PERFORMANCE EVALUATION OF FRACTIONAL-ORDER DIGITAL FILTER FOR SIGNAL AND IMAGE PROCESSING APPLICATIONS

Thesis Submitted

in fulfillment of the requirements for the award of the degree

of

DOCTOR OF PHILOSOPHY

by

ANMOL GUPTA

REG. NO. 901706005

UNDER THE SUPERVISION OF

DR. SANJAY KUMAR

ASSOCIATE PROFESSOR

ECED

Department of Electronics and Communication Engineering

THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY

PATIALA-147004

JULY 2022

DECLARATION

I hereby declare that the work which is being presented in this thesis entitled **"Performance Evaluation of Fractional-Order Digital Filter for Signal and Image Processing Applications"**, in fulfillment of the requirements for the award of the Degree of **Doctor of Philosophy** submitted in the *Department of Electronics and Communication Engineering of the Thapar Institute of Engineering and Technology, Patiala* is an authentic record of my own research work carried out during a period from July, 2017 to July, 2022 under the supervision of Dr. Sanjay Kumar, Associate Professor, Department of Electronics and Communication Engineering, Thapar Institute of Engineering and Technology, Patiala. The matter presented in the thesis has not been submitted in part or full to any other University or Institute for the award of any degree in India or abroad.

Dated: 29-07-22

Gupta 29/07/22

Anmol Gupta

(901706005)

It is certified that the above statement made by the candidate is correct to the best of my knowledge and belief.

Dated: 29/July/2022.

29/July/2022.
Kumar
(Dr. Sanjay Kumar).
ECED.

Dr. Sanjay Kumar

(Supervisor)

CERTIFICATE

It is hereby certified that the thesis entitled **"Performance Evaluation of Fractional-Order Digital Filter for Signal and Image Processing Applications"**, submitted by Ms. Anmol Gupta, a research scholar in the *Department of Electronics and Communication Engineering, Thapar Institute of Engineering and Technology, Patiala*, for the award of the degree of **Doctor of Philosophy**, is a record of bonafide research work carried out by her under my supervision and guidance. She has fulfilled the requirements for the submission of this thesis which has reached the requisite standard. The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

Dated: 28/July/2022.

(Dr. Sanjay Kumar)
ECED.

28/July/2022.

Dr. Sanjay Kumar
Associate Professor, ECED
Thapar Institute of Engineering and Technology
Patiala, Punjab - 147004
India.

ACKNOWLEDGEMENTS

It is my proud privilege to acknowledge and extend my gratitude to several persons who helped me directly or indirectly in completion of this thesis. Initially, I would like to bow in front of the Almighty God who made me capable in every aspect of life. I express my heartfelt gratitude to Him for always showering His blessings on me whenever needed.

I would like to express my heart full indebtedness and owe a deep sense of gratitude to my teacher and my guide Dr. Sanjay Kumar for his sincere guidance and support with encouragement to go ahead. I am thankful to him for instilling in me some of the best qualities that one needs in the course of his life, for example, how to be disciplined in one's life, how to present yourself in a room of well-experienced people, how to write research papers effectively and make presentations that are eye-catching, and last but not the least, how to grow personally as well as professionally with grace. I am also grateful to him for enriching my spiritual beliefs that helped me get through difficult situations in life.

I am also thankful to Dr. Alpana Agarwal, Professor and Head, ECED, for providing me with the adequate infrastructure for carrying out the research work effectively. I would like to express my sincere thankfulness to my doctoral committee members Dr. Vinay Kumar, Dr. Ashutosh Kumar Singh, Dr. Harish Garg for always providing me with valuable feedback during the course of my PhD. Also, I am thankful to ECED faculty members Dr. Navneet Sharma and Dr. Dinesh Bhardwaj for helping me with my doubts in research.

The entire course of this study would not have been completed without the love, support and encouragement of my family. I will always be deeply indebted to my parents who supported me in every aspect of my life and prayed for my wellness every now and then. I am immensely proud to have younger siblings Ruchika Gupta and Parul Gupta who helped

me get through difficult times by listening to me and motivating me whenever needed. Their presence and unrelenting support made me deal with every situation in an effortless manner.

I would like to express my heartfelt gratitude for my friend and soulmate Dr. Ishan Chawla, who motivated me to go for higher studies and helped me deal every situation with patience. He also helped me in widening the perspective of my research and motivated me to keep my spirits high even in the hardest of the times. During times when I was in doubt, he believed in me more than I did and helped me in achieving various milestones in my research. He has always been my biggest cheerleader in every part of my success. This course of study would not have been possible without his support, guidance, and motivation.

I would also like to thank my best friend Gursimar Kaur for her unconditional care and support in every aspect of my life and career. I am grateful to my friends Ms. Aditi Bajaj and Ms. Amandeep Kaur for their relentless support technically as well as emotionally who were there with me in my hardest of times and also, enjoyed every milestone together with me. I am grateful for the times I spent laughing and crying with my friends Ms. Aditi Bajaj, Ms. Amandeep Kaur, Ms. Deepika Singh, Dr. Abhinav, Mr. Ankush during the course of my study. I owe a deep sense of gratitude towards Ms. Anjana Bhatia and Dr. R.K. Bahl for always motivating and providing us with life lessons.

Lastly, I would like to thank my parents and my parents-in-laws who stood by my side through every thick and thin of life. I owe my sincere gratitude to them for helping me believe in my own self. I would again like to thank my supervisor and guide Dr. Sanjay Kumar, who not only guided me throughout my career but also helped me gained perspective for my bright future. I would also like to express my gratitude towards Deepti Ma'am for helping me as and when required. She made our time cheerful by interacting with us every now and then. I pay my sincere gratitude towards everyone who helped me in one way or another.

This study has indeed helped me to grow personally as well as professionally where I have explored various avenues related to my topic and I am sure it will help me in my future endeavors.

Anmol Gupta

ABSTRACT

Fractional-order calculus (FOC), which is often contemplated as a branch of pure mathematics, has encapsulated the attention of eminent researchers from several backgrounds due to its profound applications in various fields of science and engineering. The mathematical tool of FOC, which is, in fact, the generalization of integer-order calculus, is concerned with differentiation and integration of non-integer orders. FOC concept emerged from the generalization of integer-order calculus in two most prominent ways: One based upon the direct generalization of the limit of difference quotients defining integer-order derivatives and another one based upon the inversion of the generalization of n-fold integer-order successive integration rule (a.k.a. Cauchy's formula for repeated integration). These generalizations provided two different ways to define fractional-order derivatives and hence, referred to as Grünwald-Letnikov (GL) and Riemann-Liouville (RL) fractional-order derivatives, respectively.

From signal and image processing viewpoint, the definitions of fractional-order derivatives can be viewed in terms of convolution, where the filter or kernel function is of power-law type. A natural query that arises is: Why the kernel is of power-law type? This is because of the association of power-law in Cauchy's integral formula. Fractional-order operators exhibiting power-law kernel have found tremendous applications in natural and man-made phenomena. However, it has been pointed out recently that there exists an artificial singularity at the initial point of the power-law kernel, due to which many real-life phenomena cannot be implemented exactly. Therefore, several mathematicians made an attempt to replace the power-law kernel with other appropriate kernels (such as, exponential kernel, Mittag-Leffler kernel) and hence, led to the development of various fractional-order operators based upon different kernel functions.

On the other hand, a significant attention has been paid by the research community on the study of fractional-order digital differentiators (FODDs), i.e., digital discrete-time systems that can perform non-integer order differentiation. The fundamental advantage of FODDs is that they provide an extra degree of freedom which helps in optimizing the performance of methods based upon traditional integer-order calculus. Therefore, research work carried out in this thesis emphasis on the design of power-law kernel and Mittag-Leffler kernel based fractional-order digital filters for signal and image processing applications.

Out of power-law kernel based GL and RL fractional-order derivatives, GL fractional-order derivative has been intensively-studied and extensively-utilized for the design of FODD due to its discrete nature. However, there seems to be an improvement window for designing filters based upon RL fractional-order differential operator. Hence, initial research work in this thesis aims at establishing a closed-form analytical formulation for the design of Riemann-Liouville based fractional-order digital differentiator and further authenticates the efficacy of the proposed design through simulation results as compared to GL-based methods.

Furthermore, to explore the applicability of recently introduced fractional-order operators, non-local and non-singular Mittag-Leffler kernel based Atangana-Baleanu-Caputo fractional-order digital filter (herein, referred to as ABC-FODF) is proposed. Atangana-Baleanu fractional-order differential operator has been extensively investigated in various fields of science and engineering. However, applications of AB fractional-order operator in signal and image processing applications are limited, and hence, the research work carried out in this thesis aims to add to its applications by establishing the design of ABC-FODF. Furthermore, the applications of the proposed ABC-FODF are discussed for R-peak detection in electrocardiogram (ECG) signals and digital image sharpening.

After establishing closed-form analytical formulations for the design of fractional-order digital filters, further research work aims at studying various fractional-order integral operators for image denoising applications. For this, a generalized fractional integral (GFI) mask which incorporates both RL and AB fractional-order integral operators is proposed. The proposed GFI mask provide a more flexible and appropriate tool for image denoising applications and hence, avoids the construction of different masks for each definition separately. Furthermore, a GFI-based adaptive mask (GFI-AM) is put forward to select the fractional order based upon local features of an image. The performance of various fractional-order

operators i.e., GL, RL, and AB is tested for images affected by Gaussian and salt and pepper noise. From the simulation studies conducted, it is seen that the proposed GFI-AM incorporating AB fractional integral operator outperforms the conventional operators in removal of noise from images.

Furthermore, as a major contribution of this research work, a generalized fractional integral and fractional differential based adaptive algorithm (GFIFD-AA) is proposed for the detection and removal of salt and pepper noise in images. The proposed algorithm is segregated into two parts: In the first part, a novel noise detection method for the detection of salt and pepper noise (SP) is introduced. In the second part, the detected noisy pixels are processed by utilizing the proposed GFI-AM. In order to preserve the texture and edge details of the images, a generalized fractional differential based adaptive mask (GFD-AM) is proposed, and hence, the non-noisy pixels are processed by utilizing the proposed GFD-AM. Therefore, the proposed GFI-AM and GFD-AM are utilized together in the framework of proposed GFIFD-AA for the effective removal of salt and pepper noise and preservation of details in images. The proposed method exhibits a profound advantage in avoiding misclassification of original pixels with intensity values 0 or 255 as noisy pixels and hence, is effective in detecting SP noise in black and white images as well as synthetic images (where every pixel of an image is of value 0 or 255). The simulation studies conducted on several standard as well as synthetic images indicated the superiority of the proposed method against well-established state-of-the-art methods.

Hence, the study carried out in this thesis has unfolded significant advantages of fractional-order calculus for signal and image processing applications and hence, opens up a new paradigm of research based on recently introduced fractional-order operators for various signal filtering, image enhancement and adaptive image denoising applications.

LIST OF PUBLICATIONS

International Journals

1. **A. Gupta** and S. Kumar, "Closed-form analytical formulation for Riemann–Liouville-based fractional-order digital differentiator using fractional sample delay interpolation," *Circuits, Systems, and Signal Processing*, vol. 40, no. 5, pp. 2535-2563, 2020. DOI:10.1007/s00034-020-01589-2. **(SPRINGER–SCI Impact Factor: 2.311)**

2. **A. Gupta** and S. Kumar, "Design of Atangana-Baleanu-Caputo fractional-order digital filter," *ISA Transactions*, vol. 112, pp. 74-88, 2021. DOI:10.1016/j.isatra.2020.11.021. **(ELSEVIER–SCI Impact Factor: 5.911)**

3. **A. Gupta** and S. Kumar, "Generalized framework for the design of adaptive fractional-order masks for image denoising," *Digital Signal Processing*, vol. 121, p. 103305, 2022. DOI: 10.1016/j.dsp.2021.103305. **(ELSEVIER–SCI Impact Factor: 2.920)**

4. **A. Gupta** and S. Kumar, "Design of Mittag–Leffler kernel-based fractional-order digital filter using fractional delay interpolation," *Circuits, Systems, and Signal Processing*, vol. 41, pp. 3415-3445, 2022. DOI: 10.1007/s00034-021-01942-z. **(SPRINGER–SCI Impact Factor: 2.311)**

Conferences

1. **A. Gupta** and S. Kumar, "Generalized framework for higher-order fractional derivatives–From Riemann–Liouville to Atangana–Baleanu", *2019 5th International Conference on Signal Processing, Computing and Control (ISPCC)*, IEEE, Jaypee University of Information Technology, Waknaghat, Solan, India, 10-12 October 2019, pp. 114-118. DOI:10.1109/ISPCC48220.2019.8988303.

LIST OF ABBREVIATIONS

1D	:	One-dimensional
2D	:	Two-dimensional
AB	:	Atangana-Baleanu
AM	:	Adaptive Mask
ABC	:	Atangana-Baleanu in Caputo sense
ABR	:	Atangana-Baleanu in Riemann-Liouville sense
BW	:	Baseline Wander
CF	:	Caputo-Fabrizio
DCT	:	Discrete Cosine Transform
DFT	:	Discrete Fourier Transform
DHT	:	Discrete Hartely Transform
DST	:	Discrete Sine Transform
ER	:	Error Rate
ECG	:	Electrocardiogram
EEG	:	Electroencephalography
EME	:	Enhancement Measure
fBm	:	Fractional Brownian motion
FDI	:	Fractional Delay Interpolation
FDM	:	Finite Difference Method
FIR	:	Finite Impulse Response
FOC	:	Fractional-Order Calculus
FODD	:	Fractional-Order Digital Differentiator

FODF	:	Fractional-Order Digital Filter
FPGA	:	Field Programmable Gate Array
GL	:	Grünwald-Letnikov
GFD	:	Generalized Fractional Differential
GFI	:	Generalized Fractional Integral
GLCM	:	Gray Level Co-occurrence Matrix
IE	:	Information Entropy
IIR	:	Infinite Impulse Response
MAE	:	Mean Absolute Error
MEG	:	Magnetoencephalography
MFOD	:	Matrix Fractional-Order Differentiators
MSE	:	Mean Square Error
MSSIM	:	Mean Structural Similarity Index Measure
NDM	:	Noise Detection Method
NRMSE	:	Normalized Root Mean Square Error
PLI	:	Power Line Interference
PP	:	Positive Predictivity
PSNR	:	Peak Signal to Noise Ratio
RBF	:	Radial Basis Function
RL	:	Riemann-Liouville
RMSE	:	Root Mean Square Error
S	:	Sensitivity
SP	:	Salt and Pepper
SSIM	:	Structural Similarity Index Measure

GLOSSARY OF SYMBOLS

α	:	Fractional-Order Operator
Γ	:	Gamma Function
$E_{a,b}(z)$	:	Two-parameter Mittag-Leffler Function
∇^2	:	Laplacian Operator
T	:	Sampling Period
L	:	Prescribed Filter Length
h	:	Step-size
μ	:	Total delay
D	:	Integer part of μ
d	:	Fractional part of μ
κ	:	Kernel Function
η	:	Iteration Number
ϵ	:	Threshold

List of Figures

Figure 2.1: Flowchart of the research work carried out in thesis 34

Figure 3.1: Bode diagram representing ideal (a) magnitude and (b) phase responses of FODD. 36

Figure 3.2: Nomenclature utilized for partitioning interval $[0, t]$ into subintervals $[p_k, p_{k+1}]$. 38

Figure 3.3: Magnitude responses (solid lines) of the (a) *proposed* RL-FODD and (b) GL-FODD. The dashed lines represent the ideal magnitude response of FODD. 42

Figure 3.4: Quantitative comparison between the magnitude response approximations attained by the *proposed* RL-FODD and conventional GL-FODD at various fractional orders. 44

Figure 3.5: Computation of fractional-order derivatives using FODD. 45

Figure 3.6: Ideal and estimated fractional-order derivatives of (a) $p_1(n)$ and (b) $p_2(n)$ for $\alpha = 0.8$ by using GL-FODD and RL-FODD. 47

Figure 3.7: (a) Magnitude response and (b) phase response (solid lines) of the *proposed* L-RL-FODD using parameters $L = 100, D = 20, d = 0.2, \alpha = 0.8$; (c) magnitude response and (d) phase response of the conventional L-GL-FODD. The dashed lines represent the ideal response of FODD. 52

Figure 3.8: (a) Magnitude response and (b) Phase response errors of the conventional L-GL-FODD and the *proposed* L-RL-FODD with respect to normalized frequencies (ω/π). 54

Figure 3.9: (a) Magnitude response and (b) phase response (solid lines) of the *proposed* RBF-RL-FODD using parameters $N = 100, L = 620, D = 30, d = 0.05, \alpha = 0.8$; (c) magnitude response and (d) phase response of the conventional RBF-GL-FODD. The dashed lines represent the ideal response of FODD. 57

Figure 3.10: (a) Magnitude response and (b) Phase response errors of the conventional RBF-GL-FODD and the *proposed* RBF-RL-FODD with respect to normalized frequencies (ω/π). 58

Figure 3.11: (a) Comparison among magnitude responses and (b) phase responses (in degrees) of L-GL-FODD with that of the *proposed* L-RL-FODD for $\alpha = 0.5$; (c) comparison among magnitude responses and (d) phase responses (in degrees) of RBF-GL-FODD with that of the *proposed* RBF-RL-FODD for $\alpha = 0.5$. 60

Figure 4.1: Comparison between (a) power-law and (b) Mittag-Leffler kernels for various values of α. 69

Figure 4.2: (a) Magnitude and (b) Phase response approximations of the *proposed L-ABC-FODF* for $\alpha = 0.5$; the dashed lines represent the ideal magnitude and phase response respectively. 76

Figure 4.3: MSE of the *proposed L-ABC-FODF* design for parameter d and $\alpha = 0.5, 0.7$. 76

Figure 4.4: (a) Magnitude and (b) Phase response approximations of the *proposed RBF-ABC-FODF* for $\alpha = 0.5$; the dashed lines represent the ideal magnitude and phase response. 78

Figure 4.5: MSE of the proposed *RBF-ABC-FODF* design for parameter d and $\alpha = 0.5, 0.7$. 79

Figure 4.6: MSE comparison between magnitude response approximation of proposed *L-ABC-FODF* and *RBF-ABC-FODF* for various fractional orders. 81

Figure 4.7: Block diagram for ECG R-peak detection using the proposed ABC-FODF. 82

Figure 4.8: ECG R-peak detection for segment of record no. 100 from MIT-BIH arrhythmia database. 83

Figure 4.9: ECG R-peak detection for segment of record no. 200 from MIT-BIH arrhythmia database. 84

Figure 4.10: Magnitude response of the *proposed* filter at $\alpha = 0.6$ and $\alpha = 1$. . . 87

