ρ= 0.05, 0.01, 0.001, 0.0001, 0.00001, 0.000001 93

Figure 3.5.2 (Result for the Ambrosio Tortorelli Approximation of the Mumford Shah) The results for T=100, Δt=0.4, α=2, β=0.025, ρ= 0.05, 0.01, 0.001, 0.0001, 0.00001, 0.000001 93

Figure 3.5.3 (Result for the Ambrosio Tortorelli Approximation of the Mumford Shah) The results for T=100, Δt=0.4, α=32, β=0.4, ρ= 0.05, 0.01, 0.001, 0.0001, 0.00001, 0.000001 94

Figure 3.5.4 (Result for the Ambrosio Tortorelli Approximation of the Mumford Shah) The results for T=100, Δt=0.4, α=8, 16, 32, 64, β=0.025, ρ= 0.000001 94

Figure 3.5.5 (Result for the Ambrosio Tortorelli Approximation of the Mumford Shah) The results for T=100, Δt=0.4, α=8, β=4, 8, 16, 32, 64, ρ= 0.000001 95

Figure 3.5.6 (Result for the Ambrosio Tortorelli Approximation of the Mumford Shah) The result for T=100, Δt=0.4, α=8, β=0.1, ρ= 0.000000001 95

Figure 3.5.7 (Result for the Ambrosio Tortorelli Approximation of the Mumford Shah), (a) Smoothed Displacement Data Result, (b) The samples, between 190 and 380, of Figure 3.5.7.a 96

Figure 3.5.8 (Result for the Ambrosio Tortorelli Approximation of the Mumford Shah), (a) Smoothed Displacement Data Result, (b) The samples, between 190 and 380, of Figure 3.5.8.a 96

Figure 3.5.9 (Ambrosio Tortorelli Approximation of the Mumford Shah). (a) Second Derivative Result of Smoothed Displacement Data (b) Same as Figure 3.5.9.a except Original Displacement Data (Black Line) 97

Figure 3.5.10 (Ambrosio Tortorelli Approximation of the Mumford Shah). The Comparative Second Derivative Results of Smoothed Displacement Data and Original Acceleration Data 98

Figure 3.5.11 (Ambrosio Tortorelli Approximation of the Mumford Shah). The Comparative Second Derivative Results of Smoothed Displacement Data and Original Acceleration Data 98

Figure 3.6.1 (Perona-Malik Equation vs. Ambrosio-Tortorelli Equation) Green Line is the Ambrosio-Tortorelli Result: T=200, Δt=0.4, α=8, β=0.05, ρ= 0.000000001 Red Line is the Perona-Malik Result: T=200, Δt=0.4, λ=10, and σ=18 100

Figure 3.6.2 (Result for the Ambrosio Tortorelli Approximation of the Mumford Shah) The result for Result: T=200, Δt=0.4, α =8, β=0.05, ρ= 0.000001 100

Figure 3.6.3 (Results for the Ambrosio Tortorelli Approximation of the Mumford Shah) (a) Mean Value vs. Iteration (b) Variance vs. Iteration (c) Gradient vs. Iteration (d) Total Gradient vs. Iteration (e) Standard Deviation vs. Iteration (f) Entropy vs. Iteration 101

Figure 3.6.4 (The Comparative Second Derivative Results of Displacement Data by using some of the experimented Non-Linear Smoothing Filters). Black Line: Original Acceleration Data, (a) Sigma Filter Result by using σ=0.5, Filter size=91, (b) Perona Malik Filtering Result by using λ=10, T=18, Δt=0.25 and σ=3, (c) Ambrosio Tortorelli Approximation by using T=100, Δt=0.4, α=8, β=50, ρ=0.033, (d) Comparison of all results from Figure 3.6.4.a to Figure 3.6.4.c 102

Figure 3.7.1 (Baklava Filtering vs. Kuwahara Filtering) 104

Figure 3.7.2 (The samples, between 16 and 22, of Figure 3.7.1) 105

Figure 3.7.3 (Baklava Filtering vs. Kuwahara Filtering) (a) Smoothed Displacement Data Result, (b) The samples, between 105 and 125, of Figure 3.7.3.a 105

Figure 3.7.4 (Baklava Filtering vs. Kuwahara Filtering) (a) Smoothed Displacement Data Result, (b) The samples, between 0 and 110, of Figure 3.7.4.a 106

Figure 3.7.5 (Baklava Filtering vs. Kuwahara Filtering) (a) The samples, between 130 and 230, of Figure 3.7.4.a., (b) The samples, between 445 and 555 of Figure 3.7.4.a 106

Figure 3.7.6 (Baklava Filtering vs. Kuwahara Filtering) 107

Figure 3.7.7 (The samples, between 190 and 230, of Figure 3.7.6) 107

Figure 3.7.8 (Baklava Filtering vs. Kuwahara Filtering). (a) Second Derivative Result of Smoothed Golf Ball Data using Baklava Filtering & Kuwahara Filtering for the Mask Size=7 (b) Same as Figure 3.7.8.a except Kuwahara Filtering Result 108

FIGURE 3.7.9 (Baklava Filtering vs. Kuwahara Filtering). (a) Second Derivative Result of Smoothed Displacement Data using Baklava Filtering & Kuwahara Filtering for the Mask Size=7 (b) Same as Figure 3.7.9.a except Kuwahara Filtering Result 108

CHAPTER 1

INTRODUCTION

Data, resulting from quantitative measurements of a physical phenomenon, play a critical role in scientific investigation. These data typically exhibit randomness, either due to a limitation of the measuring device or a limitation on the measurability of the physical phenomena considered.

In the most simplistic scenario, a single quantity is being searched. For example, one tries to measure a quantity such as weight. When the second measurement for the same quantity is made, the result may be identical to the first measurement up to a certain accuracy, e.g. two measurements (0.482 and 0.485) are identical up to 2 decimal places however in many cases the second measurement will be substantially different because each measurement (X_i) is a sum of unknown true quantity and a measurement error ε_i. In such a situation, one often takes as many measurements as she or he can, and uses these measurements together in order to obtain a single number (e.g. average of all measurements), which reflects the measured quantity better than any of the obtained measurements. As the number of measurements increases, the accuracy of this number increases (more on this in Chapter 2).

In a second scenario, one attempts to measure a dynamic phenomenon. *i.e. a phenomena that produce time changing patterns such that the characteristics of the pattern at one time being interrelated with those at other times* [1]. Data, resulting from a measurement of a dynamic phenomenon, have a natural temporal ordering; thus they are called as time series. As in the first scenario where a single quantity is measured, the measurement taken at each time instant t_k reflects a noisy measurement:

$$f_M(t_k) = f_T(t_k) + \mathcal{E}(t_k)$$

where,

$f_M(t_k)$: measured value at time $\mathrm{t_k}$,
$f_T(t_k)$: true value at time t_k,
$\mathcal{E}(t_k)$: error at time t_k.

If one can afford taking multiple measurements,

$$f_M^i(t_k) \qquad i = 1, 2, 3, \ldots$$

then one can estimate the true value $f_T(t_k)$ e.g. by averaging.

More over, obtaining a second measurement is not possible. Hence, one has to exploit the relation among values at different time instances as the characteristics of the pattern at one time is interrelated with those at other times. Data smoothing techniques should exploit the relation among data values at different time instances. Smoothing should preserve critical features in the data set while removing noise. On one hand, when a data is smoothed less, it contains more noise. On the other hand, when a data is smoothed more, critical features are lost. There should be a balance in terms of how much noise could be tolerated versus what feature should be preserved.

The level of smoothing is especially critical if higher order derivatives are to be estimated from the smoothed data, and features of interest are sudden changes in the derivatives. First derivative is the rate at which the dependent variable changes with respect to a small change in an independent variable. If the independent variable corresponds to time and the dependent variable corresponds to positional displacement, the first derivative corresponds to velocity. The first derivative of velocity with respect to time is the second derivative of position and corresponds to acceleration. Therefore, being able to estimate the second derivative of a position data is important.

Unfortunately, numerical differentiation is not robust to noise. A demonstration is given below using a Gaussian function (Figure 1.1) as the function to be differentiated numerically. Plots, shown between the Figure 1.1 and Figure 1.7, are taken from [19] and modified. These plots can be found in image processing textbooks as well. The first derivative result is depicted in Figure 1.2, and the second derivative result is depicted in Figure 1.3. It is seen that both the first derivative and the second derivative results seem correct, because the function to be differentiated does not contain noise. In Figure 1.4, noise added form of the function in Figure 1.1 is shown.

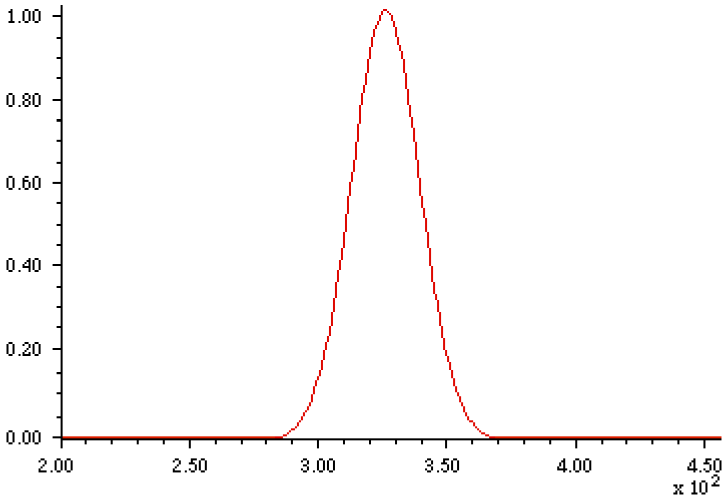


Figure 1.1: **Original Data.**

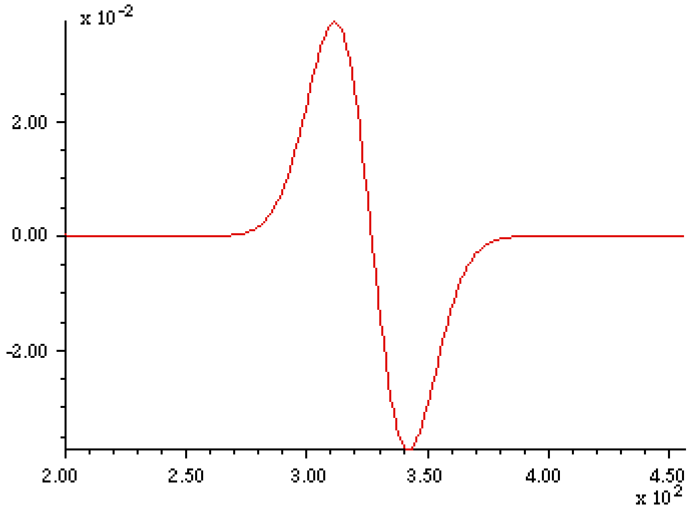


Figure 1.2: First Derivative Result of Figure 1.1

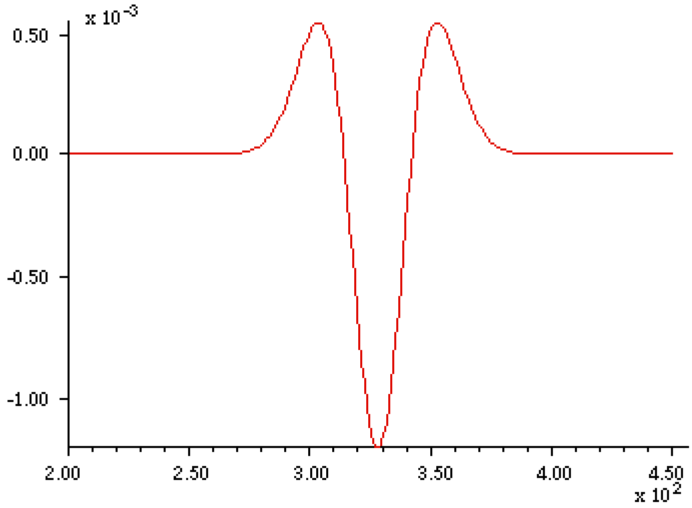


Figure 1.3: Second Derivative Result of Figure 1.1

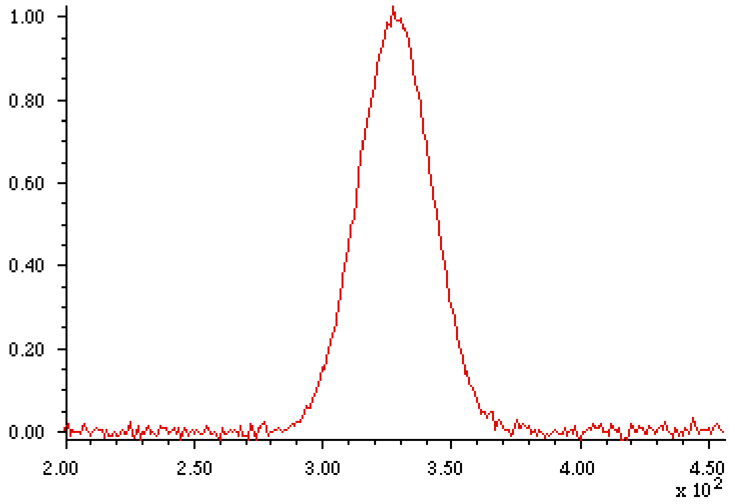


Figure 1.4: Noisy Data.

After some smoothing, numerical differentiation gives the results shown in Figures 1.5 and 1.6 for the first and the second derivatives respectively. Notice that the higher order derivative is more sensitive to noise because differentiation amplifies high frequency components, hence amplifies noise. Specifically, notice that the second derivative estimate depicted in Figure 1.6 oscillates roughly in the range [$-3.5\text{x}10^{-2}$, $3.5\text{x}10^{-2}$] which is nearly 50 times larger than the range of the correct estimate depicted in Figure 1.3.

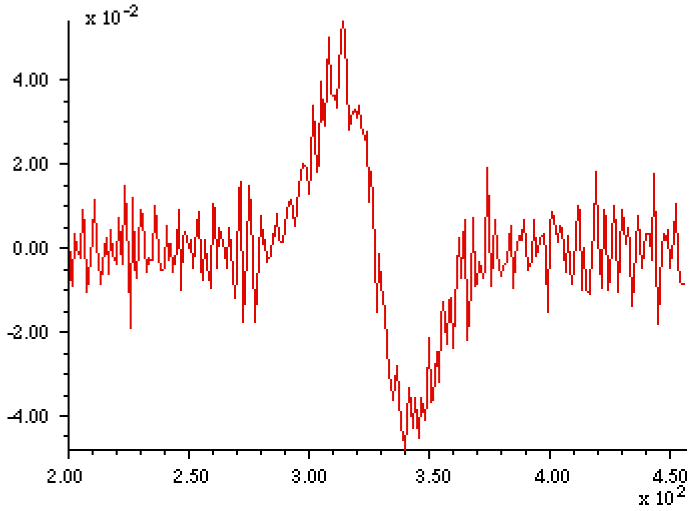


Figure 1.5: First Derivative Result after Smoothing in Figure 1.4

The signal to noise ratio of a derivative can be significantly poorer in inadequate smoothing than that of the derivative of the original signal. However with a sufficient smoothing operation, the signal to noise ratio of the smoothed derivative can decrease to acceptable level. Better derivative estimates can be obtained using a different smoothing strategy. A second derivative estimate obtained after smoothing the data in Figure 1.4 more, which is depicted in Figure 1.7,

Notice that oscillations are significantly reduced compared to those in Figure 1.6. The new estimate is roughly in [$-2x10^{-3}$, $1x10^{-3}$] range, which is only twice large than the interval of the correct second derivative depicted in Figure 1.3. The stationary nature of the signal in Figure 1.1, whose noisy form is in Figure 1.4, makes it possible for us to find the correct smoothing level that removes the noise thus enhance the signal to noise ratio.

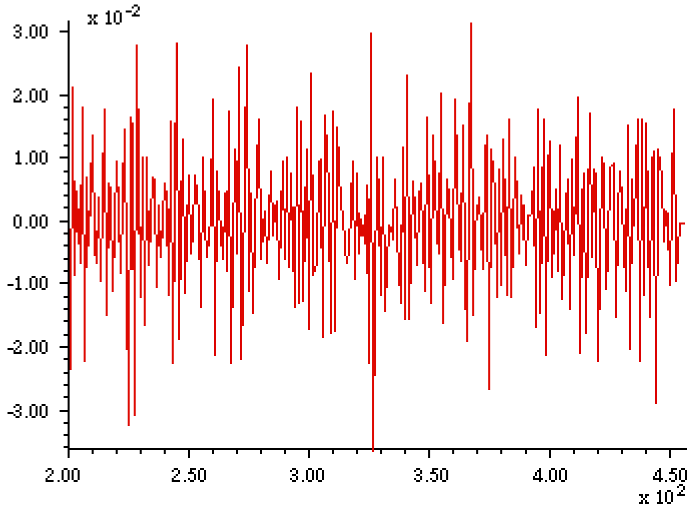


Figure 1.6: Second Derivative Result after Smoothing less in Figure 1.4

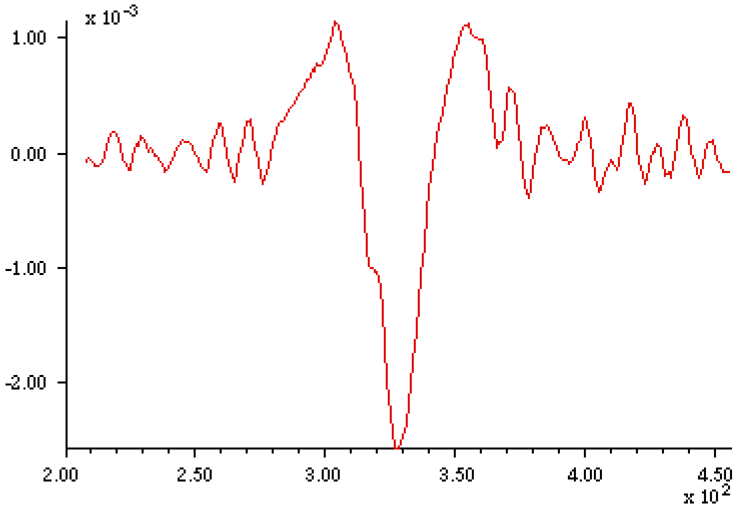


Figure 1.7: Second Derivative Result after Smoothing in Figure 1.4

Unfortunately many interesting signals and time series are not stationary. For non-stationary signals, one can not make a clear distinction between noise and signal.

In this thesis, the effect of different smoothing strategies on a one dimensional time series representing a positional displacement and on its estimated second derivative is experimentally investigated.

Experiments are performed on two data which are presented in [2, 3]. The second derivatives are estimated applying a Smoothing filter. The first data [2] is the displacement for a free falling golf ball. Thus the double differentiation should give the gravitational acceleration which is 9.81 m/s^2. The second data set is an angular displacement for a swinging pendulum whose motion is impacted by a solid barrier during its motion. Reference acceleration is measured using an accelerometer. Double differentiation of the Displacement data should be close to the reference acceleration.

Smoothing filters are grouped into two categories: linear and non-linear. Widely used filters developed for handling piecewise smooth data, where the jumps at the first derivative are of interest, are experimented with. These filters have been studied in terms of their effectiveness in the context of image processing [18].

Smoothing is especially important when acceleration (the second order derivative) is of interest which is estimated from the smoothed data. Smoothing should preserve critical features while reducing noise. Smoothing is the initial and most important part of the data analyzing. When the data are smoothed less, it contain more noise. On the other hand when the data are smoothed more, critical features are lost. So this balance is important to determine the amount of smoothing.

The complexity of the problem comes from non-stationary nature of the selected displacement signals. Popular filtering algorithms are successful especially when the data to be processed is stationary (Woltring, 1990, 1995), but the signals experimented in this thesis are non-stationary. That is, some parts of the signal contain higher frequency components than the other parts. Noise and signal spectrums overlap. Locally at some point, interesting feature is a high frequency component, thus it is hard to separate this feature from the noise.

Main topics covered in this thesis can be summarized as follows: The effects of widely used Linear smoothing algorithms are reviewed, and estimated second derivative results are experimentally investigated for a one dimensional data set. And the same processes are done for some Non-linear smoothing algorithms. Perona Malik filtering and Ambrosio Tortorelli approximation of Mumford Shah model are reviewed and experimented for the same data set too. An alternative method to the Kuwahara filter is proposed and analyzed on the second derivative results. The success of the smoothing methods is analyzed by considering the second derivative of the smoothed data sets.

1.1 EXPERIMENTAL DATA SETS

Experiments are performed on two data, both taken from [2, 3].

- The first data **(Golf ball drop data)**, is the recorded position of a dropped golf ball falling under gravitational force, shown in Figure 1.1.1.

The position d (meters) is recorded from time=0.00, to time=0.48265 (seconds). In total, these are 50 equally spaced measurements with $\Delta t = 0.00985$ (seconds) [2]. Double differentiation should produce constant acceleration.

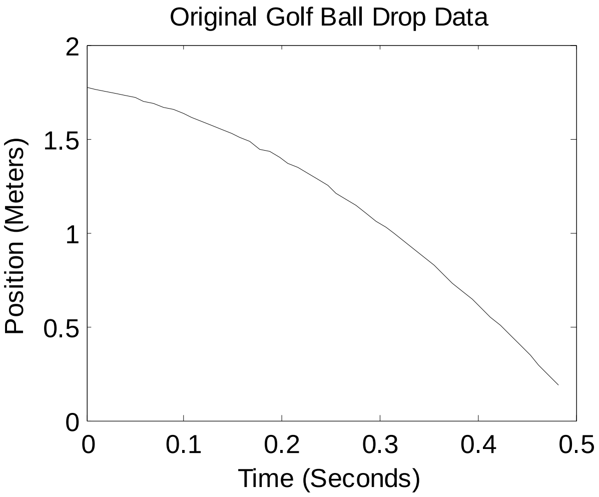


Figure 1.1.1: Original Golf Ball Drop Data

- The second data (**Accelerometer data)** which is taken from an experiment in Biomechanics [3] contains two parts: angular displacement and acceleration measured by accelerometer.

In the Accelerometer data set, the units of the angular displacement are in radians, and the units of the acceleration are in radians per seconds^2. In total, there are 600 equally spaced measurements with $\Delta t = \frac{1}{512}$ (seconds), in Figure 1.1.2.a and Figure 1.1.2.b. While a solid is swinging forward in horizontal direction, suddenly a soft barrier impacts its motion and it stops a little time before it swings back.

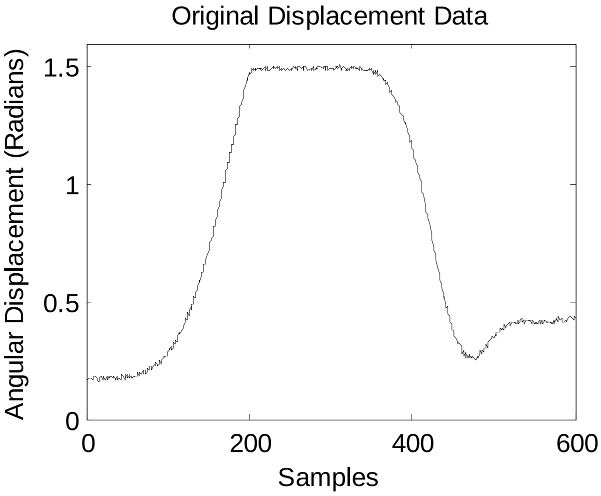


Figure 1.1.2.a: Original Angular Displacement Data

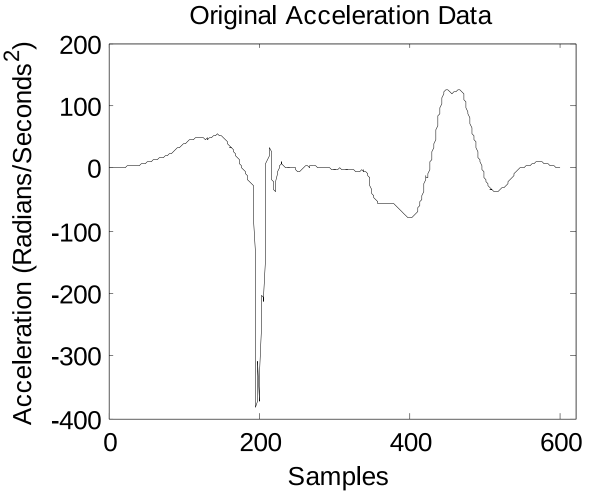


Figure 1.1.2.b: Original Acceleration Data

1.2 ORGANIZATION

The organization of thesis is as follows: In Chapter 2, some Linear smoothing filters are reviewed and implemented by using two data taken from Biomechanics Repository and second derivative results are presented. Same works in Chapter 2 are applied for Non-linear smoothing filters in Chapter 3, and a new method, alternative to Kuwahara filter, is proposed and implemented. The thesis concludes with Chapter 4 in which summary and discussions are provided.

CHAPTER 2

LINEAR SMOOTHING FILTERS

Linear and Time Invariant (LTI) filter can be defined as [4]:

$$Output(x) = \sum_{i=-\infty}^{\infty} C_i \cdot Input(x-i), \text{ where } x \in \mathbf{Z}. \tag{2.1}$$

$Input(x),$ where $x \in \mathbf{Z}$: is the input data,
$C_i,$ where $i \in \mathbf{Z}$: is a set of real numbers,
$Output(x),$ where $x \in \mathbf{Z}$: is the output data.

Notice that the filtered value at each point x is calculated by taking the weighted average of input values [5]. Further notice that the weights are merely a function of the distance (offset) $\boldsymbol{i}$ from the position $\boldsymbol{x}$.

Thus, a Linear and Time Invariant (LTI) filter does not take the characteristic properties of the given data into account. Each data point is treated equally without paying attention to the context (i.e. a pattern which is formed around each point).

Depending on the choice of weight function and the mathematical model, linear filters take various forms.

In this chapter, six kinds of Linear smoothing filters are experimented.

1. Mean Filter
2. Iterated Mean Filter
3. Gaussian Filter
4. Ideal Low Pass Filter
5. Butterworth Low Pass Filter
6. Linear Diffusion Filter

Our test data do not include the data, eliminated from all kinds of noise. So, to be able to see the effect of the Linear smoothing filters, a sinusoidal test data is used with some added noise. Signal to noise ratio of the original noisy test data is equal to 10.9379.

The Linear smoothing filter results are:

Signal to noise ratio of the test data, filtered by Iterated Mean filter, is 19.1682. Signal to noise ratio of the test data, filtered by Gaussian filter, is 19.9133. Signal to noise ratio of the test data, filtered by Ideal Low Pass filter, is 23.2372. Signal to noise ratio of the test data, filtered by Butterworth Low Pass filter, is 23.2155. Signal to noise ratio of the test data, filtered by Linear Diffusion filter, is 19.9279.

2.1 MEAN FILTER

Typically due to noise, one measurement is not enough to get the correct value. Thus series of measurements (X_i) are collected. These measurements are associated with a single point on a time line as seen in Figure 2.1.1.

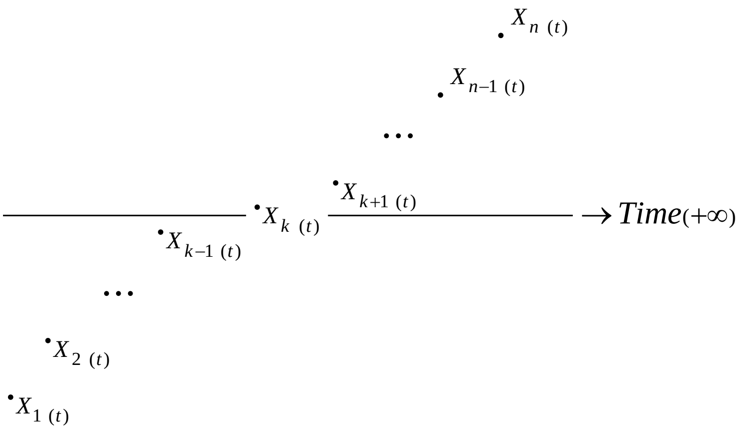


Figure 2.1.1: Measurements.

X_i : The measurements where $i = 1,\ 2,\ 3,\ 4,\ 5,\ \ ...\ ,n$
n : The number of measurements.

It is seen that, these measurements are collected around the real value (μ), which is nearly at the center of the measurements.

Sum of the difference among the measurements, less than the value of μ, and μ is equal to the sum of the difference among the measurements, greater than the value of μ, and μ [1].

$$(\mu - X_1) + (\mu - X_2) + \ ...\ + (\mu - X_{k-1}) = (X_{k+1} - \mu) + (X_{k+2} - \mu) + \ ...\ + (X_n - \mu) \quad (2.2)$$

$$\mu = \frac{1}{n}\sum_{i=1}^{n} X_i \quad (2.3)$$

So, (2.3) is the arithmetic mean of the measurements. This is called as a sample mean.

The difference between a measurement and the center of cluster, μ, gives error, (2.4).

$$\mathcal{E}rror_i = X_i - \mu \quad (2.4)$$

Total $\mathcal{E}$*rror* can be calculated by using (2.5),

$$Total\_\mathcal{E}rror = \sum_{i=1}^{n} \mathcal{E}rror_i^2 \tag{2.5}$$

In the absence of multiple measurements, the following assumption can be made:

<u>*Assumption*</u>: In a time series, consecutive measurements are close to each other. The measured values are collected around the real value (μ), which is nearly at the center of the measurements (That is the measured phenomenon varies slowly).

One can consider the smallest window size of 3. Then the Mean filter takes the form in (2.6).

$$\frac{1}{3}\begin{pmatrix} 1 & \boxed{1} & 1 \end{pmatrix} \tag{2.6}$$

Mean filtering reduces the change of intensity variation between 2 neighbor data, and simply replaces each data value with the average value of its neighbors and its own. Consequently, Mean filtering can be thought as computing by moving averages over a finite window size. For example, when (2.6) is used for the data in (2.7),

$$\begin{bmatrix} \ldots\ g\ \boxed{\mathrm{f}}\boxed{e}\boxed{a}\boxed{b}\boxed{c}\ d\ \ldots \end{bmatrix} \tag{2.7}$$

$$\begin{bmatrix} \ldots\ (g)\ \boxed{\dfrac{g+\mathbf{f}+e}{3}}\boxed{\dfrac{\mathrm{f}+\boldsymbol{e}+a}{3}}\boxed{\dfrac{e+\boldsymbol{a}+b}{3}}\boxed{\dfrac{a+\boldsymbol{b}+c}{3}}\boxed{\dfrac{b+\boldsymbol{c}+d}{3}}\ (d)\ \ldots \end{bmatrix} \tag{2.8}$$

The larger the mask size, smoother the result. The result of the Mean filter is seen in (2.9), when the mask size is increased from 3 to 5.

$$\frac{1}{5}\begin{bmatrix} \ldots\ (g)\ (\mathrm{f})\ \boxed{\mathrm{g+f}+\boldsymbol{e}+a+b}\boxed{\mathrm{f}+e+\boldsymbol{a}+b+c}\boxed{e+a+\boldsymbol{b}+c+d}\ (c)\ (d)\ \ldots \end{bmatrix} \tag{2.9}$$

If the mask size is defined as the size of the data (N), the data goes to an average value. It means that it goes to a line.

$$\frac{1}{N}\begin{bmatrix} \ldots\ \boxed{1}\ \boxed{1}\ \boxed{1}\ \boxed{1}\ \boxed{1}\ \boxed{1}\ \boxed{1}\ \boxed{1}\ \ldots \end{bmatrix} \tag{2.10}$$

Filter results can be seen at the end of the Section 2.2 by comparing with the Iterated Mean filter.

2.2 ITERATED MEAN FILTER

Consider a Mean filter with window size of 3. Single application of Mean filter to (2.7) gives (2.11). If the same filter is applied to the output of the filter given in (2.11), new output is obtained as (2.12).

$$\frac{1}{3}\left[\dots\ (g)\ \boxed{g+\mathbf{f}+e}\boxed{\mathrm{f}+\boldsymbol{e}+a}\boxed{e+\boldsymbol{a}+b}\boxed{a+\boldsymbol{b}+c}\boxed{b+\boldsymbol{c}+d}\ (d)\ \dots\right] \tag{2.11}$$

$$\frac{1}{9}\left[..\ (g+\mathrm{f}+e)\ \boxed{g+2\mathrm{f}+3\boldsymbol{e}+2a+b}\boxed{\mathrm{f}+2e+3\boldsymbol{a}+2b+c}\boxed{e+2a+3\boldsymbol{b}+2c+d}\ (b+c+d)\ ..\right] \tag{2.12}$$

As you see from (2.12), when iteration number is increased, then the data, which are processed, start to be affected from distant data. For example, in (2.12), the value of e is effected by g, f, a and b in second iteration, but a and f affects it more than g and b. If the iteration number continues increasing, the value of e will be affected from all the data.

In my experiment for the Iterated Mean filter, 2 kinds of filter masks are used.

$$\frac{1}{3}\begin{pmatrix}1 & \boxed{1} & 1\end{pmatrix} \quad \text{and} \quad \frac{1}{5}\begin{pmatrix}1 & 1 & \boxed{1} & 1 & 1\end{pmatrix}$$

Mask 1 Mask 2

The results are seen in Figure 2.2.1 and in Figure 2.2.2, after applying Mean filter and Iterated Mean filter to our datasets by using Mask 1 and Mask 2.

2.2.1 Golf Ball Data Results for Mean Filtering and Iterated Mean Filtering

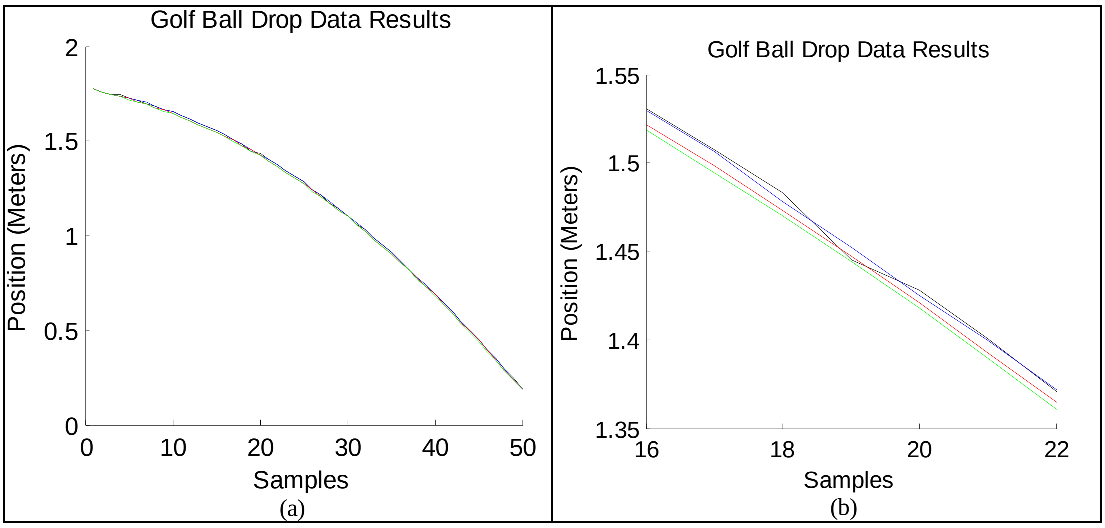


Figure 2.2.1: (Mean Filter vs. Iterated Mean Filter). (a) Golf Ball Data Result using the mask 1, **(b)** The samples, between 16 and 22, of Figure 2.2.1.a.

Black Line: The Original Golf Ball Drop Data.

Blue Line: Smoothed Golf Data by using the mask 1 (1x3),

Red Line: Smoothed Golf Data by using the mask 1 for 20 iterations,

Green Line: Smoothed Golf Data by using the mask 1 for 30 iterations

To be able to realize the difference between two figures, check the data values in Tables 2.2.1.1, 2.2.1.2, 2.2.1.3 and 2.2.1.4.

Table 2.2.1.1: Original Golf Ball Data Values

Height (meters)	:	...	1,7260	1,7150	1,6980	1,6830	1,6670	1,6510	…
Time (seconds)	:	...	0.0394	0.04925	0.0591	0.06895	0.0788	0.08865	...
Sample #	:	...	5	6	7	8	9	10	...

Table 2.2.1.2: Smoothed Golf Data after applying Mean filter by using mask 1

Height (meters)	:	...	1.727	1.713	1.6987	1.6827	1.667	1.65	...
Time (seconds)	:	...	0.0394	0.04925	0.0591	0.06895	0.0788	0.08865	...

Table 2.2.1.3: Smoothed Golf Data after applying Mean filter by using mask 1 for 20 iterations

Height (meters)	:	...	1,7206	1,7069	1,6925	1,6772	1,6610	1,6439	...
Time (seconds)	:	...	0.0394	0.04925	0.0591	0.06895	0.0788	0.08865	...

Table 2.2.1.4: Smoothed Golf Data after applying Mean filter by using mask 1 for 30 iterations

Height (meters)	:	...	1,7185	1,7045	1,6898	1,6743	1,6581	1,6409	...
Time (seconds)	:	...	0.0394	0.04925	0.0591	0.06895	0.0788	0.08865	...

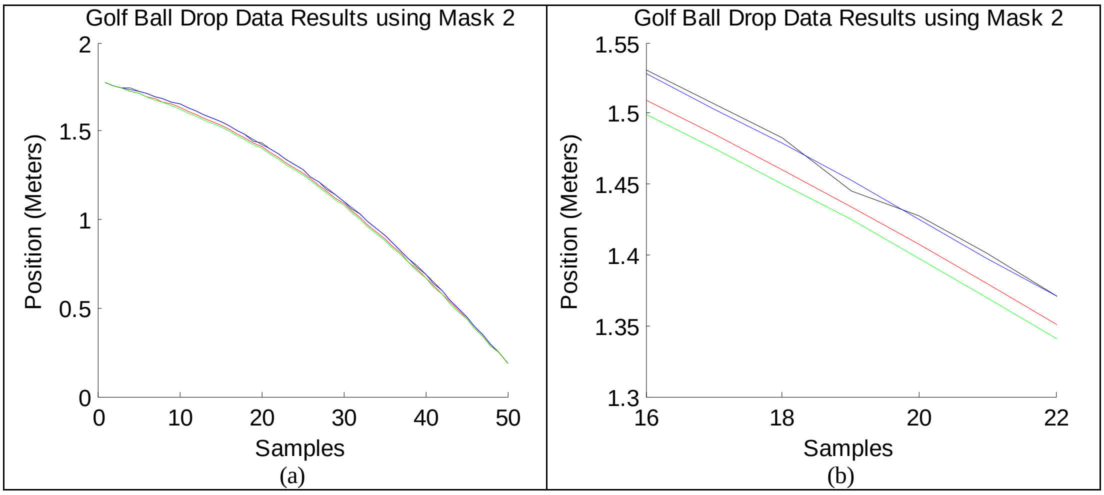


Figure 2.2.2: (Mean Filter vs. Iterated Mean Filter). (a) Golf Ball Data Result using the mask 2, **(b)** The samples, between 16 and 22, of Figure 2.2.2.a.

Black Line: The Original Golf Ball Drop Data.

Blue Line: Smoothed Golf Data by using the mask 2 (1x5),

Red Line: Smoothed Golf Data by using the mask 2 for 20 iterations,

Green Line: Smoothed Golf Data by using the mask 2 for 30 iterations

To be able to realize the difference between two figures, check the data values in Tables 2.2.1.1, 2.2.1.5, 2.2.1.6 and 2.2.1.7.

Table 2.2.1.5: Smoothed Golf Data after applying Mean filter by using mask 2

Height (meters)	**:**	...	1,7254	1,7124	1,6978	1,6828	1,6662	1,6490	...
Time (seconds)	**:**	...	0.0394	0.04925	0.0591	0.06895	0.0788	0.08865	...

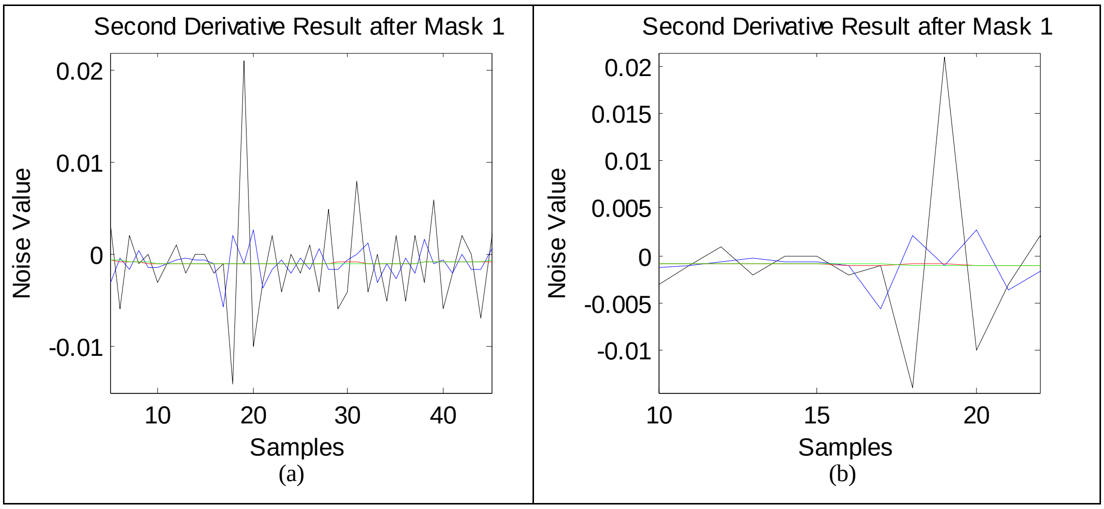
Table 2.2.1.6: Smoothed Golf Data after applying Mean filter by using mask 2 for 20 iterations

Height (meters)	**:**	...	1,7134	1,6983	1,6827	1,6666	1,6498	1,6322	...
Time (seconds)	**:**	...	0.0394	0.04925	0.0591	0.06895	0.0788	0.08865	...

Table 2.2.1.7: Smoothed Golf Data after applying Mean filter by using mask 2 for 30 iterations

Height (meters)	**:**	...	1,7098	1,6938	1,6773	1,6604	1,6428	1,6247	...
Time (seconds)	**:**	...	0.0394	0.04925	0.0591	0.06895	0.0788	0.08865	...

Black line is the second derivative of the input data. There are so many noises. As you see from Figure 2.2.3, Figure 2.2.4 and Figure 2.2.5, when smoothing effect of the Mean filter is increased then the second derivative of the data become less noisy, and it goes to a line.



(a)

(b)

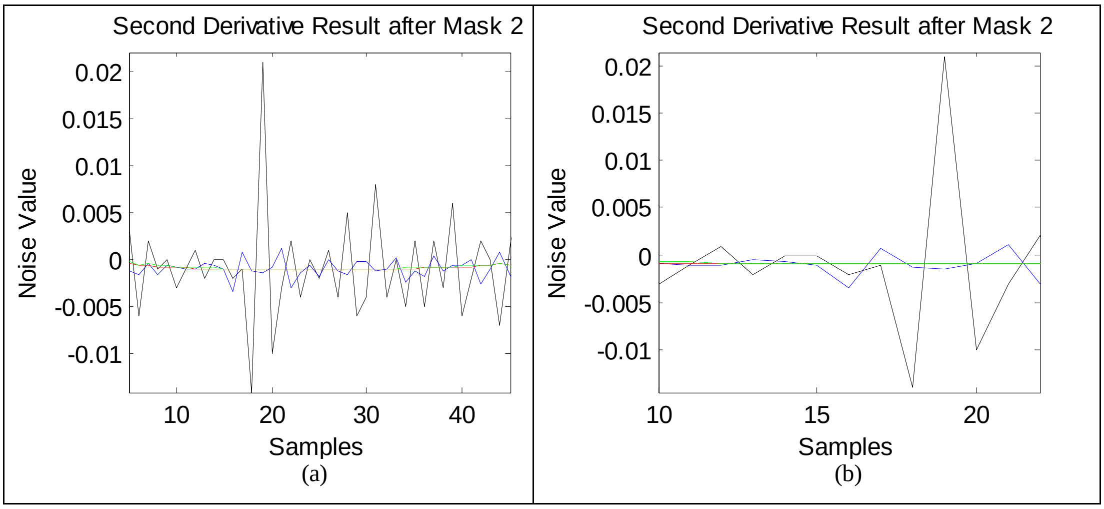
Figure 2.2.3: (Mean Filter vs. Iterated Mean Filter). (a) Second Derivative Result of Smoothed Golf Ball Data using the mask 1, **(b)** The samples, between 10 and 22, of Figure 2.2.3.a.

Black Line: Second Derivative of the Original Golf Ball Drop Data.

Blue Line: Second Derivative of the Data by using the mask 1 (1x3),

Red Line: Second Derivative of the Data by using the mask 1 for 20 iterations,

Green Line: Second Derivative of the Data by using the mask 1 for 30 iterations



(a)

(b)

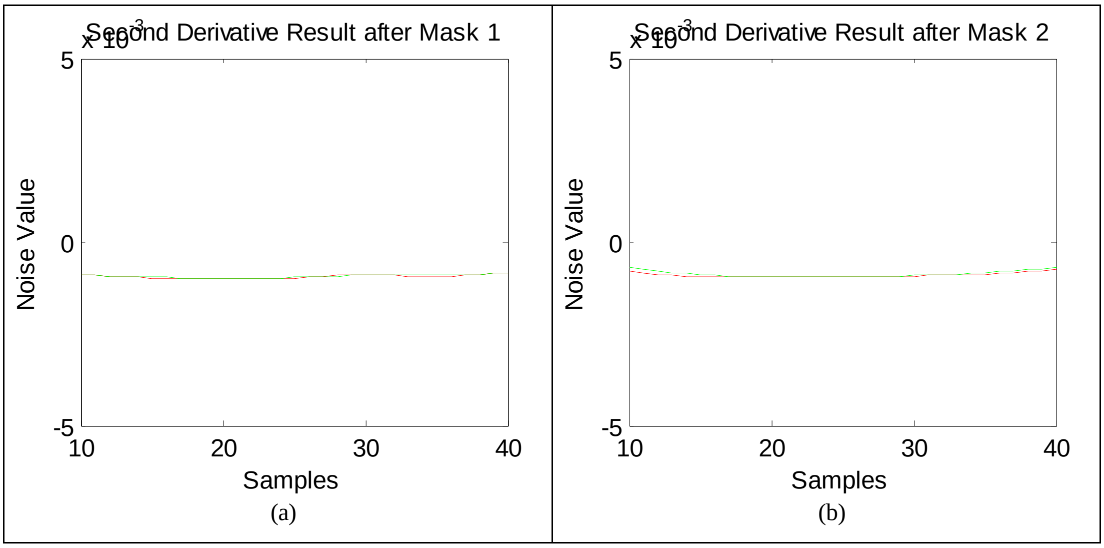
Figure 2.2.4: (Mean Filter vs. Iterated Mean Filter). (a) Second Derivative Result of Smoothed Golf Ball Data using the mask 2, **(b)** The samples, between 10 and 22, of Figure 2.2.4.a.

Black Line: Second Derivative of the Original Golf Ball Drop Data.

Blue Line: Second Derivative of the Data by using the mask 2 (1x5),

Red Line: Second Derivative of the Data by using the mask 2 for 20 iterations,

Green Line: Second Derivative of the Data by using the mask 2 for 30 iterations



(a)

(b)

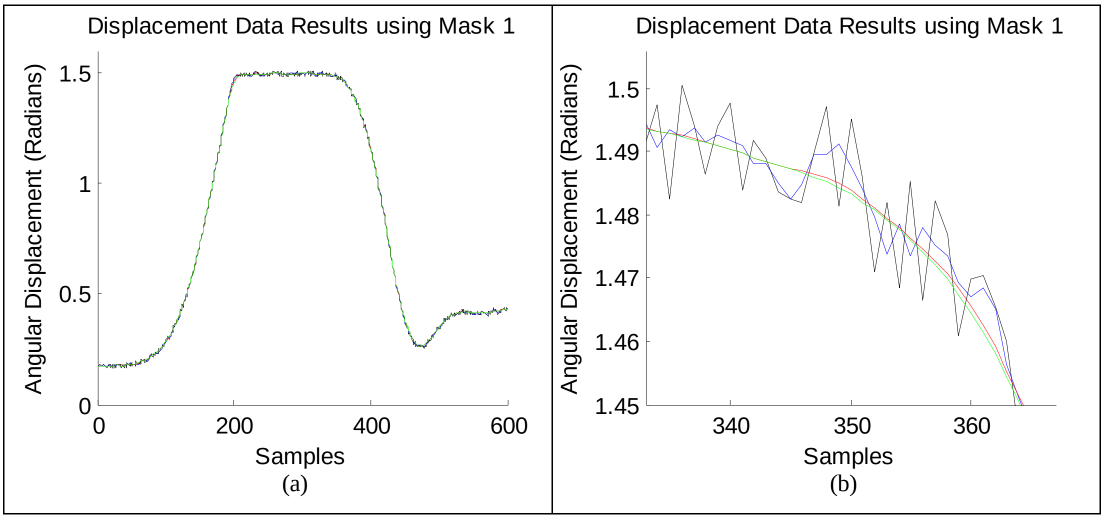
Figure 2.2.5: (Iterated Mean Filter). (a) Second Derivative Result of Smoothed Displacement Data using the **mask 1** (1x3), **(b)** Second Derivative Result of Smoothed Displacement Data using the **mask 2** (1x5).

Red Line: Second Derivative Result of the Displacement Data for 20 iterations,

Green Line: Second Derivative Result of the Displacement Data for 30 iterations.

2.2.2 Displacement Data Results for Mean Filtering

Comparative results of the Mean filter and Iterated Mean filter for various filter size and iteration can be seen in Figure 2.2.6 and Figure 2.2.7.



(a)

(b)

Figure 2.2.6: (Mean Filter vs. Iterated Mean Filter). (a) Displacement Data Result using the mask 1, **(b)** The samples, between 333 and 364, of Figure 2.2.6.a.

Black Line: The Original Displacement Data.

Blue Line: Smoothed Displacement Data by using the mask 1 (1x3),

Red Line: Smoothed Displacement Data by using the mask 1 for 20 iterations,

Green Line: Smoothed Displacement Data by using the mask 1 for 30 iterations

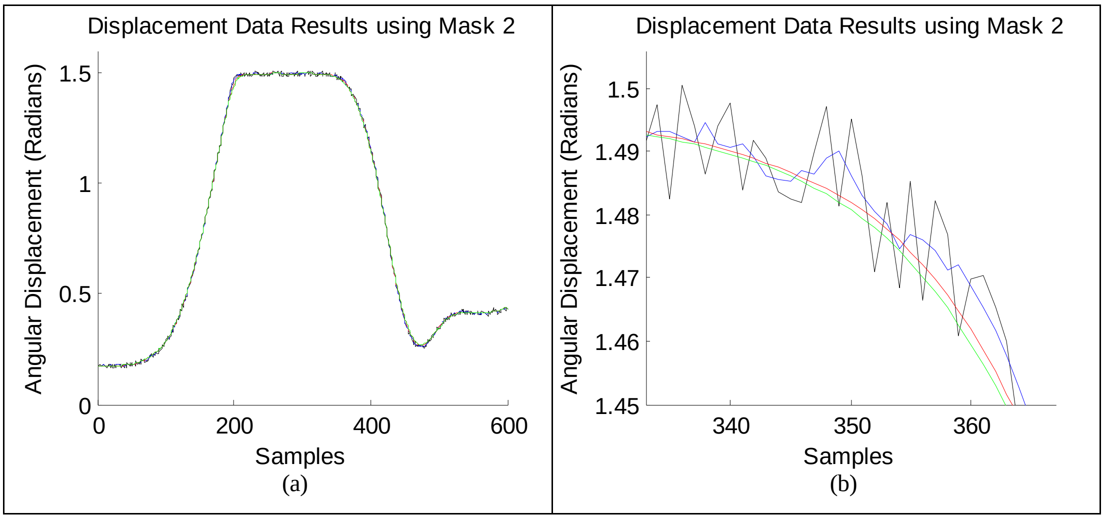


Figure 2.2.7: (Mean Filter vs. Iterated Mean Filter). (a) Displacement Data Result using the mask 2, **(b)** The samples, between 333 and 364, of Figure 2.2.7.a.

Black Line: The Original Displacement Data.

Blue Line: Smoothed Displacement Data by using the mask 2 (1x5),

Red Line: Smoothed Displacement Data by using the mask 2 for 20 iterations,

Green Line: Smoothed Displacement Data by using the mask 2 for 30 iterations

Comparative Second Derivative Result of the Smoothed Displacement data by using the Mean filter and Iterated Mean filter for various filter size and iteration can be seen in Figure 2.2.8 and Figure 2.2.9.

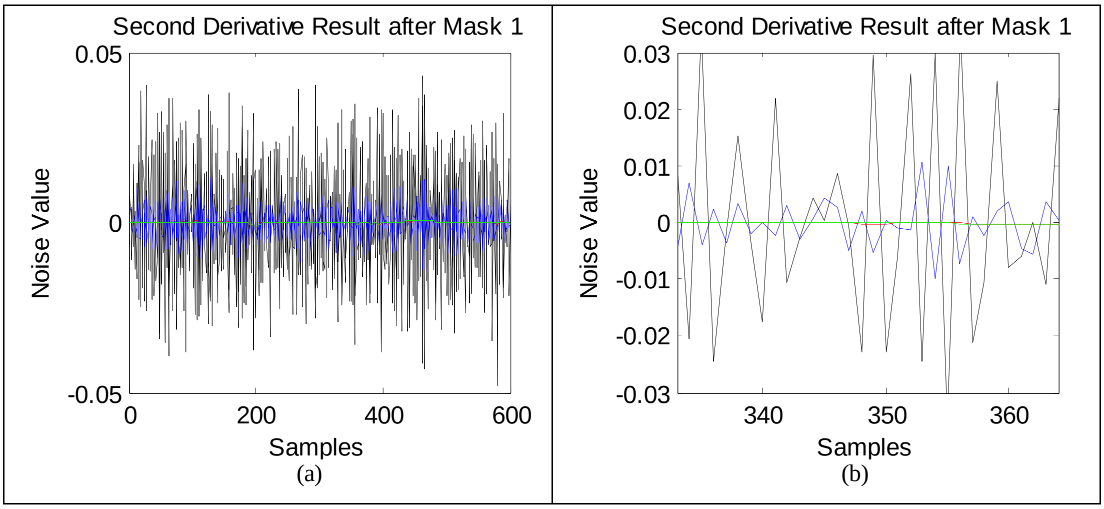


Figure 2.2.8: (Mean Filter vs. Iterated Mean Filter). (a) Second Derivative Result of Smoothed Displacement Data using the mask 1, **(b)** The samples, between 333 and 364, of Figure 2.2.8.a.

Black Line: Second Derivative of the Original Displacement Data.

Blue Line: Second Derivative of the Data by using the mask 1 (1x3),

Red Line: Second Derivative of the Data by using the mask 1 for 20 iterations,

Green Line: Second Derivative of the Data by using the mask 1 for 30 iterations

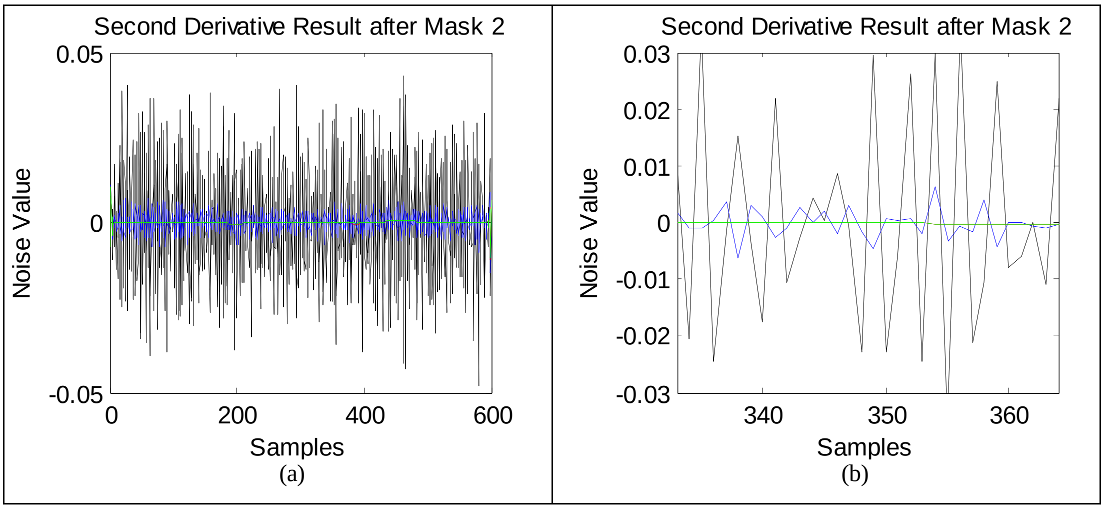


Figure 2.2.9: (Mean Filter vs. Iterated Mean Filter). (a) Second Derivative Result of Smoothed Displacement Data using the mask 2, **(b)** The samples, between 333 and 364, of Figure 2.2.9.a.

Black Line: Second Derivative of the Original Displacement Data.

Blue Line: Second Derivative of the Data by using the mask 2 (1x5),

Red Line: Second Derivative of the Data by using the mask 2 for 20 iterations,

Green Line: Second Derivative of the Data by using the mask 2 for 30 iterations

Second Derivative Result of the Smoothed Displacement data by using the Iterated Mean filter for various filter size and iteration can be seen in Figure 2.2.10.

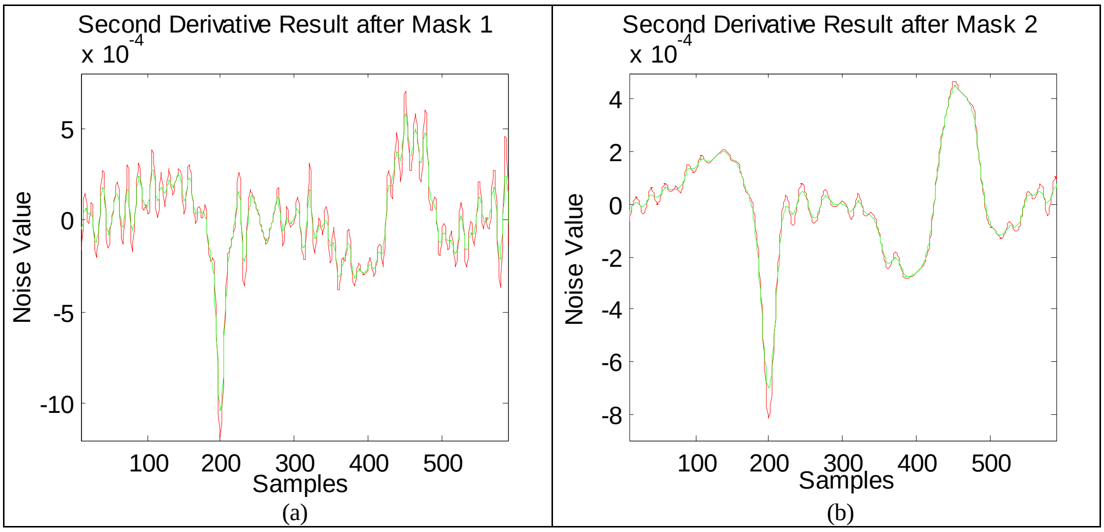


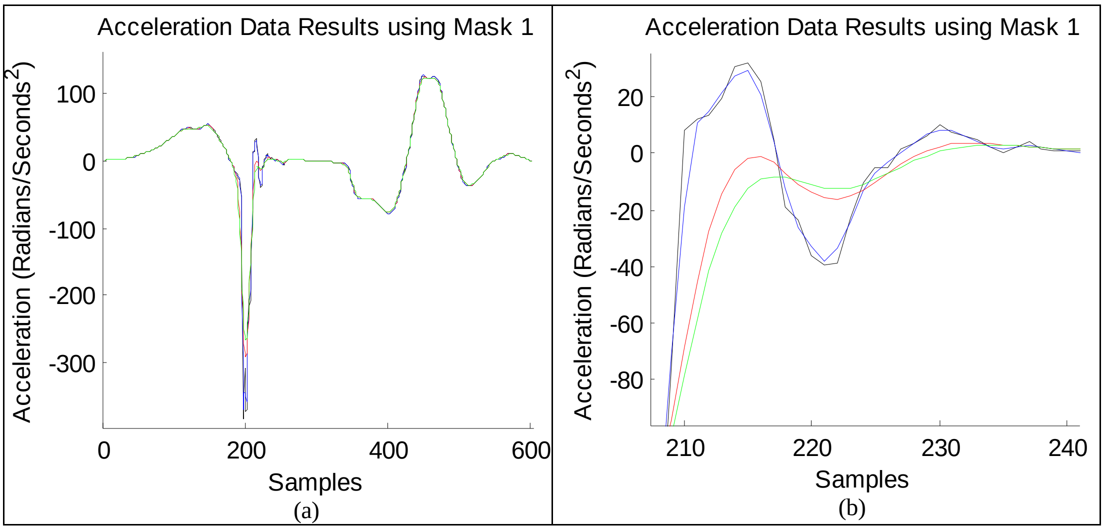
Figure 2.2.10: (Iterated Mean Filter). (a) Second Derivative Result of Smoothed Displacement Data using the **mask 1** (1x3), **(b)** Second Derivative Result of Smoothed Displacement Data using the **mask 2** (1x5).

Red Line: Second Derivative Result of the Displacement Data for 20 iterations,

Green Line: Second Derivative Result of the Displacement Data for 30 iterations.

2.2.3 Acceleration Data Results for Mean Filtering

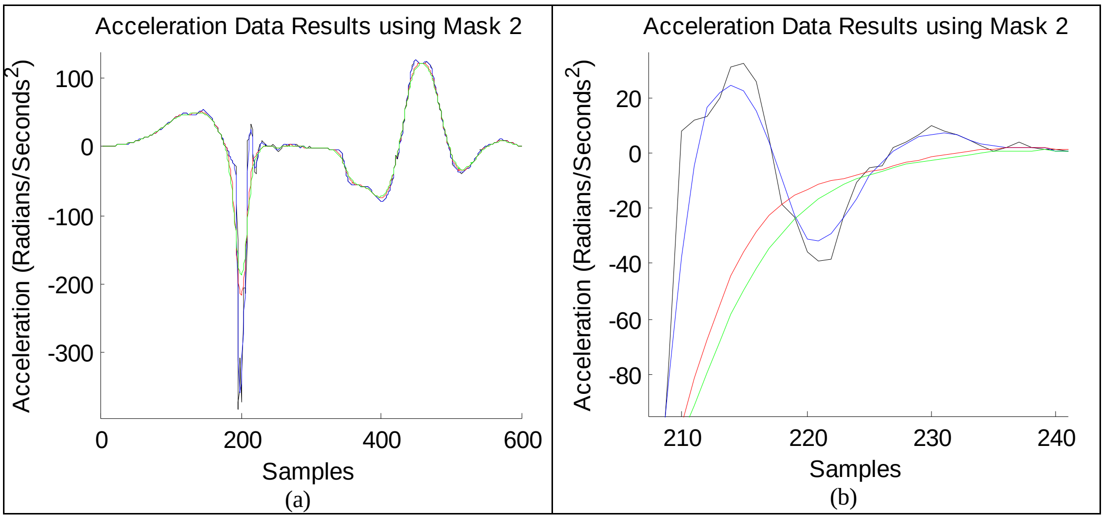
As it is seen from Figure 2.2.11 and Figure 2.2.12, when the filter size is increased, fluctuations and the instant noise start to decrease. However, at the same time, sudden changes evaporate and the fluctuating transitions of the data become quite soft.



(a) (b)

Figure 2.2.11: (Mean Filter vs. Iterated Mean Filter). (a) Acceleration Data Result using the mask 1, **(b)** The samples, between 207 and 242, of Figure 2.2.11.a.

Black Line: The Original Acceleration Data.
Blue Line: Smoothed Acceleration Data by using the mask 1 (1x3),
Red Line: Smoothed Acceleration Data by using the mask 1 for 20 iterations,
Green Line: Smoothed Acceleration Data by using the mask 1 for 30 iterations.



(a) (b)

Figure 2.2.12: (Mean Filter vs. Iterated Mean Filter). (a) Acceleration Data Result using the mask 2, **(b)** The samples, between 207 and 242, of Figure 2.2.12.a.

Black Line: The Original Acceleration Data.
Blue Line: Smoothed Acceleration Data by using the mask 2 (1x5),
Red Line: Smoothed Acceleration Data by using the mask 2 for 20 iterations,
Green Line: Smoothed Acceleration Data by using the mask 2 for 30 iterations.

2.2.4 Comparing Mean and Iterated Mean Filters

Both the amount of smoothing and choice of smoothing procedure are important. Results are shown in Figures 2.2.13, 2.2.14 and 2.2.15. The peak, seen in the Figure 2.2.13 in the area of 1, is the critical feature of the Acceleration data. This peak should be preserved after smoothing operation, but the peak, seen in the area of 2, is the noise that should be smoothed as much as possible. This is the major difficulty of this problem. In the area of 2, the results of the filtering are not successful, but Iterated Mean filtering result, seen by blue line in Figure 2.2.15, gives comparatively better result than others, but the same line is not successful in the area of 1. The peak in the area of 1 is not a noise, but the filtered signal lost this critical feature as shown by blue line. The result of the filtering, red line in Figure 2.2.14, gives relatively good result while giving the worst result in the area of 2. Also the results of filtered Displacement data can also be seen in Figure 2.2.6 and in Figure 2.2.7 by using the same parameters.

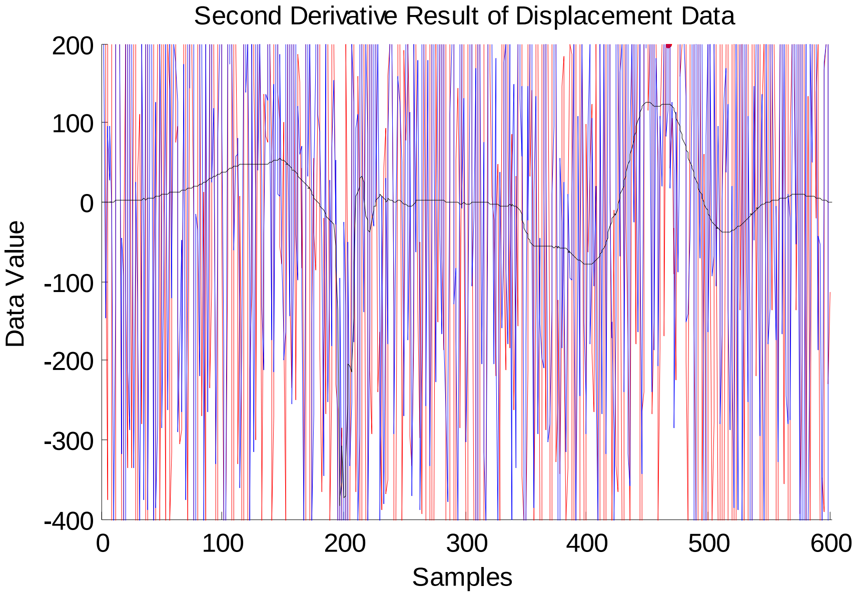


Figure 2.2.13: (Mean Filtering). The Comparative Second Derivative Results of Smoothed Displacement Data and Original Acceleration Data.

Black Line: Original Acceleration Data

Red Line: Second Derivative of Smoothed Displacement Data by using the mask 1 (1x3)

Blue Line: Second Derivative of Smoothed Displacement Data by using the mask 2 (1x5)

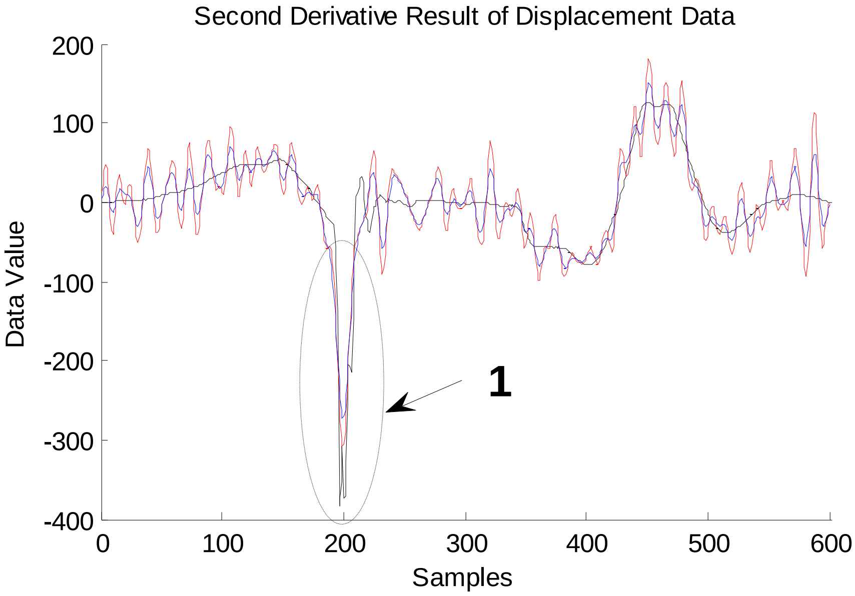


Figure 2.2.14: (Iterated Mean Filtering). The Comparative Second Derivative Results of Smoothed Displacement Data and Original Acceleration Data.

Black Line: Original Acceleration Data

Red Line: Second Derivative of Smoothed Displacement Data by using the mask 1 for 20 iterations

Blue Line: Second Derivative of Smoothed Displacement Data by using the mask 1 for 30 iterations

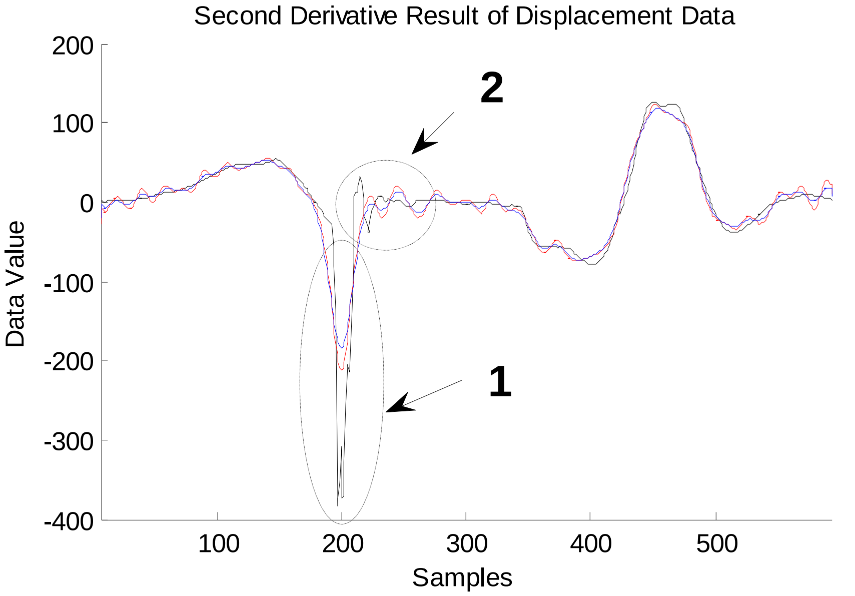


Figure 2.2.15: (Iterated Mean Filtering). The Comparative Second Derivative Results of Smoothed Displacement Data and Original Acceleration Data.

Black Line: Original Acceleration Data

Red Line: Second Derivative of Smoothed Displacement Data by using the mask 2 for 20 iterations

Blue Line: Second Derivative of Smoothed Displacement Data by using the mask 2 for 30 iterations

2.3 GAUSSIAN FILTER

Gaussian filtering is another linear smoothing method. Filter weights are taken from Gaussian distribution. The characteristic feature of this filter is that the data, which are near the center, is more effective than the data, which are far away [9].

Gaussian has the form:

$$g(x) = \frac{1}{\sqrt{2 \cdot \pi} \cdot \sigma} \cdot e^{-\frac{x^2}{2\sigma^2}} \tag{2.13}$$

In (2.13), σ is the Standard Deviation of the Gaussian distribution and x is the distance from the origin in the horizontal axis. Gaussian filter smoothes the signal by convolving the input data with a Gaussian function.

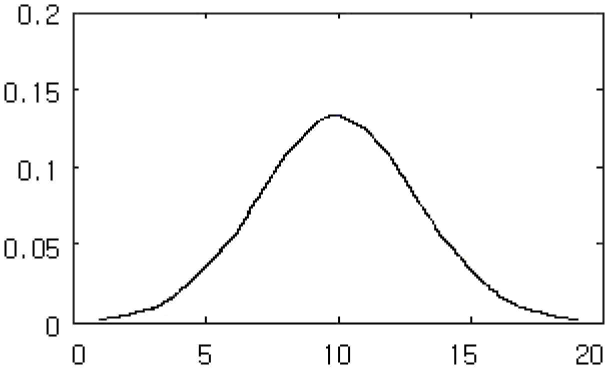


Figure 2.3.1: Gaussian distribution where σ =3, Size=6*σ+1

$$Kernel\_Size = 6 \cdot \sigma + 1 \tag{2.14}$$

Actually, Gaussian distribution behaves like weighted Mean filtering. The difference between them is that the weights on the Gaussian filter related to the filter size, which is in relation with the standard deviation, are calculated automatically. On the other hand, the weights on the Weighted Mean filter are defined manually. Larger standard deviations require larger convolution kernels. So, bigger standard deviation gives us smoother result.

2.3.1 Displacement Data Results for Gaussian Filtering

Results of the Gaussian filtering for various σ values can be seen in Figure 2.3.2 and Figure 2.3.3.

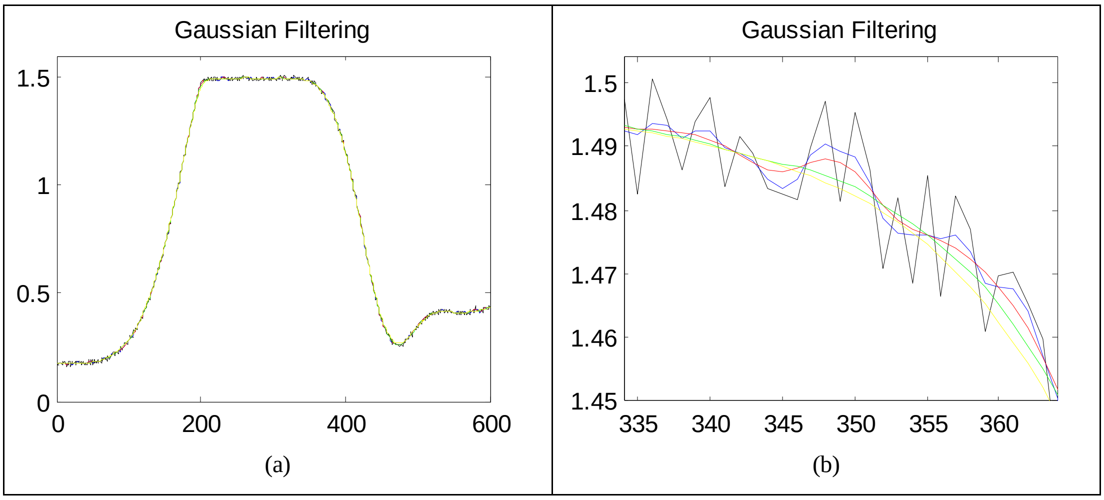


Figure 2.3.2: (Gaussian Filtering). (a) Displacement Data Results for σ=1, 2, 4, 6, size=6*σ+1, **(b)** The samples, between 333 and 364, of Figure 2.3.2.a.

Black Line: The Original Displacement Data.

Blue Line: Smoothed Displacement Data by using σ=1,

Red Line: Smoothed Displacement Data by using σ=2,

Green Line: Smoothed Displacement Data by using σ=4,

Yellow Line: Smoothed Displacement Data by using σ=6.

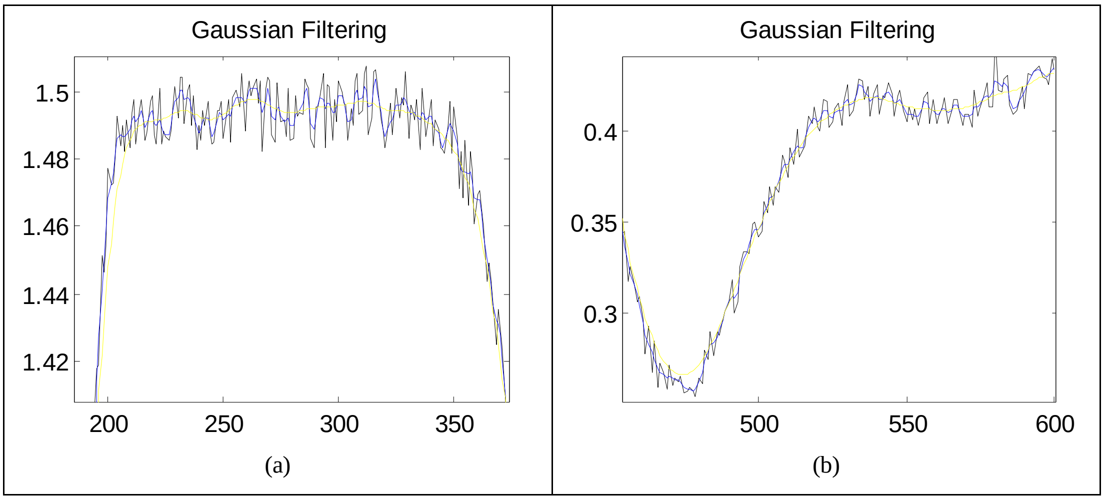


Figure 2.3.3: (Gaussian Filtering). (a) Displacement Data Results between 180 and 370 for σ=1, 6, size=6*σ+1, **(b)** Displacement Data Results between 460 and 600.

Black Line: The Original Displacement Data.

Blue Line: Smoothed Displacement Data by using σ=1,

Yellow Line: Smoothed Displacement Data by using σ=6.

2.3.2 Second Derivative Results for Gaussian Filtering

Second Derivative Result of Smoothed Displacement data by using Gaussian filter for various σ values can be seen in Figure 2.3.4 and Figure 2.3.5.

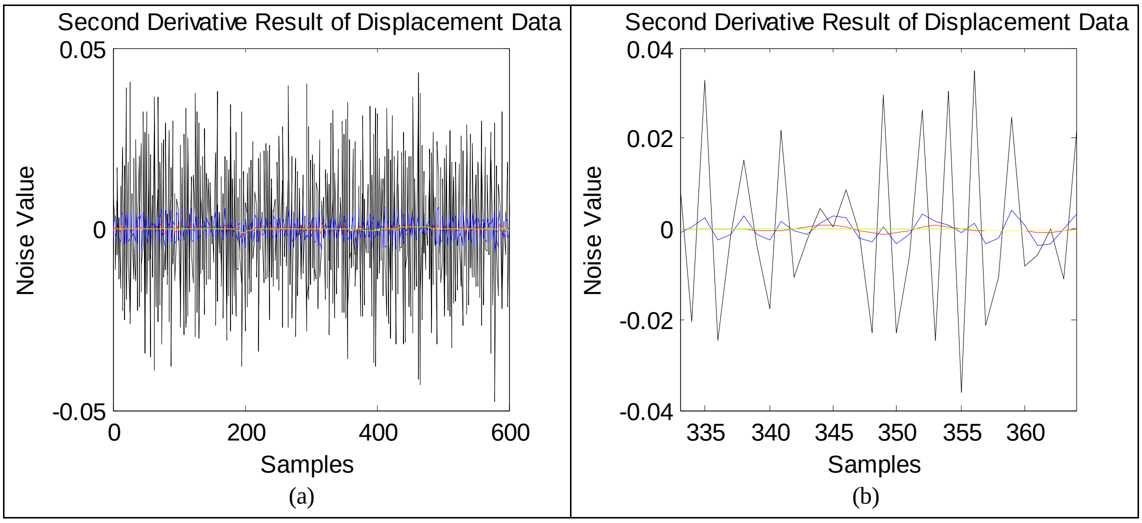


Figure 2.3.4: (Gaussian Filtering). (a) Second Derivative Result of Smoothed Displacement Data for σ=1, 2, 4, 6, size=6*σ+1, **(b)** The samples, between 333 and 364, of Figure 2.3.4.a.

Black Line: Second Derivative of the Original Displacement Data.

Blue Line: Second Derivative of the Data by using σ=1,

Red Line: Second Derivative of the Data by using σ=2,

Green Line: Second Derivative of the Data by using σ=4,

Yellow Line: Second Derivative of the Data by using σ=6,

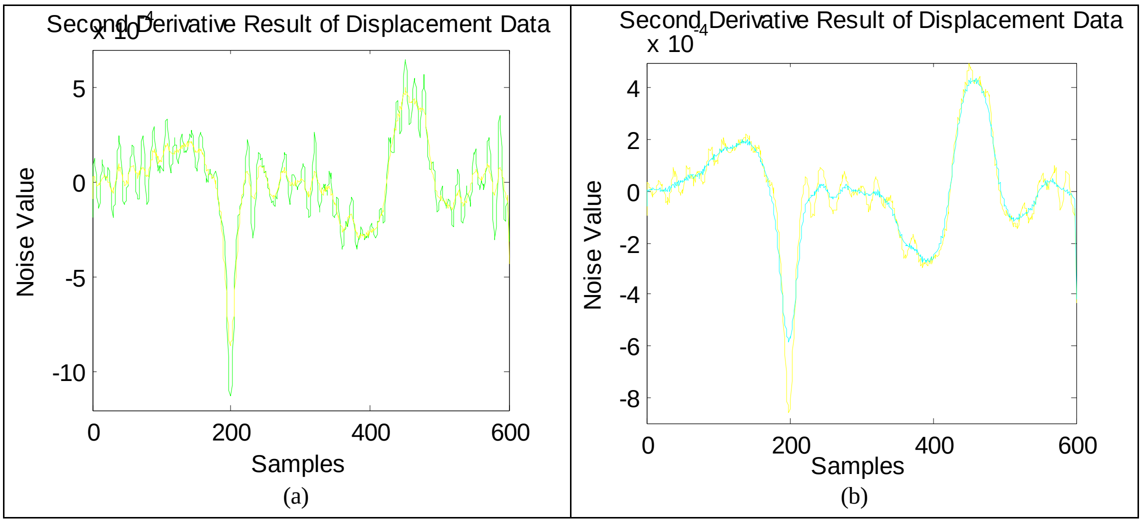


Figure 2.3.5: (Gaussian Filtering). (a) Second Derivative Result of Smoothed Displacement Data for σ=4, 6, size=6*σ+1, **(b)** Second Derivative Result of Smoothed Displacement Data for σ=6, 10 size=6*σ+1,

Green Line: Second Derivative of the Data by using σ=4,

Yellow Line: Second Derivative of the Data by using σ=6,

Cyan Line: Second Derivative of the Data by using σ=10,

2.3.3 Acceleration Data Results for Gaussian Filtering

After applying the Gaussian filter to the Acceleration data, in Figure 1.1.2.b, the following results are obtained.

Gauss Convolution results for various σ=1, 2, 3, 4 and size=6*σ+1 are seen in Figure 2.3.6. All the results are overlapped and seen in Figure 2.3.8. When the standard deviation is increased, the filter size also increases as in (2.14). As increase in filter size means that the data in the center are affected by the data which are far away, so our data will become smoother. In Figure 2.3.6 and in Figure 2.3.8, it is seen how the data become smoother when the standard deviation increases. In Figure 2.3.7, in order to see the difference in the data, when σ=1, 2, 3, 4 more clearly, the spaces between the samples of 195 and 215 are overlapped and showed with different colors, and in Figure 2.3.8.b the spaces between the samples of 185 and 230 are overlapped too.

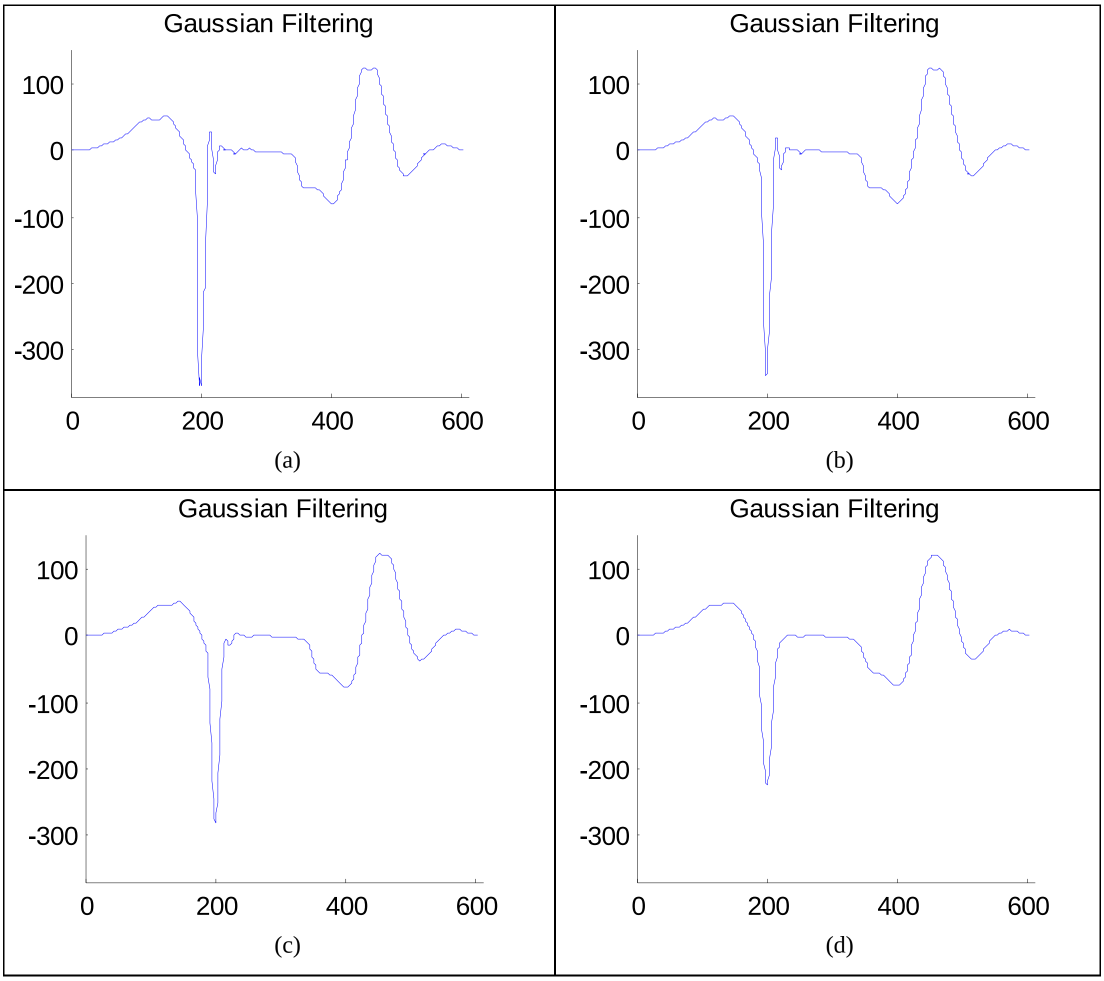


Figure 2.3.6: (Gaussian Filtering). Filter Size is calculated by using (2.14).
(a) σ = 1, **(b)** σ = 2, **(c)** σ = 4, **(d)** σ = 6.

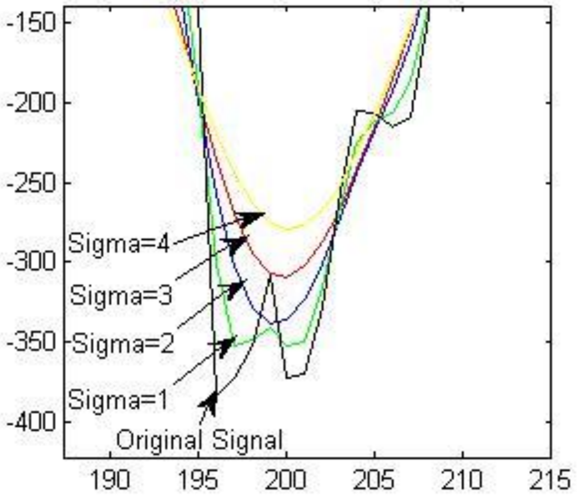
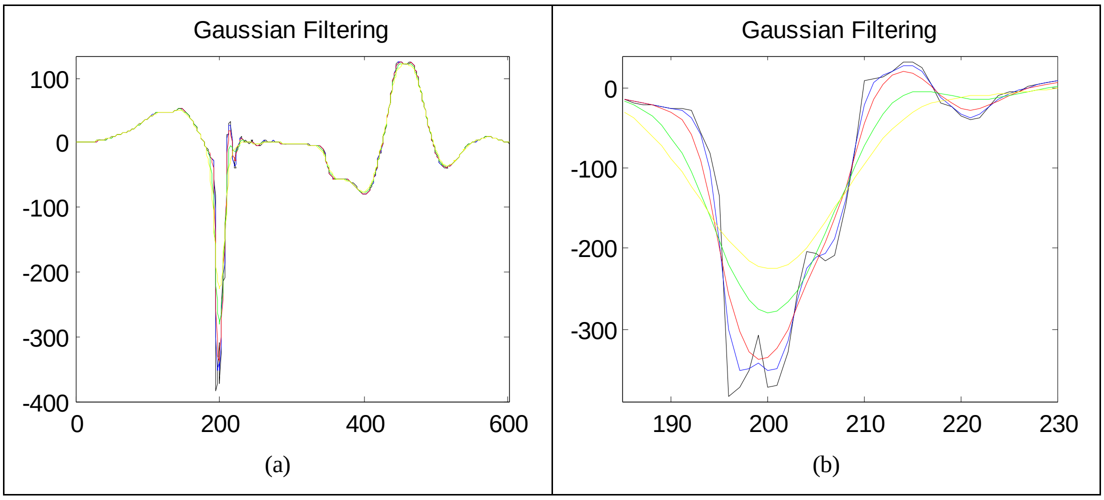


Figure 2.3.7: (Gaussian Filtering). Results between 186 and 215 for σ=1, 2, 3, 4, size=6*σ+1.