Figure 4.11: Digital image sharpening approach. 89

Figure 4.12: (a) Grayscale "Lena" image (512×512) ; Sharpened grayscale images at (b) $\alpha = 0.3$ (c) $\alpha = 0.6$ (d) $\alpha = 0.9$ by utilizing the *proposed* approach. 91

Figure 4.13: (a) Colored "Lena" image ($512 \times 512 \times 3$) ; Sharpened colored images at (b) $\alpha = 0.3$ (c) $\alpha = 0.6$ (d) $\alpha = 0.9$ by utilizing the *proposed* approach. 93

Figure 4.14: Diabetic retinopathy (a) Original image ($605 \times 700 \times 3$); Sharpened images at (b) $\alpha = 0.3$ (c) $\alpha = 0.6$ (d) $\alpha = 0.9$ by utilizing the *proposed* method. 94

Figure 4.15: MRI image (a) Original image (286×276); Sharpened images at (b) $\alpha = 0.3$ (c) $\alpha = 0.6$ (d) $\alpha = 0.9$ by utilizing the *proposed* method. 94

Figure 4.16: Test images used for simulation from Set12 dataset. 95

Figure 5.1: Denoising performance of various methods on Image 5 of Set12 dataset corrupted by 0.03 variance Gaussian noise. 106

Figure 5.2: Noise variance versus (a) PSNR and (b) SSIM plots for comparison among various methods on Image 5 of Set12 dataset corrupted by Gaussian noise. 107

Figure 5.3: Denoising performance of various fractional-order masks on Lena image corrupted by salt and pepper noise at noise density 0.03. . . 109

Figure 5.4: Denoising performance of various fractional-order masks on MRI image corrupted by SP noise at density 0.04. 109

Figure 5.5: Denoising performance of various adaptive fractional-order masks on Lena image corrupted by salt and pepper noise at noise density 0.03. 112

Figure 5.6: Denoising performance of various fractional-order masks on MRI image corrupted by SP noise at density 0.04. 112

Figure 5.7: (a) PSNR and (b) SSIM comparison among various methods on all images of Set12 dataset corrupted by salt and pepper noise at density 0.07. 113

Figure 5.8: (a) PSNR and (b) SSIM comparisons among various methods on Lena Image affected by salt and pepper noise at various noise densities. 114

Figure 5.9: (a) PSNR and (b) SSIM comparison among several methods on MRI image affected by salt and pepper noise at various noise densities. 114

Figure 6.1: Middle (MD), Boundary (BD) and Corner (CR) pixels in an image and their neighboring pixels 126

Figure 6.2: Flowchart of the proposed GFIFD-AA 133

Figure 6.3: (a-e) Noisy MRI images corrupted by SP noise at various noise densities (ND) (f-j) Corresponding denoised images obtained by utilizing proposed GFIFD-AA along with the values of PSNR and SSIM. 136

Figure 6.4: Denoising performance of various techniques on Image 9 of Set12 dataset corrupted by SP noise at 90% noise density. 137

Figure 6.5: Denoising performance of various techniques on Image 68 of BSD68 dataset corrupted by SP at 50% noise density. 144

Figure 6.6: Denoising performance of various methods on chessboard image corrupted by SP noise at 50% noise density. 145

Figure 6.7: Denoising performance of various methods on synthetic image 1 corrupted by SP noise at 50% noise density. 146

Figure 6.8: Denoising performance of various methods on synthetic image 2 corrupted by SP noise at 50% noise density. 147

Figure 6.9: Denoising performance of various methods on synthetic image 3 corrupted by SP noise at 50% noise density. 148

Figure 6.10: Denoising performance of various methods on synthetic image 4 corrupted by SP noise at 50% noise density. 149

Figure 6.11: Denoising performance of various methods on synthetic image 5 corrupted by SP noise at 50% noise density. 149

Figure 6.12: Average processing time of various methods for all the images of Set12 dataset with all noise levels from 10% to 90% 151

List of Tables

Table 3.1: Quantitative comparison between GL-FODD and *proposed* RL-FODD in terms of MSE at various fractional orders 44

Table 3.2: Quantitative comparison between GL-FODD and the *proposed* RL-FODD at $\alpha = 0.8$ and varying values of h 46

Table 3.3: Quantitative comparison between GL-FODD and the *proposed* RL-FODD at various fractional orders and $h = 0.05$ 47

Table 3.4: MSE comparison between magnitude and phase responses of L-GL-FODD and the *proposed* L-RL-FODD 61

Table 3.5: MSE comparison between magnitude and phase responses of RBF-GL-FODD and the *proposed* RBF-RL-FODD 62

Table 4.1: Quantitative comparison among magnitude response approximations of the *proposed L-ABC-FODF* and *RBF-ABC-FODF* design methods for various fractional orders. 80

Table 4.2: Experimental results of the *proposed RBF-ABC-FODF* based R-peak detection algorithm 85

Table 4.3: Comparative analysis of the *proposed* method against well-established methods from the literature 86

Table 4.4: Image assessment parameters for grayscale “Lena” image at various fractional orders 92

Table 4.5: Comparative analysis of the *proposed* image sharpening method against well-established methods from the literature 96

Table 5.1: Fractional-order integral operators and their corresponding constants 102

Table 5.2: Superposition of GFI mask in all possible eight directions 104

Table 5.3: Denoising performance of various methods on Set12 dataset corrupted by zero mean Gaussian noise of variance 0.03. 108
Table 5.4: Quantitative comparison among different methods for an MRI image corrupted by salt and pepper noise. 115

Table 6.1: Fractional-differential operators and their corresponding parameters 119
Table 6.2: Comparative analysis between non-adaptive and adaptive fractional-order differential masks 124
Table 6.3: Quantitative comparison for Image 9 of Set12 dataset corrupted by SP noise at various noise densities. 140
Table 6.4: Quantitative comparison among different methods for all images of Set12 dataset corrupted by SP noise at 90% noise density. 142

Table of Contents

Declaration **iv**

Certificate **vi**

Acknowledgements **ix**

Abstract **xii**

List of Publications **xiv**

List of Abbreviations **xvii**

Glossary of Symbols **xviii**

List of Figures **xix**

List of Tables **xxv**

Table of Contents **xxvii**

1 Introduction **1**

1.1 Historic Background of Fractional-Order Calculus 1

1.2 Concept of Fractional-Order Differ-Integrals 3

1.3 Fractional-Order Signal and Image Processing 4

1.4 Contribution of Research Work 6

1.5 Organization of Thesis 8

2 Literature Survey **11**

2.1 Preliminaries of FOC 11

2.1.1 Special Mathematical Functions 11

2.1.1.1 Mittag-Leffler Function 11
2.1.1.2 Gamma Function 12
2.1.2 Definitions of Fractional-Order Operators 13
2.1.2.1 Riemann-Liouville (RL) Fractional-Order Operator 13
2.1.2.2 Grünwald-Letnikov (GL) Fractional-Order Operator . . . 14
2.1.2.3 Caputo Fractional-Order Operator 14
2.1.2.4 Caputo-Fabrizio (CF) Fractional-Order Operator 15
2.1.2.5 Atangana-Baleanu (AB) Fractional-Order Operator 15
2.1.3 Concept of Fractional Delay Interpolation 17
2.2 Preliminaries of Fractional-Order Digital Differentiators (FODD) 19
2.2.1 Applications of FODD 20
2.2.1.1 One-Dimensional Applications 20
2.2.1.2 Two-Dimensional Applications 21
2.2.2 Design methods of FODD 22
2.3 Preliminaries of Adaptive Fractional-Order Calculus 28
2.3.1 Adaptive Fractional-Order Image Processing 28
2.4 Motivation 29
2.4.1 Research Gaps 31
2.4.2 Objectives of Research 32
2.5 Research Methodolody 33

3 Riemann-Liouville Fractional Differential-Based Design of FODD 35
3.1 Concept of FODD: Signal Processing Perspective 35
3.2 Design of RL-based FODD (RL-FODD) using Finite Difference Method . . 37
3.2.1 Mathematical Investigation of RL-FODD 38
3.2.2 RL-FODD and GL-FODD − A Comparative Analysis 42
3.3 Need of Fractional Sample Delay Interpolation 48
3.4 Design of RL-FODD using Fractional Delay Interpolation (FDI) 50
3.4.1 Lagrange Interpolation based Design of RL-FODD (L-RL-FODD) . 50
3.4.1.1 Mathematical Investigation of L-RL-FODD 51
3.4.1.2 L-RL-FODD and L-GL-FODD – A Comparative Analysis 51
3.4.2 Radial Basis Function based Design of RL-FODD (RBF-RL-FODD) 55

3.4.2.1 Mathematical Investigation of RBF-RL-FODD 55
3.4.2.2 RBF-RL-FODD and RBF-GL-FODD – A Comparative Analysis 56
3.5 FDI based RL-FODD and GL-FODD − A Comparative Analysis 59
3.6 Summary 63

4 Atangana-Baleanu Fractional Differential-Based Design of Fractional-Order Digital Filter (FODF) 65
4.1 Atangana-Baleanu Fractional-Order Differential Operator 65
4.1.1 Atangana-Baleanu Fractional-Order Differential in RL sense 66
4.1.2 Atangana-Baleanu Fractional-Order Differential in Caputo sense . . 66
4.1.3 ABR and ABC Fractional-Order Differential Operators 67
4.1.4 Properties of AB Fractional-Order Differential Operator 68
4.2 Design of ABC-FODF using Finite Difference Method 70
4.3 Design of ABC-FODF using Fractional Delay Interpolation 73
4.3.1 Lagrange Interpolation Based Design of ABC-FODF (L-ABC-FODF) 74
4.3.2 Radial Basis Function Based Design of ABC-FODF (RBF-ABC-FODF) 77
4.4 Simulation Results 79
4.5 Applications of Proposed ABC-FODF 81
4.5.1 R-peak Detection in Electrocardiogram (ECG) Signals 81
4.5.1.1 Performance Metrics 85
4.5.1.2 Simulation Results and Comparative Analysis 85
4.5.2 Digital Image Sharpening 87
4.5.2.1 Performance Metrics 89
4.5.2.2 Simulation Results and Comparative Analysis 90
4.6 Summary 97

5 Generalization of Fractional Integrals: From Riemann-Liouville to Atangana-Baleanu for Image Denoising 99
5.1 Concept of Image Denoising: Two-Dimensional Signal Processing 99
5.2 Analytical Foundation of Generalized Fractional Integral (GFI) Mask 101

5.3 Comparative Analysis between GFI-RL and GFI-AB Fractional-Order Masks 104
5.3.1 Performance Metrics 104
5.3.2 Simulation Results Considering Gaussian Noise 105
5.3.3 Simulation Results Considering Salt and Pepper Noise 108
5.4 Adaptive Fractional-Order for Image Denoising 110
5.4.1 Mathematical Investigation of GFI-based Adaptive Mask (GFI-AM) 110
5.4.2 Comparative Analysis between GFI-AM-RL and GFI-AM-AB . . . 111
5.5 Comparative Analysis between GFI and GFI-AM Fractional-Order Masks . 113
5.6 Summary 116

6 Generalized Fractional Integral and Fractional Differential (GFIFD) based Adaptive Algorithm for Image Denoising 117
6.1 Generalization of Fractional-Order Differential Operators 118
6.1.1 Analytical Foundation of Generalized Fractional Differential (GFD) Mask 119
6.1.2 Mathematical Investigation of GFD-based Adaptive Mask (GFD-AM) 121
6.1.3 Image Enhancement using GFD-based Adaptive Mask 123
6.2 Proposed GFIFD-based Adaptive Algorithm (GFIFD-AA) 124
6.2.1 Noise Detection Method (NDM) for Salt and Pepper Noise 125
6.2.1.1 Probability Strategy for the Prediction of Noisy pixels . . 125
6.2.1.2 Detection of Noisy Pixels using Gradient Information . . 128
6.2.2 GFI-AM and GFD-AM based Image Denoising 131
6.3 Performance Analysis of the Proposed GFIFD-AA 133
6.3.1 Based on Various Levels of Noise Density 135
6.3.2 On Images with Different Characteristics 136
6.4 Comparative Analysis 138
6.4.1 Performance Metrics 138
6.4.2 Simulation Results and Comparison 139
6.4.2.1 On Images from Standard Datasets 139
6.4.2.2 On Black and White Images 143
6.4.2.3 On Synthetic Images 145
6.4.3 Computational Time 150

6.5 Summary 151

7 Conclusions and Future Work **153**

7.1 Conclusion 153

7.2 Future Work 156

Bibliography **159**

Chapter 1

Introduction

OVER THE PAST FEW DECADES, the domain of fractional-order calculus (FOC) has drawn a huge interest of all generations of scientists and has found significant applications in various fields, including but not limited to: control theory, signal and image processing, filter design, chaos synchronization, and biomedical applications [1, 2, 3, 4, 5]. Now, the domain of FOC is at the stage where new applications are continuing to stimulate the minds of contemporary researchers and existing applications are continuing to expand in the diverse areas of science and engineering. This chapter reviews the historical context of FOC, concept of fractional-order digital filters and its tremendous applications in signal and image processing area with emphasis on its ability to enhance useful information and remove noise.

1.1 Historic Background of Fractional-Order Calculus

FOC, originated from the branch of pure mathematics, has kept the attention of eminent researchers from various fields of science and engineering even after 325 years of its existence. The theory of FOC has a long and prominent history which can be traced back to the very origins of traditional Newtonian calculus itself.

Historically, in 17th century, Issac Newton and Gottfried Leibniz discovered calculus independently. However, in 1965, in a letter to L'Hôpital, Leibniz posed a query about the possibility of generalizing the order of derivatives and integrals to non-integer orders. L'Hôpital, somewhat curious, replied by posing another question to Leibniz: *"What if the order will be 1/2?"* The prophetical answer of Leibniz, i.e., *"It will lead to a paradox, a paradox from*

which one day useful consequences will be drawn", paved the way to the concept of FOC [6]. The magnificent response of Leibniz attracted the attention of various mathematical giants, for instance, Euler (1738), Laplace (1812), Lacroix (1819), Fourier (1822), Abel (1823), Liouville (1832-1855), Grünwald (1867), Letnikov (1868-1872), Riemann (1876), Laurent (1884), Heaviside (1893-1912), Weyl (1917), Lévy (1923), Marchaud (1927), Davis (1924-1936), Post (1930), Zygmund (1935-1945), Love (1938-1996), Erdelyi (1939-1965), Kober (1940), Widder (1941), Riesz (1949), Feller (1952) during the 18th and 19th centuries. In 1823, Abel became the first researcher to introduce the first application of FOC by investigating the tautochrone problem using the concepts of FOC [7].

Since the 70s only, FOC has been the subject of specialized conferences. For the first conference, the credit goes to B. Ross, who organized the first conference on *Fractional Calculus and its Applications* in June 1974. For the first monograph, the credit is ascribed to the joint collabration of K.B. Oldham and J. Spanier, who published a book devoted to FOC in 1974 [8]. FOC underwent two stages: From its beginning to the 1970s, when several mathematicians studied FOC as an abstract concept; and after 1970s, which shifted a paradigm of research from pure mathematical manipulations of FOC to applications in various areas of science and engineering [9].

During the last few decades, the success of FOC in various fields has become unquestionable with numerous applications. This is also due to the fact that it can describe many natural and man-made phenomena accurately that embeds long range memory effects, which are neglected by traditional integer-order calculus. This has unfolded plethora of models in the domain of FOC and hence, has led to its huge development, in theoretical, numerical and applied perspectives [10].

Since FOC was developed from the pure mathematical viewpoint, there exists number of competing definitions of fractional-order derivatives and integrals proposed by many famous mathematicians. The predominant definitions that are present since the beginning of the era of FOC includes Grünwald-Letnikov (GL) and Riemann-Liouville (RL) fractional operators [8]. After that, many other variants have evolved, for example, Caputo, Weyl, Riesz, Marchaud-Hadamard, to name a few. The predominant definitions of fractional-order operators, i.e., GL, RL, Caputo, and many others involve singular power-law as their kernel functions. Therefore, more recently, many mathematicians put forward the definitions of

fractional-order operators based upon various non-singular kernels such as, exponential kernel and Mittag-Leffler kernel. Each definition has its own advantages and disadvantages that have been explored by the research community in various fields of science and engineering [11].

On the other hand, from signal and image processing viewpoint, most of the research work is carried out by taking power-law kernel based fractional-order operators [12, 13] into consideration. However, very little emphasis has been laid on the performance of various other kernels for filtering applications. Therefore, the focus of this thesis is to establish fractional-order digital filters based upon singular and non-singular fractional-order differential operators and to investigate their performance analysis for signal and image processing applications.

1.2 Concept of Fractional-Order Differ-Integrals

The mathematical tool of FOC which is, in fact, the generalization of integer-order calculus, is concerned with differentiation and integration of non-integer orders. In simple terms, one can easily compute first-order, second-order or even *n*th-order derivative/integral of a simple function, say $f(x) = x^3$, by utilizing the concepts of integer-order calculus. However, if one wants to compute half-order derivative or half-order integral of the given function, then how is it possible? This curosity of Leibniz gave birth to the idea of FOC and hence, led to the concept of fractional-order differentiation and integration [8].

For someone who is hearing about FOC and its applications for the first time, it can be explained in lucid terms, as follows:

Recalling the concept of classical integration, where integration (i.e., $f^{(-1)}(x) = \int f(x)dx$) of a function, say $f(x)$, simply implies area under the curve i.e., all values of the function from the present point to start of the function are taken into account and are added together to carry out the integration process. On the other hand, fractional-order integration, just like classical integration, is also the area under the curve, but it is an *'area under the shape-changing curve'* [11]. Fractional-order integration also takes into account all past values of the function but in this case, different weights (having decreasing values) are multiplied by previous and past values of the function. Thus, it can be said that fractional-order integration

is a process with 'fading memory', which is in analogy with reality where past memory fades as the time passes by [2].

Furthermore, recalling the concept of classical differentiation, where differentiation (i.e., $f^{(1)}(x) = \frac{d}{dx}f(x)$) is defined as the slope of the function at a particular point. Unlike classical calculus, where differentiation at one point relies only on the values in the vicinity of that point, fractional-order differentiation at a point depends upon the characteristics of the entire function [8]. Fractional-order differentiation has an in-built fractional-order integration process and hence, it is also termed as fractional-order differ-integral. In this case, all the past values of the functions (multiplied with decreasing weights) are taken into account in order to compute fractional-order derivative at a particular point. Thus, it can be said that fractional-order differentiation is non-local in nature in contrast to classical differentiation which is local [11].

Owing to these inherent characteristics of memory and non-locality, FOC has shown remarkable progress in various scientific areas such as fractional control, viscoelasticity theory, diffusion processes, fractal dynamics, fractance, signal processing and image processing [10]. Its popularity in many discplines verified that FOC-based models can capture properties and phenomena that the classical integer-order calculus based models merely neglects.

1.3 Fractional-Order Signal and Image Processing

Digital filters have been the essence of signal and image processing applications from the past few decades. Digital filters aim at extricating valuable parts and removing unwanted parts of the signal and are widely popular in areas such as, signal and image processing, control systems, computer vision, seismic data processing, bio-informatics, communications etc. [14, 15, 16, 17, 18]. Filters based upon traditional integer-order calculus have found tremendous applications in various fields. However, they possess various shortcomings, for example, ringing artifacts and staircase effects [2, 19], due to which they are not able to model many real-world challenges accurately.

Therefore, in recent years filters based upon the concept of FOC have gained powerful insight in the research community due to their significant advantages over integer-order counterparts. Due to its outstanding features, the domain of FOC has become ubiquitous, seeing

considerable number of applications in all kinds of fields, from fractional diffusion equations and control systems to signal processing and biomedical applications [10, 20].

From the perspective of signal and image processing literature [2, 3], the generalization from integer orders to non-integer order has found abundant applications in areas such as image edge detection [21], in modeling of voltage source converters [22], in synthesis of multifractional Gaussian noise [23], in 2-D block FIR filters using distributed arithmetic [24, 25], in digital image sharpening [13], in dielectric spectroscopy [26], in modeling of speech signals and detection of electrocardiogram (ECG) waves [27], to name a few. Apart from these encouraging applications that have rendered FOC as an interesting research topic, the study of fractional-order digital differentiators (FODD) has gained momentum as it provides enhanced flexibility in desigining digital differentiating systems. They provide an extra degree of freedom which helps in optimizing the performance of methods based upon traditional integer-order calculus [4, 12].

Therefore, significant attention has been paid by the research community on design and development of FODD i.e., digital discrete-time systems that can perform non-integer order differentiation. Fractional/non-integer order differentiation implies that the integer order n of the derivative $D^n f(x) = d^n f(x)/dx^n$ of the function $f(x)$ is generalized to the fractional-order $D^\alpha f(x)$ in the research domain of FOC, where α represents a real-number [8]. Furthermore, it has become an important mathematical tool that can provide more specific details in comparison to integer-order counterparts [12].

For this reason, the design of FODD has become the main focus of interest among signal and image processing research community from past few decades [4]. To understand why the domain of FOC is catching on among signal and image processing research community, some of the reasons can be listed as follows:

1. Most of the real-time signals, for example, ECG, speech and music, etc. have spectra that do not decrease or increase by the multiples of 20 dB/decade. FOC provides substantially important flexibility in this aspect which has been proved to be beneficial for modeling and identification research [7].

2. Fractional Brownian motion (fBm) has attracted attention of various researchers due to its importance in many practical systems. The analysis and synthesis of such signals

are done by utilizing various methods such as, wavelets. But, modeling of these signals by using fractional derivatives have been proved to be most efficient [28].

3. In a digital image, texture details and weak edges correspond to low-frequency parts; boundaries and noise correspond to high-frequency ones. If integer-order derivative based methods are employed, then weak edges and texture details tend to suffer immensely; and noise tends to strengthen enormously. However, fractional-order differential based methods have been proved to be appropriate for overcoming these disadvantages because they have the tendency to preserve texture and edge details as well as to oppose noise to some degree [10, 13].

Apart from this, there is a growing list of applications indicating the enormous success of FOC in signal and image processing areas. However, very few studies have been conducted on the design and development of fractional-order digital filters based on different kernel functions for signal and image processing applications. Therefore, focus of this research work is to explore the applicability of various fractional-order kernels in signal and image processing applications with emphasis on their ability to enhance useful information and remove noise.

1.4 Contribution of Research Work

FOC concept emerged from the generalization of integer-order calculus, where it is often contemplated as branch of mathematical analysis dealing with integro-differential equations [8]. In FOC, fractional-order integral is defined in terms of convolution, where the filter or kernel function is of power-law type. A natural question that arises is: Why the kernel is of power-law type? A direct response to this query lies in the fact that fractional-order integral originated from the generalization of $n-$fold integer-order successive integration rule *a.k.a.* Cauchy's integral formula. Therefore, association of power-law in Cauchy's integral formula led to the kernel of power-law type in fractional-order integral and differential operators. Fractional-order operators exhibiting power-law kernel have found tremendous applications in many natural and man-made phenomena [29].

However, it has been pointed out recently by various researchers that there exists an artificial singularity at the initial point of the power-law kernel [30, 31], due to which many real-life

phenomena cannot be implemented exactly. On the other hand, different physical scenarios require different distributions [29, 32]. Therefore, several mathematicians made an attempt to replace the power-law kernel with other appropriate kernels (such as, exponential kernel, Mittag-Leffler kernel) and hence, led to the development of various fractional-order operators based upon different kernel functions [33]. These recently introduced fractional-order operators have found many applications in modelling many real-life phenomena in the fields of heat transfer [31], chaos theory [34], variational problems [35, 36], biomedical image analysis [37], and to name a few.

However, the profound advantages of these fractional-order operators are yet to be explored in detail in the field of signal and image processing. Therefore, based upon these aforementioned concepts, the initial research work in this thesis aims at designing fractional-order digital filters based upon singular and non-singular fractional-order kernels for one-dimensional (1D) and two-dimensional (2D) applications.

Furthermore, as there exist various definitions of fractional-order operators, several fractional-order masks exist in literature corresponding to these definitions for image processing applications. Therefore, to avoid the construction of fractional-order masks for each definition separately and to explore the potential of recently introduced fractional-order kernels for image processing applications, further research is carried out to put forward a framework for the design of generalized fractional integral (GFI) and generalized fractional differential (GFD) masks. The proposed generalized fractional-order masks aims to provide a more flexible and appropriate tool for image denoising and image enhancement applications and hence, avoids the construction of different masks for each kernel separately.

On the other hand, most of the traditional image processing methods based on fractional-order masks employ a fixed fractional-order to deal with the noise present in the images. However, the characteristics of an image varies locally. Hence, it is difficult to obtain better image denosing effect by using a fixed fractional-order to process noisy pixels in distinct regions. This is due to the fact that if higher fractional-order is chosen, the texture details are lost. On the other hand, if lower fractional-order is chosen, denoising performance is degraded. Hence, the proposed generalized masks are further extended to GFI-based adaptive mask (GFI-AM) and GFD-based adaptive mask (GFD-AM), respectively.

Subsequently, the proposed GFI-AM and GFD-AM masks are utilized to develop a general-

ized fractional integral and fractional differential based adaptive algorithm (GFIFD-AA) for the detection and removal of salt and pepper noise in images. The proposed GFIFD-AA consists of novel noise detection method that helps in detecting as well as locating noisy pixels in natural, black and white, and synthetic images effectively. The proposed GFIFD-AA not only removes the noise, but also preserves and enhances the texture and edge information of the image simultaneously. Various types of images from standard datasets such as, natural, medical and synthetic images are taken into account to illustrate the effectiveness and robustness of the proposed method against state-of-the-art methods.

1.5 Organization of Thesis

The research work carried out in this thesis is assembled in seven chapters, as follows:

Chapter 2: Literature Survey

To well situate the reader with the notion of FOC, this chapter provides the in-depth study of various definitions of fractional-order operators, design and applications of fractional-order digital differentiators (FODD), and the concept of adaptive fractional-order filters. The work already reported in the past covering these areas is reviewed to consolidate the work carried out in this thesis. Furthermore, the motivation behind this research work, research gaps, objectives and research methodology are also discussed in this chapter.

Chapter 3: Riemann-Liouville Fractional Differential-Based Design of FODD

This chapter intends to apply a new mathematical approach based on RL fractional-order differential operator incorporating power-law kernel to the design of fractional-order digital differentiator (FODD). A closed-form analytical expression is formulated for computing finite impulse response (FIR) filter coefficients of RL-based FODD. Then, the performance of the proposed RL-FODD is compared against well-established designs of GL-based FODD, based upon their ability to yield ideal frequency response of FODD. Furthermore, the efficacy of the proposed method is validated by computing fractional-order derivatives of the given signals by utilizing the proposed RL-FODD.

Chapter 4: Atangana-Baleanu Fractional Differential-Based Design of Fractional-order Digital Filter (FODF)

In this chapter, fractional-order digital filter (FODF) based upon Atangana–Baleanu

fractional-order differential operator is proposed, which incorporates Mittag-Leffler kernel instead of standard power-law kernel. The closed-form expression for the proposed FODF is derived by utilizing backward finite difference method and fractional sample delay interpolation techniques. Furthermore, preliminary applications of the proposed method in accurate delineation of R-peaks in ECG signals and digital image sharpening are reported.

Chapter 5: Generalization of Fractional Integrals: From Riemann-Liouville to Atangana-Baleanu for Image Denoising

This chapter puts forward a generalized fractional integral (GFI) mask which incorporates fractional-order kernels such as power-law and Mittag-Leffler law corresponding to RL and AB fractional integral operators, respectively. Then, the denoising performance of both kernels is validated and compared against each other for images affected by impulse (*a.k.a.* salt and pepper) noise and Gaussian noise. Furthermore, to avoid the difficulty of choosing fractional-order manually, GFI-based adaptive mask (GFI-AM) is presented, where fractional-order is chosen based upon local features of the given image. Several standard and medical images are taken into account to illustrate the effectiveness of the proposed masks in comparison to state-of-the-art methods.

Chapter 6: Generalized Fractional Integral and Fractional Differential based Adaptive Algorithm (GFIFD-AA) for Image Denoising

In this chapter, a generalized framework for the design of fractional differential mask is proposed which incorporates various definitions of fractional-order differential operators, such as, Riemann-Liouville (RL)/Caputo, Caputo-Fabrizio (CF), Atangana-Baleanu (AB), etc. Furthermore, for the detection of salt and pepper noise from images as well as to obtain better denoising effect, a generalized fractional integral and fractional differential based adaptive algorithm (GFIFD-AA) is put forward.

In the proposed GFIFD-AA, a novel noise detection method (NDM) is developed to detect noisy pixels in the image corrupted by salt and pepper noise. The detected noisy pixels are then processed by utilizing GFI-based adaptive mask (GFI-AM). The noise-free pixels are then updated by utilizing the proposed GFD-based adaptive mask (GFD-AM) so as to preserve and enhance the texture and edge details of an image simultaneously. To show the effectiveness of the proposed GFIFD-AA method, several standard and medical images are taken into account. The advantages of the proposed method also lies in the fact that it works

well for black and white images as well as synthetic images. The results of the proposed method are validated in terms of various quantitative parameters to illustrate the effectiveness of the proposed approach.

Chapter 7: Conclusion and Future Work

This chapter summarizes the main contribution of the research work carried out in this thesis and suggests a few pathways to the future research.

Chapter 2

Literature Survey

THIS CHAPTER PROVIDES THE IN-DEPTH STUDY of various definitions of fractional-order operators, design and applications of fractional-order digital differentiators, and the significant advantages of utilizing adaptive fractional-order for image processing applications. Basically, this chapter elaborates the work already reported in the literature in order to consolidate the research carried out in this thesis. Furthermore, the motivation behind the research work including gaps and objectives of research as well as research methodology are presented in this chapter.

2.1 Preliminaries of FOC

To travel through the world of FOC, this section is segregated into three main components corresponding to some special mathematical functions, various definitions of fractional-order operators and the concept of fractional delay interpolation.

2.1.1 Special Mathematical Functions

The mathematical functions that have found wide applicability in the research domain of FOC are described as follows [2]:

2.1.1.1 Mittag-Leffler Function

In traditional integer-order calculus, exponential functions play a significant role when dealing with ordinary differential equations [2]. Similarly, in FOC, Mittag-Leffler function is

utilized as the solution of fractional-order differential equations. It can be said that in FOC, Mittag-Leffler plays the same role as exponential functions play for integer-order systems [2]. This function is named after the great Swedish mathematician Gösta Magnus Mittag-Leffler who studied its properties in 1902-1905 [38].

During the first half of the twentieth century, this function remained almost unknown among the research community. It was only later in the 1960s that this function gained some recognition. However, the real importance of Mittag-Leffler function became evident when its special role in FOC was discovered [39]. Now, the classical Mittag-Leffler function is regarded as the Queen Function of FOC. The one-parameter Mittag-Leffler function is defined as

$$E_a(z) = \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(ak+1)} \qquad (z, a \in \mathbb{C}; Re(a) > 0) \tag{2.1}$$

For $a = 1$, the Mittag-Leffler function reduces to the exponential function. Therefore, it can be said that the Mittag-Leffler function provides a simple generalization of the exponentional function.

Furthermore, two-parameter Mittag-Leffler function is described as

$$E_{a,b}(z) = \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(ak+b)} \qquad (z, b \in \mathbb{C}; Re(a) > 0) \tag{2.2}$$

When $b = 1$, it can be seen from (2.1) and (2.2) that $E_{a,1}(z) = E_a(z)$. Therefore, it can be said that one-parameter Mittag-Leffler function is a special case of two-parameter Mittag-Leffler function. In some applications, Mittag-Leffler functions with three or four parameters are used, the comprehensive details of which can be referred from [38].

2.1.1.2 Gamma Function

Gamma function, introduced by L. Euler in 1729, is one of the fundamental functions of FOC. It provides the extension of factorial function for non-integer numbers [8]. It is defined by Euler's integral of second kind as follows:

$$\Gamma(z) = \int_0^{\infty} t^{z-1} e^{-t} dt, \qquad z \in \mathbb{C} \tag{2.3}$$

where $t^{z-1} = e^{(z-1)log(t)}$. The integral presented in (2.3) is convergent in the right half of the complex plane i.e., for $\mathrm{Re}(z) > 0$ [2, 8]. Furthermore, some of the properties of Gamma function are listed as follows:

1. $\Gamma(z+1) = z\Gamma(z)$

2. $\Gamma(1) = 1$; if $n \in \mathbb{Z}$, then $\Gamma(n+1) = n!$

3. $\Gamma(1/2) = \sqrt{\pi}$

2.1.2 Definitions of Fractional-Order Operators

Since the existence of FOC, various definitions of fractional-order operators (i.e., fractional-order differentials and fractional-order integrals) have been proposed in the literature. The most intensively-studied and extensively-utilized definitions of fractional-order operators include Riemann-Liouville (RL), Grünwald-Letnikov (GL) and Caputo fractional-order operators [8]. However, in recent years, several other variants of fractional-order operators, such as Caputo-Fabrizio (CF) [30] and Atangana-Baleanu (AB) [31] have been put forward. The detailed overview of various defintions of fractional-order operators is presented as follows:

2.1.2.1 Riemann-Liouville (RL) Fractional-Order Operator

RL definition of fractional-order operators, which originated in the late 19th century, is the most common way to define the concept of FOC. The RL definition of fractional-order integral emerged from the generalization of Cauchy's formula for repeated integration [8, 32]. The RL-based fractional-order integral (denoted by I^{α}_{RL}) of the function $s(t)$ with respect to t is defined as [8]

$$I^{\alpha}_{RL}s(t) = \frac{1}{\Gamma(\alpha)}\int_0^t (t-\zeta)^{\alpha-1}s(\zeta)d\zeta \tag{2.4}$$

where α represents the fractional-order such that $\mathrm{Re}(\alpha) > 0$ and $\Gamma(\cdot)$ represents the Gamma function, as defined in (2.3). Furthermore, in FOC, fractional-order differentials are generally defined via fractional-order integrals. The α^{th}-order RL-based fractional-order differential (denoted by D^{α}_{RL}) emerged from the inversion of the generalization of n-fold integer-

order successive integration rule [8] and is defined as

$$\begin{aligned} D^{\alpha}_{RL}s(t) &= \frac{d^n}{dt^n} I^{n-\alpha}_{RL} s(t) \\ &= \frac{1}{\Gamma(n-\alpha)} \frac{d^n}{dt^n} \int_0^t (t-\zeta)^{n-\alpha-1} s(\zeta) d\zeta \end{aligned} \tag{2.5}$$

where $n-1 < \alpha < n$ and $n \in \mathbb{N}$ [8].

2.1.2.2 Grünwald-Letnikov (GL) Fractional-Order Operator

Another most popular definition of fractional-order operators emerged from the direct generalization of the limit of difference quotients defining integer-order differentials [8], which is most commonly known as Grünwald−Letnikov (GL) fractional-order differential operator. The GL-based fractional-order differential (denoted by D^{α}_{GL}) of the function $s(t)$ with respect to t is defined as [8].

$$D^{\alpha}_{GL}s(t) = \lim_{h \to 0} \frac{h^{-\alpha}}{\Gamma(-\alpha)} \sum_{k=0}^{t/h} \frac{\Gamma(k-\alpha)}{\Gamma(k+1)} s\left(t - kh\right) \tag{2.6}$$

where h represents the step size. In this case, when the fractional-order α is positive, then the above definition corresponds to GL-based fractional-order differential. However, when α is negative, it corresponds to GL-based fractional-order integral. Due to its discrete-form and flexibility, the GL-based fractional-order differential has been widely utilized in various applications, such as control theory, signal and image processing, etc. [2, 12].

2.1.2.3 Caputo Fractional-Order Operator

Caputo fractional-order differential, introduced by M. Caputo in 1967, appeared as a modification of RL fractional-order differential operator. Caputo-based fractional-order differential (denoted by D^{α}_{C}) is defined as

$$D^{\alpha}_{C}s(t) = \frac{1}{\Gamma(n-\alpha)} \int_0^t (t-\zeta)^{n-\alpha-1} s^{(n)}(\zeta) d\zeta \tag{2.7}$$

In contrast to RL fractional-order differential, the initial conditions in Caputo fractional-order differential are formulated in terms of integer-order derivatives and hence, it is useful in finding solutions to fractional-differential equations dealing with initial value problems

[2].

The RL, GL, and Caputo fractional-order differentials have been widely utilizied to deal with processes with power-law behaviour, as they exhibit power-law as their kernel function. However, there exist many natural processes that exhibit different distributions and hence, cannot be simply described by power-law functions [30, 31, 40].

Therefore, in the recent past, the trend is to go beyond power-law kernels and to consider various other kernel functions in the FOC research community.

2.1.2.4 Caputo-Fabrizio (CF) Fractional-Order Operator

In 2015, Caputo and Fabrizio [30] introduced yet another definition of fractional-order differentials whose kernel function is of exponential type. The α^{th}-order CF fractional-order differential (denoted by D^{α}_{CF}) is defined as [30]:

$$D^{\alpha}_{CF}s(t) = \frac{M(\alpha)}{1-\alpha}\int_{a}^{t} exp\left(\frac{-\alpha(t-\zeta)}{1-\alpha}\right)s'(\zeta)d\zeta \tag{2.8}$$

where $M(\alpha)$ is the normalization constant such that $M(0) = M(1) = 1$ [30] and $s'(\zeta)$ represents the first-order derivative of the function. The CF fractional-order differential eliminates the limitation of singularity in power-law kernel and proved to be efficient in describing the behaviour of classical viscoelastic materials, electromagnetic systems and thermal media, etc. [30] However, a couple of issues were indicated against this fractional-derivative lately, where it was stated that the exponential kernel associated with CF fractional-order differential is not non-local in nature, and the fractional-order integral associated with this definition is not a fractional operator [41, 42].