(a)

(b)

Figure 2.3.8: (Gaussian Filtering). (a) Acceleration Data Results for σ=1, 2, 4, 6, size=6*σ+1, **(b)** The samples, between 185 and 230, of Figure 2.3.8.a.

Black Line: The Original Acceleration Data.

Blue Line: Smoothed Acceleration Data by using σ=1,

Red Line: Smoothed Acceleration Data by using σ=2,

Green Line: Smoothed Acceleration Data by using σ=4,

Yellow Line: Smoothed Acceleration Data by using σ=6.

Fourier transform of a Gaussian function is also a Gaussian. This makes working on Gaussian function both in spatial and frequency domain possible by using the same function [9]. Gaussian filtering decreases high frequencies more than low frequency.

2.3.4 Comparing Gaussian Filters of Varying Standard Deviation

The Comparative Second Derivative Results of Smoothed Displacement data and Original Acceleration data are seen in Figure 2.3.9. Gaussian filtering result, seen by yellow line in Figure 2.3.9, gives comparatively better result in the area of 2 than the others, but the same line is not successful in the area of 1. The result of filtering, seen by green line in Figure 2.3.9, gives best result in the area of 1 while giving worst result in the area of 2. Also results of filtered Displacement data can also be seen in Figure 2.3.2 and in Figure 2.3.3 by using the same parameters.

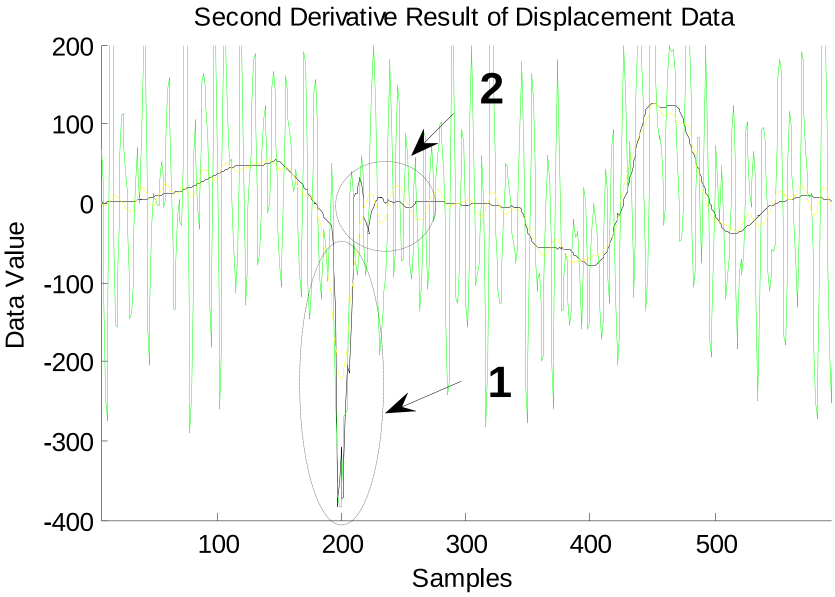


Figure 2.3.9: (Gaussian Filtering). The Comparative Second Derivative Results of Smoothed Displacement Data and Original Acceleration Data.

Black Line: Original Acceleration Data

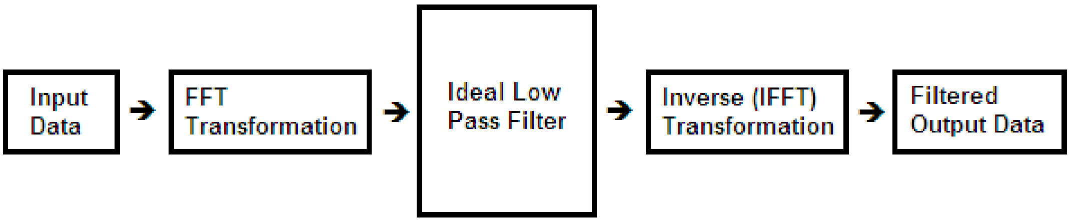
Green Line: Second Derivative of Smoothed Displacement Data by using σ=4, (size=6*σ+1)

Yellow Line: Second Derivative of Smoothed Displacement Data by using σ=6

2.4 IDEAL LOW PASS FILTER

Ideal Low Pass filter works on the frequency domain. This filter works like a smoothing filter. The Ideal Low Pass filter removes all the frequency components above a Cut off frequency which is defined by the user [9]. Removing frequency components above a Cut off frequency means removing sudden change beside the noise because noise and sudden change in the data correspond to high frequency components. Removing these high frequency components gives a result for smoothing the output data.

Table 2.4.1: Ideal Low Pass Filter



In Spatial Domain ⇔ **In Frequency Domain**

$$f(x) \quad \rightarrow \quad F(u) \qquad (2.15)$$

$$H(u) = \begin{cases} 1, & if \ \ D(u) \leq CutOff\_Frequency \\ 0, & if \ \ D(u) > CutOff\_Frequency \end{cases} \qquad (2.16)$$

$$D(u) = \sqrt{u^2} \qquad (2.17)$$

$$G(u) = F(u) \cdot H(u) \qquad (2.18)$$

$$g(x) \quad \leftarrow \quad G(u) \qquad (2.19)$$

$H(u)$ is the ideal filter and created by using (2.16).

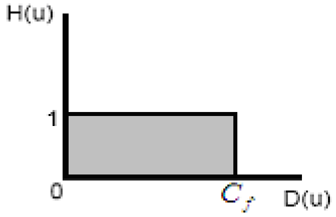


Figure 2.4.1: Ideal Low Pass Filter Cross Section

Because of working in frequency domain, Fourier transform of the input data given in (2.15) is taken. After that all frequency components above a Cut off frequency are removed and then Inverse Fourier transform is taken to get the result given in (2.19). Increasing the Cut off frequency means that more frequency components will pass, given in Figure 2.4.1. So that, more data in detail but less smoothed result will be obtained.

2.4.1 Golf Ball Data Results for Ideal Low Pass Filtering

After applying the Ideal Low Pass filter to the Golf ball data, in Figure 1.1.1, following results are obtained, in Figure 2.4.2 and 2.4.3.

To be able to realize the difference between figures, check the Tables 2.4.1.1, 2.4.1.2, 2.4.1.3, 2.4.1.4 and 2.4.1.5.

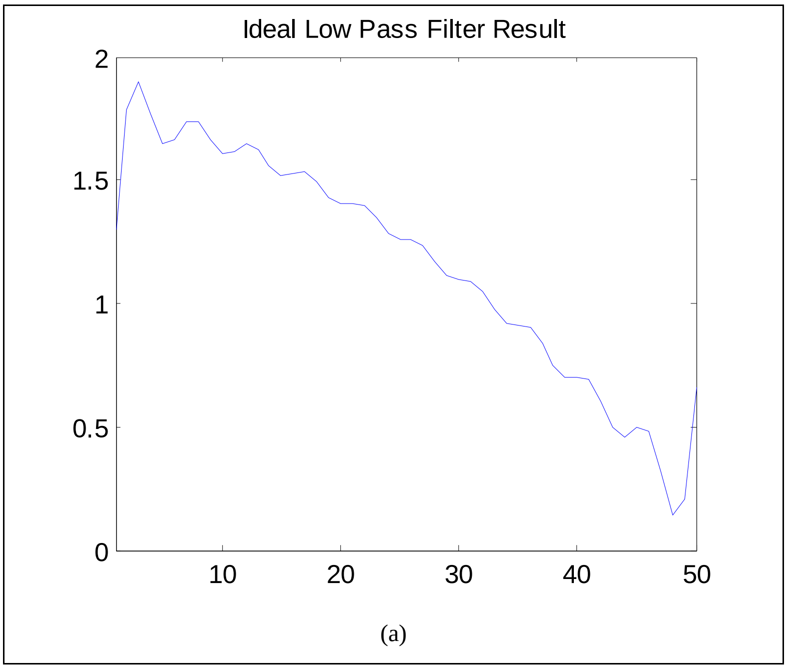
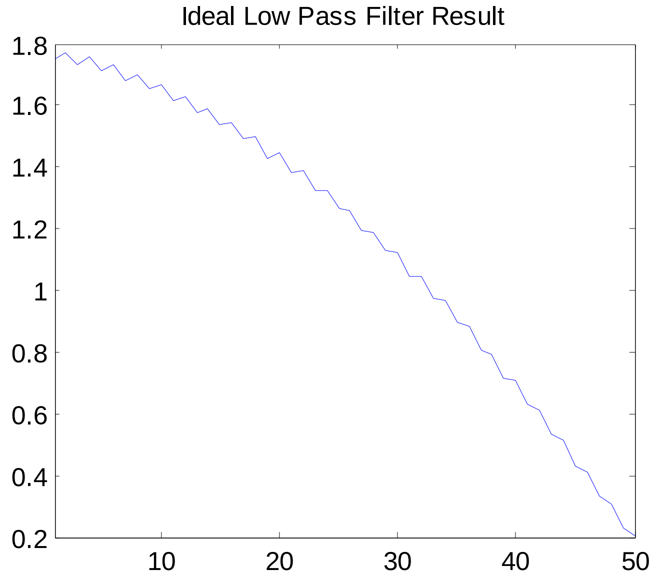


Figure 2.4.2: (Ideal Low Pass Filtering). (a) Smoothed Golf Data by using the Cut off Frequency=10, **(b)** Smoothed Golf Data by using the Cut off Frequency=24.



(b)

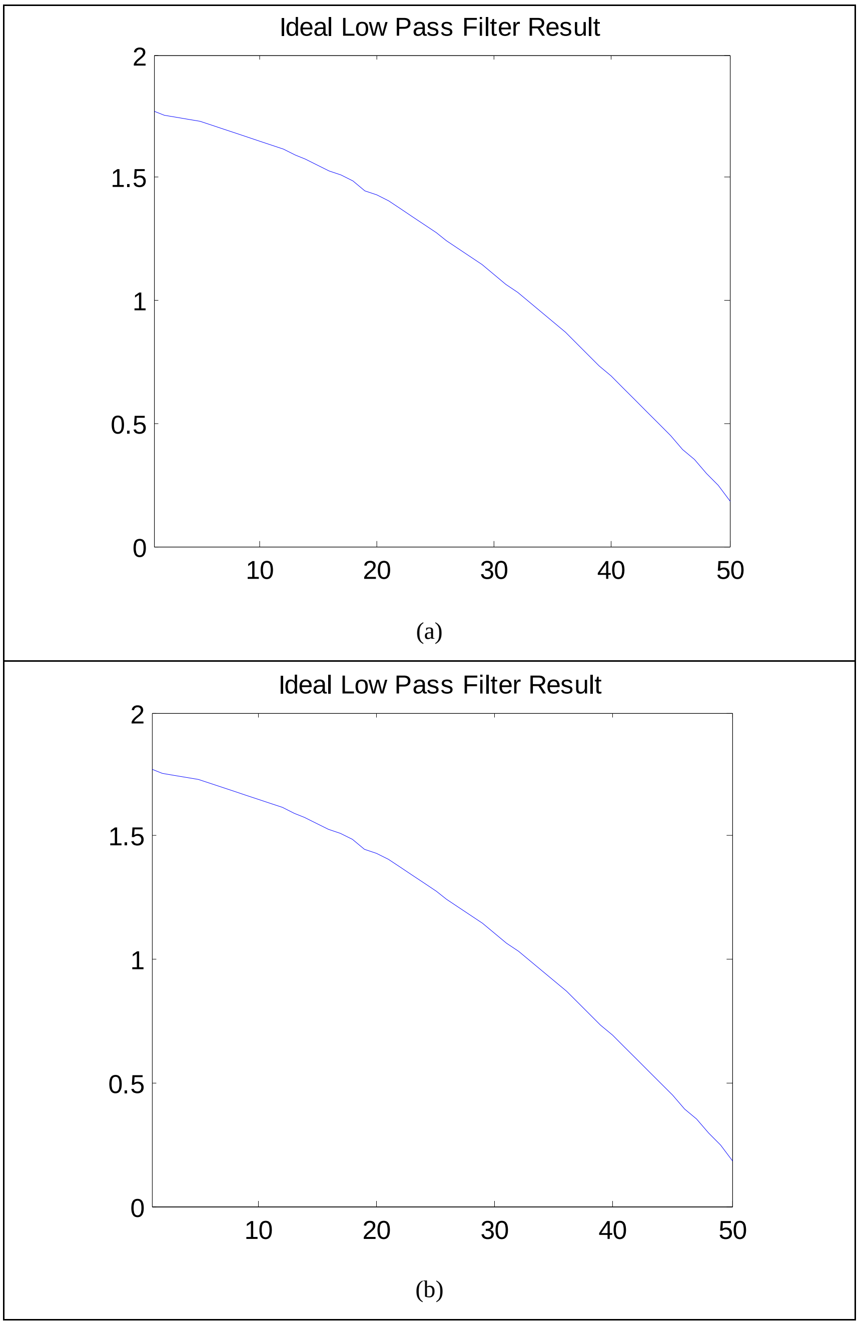
Figure 2.4.2: (Continued)

Table 2.4.1.1: Smoothed Golf Data after applying ILP filter by using Cut off Frequency=10

Height (meters)	**:**	...	1,6472	1,6641	1,7361	1,7358	1,6601	1,6045	...
Sample #	**:**	...	5	6	7	8	9	10	...

Table 2.4.1.2: Smoothed Golf Data after applying ILP filter by using Cut off Frequency=24

Height (meters)	**:**	...	1,7101	1,7309	1,6821	1,6989	1,6511	1,6669	...
Sample #	**:**	...	5	6	7	8	9	10	...



(a)

(b)

Figure 2.4.3: (Ideal Low Pass Filtering). (a) Smoothed Golf Data by using the Cut off Frequency=25, **(b)** Smoothed Golf Data by using the Cut off Frequency=40

Table 2.4.1.3: Smoothed Golf Data after applying ILP filter by using Cut off Frequency=25

Height (meters)	**:**	...	1,7260	1,7150	1,6980	1,6830	1,6670	1,6510	...
Sample #	**:**	...	5	6	7	8	9	10	...

Table 2.4.1.4: Smoothed Golf Data after applying ILP filter by using Cut off Frequency=40

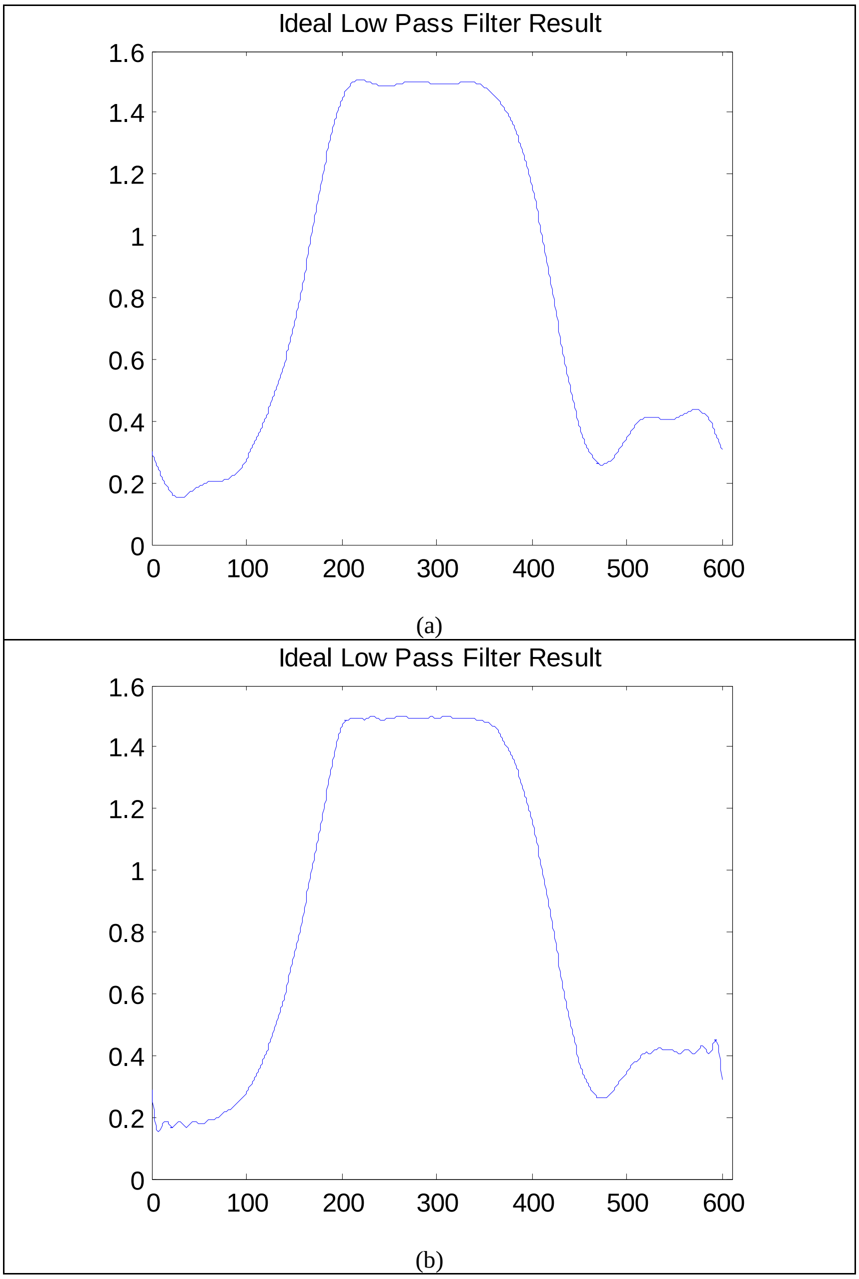
Height (meters)	**:**	...	1,7260	1,7150	1,6980	1,6830	1,6670	1,6510	...
Sample #	**:**	...	5	6	7	8	9	10	...

2.4.2 Displacement Data Results for Ideal Low Pass Filtering

After applying the Ideal Low Pass filter to the Displacement data, in Figure 1.1.2.a, following results are obtained, in Figure 2.4.4 and in Tables 2.4.2.2, 2.4.2.3, 2.4.2.4.

Table 2.4.2.1: Original Displacement Data Values

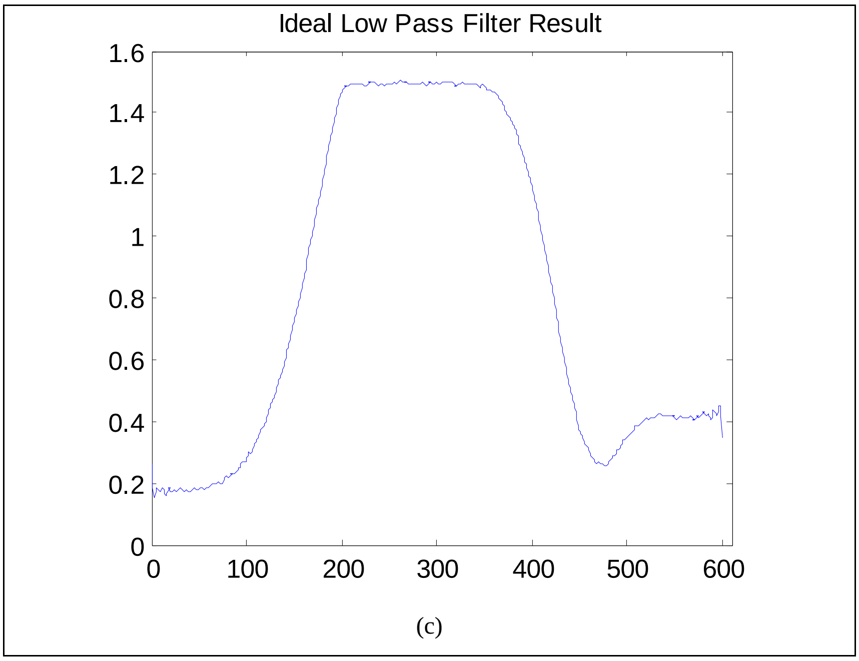
Height (meters)	**:**	...	1.482	1.4685	1.4854	1.4664	1.4823	1.4769	...
Sample #	**:**	...	353	354	355	356	357	358	...



(a)

(b)

Figure 2.4.4: (Ideal Low Pass Filtering). (a) Smoothed Displacement Data by using the Cut off Frequency=10, **(b)** Smoothed Displacement Data by using the Cut off Frequency=40, **(c)** Smoothed Displacement Data by using the Cut off Frequency=100.



(c)

Figure 2.4.4: (Continued).

Table 2.4.2.2: Smoothed Displacement Data after applying ILP filter by using Cut off Freq.=10

Height (meters)	**:**	...	1,4790	1,4766	1,4741	1,4715	1,4688	1,4659	...
Sample #	**:**	...	353	354	355	356	357	358	...

Table 2.4.2.3: Smoothed Displacement Data after applying ILP filter by using Cut off Freq.=40

Height (meters)	**:**	...	1,4807	1,4795	1,4781	1,4765	1,4746	1,4725	...
Sample #	**:**	...	353	354	355	356	357	358	...

Table 2.4.2.4: Smoothed Displacement Data after applying ILP filter by using Cut off Freq.=100

Height (meters)	**:**	...	1,4756	1,4755	1,4765	1,4768	1,4755	1,4729	...
Sample #	**:**	...	353	354	355	356	357	358	...

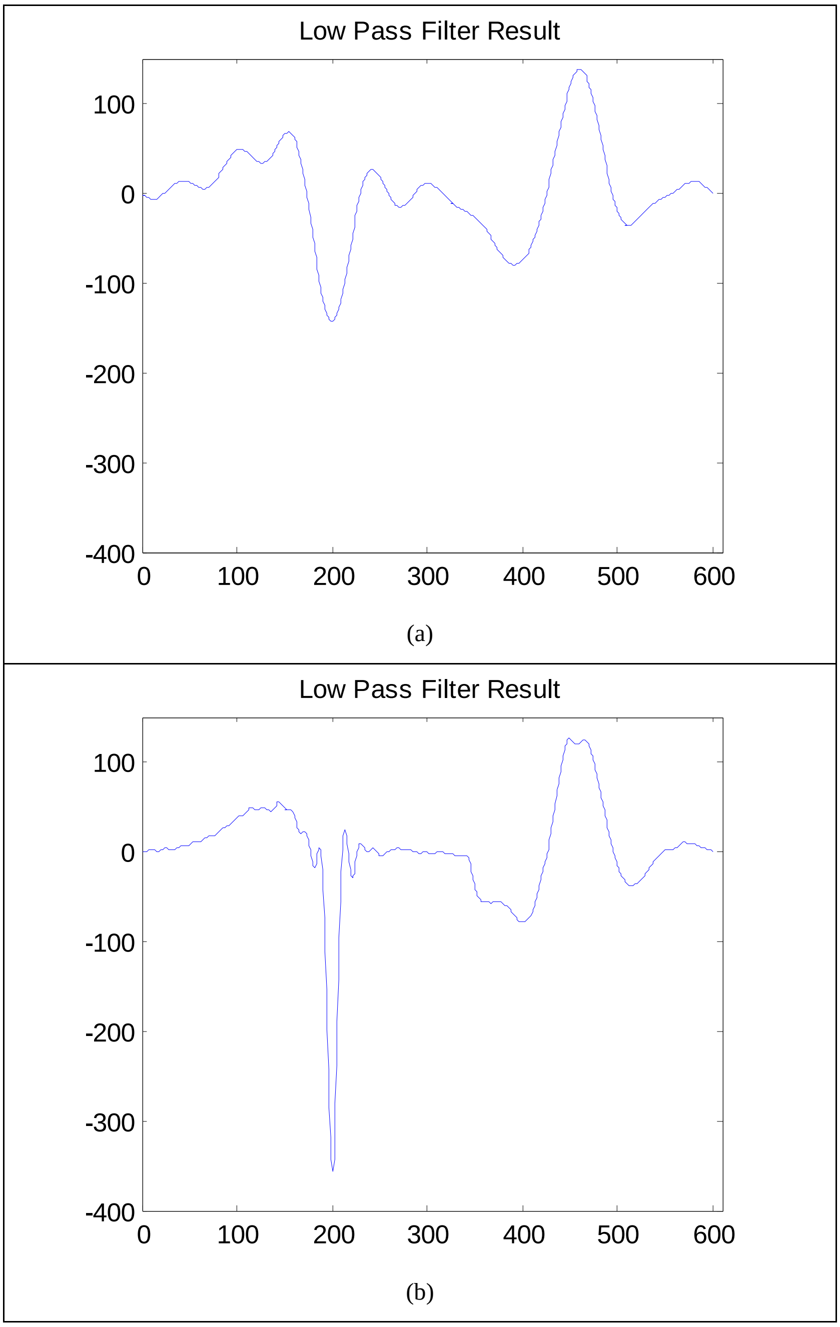
2.4.3 Acceleration Data Results for Ideal Low Pass Filtering

After applying the Ideal Low Pass filter to the Acceleration data, in Figure 1.1.2.b, results are obtained in Figure 2.4.5, 2.4.6.

Table 2.4.3.1: Original Acceleration Data Values

Height (meters)	**:**	...	-135.4	-383.5	-372.8	-352.2	-308.4	-372.4	...
Sample #	**:**	...	195	196	197	198	199	200	...

As it is seen from Figure 2.4.3, after a definite Cut off frequency which is 25 for Golf ball data, actually, the result does not change because this data have not so much noise. On the other hand, Acceleration data affected these Cut off frequency values severely because the intensity of fluctuation in Acceleration data is higher than the other 2 datasets as in Figure 2.4.5 and Figure 2.4.6.



Low Pass Filter Result

(a)

Low Pass Filter Result

(b)

Figure 2.4.5: (Ideal Low Pass Filtering). (a) Smoothed Acceleration Data by using the Cut off Frequency=10, **(b)** Smoothed Acceleration Data by using the Cut off Frequency=40.

Table 2.4.3.2: Smoothed Acceleration Data after applying ILP filter by using Cut off Freq.=10

Height (meters)	**:**	...	-136,53	-139,06	-141,05	-142,47	-143,33	-143,62	...
Sample #	**:**	...	195	196	197	198	199	200	...

Table 2.4.3.3: Smoothed Acceleration Data after applying ILP filter by using Cut off Freq.=40

Height (meters)	**:**	...	-198,71	-243,57	-284,71	-318,92	-343,42	-356,11	...
Sample #	**:**	...	195	196	197	198	199	200	...

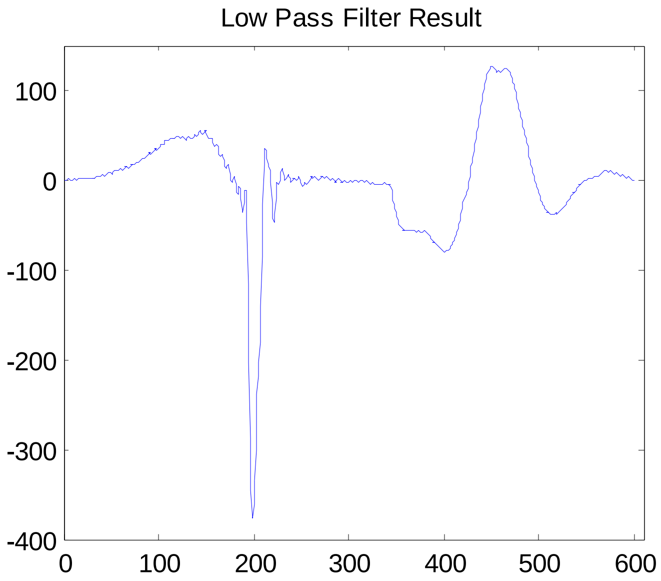


Figure 2.4.6: (Ideal Low Pass Filtering). Smoothed Acceleration Data by using the Cut off Frequency=100.

Table 2.4.3.4: Smoothed Acceleration Data after applying ILP filter by using Cut off Freq.=100

Height (meters)	**:**	...	-203,78	-287,03	-346,13	-374,62	-376,91	-361,17	...
Sample #	**:**	...	195	196	197	198	199	200	...

2.4.4 Second Derivative Results for Ideal Low Pass Filtering

The Second Derivative Result of the Smoothed Displacement data can be seen in Figure 2.4.7 and Figure 2.4.8 by using the Ideal Low Pass filter for various Cut off Frequencies.

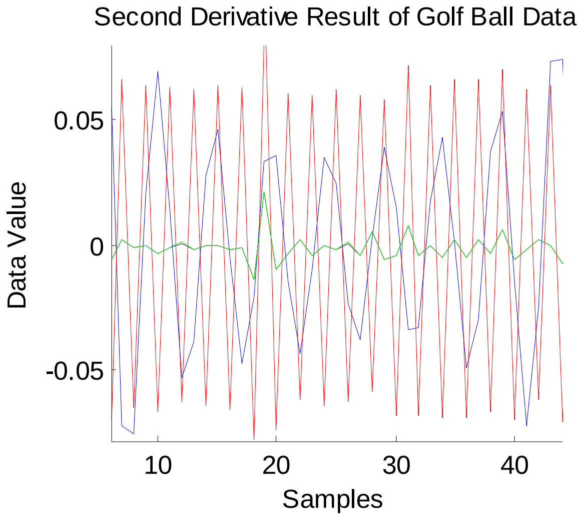


Figure 2.4.7: (Ideal Low Pass Filtering). Second Derivative Result of Smoothed **Golf Ball Data** using Cut off Frequency=10, 24 and 25.

Black Line: Second Derivative of the Original Golf Ball Drop Data.

Blue Line: Second Derivative of the Data by using the Cut off Frequency=10,

Red Line: Second Derivative of the Data by using the Cut off Frequency=24,

Green Line: Second Derivative of the Data by using the Cut off Frequency=25.

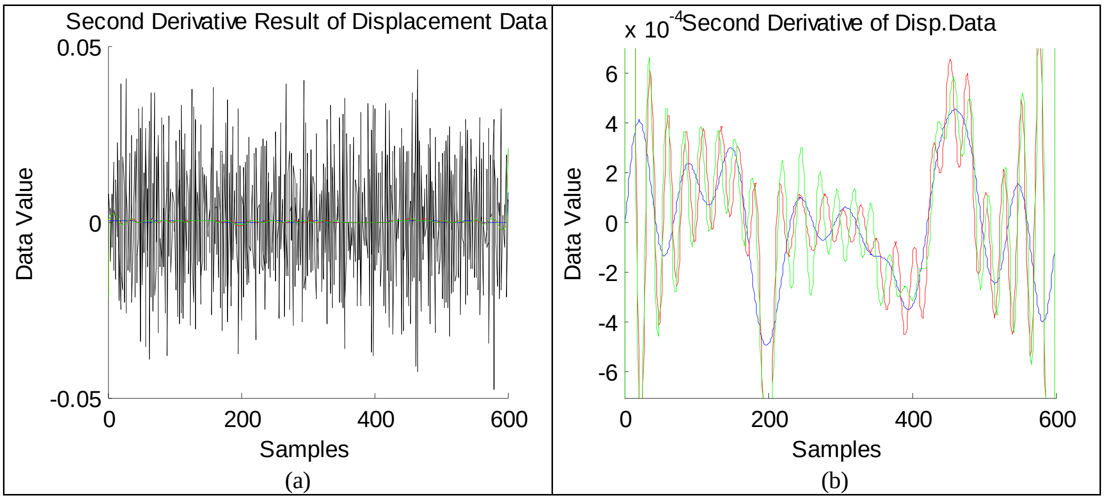


Figure 2.4.8: (Ideal Low Pass Filtering). (a) Second Derivative Result of Smoothed **Displacement Data** using Cut off Frequency=10, 24 and 25, **(b)** Same as Figure **2.4.8.**a except Original Displacement Data (Black Line).

Black Line: Second Derivative of the Original Displacement Data.

Blue Line: Second Derivative of the Data by using the Cut off Frequency=10,

Red Line: Second Derivative of the Data by using the Cut off Frequency=24,

Green Line: Second Derivative of the Data by using the Cut off Frequency=25.

2.4.5 Comparing Ideal Low Pass Filters of Varying Cut off Frequency

The Comparative Second Derivative Results of the Smoothed Displacement data and Original Acceleration data are seen in Figure 2.4.9. Ideal Low Pass filtering result, seen by red line in Figure 2.4.9, gives comparatively better result in the area of 1 than others, but the same line is not successful at any other part of the data.

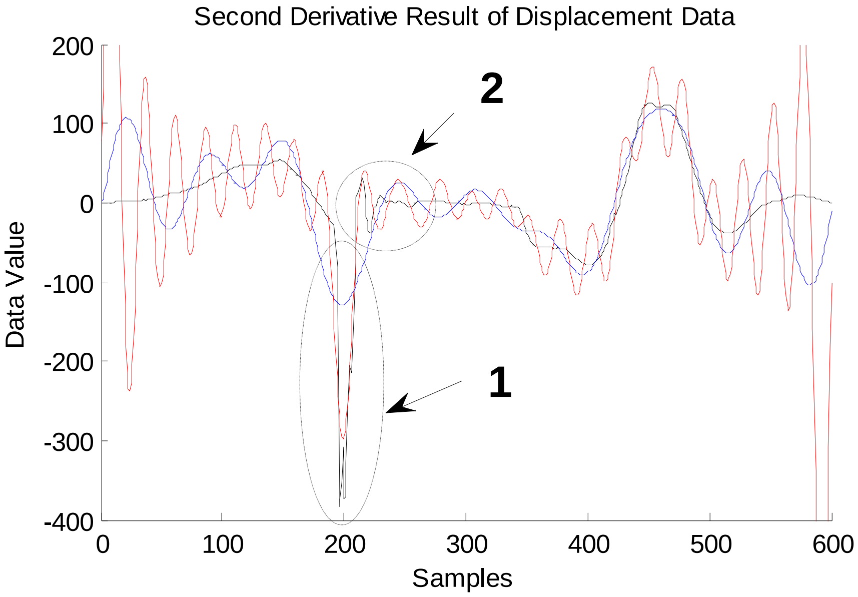


Figure 2.4.9: (Ideal Low Pass Filtering). The Comparative Second Derivative Results of Smoothed Displacement Data and Original Acceleration Data.

Black Line: Original Acceleration Data

Blue Line: Second Derivative of Smoothed Displacement Data by using the Cut off Frequency=10,

Red Line: Second Derivative of Smoothed Displacement Data by using the Cut off Frequency=24

2.5 BUTTERWORTH LOW PASS FILTER

Butterworth Low Pass filter works on the frequency domain like Ideal Low Pass filter. This filter works like a smoothing filter. The difference between the Ideal Low Pass filter and the Butterworth Low Pass filter is the order parameter. This order parameter decides the shape of the filter's cross section. For the large values of order, shape of the Butterworth Low Pass filter's cross section resembles to the shape of the Ideal Low Pass filter because increasing the order means that value of the filter, $H(u)$, converges to 1 or 0. However in the Ideal Low Pass filter, the value of $H(u)$ is 1 or 0 precisely, but the value of $H(u)$ is calculated in Butterworth Low Pass filter by using the formula in (2.20).

$$H(u) = \frac{1}{1 + (D(u) / D_0)^{2n}} \tag{2.20}$$

Decreasing the value of order parameter resembles to the Gaussian smoothing in spatial domain [9]. Ideal Low Pass filter removes all frequency components above a Cut off frequency. However in Butterworth Low Pass filter, neither any frequency components above a Cut off frequency are blocked, nor all frequency components below a Cut off frequency are passed.

Noise and sudden change in the data correspond to high frequency components. Removing these high frequency components gives a result for smoothing the output data.

In Spatial Domain ⇔ **In Frequency Domain**

$$f(x) \quad \rightarrow \quad F(u) \tag{2.21}$$

$$H(u) = \frac{1}{1 + (D(u) / D_0)^{2n}} \tag{2.22}$$

$$D(u) = \sqrt{u^2} \tag{2.23}$$

$$G(u) = F(u) \cdot H(u) \tag{2.24}$$

$$g(x) \quad \leftarrow \quad G(u) \tag{2.25}$$

n is the order parameters. $H(u)$ is the Butterworth filter and created by using (2.22).

Because of working in frequency domain, the Fourier transform of the input data is taken, (2.21). After removing some frequency components above a Cut off frequency, the Inverse Fourier transform is taken to get the result, (2.25). Increasing the Cut off frequency means more frequency components will pass, Figure 2.5.1, so that, more data in detail but less smoothed result will be obtained.

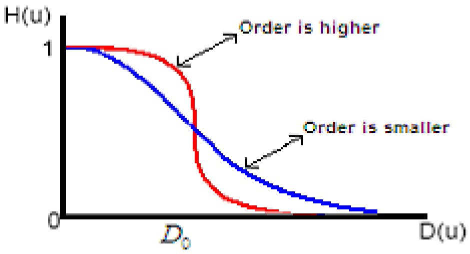
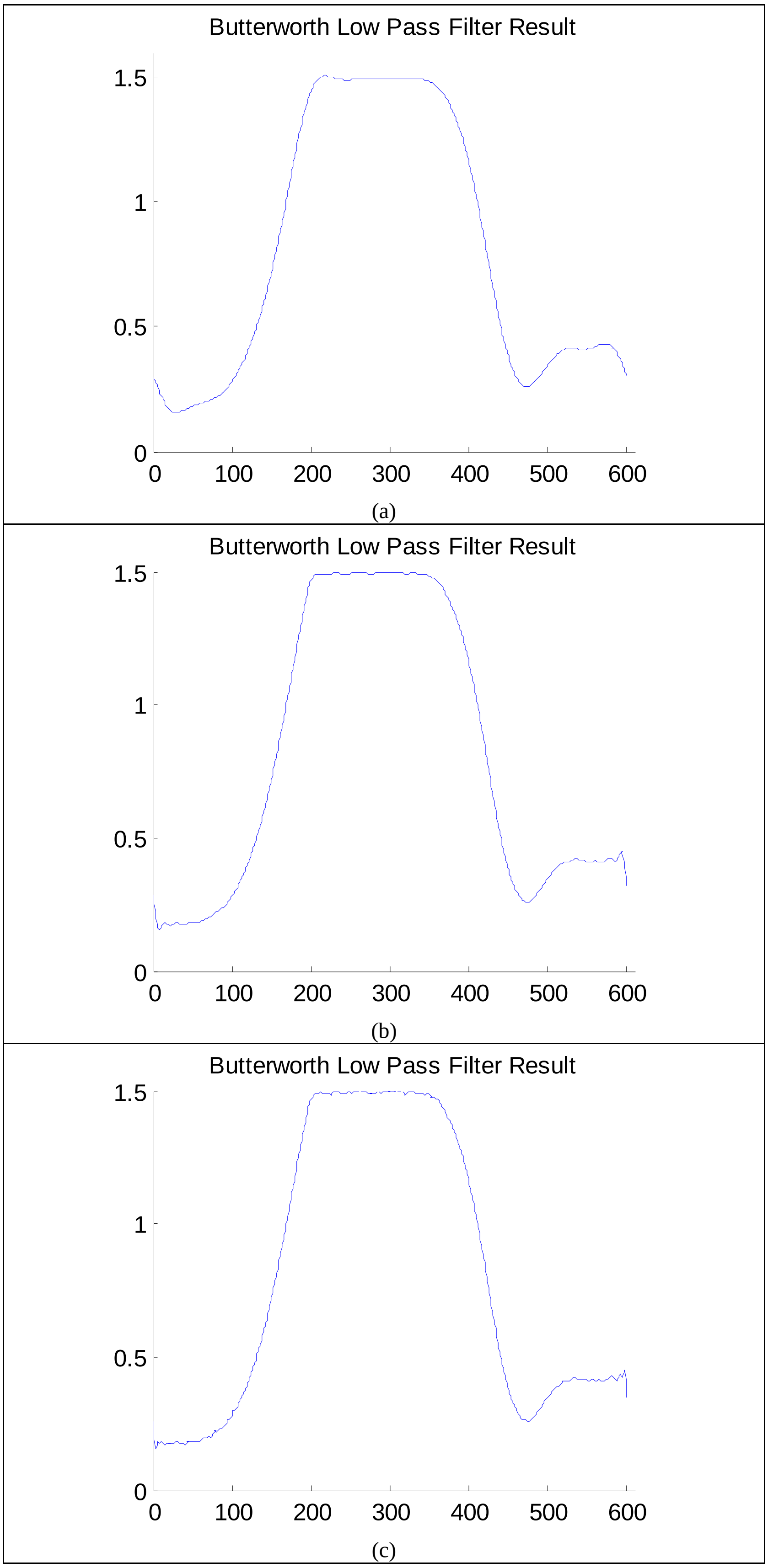


Figure 2.5.1: Butterworth Low Pass Filter Cross Section

By keeping constant the Cut off frequency and changing the order parameter, Butterworth Low Pass filter behaves like Ideal Low Pass filter for high values of order. For small values of order, it behaves like Gaussian filter.

2.5.1 Displacement Data Results for Butterworth Low Pass Filtering

After applying the Butterworth Low Pass filter to the Displacement data, in Figure 1.1.2.a, following results are obtained, in Figure 2.5.2 and in Tables 2.5.1.1, 2.5.1.2.



(a)

(b)

(c)

Figure 2.5.2: (Butterworth Low Pass Filtering). (a) Smoothed Displacement Data by using the Cut off Frequency=10 and Order=3, **(b)** Smoothed Displacement Data by using the Cut off Frequency=40 and Order=3, **(c)** Smoothed Displacement Data by using the Cut off Frequency=100 and Order=3.

Table 2.5.1.1: Smoothed Displacement Data after applying Butterworth Low Pass filter by using Cut off Frequency=10 and Order=3

Height (meters)	**:**	...	1,4805	1,4790	1,4775	1,4759	1,4741	1,4722	...
Sample #	**:**	...	353	354	355	356	357	358	...

Table 2.5.1.2: Smoothed Displacement Data after applying Butterworth Low Pass filter by using Cut off Frequency=100 and Order=3

Height (meters)	**:**	...	1,4756	1,4753	1,4762	1,4767	1,4754	1,4729	...
Sample #	**:**	...	353	354	355	356	357	358	...

After keeping the value of Cut off frequency constant and increasing the order parameter by taking it to infinity, Butterworth Low Pass filter starts to behave like Ideal Low Pass filter, as in Figure 2.5.3.