2.1.2.5 Atangana-Baleanu (AB) Fractional-Order Operator

Atangana and Baleanu, in 2016 suggested the Mittag-Leffler function as a new kernel and hence, put forward the definition of AB-based fractional-order operators [31]. Atangana and Baleanu proposed two novel fractional-order differential operators i.e., Atangana-Baleanu in Caputo sense (i.e., ABC)) and Atangana-Baleanu in RL sense (i.e., ABR) based upon non-local and non-singular Mittag-Leffler function which itself possesses a strong connection with fractional differential theory [40]. The Atangana-Baleanu fractional-order differential

(denoted by D^{α}_{ABC}) of the function $s(t)$ in Caputo sense (ABC) is defined as [31]

$$D^{\alpha}_{ABC}s(t) = \frac{M(\alpha)}{1-\alpha}\int_0^t s'(\zeta)E_{\alpha}\left(-\frac{\alpha}{1-\alpha}(t-\zeta)^{\alpha}\right)d\zeta \tag{2.9}$$

where α represents the fractional-order such that $0 < \alpha < 1$, $E_{\alpha}(z) = \sum_{k=0}^{\infty}\frac{z^k}{\Gamma(\alpha k+1)}$ represents the one-parameter Mittag-Leffler function, $M(\alpha)$ is a normalization constant defined as $M(\alpha) = 1-\alpha+\frac{\alpha}{1-\alpha}$ [40] and $s'(\zeta)$ represents the first-order derivative of the function.

Furthermore, Atangana-Baleanu fractional-order differential (denoted by D^{α}_{ABR}) in Riemann–Liouville sense (ABR) is defined as [31]

$$D^{\alpha}_{ABR}s(t) = \frac{M(\alpha)}{1-\alpha}\frac{d}{dt}\int_0^t s(\zeta)E_{\alpha}\left(-\frac{\alpha}{1-\alpha}(t-\zeta)^{\alpha}\right)d\zeta \tag{2.10}$$

Both definitions have their own advantages, and the detailed analytical study of ABC and ABR fractional-order differentials can be referred from [31, 40].

On the other hand, the fractional-order integral associated with Atangana-Baleanu fractional-order differential operators is defined by [31]

$$I^{\alpha}_{AB}s(t) = \frac{1-\alpha}{M(\alpha)}s(t) + \frac{\alpha}{M(\alpha)\Gamma(\alpha)}\int_0^t s(\zeta)(t-\zeta)^{\alpha-1}d\zeta \tag{2.11}$$

The AB fractional-order integral is a more generalized version than RL fractional-order integral, as it can be observed that it provides an average between a given function and its RL fractional-order integral [40].

The AB fractional-order operators have found extensive applications in areas such as, heat transfer [31], chaos theory [34], variational problems [35, 36], biomedical analysis [37] etc. The AB fractional-order operators exhibit all the benefits of RL and Caputo fractional-order differentials, in addition, the kernel associated is non-singular. Moreover, it inherits all the advantages of CF fractional-order differential; in addition, the kernel associated is non-local [40]. Furthermore, fundamental differences among power-law, exponential-law, and Mittag-Leffler kernels have been presented in [43] along with their potential applications in various fields. Based on these aforementioned concepts, the potential of AB fractional-order operators is explored in this thesis for various signal and image processing applications.

2.1.3 Concept of Fractional Delay Interpolation

Fractional delay interpolation possesses a wide range of applications in digital signal processing and communications, where a delay that is a fraction of the sampling period is required. Delaying a digital signal by a real amount of the sampling interval is the main feature of digital fractional delay filters [44, 45]. In recent years, fractional delay filters have become an essential device in various applications which include, speech coding and synthesis [45, 46], timing synchronization in digital receivers [47], beemsteering of antenna arrays [48], comb filter design [49], analog to digital conversion [50], design of optimum integrators and differentiators [51], and to name a few.

The transfer function of an ideal fractional delay filter is given by [44]

$$M(z) = z^{-\mu}, \mu \in \mathbb{R}^{+} \tag{2.12}$$

where μ represents a delay that can be split into integer part D and fractional part d.

Despite the apparent simplicity of the above transfer function, it is not easy to implement digitally. Therefore, many digital finite impulse response (FIR) and infinite impulse response (IIR) filter design techniques have been put forward in the research community to approximate the ideal transfer function of the fractional delay filter. A comprehensive survey on the definition of fractional delay, its characteristics and the design of fractional delay filters is available in [44].

The design problem of fractional delay filter involves finding a digital filter such that it provides an approximation of the ideal fractional delay μ, presented in (2.12), as precisely as possible. The z-domain transfer function of the N^{th}-order FIR fractional delay filter can be expressed as

$$M(z) = \sum_{m=0}^{N} q(m) z^{-m} \tag{2.13}$$

where $q(m)$ represents the FIR filter coefficients of the fractional delay filter that will approximate the transfer function of an ideal fractional delay filter, presented in (2.12).

There exist various design methods for obtaining filter coefficients of the fractional delay filters [44, 52, 53, 54]. The fractional-delay interpolation methods that have been utilized in this work are discussed here as presented below:

Lagrange Interpolation

Lagrange interpolation, usually implemented using a direct-form FIR filter structure, is one of the techniques often used for fractional delay approximation. Lagrange interpolation is equivalent to maximally flat approximation and is obtained by setting the error function and its derivatives equal to zero at $\omega = 0$ [44]. The derivation of the maximally flat Lagrange FIR filter is detailed in [44]. The closed-form formula to compute the coefficients of an N^{th}-order Lagrange fractional delay filter is given as

$$q(m) = \prod_{l=0, l\neq m}^{N} \frac{\mu - l}{m - l}, m = 0, 1, ..., N \tag{2.14}$$

where N represents the order of the filter and μ is the desired total delay. It is interesting to note that the coefficients of FIR filter obtained by this method are equivalent to the weighting coefficients of the classical Lagrange interpolation. Lagrange interpolation is one of the widely utilized methods in signal processing algorithms [2].

Radial Basis Function (RBF) Interpolation

Radial basis functions are means to approximate multivariable functions by linear combinations of terms based on a single univariate function, also known as the radial basis function [45]. A radial basis function $\phi(t)$ is a real-valued function whose value depends only on the distance from the origin, i.e., $\phi(t) = \phi(|t|)$, where $|.|$ denotes the absolute value. The most commonly used radial basis functions include Gaussian and inverse multiquadratic functions defined as [54]

$$\begin{aligned} \text{Gaussian} : \phi(t) &= e^{-t^2/\sigma^2} \\ \text{Inverse multi-quadratic} : \phi(t) &= \frac{\sigma}{\sqrt{t^2 + \sigma^2}} \end{aligned} \tag{2.15}$$

where σ is utilized to adjust the shape of $\phi(t)$ and hence, is known as shape parameter.

Given a set of $N + 1$ different points $\{t_0, t_1, ..., t_N\}$ and the corresponding real numbers $\{s_0, s_1, ..., s_N\}$, the interpolation problem is to find a function $s(t)$ that satisfies

$$s(t_k) = s_k, \quad k = 0, 1, ..., N \tag{2.16}$$

The RBF interpolation method consists of choosing a function $s(t)$ that has the following form [45, 54]

$$s(t) = \sum_{k=0}^{N} w_k \phi(|t - t_k|) \tag{2.17}$$

where t_k represents the data points where the value of $s(t)$ is known, t represents the data points where the value of $s(t)$ needs to be approximated, ϕ is a univariate radial basis function, and w_k represents the weighting coefficients. Given this information, the function $s(t)$ is represented as a sum of $N + 1$ radial basis functions weighted by the corresponding coefficients w_k. Substituting (2.16) into (2.17), provides a linear system of equations that can be used to compute the value of coefficients w_k, the details of which can be found in [45].

2.2 Preliminaries of Fractional-Order Digital Differentiators (FODD)

Digital differentiator is one of the most significant mathematical tools that is utilized to determine and estimate the time derivatives of the given signal. From the very beginning, digital differentiators have been utilized in various applications (for examples, radar and sonar applications) in order to compute velocity and acceleration from position measurements. Several methods exist in the literature to design digital differentiators for various applications. An excellent survey on design methods of digital differentiators is presented in [55].

However, over the past few years, the research has been shifted towards investigating the behaviour of digital differentiators in the fractional domain. The study of fractional-order digital differentiators (FODD) have gained momentum as they provide an extra degree of freedom to alter the characteristics of integer-order digital differentiators and hence, proves to be efficient in providing peculiar details as compared to integer-order counterparts [56].

FODD is basically concerned with estimating fractional-order derivatives of the given signal, where fractional-order differentiation implies that the integer-order n of the derivative $D^n s(t) = d^n s(t)/dt^n$ of the signal $s(t)$ is generalized to the fractional-order $D^\alpha s(t)$, where α represents the real-number [54]. It has turned out to be an important mathematical tool in signal and image processing research community due to its wide applicability in various applications. The profound applications of FODD can be listed here as follows:

2.2.1 Applications of FODD

FOC is a fast growing mathematical discipline, mostly because of its applications in broad areas of science and engineering. Employing FODD to signal and image processing applications is a burgeoning subject under discourse. The one-dimensional and two-dimensional applications of FODD are listed as follows:

2.2.1.1 One-Dimensional Applications

The focus of this section is to outline some applications of FODD to signal processing with emphasis on the ability of FODD to enhance useful information and remove noise. Therefore, the one-dimensional applications of FODD are presented here as follows:

(a) QRS detection in electrocardiogram (ECG) signals

ECG is an electrical signal of human heart activities and is composed of various pulses known as P, Q, R, S, and T waves [57]. In real-world scenarios, various artifacts such as power line interference (PLI), baseline wander (BW), instrumentation noise, etc. severely affect the morphology of ECG signals, and hence making its analysis a complex task [57, 58]. Therefore, for the detection of several cardiovascular diseases, an accurate delineation of R-peaks becomes a crucial job to investigate. Hence, there has been a surge of interest in applying the concepts of FODD for the accurate delineation of ECG signals [4, 59, 60] in the research community.

(b) Parameter estimation of fractional noise process

The discrete-time equivalent of fractional Brownian noise is commonly referred to as fractional noise process. Filtered versions of these processes are suitable to model signals that possess various short-term and long-term correlation structures [61].

In general, the fractional noise $f(n)$ is modelled as the output of the fractional-order integrator with frequency response $1/(j\omega)^{\alpha}$ driven by the zero-mean white noise $\zeta(n)$ with unit variance σ_f^2. The power spectral density of $f(n)$ is described as: $S_f(\omega) = \frac{\sigma_f^2}{\omega^{2\beta}}$. In order to estimate the parameters β and σ_f^2, FODD has found various advantages and has been widely employed in the literature [54, 56].

(c) Computing fractional-order derivatives of noisy signals

Chen *et al.* [62] amalgamated the concepts of FOC and Savitzky-Golay filter in order to estimate fractional-order derivatives of contaminated signal accurately. It utilizes least-squares method and RL definition of fractional-order operators to generalize the integer-order Savitzky-Golay differentiator to fractional-order. From the experimental studies conducted, it has been demonstrated that the proposed digital fractional-order Savitzky-Golay differentiator can efficiently estimate the fractional-order derivatives of both noise-free and contaminated signals.

2.2.1.2 Two-Dimensional Applications

Owing to the advantages of fractional-order operators in preserving smooth details of the images [13], they have been extensively employed for image enhancement, edge detection and image denoising applications. The two-dimensional applications of FODD are briefly summarized here as follows:

(a) Digital Image Sharpening

Digital image sharpening has found extensive applications in the areas as diverse as electronic printing, industrial inspection and medical imaging, etc. [63]. In digital image processing applications, the sharpening method is usually utilized to highlight fine details in digital images. Conventionally, spatial filters based on second-order derivatives (commonly known as Laplacian filter) were utilized to enhance fine details in the images that were blurred during acquisition of an image. However, over the past few decades, the trend has been shifted to perform the task of image sharpening in fractional domain after Pu *et al.* [13] introduced six fractional-order mask for the enhancement of texture rich images.

(b) Edge Detection in images

Edge detection comprises of a set of mathematical methods which aim at determining sharp discontinuities in an image. These discontinuities may arise at those points in an image where intensity or color of the image changes sharply. These points are arranged into a curved line segments known as edges [63]. In image processing, edge detection is mostly done by utilizing integer-order differentiation operators. However, these operators suffer from loss of

image details and are highly sensitive to noise. To overcome these disadvantages, fractional-order operators have been extensively employed for developing new image edge detection methods [21].

(c) Image denoising

The task of image denoising (i.e., removing noise and recovering the original image) is extremely important because quality of a digital image gets inevitably degraded during the process of acquisition and transmission [63]. The conventional image denoising methods adopt integer-order integrals directly or indirectly during the construction of a denoising model. On the contrary, fractional-order filters have been proved to be advantagenous because of their ability to preserve the information while denoising the image [64].

2.2.2 Design methods of FODD

As FODDs have gained significant importance in the reseacrh community due to their wide applicability in various signal and image processing areas, the design of FODD has attracted the attention of many researchers worldwide.

In recent decades, several methods have been developed in the literature for the design of FODD. From the perspective of signal and image processing applications, the pioneer work of A. Oustaloup [65] and C.C. Tseng [56, 66] can be cited, who studied the applications of fractional-order derivatives from frequency response viewpoint.

Oustaloup *et al.* [65] presented an initial comprehensive study on the synthesis of differentiators with integer, non-integer, real or complex orders. Tseng *et al.* [66] proposed a method to compute fractional-order derivative of a function by utilizing fractional differentiation property of the Fourier transform. Then, a least-squares method has been developed to design fractional-order digital differentiator. Finally, the proposed FODD has been utilized to generate a random fractal process.

Furthermore, Tseng [56] investigated the design of fractional-order FIR digital differentiator by utilizing the fractional derivative of the power function. Then, the impulse response of the proposed FODD has been obtained by solving linear equations. Finally, it has been demonstrated that fractional-order derivatives of digital signals can be computed by utilizing the proposed filtering technique.

Furthermore, various other methods for the design of FODD exist, which have been widely categorized into two parts in the literature: One dealing with continuous-time models, which include Carlson's method [67], Roy's method [68], Chareff's method [69], Matsuda's method [70] and Oustaloup's method [65] and the others dealing with discrete-time models, which include fractional differencing formula [71], Tustin method [72], continued fraction expansion [72], Newton's method [73], numerical integration method [74] and least-squares method [75]. An excellent survey on the study of fractional-order differentiators and integrators is presented in [12].

In this thesis, the design methods of FODD are segregated based upon various definitions of fractional-order operators being utilized.

Based on GL Fractional-Order Operator

A significant amount of work on the design of FODD has been done by C. C. Tseng by taking GL fractional-order differential into account. A brief summary on the work reported by Tseng is presented as follows:

Tseng improved the performance of conventional fractional differencing methods by incorporating fractional sample delay into the design process of FODD [53]. Furthermore, various tools of digital signal processing have been utilized to design FODD. For instance, discrete Fourier transform (DFT) [76], which is an important mathematical tool in digital signal processing area, has been utilizied to estimate non-integer delay samples of the signal. Then, GL definition of fractional-order differential and DFT based non-integer delay sample estimation have been utilized to obtain transfer function of FODD.

Furthermore, the non-integer delay samples have been estimated by utilizing various other transforms, such as, discrete Hartley transform (DHT) [77], discrete cosine transform (DCT) [78] and discrete sine transform (DST) [79]. These non-integer delay sample estimation and GL fractional-order differential have been utilized to put forward various designs of FODD.

On the other hand, Tseng put forward the design of FODD based upon radial basis function (RBF) interpolation method [54]. The RBF based non-integer delay sample estimation and GL fractional-order differential are applied in order to attain the transfer function of the FODD. The proper selection of shape parameter of RBF helps in reducing the design error of FODD. Finally, the applications in digital image sharpening and parameter estimation of

fractional noise process are investigated to illustrate the utility of the design method.

In addition, matrix fractional-order differentiators (MFOD) have been designed for computing fractional-order derivatives of digital signals [80]. To improve the performance of conventional FIR and IIR filters, DST, DCT and optimization methods are presented to design MFOD. Furthermore, numerical examples are presented to compare the performances of these methods and also, the design of variable MFOD is investigated. To show the effectiveness of MFOD, digital image sharpening and signal denoising applications are taken into account. Various other well-established methods have also been utilized for enhancement of signals and images [13, 81].

Apart from that, Tolba *et al.* [82] provided field programmable gate array (FPGA) implementation of fractional-order differentiator/integrator based on GL fractional-order operator. Two algorithms have been put forward to realize the GL operator: One based upon look-up table and another one based upon quadratic and piece-wise linear approximation approaches. To guarantee the validity of the design method, it has been further implemented for various input functions.

In general, the GL fractional-order operator appears to be an ideal generalization of the integer-order derivative. Furthermore, it is easy to implement digitally. Therefore, many numerical fractional differential algorithms that have been put forward in the research community are mostly based upon the GL fractional-order differential operator. Additionally, GL-based fractional-order differential provides a more comprehensive way to understand the nature of FOC [2].

However, several issues have been addressed in the well-established literature against the use of GL fractional-order differential operator in most applications. The issues reported are described as: (i) The implementation of $D^{\alpha}_{GL}s(t)$ at any point t requires the knowledge of the function $s(t)$ over the whole interval $(-\infty, t]$ [83]. (ii) The series presented in (2.6) converges only for a restricted number of functions, i.e. for functions that are bounded in $(-\infty, t]$ [83]. (iii) The GL-based approach does not yield exact differentiators for some signals, particularly those that can be modelled by finite-degree polynomials [84]. (iv) The GL-based scheme is not robust against noises [85], i.e. a minute error in the observation can result into a significant error in the computed fractional-order derivatives.

Therefore, efforts have been made by various researchers to put forward the design of FODD

based upon RL fractional-order differential operator.

Based on RL and Caputo Fractional-Order Operators

Samadi *et al.* [84] developed a discrete-time fractional-order differentiator to compute fractional-order derivatives of RL type for a uniformly sampled polynomial signal. In this approach, the input signal is represented in continuous-time by utilizing Newton series. The output is discrete-time signal equivalent to the fractional-order derivative of input polynomial at uniform intervals. Closed-form expressions to compute the coefficients of both integer-order and fractional-order differentiators have been obtained. However, the approach is based upon the assumption that the signal $s(t)$ is represented by truncated Newton series.

Chen *et al.* [62] utilized least-squares method and RL definition of fractional-order operators to generalize the integer-order Savitzky-Golay differentiator to fractional order for computing fractional-order derivatives of the contaminated signal. From the experimental studies conducted, it has been demonstrated that the proposed digital fractional-order Savitzky-Golay differentiator can efficiently estimate fractional-order derivatives of both noise-free and noisy signals.

Furthermore, robust fractional-order Jacobi differentiators have been designed in [85] to compute the RL and Caputo fractional-order derivatives of contaminated signals. This idea has also been extended using fractional-order Jacobi orthonormal functions in [85] to determine the fractional-order derivatives of signals contaminated with noise. These methods filtered the noisy signals, which are power functions, and their fractional-order derivatives also result in the form of power functions.

On the other hand, from the perspective of image processing applications, Amoako-Yirenkyi *et al.* [86] and Nandal *et al.* [87] proposed various fractional differential masks based on RL fractional differential operator for image enhancement and edge detection applications.

However, the general approach to design FODD based upon RL fractional–order differential operator has not been considered so far. Therefore, the initial research work in this thesis aims at devising an analytical framework for RL-based FODD and compares it with the conventional well-established design methods that are based upon GL fractional–order differential operator (GL-FODD). Since there exists a close relationship between RL and truncated GL fractional-order derivatives [83], the performance of both the methods in ap-

proximating the ideal frequency response of FODD has been compared rigorously.

On the other hand, the recent trend in the FOC domain has emerged where the interest has been shifted from singular kernel functions to non-singular kernel functions. Therefore, various new definitions have been developed that have shown significant contribution in signal and image processing applications.

Based on CF Fractional-Order Operator

Cruz-Duarte *et al.* [88] proposed a closed-form formulation for computing $(n + \alpha)$-order fractional derivative of Gaussian function by utilizing CF fractional-order differential operator. This study is based on the fact that Gaussian function and its derivatives have been broadly employed in various applications, specifically in signal processing. Finally, three illustrative signal processing applications of the deduced formula have been studied which include, customisable wavelets, image processing filters and probability distribution functions like the fractional Rayleigh distribution.

Zhang *et al.* [89] presented a highly selective and noise-robust edge-relevant structure feature detection method based on fractional-order Gaussian derivatives by utilizing the definition of CF fractional-order differential operator. The closed-form expression of fractional-order Gaussian derivatives has been derived by using non-singular kernel based CF fractional-order differential operator. Experimental results demonstrated that the proposed method has strong edge feature selectivity as well as strong noise immunity.

Lavìn-Delgado *et al.* [90] considered CF fractional-order differential operator to generalize the first-order Sobel edge detector operator. The fractional mask based on non-singular exponential kernel has been utilized to carry out edge detection of medical images such as mammograms, angiograms, and retinal images. From the simulation studies conducted, it has been validated that the proposed operator has higher edge detection capability and hence, provides superior performance over conventional integer-order counterparts.

Apart from the widespread popularity of CF fractional-order operator in signal and image processing applications, a couple of issues were indicated against this fractional-derivative lately [41, 42].

Therefore, to solve the above-mentioned problems of singularity and locality in fractional-

differential operators, Atangana and Baleanu [31] proposed a novel fractional-order differential operator based upon non-local and non-singular Mittag-Leffler function which itself possesses a strong connection with fractional differential theory [38]. The AB fractional-order operator exhibits all the benefits of RL and Caputo fractional-order differential operators, as well as the kernel associated is non-singular. Moreover, it inherits all advantages of the CF fractional-order differential operator; in addition, the kernel associated is non-local.

The applications of AB fractional-order differential operator pertaining to signal and image processing areas are presented as follows:

Based on AB Fractional-Order Operator

Solìs-Pèrez *et al.* [37] put forward a novel approch based on fractional Gaussian kernel and fractional Gaussian Hessian matrix based on AB fractional-order differential operator for the detection of blood vessels in retinal images. The fractional orders of the fractional Gaussian mask for retinal image denoising and fractional Gaussian Hessian matrix have been optimized by utilizing Cuckoo search algorithm. Different databases have been used to demonstrate the effectiveness of the proposed method.

Ghanbari *et al.* [91] developed several image denoising algorithms based on AB fractional-order integral operator. The AB-based fractional-order masks have been applied on images affected by Gaussian noise and the performance of the masks have been tested and compared against various standard and fractional smoothing filters.

Furthermore, fundamental differences among power-law, exponential-law, and Mittag-Leffler kernels have been presented in [43] along with their potential applications in various fields. From the viewpoint of signal processing and filtering applications, a detailed study has been presented in [92], where it has been pointed out that the transfer function associated to AB fractional differential operator is yet to be explored and analyzed in-depth.

Therefore, the work done in this thesis aims to extend the domain of fractional-order signal processing by developing novel fractional-order digital filter based upon recently introduced AB fractional-differential operator, which could be beneficial for various signal and image processing applications. Furthermore, the applicability of the proposed filter is tested for one-dimensional (1D) and two-dimensional (2D) applications.

2.3 Preliminaries of Adaptive Fractional-Order Calculus

Although FOC provides one extra degree of freedom, finding an optimum fractional-order for various applications is still a challenging task. Therefore, the study of adaptive FOC has become a burgeoning subject under consideration among various fields of the research community from the past few years.

2.3.1 Adaptive Fractional-Order Image Processing

From the perspective of image processing applications, it is known that the characteristics of an image varies locally. Therefore, if the quality of a digital image gets inevitably degraded during the process of acquisition and transmission, it becomes difficult to attain better image denoising effect by utilizing a fixed fractional-order to process noisy pixels in distinct regions.

This is due to the fact that if higher fractional-order is chosen, the texture details are lost. On the other hand, if lower fractional-order is chosen, denoising performance is degraded [13]. Therefore, few attempts have been made by some well-renowed researchers to incorporate adaptive fractional-order for image processing applications, the details of which are presented here as follows:

Li *et al.* [93] made an attempt to design a global adaptive fractional integral algorithm (GAFIA) to deal with various types of noises adaptively. In this, the optimal integral order is selected based upon local average gradient of the image. Furthermore, an image denoising and image enhancement algorithm based on adaptive fractional calculus of small probability strategy (AFC-SPS) is proposed to deal with salt and pepper noise of low density (i.e., 1% to 5%). From the experimental studies conducted, it has been shown that both methods provide good image denoising effect. Apart from that, it has been demonstrated that the AFC-SPS algorithm provides better effect in preserving the texture and enhancing the edge details of the image as well.

Yu *et al.* [94] proposed an adaptive algorithm based on entropy and gradient features (EN-AFC) for the removal of salt and pepper (SP) noise in images. Initially, the effects of noise on the entropy and gradient of the image are analyzed. Then, an adaptive FOC based image denoising algorithm is proposed to deal with the low density SP noise. The experimental re-

sults demonstrated that the proposed method not only preseves the smooth regions, textures and edge details of the image, but also provides a better visualisation.

Shukla *et al.* [95] designed an adaptive image denoising algorithm based on fractional K-operator. In this, a noise detection method has been designed which selects the noisy pixels and then, updates them by utilizing adaptive fractional integral mask. The rest of the pixels are updated by utilizing adaptive fractional differential mask so as to recover the distortions incurred by noise. Experimental results validated that the proposed method outperformed previously reported GAFIA [93], AFC-SPS [93], and EN-AFC [94] methods when dealing with SP noise of various densities.

Lin *et al.* [96] proposed an adaptive algorithm for image denoising based on AB fractional-order integral operator. An adaptive fractional-order function has been constructed based on various features such as global entropy, local entropy, edge intensity, local variance, and to name a few. The designed fractional-order masks have been applied on images affected by Gaussian noise and the results reported verified the effectiveness of the denoising method.

Furthermore, a significant research has also been carried out to develop adaptive fractional-order differential based image enhancement methods. The details corresponding to applications of adaptive FOC for image enhancement applications can be refered from [97, 98, 99].

Due to the widespread popularity of adaptive FOC in image processing applications, the research work carried out in this thesis aims at developing generalized fractional-order adaptive masks for image processing applications.

2.4 Motivation

The comprehensive study conducted elucidates the fact that various definitions of fractional-order operators have found numerous applications in several areas of science and engineering. However, the profound applications of recently introduced fractional-order operators are yet to be explored in detail for signal and image processing applications.

Considering power-law kernel based fractional-order operators, the GL-based methods have been extensively investigated by the research community to design FODD [54, 76, 78, 79, 80]. Following this, a genuine curiosity arises to delve into the design of FODD based on RL fractional-order differential operator. Therefore, the initial work in this thesis aims at

establishing a generalized framework for the design of RL-based FODD (RL-FODD) and compares its performance with well-established GL-FODD designs, based upon their ability to yield ideal frequency response of FODD. Furthermore, considering Mittag-Leffler kernel based fractional-order operator, a detailed study has been presented in [92], where it has been pointed out that the transfer function associated to AB fractional differential operator is yet to be explored and analyzed in-depth. Therefore, the domain of fractional-order signal processing is extended by developing novel fractional-order digital filters based upon recently introduced fractional-differential operators, which could be beneficial for various signal processing applications.

On the other hand, from the viewpoint of image processing applications, the existence of various fractional-order masks for each definition separately, as presented in [13, 90, 91], manifested the necessity to develop a generalized framework for fractional-order masks to avoid the need to numerically approximate each fractional-order operator separately for the construction of masks. Therefore, further research work aims at establishing framework for the design of generalized fractional-order masks for image processing applications.

Additionally, the impulse noise is one of the most prominent types of noises which greatly impact the performance of various image processing techniques. One such type of impulse noise is salt and pepper (SP) noise. Several traditional methods exist in the literature that have been applied to deal with SP noise effectively.

Although these methods can remove noise efficiently and possess several advantages, the preservation of edge and texture details of an image still remains a challenging issue. On the other hand, most of the methods in the literature assume all pixels with intensity values 0 and 255 in natural images as noisy [100, 101, 102, 103]. Therefore, these methods misclassify non-noisy pixels (of values 0 or 255) as noisy pixels. As a result, the performance of these methods get degraded when images containing pixels with intensity values 0 or 255 are taken into account. Furthermore, a limited research has been carried out to deal with SP noise in synthetic images (i.e., which contains all the pixels with intensity values 0 and 255) [104, 105].

Therefore, this aroused interest to develop a robust noise detection method (NDM) for the efficient detection of noisy pixels in natural as well as synthetic images. Hence, a novel NDM has been proposed which exhibits a profound advantange in avoiding the misclassification of

original pixels as noisy pixels when the original image itself has some pixels with intensity values 0 or 255.

On the other hand, most of the traditional image denoising methods based on fractional-order masks employ a fixed fractional-order to deal with the noise present in the images. However, the characteristics of an image varies locally. Therefore, Li *et al.* [93], Yu *et al.* [94], Shukla *et al.* [95], Lin *et al.* [96] made some attempts to propose adaptive algorithms for image denoising. Although these methods can remove noise adaptively, the methods presented in [93] and [94] are applicable only when the SP noise density is small i.e., 0.01-0.05. Furthermore, the methods presented in [93, 94, 95] are based upon conventional fractional-order operators. On the other hand, the method proposed in [96] is applicable only for images affected by Gaussian noise.

Therefore, motivated by the research presented in [93, 94, 95], a generalized fractional integral and fractional differential based adaptive algorithm (GFIFD-AA) has been proposed to deal with SP noise effectively. The proposed method not only extends the range of SP noise density considered in [93] and [94] to 10%-95%, but also incorporates recently introduced definitions of fractional-order operators in generalized framework in contrast to the method presented in [95].

Therefore, based on the above-mentioned concepts, the gaps encountered in the study are outlined and the objectives of research work are put forward.

2.4.1 Research Gaps

Based on the survey conducted, some of the research gaps that have been identified are as follows:

- Most of the literature available provide emphasis on particular definitions of fractional-order differentials for the design of FODD. Therefore, there is a scope of research for the design of fractional-order digital filters based upon singular and non-singular fractional-order kernels for signal and image processing applications.

- Several fractional-order masks corresponding to various fractional-order derivatives exist in the literature but in isolation i.e., for each definition of fractional-order derivative, fractional-order masks have been constructed separately. Therefore, there is a

need to develop a generalized framework for fractional-order masks that can avoid the need to numerically approximate each fractional-order operator separately for the construction of masks.

- Although most of the traditional methods that exist in the literature for the detection and removal of SP noise in images possess several advantages, the preservation of edge and texture details of an image still remains a challenging issue. Therefore, there is a scope of research to establish an image denoising method that can preserve the texture and edge details of the image while denoising the image simultaneously.

- On the other hand, most of the methods in the literature assume all pixels with intensity values 0 and 255 as noisy. Therefore, these methods misclassify non-noisy pixels (of values 0 and 255) as noisy pixels. As a result, the performance of these methods get degraded when images containing pixels with intensity values 0 or 255 are taken into account. Therefore, there is a need to develop a robust noise detection method for the efficient detection of noisy pixels in the image.

- Most of the traditional image denoising methods based on fractional-order masks employ a fixed fractional-order to deal with the noise present in the images. However, the characteristics of an image varies locally. Therefore, few attempts have been made in the literature to deal with the noise adaptively. Although these methods can remove noise adaptively, most of the existing methods are applicable only when the noise density is small i.e., 0.01-0.05. Furthermore, the existing methods based on fractional-order masks take conventional fractional-order operators into consideration. Therefore, there is a scope to extend the range of noisy density as well as definitions of fractional-order operators considered so far for efficient denoising of images.

2.4.2 Objectives of Research

Based on the basic principles, literature survey conducted and the insight developed regarding FOC, the following objectives have been put forward:

1. To study the various design methods for implementing digital filter in the fractional domain.

2. To develop a mathematical model of the proposed fractional-order digital filter.

3. To implement the designed fractional-order filter for one-dimensional (1D) and two-dimensional (2D) applications.

4. To validate the proposed fractional-order filter design with the state-of-the-art methods.

2.5 Research Methodolody

In order to attain the above-mentioned objectives of research, the key component is to conduct a thorough survey of prevalent literature in order to garner the knowledge that assists in understanding the fundamental concepts of topic under investigation. The flowchart of the work carried out in this thesis is presented in Figure 2.1. Initially, the comprehensive study of various definitions of fractional-order operators, some special mathematical functions utilized in FOC and interpolation methods is conducted. Furthermore, the fundamental concepts of FODD and the methods utilized in literature for their design are studied to gain significant perspective about their implementation for signal and image processing applications. This in-depth analysis led to the formulation of certain solutions in order to satisfy the research gaps encountered during the course of study.

The research work carried out in this thesis emphasizes on the design of singular power-law kernel and non-singular Mittag-Leffler kernel based fractional-order digital filters for signal filtering applications. The closed-form analytical formulations are proposed to obtain FIR filter coefficients of the designed filters. Then, the applicability of the proposed filters are validated for various signal filtering applications. Then, a generalized fractional integral and fractional differential based adaptive algorithm (GFIFD-AA) is put forward for the application of image denoising. The proposed GFIFD-AA is further segregated into two parts where in the first part, generalized frameworks for the design of fractional integral and fractional differential based adaptive masks are constructed. Then, a robust noise detection method based on probability strategy is put forward to detect SP noise in images. The efficacy of the proposed methods is evaluated both qualitatively and quantitatively by taking standard images from various datasets into account. Furthermore, the effectiveness of the proposed method is compared against well-established state-of-the-art methods in the literature. All

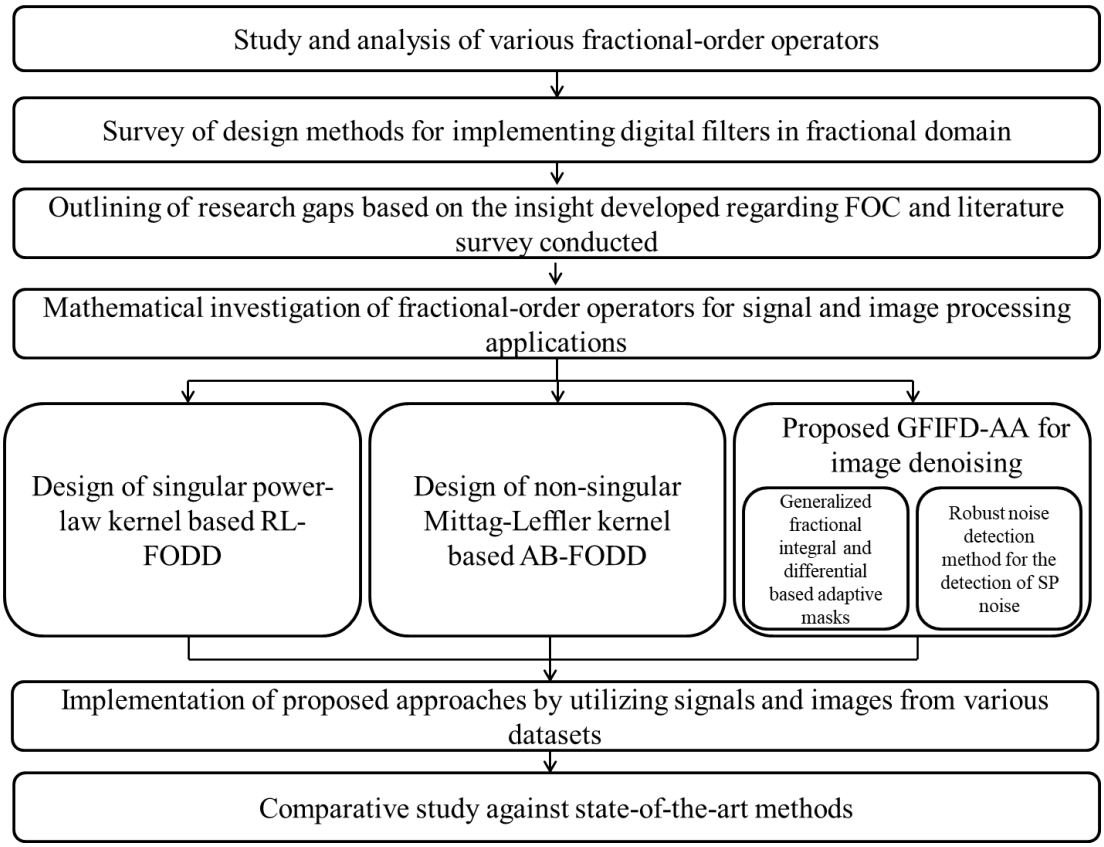


Figure 2.1: Flowchart of the research work carried out in thesis

the simulations are performed by utilizing software version 2020b of MATLAB on a personal computer with Intel Core i5 and 1.60 GHz CPU processor with 8 GB RAM.