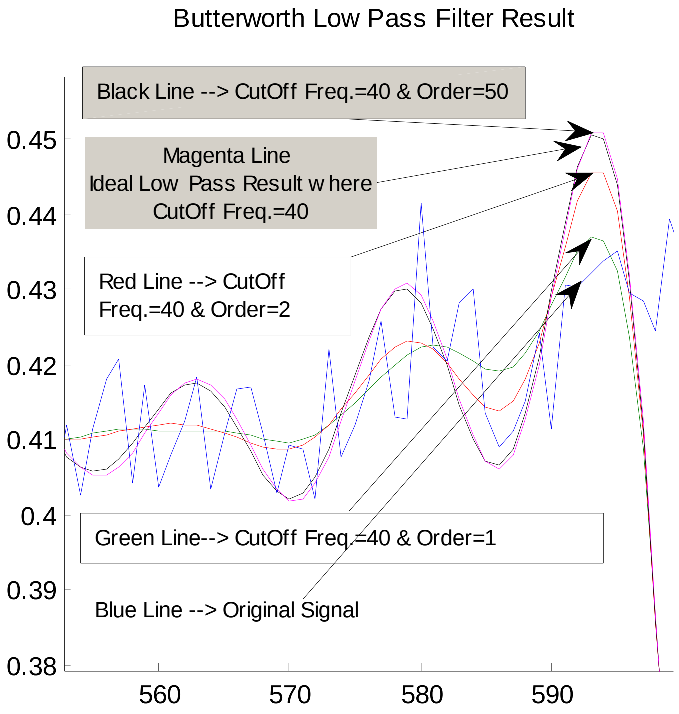
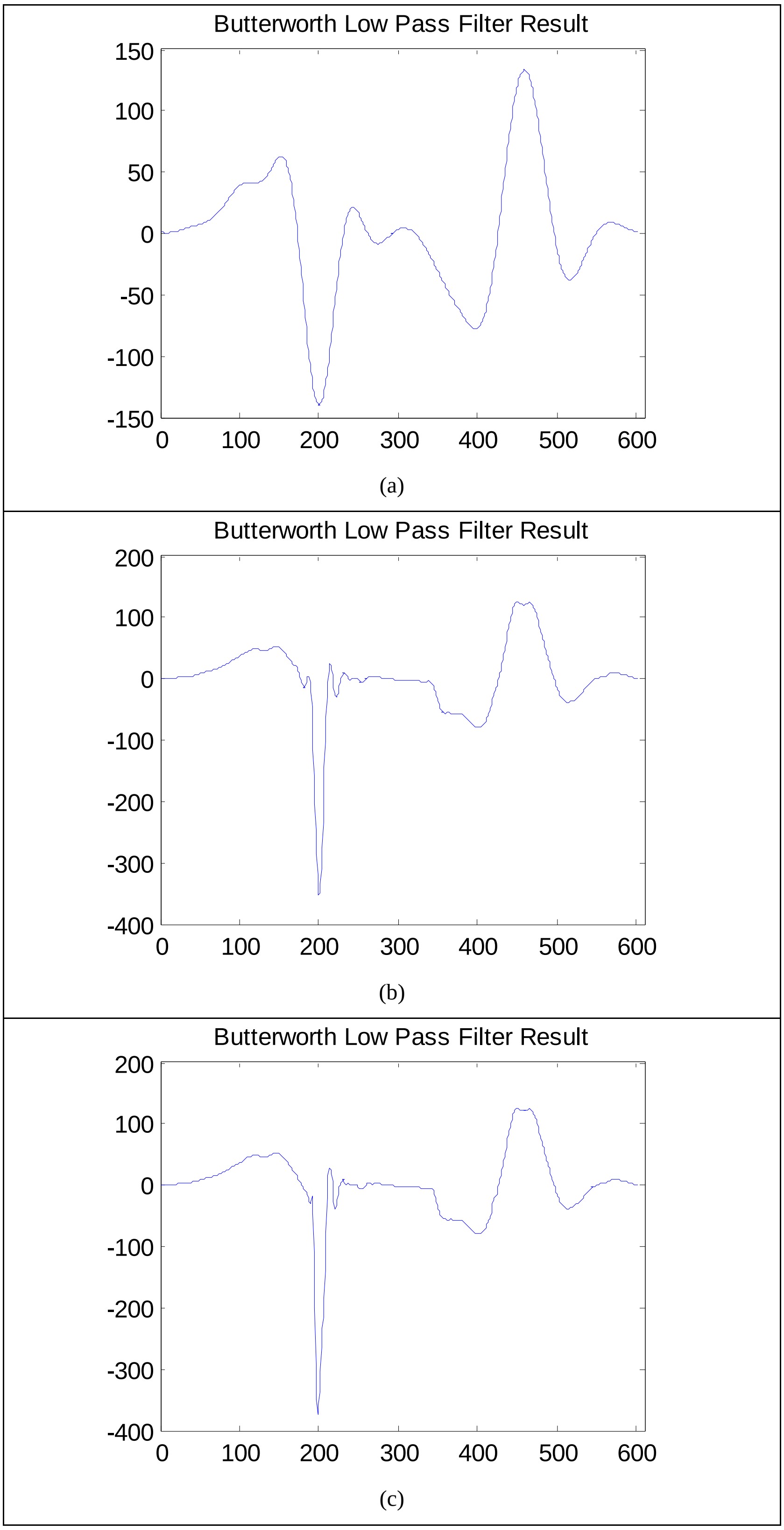


Figure 2.5.3: (Butterworth Low Pass Filtering). Blue Line is the Original input data. Green Line, Red Line and Black Line are the Smoothed Displacement Data by using the Cut off Frequency is 40 and Orders are 1, 2 and 50. Magenta Line is the Ideal Low Pass Filter Result where Cut off Frequency=40.

2.5.2 Acceleration Data Results for Butterworth Low Pass Filtering

Results of Butterworth Low Pass filter by using various Cut off Frequencies are in Figure 2.5.4, 2.5.5 and in Tables 2.5.2.1, 2.5.2.2, 2.5.2.3.



(a)

(b)

(c)

Figure 2.5.4: (Butterworth Low Pass Filtering).

(a) Smoothed Acceleration Data by using the Cut off Frequency=10 and Order=3,
(b) Smoothed Acceleration Data by using the Cut off Frequency=40 and Order=3,
(c) Smoothed Acceleration Data by using the Cut off Frequency=100 and Order=3.

Table 2.5.2.1: Smoothed Acceleration Data after applying Butterworth Low Pass filter by using Cut off Frequency=10 and Order=3

Height (meters)	**:**	...	-132,29	-134,83	-136,84	-138,32	-139,24	-139,61	...
Sample #	**:**	...	195	196	197	198	199	200	...

Table 2.5.2.2: Smoothed Acceleration Data after applying Butterworth Low Pass filter by using Cut off Frequency=40 and Order=3

Height (meters)	**:**	...	-202,62	-246,74	-286,47	-318,73	-340,97	-351,50	...
Sample #	**:**	...	195	196	197	198	199	200	...

Table 2.5.2.3: Smoothed Acceleration Data after applying Butterworth Low Pass filter by using Cut off Frequency=100 and Order=3

Height (meters)	**:**	...	-201,74	-291,15	-350,59	-373,42	-371,87	-359,14	...
Sample #	**:**	...	195	196	197	198	199	200	...

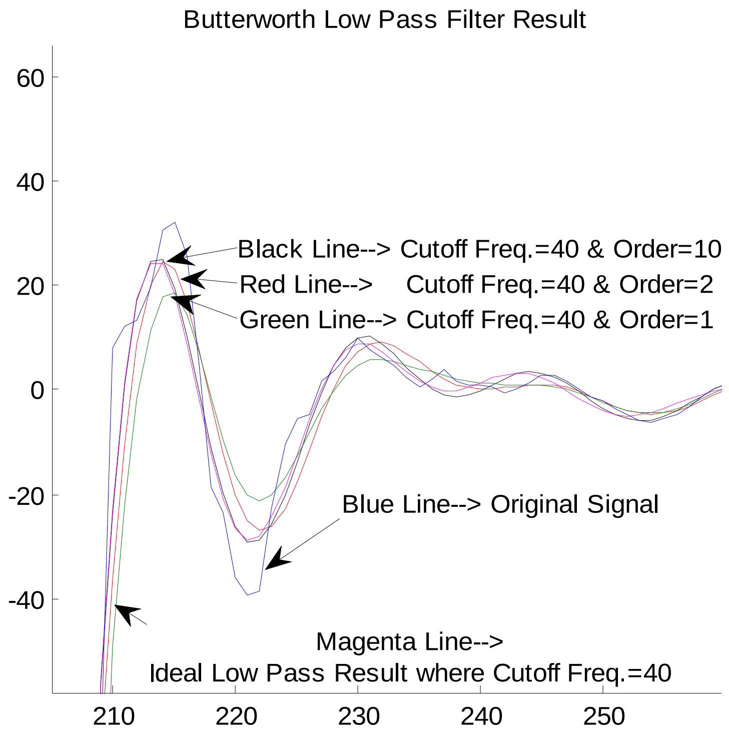
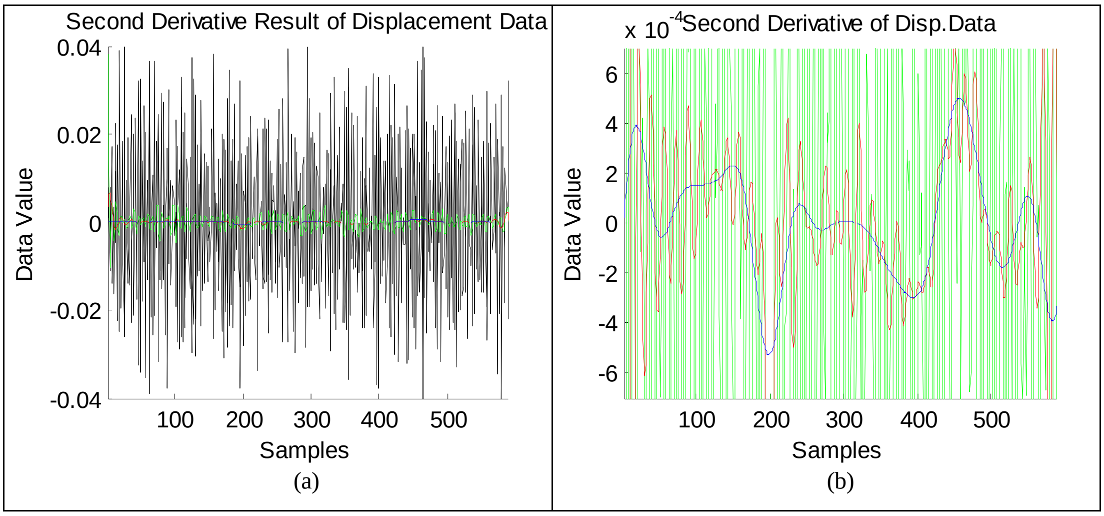


Figure 2.5.5: (Butterworth Low Pass Filtering). <u>Zoom in</u> Smoothed Acceleration Data by using Cut off Frequency=40 and Order=1,2 and 10.

2.5.3 Second Derivative Results for Butterworth Low Pass Filtering

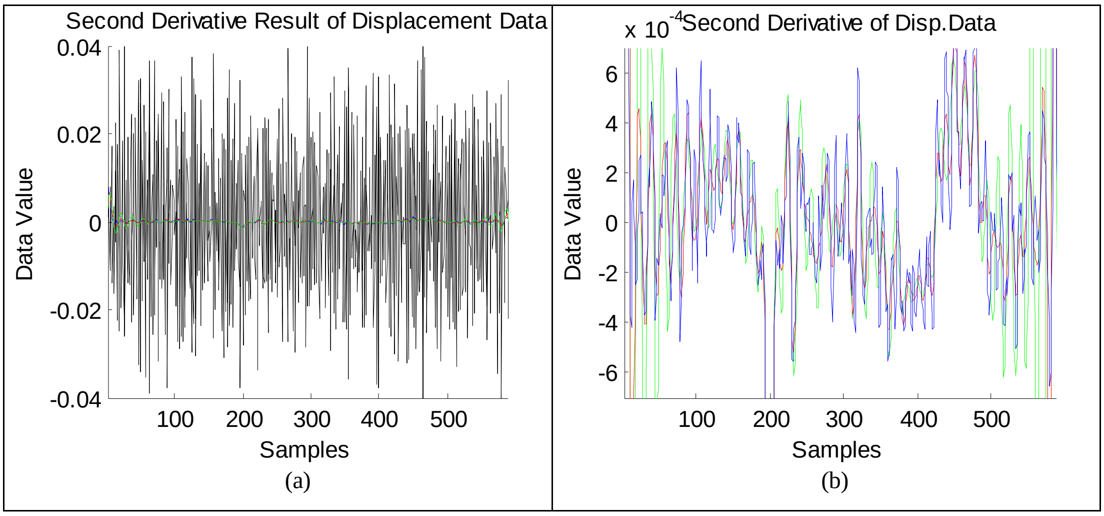
Second Derivative Results of Smoothed Displacement data can be seen in Figure 2.5.6 and Figure 2.5.7 by using the Butterworth Low Pass filtering for various Cut off Frequency and order parameters.



(a) (b)

Figure 2.5.6: (Butterworth Low Pass Filtering). (a) Second Derivative Result of Smoothed **Displacement Data** using Cut off Frequency=10, 40, 100, and Order=3 **(b)** Same as Figure 2.5.6.a except Original Displacement Data (Black Line).

Black Line: Second Derivative of the Original Displacement Data.
Blue Line: Second Derivative of the Data by using the Cut off Frequency=10 and Order=3,
Red Line: Second Derivative of the Data by using the Cut off Frequency=40 and Order=3,
Green Line: Second Derivative of the Data by using the Cut off Frequency=100 and Order=3.



(a) (b)

Figure 2.5.7: (Butterworth Low Pass Filtering). (a) Second Derivative Result of Smoothed **Displacement Data** using Cut off Frequency=40, and Order=1, 2, 50 **(b)** Same as Figure 2.5.7.a except Original Displacement Data (Black Line).

Black Line: Second Derivative of the Original Displacement Data.
Blue Line: Second Derivative of the Data by using the Cut off Frequency=40 and Order=1,
Red Line: Second Derivative of the Data by using the Cut off Frequency=40 and Order=2,
Green Line: Second Derivative of the Data by using the Cut off Frequency=40 and Order=50.

2.5.4 Comparing Butterworth Low Pass Filters of Varying Cut off and Order Parameters

The Comparative Second Derivative Results of Smoothed Displacement data and Original Acceleration data are seen in Figure 2.5.8. Butterworth Low Pass filtering result, blue line, gives comparatively better result in the area of 2, but the same line is not successful in the area of 1. The result of filtering, red line, gives good result in the area of 1 while giving bad result in the area of 2. Also the results of the filtered Displacement data can also be seen in Figure 2.5.2 by using the same parameters.

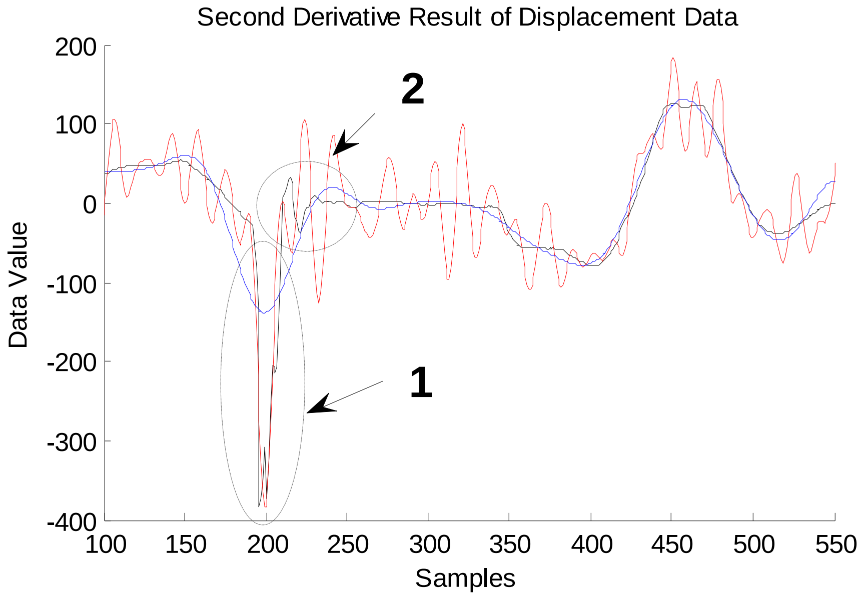


Figure 2.5.8: (Butterworth Low Pass Filtering). The Comparative Second Derivative Results of Smoothed Displacement Data and Original Acceleration Data.

Black Line : Original Acceleration Data

Blue Line : Second Derivative of Smoothed Displacement Data by using the Cut off Frequency=10 and Order=3,

Red Line : Second Derivative of Smoothed Displacement Data by using the Cut off Frequency=40 and Order=3.

2.6 LINEAR DIFFUSION FILTER

Diffusion is a physical process which balances the concentration difference without destroying the data. Smoothing process can be considered as a diffusion process, which decreases the number of maxima and minima in the data without increasing the difference between maxima and minima. Diffusion process equation can be considered in one dimension as [15]:

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}(c.\frac{\partial}{\partial x}u) \tag{2.26}$$

Where **c** is the diffusivity control and **u** is the input data.

$$\frac{\partial u}{\partial t} = c.(\frac{\partial^2 u}{\partial x^2}) \tag{2.27}$$

Explicit scheme discretization is,

$$\frac{\partial u}{\partial t} \cong \frac{u(x,t+\Delta t)-u(x,t)}{\Delta t} \tag{2.28}$$

$$\frac{\partial u}{\partial t} \cong \frac{u^{t+1}[x]-u^{t}[x]}{\Delta t} \tag{2.29}$$

$$\frac{u^{t+1}[x]-u^{t}[x]}{\Delta t} = c\cdot(u[x+1]+u[x-1]-2u[x]) \tag{2.30}$$

Partial differential equation of linear diffusion equation, calculated from (2.30), is:

$$u^{t+1}[x] = (c.\Delta t)\cdot(u^{t}[x+1]+u^{t}[x-1])-((2c\cdot\Delta t)\cdot u^{t}[x])+u^{t}[x] \tag{2.31}$$

$$u^{t+1}[x] = (c.\Delta t)\cdot(u^{t}[x+1]+u^{t}[x-1])+(1-(2c.\Delta t))\cdot u^{t}[x] \tag{2.32}$$

$u^{t}[x]$ is the value of input data at the location of x at time t.

Δt is the time step and c is a constant which controls the diffusivity. In my experiments, the value of c is assumed as 1.

In Figure 2.6.1, Figure 2.6.2 and Figure 2.6.3, there are nearly no change by keeping T constant and changing n and Δt accordingly.

$$n=T/\Delta t \tag{2.33}$$

2.6.1 Acceleration Data Results for Linear Diffusion Filtering

Smoothing is applied to Acceleration Data to see the smoothing effect of Linear Diffusion Equation. The result of the change in T can be seen in Figure 2.6.4 while keeping Δt constant. The output changes when T is increased then accordingly the smoothing effect increases.

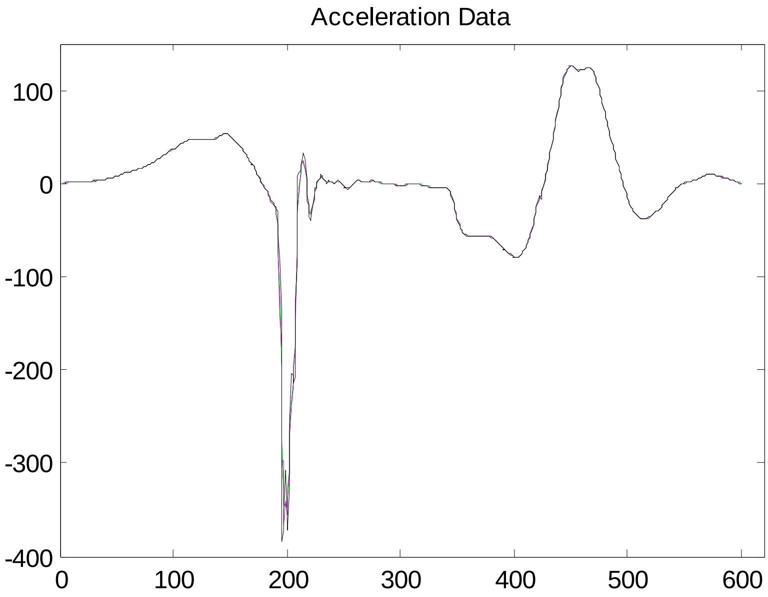


Figure 2.6.1: (Linear Diffusion Filtering). T=1 & Δt=0.01, 0.1, 0.2, 0.5

Black Line: The Original Data. Red Line: T=1, Δt=0.01, n=100 Magenta Line: T=1, *Δt=0.50*, n=2
Blue Line: T=1, Δt=0.10, n=10 Green Line: T=1, Δt=0.20, n=5

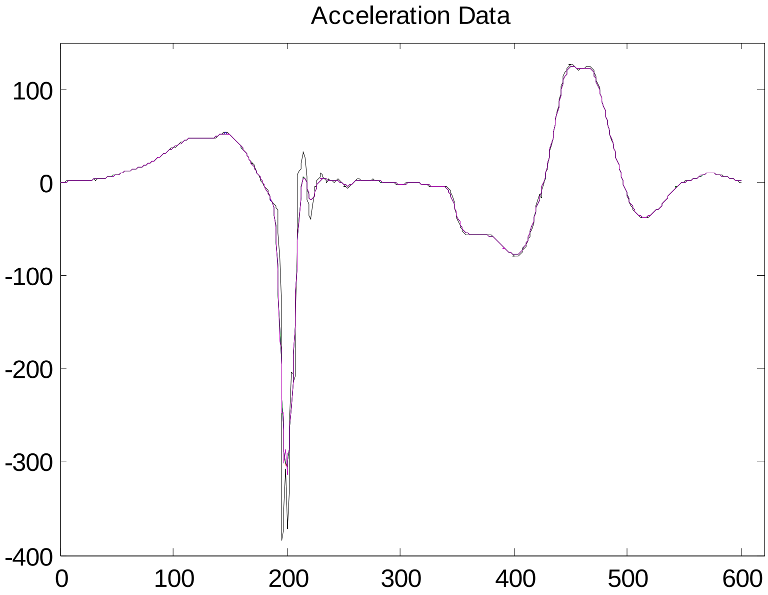


Figure 2.6.2: (Linear Diffusion Filtering). T=5 & Δt=0.01, 0.1, 0.2, 0.5

Black Line: The Original Data. Red Line: T=5, Δt=0.01, n=500. Magenta Line: T=5, *Δt=0.50*, n=10
Blue Line: T=5, Δt=0.10, n=50. Green Line: T=5, Δt=0.20, n=25.

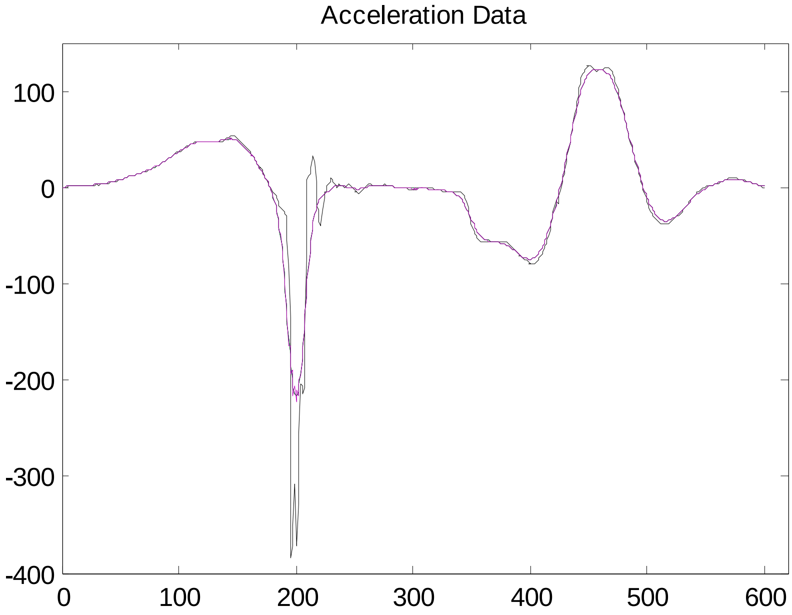


Figure 2.6.3: (Linear Diffusion Filtering). T=20 & Δt=0.01, 0.1, 0.2, 0.5

Black Line: The Original Data. Red Line: T=20, Δt=0.01, n=2000 Magenta Line: T=20, Δ*t=0.50*, n=80

Blue Line: T=20, Δt=0.10, n=200 Green Line: T=20, Δt=0.20, n=100

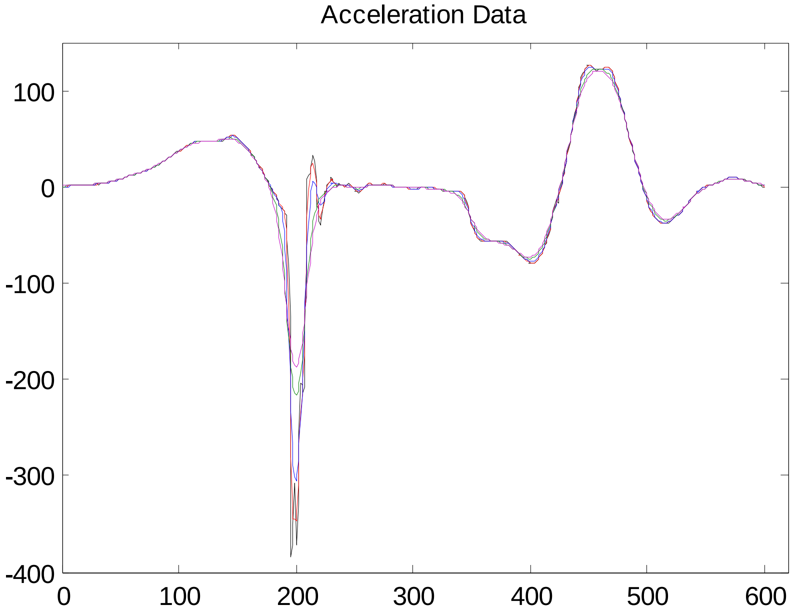


Figure 2.6.4: (Linear Diffusion Filtering). Δt=0.1 & T=1, 5, 20, 30

Black Line: The Original Data. Red Line: T=1, Δt=0.1, n=10 Magenta Line: T=30, Δ*t=0.1*, n=300

Blue Line: T=5, Δt=0.1, n=50 Green Line: T=20, Δt=0.1, n=200

Linear Diffusion process is not invertible. When some negative values are set to Δt, some undesirable outputs are obtained, seen in Figure 2.6.5.

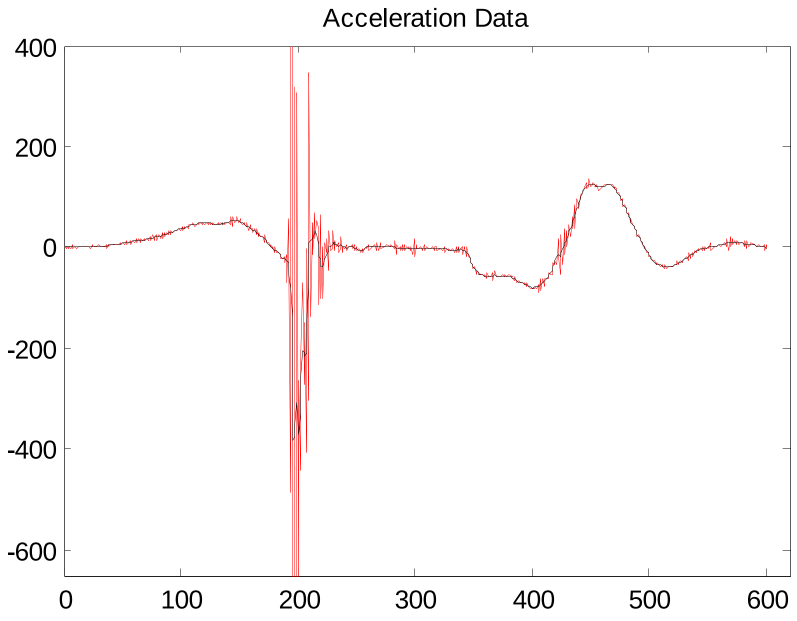


Figure 2.6.5: (Linear Diffusion Filtering). T=1, Δt=-0.05

Black Line is the Original Data.

Red Line: T=1, Δt=-0.05

For the condition c=1, the stability limit for Δt is 0.50. So, when Δt is set to a greater value than 0.50, Figure 2.6.6 and Figure 2.6.7, some unstable results are obtained.

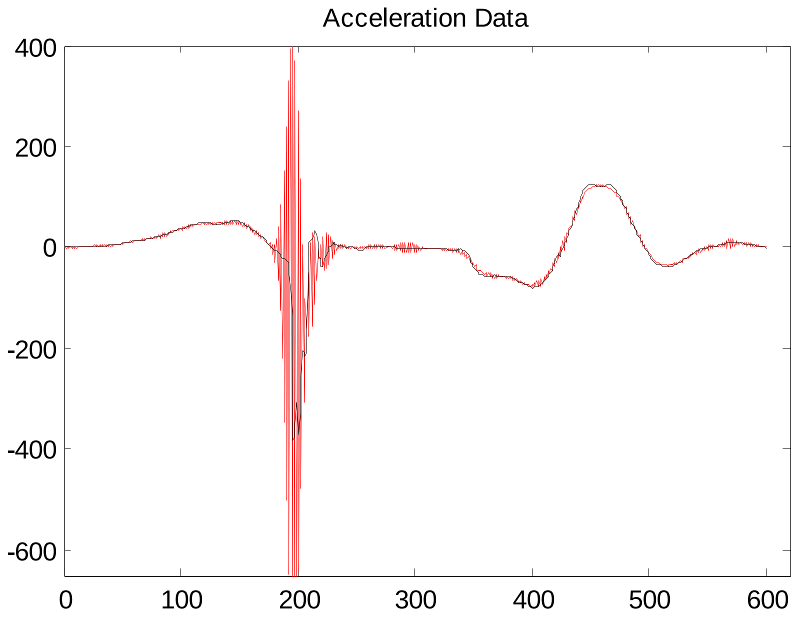


Figure 2.6.6: (Linear Diffusion Filtering). T=20, Δt=0.53

Black Line is the Original Data.

Red Line: T=20, Δt=0.53, n=37

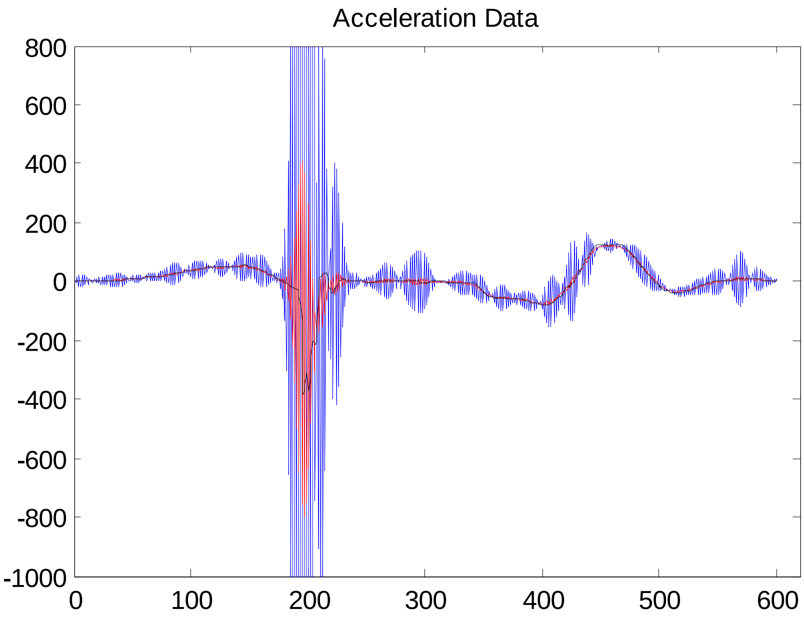


Figure 2.6.7: (Linear Diffusion Filtering). T=20 & Δt=0.53, 0.55

Black Line is the Original Data.

Red Line: T=20, Δt=0.53, n=37 Blue Line: T=20, Δt=0.55, n=36

2.6.2 Displacement Data Results for Linear Diffusion Filtering

After applying the Linear Diffusion filtering to the Displacement data, Figure 1.1.2.a, following results are obtained, Figure 2.6.8 and Figure 2.6.9.

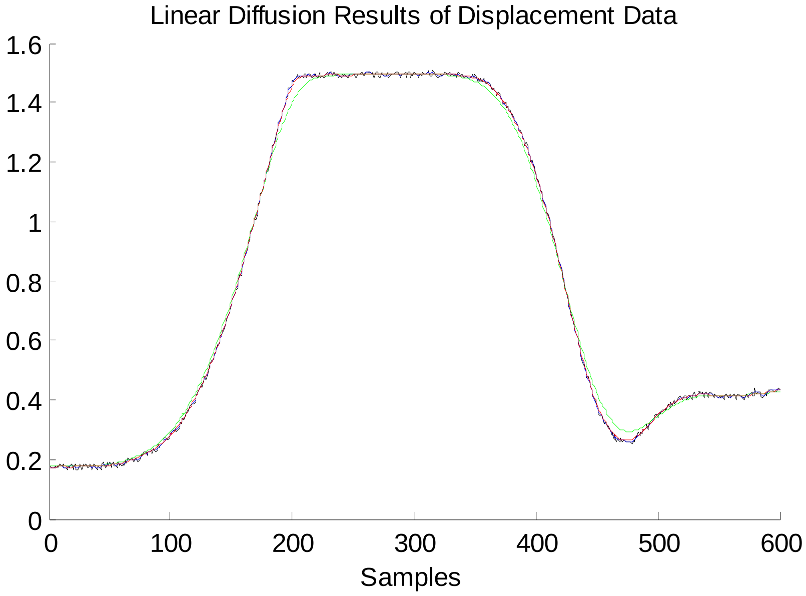


Figure 2.6.8: (Linear Diffusion Filtering)

Black Line: Original Displacement Data. Red Line: The result for T=10, Δt=0.1

Blue Line: The result for T=1, Δt=0.1 Green Line: The result for T=100, Δt=0.1

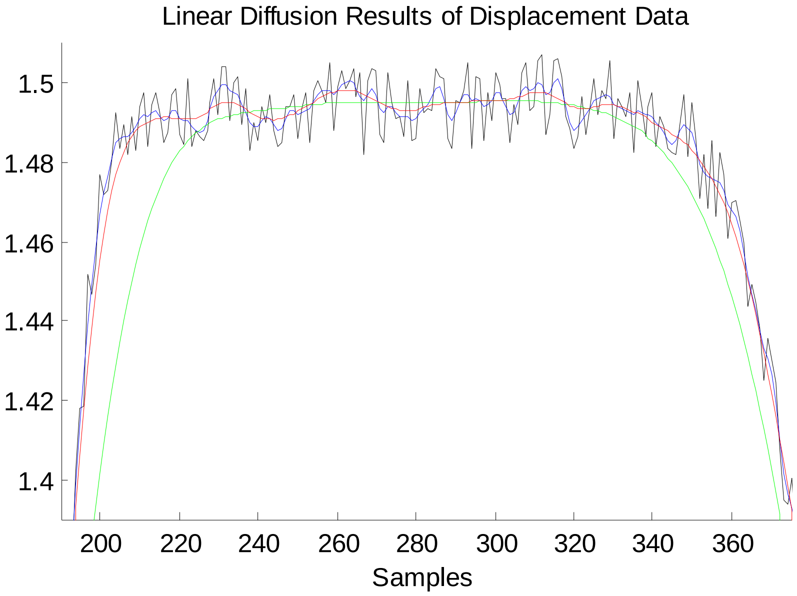


Figure 2.6.9: (Linear Diffusion Filtering) The samples, between 190 and 380, of Figure 2.6.8

Black Line: Original Displacement Data Red Line: The result for T=10, Δt=0.1,

Blue Line: The result for T=1, Δt=0.1, Green Line: The result for T=100, Δt=0.1.

When Linear Diffusion is applied to a data, for each of the iteration, mean value of the data doesn't change (because of the Neumann BC). Variance, standard deviation and total gradient, $\Sigma|\Delta|^2$ monotonically decreases.

Figure 2.6.10 is the result of Linear Diffusion filter by using n=10 and Δt=0.1. Mean, variance, total gradient and standard deviation are given in Figure 2.6.11 by using Figure 2.6.10, but entropy is the result of Linear Diffusion by using n=200 and Δt=0.01.

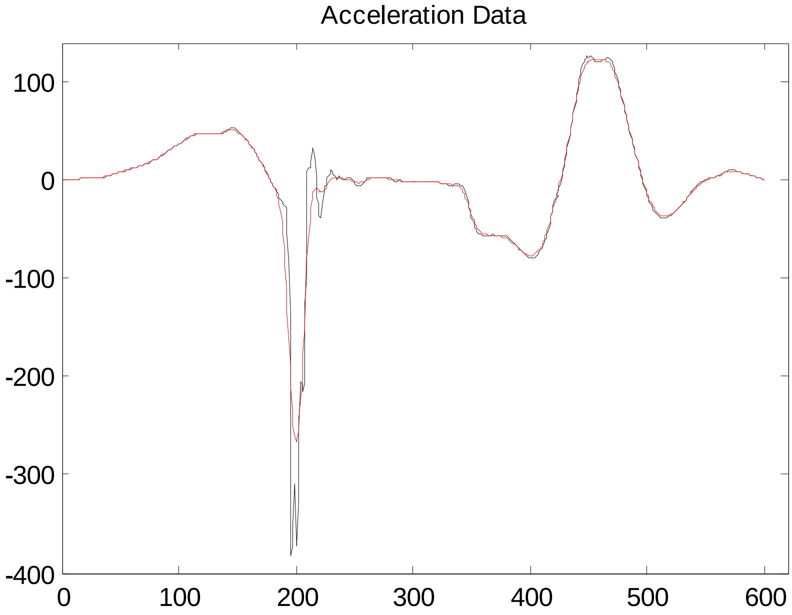


Figure 2.6.10: (Linear Diffusion Filtering). T=10, Δt=0.1

Black Line is the Original Data. Red Line is result for T=10, Δt=0.1, n=100

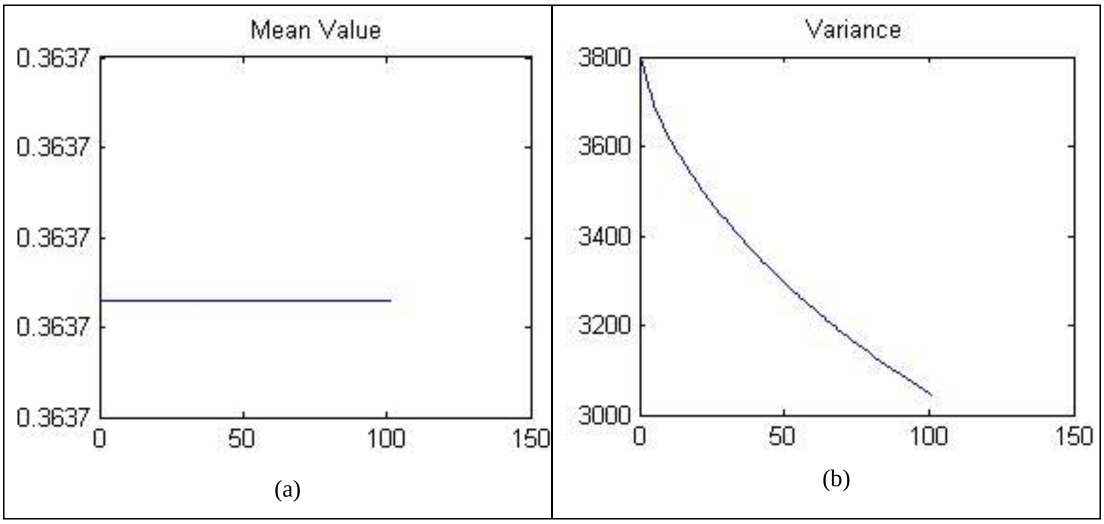
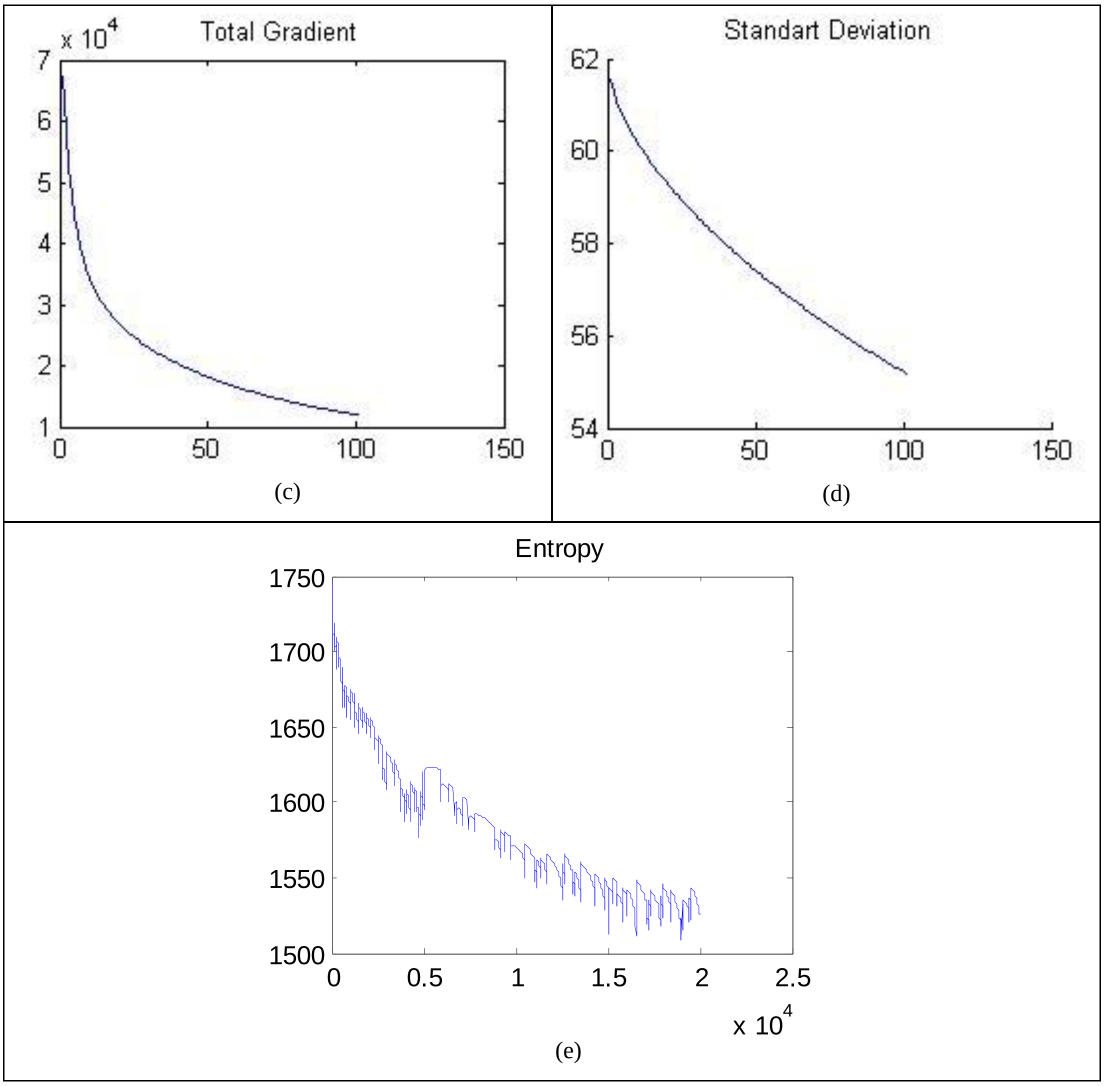


Figure 2.6.11: (Linear Diffusion Filtering). T=10, Δt=0.1 **(a)** Mean Value, **(b)** Variance, **(c)** Total Gradient, **(d)** Standard Deviation, **(e)** Entropy for the values T=200 and Δt=0.01.



(c)

(d)

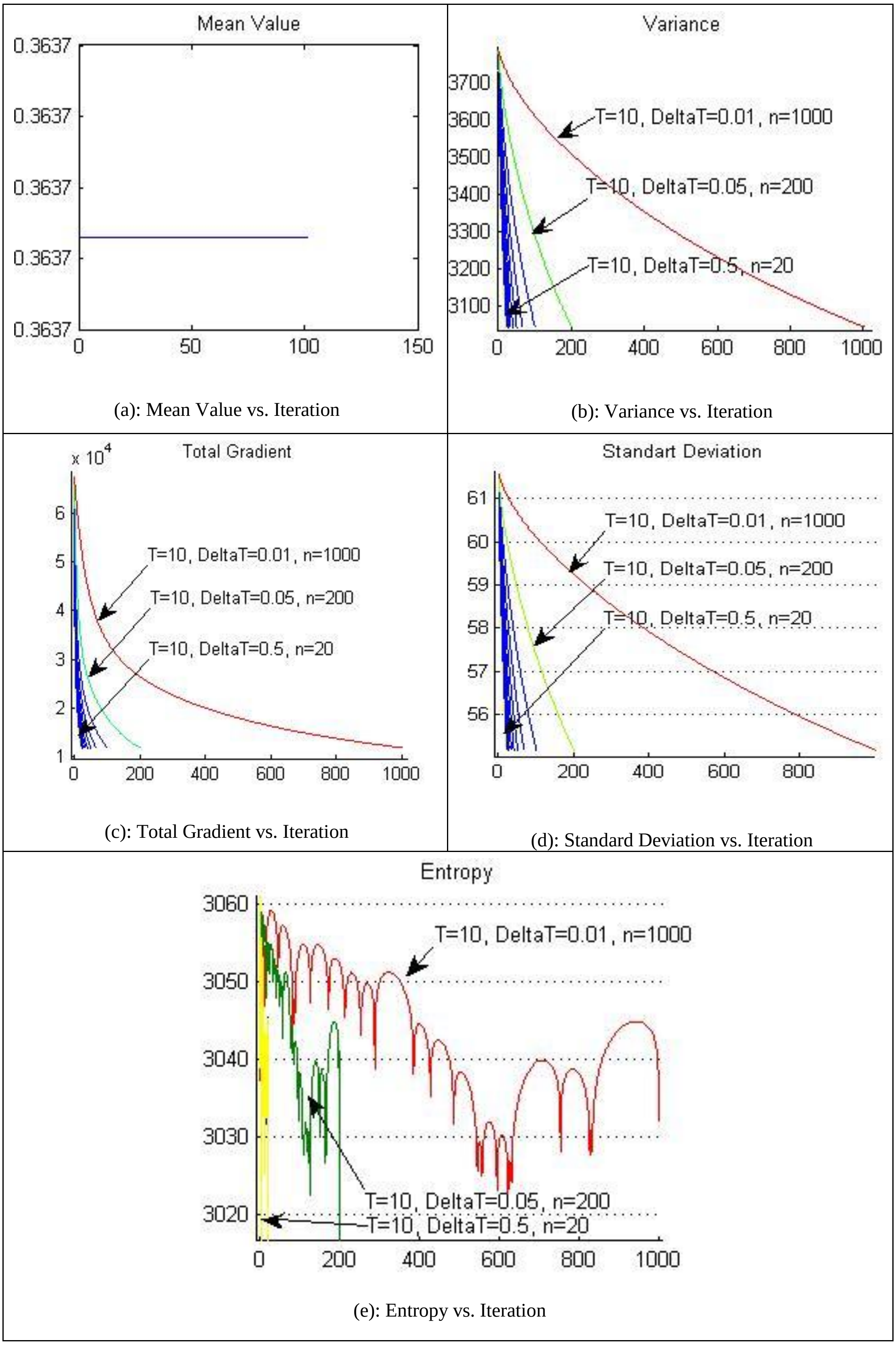
(e)

Figure 2.6.11: (Continued).

Changes in mean, variance, standard deviation, total gradient and entropy are given in Figure 2.6.12 by using T=10 and various Δt.

Δt=[0.01, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5]

There is no change in mean for all the iterations. Last Iteration values of Variance, Total Gradient and Standard Deviation, given in Figure 2.6.12.b, Figure 2.6.12.c and Figure 2.6.12.d, for the various Δt and n in case of constant T are nearly same.



(a): Mean Value vs. Iteration

(b): Variance vs. Iteration

(c): Total Gradient vs. Iteration

(d): Standard Deviation vs. Iteration

(e): Entropy vs. Iteration

Figure 2.6.12: (Linear Diffusion Filtering). T=10, Δt=0.01, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5 **(a)**Mean Value, **(b)**Variance, **(c)** Total Gradient, **(d)** Standard Deviation, **(e)** Entropy.

2.6.3 Second Derivative Results for Linear Diffusion Filtering

Second Derivative Result of Smoothed Displacement data by using the Linear Diffusion filtering for various T and Δt values can be seen in Figure 2.2.13.

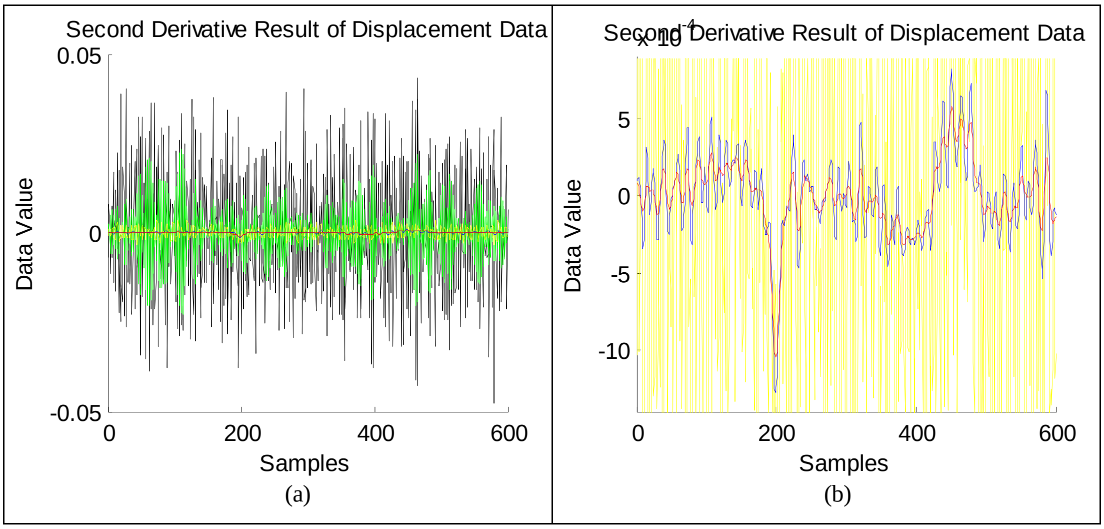


Figure 2.6.13: (Linear Diffusion Filtering). (a) Second Derivative Result of Smoothed **Displacement Data (b)** Same as Figure 2.6.13.a except Original Displacement Data (Black Line) and Green Line.

Black Line: Second Derivative of the Original Displacement Data,
Green Line: Second Derivative of the Data by using the **T=5**, ((Δt=0.50, n=10), **Δt.n=T),**
Yellow Line: Second Derivative of the Data by using the **T=1**, (Δt=0.01, n=100),
Blue Line: Second Derivative of the Data by using the **T =5**, (Δt=0.1, n=50),
Red Line: Second Derivative of the Data by using the **T=10**, (Δt=0.1, n=100).

2.6.4 Comparing Linear Diffusion Filters of Varying Parameters

The Comparative Second Derivative Results of Smoothed Displacement data and Original Acceleration data are seen in Figure 2.6.14. The peak, seen in the Figure 2.6.14 in the area of 1, is the critical feature of Acceleration data. This peak should be preserved after smoothing operation, but the peak, in the area of 2, is the noise that should be smoothed as much as possible. This is the major difficulty of this problem. In the area of 2, the result of the filtering shown by green line is successful because it is successfully got rid of the noise in that area, but the same line is not successful in the area of 1. The data in the area of 1 is not a noise, but the filtered signal as shown by green line lost this critical feature. The result of filtering shown by red line gives relatively good result while giving the worst result in the area of 2. Also the results of the filtered Displacement data can also be seen in Figure 2.6.8 and in Figure 2.6.9 by using the same parameters, except magenta line.

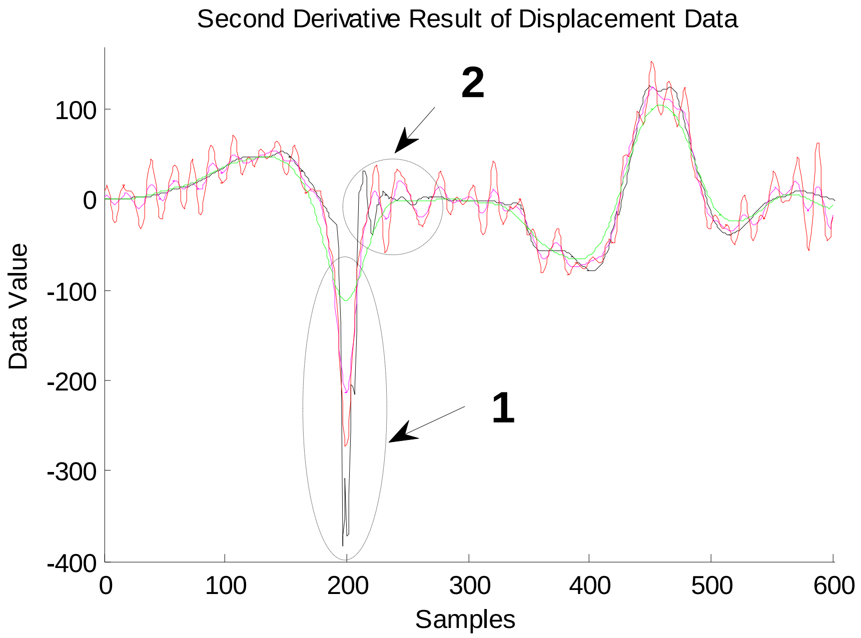


Figure 2.6.14: (Linear Diffusion Filtering). The Comparative Second Derivative Results of Smoothed Displacement Data and Original Acceleration Data.

Black Line: Original Acceleration Data

Red Line: Second Derivative of Smoothed Displacement Data by using the **T=10**, Δt=0.1

Magenta Line: Second Derivative of Smoothed Displacement Data by using the **T=20**, Δt=0.1

Green Line: Second Derivative of Smoothed Displacement Data by using the **T=100**, Δt=0.1

2.7 GENERAL REVIEW OF LINEAR SMOOTHING FILTERS

In this chapter, Linear smoothing filters are experimented with. Linear smoothing filters eliminate high frequency components. Removing noise from an input signal, without eliminating important features, is the first step of signal processing. Linear smoothing filters are not good at discriminating the noise and the important features in the signal. They remove the critical features. This is an undesirable result of the Linear smoothing filters.

Both the spatial domain filters and frequency domain filters are also experimented with. The transition between spatial domain and frequency domain is provided by the convolution theorem.

Let $f(x)$ and $h(x)$ be functions in spatial domain. If the Fourier transform, (2.34), of $f(x)$ and $h(x)$ in spatial domain is taken, F(u) and H(u) are obtained in frequency domain. Convolution of f(x)*h(x) in spatial domain is equal to the multiplication of $\mathrm{F(u)} \cdot \mathrm{H(u)}$ in frequency domain, (2.35).

$$F(u) = \frac{1}{N}\sum_{x=0}^{N-1} f(x) \cdot e^{-i \cdot 2\pi (\frac{u \cdot x}{N})} \tag{2.34}$$

$$\begin{gathered} f(x) \Leftrightarrow F(u) \qquad\qquad h(x) \Leftrightarrow H(u) \\ \mathrm{f(x)*h(x)} \Leftrightarrow \mathrm{F(u)} \cdot \mathrm{H(u)} \end{gathered} \tag{2.35}$$

Similar to (2.34), $f(x)$ and $h(x)$ are obtained in spatial domain by taking the Inverse Fourier transform of F(u) and H(u) in frequency domain using (2.36).

$$f(x) = \frac{1}{N}\sum_{u=0}^{N-1} F(u) \cdot e^{i \cdot 2\pi (\frac{u \cdot x}{N})} \tag{2.36}$$

N : Size of the data.
f(x) : Data in Spatial Domain
F(u): Data in Frequency Domain

The Fourier transform is used to separate data into its sine and cosine components. In the Fourier Domain data, each point represents a particular frequency contained in the spatial domain data [6].

The solution for Linear Diffusion Equation can be obtained by convolving the input data by a Gaussian Kernel on condition that $T = \frac{\sigma^2}{2 \cdot c}$, where c is assumed as 1. So the equation is $T = \frac{\sigma^2}{2}$ and $\sigma = \sqrt{2 \cdot T}$.

When the equation $T = \frac{\sigma^2}{2}$ is provided, nearly the same results are obtained. Linear Diffusion parameter, which is Δt, is defined as 0.25 for all of the experiments. And Gaussian filter size is defined as 6σ+1, in Figure 2.7.1.

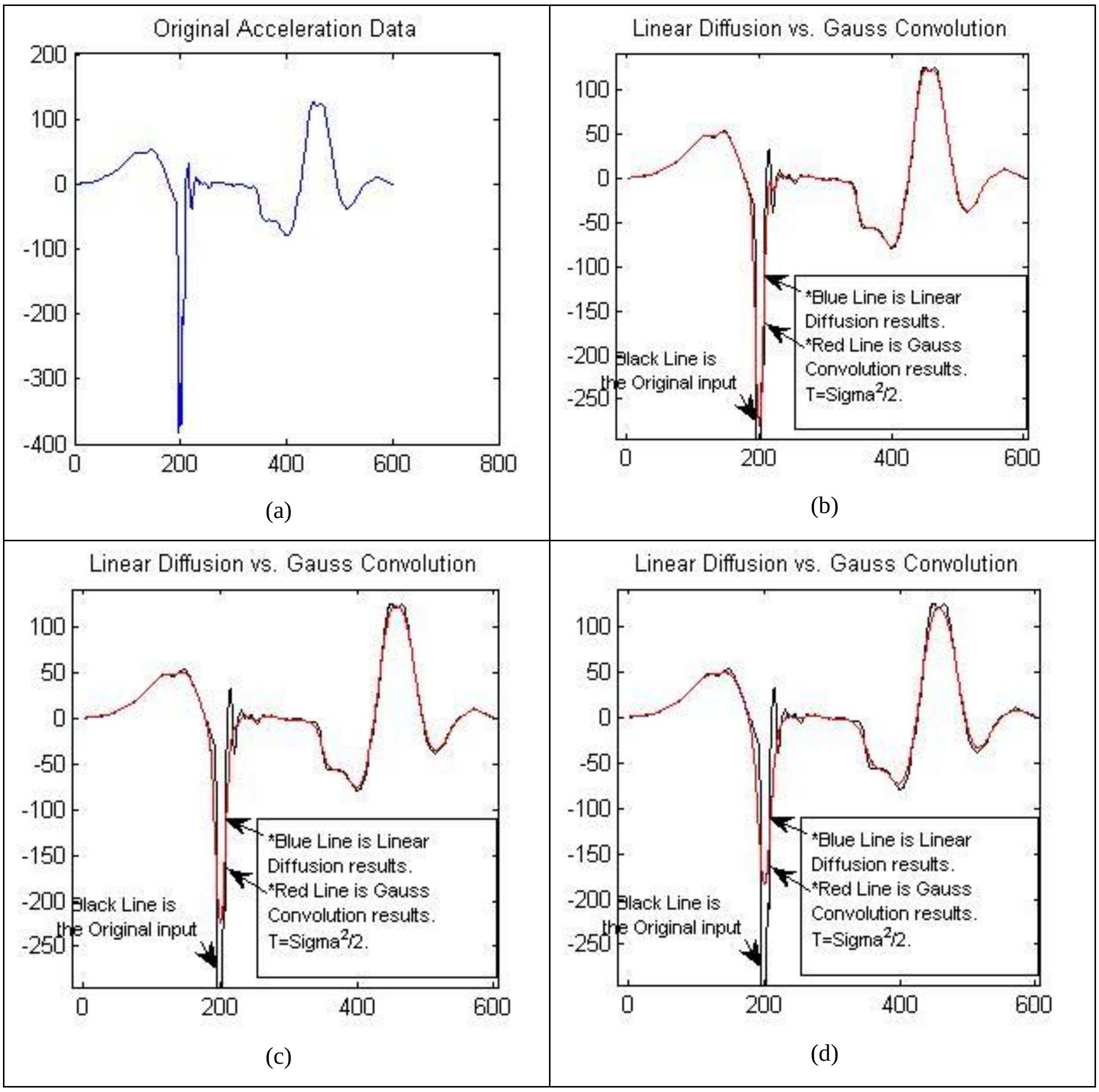
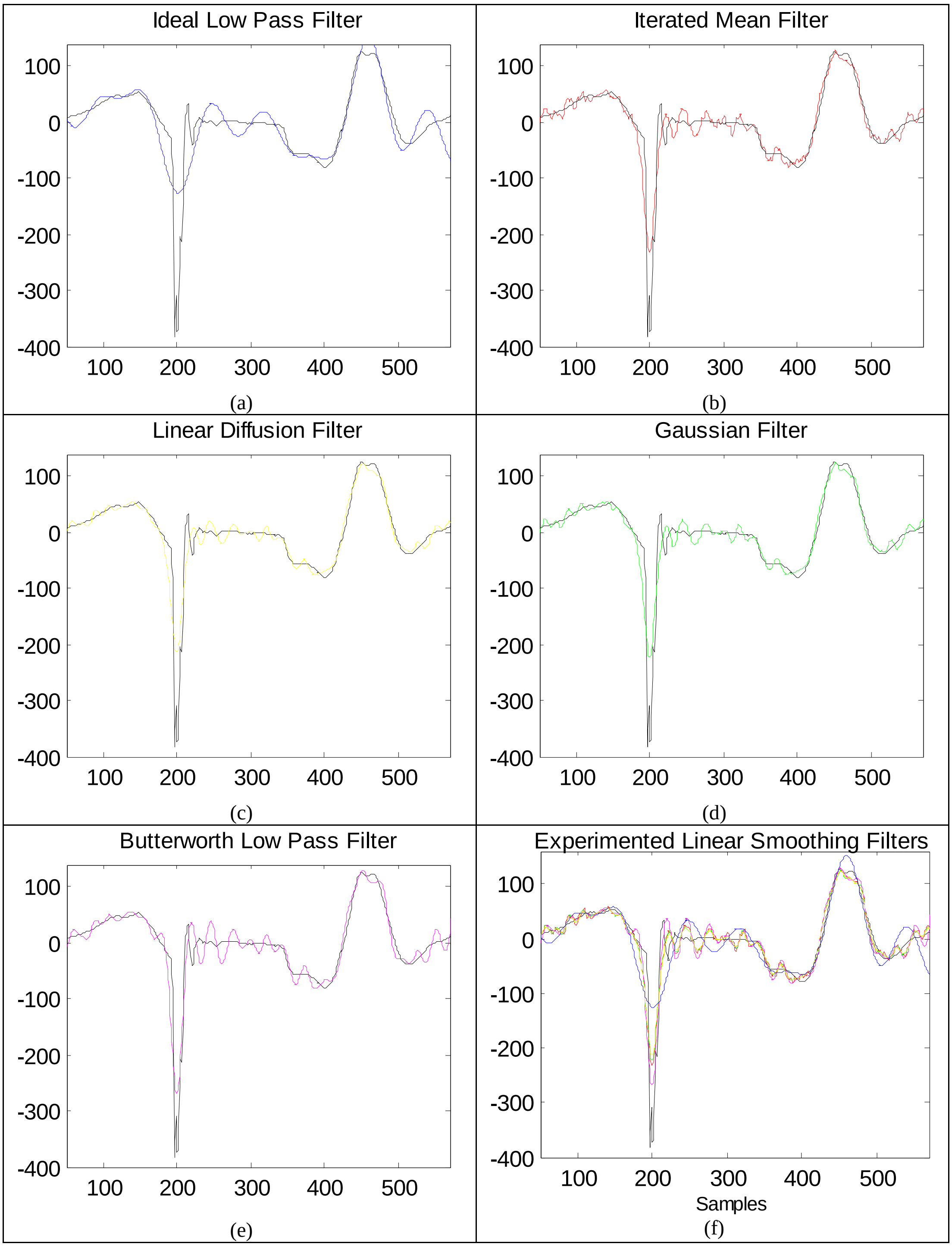


Figure 2.7.1: (Linear Diffusion Filtering vs. Gauss Convolution). T=10, Δt=0.25,
(a) Original Acceleration Data,
(b) Linear Diffusion for T=8 vs. Gauss Convolution for σ=4,
(c) Linear Diffusion for T=18 vs. Gauss Convolution for σ=6,
(d) Linear Diffusion for T=32 vs. Gauss Convolution for σ=8.

Which Linear smoothing algorithms give better result than the others can be seen in Figure 2.7.2, when the noise level of the smoothed Displacement data is hold similarly for the first 150 samples and it is zoomed around the point of impact level.



(a) (b) (c) (d) (e) (f)

Figure 2.7.2: (The Comparative Second Derivative Results of Displacement Data by using All Experimented Linear Smoothing Filters). Black Line: Original Acceleration Data,

(a) Ideal Low Pass Filter Result by using the Cut off Frequency=8, **(b)** Iterated Mean Filter Result by using Filter Size=11 for 3 iterations, **(c)** Linear Diffusion Filter Result by using the T=20, Δt=0.1, **(d)** Gaussian Filter Result by using σ=6, **(e)** Butterworth Low Pass Filter Result by using the Cut off Frequency=24 and Order=2, **(f)** Comparison of all results from Figure 2.7.2.a to Figure 2.7.2.e

CHAPTER 3

NON-LINEAR SMOOTHING FILTERS

In the previous chapter, chapter 2, Linear smoothing filters are given. Linear smoothing filters are not good at saving the important part in the data. Non-linear smoothing filters are important at smoothing the data. Linear smoothing filters can not determine the data, whether it is a noise or an important part of the data. The condition, which is expected, is smoothing the data while keeping the distinctive data. Non-linear smoothing filters provide this condition.

In this section, I am going to examine some kinds of Non-linear smoothing filters, which are;

1. Kuwahara Filter
2. Sigma Filter
3. Median Filter
4. Perona Malik Filter
5. The Ambrosio Tortorelli Approximation of the Mumford Shah Model

Our test data does not include the data which is eliminated from all kinds of noise. So, to be able to see the effect of Non-linear smoothing filters, a sinusoidal test data is used with some added noise. Signal to noise ratio of the original noisy test data is equal to 10.9379.

The Non-linear smoothing filter results are:

Signal to noise ratio of the test data, filtered by Kuwahara filter, is 14.0394. Signal to noise ratio of the test data, filtered by Sigma filter, is 19.7889. Signal to noise ratio of the test data, filtered by Median filter, is 17.5168. Signal to noise ratio of the test data, filtered by Perona Malik filter, is 19.8506. Signal to noise ratio of the test data, filtered by Ambrosio Tortorelli approximation of Mumford Shah model, is 19.8093.

3.1 KUWAHARA FILTER

Kuwahara is a Non-linear smoothing filtering. One dimensional Kuwahara filter works on an array which is divided into two overlapping sub array parts [16]. In each sub array part, the mean and the variance are computed. The output value (located at the center of the array) is set to the mean value of the sub array where the variance is smaller than the other part. For the condition that mask size is 5:

$$\square \left[X_i \quad X_{i+1} \quad \overline{X_{i+2}} \right] \left[\overline{Y_{i+2}} \quad Y_{i+3} \quad Y_{i+4} \right] \square \tag{3.1}$$

X_{i+2} *and* Y_{i+2} are point the same address.

Compute the following values:

$mean_X$: Mean the value of $\left[X_i \quad X_{i+1} \quad X_{i+2} \right]$ (3.2)

$mean_Y$: Mean the value of $\left[Y_{i+2} \quad Y_{i+3} \quad Y_{i+4} \right]$ (3.3)

var_X : Variance the value of $\left[X_i \quad X_{i+1} \quad X_{i+2} \right]$ (3.4)

var_Y : Variance the value of $\left[Y_{i+2} \quad Y_{i+3} \quad Y_{i+4} \right]$ (3.5)

var_X and var_Y are compared and the mean value of the smaller variance is copied to the address which is defined by X_{i+2} *and* Y_{i+2}. These X_{i+2} *and* Y_{i+2} values refer to the same address which is located at the center of the array.

3.1.1 Displacement Data Results for Kuwahara Filtering

After applying the Kuwahara filter on the Displacement data, Figure 1.1.2.a, the results are in Figure 3.1.1 and in Figure 3.1.2:

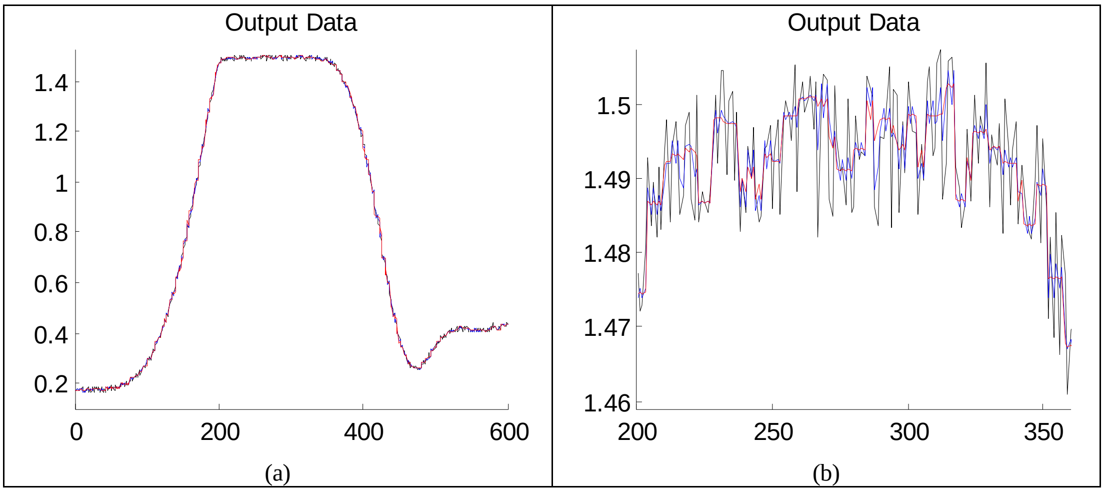


Figure 3.1.1: (Kuwahara Filtering). (a) Smoothed Displacement Data Result, **(b)** The samples, between 200 and 360, of Figure 3.1.1.a.

Black Line: The Original Displacement Data,

Blue Line: Smoothed Displacement Data by using the Mask Size=5,

Red Line: Smoothed Displacement Data by using the Mask Size=5 for 3 iterations.

Table 3.1.1.1: Kuwahara Filter Results by using Mask Size=5

Height (meters)	**:**	...	1,4797	1,4738	1,4786	1,4752	1,4780	1,4692	...
Sample #	**:**	...	353	354	355	356	357	358	...

Table 3.1.1.2: Kuwahara Filter Results by using Mask Size=5 for 3 iterations

Height (meters)	**:**	...	1,4769	1,4764	1,4768	1,4764	1,4768	1,4678	...
Sample #	**:**	...	353	354	355	356	357	358	...

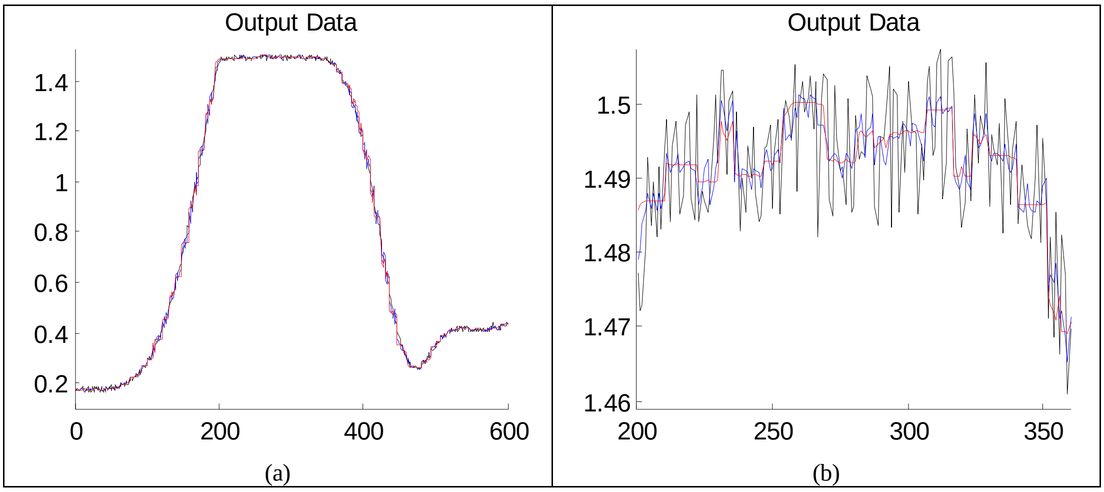


Figure 3.1.2: (Kuwahara Filtering). (a) Smoothed Displacement Data Result, **(b)** The samples, between 200 and 360, of Figure 3.1.2.a.

Black Line: The Original Displacement Data,

Blue Line: Smoothed Displacement Data by using the Mask Size=9,

Red Line: Smoothed Displacement Data by using the Mask Size=9 for 3 iterations.

Table 3.1.1.3: Kuwahara Filter Results by using Mask Size=9

Height (meters)	**:**	...	1,4769	1,4759	1,4786	1,4712	1,4720	1,4686	...
Sample #	**:**	...	353	354	355	356	357	358	...

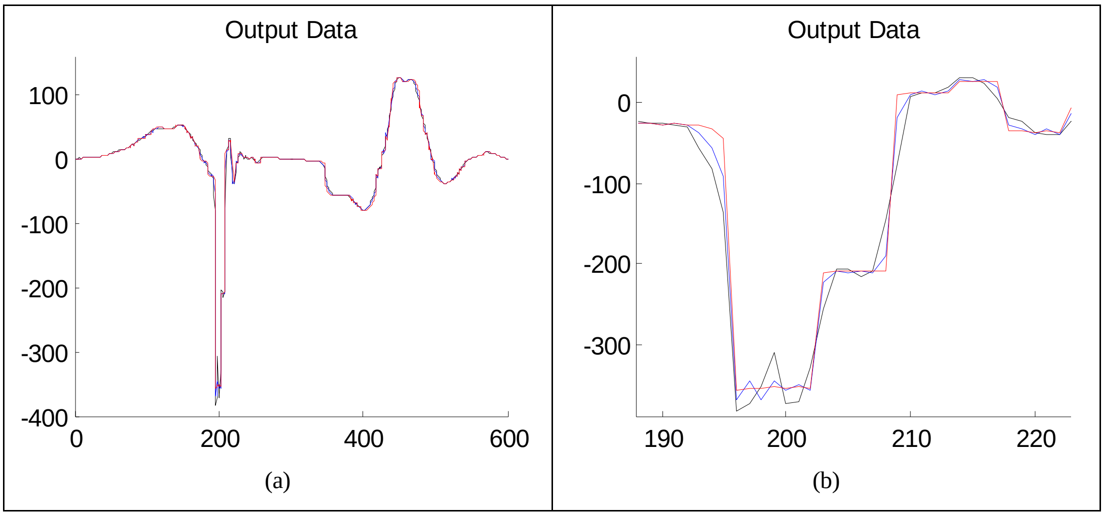
Table 3.1.1.4: Kuwahara Filter Results by using Mask Size=9 for 3 iterations

Height (meters)	**:**	...	1,4729	1,4718	1,4708	1,4741	1,4694	1,4693	...
Sample #	**:**	...	353	354	355	356	357	358	...

3.1.2 Acceleration Data Results for Kuwahara Filtering

After applying the Kuwahara filter to the Acceleration data to see the smoothing effect, in Figure 1.1.2.b, results are following.

As in Figure 3.1.1, figure 3.1.2, figure 3.1.3 and Figure 3.1.4, when Mask size is increased, the difference between maxima and minima does not decrease much, however, a corruption on the signal itself occurs.



(a)

(b)

Figure 3.1.3: (Kuwahara Filtering). (a) Smoothed Acceleration Data Result, **(b)** The samples, between 190 and 220, of Figure 3.1.3.a.

Black Line: The Original Acceleration Data,
Blue Line: Smoothed Acceleration Data by using the Mask Size=5,
Red Line: Smoothed Acceleration Data by using the Mask Size=5 for 3 iterations.

Table 3.1.2.1: Kuwahara Filter Results by using Mask Size=5

Height (meters)	**:**	...	-90,63	-369,5	-344,5	-369,5	-344,5	-356,7	...
Sample #	**:**	...	195	196	197	198	199	200	...

Table 3.1.2.2: Kuwahara Filter Results by using Mask Size=5 for 3 iterations

Height (meters)	**:**	...	-43,34	-356,96	-353,4	-354	-351,83	-353,2	...
Sample #	**:**	...	195	196	197	198	199	200	...

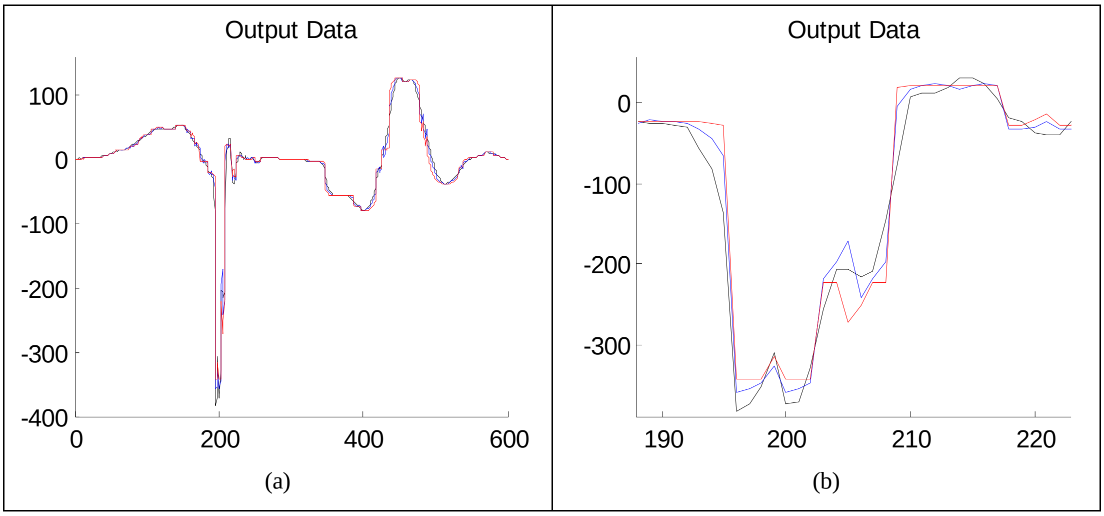


Figure 3.1.4: (Kuwahara Filtering). (a) Smoothed Acceleration Data Result, **(b)** The samples, between 190 and 220, of Figure 3.1.4.a.

Black Line: The Original Acceleration Data,

Blue Line: Smoothed Acceleration Data by using the Mask Size=9,

Red Line: Smoothed Acceleration Data by using the Mask Size=9 for 3 iterations.

Table 3.1.2.3: Kuwahara Filter Results by using Mask Size=9

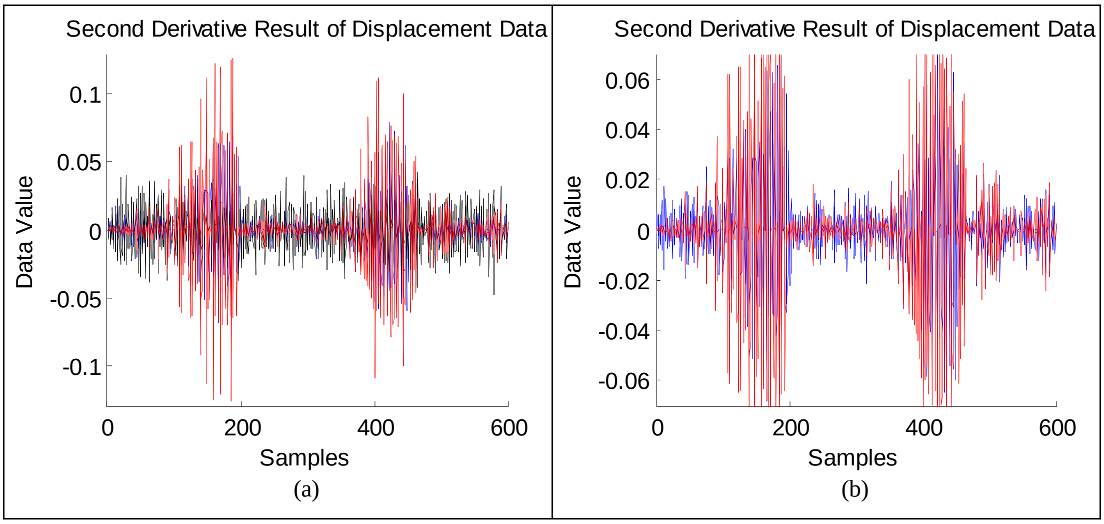
Height (meters)	**:**	...	-65,42	-357,86	-355,12	-346,2	-326,76	-357,86	...
Sample #	**:**	...	195	196	197	198	199	200	...

Table 3.1.2.4: Kuwahara Filter Results by using Mask Size=9 for 3 iterations

Height (meters)	**:**	...	-27,501	-342,58	-342,47	-342,1	-314,61	-342,58	...
Sample #	**:**	...	195	196	197	198	199	200	...

3.1.3 Second Derivative Results for Kuwahara Filtering

As it is seen in Figure 3.1.5 and Figure 3.1.6, if the value of Mask Size is increased, Kuwahara filter result starts to corrupt.



(a)

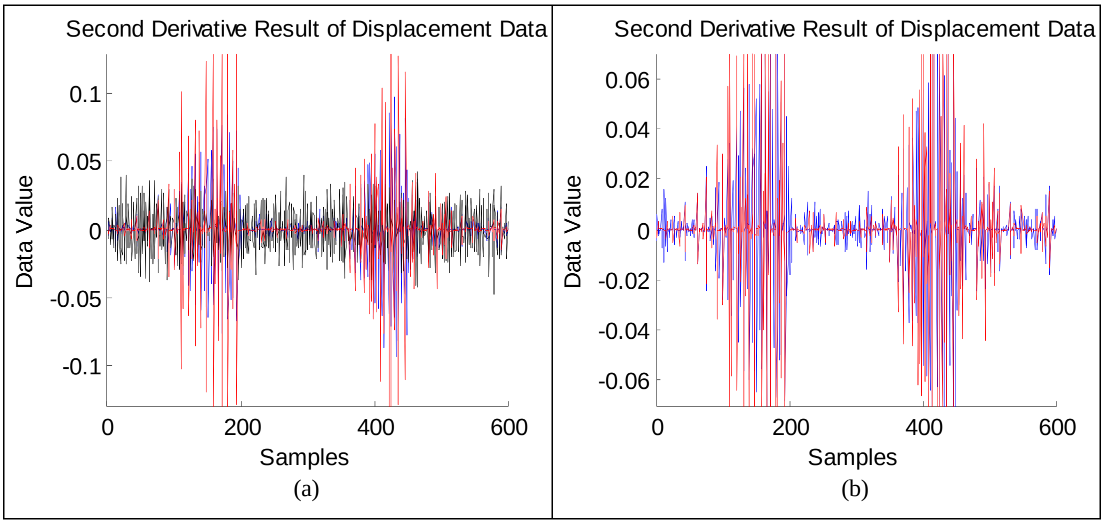
(b)

Figure 3.1.5: (Kuwahara Filtering). (a) Second Derivative Result of Smoothed **Displacement Data** using Mask Size= 5 and 9 **(b)** Same as Figure 3.1.5.a except Original Displacement Data (Black Line).

Black Line: Second Derivative of the Original Displacement Data,

Blue Line: Second Derivative of the Data by using the Mask Size=5,

Red Line: Second Derivative of the Data by using the Mask Size=9.



(a)

(b)

Figure 3.1.6: (Kuwahara Filtering). (a) Second Derivative Result of Smoothed **Displacement Data** using Mask Size= 5 for 3 iteration and Mask Size= 9 for 3 iteration **(b)** Same as Figure 3.1.6.a except Original Displacement Data (Black Line).

Black Line: Second Derivative of the Original Displacement Data,

Blue Line: Second Derivative of the Data by using the Mask Size=5 for 3 iteration,

Red Line: Second Derivative of the Data by using the Mask Size=9 for 3 iteration.

3.1.4 Comparing Kuwahara Filters of Varying Mask Size and Iteration

The Comparative Second Derivative Results of Smoothed Displacement data and Original Acceleration data are seen in Figure 3.1.7 and Figure 3.1.8. The results of the Kuwahara filtering and Iterative Kuwahara filtering are not clear at all. Iterative Kuwahara filter gives comparatively better result than Kuwahara filter.

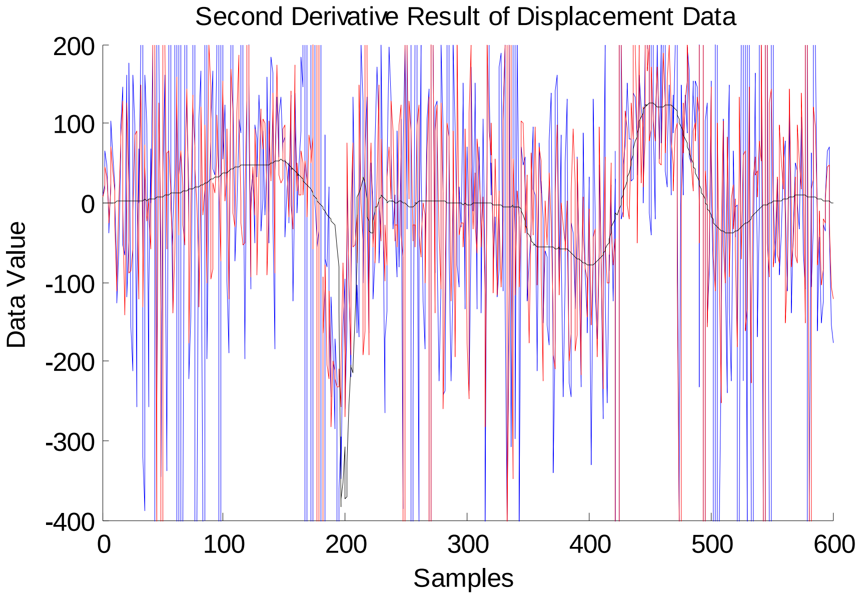


Figure 3.1.7: (Kuwahara Filtering). The Comparative Second Derivative Results of Smoothed Displacement Data and Original Acceleration Data. Black Line is the Original Acceleration Data.

Blue Line: Second Derivative of Smoothed Displacement Data by using the Mask Size=21
Red Line: Second Derivative of Smoothed Displacement Data by using the Mask Size=31

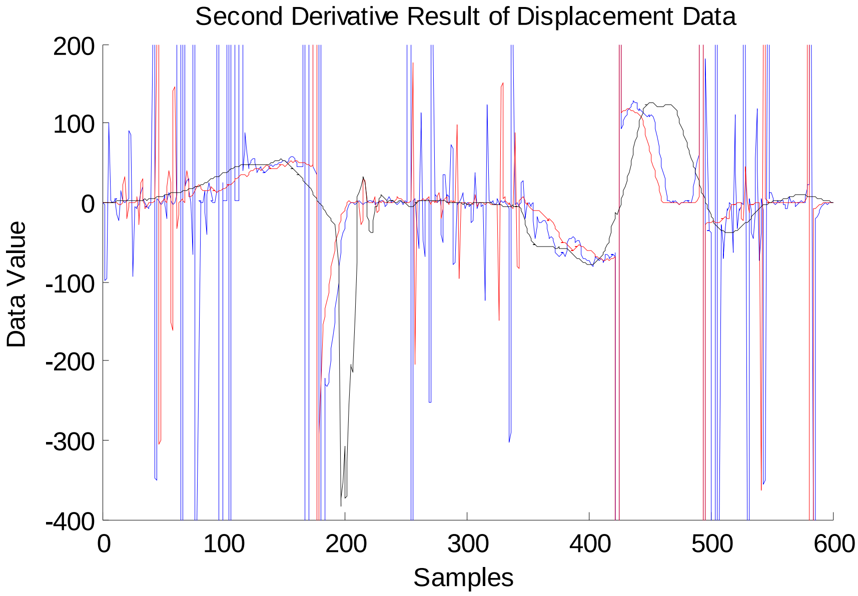


Figure 3.1.8: (Iterative Kuwahara Filtering). The Comparative Second Derivative Results of Smoothed Displacement Data and Original Acceleration Data. Black Line is the Original Acceleration Data.

Blue Line: Second Derivative of Smoothed Displacement Data by using the Mask Size=41 for 3 iteration
Red Line: Second Derivative of Smoothed Displacement Data by using the Mask Size=61 for 3 iteration

3.2 SIGMA FILTER

The main idea of the Sigma filter is based on the fact that 95.5% of the area of Gaussian distribution curve with mean (μ) and variance (σ^2) is between [μ-2σ] and [μ+2σ], in Figure 3.2.1.

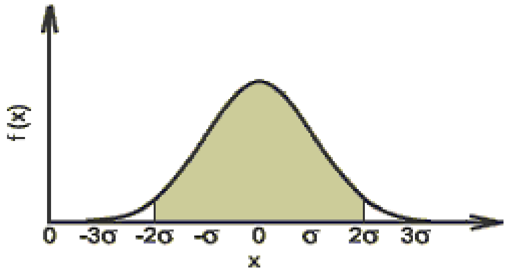


Figure 3.2.1: (Gaussian (Normal) Distribution Curve)

Applying this observation, Sigma filter works by computing the local average of neighboring data, inside the interval of [input(i)-2σ, input(i)+2σ], and replaces the corresponding data, output(i), with the local average value [17].