Chapter 3

Riemann-Liouville Fractional Differential-Based Design of FODD

FRACTIONAL-ORDER DIFFERENTIAL operators, their numerical implementations, and the design of fractional-order digital differentiators/filters is a burgeoning subject under consideration among various fields of research from the past few decades [8, 12, 106]. The mathematical tool of fractional-order calculus (FOC), which is, in fact, the generalization of integer-order calculus, is concerned with differentiation and integration of non-integer orders.

Under the research area of FOC, Grünwald−Letnikov (GL) and Riemann−Liouville (RL) are the most widely utilized definitions of fractional differential operators [8]. The GL fractional differential operator gained a significant attention in the research community for designing fractional-order digital differentiators (FODD). Following this, a genuine curiosity arises to delve into the design of FODD pertaining to RL fractional differential operator. Therefore, this chapter establishes a closed-form analytical formulation for the design of RL fractional differential based FODD (*herein, referred to as RL-FODD*) and further establishes the efficacy of the *proposed* design through simulation results.

3.1 Concept of FODD: Signal Processing Perspective

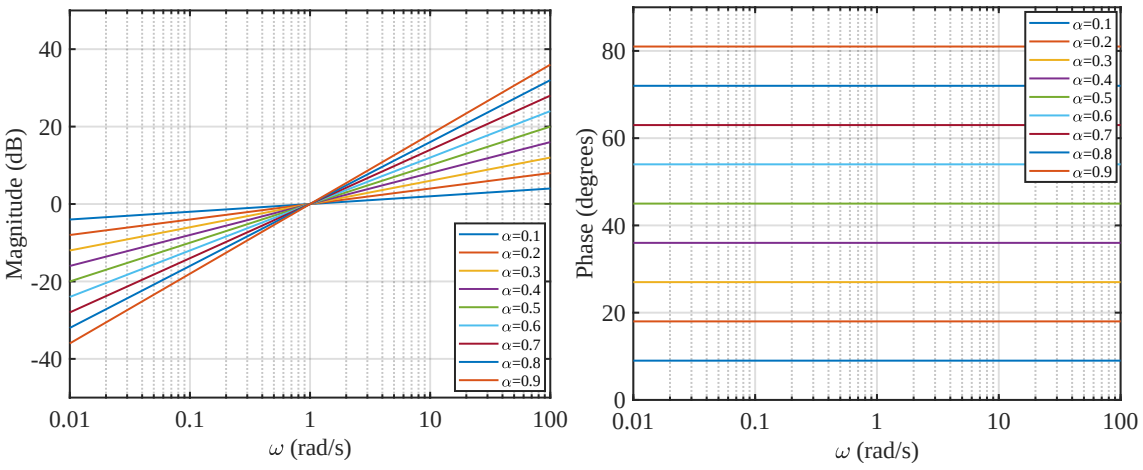
From the perspective of signal processing literature [2], the generalization from integer-orders to non-integer order has found abundant applications in areas such as image edge detection [21], in synthesis of multifractional Gaussian noise [23], in digital image sharpen-

ing [13] and in dielectric spectroscopy [26], to name a few. Apart from these encouraging applications that have rendered FOC an interesting research topic, the study of FODDs has gained momentum as it provides enhanced flexibility in desigining digital differentiating systems [56]. Furthermore, it has become an important mathematical tool that can provide more specific details in comparison to integer-order counterparts [13]. Therefore, the design of FODDs has become the main focus of interest among signal processing research community from the past few decades [4].

The design problem of FODD involves designing a digital filter such that its frequency response matches the ideal frequency response of FODD, as closely as possible. As it is already established, the ideal frequency response of power-law kernel based FODD is given by [2, 107]

$$H_d(\omega) = (j\omega)^{\alpha} = \omega^{\alpha} e^{\frac{j\alpha\pi}{2}} \tag{3.1}$$

where $j = \sqrt{-1}$ and α is a real number. For integer-order digital differentiator, $\alpha = 1$, i.e., integer-order digital differentiator is a special case of FODD. The presence of additional tuning knob in FODD facilitates in altering the characteristics of integer-order digital differentiator and hence, provides one more degree of freedom for shaping the frequency response of the system [4]. The Bode diagrams depicting the ideal magnitude and phase response of $H_d(\omega)$, presented in (3.1) are illustrated in Figure 3.1.



(a) **(b)**

Figure 3.1: Bode diagram representing ideal (a) magnitude and (b) phase responses of FODD.

As mentioned in Section 2.2.2, many researchers have paid their attention preferably to GL

fractional-order differential operator for the design of FODD [12, 53, 54]. However, it has been pointed out that the discrete approximations derived from GL fractional differential operator present some limitations [108] and the order of accuracy of such approaches is never higher than one [83, 106, 109].

Although there exists a close relationship between truncated GL and RL fractional-order differential operators, the choice of different nodes (sampling instants) and different coefficients may lead to different accuracies [109]. From the perspective of image processing applications, a class of fractional-order differential masks has been proposed in [13] based upon GL and RL fractional-order differential and integral operators. A rigorous comparison among the proposed fractional differential masks has also been made based on their capability to enhance texture details in the images.

However, in the signal processing literature, the general approach to design FODD has only been put forward by considering GL fractional-order differential operator. Therefore, there arises a genuine desire to explore the design of FODD based on yet another most popular definition of fractional-order derivatives, i.e., RL fractional-order differential operator. Unlike GL fractional-order differential operator, which due to its discrete formulation, can immediately yield coefficients of FODD, RL fractional-order differential operator needs to be numerically approximated to obtain filter coefficients of the FODD. Therefore, the main intent of this chapter is to devise an analytical framework for RL-FODD and compare it with conventional well-established designs based on GL fractional-order differential operator.

3.2 Design of RL-based FODD (RL-FODD) using Finite Difference Method

Before delving into the design details of RL-FODD, the RL fractional-order differential operator, presented in (2.5) is first numerically approximated. The α^{th}-order RL-based fractional-order differential of the signal $s(t)$ can also be written as [8, 13]

$$\begin{aligned} D^{\alpha}_{RL}p(t) &= \frac{d^{\alpha}}{[d(t-c)]^{\alpha}}p(t)\bigg|_{RL} = \frac{d^{n}}{dt^{n}}\frac{d^{\alpha-n}}{d(t-c)^{\alpha-n}}p(t) \\ &= \sum_{k=0}^{n-1}\frac{(t-c)^{k-\alpha}p^{(k)}(c)}{\Gamma(k-\alpha+1)} + \frac{1}{\Gamma(n-\alpha)}\int_{c}^{t}\frac{p^{(n)}(\zeta)}{(t-\zeta)^{\alpha-n+1}}d\zeta \end{aligned} \tag{3.2}$$

where D^{α}_{RL} represents RL-based fractional differential operator and $n-1 < \alpha < n$ (where $n = \lceil \alpha \rceil$ and $\lceil . \rceil$ denotes the *ceil* operator). The continuous integration in the definition of RL fractional-order differential is changed into discrete sum of products and the differential inside the integral is approximated by using difference equation of the first-order derivative. Then, it is analytically formulated to yield the finite impulse response (FIR) filter coefficients of the *proposed* RL-FODD.

3.2.1 Mathematical Investigation of RL-FODD

The RL fractional-order differential operator, presented in (3.2), for the signal $p(t)$, where t represents the time variable can be rewritten as

$$D^{\alpha}_{RL}p(t) = \sum_{k=0}^{n-1} \frac{(t-c)^{k-\alpha} p^{(k)}(c)}{\Gamma(k-\alpha+1)} + \frac{1}{\Gamma(n-\alpha)} \int_{c}^{t} \frac{p^{(n)}(\zeta)}{(t-\zeta)^{\alpha-n+1}} d\zeta \tag{3.3}$$

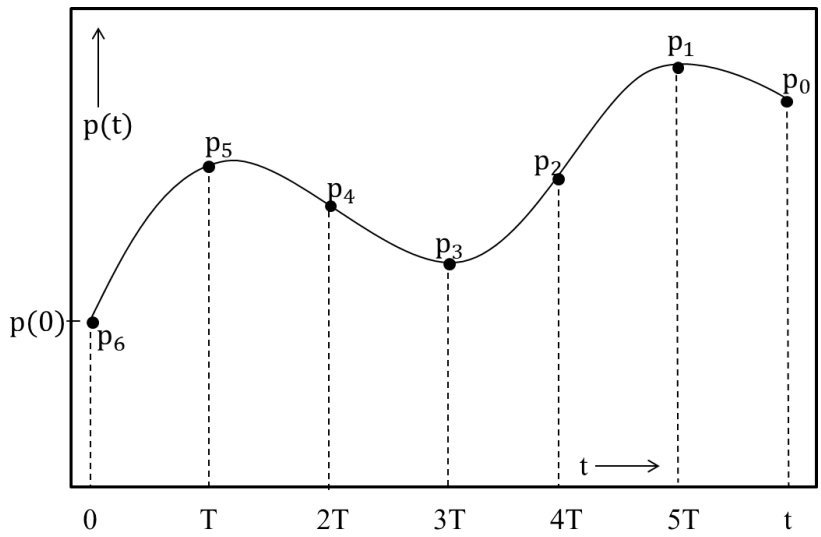


Figure 3.2: Nomenclature utilized for partitioning interval $[0, t]$ into subintervals $[p_k, p_{k+1}]$.

In general, let the lower limit $c = 0$ and the value of the signal $p(t)$ is known at $L+1$ evenly spaced points in the range $[0, t]$ with sampling interval T. Thus, the $L+1$ causal samples of

the signal can be written as

$$\begin{aligned} p_L &\equiv p(0) \\ p_{L-1} &\equiv p(T) \\ &\vdots \\ p_k &\equiv p(t-kT) \\ &\vdots \\ p_0 &\equiv p(t) \end{aligned}$$

To get a more intuitive idea about this nomenclature, Figure 3.2 is presented by taking $L = 6$ as an example.

For $c = 0$ and $n = 1$, the RL fractional-order differential operator for $0 \leq \alpha < 1$, presented in (3.3) can be written as [8]

$$\begin{aligned} D^{\alpha}_{RL}p(t) &= \frac{t^{-\alpha}p(0)}{\Gamma(1-\alpha)} + \frac{1}{\Gamma(1-\alpha)}\int_0^t \frac{p^{(1)}(\zeta)}{(t-\zeta)^{\alpha}}d\zeta \\ &= \frac{t^{-\alpha}p(0)}{\Gamma(1-\alpha)} + \frac{1}{\Gamma(1-\alpha)}\int_0^t \frac{p^{(1)}(t-\zeta)}{\zeta^{\alpha}}d\zeta \end{aligned} \tag{3.4}$$

The RL fractional-order differential operator, presented in (3.4) is then numerically approximated by changing the continuous integration into discrete sum of products (i.e., the interval $[0, t]$ is uniformly partitioned into L sub-intervals $[p_k, p_{k+1}](k = 0, 1, ..., L-1)$ and the length of each sub-interval is $h = (t-0)/L$) as given by

$$D^{\alpha}_{RL}p(t) = \frac{1}{\Gamma(1-\alpha)}\left[\frac{p(0)}{t^{\alpha}} + \sum_{k=0}^{L-1}\int_{kh}^{(k+1)h}\left(\frac{d}{d\zeta}p(t-\zeta)\right)\frac{d\zeta}{\zeta^{\alpha}}\right] \tag{3.5}$$

The first-order derivative inside the integral in (3.5) can then be written by utilizing backward difference equation of the first-order derivative (with reference to the nomenclature utilized for $[p_k, p_{k+1}]$, as presented in Figure 3.2) as

$$\begin{aligned} \int_{kh}^{(k+1)h}\left(\frac{d}{d\zeta}p(t-\zeta)\right)\frac{d\zeta}{\zeta^{\alpha}} &\cong \frac{p(t-kh)-p(t-h-kh)}{h}\int_{kh}^{(k+1)h}\frac{d\zeta}{\zeta^{\alpha}} \\ &= \frac{p_k - p_{k+1}}{h}\left[\frac{((k+1)h)^{(1-\alpha)} - (kh)^{(1-\alpha)}}{(1-\alpha)}\right] \\ &= \frac{p_k - p_{k+1}}{h}\frac{h^{(1-\alpha)}}{1-\alpha}[(k+1)^{1-\alpha} - k^{1-\alpha}] \\ &= \frac{h^{-\alpha}}{1-\alpha}(p_k - p_{k+1})[(k+1)^{1-\alpha} - k^{1-\alpha}] \end{aligned} \tag{3.6}$$

Substituting (3.6) into (3.5), implies

$$D^{\alpha}_{RL}p(t) \cong \frac{h^{-\alpha}}{\Gamma(2-\alpha)}\left[\frac{1-\alpha}{L^{\alpha}}p_L + \sum_{k=0}^{L-1}(p_k - p_{k+1})\left[(k+1)^{1-\alpha} - k^{1-\alpha}\right]\right] \tag{3.7}$$

Now, the $L+1$ approximate backward differences of RL fractional-order differential operator can be expressed as

$$\begin{aligned} D^{\alpha}_{RL}p(t) \cong \frac{h^{-\alpha}}{\Gamma(2-\alpha)}p(t) + \frac{h^{-\alpha}}{\Gamma(2-\alpha)}\Bigg\{\sum_{k=1}^{L-1}\left[(k+1)^{1-\alpha} - 2k^{1-\alpha} + (k-1)^{1-\alpha}\right]p(t-kh) \\ + \left[(1-\alpha)L^{-\alpha} - (L-1)^{1-\alpha} + (L-1)^{1-\alpha}\right]p(t-Lh)\Bigg\} \end{aligned} \tag{3.8}$$

Rewriting the above equation in compact form yields

$$D^{\alpha}_{RL}p(t) \cong \sum_{k=0}^{L}\frac{a(k)}{h^{\alpha}}p(t-kh) \tag{3.9}$$

where, $a(k)$ represents the *proposed* RL-based FIR filter coefficients of FODD, which are defined as

$$a(k) = \begin{cases} \frac{1}{\Gamma(2-\alpha)} & ; k = 0 \\ \frac{1}{\Gamma(2-\alpha)}[(k+1)^{1-\alpha} - 2k^{1-\alpha} + (k-1)^{1-\alpha}] & ; 1 \le k \le L-1 \\ \frac{1}{\Gamma(2-\alpha)}[(1-\alpha)L^{-\alpha} - L^{1-\alpha} + (L-1)^{1-\alpha}] & ; k = L \end{cases} \tag{3.10}$$

Thus, (3.9) represents the *proposed* framework for RL-based FIR FODD, where FIR filter coefficients can be computed from (3.10). These coefficients are further utilized to compute the transfer function of the *proposed* RL-FODD. By considering $t = n - D$ in (3.9), where D represents the integer delay of the system, the discrete-time RL fractional-order differential can be defined by

$$D^{\alpha}_{RL}p[n-D] = \sum_{k=0}^{L}\frac{a[k]}{h^{\alpha}}p[n-D-kh] \tag{3.11}$$

Now, computing the z transform on both sides of (3.11) yields

$$Y(z) = \left(\sum_{k=0}^{L}\frac{a[k]}{h^{\alpha}}z^{-(D+kh)}\right)P(z) \tag{3.12}$$

where $Y(z)$ and $P(z)$ represent the z-transform of $D^{\alpha}_{RL}p[n-D]$ and $p[n]$, respectively. The above equation can further be expressed in a compact form as

$$Y(z) = H(z)P(z) \tag{3.13}$$

where

$$H(z) = \sum_{k=0}^{L} \frac{a[k]}{h^{\alpha}} z^{-(D+kh)} \tag{3.14}$$

Thus, $H(z)$ represents the transfer function of the *proposed* RL-FODD for fractional orders $0 \leq \alpha < 1$, whose frequency response approximates the ideal frequency response $H_d(\omega)$ of FODD. The frequency response of the *proposed* RL-FODD can then be computed by utilizing $z = e^{j\omega}$, as is represented by $H(e^{j\omega})$.

On the other hand, the discrete-time GL fractional-order differential [54] can be defined as

$$D^{\alpha}_{GL}p[n-D] = \sum_{k=0}^{L} \frac{b[k]}{h^{\alpha}} p[n-D-kh] \tag{3.15}$$

where $b[k]$ represents the FIR filter coefficients of GL fractional-order differential based FODD (*herein, referred to as GL-FODD*), given as

$$b[k] = (-1)^k \frac{\Gamma(\alpha+1)}{\Gamma(k+1)\Gamma(\alpha-k+1)} \tag{3.16}$$

In the similar manner, the transfer function of GL-FODD can be obtained and is represented by

$$H(z) = \sum_{k=0}^{L} \frac{b[k]}{h^{\alpha}} z^{-(D+kh)} \tag{3.17}$$

Therefore, it can be said that there are two ways to obtain FIR filter coefficients of FODD in order to obtain fractional-order derivatives of the given signal. One based on GL formula and the other based on numerical approximation of RL fractional-order differential, as presented in (3.10). The *proposed* RL-FODD and the conventional GL-FODD differ only in the terms of filter coefficients. The coefficients of GL-FODD, as presented in (3.16), require the arithmetic computation of factorial/Gamma functions, i.e. $\Gamma(k+1)$, where $k = 0, 1, \ldots, L$, which imposes a computational limitation for larger values of L. This is because the computation of factorial/Gamma function causes the overflow and hence, limits the number of coefficients

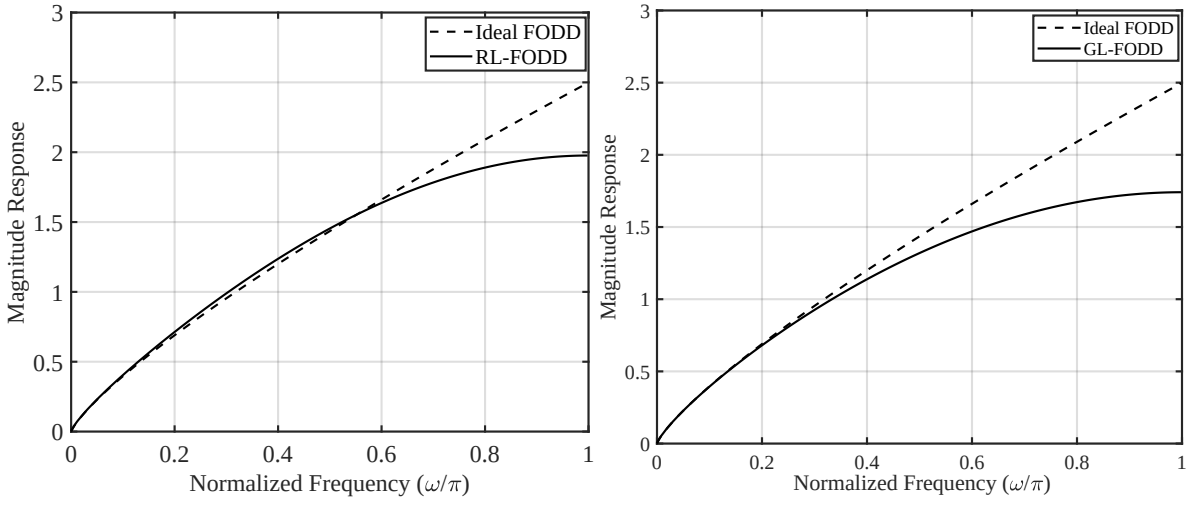
$L+1$ to 170 [110]. Therefore, recursive relation needs to be applied for computing filter coefficients of GL-based FIR FODD. This recursive approach is suitable for fixed value of α. However, in some problems where α must be found (for example, in system identification), various values of α needs to be considered and hence, recurrence relationships are not very suitable [108, 110].