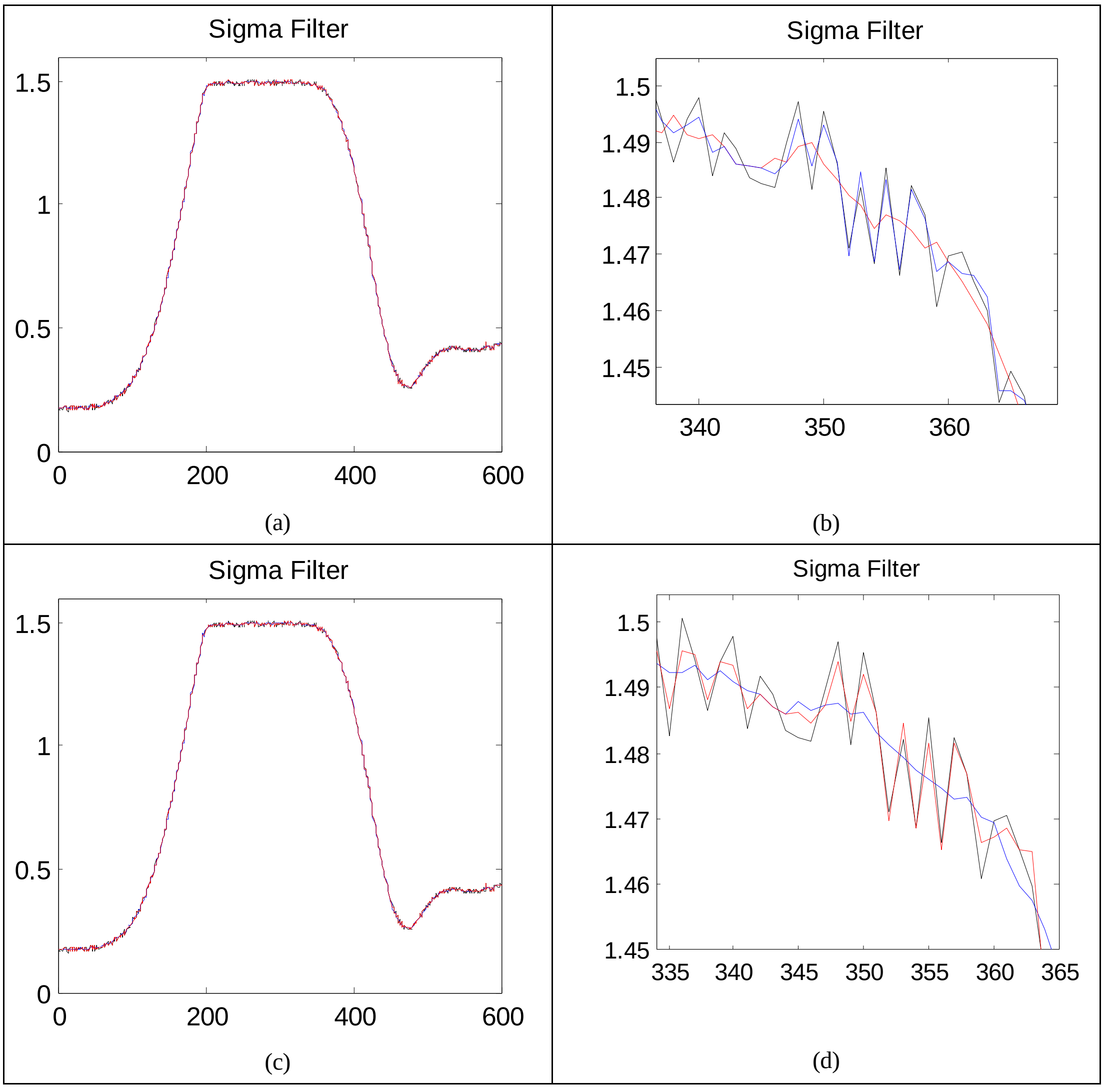
For example, if the mask size is defined as 7 and sigma value as 2, three data from both in the right and in the left side of the centered data whose value will be changed, are taken, and then the neighbor data values will be found, between [(input(i)-4), (input(i)+4)]. As a result, the average of these neighbor values are taken and put into the place of the output(i).

Sigma filtering is based on the assumption that the value of the data in input(i) is a good estimate of the local neighbor mean. So, choosing the appropriate Mask size and the sigma value is important.

3.2.1 Displacement Data Results for Sigma Filtering

After the application of Sigma filter on the Displacement data, the results are seen in Figure 1.1.2.a:

Keeping the value of Mask Size is constant in Sigma filter and decreases sigma parameter, Sigma filter result looks like the original input, Figure 3.2.2. Increasing the value of the sigma, the data do not get smoother and stay the same after a certain value.



(a)

(b)

(c)

(d)

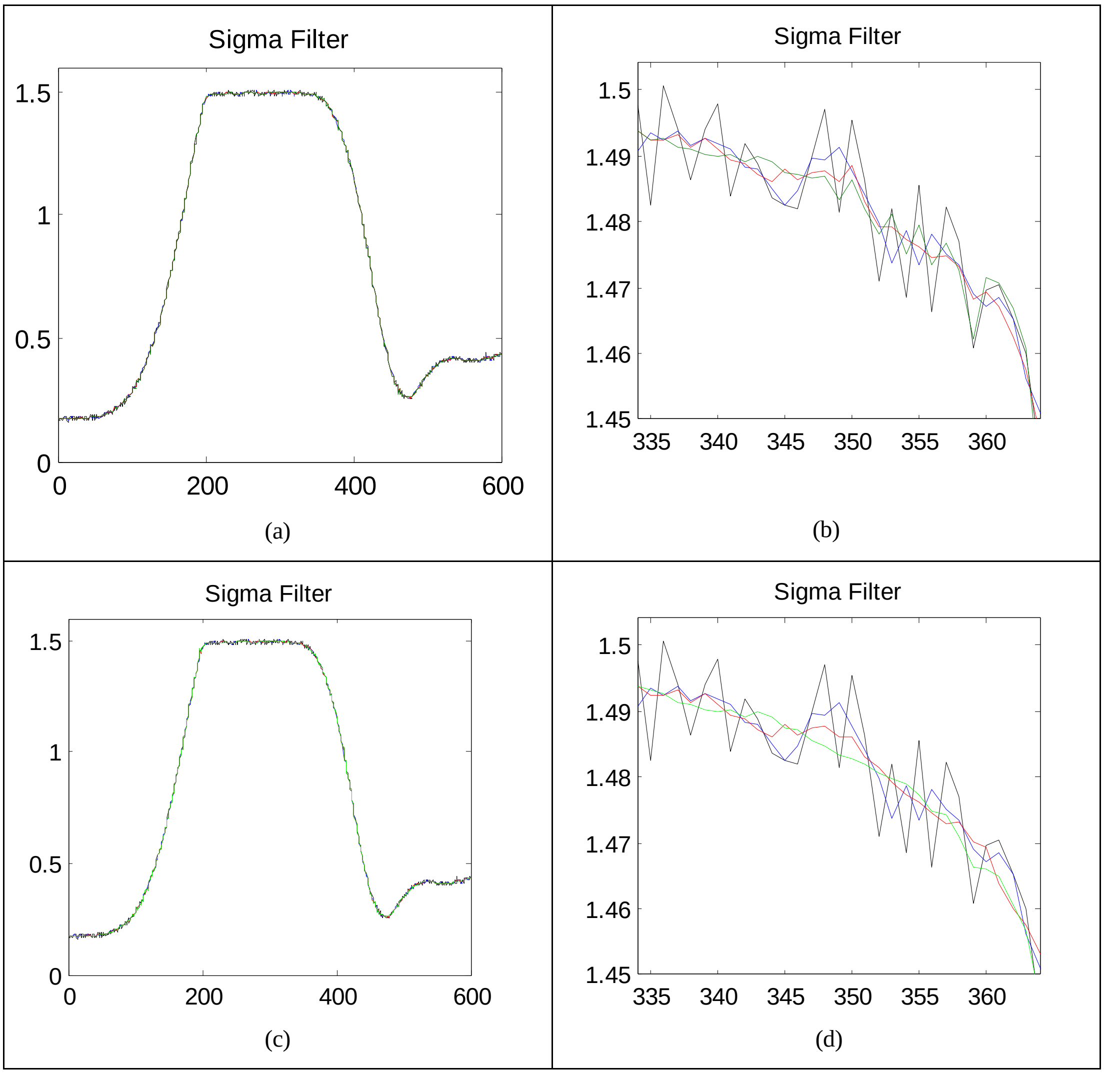
Figure 3.2.2: (Sigma Filtering), Mask Size is constant.

(a) Smoothed Displacement Data by using the values:
Black Line: The Original Displacement Data,
Blue Line: σ=0.015, Mask Size= 5
Red Line: σ=0.005, Mask Size= 5

(b) The samples, between 333 and 364, of Figure 3.2.2.a,

(c) Smoothed Displacement Data by using the values:
Black Line: The Original Displacement Data,
Blue Line: σ=0.015, Mask Size= 7
Red Line: σ=0.005, Mask Size= 7

(d) The samples, between 333 and 364, of Figure 3.2.2.c, As it is seen in Figure 3.2.3, when the value of sigma is kept constant in Sigma filter, and the Mask Size parameter is increased, Sigma filter result continues smoothing.



(a)

(b)

(c)

(d)

Figure 3.2.3: (Sigma Filtering), Sigma is constant.

(a) Smoothed Displacement Data by using the values:
Black Line: The Original Data,
Blue Line: σ=0.01, Mask Size= 3
Red Line: σ=0.01, Mask Size= 7
Green Line: σ=0.01, Mask Size= 15

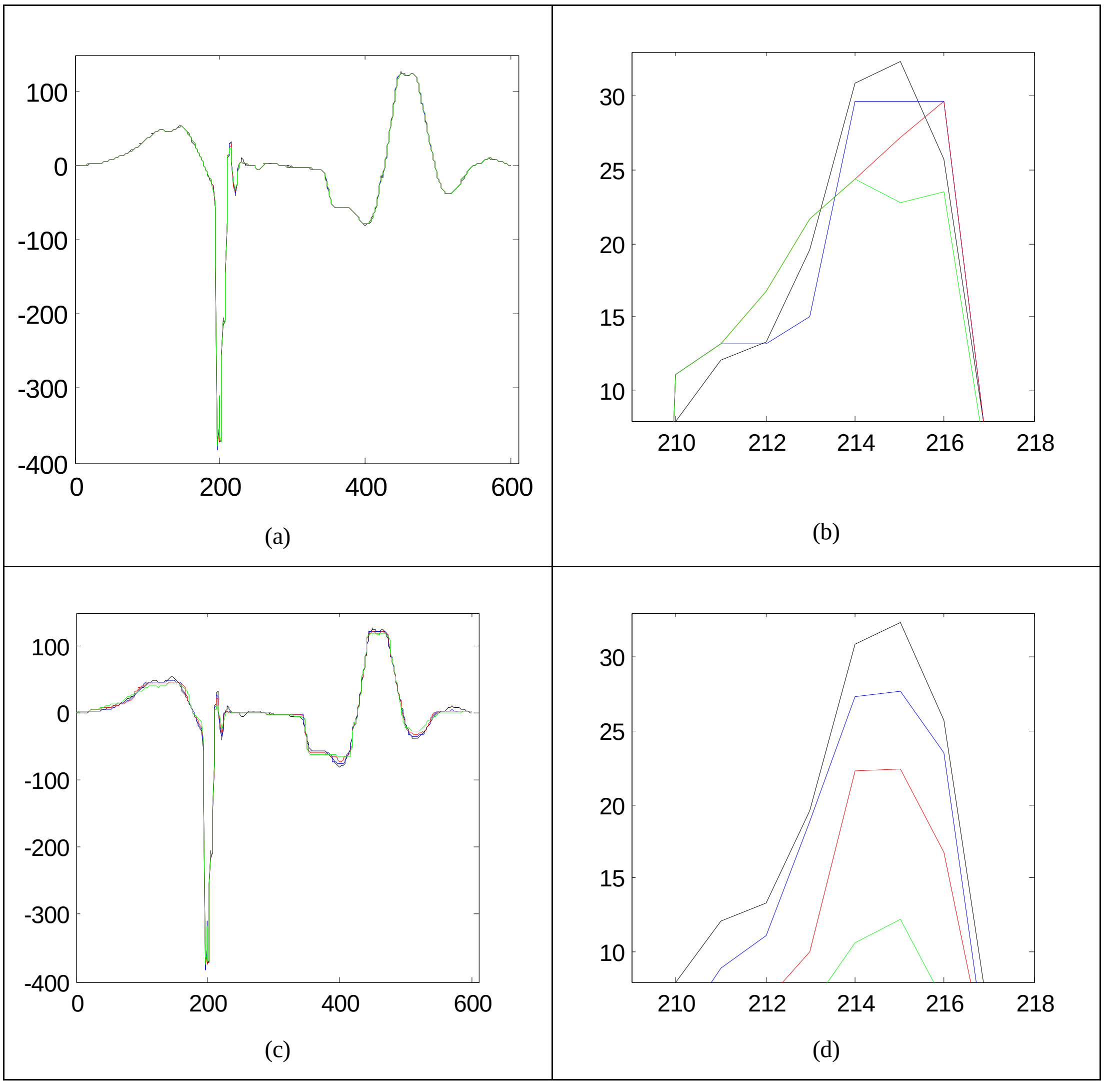
(b) The samples, between 333 and 364, of Figure 3.2.3.a,

(c) Smoothed Displacement Data by using the values:
Black Line: The Original Data,
Blue Line: σ=0.015, Mask Size= 3
Red Line: σ=0.015, Mask Size= 7
Green Line: σ=0.015, Mask Size= 15

(d) The samples, between 333 and 364, of Figure 3.2.3.c,

3.2.2 Acceleration Data Results for Sigma Filtering

After applying the Sigma filter on the Acceleration data, in Figure 1.1.2.a, the results are in Figure 3.2.4 and in Figure 3.2.5.



(a)

(b)

(c)

(d)

Figure 3.2.4: (Sigma Filtering), Mask Size is constant.

(a) Smoothed Acceleration Data by using the values:
Black Line: The Original Data,
Blue Line: σ=5, Mask Size= 5,
Red Line: σ=10, Mask Size= 5,
Green Line: σ=15, Mask Size= 5.

(b) The samples, between 209 and 218, of Figure 3.2.4.a

(c) Smoothed Acceleration Data by using the values:
Black Line: The Original Data,
Blue Line: σ=5, Mask Size= 101,
Red Line: σ=10, Mask Size= 101,
Green Line: σ=15, Mask Size= 101.

(d) The samples, between 209 and 218, of Figure 3.2.4.c

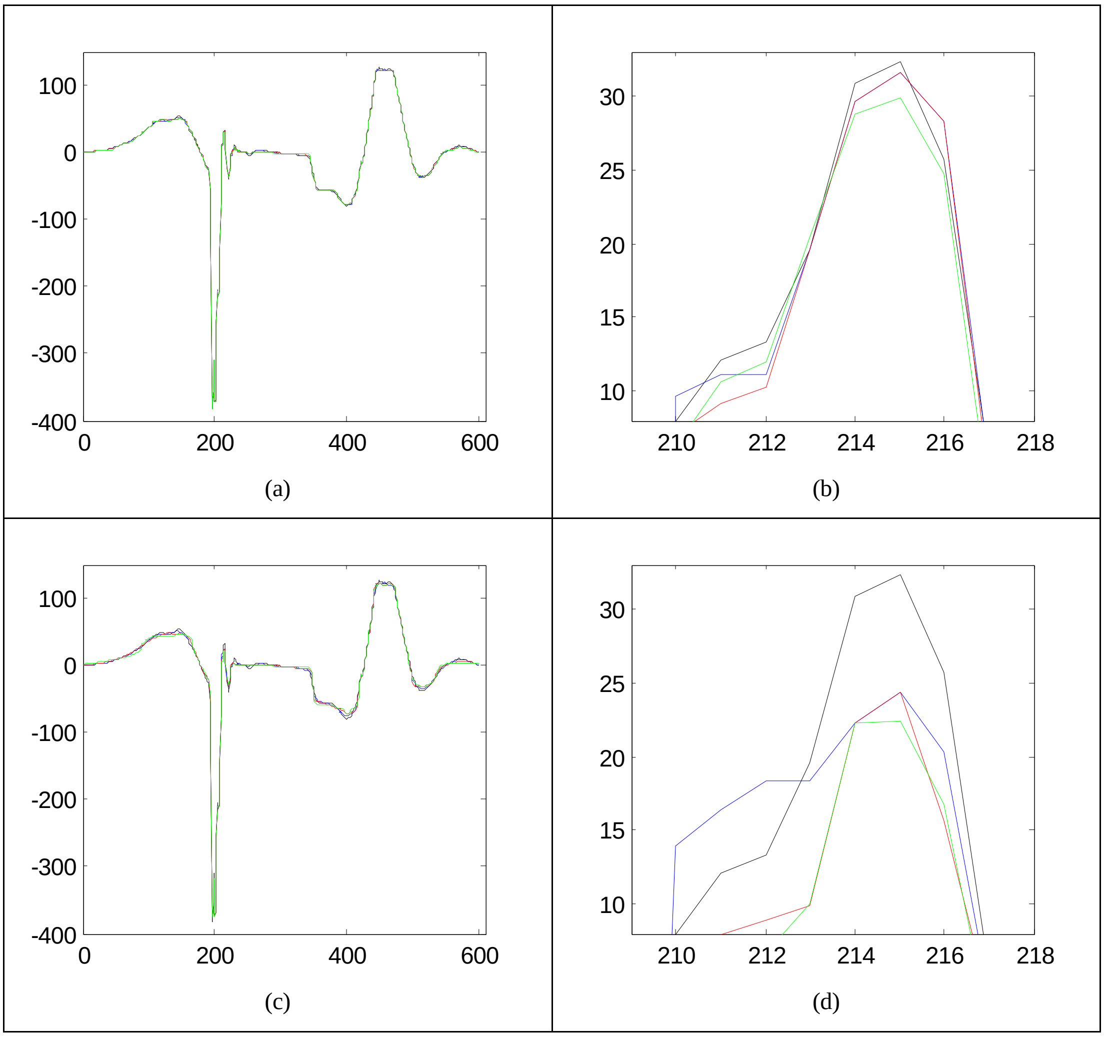


Figure 3.2.5: (Sigma Filtering), Sigma is constant.

(a) Smoothed Acceleration Data by using the values:
Black Line: The Original Data,
Blue Line: σ=3, Mask Size= 21,
Red Line: σ=3, Mask Size= 51,
Green Line: σ=3, Mask Size= 101.

(b) The samples, between 209 and 218, of Figure 3.2.5.a

(c) Smoothed Acceleration Data by using the values:
Black Line: The Original Data,
Blue Line: σ=10, Mask Size= 21,
Red Line: σ=10, Mask Size= 51,
Green Line: σ=10, Mask Size= 101.

(d) The samples, between 209 and 218, of Figure 3.2.5.c

3.2.3 Second Derivative Results for Sigma Filtering

The Second Derivative Result of Smoothed Displacement data can be seen in Figure 3.2.6 by using the Sigma filter for various σ values.

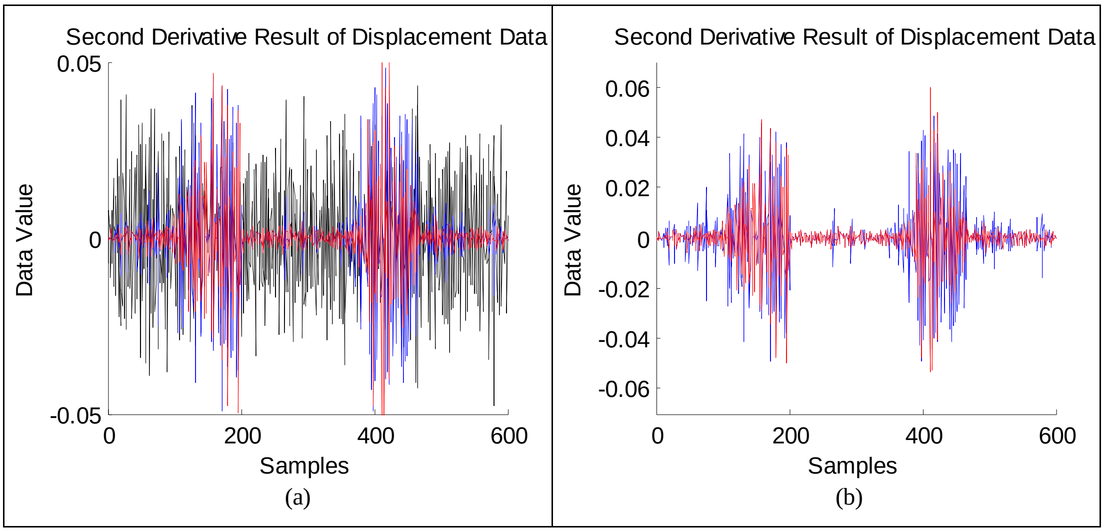


Figure 3.2.6: (Sigma Filtering). (a) Second Derivative Result of Smoothed **Displacement Data** using σ= 0.015, 0.01 and Mask Size = 7 **(b)** Same as Figure 3.2.6.a except Original Displacement Data (Black Line).

Black Line: Second Derivative of the Original Displacement Data,

Blue Line: Second Derivative of the Data by using the σ=0.01 and Mask Size= 7,

Red Line: Second Derivative of the Data by using the σ=0.015 and Mask Size= 7.

3.2.4 Comparing Sigma Filters of Varying Sigma and Mask Size Parameters

The Comparative Second Derivative Results of Smoothed Displacement data and Original Acceleration data are seen in Figure 3.2.7. The results of the filtered Displacement data can also be seen in Figure 3.2.2.d and Figure 3.2.3.d by using the same parameters.

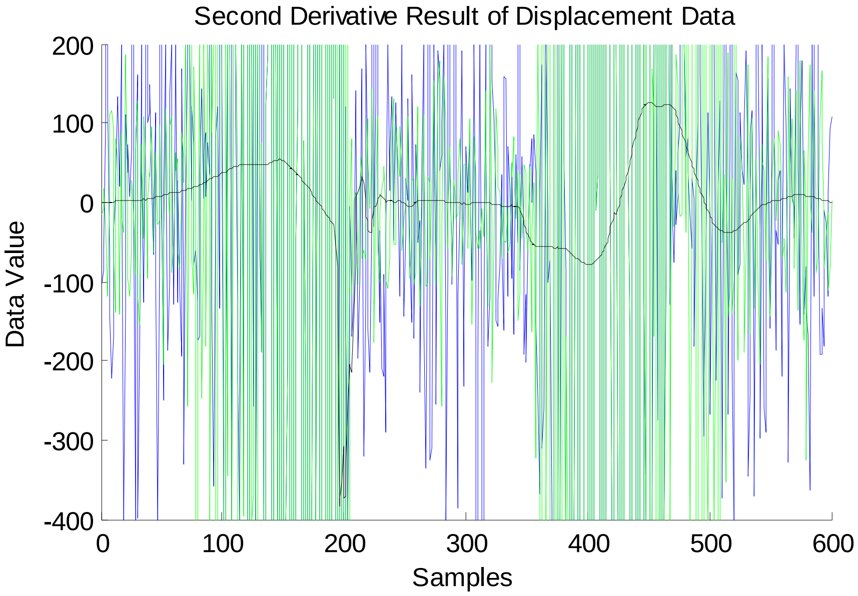


Figure 3.2.7: (Sigma Filtering). The Comparative Second Derivative Results of Smoothed Displacement Data and Original Acceleration Data.

Black Line: Original Acceleration Data

Blue Line: Second Derivative of Smoothed Displacement Data by using the σ=0.015 and Mask Size=7

Green Line: Second Derivative of Smoothed Displacement Data by using the σ=0.015 and Mask Size=15

3.3 MEDIAN FILTER

Median filter is a kind of Non-linear smoothing filter, which removes impulsive noise. Mask size, defined by the Median filter, defines the number of data, placed in the left and right sides of the center data. These data are sorted in terms of their values and then median value, which comes out after sorting, is written in the place of the centered data and this process is done through all the data [7, 8]. For example, if the filter size is defined as 5, taken two data from both in the right and in the left side of the centered data, whose value will be changed, and then these data will be put in order, according to their size. The third biggest data is taken and put into the place of the centered data and go right then the same process is applied to whole signal again and again.

For the data which have a lot of noise in it, Median filter may not give the expected result. However for the data which have less density impulsive noise, Median filter gives the expected results. As long as the density of the impulsive noise in our data increases; the results that are obtained by using Median filter decreases [9].

Median Filter Masks are:

Filter size=5

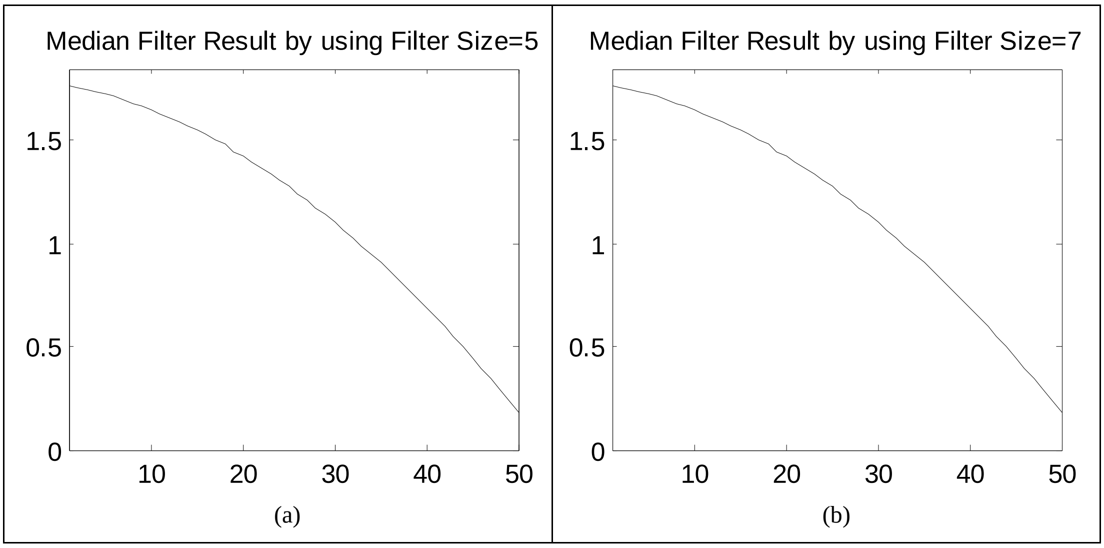
$$\begin{bmatrix} x_1 & x_2 & \boxed{x_3} & x_4 & x_5 \end{bmatrix} \tag{3.6}$$

Filter size=7

$$\begin{bmatrix} x_1 & x_2 & x_3 & \boxed{x_4} & x_5 & x_6 & x_7 \end{bmatrix} \tag{3.7}$$

3.3.1 Golf Ball Data Results for Median Filtering

Median filter removes impulsive noise. However as it is seen in Figure 3.3.1, the inputs, which have not impulsive noise, can not be corrected. After applying the Median filter by using (3.6) and (3.7), no change in the result value of the Golf ball data occurs. The reason for this is that there are not sudden changes but small noises in the data.



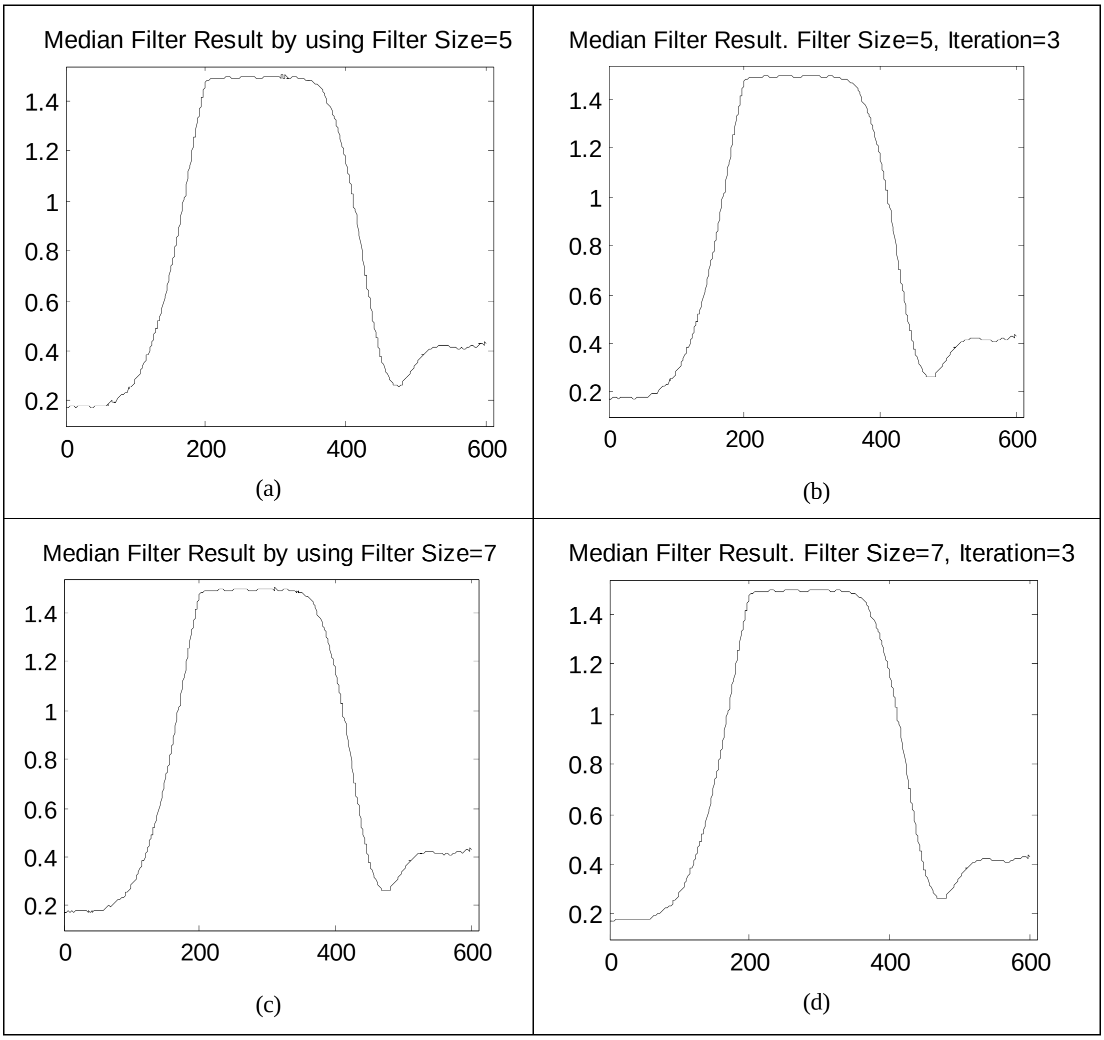
(a)

(b)

Figure 3.3.1: (Median Filtering). (a) Filter Size=5, **(b)** Filter Size=7.

3.3.2 Displacement Data Results for Median Filtering

After applying the Median filter to the Displacement data, in Figure 1.1.2.a, the results are in Figure 3.3.2.



(a) (b) (c) (d)

Figure 3.3.2: (Median Filtering). (a) Filter Size=5, **(b)** Filter Size=5 & Iteration number=3, **(c)** Filter Size=7, **(d)** Filter Size=7 & Iteration number=3.

3.3.3 Acceleration Data Results for Median Filtering

After applying the Median filter to the Acceleration data, in Figure 1.1.2.b, the results are in Figure 3.3.3.

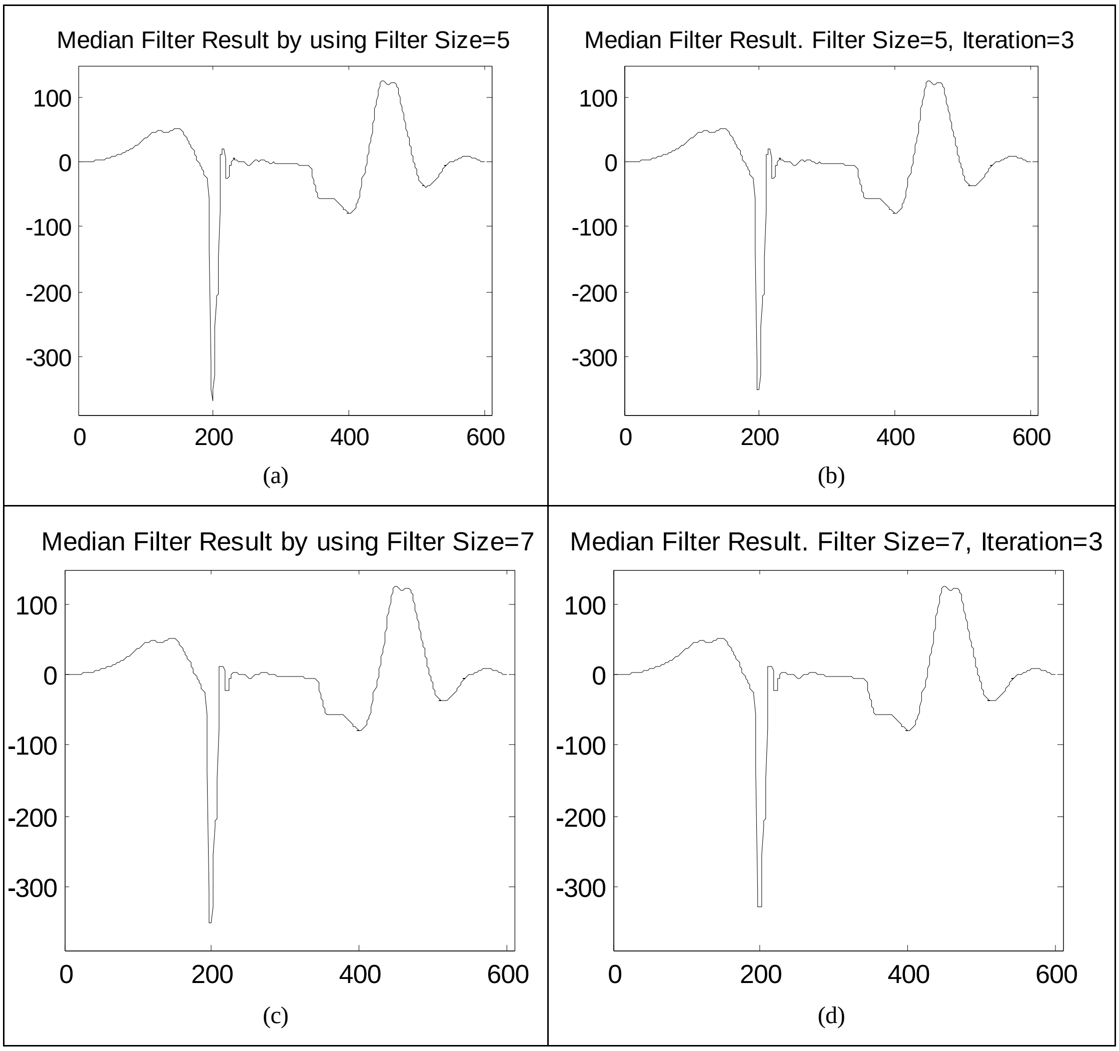


Figure 3.3.3: (Median Filtering). (a) Filter Size=5, **(b)** Filter Size=5 & Iteration number=3, **(c)** Filter Size=7, **(d)** Filter Size=7 & Iteration number=3.

As it is seen from the results of Median filter that is applied to the displacement and the Acceleration data, in the area of impulsive noise, Median filter corrected these noises. The iterations of the filter can make the result better, however after some iterations, the result does not change. At this point, where the result does not change, filter size should be increased instead of continuing to iterations in order to continue filtering,. So, Median filter can continue giving better results. While, increase in the filter size means the decrease in fluctuations in the signal, on the other hand, it means increase in loss in data. Instead of this, if filter size is kept small and iteration number is increased, better results can be obtained. However, if filter size is kept too small, it makes us get an output which is not debugged from the noise enough, and this is an unexpected situation. As a result, in order to get a good result, these two criteria (filter size and iteration number) should be balanced very well.

Keeping the filter size as 21 and iteration number as 7, better results than Figure 3.3.3 can be obtained in Figure 3.3.4. However, the result in which the same values are used but Acceleration data is also used are worse than the results before.

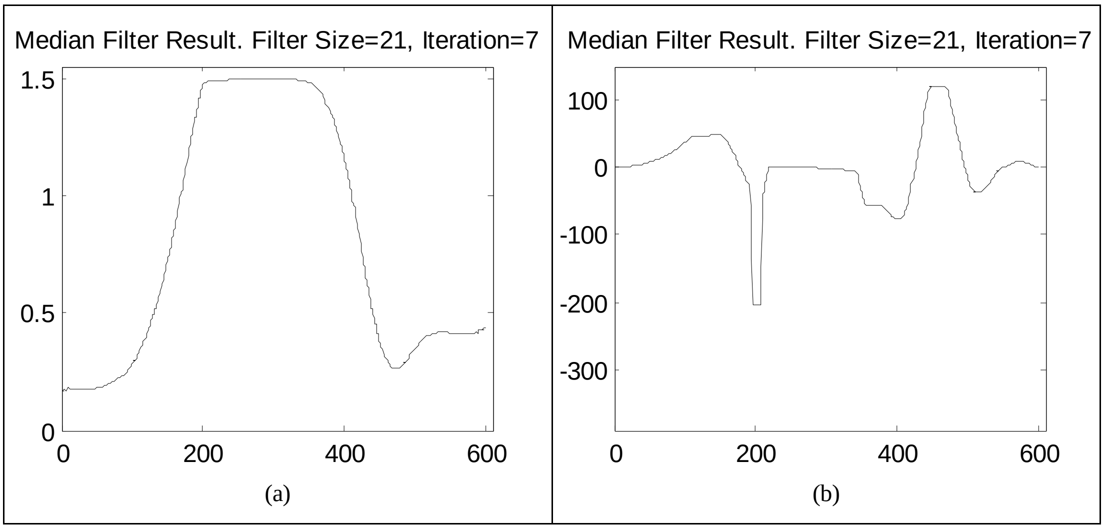


Figure 3.3.4: (Median Filtering).

(a) Smoothed Displacement Data by using Filter Size=21 and Iteration number=7,
(b) Smoothed Acceleration Data by using Filter Size=21 and Iteration number=7.

3.3.4 Second Derivative Results for Median Filtering

Second Derivative Result of Smoothed Displacement data can be seen in Figure 3.3.5 and in Figure 3.3.6 by using the Median filter for various filter size and iteration.

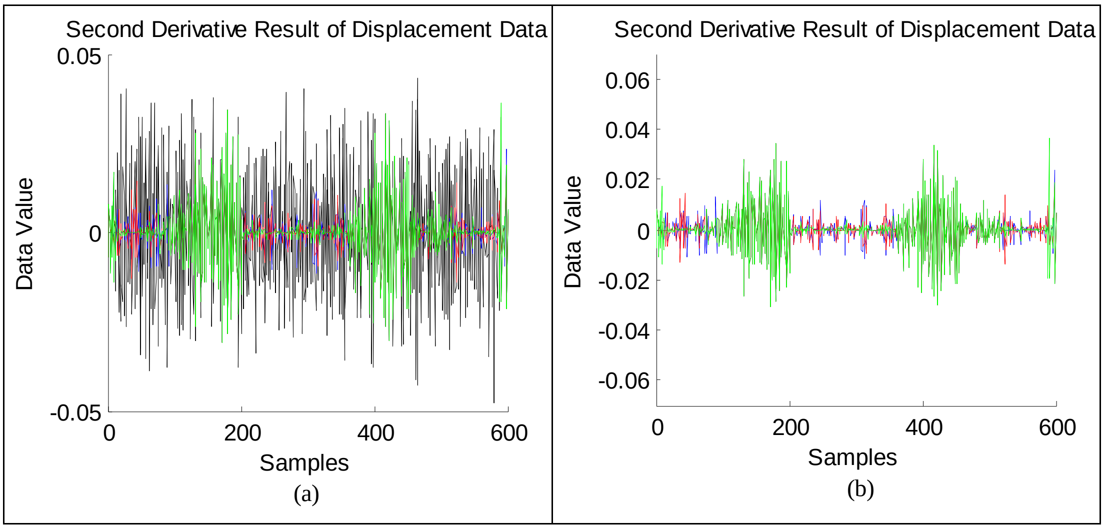


Figure 3.3.5: (Median Filtering). (a) Second Derivative Result of Smoothed **Displacement Data** using Filter Size = 5, 7, 21 **(b)** Same as Figure 3.3.5.a except Original Displacement Data (Black Line).

Black Line: Second Derivative of the Original Displacement Data,
Blue Line: Second Derivative of the Data by using the Filter Size= 5,
Red Line: Second Derivative of the Data by using the Filter Size= 7,
Green Line: Second Derivative of the Data by using the Filter Size= 21.

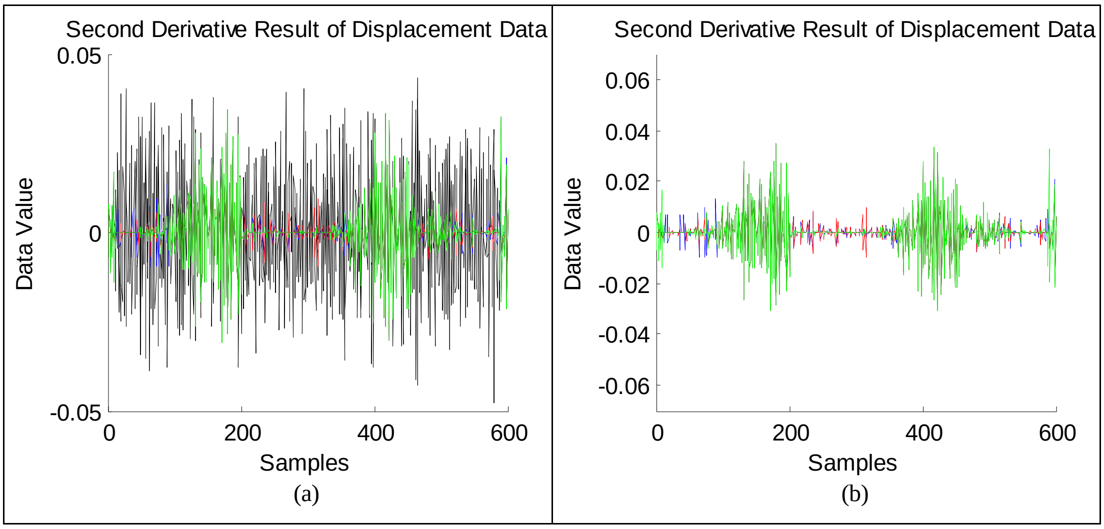


Figure 3.3.6: (Median Filtering). (a) Second Derivative Result of Smoothed **Displacement Data** using Filter Size = 5, 7, 21 for 3 iteration **(b)** Same as Figure 3.3.6.a except Original Displacement Data (Black Line).

Black Line: Second Derivative of the Original Displacement Data,
Blue Line: Second Derivative of the Data by using the Filter Size= 5 for 3 iteration,
Red Line: Second Derivative of the Data by using the Filter Size= 7 for 3 iteration,
Green Line: Second Derivative of the Data by using the Filter Size= 21 for 3 iteration.

3.3.5 Comparing Median Filters of Varying Filter Size and Iteration Parameters

The Comparative Second Derivative Results of Smoothed Displacement data and Original Acceleration data are seen in Figure 3.3.7 and in Figure 3.3.8.

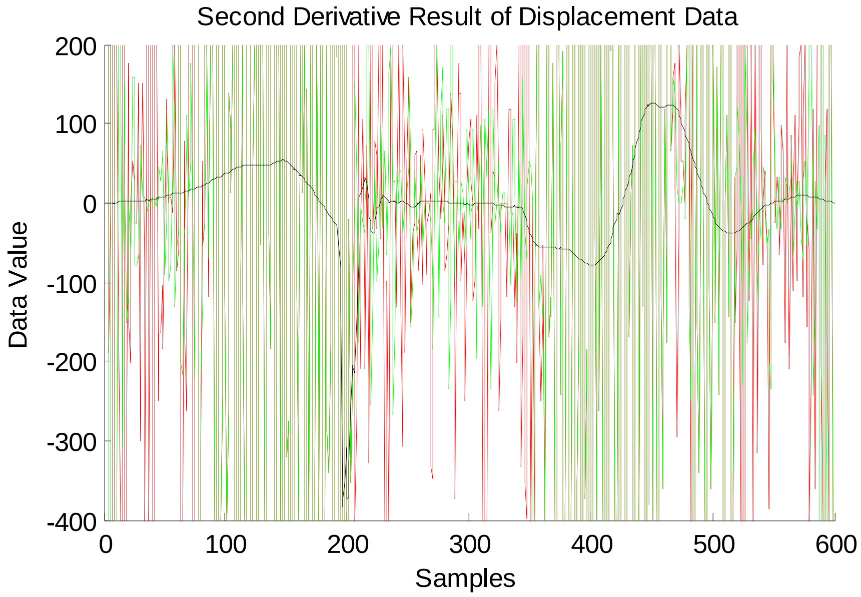


Figure 3.3.7: (Median Filtering). The Comparative Second Derivative Results of Smoothed Displacement Data and Original Acceleration Data. Black Line is the Original Acceleration Data.

Red Line: Second Derivative of Smoothed Displacement Data by using Filter Size=7

Green Line: Second Derivative of Smoothed Displacement Data by using Filter Size=21

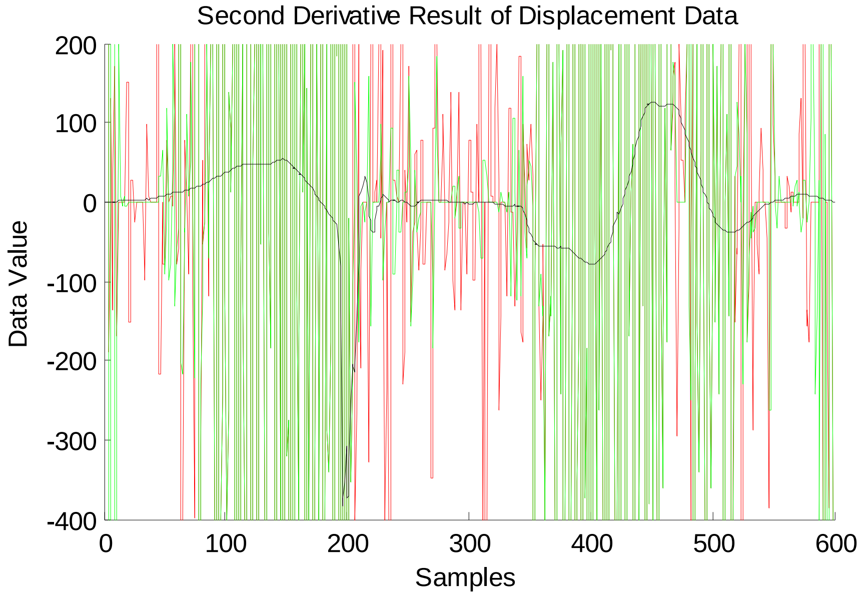


Figure 3.3.8: (Iterative Median Filtering). The Comparative Second Derivative Results of Smoothed Displacement Data and Original Acceleration Data. Black Line is the Original Acceleration Data.

Red Line: Second Derivative of Smoothed Displacement Data by using Filter Size=7 for 3 iteration

Green Line: Second Derivative of Smoothed Displacement Data by using Filter Size=21 for 3 iteration

3.4 PERONA MALIK FILTER

Diffusion processes derive from Fick’s law [10]. Smoothing is the result of a diffusion process as mentioned in Linear Diffusion filtering. Diffusion is a physical process which balances the concentration difference without destroying the data, and smoothing process can be considered as a diffusion process. Perona-Malik Diffusion is the space variant smoothing filter depending on the data content [11]. This technique reduces the high frequency components, like noise, without removing significant parts of the data content.

Diffusion equation can be considered as [12]:

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}(C.\frac{\partial}{\partial x}u) \qquad (3.8)$$

where C is the diffusion coefficient, u is the input data. If the value of C=1 then the result will be the same as the Linear Diffusion filter, and Diffusion Equation will be

$$\frac{\partial u}{\partial t} = \frac{\partial^2}{\partial x^2}u \qquad (3.9)$$

The idea of Perona Malik is making the diffusivity signal dependent. The value of C must be between 0 and 1. When the process is near significant parts of the data, the value of C converges to 0 in the other parts of the data it converges to 1.

The value of C is determined by using the magnitude of the first derivative

$$C = g(\frac{\partial}{\partial x}u)^2 \qquad (3.10)$$

By using (3.10) in (3.8), anisotropic Perona Malik Diffusion Equation is [13].

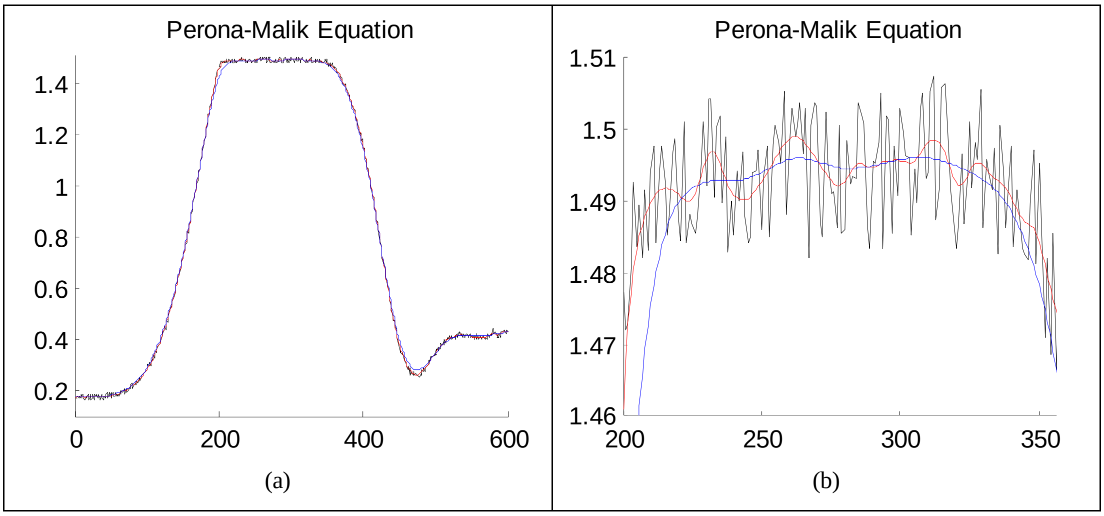
$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}(g(\frac{\partial}{\partial x}u)^2) \qquad (3.11)$$

Where

$$g(\frac{\partial}{\partial x}u) = \frac{1}{(1+(\frac{(\frac{\partial}{\partial x}u)^2}{\lambda^2}))} \qquad (3.12)$$

3.4.1 Displacement Data Results for Perona Malik Filtering

After applying the Perona Malik filter to the Displacement data, Figure 1.1.2.a, the results are in Figure 3.4.1 and in Figure 3.4.2.



(a)

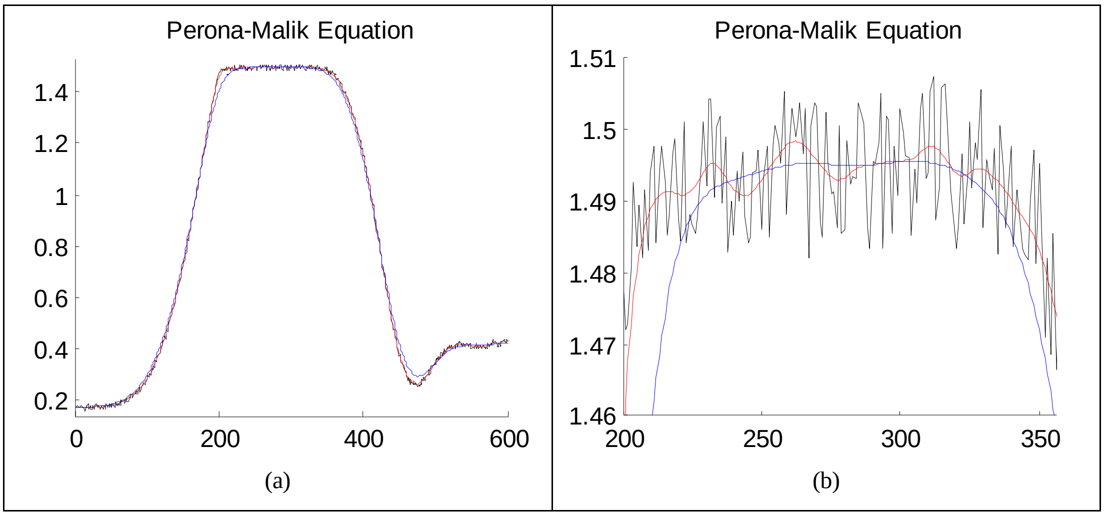
(b)

Figure 3.4.1: (Perona Malik Filtering), (a) Smoothed Displacement Data Result, **(b)** The samples, between 200 and 360, of Figure 3.4.1.a.

Black Line: The Original Displacement Data,

Red Line : Smoothed Displacement Data by using the λ=10, T=5, Δt=0.25 and *σ=3*,

Blue Line : Smoothed Displacement Data by using the λ=10, T=50, Δt=0.05 and *σ=3*.



(a)

(b)

Figure 3.4.2: (Perona Malik Filtering), (a) Smoothed Displacement Data Result, **(b)** The samples, between 200 and 360, of Figure 3.4.2.a.

Black Line: The Original Displacement Data,

Red Line : Smoothed Displacement Data by using the λ=10, T=10, Δt=0.25 and *σ=3*,

Blue Line : Smoothed Displacement Data by using the λ=80, T=100, Δt=0.25 and *σ=1*.

3.4.2 Acceleration Data Results for Perona Malik Filtering

After applying the Perona Malik filter to the Acceleration data, in Figure 1.1.2.b, the results are in Figure 3.4.3 and in Figure 3.4.4.

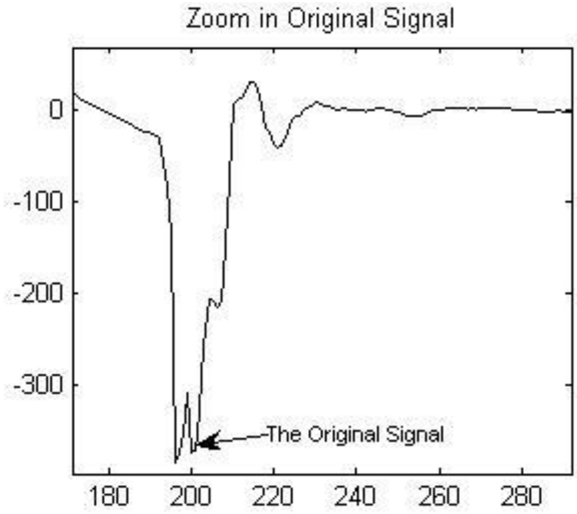


Figure 3.4.3: (Original Acceleration Data between the samples of 176 and 284)

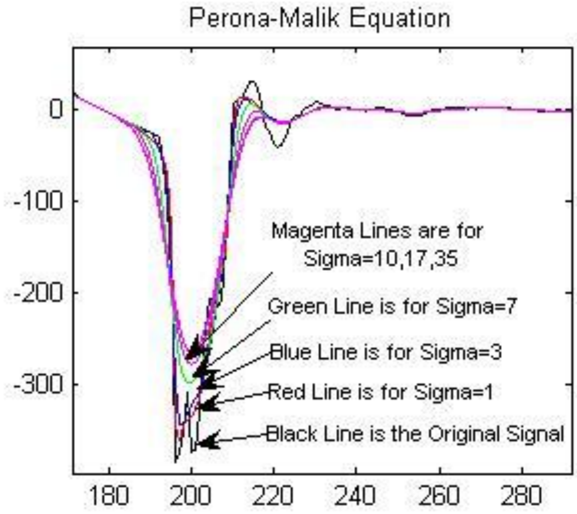


Figure 3.4.4: (Perona Malik Filtering). Results between the samples of 176 and 284 for λ=10, T=10, Δt=0.25 and various *σ=1, 3, 7, 10, 17, 35*.

λ and σ are the parameters of Regularized Perona-Malik beside Δt and T.

Sigma (σ) which is a scale parameter in the Regularized Perona-Malik equation can be seen as a smoothing factor.

As it is seen in Figure 3.4.3, there are changes only in curvature points, but actually there is no change on straight lines. It means, this is a Non-linear smoothing filter. After a while the signal doesn't change so much, also if the value of σ is increased to some larger value for the fix λ.

Lambda (λ) term in Perona Malik equation can be thought as a contrast parameter separating regions of forward diffusion from regions of backward diffusion. It decides on whether an edge or curve in the diffusion process is preserved or not.

To see the effect of the λ, Regularized Perona-Malik is applied to Acceleration data for some various λ values by keeping σ, Δt and T constant, in Figure 3.4.5.

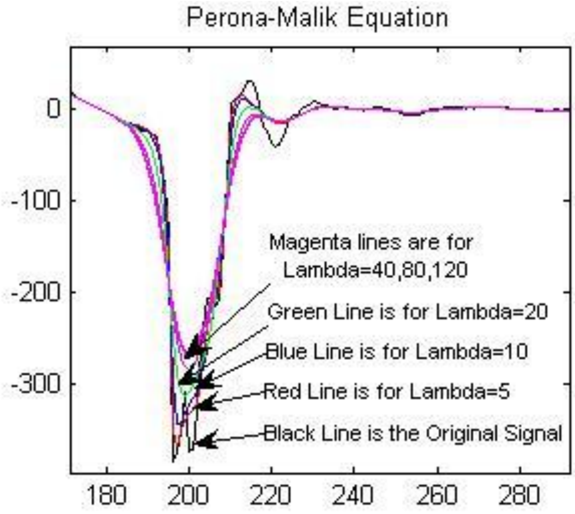


Figure 3.4.5: (Perona Malik Filtering). Results between the samples of 176 and 284 for σ= 3, T=10, Δt=0.25 and various *λ=5, 10, 20, 40, 80, 120*.

After a definite value of λ, the effect of the λ becomes smaller even if the value of λ is set to a larger value for the fix σ. So the effect of σ and λ does not increase linearly by their values.

The mean, variance, standard deviation, total gradient and entropy are examined, Figure 3.4.7, by using the parameters σ=3, λ=10, T=50 and Δt=0.05 on the Acceleration data signal by applying Perona-Malik, Figure 3.4.6.

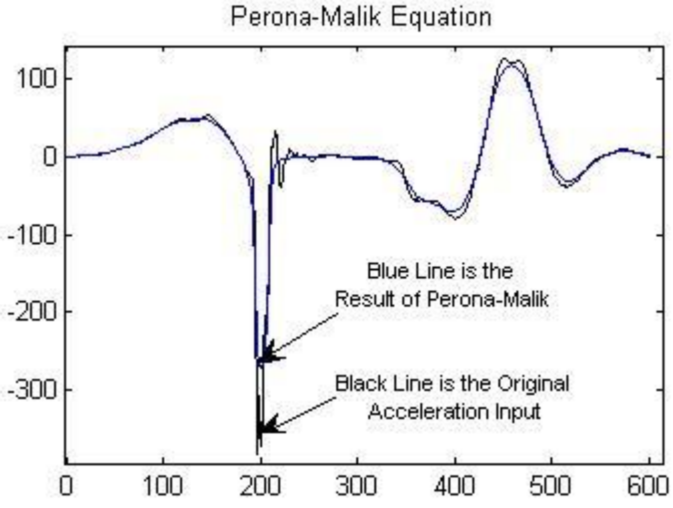


Figure 3.4.6: (Perona Malik Filtering).) Smoothed Acceleration Data by using σ=3, λ=10, T=50 and Δt=0.05.

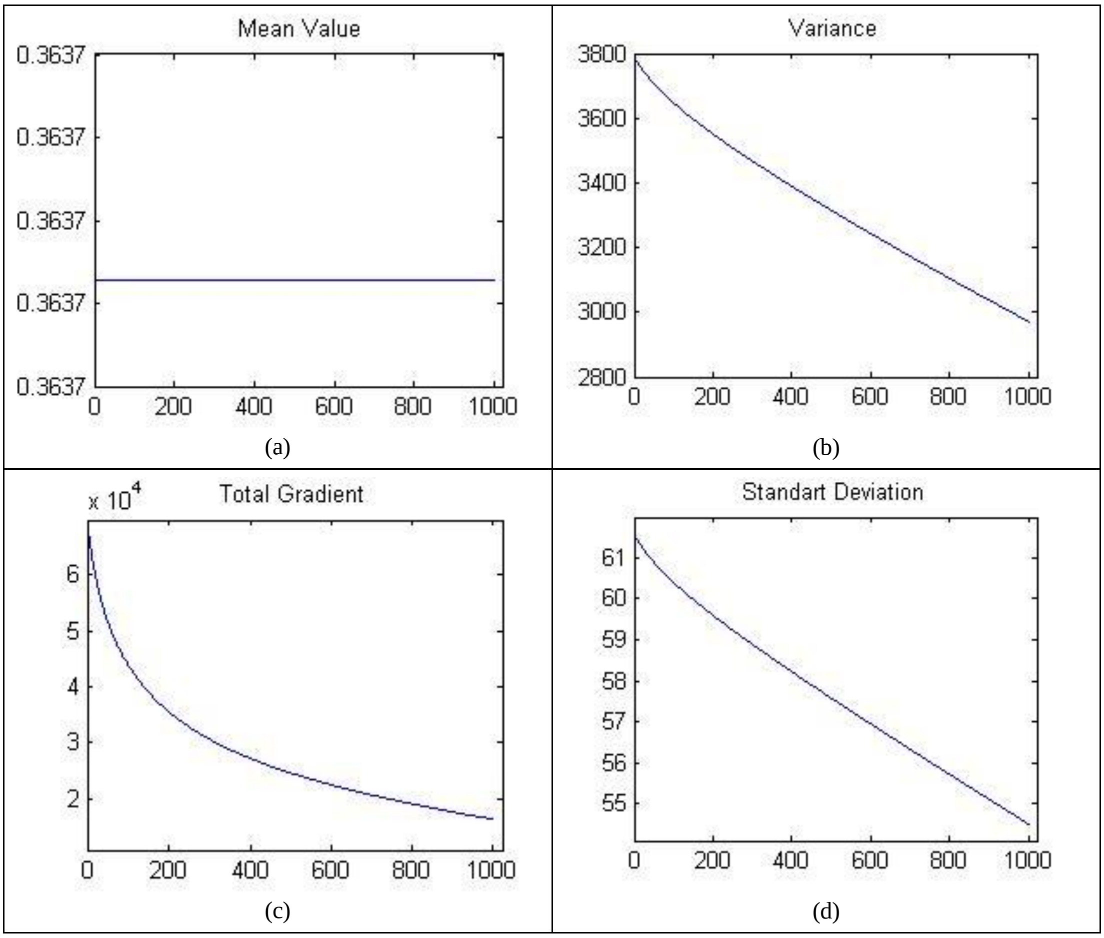
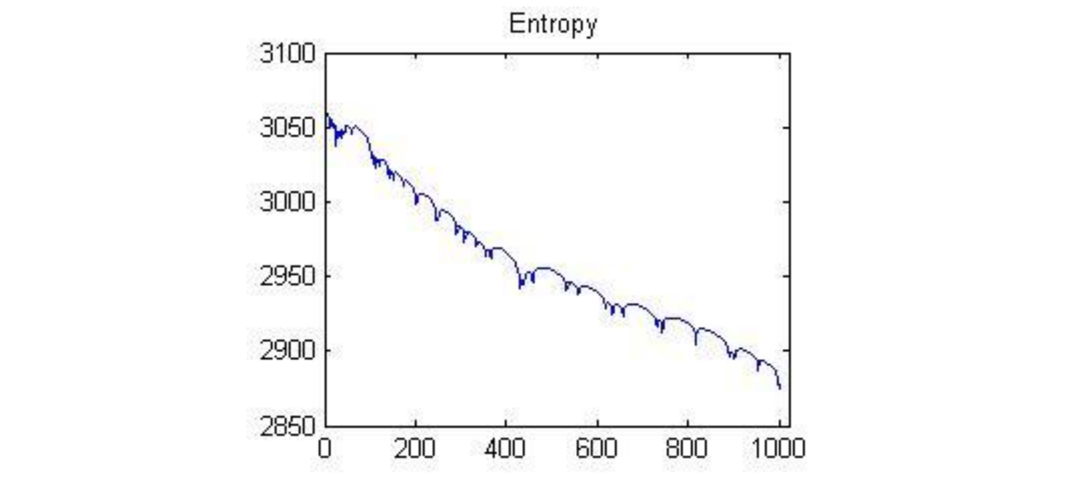


Figure 3.4.7: (Perona Malik Filtering).
(a) Mean Value vs. Iteration for σ=3, λ=10, T=50 and Δt=0.05
(b) Variance vs. Iteration for σ=3, λ=10, T=50 and Δt=0.05
(c) Total Gradient vs. Iteration for σ=3, λ=10, T=50 and Δt=0.05
(d) Standard Deviation vs. Iteration for σ=3, λ=10, T=50 and Δt=0.05
(e) Entropy vs. Iteration for σ=3, λ=10, T=50 and Δt=0.05



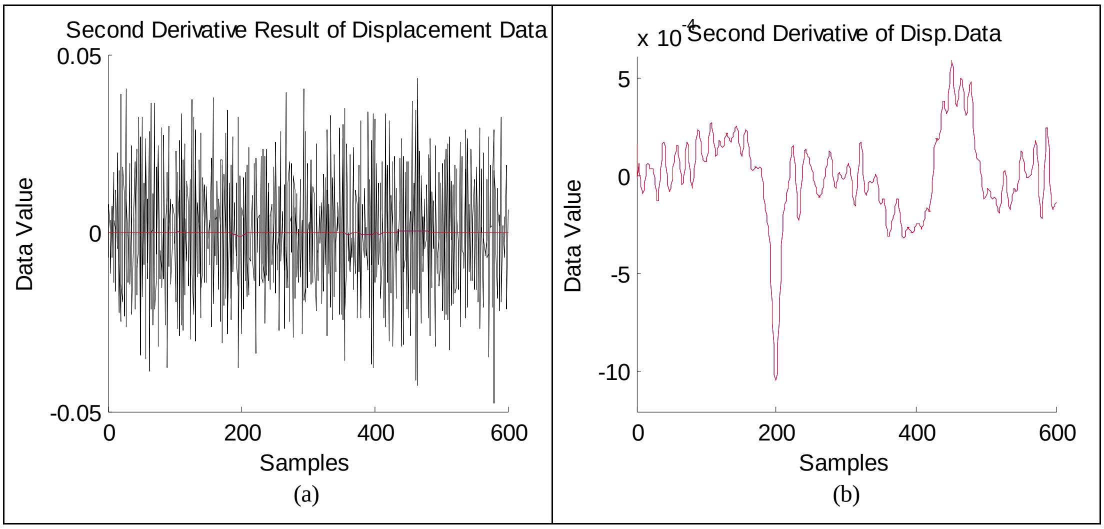
(e)

Figure 3.4.7: (Continued).

When the Perona-Malik filter is applied to a signal, for the each iteration, mean value of the signal doesn't change. Variance, standard deviation and total gradient, $\Sigma|\Delta|^2$ monotonically decrease.

3.4.3 Second Derivative Results for Perona Malik Filtering

Second Derivative Result of Smoothed Displacement data can be seen in Figure 3.4.8 by using Perona Malik filter.



(a)

(b)

Figure 3.4.8: (Perona Malik Filtering). (a) Second Derivative Result of Smoothed **Displacement Data (b)** Same as Figure 3.4.8.a except Original Displacement Data (Black Line).

Black Line: Second Derivative of the Original Displacement Data,

Blue Line: Second Derivative of the Data by using the λ=10, T=10, Δt=0.25 and σ=3,

Red Line: Second Derivative of the Data by using the λ=80, T=10, Δt=0.25 and σ=1.

3.4.4 Comparing Perona Malik Filters of Varying Parameters

The Comparative Second Derivative Results of Smoothed Displacement data and Original Acceleration data are seen in Figure 3.4.9. Perona Malik filtering result, blue line in Figure 3.4.9, gives comparatively better result in the area of 2, because it is successfully got rid of the noise, but it is not successful in the area of 1, because the critical feature is lost. The result of filtering, red line, gives good result in the area of 1 while giving bad result in the area of 2. The results of the smoothed Displacement data can also be seen in Figure 3.4.1 by using the same parameters.

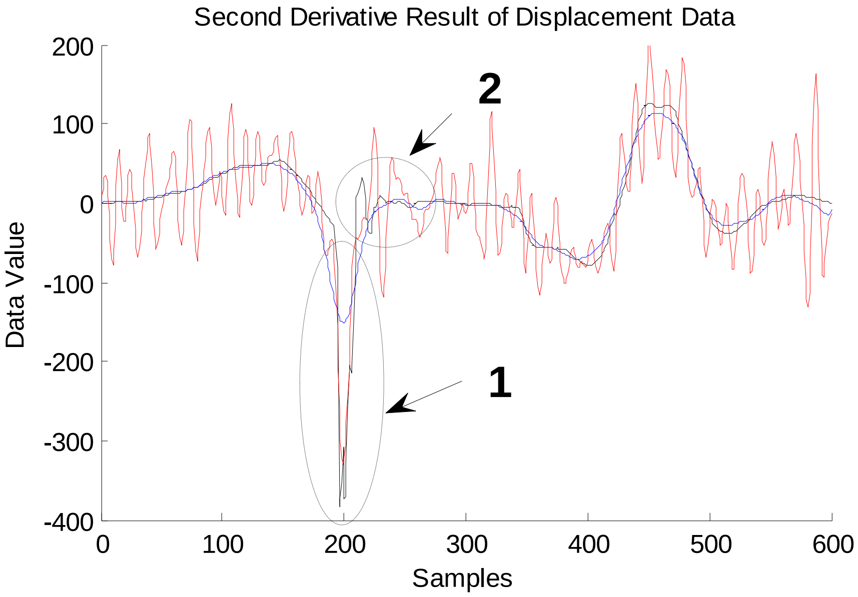


Figure 3.4.9: (Perona Malik Filtering). The Comparative Second Derivative Results of Smoothed Displacement Data and Original Acceleration Data.

Black Line: Original Acceleration Data

Red Line: Second Derivative of Smoothed Displacement Data by using λ=10, T=5, Δt=0.25 and σ=3,

Blue Line: Second Derivative of Smoothed Displacement Data by using λ=10, T=50, Δt=0.05 and σ=3.

3.5 THE AMBROSIO-TORTORELLI APPROXIMATION OF THE MUMFORD-SHAH MODEL

Ambrosio-Tortorelli is using smoothing based Partial Differential Equation by using coupled Partial Differential Equation to solve the diffusion process [14]:

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x} \cdot ((1-v)^2 (\frac{\partial}{\partial x} u)) - \frac{\beta}{\alpha}(u-g) \tag{3.13}$$

$$\frac{\partial v}{\partial t} = \frac{\partial^2}{\partial x^2} v + \frac{2\alpha(\frac{\partial}{\partial x} u)^2}{\rho}(1-v) - \frac{v}{\rho^2} \tag{3.14}$$

Parameters in this Partial Differential Equation are:

g: is the input data.
u: is the nonlinearly smoothed version of input data.
v: is an estimate of the first derivative.

v is the result of regularized first derivative function. The regularization of v can be ignored:

$$v \approx \frac{2\alpha\rho(\frac{\partial}{\partial x} u)^2}{1 + 2\alpha\rho(\frac{\partial}{\partial x} u)^2} \tag{3.15}$$

Using the equation (3.13):

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x} \cdot \boldsymbol{((1-v)^2 (\frac{\partial}{\partial x} u))} - \frac{\beta}{\alpha}(u-g) \tag{3.16}$$

Bold part can be written as:

$$\frac{\partial}{\partial x} \cdot (g((\frac{\partial}{\partial x} u)^2) \frac{\partial}{\partial x} u) \tag{3.17}$$

where

$$g(\frac{\partial}{\partial x} u) = \frac{1}{1 + 2\alpha\rho^2 (\frac{\partial}{\partial x} u)^2} \tag{3.18}$$

Perona Malik Diffusion Equation, in (3.12), was:

$$g(\frac{\partial}{\partial x}u)=\frac{1}{1+(\frac{(\frac{\partial}{\partial x}u)^2}{\lambda^2})} \tag{3.19}$$

So, in Perona Malik Diffusion, when λ is replaced with

$$\lambda=\sqrt{\frac{1}{2\cdot\alpha\cdot\rho^2}} \tag{3.20}$$

Then the result will be the Ambrosio Tortorelli Partial differential Equation [14].

$$\frac{\partial}{\partial t}u=\frac{\partial}{\partial x}\cdot((\frac{1}{1+\frac{(\frac{\partial}{\partial x}u)^2}{(\sqrt{\frac{1}{2\cdot\alpha\cdot\rho^2}})^2}})^2\frac{\partial}{\partial x}u)-\frac{\beta}{\alpha}(u-g) \tag{3.21}$$

Equation (3.21) is equal to:

$$\frac{\partial}{\partial t}u=\frac{\partial}{\partial x}\cdot((\frac{1}{1+2\cdot\alpha\cdot\rho^2(\frac{\partial}{\partial x}u)^2})^2\frac{\partial}{\partial x}u)-\frac{\beta}{\alpha}(u-g) \tag{3.22}$$

where $\frac{\beta}{\alpha}$ is the stopping criteria. Equation (3.23) is about the time to stop.

$$\frac{\alpha}{\beta}=\frac{\sigma^2}{2} \tag{3.23}$$

3.5.1 Acceleration Data Results for Ambrosio Tortorelli Approximation of the Mumford Shah

After applying the Ambrosia Tortorelli to the Acceleration data, in Figure 1.1.2.b, following results are obtained.

Seeing the result of Ambrosio Tortorelli, when $\frac{\alpha}{\beta}$ is constant, and change the value of ρ:

If parameter ρ goes to 0, Ambrosio Tortorelli filter works better, as in Figure 3.5.1, Figure 3.5.2 and Figure 3.5.3. When the parameter ρ decreases, the signal smoothes and less noise appears.

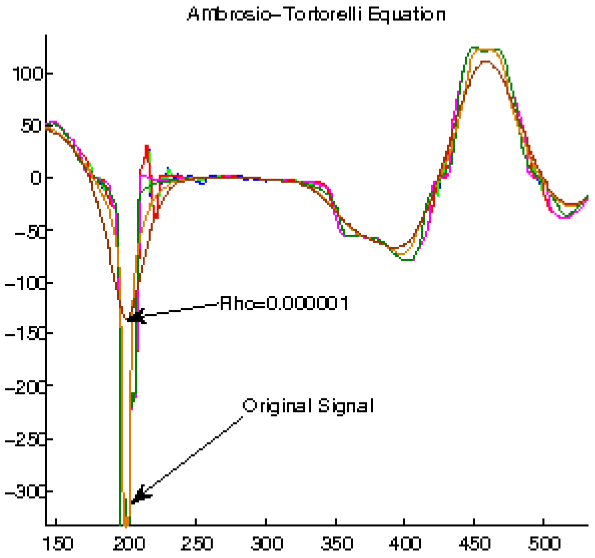


Figure 3.5.1: (Result for the Ambrosio Tortorelli Approximation of the Mumford Shah)
The results for T=100, Δt=0.4, α=8, β=0.1, ρ= 0.05, 0.01, 0.001, 0.0001, 0.00001, 0.000001

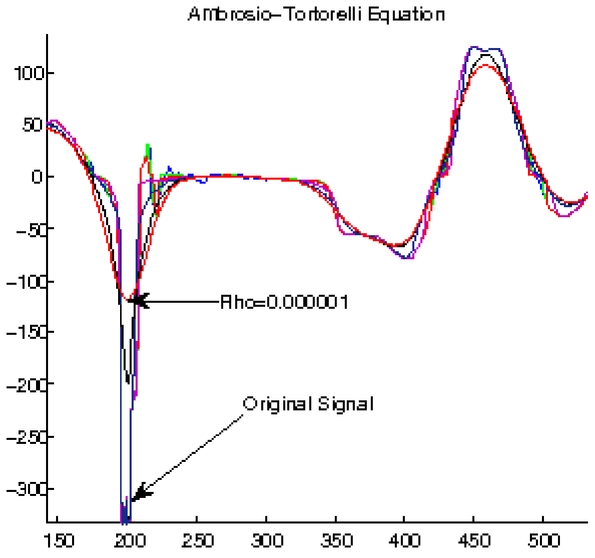


Figure 3.5.2: (Result for the Ambrosio Tortorelli Approximation of the Mumford Shah)
The results for T=100, Δt=0.4, α=2, β=0.025, ρ= 0.05, 0.01, 0.001, 0.0001, 0.00001, 0.000001

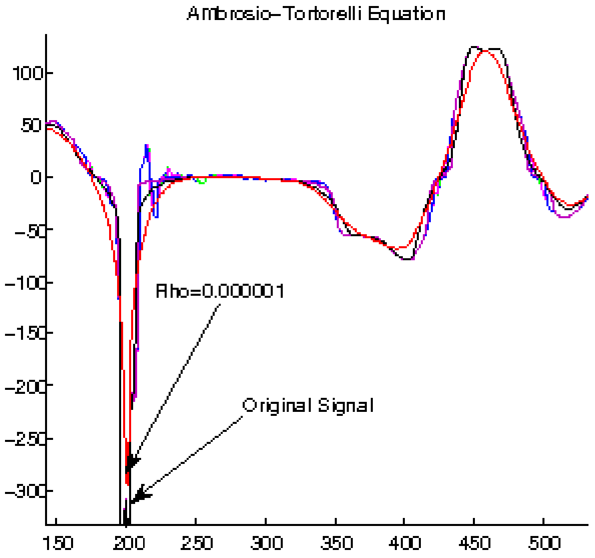


Figure 3.5.3: (Result for the Ambrosio Tortorelli Approximation of the Mumford Shah)
The results for T=100, Δt=0.4, α=32, β=0.4, ρ= 0.05, 0.01, 0.001, 0.0001, 0.00001, 0.000001

After changing the value of α, the result of Ambrosio Tortorelli is as in Figure 3.5.4.

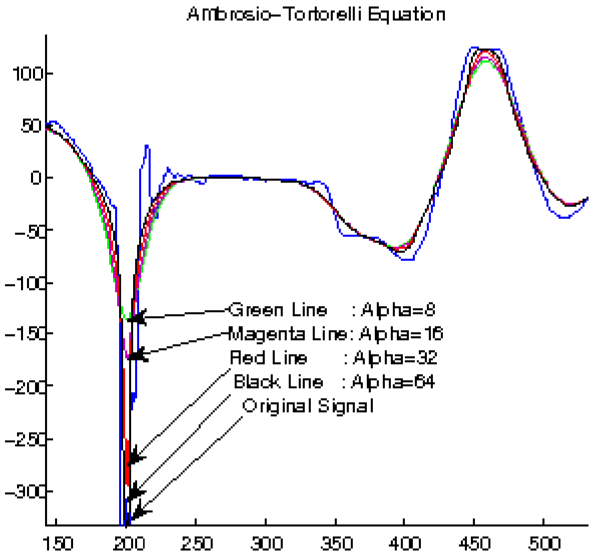


Figure 3.5.4: (Result for the Ambrosio Tortorelli Approximation of the Mumford Shah)
The results for T=100, Δt=0.4, α=8, 16, 32, 64, β=0.025, ρ= 0.000001.

After changing the value of β, the result of Ambrosio Tortorelli is as in Figure 3.5.5 and in Figure 3.5.6.

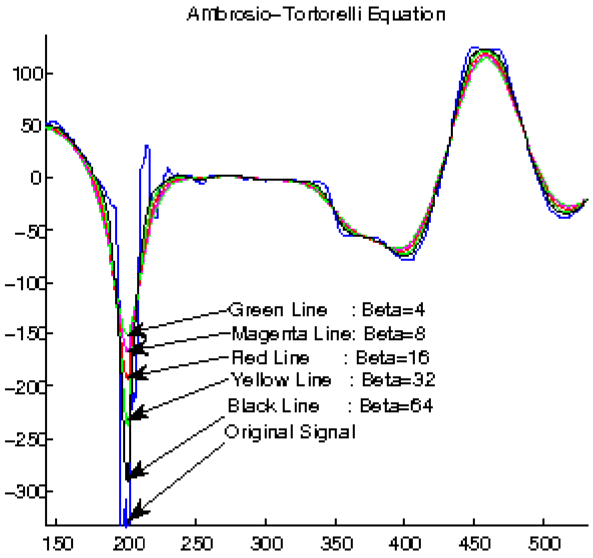


Figure 3.5.5: (Result for the Ambrosio Tortorelli Approximation of the Mumford Shah)
The results for T=100, Δt=0.4, α=8, β=4, 8, 16, 32, 64, ρ= 0.000001

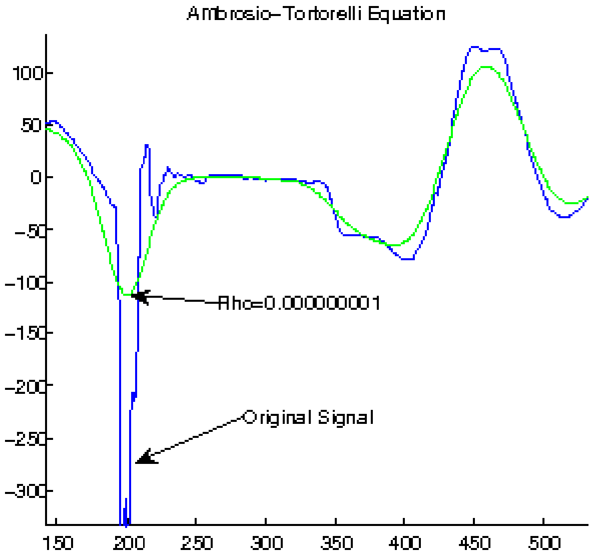
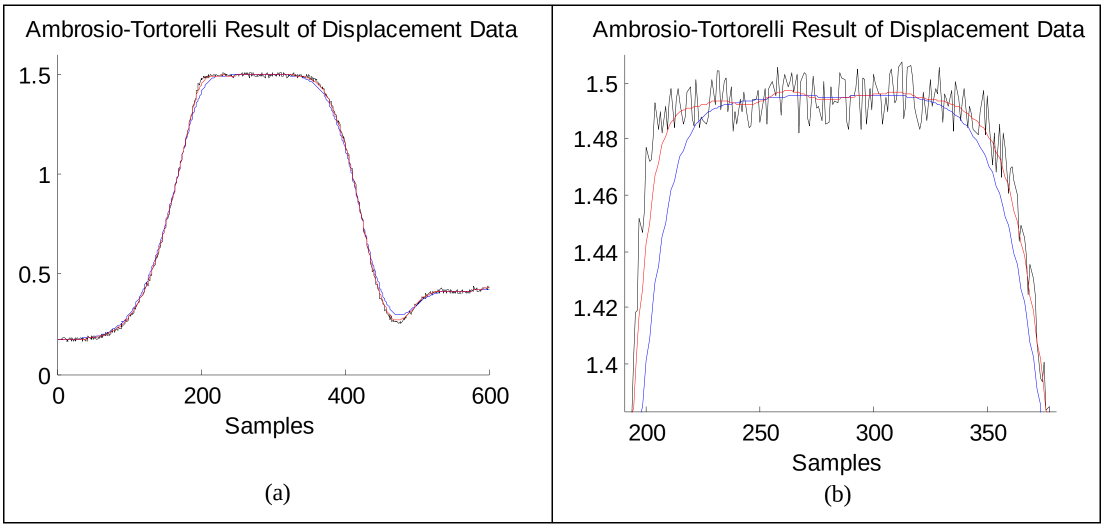


Figure 3.5.6: (Result for the Ambrosio Tortorelli Approximation of the Mumford Shah)
The result for T=100, Δt=0.4, α=8, β=0.1, ρ= 0.000000001

3.5.2 Displacement Data Results for Ambrosio Tortorelli Approximation of the Mumford Shah

After applying the Ambrosia Tortorelli to the Displacement data, Figure 1.1.2.a, the results are in Figure 3.5.7 and in Figure 3.5.8.



(a)

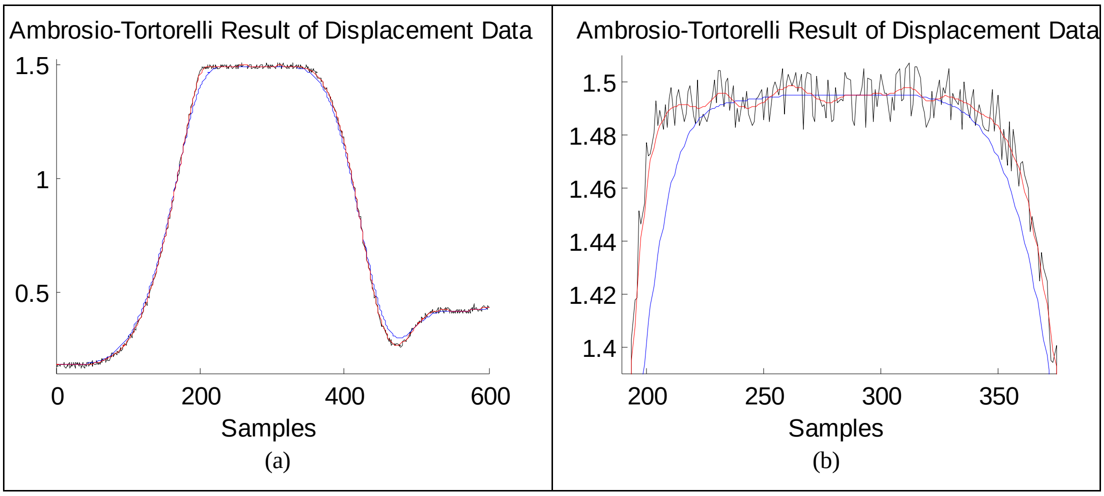
(b)

Figure 3.5.7: (Result for the Ambrosio Tortorelli Approximation of the Mumford Shah), (a) Smoothed Displacement Data Result, **(b)** The samples, between 190 and 380, of Figure 3.5.7.a.

Black Line : The Original Displacement Data,

Blue Line : Smoothed Displacement Data by using the T=100, Δt=0.4, α=64, β=0.025, ρ=0.000001,

Red Line : Smoothed Displacement Data by using the T=100, Δt=0.4, α=8, β=64, ρ=0.000001.



(a)

(b)

Figure 3.5.8: (Result for the Ambrosio Tortorelli Approximation of the Mumford Shah), (a) Smoothed Displacement Data Result, **(b)** The samples, between 190 and 380, of Figure 3.5.8.a.

Black Line : The Original Displacement Data,

Blue Line : Smoothed Displacement Data by using the T=100, Δt=0.4, α=64, β=0.025, ρ=0.000001,

Red Line : Smoothed Displacement Data by using the T=100, Δt=0.4, α=8, β=256, ρ=0.000001.

3.5.3 Second Derivative Results for Ambrosio Tortorelli Approximation of the Mumford Shah

The Second Derivative Result of Smoothed Displacement data can be seen in Figure 3.5.9 by using the Ambrosio Tortorelli approximation of the Mumford Shah.

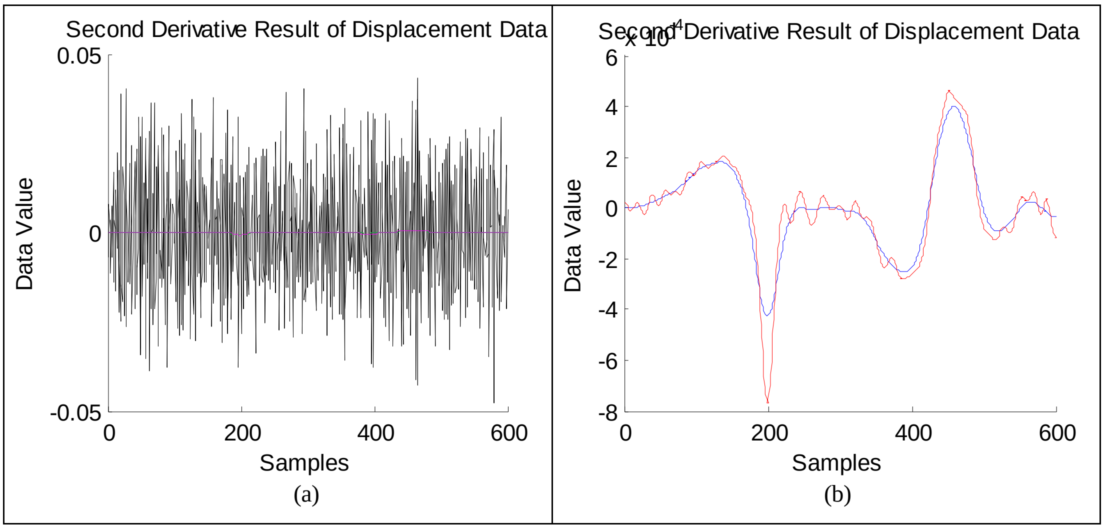


Figure 3.5.9: (Ambrosio Tortorelli Approximation of the Mumford Shah). (a) Second Derivative Result of Smoothed **Displacement Data (b)** Same as Figure 3.5.9.a except Original Displacement Data (Black Line).