On the other hand, the *proposed* RL-based filter coefficients, as presented in (3.10), do not impose such kind of limitations for larger values of L, which makes the *proposed* RL-based filter coefficients simple and easy to compute.

3.2.2 RL-FODD and GL-FODD − A Comparative Analysis

After establishing the closed-form analytical formulation for the design of RL-FODD, the next aim is to validate the efficiency of the *proposed* RL-FODD against that of well-established design of GL-FODD.

To illustrate the efficiency of the *proposed* RL-FODD design, an example with design parameters $h = T = 1, L = 100$ and $\alpha = 0.8$ is presented. Furthermore, the design of the *proposed* RL-FODD is compared against the design of GL-FODD by utilizing the same design parameters. Figure 3.3 illustrates the magnitude response approximations with respect to normalized frequencies (solid lines) attained by the *proposed* RL-FODD and the conventional design of GL-FODD.



(a) **(b)**

Figure 3.3: Magnitude responses (solid lines) of the (a) *proposed* RL-FODD and (b) GL-FODD. The dashed lines represent the ideal magnitude response of FODD.

From the result presented in Figure 3.3a, it can be seen that the *proposed* RL-FODD approximates the ideal frequency response of FODD well in the low- and mid-frequency range. However, in the high-frequency region, i.e., near $\omega = \pi$, the approximation error is quite large. On the other hand, Figure 3.3b represents the corresponding magnitude response approximation of the conventional GL-FODD. From Figure 3.3b, it can be observed that the magnitude response approximation of GL-FODD imitates the ideal frequency response in the low-frequency range only. However, in the mid- and high-frequency regions, the approximation error is relatively large.

Comparing the design approximations illustrated in Figure 3.3a and Figure 3.3b, it can be seen that the *proposed* RL-FODD provides better approximation of ideal FODD in comparison to that of GL-FODD in the mid- and high-frequency range. This better approximation attained by the *proposed* RL-FODD can further be evaluated quantitatively by considering the performance metric known as mean square error (MSE), which is defined as

$$E = \int_0^{\pi} \left(|H_d(\omega)| - \left| H(e^{j\omega}) \right| \right)^2 d\omega \tag{3.18}$$

where $|H(e^{j\omega})|$ represents the magnitude response approximation attained by the design method and $|H_d(\omega)|$ represents the ideal magnitude response of FODD. The smaller MSE indicates better performance of the design method.

For $\alpha = 0.8$, the MSE between the *proposed* RL-FODD and ideal FODD is $E = 0.0286$, which is comparatively less than the MSE value of $E = 0.0897$ attained by the conventional GL-FODD over the entire frequency band of interest, i.e., $[0, \pi]$. Therefore, it can be said that the *proposed* RL-FODD has shown a percentage improvement of 68.11% for fractional-order $\alpha = 0.8$ over that of conventional GL-FODD.

Furthermore, the comparison between the *proposed* RL-FODD and conventional GL-FODD is illustrated in Figure 3.4, where MSE values are plotted for different values of fractional order α, which varies from 0 to 1. From the results, it can be visualized that the performance of *proposed* RL-FODD is better than that of the conventional GL-FODD for fractional orders $\alpha \geq 0.3$. However, for fractional orders $\alpha < 0.3$, the difference between the MSE values for both the methods is almost negligible.

For better clarity, the MSE values for both the designs are reported in Table 3.1 for various

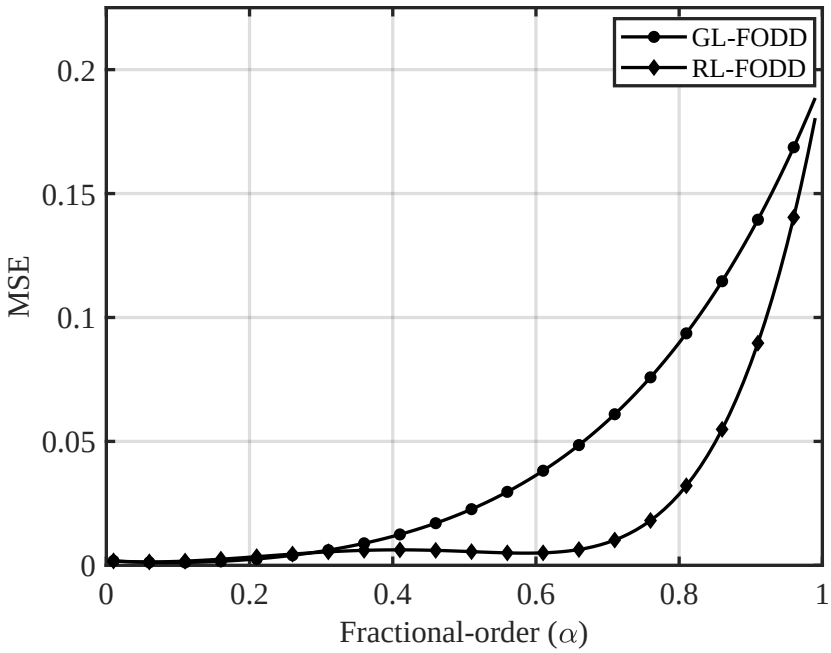


Figure 3.4: Quantitative comparison between the magnitude response approximations attained by the *proposed* RL-FODD and conventional GL-FODD at various fractional orders.

fractional orders. From the values tabulated in Table 3.1, it can be seen that the proposed RL-FODD outperforms the conventional GL-FODD at fractional orders $\alpha \geq 0.3$. However, for $\alpha < 0.3$, the variation between the *proposed* method and the conventional method is not more than 0.001. *This clearly indicates the effectiveness of the proposed RL-FODD over that of conventional well-established GL-FODD.*

Table 3.1: Quantitative comparison between GL-FODD and *proposed* RL-FODD in terms of MSE at various fractional orders

Fractional-order α	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
GL-FODD	**0.0011**	**0.0023**	0.0056	0.0116	0.0214	0.0363	0.0582	0.0897	0.1341
RL-FODD	0.0015	0.0032	**0.0052**	**0.0062**	**0.0056**	**0.0049**	**0.0091**	**0.0286**	**0.0815**

From the analytical and simulation study conducted, it can be said that the *proposed* RL-FODD provides a better approximation of the ideal frequency response of FODD in comparison to the well-established GL-FODD design presented in the literature. Nevertheless, the approximation error is comparatively high in the high-frequency region for both cases. Hence, it becomes a crucial task to lower the approximation error in the high-frequency region.

This large approximation error can however, be reduced by decreasing the step-size h in (3.5) and (3.15), i.e., by choosing $h \rightarrow 0$. The accuracy of the *proposed* design will increase as h decreases.

As an illustration, the *proposed* RL-FODD is utilized to estimate fractional-order derivatives of the given signal by taking various values of h into account. The intent of the FODD is to estimate fractional-order derivatives of the given signal, i.e., to compute the α^{th}-order derivative of the discrete-time signal $p[n] = p(t)|_{t=t_n}$ such that the output samples $y[n] = y(t)|_{t=t_n}$ are as close to the output $D^{\alpha}p(t)$ at the sampling intervals $t_n = nT$ as possible. Figure 3.5 represents the block diagram to compute fractional-order derivatives of the given signal by utilizing the *proposed* RL-FODD.

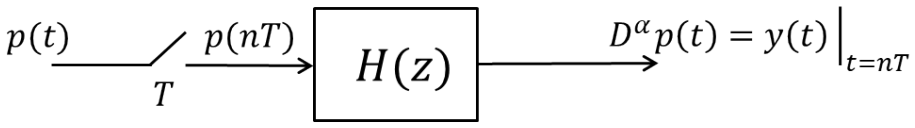


Figure 3.5: Computation of fractional-order derivatives using FODD.

Now, the problem statement is to determine the fractional-order derivatives of various signals (i.e. $D^{\alpha}p(n)$) by choosing various values of h and then, to determine the efficacy of the *proposed* RL-FODD design against that of the conventional design of GL-FODD.

Two types of signal, i.e., a signal comprising of sum of two sinusoids and a power function are considered, which are defined as:

$$\begin{aligned} p_1(t) &= -(1/2)(t^{3\alpha}) + (1/5)t^{4\alpha} \\ p_2(t) &= 2sin(0.1\pi t) + 2sin(0.2\pi t) \end{aligned} \tag{3.19}$$

The exact fractional-order derivatives of these signals are given by [8, 111]

$$\begin{aligned} D^{\alpha}p_1(t) &= -(1/2)\frac{\Gamma(3\alpha+1)(t^{2\alpha})}{\Gamma(2\alpha+1)} + (1/5)\frac{\Gamma(4\alpha+1)(t^{3\alpha})}{\Gamma(3\alpha+1)} \\ D^{\alpha}p_2(t) &= 2(0.1\pi)^{\alpha}sin(0.1\pi t + \alpha\pi/2) + 2(0.2\pi)^{\alpha}sin(0.2\pi t + \alpha\pi/2) \end{aligned} \tag{3.20}$$

Table 3.2 presents the RMSE and MAE values between the exact fractional-order derivatives of the signals and estimated fractional-order derivatives attained by utilizing GL-FODD (denoted by D^{α}_{GL}) and the *proposed* RL-FODD (denoted by D^{α}_{RL}), at $\alpha = 0.8$. From the results tabulated in Table 3.2, it can be seen that as the value of h decreases, i.e., as $h \to 0$, the accuracy of the methods increases.

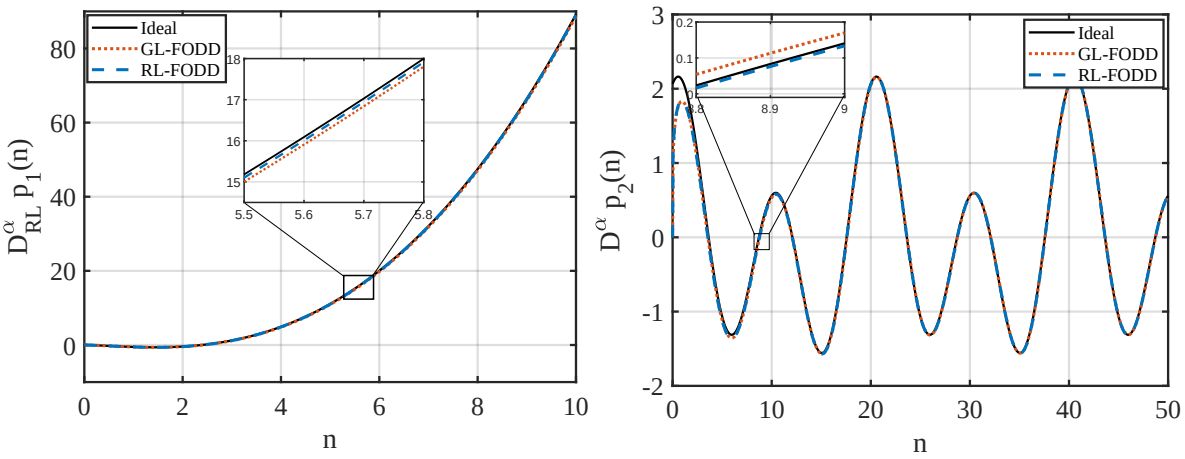
Table 3.2: Quantitative comparison between GL-FODD and the *proposed* RL-FODD at $\alpha = 0.8$ and varying values of h

	$p_1(n)$				$p_2(n)$			
	RMSE		MAE		RMSE		MAE	
h	D^{α}_{GL}	D^{α}_{RL}	D^{α}_{GL}	D^{α}_{RL}	D^{α}_{GL}	D^{α}_{RL}	D^{α}_{GL}	D^{α}_{RL}
1	56.6657	**29.2233**	45.0018	**23.9021**	0.3791	**0.3524**	0.2553	**0.2184**
0.5	28.3541	**12.7941**	22.6031	**10.5121**	0.2461	**0.2284**	0.1340	**0.1036**
0.25	14.1823	**5.5879**	11.3275	**4.6023**	0.1734	**0.1634**	0.0743	**0.0574**
0.125	7.0925	**2.4371**	5.6703	**2.0098**	0.1332	**0.1277**	0.0484	**0.0387**
0.0625	3.5466	**1.0620**	2.8368	**0.8764**	0.1110	**0.1079**	0.0361	**0.0313**
0.03125	1.7734	**0.4626**	1.4188	**0.3819**	0.0991	**0.0973**	0.0308	**0.0285**
0.015625	0.8867	**0.2014**	0.7095	**0.1663**	0.0929	**0.0918**	0.0284	**0.0277**

Furthermore, comparing the results presented in Table 3.2, it is evident that the *proposed* RL-FODD yields smaller values of RMSE and MAE, and hence, validates the superior performance of the *proposed* RL-FODD than that of the conventional GL-FODD in computing fractional-order derivatives of the given signals.

As an illustration, Figure 3.6a and 3.6b represent the ideal fractional-order derivatives of the signals $p_1(n)$ and $p_2(n)$ and their corresponding approximations obtained by utilizing conventional GL-FODD and the *proposed* RL-FODD for $\alpha = 0.8$ and $h = 0.05$. From the results depicted in Figure 3.6, it can be seen that the conventional GL-FODD and the *proposed* RL-FODD imitate the ideal fractional-order derivative of the signal effectively. However, the approximation attained by the *proposed* RL-FODD is better than that of the conventional GL-FODD, as can be visualized from the highlighted portion presented in Figure 3.6a. Similar pattern can be observed for the signal $p_2(n)$, as presented in Figure 3.6b. Hence, it can be said that the *proposed* RL-FODD yields a better estimation of the ideal fractional derivative than the conventional GL-FODD.

Furthermore, to illustrate the effectiveness of the *proposed* method at various fractional or-



(a) **(b)**

Figure 3.6: Ideal and estimated fractional-order derivatives of (a) $p_1(n)$ and (b) $p_2(n)$ for $\alpha = 0.8$ by using GL-FODD and RL-FODD.

ders, Table 3.3 represents RMSE and MAE values for both the signals at various values of α. From the results, it can be seen that the *proposed* method obtains better approximation to the ideal fractional-order derivatives for each α as compared to GL-FODD. This is evident from the values of RMSE and MAE obtained by the *proposed* method.

Table 3.3: Quantitative comparison between GL-FODD and the *proposed* RL-FODD at various fractional orders and $h = 0.05$

	$p_1(n)$				$p_2(n)$			
	RMSE		MAE		RMSE		MAE	
α	D^{α}_{GL}	D^{α}_{RL}	D^{α}_{GL}	D^{α}_{RL}	D^{α}_{GL}	D^{α}_{RL}	D^{α}_{GL}	D^{α}_{RL}
0.1	0.0877	**0.0873**	0.0515	0.0515	0.0100	**0.0037**	0.0074	**0.0016**
0.2	0.1427	**0.1419**	0.0767	0.0767	0.0145	**0.0037**	0.0103	**0.0017**
0.3	0.1725	**0.1713**	0.0847	0.0847	0.0164	**0.0028**	0.0109	**0.0014**
0.4	0.1826	**0.1809**	0.0816	0.0816	0.0165	**0.0019**	0.0103	**0.0009**
0.5	0.1774	**0.1753**	0.0723	**0.0719**	0.0155	**0.0012**	0.0090	**0.0005**
0.6	0.1608	**0.1583**	0.0599	**0.0585**	0.0136	**0.0008**	0.0077	**0.0002**
0.7	0.1360	**0.1332**	0.0469	**0.0437**	0.0110	**0.0006**	0.0062	**0.0003**
0.8	0.1063	**0.1037**	0.0338	**0.0301**	0.0082	**0.0006**	0.0047	**0.0004**
0.9	0.0772	**0.0756**	0.0221	**0.0190**	0.0056	**0.0014**	0.0034	**0.0011**

Therefore, it can be said that the *proposed* RL-FODD not only provides better approximation to the ideal FODD but also provides better results in estimating fractional-order derivatives of the given signal. Furthermore, it has been observed that as $h \rightarrow 0$, the approximation

error decreases. However, choosing very small value of h exhibits various challenges that needs to be addressed and hence, are discussed in the subsequent section.

Additionally, it has also been verified that for the fractional derivatives of order $\alpha (0 < \alpha < 1)$, the GL formula converge with order one with respect to h [106, 109], i.e., the convergence of GL-algorithm is of first-order, which originates from the fact that the approximation error at a point halves when the step-size h is halved (the order of convergence is obtained by computing $\log_2(E(h)/E(h/2))$, where $E(h)$ represents the error corresponding to step-size h [109]).

$$D^{\alpha}_{GL} p(t) = \sum_{k=0}^{L} \frac{b(k)}{h^{\alpha}} p(t - kh) + O(h) \tag{3.21}$$

where, $b(k)$ represents the filter coefficients of GL-FODD, as presented in (3.16).

On the other hand, the numerical approximation of RL definition based upon backward finite difference method can achieve an improved convergence of order $(2 - \alpha)$ [106, 109], i.e.,

$$D^{\alpha}_{RL} p(t) = \sum_{k=0}^{L} \frac{a(k)}{h^{\alpha}} p(t - kh) + O(h^{2-\alpha}) \tag{3.22}$$

where, $a(k)$ represents the *proposed* FIR filter coefficients of RL-FODD, presented in (3.10). The quality of this approach deteriorates as α increases. However, for $0 < \alpha < 1$, quite good convergence properties have been observed. This is in conformity with the well-established literature on fractional-order calculus [8, 106, 109] and hence, also conforms the analytical and simulation results presented in this section.

3.3 Need of Fractional Sample Delay Interpolation

As discussed in Section 3.2.2, the large approximation error in the high-frequency region can be reduced by decreasing the size of sampling interval h in (3.5), i.e., by choosing $h \to 0$. However, choosing very small value of h would require the need of high-end hardware to incorporate very large sampling rates. Therefore, an appropriate selection of h needs to be done in order to maintain an appropriate balance between the accuracy and hardware cost.

On the other hand, decreasing the value of h implies that for a given t, a large value of L is required (as $h = t/L$). However, a larger value of L would take more time for multiplication

and accumulation of large number set and hence, one would be left with a lower sampling rate and there may be issues to represent the signal correctly on account of the poor sampling rate alone. Furthermore, it will increase the implementation complexity of the method. Hence, a trade-off needs to be set for the choice of h and L so that the desired accuracy can be obtained without increasing complexity of the method.