Black Line: Second Derivative of the Original Displacement Data,

Blue Line: Second Derivative of the Data by using the T=100, Δt=0.4, α=64, β=0.025, ρ=0.000001,

Red Line: Second Derivative of the Data by using the T=100, Δt=0.4, α=8, β=256, ρ=0.000001.

3.5.4 Comparing Ambrosio Tortorelli Approximation of Mumford Shah of Varying Parameters

The Comparative Second Derivative Results of Smoothed Displacement data and Original Acceleration data are seen in Figure 3.5.10 and Figure 3.5.11. Ambrosio Tortorelli approximation of the Mumford Shah result, blue line in Figure 3.5.10, gives comparatively better result in the area of 2, because it is successfully got rid of the noise, but it is not successful in the area of 1, because the critical feature is lost. The result of filtering, red line, gives good result in the area of 1 while giving bad result in the area of 2 in Figure 3.5.10. The results of smoothed Displacement data can also be seen in Figure 3.5.8 by using the same parameters.

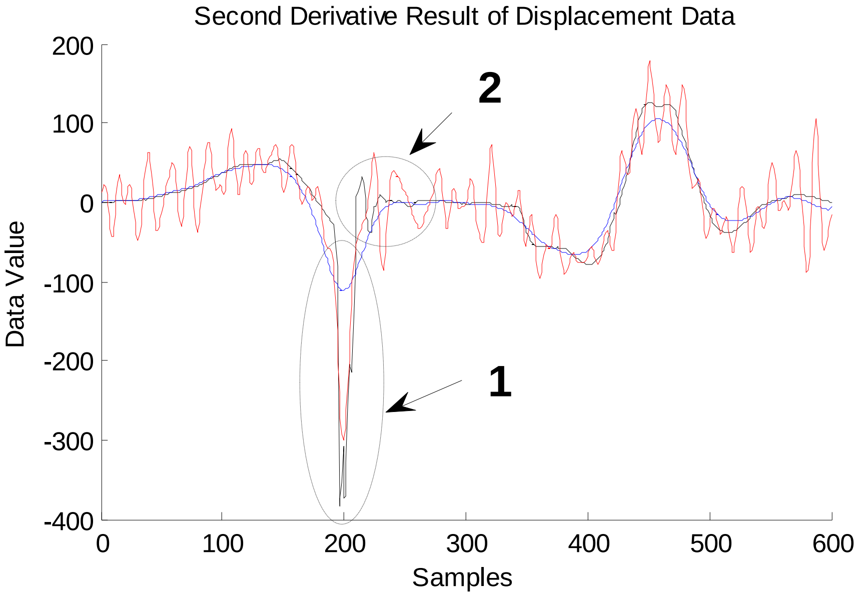


Figure 3.5.10: (Ambrosio Tortorelli Approximation of the Mumford Shah). The Comparative Second Derivative Results of Smoothed Displacement Data and Original Acceleration Data.

Black Line: Original Acceleration Data

Blue Line : Second Derivative of Smoothed Displacement Data by using T=100, Δt=0.4, α=64, β=0.025, ρ=0.000001,

Red Line : Second Derivative of Smoothed Displacement Data by using T=100, Δt=0.4, α=8, β=256, ρ=0.000001.

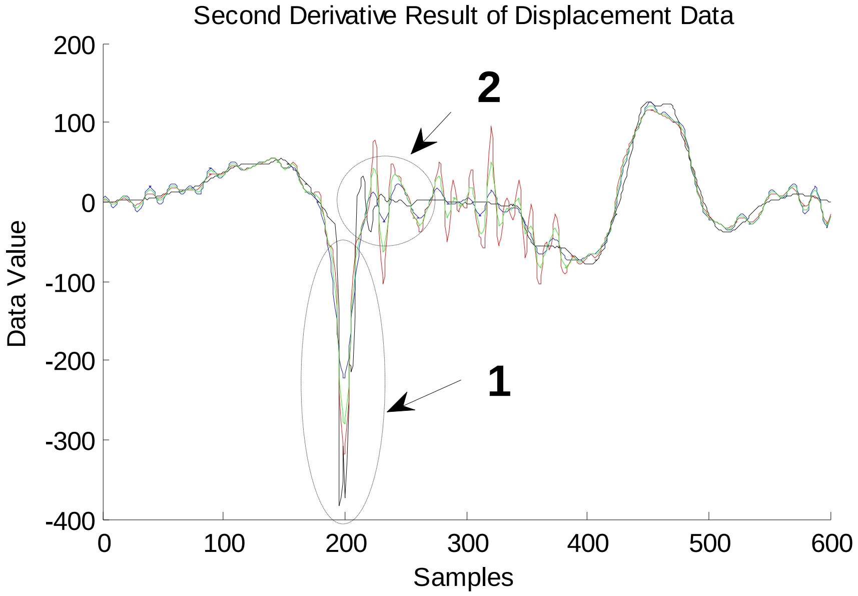


Figure 3.5.11: (Ambrosio Tortorelli Approximation of the Mumford Shah). The Comparative Second Derivative Results of Smoothed Displacement Data and Original Acceleration Data.

Black Line: Original Acceleration Data

Blue Line : Second Derivative of Smoothed Displacement Data by using T=100, Δt=0.4, α=8, β=90, ρ=0.000001,

Green Line : Second Derivative of Smoothed Displacement Data by using T=100, Δt=0.4, α=8, β=70, ρ=0.015,

Red Line : Second Derivative of Smoothed Displacement Data by using T=100, Δt=0.4, α=8, β=50, ρ=0.033.

3.6 GENERAL REVIEW OF THE NON-LINEAR SMOOTHING FILTERS

Smoothing is a process that removes high frequency components. Non-linear smoothing filters are the filters that smoothes the data without removing significant parts of the data content.

Anisotropic diffusion can be used to remove noise from the data without removing significant parts of the data. With a constant diffusion coefficient, the anisotropic diffusion equations reduce Gaussian blurring. If the value of the diffusion coefficient is determined by using magnitude of the first derivative, like Perona Malik, the resulting equations discourage diffusion at the significant part of the data and encourage diffusion in the other part of the data. Hence the significant part of the data can be kept while removing the high frequency components from the data.

Perona-Malik and Ambrosia-Tortorelli are anisotropic diffusion filters. If the contrast parameters are adjusted correctly, Perona-Malik and Ambrosia-Tortorelli give the same result. Time to stop for Perona-Malik is determined by using the equation in (3.23) at Ambrosio-Tortorelli approximation. So,

$$\sigma = \sqrt{\frac{2 \cdot \alpha}{\beta}} \tag{3.24}$$

The value of λ, a contrast parameter of Perona Malik equation, can be found experimentally, by using the equation (3.20), as seen in Figure 3.6.1.

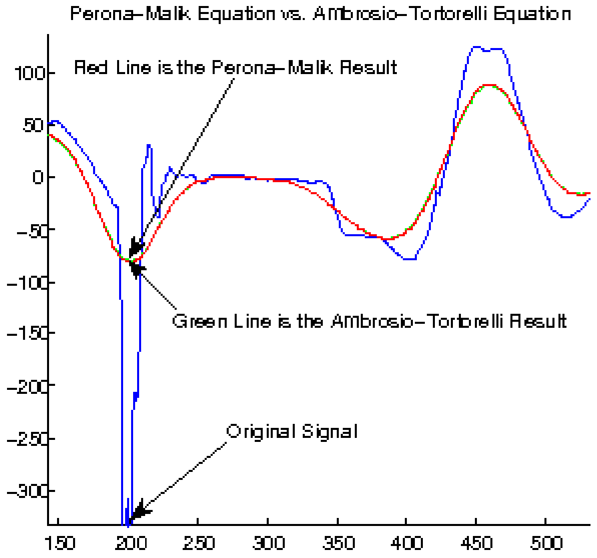


Figure 3.6.1: (Perona-Malik Equation vs. Ambrosio-Tortorelli Equation)
Green Line : Ambrosio-Tortorelli Result for T=200, Δt=0.4, α =8, β=0.05, ρ= 0.000000001
Red Line : Perona-Malik Result for T=200, Δt=0.4, λ=10, and σ=18.

When the Ambrosio-Tortorelli is applied to a signal, for each iteration, mean value of the signal doesn't change. Variance, Gradient, Total Gradient $\Sigma|\Delta|^2$, standard deviation and Entropy monotonically decrease.

The examination of these values on the Acceleration Data, using Ambrosio-Tortorelli in Figure 3.6.2, is in Figure 3.6.3.

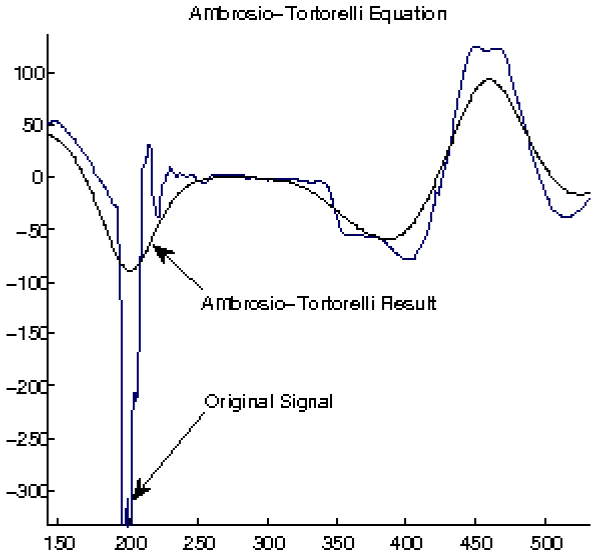
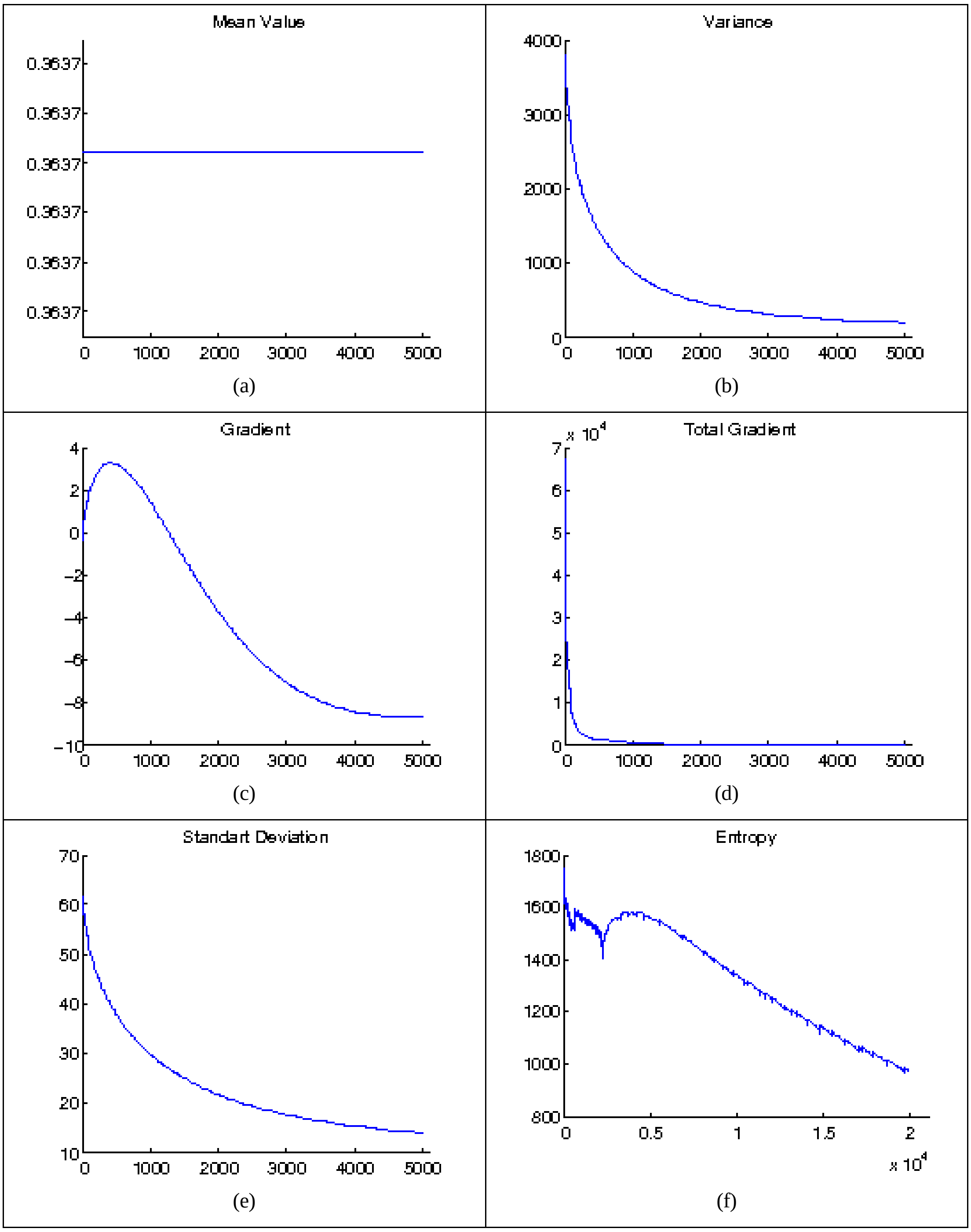


Figure 3.6.2: (Result for the Ambrosio Tortorelli Approximation of the Mumford Shah)
The result for Result: T=200, Δt=0.4, α =8, β=0.05, ρ= 0.000001



(a)

(b)

(c)

(d)

(e)

(f)

Figure 3.6.3: (Results for the Ambrosio Tortorelli Approximation of the Mumford Shah)
(a) Mean Value vs. Iteration
(b) Variance vs. Iteration
(c) Gradient vs. Iteration
(d) Total Gradient vs. Iteration
(e) Standard Deviation vs. Iteration
(f) Entropy vs. Iteration

Which Non-linear smoothing algorithm gives better result than the others can be seen in Figure 3.6.4, when the noise level of smoothed Displacement data is hold similarly for the first 150 samples and concentrated on the impulse level.

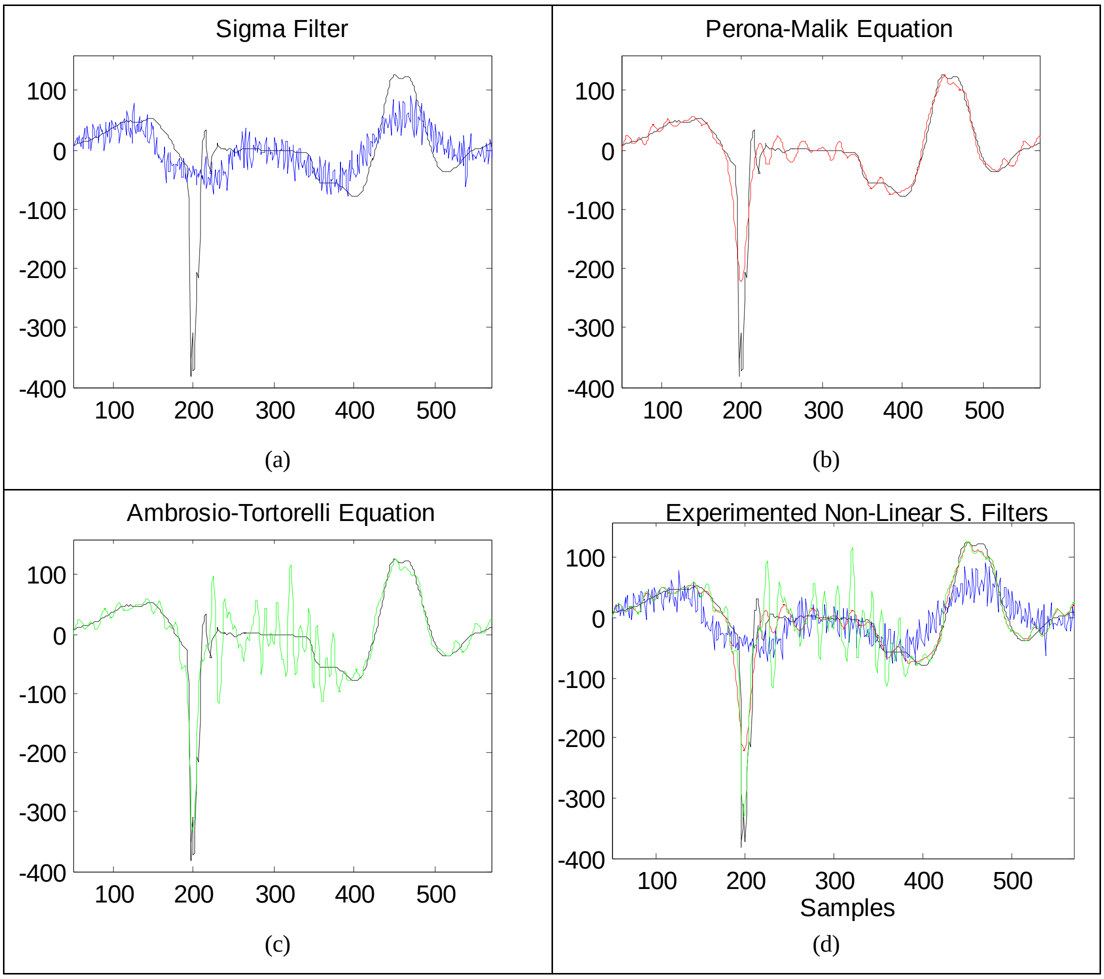


Figure 3.6.4: (The Comparative Second Derivative Results of Displacement Data by using some of the experimented Non-Linear Smoothing Filters).
Black Line: Original Acceleration Data,
(a) Sigma Filter Result by using σ=0.5, Filter size=91,
(b) Perona Malik Filtering Result by using λ=10, T=18, Δt=0.25 and σ=3,
(c) Ambrosio Tortorelli Approximation by using T=100, Δt=0.4, α=8, β=50, ρ=0.033,
(d) Comparison of all results from Figure 3.6.4.a to Figure 3.6.4.c

3.7 BAKLAVA FILTER

Baklava filtering is a Non-linear smoothing filtering. One dimensional Baklava filter works on an array which is divided into two sub array parts. In this method, filter size should be defined as an odd number. In the first array part, members are taken from the middle of the array. The second array part is consisting of all array members out of the first part. The size of the array parts are determined by dividing two equal or almost equal parts of processed array. In each sub array part, the mean and variance are computed. The output value (located at the center of the array) is set to the mean value of the sub array where the variance is the smaller than the other part. The mask size is defined as 7 for the following implementation:

$$\left[\;\left|\left(X_i \quad X_{i+1}\right)\right\rangle \left\{Y_{i+2} \quad Y_{i+3} \quad Y_{i+4}\right\} \left\langle\left(X_{i+5} \quad X_{i+6}\right)\right|\;\right] \tag{3.25}$$

So, it divides into 2 sub array parts.

First array part is:

$$\left[\; X_i \quad X_{i+1} \;\; X_{i+5} \quad X_{i+6} \;\right] \tag{3.26}$$

And second array part is:

$$\left[\; Y_{i+2} \quad Y_{i+3} \quad Y_{i+4} \;\right] \tag{3.27}$$

Last step is computing the following values:

$mean_X$: Mean the value of the first array part.

$mean_Y$: Mean the value of the second array part

var_X : Variance the value of the first array part.

var_Y : Variance the value of second array part.

var_X and var_Y are compared and the mean value of the smaller variance is copied to the address which is defined by Y_{i+3}. This Y_{i+3} value is located at the center of the array sequence.

This algorithm gives better result than Kuwahara filter. Some comparisons are done for the test data. In this algorithm filter size shouldn’t be increased so much because of the stability of the algorithm and the result.

3.7.1 Golf Ball Drop Data Results for Baklava Filtering

After applying the Baklava filter on the Golf ball drop data, in Figure 1.1.1, the results are in Figure 3.7.1.

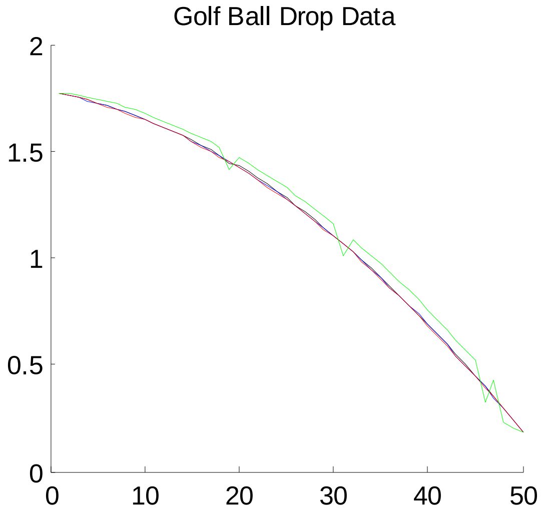


Figure 3.7.1: (Baklava Filtering vs. Kuwahara Filtering)

Black Line: Original Golf Ball Drop Data,

Blue Line: The result of the Baklava Filtering for Filter Size = 7,

Red Line: The result of the Baklava Filtering for Filter Size = 9,

Green Line: The result of the Kuwahara Filtering for Filter Size = 7,

As it is seen from Figure 3.7.1, while Kuwahara filter results are corrupted in some part of the data, Baklava filter gives better result and there is no corruption in any part of the data. This condition can be seen clearly in Figure 3.7.2.

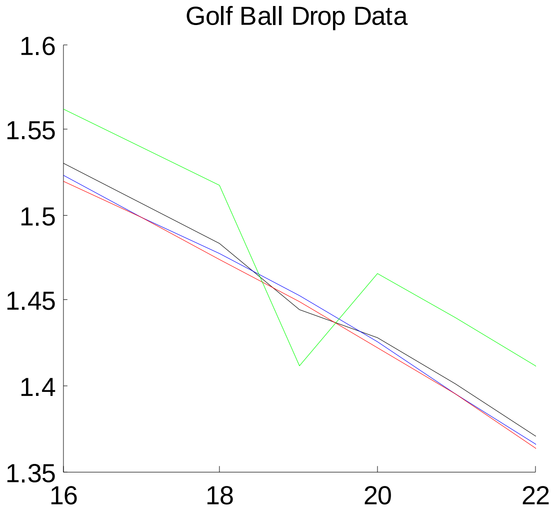


Figure 3.7.2: (The samples, between 16 and 22, of Figure 3.7.1)

Black Line: Original Golf Ball Drop Data,

Blue Line: The result of the Baklava Filtering for Filter Size = 7,

Red Line: The result of the Baklava Filtering for Filter Size = 9,

Green Line: The result of the Kuwahara Filtering for Filter Size = 7,

3.7.2 Displacement Data Results for Baklava Filtering

After applying the Baklava filter on the Displacement data, in Figure 1.1.2.a, the results are in Figure 3.7.3, Figure 3.7.4 and in Figure 3.7.5.

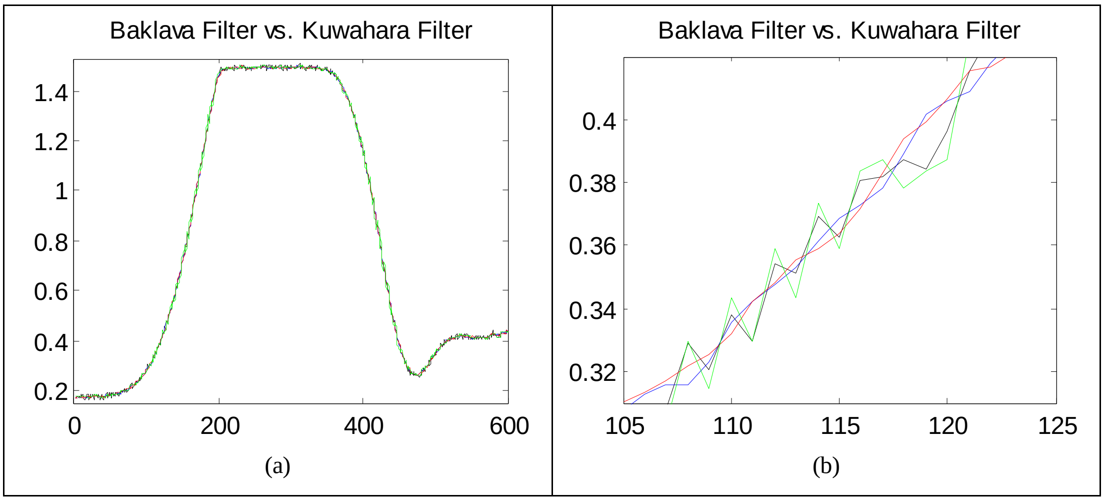


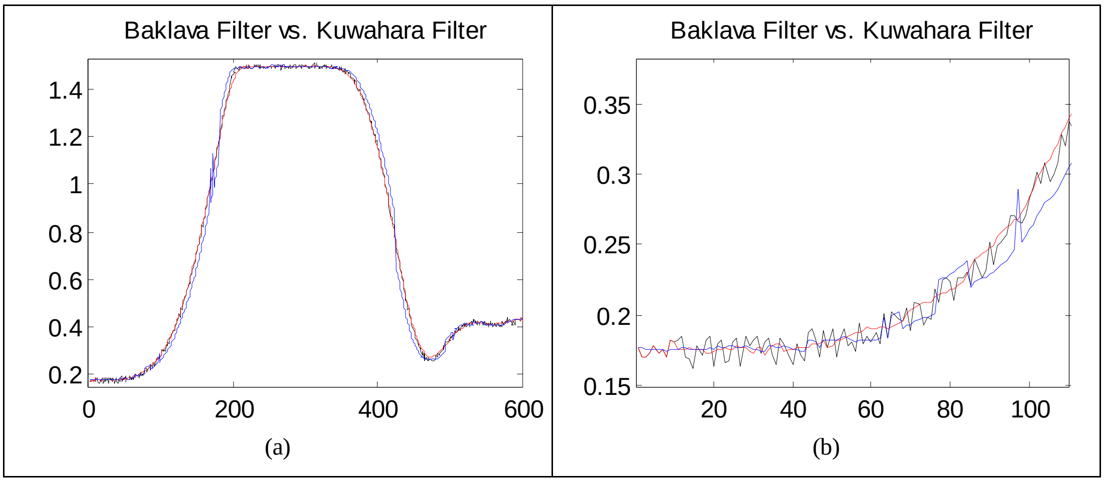
Figure 3.7.3: (Baklava Filtering vs. Kuwahara Filtering) (a) Smoothed Displacement Data Result, **(b)** The samples, between 105 and 125, of Figure 3.7.3.a.

Black Line : The Original Displacement Data,

Blue Line: The result of the Baklava Filtering for Filter Size = 7,

Red Line: The result of the Baklava Filtering for Filter Size = 9,

Green Line: The result of the Kuwahara Filtering for Filter Size = 7,



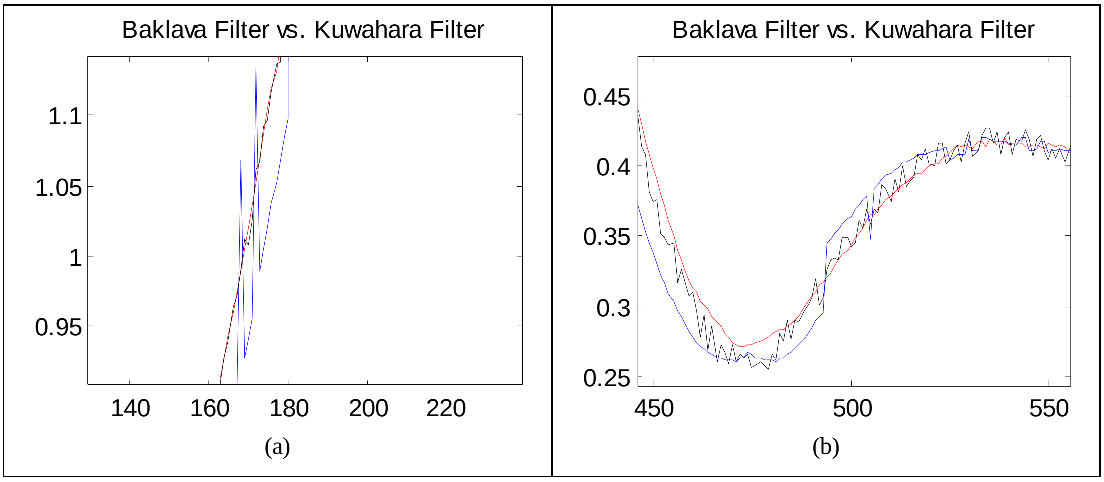
(a) (b)

Figure 3.7.4: (Baklava Filtering vs. Kuwahara Filtering) (a) Smoothed Displacement Data Result, **(b)** The samples, between 0 and 110, of Figure 3.7.4.a.

Black Line : The Original Displacement Data,

Red Line: The result of the Baklava Filtering for Filter Size = 21,

Blue Line: The result of the Kuwahara Filtering for Filter Size = 21,



(a) (b)

Figure 3.7.5: (Baklava Filtering vs. Kuwahara Filtering) (a) The samples, between 130 and 230, of Figure 3.7.4.a., **(b)** The samples, between 445 and 555 of Figure 3.7.4.a.

Black Line : The Original Displacement Data,

Red Line: The result of the Baklava Filtering for Filter Size = 21,

Blue Line: The result of the Kuwahara Filtering for Filter Size = 21,

3.7.3 Acceleration Data Results for Baklava Filtering

After applying the Baklava filter to the Acceleration data, in Figure 1.1.2.b, following results are obtained. When mask size is increased, the difference between maxima and minima does not decrease a lot, however, a corruption on the signal itself occurs, in Figure 3.7.6 and in Figure 3.7.7.

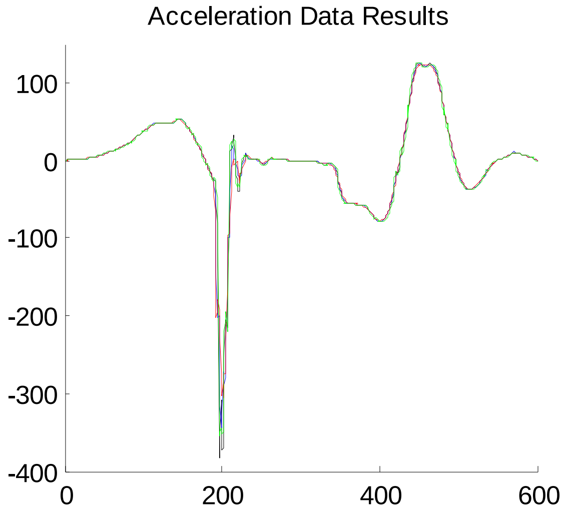


Figure 3.7.6: (Baklava Filtering vs. Kuwahara Filtering)
Black Line: Original Acceleration Data,
Blue Line: The result of the Baklava Filtering for Filter Size = 7,
Red Line: The result of the Baklava Filtering for Filter Size = 9,
Green Line: The result of the Kuwahara Filtering for Filter Size = 7.

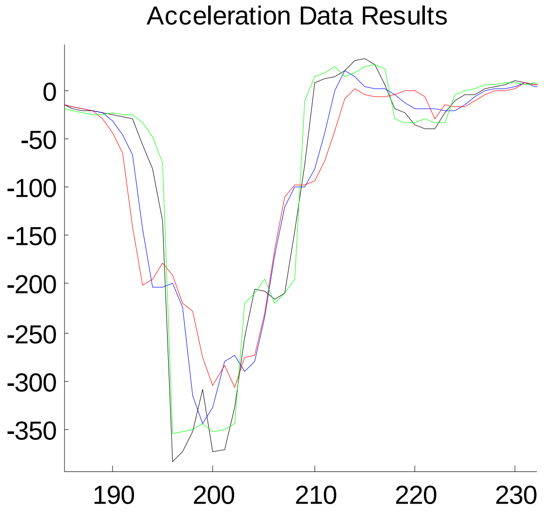
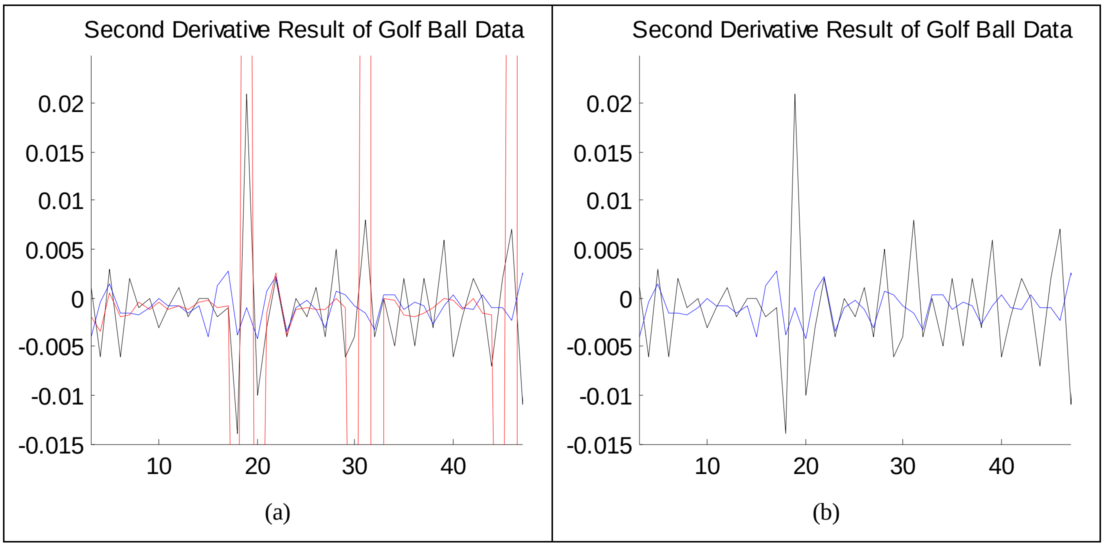


Figure 3.7.7: (The samples, between 190 and 230, of Figure 3.7.6)
Black Line: Original Acceleration Data,
Blue Line: The result of the Baklava Filtering for Filter Size = 7,
Red Line: The result of the Baklava Filtering for Filter Size = 9,
Green Line: The result of the Kuwahara Filtering for Filter Size = 7.

3.7.4 Second Derivative Results for Baklava Filtering:

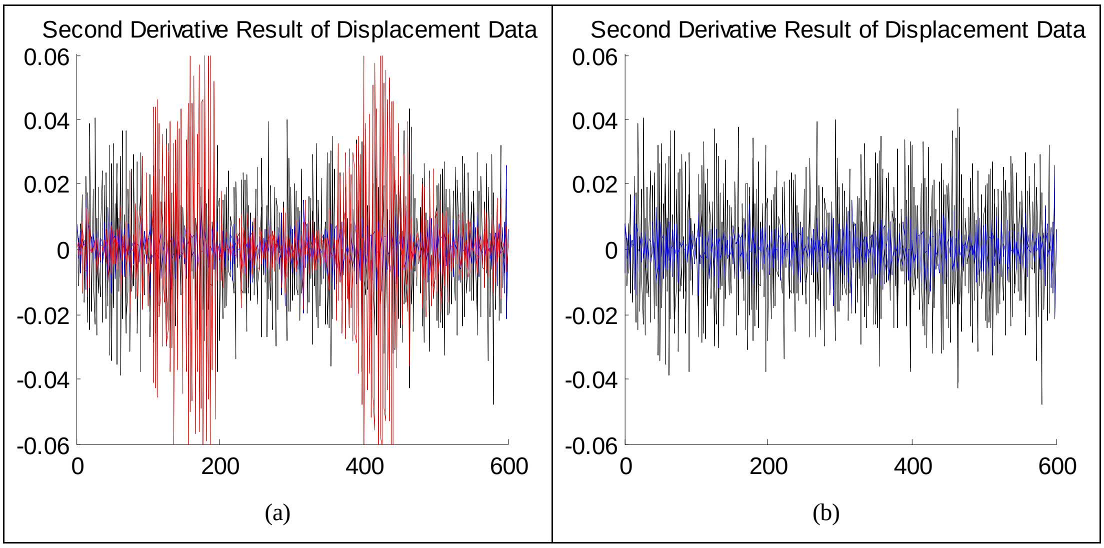
Second Derivative Result of Smoothed Displacement data by using the Baklava filtering can be seen in Figure 3.7.8 and in Figure 3.7.9.



(a) (b)

Figure 3.7.8: (Baklava Filtering vs. Kuwahara Filtering). (a) Second Derivative Result of Smoothed **Golf Ball Data** using Baklava Filtering & Kuwahara Filtering for the <u>Mask Size=7</u> **(b)** Same as Figure 3.7.8.a <u>except Kuwahara Filtering Result</u>.

<u>Black Line</u>: Second Derivative of the Original Golf Ball Drop Data,
<u>Blue Line</u>: Second Derivative of the Data by using the <u>Baklava Filtering</u> for mask Size=7,
<u>Red Line</u>: Second Derivative of the Data by using the <u>Kuwahara Filtering</u> for mask Size=7.



(a) (b)

Figure 3.7.9: (Baklava Filtering vs. Kuwahara Filtering). (a) Second Derivative Result of Smoothed **Displacement Data** using Baklava Filtering & Kuwahara Filtering for the <u>Mask Size=7</u> **(b)** Same as Figure 3.7.9.a <u>except Kuwahara Filtering Result</u>.

<u>Black Line</u>: Second Derivative of the Original Displacement Data,
<u>Blue Line</u>: Second Derivative of the Data by using the <u>Baklava Filtering</u> for mask Size=7,
<u>Red Line</u>: Second Derivative of the Data by using the <u>Kuwahara Filtering</u> for mask Size=7.

CHAPTER 4

SUMMARY AND CONCLUSION

In this thesis, the effect of different smoothing strategies on the estimated second derivative is experimentally investigated. Widely used filters developed for handling piecewise smooth data, where the jumps at the first derivative are of interest, are examined in a context where the interest is to obtain the second derivative accurately. In particular second derivative of a positional Displacement data have a physical importance because it corresponds to acceleration. Obtaining accurate estimate of the second derivative is much more challenging than obtain accurate estimate of the original data (which is the zeroth derivative) or of the first derivative. This is because differentiation amplifies high frequency content more than low frequency content, thus decreases signal to noise ratio. Naturally, differentiating twice decreases signal to noise ratio more than differentiating once. So that, second derivative of the filtered signals don't give good result while getting better result in the first derivative which used for preserving the edge. A new method is also proposed in Section 3.7, named Baklava filtering. Codes, used to demonstrate the results of the smoothing methods, are coded by using Matlab by the author and images are saved by using Enhanced Meta File (EMF) and BMP format.

Linear smoothing filters that are covered in Chapter 2 are:

2.1 Mean Filter

2.2 Iterated Mean Filter

2.3 Gaussian Filter

2.4 Ideal Low Pass Filter

2.5 Butterworth Low Pass Filter

2.6 Linear Diffusion Filter

Linear smoothing filters don’t take the characteristic properties of the given data into account. Depending on the choice of weight function and choice of the mathematical model, linear filters take various forms. These filters are not successful at preserving the critical features in the data.

When the noise level of the smoothed Displacement data is hold similarly for the first 150 samples and concentrated on the impulse level in Figure 2.7.2, Butterworth Low Pass filter seems to give better result between the samples 180 and 220, but it does not give good result between the samples 230 and 350. When the filter gives smooth result at the second area, between the samples 230 and 350, then the critical feature in the first area, between the samples 180 and 220, is lost. All of the experimented Linear smoothing algorithms have the same problem, because noise and signal spectrums are overlapping each other. Locally at some part of the data, signal has got high spectrum value. So especially in this area, it is hard to separate the signal from the noise. Finding the right cut-off value which separates the signal from the noise is not possible for this condition.

Non-linear smoothing filters that are covered in Chapter 3 are:

3.1 Kuwahara Filter

3.2 Sigma Filter

3.3 Median Filter

3.4 Perona Malik Filter

3.5 The Ambrosio Tortorelli Approximation of the Mumford Shah Model

Non-linear filters are important at smoothing the data. Linear smoothing filters can not determine the data, whether it is noise or an important part of the data. The condition, which is expected, is smoothing the data while keeping important part of the data. Non-linear smoothing filters provide this condition.

When the noise level of the smoothed Displacement data is hold similarly for the first 150 samples and concentrated on the impulse level in Figure 3.6.4, Ambrosio Tortorelli Approximation gives better result between the samples 180 and 220, but it does not give good result between the samples 230 and 350.

Actually, when the right parameters are chosen for Ambrosio Tortorelli and Perona Malik they all give the same result. If the contrast parameters are adjusted correctly, they give the same result. Time to stop for Perona-Malik is determined by using the equation in (3.23) at Ambrosio-Tortorelli approximation. So, (3.24) shows the relation between Perona Malik and Ambrosio Tortorelli Approximation.

Widely used successful filters in image processing aim at preserving discontinuities in the data and estimate the first order derivative accurately. They fail when the aim is to estimate second order derivative (acceleration) accurately.

Complexity of the problem comes from the non-stationary condition of the signal. Popular filtering algorithms are successful especially when the data to be processed are stationary (Woltring, 1990, 1995), but this data set is a non-stationary and some part of the signal contains high frequency components as a critical feature. So it is hard to remove the noise from the original data by using conventional smoothing filters. So, new methods are necessary for the second derivative problems.

REFERENCES

[1] Luenberger, David G., Introduction to Dynamic Systems: Theory, Models and Applications. *Wiley Text Books, New York, NY, USA*. Page 382, 1979.

[2] C.L. Vaughan, Smoothing and Differentiation of Displacement-time Data: an Application of Splines and Digital Filtering. *International Journal of Bio-Medical Computing* 13:375-382, 1982.

[3] An experiment that was reported in BIOMECHANICS X-B (ed:Bengt Jonsson) *Human Kinetics Publishers*, Champaign, IL. Pp.1163-1167.

[4] Statistics Department, Texas A&M University, College Station TX 77843-3143 USA *http://www.stat.tamu.edu/~jnewton/stat626/topics/topics/topic9.pdf, (25 November 2009).*

[5] N. Jackson St. Arlington 612, VA 22201 USA, *http://www.statistics.com/ resources/glossary/l/linfilt.php, (16 December 2009).*

[6] R. Fisher, S. Perkins, A. Walker and E. Wolfart, *http://homepages.inf.ed.ac.uk/ rbf/HIPR2/fourier.htm*, (11 *January* 2010).

[7] Anil K. Jain, *Fundamentals of Digital Image Processing*. Prentice Hall, 1989.

[8] Kenneth R. Castleman, *Digital Image Processing*. Prentice Hall, 1996.

[9] Richard Woods, Rafael Gonzalez. *Digital Image Processing*. Tom Robbins, 2002.

[10] G.L. Flynn. Mass Transport Phenomena and Models: Theoretical concepts. *In J. Pharm. Sci.*, 63 (4), pages 479-510, 1974.

[11] Weickert, J., 1998. *Anisotropic Diffusion in Image Processing,* Teubner, 1998.

[12] J. Malik, P. Perona. , Scale-space and Edge Detection using Anisoropic Diffusion. *IEEE Trans. Pattern Anal. And Machine Intell.*, 12(7):629–639, 1990.

[13] Allen Tannenbaum, Mostafa Kaveh, Yu-Li You, Wenyuan Xu. , Behavioral Analysis of Anisotropic Diffusion in Image Processing. *IEEE Trans. Image Processing,* 5(11): 1539–1553, November 1996.

[14] V. Tortorelli, L. Ambrosio. On the Approximation of Functionals Depending on Jumps by Elliptic Functionals via Γ-Convergence. *Commun. Pure Appl. Math.*, 43(8):999–1036, 1990.

[15] C. Kamath, S.K.Weeratunga. A Comparison of Pde-based Non-linear Anisotropic Diffusion Techniques for Image Denoising. *Image Processing: Algorithms and Systems II, SPIE Electronic*, pages 279–290, Jan 2002.

[16] S. Eiho, M. Kuwahara, K.Hachimura and M. Kinoshita. Digital Processing of Biomedical Images. *Plenum Press, New York NY*, pages 187–203, 1976.

[17] J. Lee. Digital Image Smoothing and the Sigma Filter. *Computer Vision, Graphics and Image Processing*, Vol. 24, issue 2, pages 255–269, 1983.

[18] A. S. Yılmaz, “Edge Preserving Smoothing with Directional Consistency”, Ms Thesis, *Middle East Technical University* , 2007.

[19] Chemistry and Biochemistry Department, Maryland University, College Park, 20742 USA *http://terpconnect.umd.edu/~toh/SPECTRUM/Differentiation.html, (2 April 2010).*