This problem can, however, be solved by utilizing fractional sample delay interpolation [53] in the design of FODD without requiring the need of high-end hardware. Fractional delay implies, assuming uniform sampling, a delay that is non-integer multiple of sampling interval is introduced. As already mentioned, smaller the value of h is, better is the approximation. Therefore, assuming uniform sampling of the signal with sampling period T, h is chosen such that it is non-integer multiple of the sampling interval i.e., $h = dT$, where $0 < d < 1$.

However, when h is non-integer multiple of sampling interval T, and is expressed as $h = dT$, (3.11) can be rewritten as

$$D^{\alpha}_{RL}p[n-D] = \sum_{k=0}^{L} \frac{a[k]}{d^{\alpha}} p[n-D-kd] \tag{3.23}$$

Now, $p[n-D-kd]$ represents the non-integer delay samples of the signal $p[n]$ that would lie somewhere between the existing samples of $p[n]$. The appropriate values of these samples are hence, found via interpolation, most commonly known as fractional delay interpolation [44]. Taking z-transform on both sides of (3.23) gives

$$Y(z) = \left(\sum_{k=0}^{L} \frac{a[k]}{d^{\alpha}} z^{-(D+kd)} \right) P(z) \tag{3.24}$$

where

$$H(z) = \sum_{k=0}^{L} \frac{a[k]}{d^{\alpha}} z^{-(D+kd)} \tag{3.25}$$

In general, the z-transform holds only for integral powers of z, and hence, if $0 < d < 1$ is utilized, $z^{-(D+kd)}$ cannot be implemented directly. Therefore, to solve this problem, the well-documented design techiques of fractional delay filters needs to be utilized. Employing fractional delay filters facilitates the fine-tuning of sampling instants and hence, provide the observation of signal values at arbitrary locations between the samples [44].

Therefore, fractional sample delay interpolation is incorporated into the design process of the *proposed* RL-FODD which significantly improves the frequency response approximation of the *proposed* method without requiring the need of high-end hardware to meet the specification of high sampling rates.

3.4 Design of RL-FODD using Fractional Delay Interpolation (FDI)

This section illustrates the improved design of the *proposed* RL-FODD by utilizing fractional delay interpolation techniques. For this, the delay $D + kd$, presented in (3.25) is segregated into the sum of an integer and fractional part as described below:

$$D + kd = I_k + f_k \tag{3.26}$$

where $I_k = \lfloor D + kh \rfloor$ denotes the integer part (here $\lfloor . \rfloor$ represents the $floor$ operator) and $f_k = D + kh - I_k$ denotes the fractional part of the delay value. Substituting (3.26) into (3.25), implies

$$H(z) = \sum_{k=0}^{L} \frac{a[k]}{d^{\alpha}} z^{-(I_k + f_k)} \tag{3.27}$$

Now, to implement $z^{-(I_k+f_k)}$, fractional sample delay interpolation techniques based on FIR Lagrange interpolator and radial basis function interpolation are incorporated into the design process of the *proposed* RL-FODD, the details of which are presented as follows:

3.4.1 Lagrange Interpolation based Design of RL-FODD (L-RL-FODD)

In this case, Lagrange interpolation is utilized to estimate the coefficients of an FIR filter that can approximate the fractional sample delay $z^{-(I_k+f_k)}$, presented in (3.27). Then, it is incorporated into the design process of RL-FODD so as to yield analytical formulation of Lagrange interpolation based design of RL-FODD (*herein, referred to as L-RL-FODD*).

3.4.1.1 Mathematical Investigation of L-RL-FODD

In Lagrange interpolation, the fractional sample delays $z^{-(I_k+f_k)}$, presented in (3.27) are estimated by

$$z^{-(I_k+f_k)} \cong \sum_{m=0}^{2I_k} q_k(m) z^{-m} \tag{3.28}$$

where $q_k(m)$ represents the FIR filter coefficients of Lagrange fractional delay filter deduced from (2.14), and are given by

$$q_k(m) = \prod_{l=0, l\neq m}^{2I_k} \frac{I_k + f_k - l}{m - l} \tag{3.29}$$

Substituting (3.29) into (3.27), implies

$$\begin{aligned} H_1(z) &\cong \sum_{k=0}^{L} \frac{a[k]}{d^\alpha} \sum_{m=0}^{2I_k} q_k(m) z^{-m} \\ &= \sum_{k=0}^{L} \frac{a[k]}{d^\alpha} \left[\sum_{m=0}^{2I_k} \left(\prod_{l=0, l\neq m}^{2I_k} \frac{I_k + f_k - l}{m - l} \right) z^{-m} \right] \\ &= \frac{1}{d^\alpha} \sum_{m=0}^{2I_k} \left(\sum_{k=0}^{L} a[k] \prod_{l=0, l\neq m}^{2I_k} \frac{I_k + f_k - l}{m - l} \right) z^{-m} \\ &= \sum_{m=0}^{2I_k} a_1(m) z^{-m} \end{aligned} \tag{3.30}$$

where

$$a_1(m) = \frac{1}{d^\alpha} \sum_{m=0}^{2I_k} \left(\sum_{k=0}^{L} a[k] \prod_{l=0, l\neq m}^{2I_k} \frac{I_k + f_k - l}{m - l} \right) \tag{3.31}$$

Therefore, $a_1(m)$ represents the Lagrange interpolation based FIR filter coefficients of the *proposed* RL-FODD. Now, these filter coefficients are further utilized to yield the frequency response of FODD.

3.4.1.2 L-RL-FODD and L-GL-FODD – A Comparative Analysis

To illustrate the effectiveness of the *proposed* L-RL-FODD, a design example with parameters $L = 100, D = 20, d = 0.2$ and $\alpha = 0.8$ is considered, where L represents the prescribed length of the filter, D represents the integer delay, d represents the fractional sample delay, and α represents the fractional-order. Figure 3.7a illustrates the normalized magnitude re-

sponse $|H_1(e^{j\omega})|$ (solid line) of the *proposed* L-RL-FODD. The dashed line represent the ideal response of FODD.

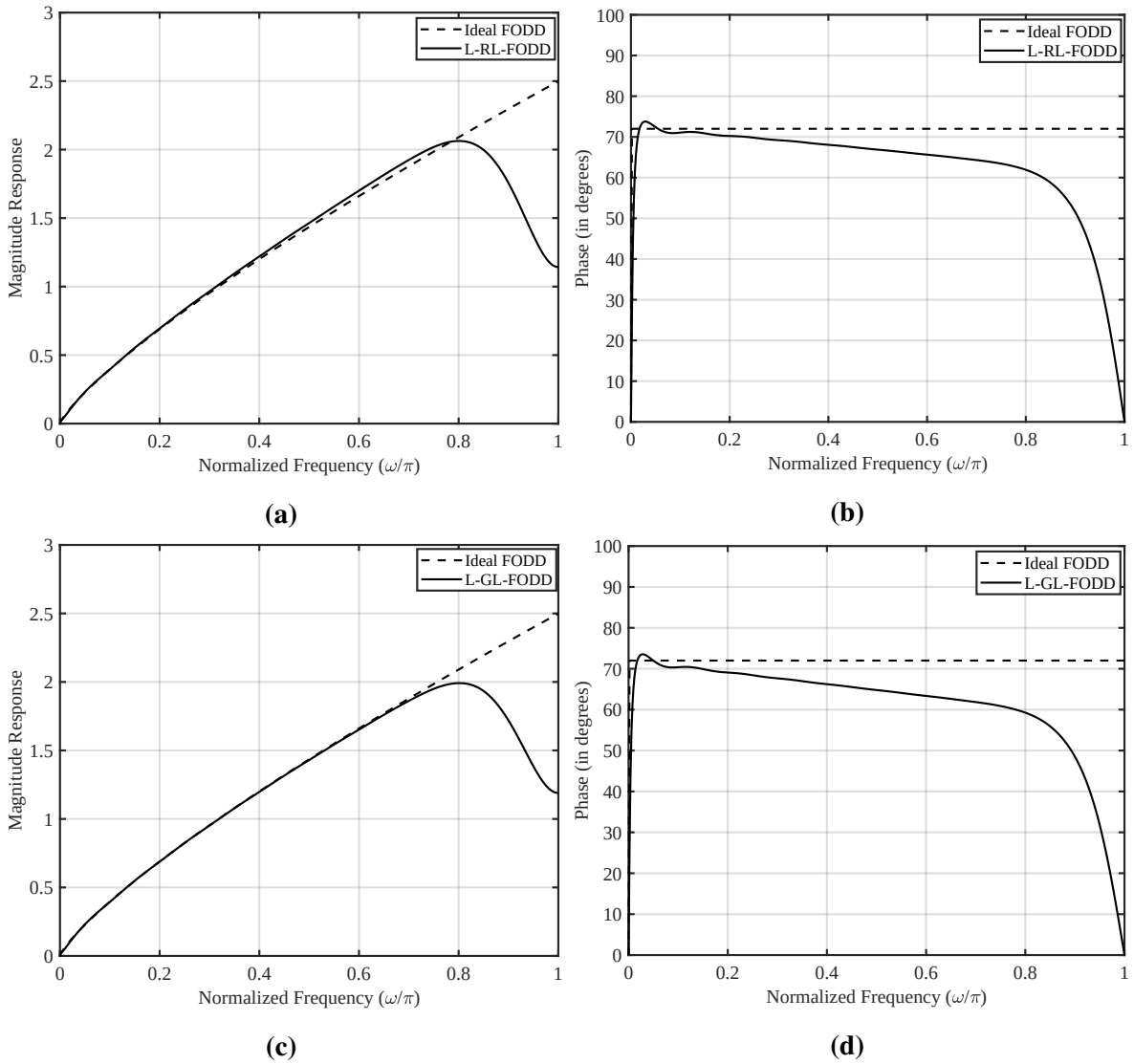


Figure 3.7: (a) Magnitude response and (b) phase response (solid lines) of the *proposed* L-RL-FODD using parameters $L = 100, D = 20, d = 0.2, \alpha = 0.8$; (c) magnitude response and (d) phase response of the conventional L-GL-FODD. The dashed lines represent the ideal response of FODD.

Furthermore, Figure 3.7b represents the phase response of the *proposed* L-RL-FODD. The normalized phase response in degrees can be estimated by computing $90 \times (\angle H(e^{j\omega}) + \omega I_k)/(\pi/2)$.

Comparing the results of the *proposed* RL-FODD and L-RL-FODD, presented in Figure 3.3a and Figure 3.7a, respectively, it can be observed that the design of L-RL-FODD has shown a significant improvement over that of RL-FODD, presented in Figure 3.3a. Therefore, *it*

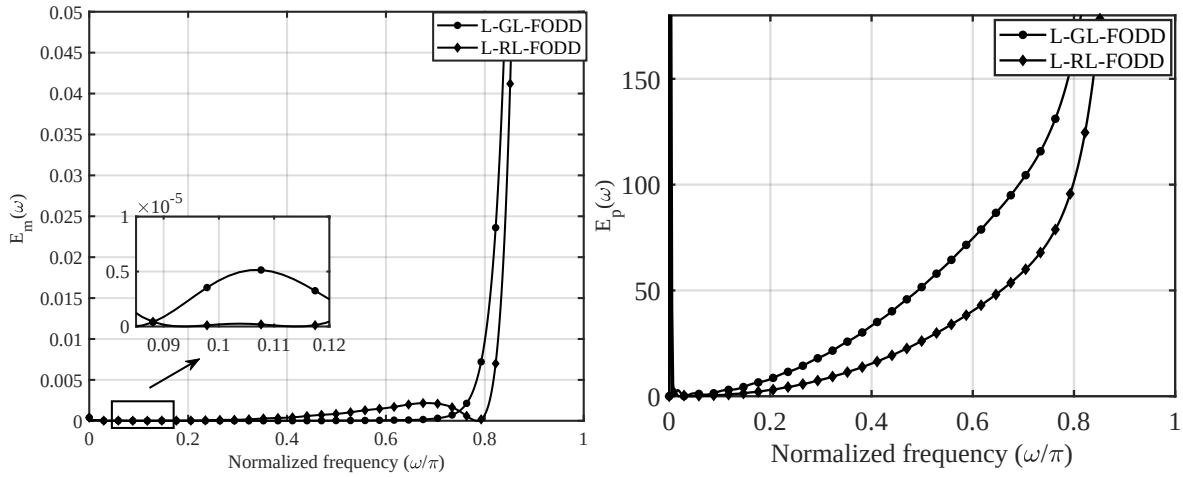
can be said that the approximation to the ideal frequency response has been significantly improved by incorporating fractional sample delay in the design process. Furthermore, it can be observed from (3.30) that the length of the *proposed* L-RL-FODD is $2I_L+1 = 2(D+Ld)+1$. Hence, by utilizing the parameters listed above, the length of the filter $H_1(z)$ illustrated in Figure 3.7a is $2(D + kd) + 1 = 81$, which is less that the length $L + 1 = 101$ of $H(z)$, as presented in (3.14). Therefore, incorporating fractional sample delay interpolation improves the design accuracy without increasing implementation complexity of the method.

Furthermore, the efficiency of the *proposed* L-RL-FODD design is tested against that of the well-established Lagrange interpolation based GL-FODD (*herein, referred to as L-GL-FODD*) design, presented in [53]. The same design parameters are chosen for conventional L-GL-FODD so as to make fair comparison between two designs. Figure 3.7c illustrates the normalized magnitude response of the conventional L-GL-FODD [53]. However, phase response is not considered in [53] and is presented here in Figure 3.7d for comparison purposes.

From the results presented in Figure 3.7, it can be observed that the magnitude response approximations of the conventional L-GL-FODD [53] and the *proposed* L-RL-FODD designs are almost similar except in the mid- and high-frequency regions. However, the magnitude response errors of the *proposed* L-RL-FODD and the conventional L-GL-FODD [53] are computed to be $E = 0.0091$ and $E = 0.0120$, respectively for the frequency band of interest, i.e., $B = [0, 0.9\pi]$. The corresponding phase response errors for the *proposed* L-RL-FODD and L-GL-FODD [53] are found to be $E = 52.07$ and $E = 80.50$, respectively.

In quantitative terms, the approximated magnitude and phase responses using the *proposed* L-RL-FODD has shown a percentage improvement of 24.17% and 35.31%, respectively, over that of L-GL-FODD, for fractional-order $\alpha = 0.8$. Therefore, it can be observed that the *proposed* L-RL-FODD provides better approximation of the ideal frequency response of FODD as compared to conventional L-GL-FODD [53]. The magnitude and phase response errors of the *proposed* L-RL-FODD for various fractional orders are further reported in Section 3.5.

To illustrate the detailed comparison between the *proposed* L-RL-FODD and the conventional L-GL-FODD [53], the squared magnitude response (denoted by $E_m(\omega)$) and phase response (denoted by $E_p(\omega)$) errors have been plotted against normalized frequencies (i.e.,



(a) **(b)**

Figure 3.8: (a) Magnitude response and (b) Phase response errors of the conventional L-GL-FODD and the *proposed* L-RL-FODD with respect to normalized frequencies (ω/π).

ω/π) for $\alpha = 0.8$ in Figure 3.8. From Figure 3.8a, it can be observed that the magnitude response approximation of the *proposed* L-RL-FODD has shown a significant improvement in the frequency ranges $0.1\pi < \omega < 0.2\pi$ and $0.75\pi < \omega < 0.85\pi$ against that of conventional L-GL-FODD [53]. However, in the mid-frequency range i.e., $0.4\pi < \omega < 0.7\pi$, the magnitude response error of the *proposed* L-RL-FODD is slightly higher than that of L-GL-FODD. Nevertheless, as mentioned previously, the overall magnitude response error of the *proposed* L-RL-FODD is less than that of conventional L-GL-FODD [53], which illustrates the superiority of the *proposed* method.

Furthermore, the *proposed* L-RL-FODD outperforms the conventional L-GL-FODD [53] in phase response approximation, as depicted in Figure 3.8b. From the figure, it can be observed that the phase response error of the *proposed* L-RL-FODD is significantly less than that of conventional L-GL-FODD [53] over the entire frequency range, which indicates the superiority of the *proposed* method in terms of frequency response approximation. Therefore, it can be said that overall frequency response approximation of the ideal differentiator has been improved by utilizing the *proposed* L-RL-FODD in comparison to the conventional L-GL-FODD [53].

3.4.2 Radial Basis Function based Design of RL-FODD (RBF-RL-FODD)

RBF interpolation, which has found utmost importance in multivariate interpolation, neural networks, control of nonlinear systems, etc. has also been utilized here to improve the efficiency of the *proposed* RL-FODD design method in the high-frequency region [54].

RBF interpolation method is utilized to estimate the non-integer delay samples $p(n-\mu)$ from the given integer delay samples $p(n), p(n-1), ...p(n-N)$, where μ denotes the delay. Using this approach, the non-integer delay samples are estimated by

$$p(n-\mu) = \sum_{m=0}^{N} g(m,\mu)p(n-m) \tag{3.32}$$

where the weights $g(m,\mu)$ are determined by utilizing RBF interpolation, as presented in [54]. Now, in the subsequent section, these non-integer delay samples estimated by RBF interpolation will be incorporated in the design process of the *proposed* RL-FODD to improve its design accuracy in the high-frequency region.

3.4.2.1 Mathematical Investigation of RBF-RL-FODD

The non-integer sample delays $p(n-D-kd)$, presented in (3.23), can be represented as

$$p(n-D-kd) = \sum_{m=0}^{N} g(m,D+kd)p(n-m) \tag{3.33}$$

where the weights $g(m,D+kd)$ are obtained by utilizing RBF interpolation, as presented in [54]. Substituting $D+kh = I_k + f_k$, as presented in (3.26), implies

$$p(n-I_k-f_k) = \sum_{m=0}^{N} g(m,I_k+f_k)p(n-m) \tag{3.34}$$

Now, taking z-transform on both sides of (3.34) gives

$$z^{-(I_k+f_k)} = \sum_{m=0}^{N} g(m,I_k+f_k)z^{-m} \tag{3.35}$$

Substituting (3.35) into (3.27), the transfer function of the *proposed* RBF-based RL-FODD (*herein, referred to as RBF-RL-FODD*) can be represented as

$$\begin{aligned} H_2(z) &\cong \sum_{k=0}^{L} \frac{a[k]}{d^{\alpha}} \sum_{m=0}^{N} g(m, I_k + f_k) z^{-m} \\ &= \sum_{m=0}^{N} \frac{1}{d^{\alpha}} \left(\sum_{k=0}^{L} a[k] g(m, I_k + f_k) \right) z^{-m} \end{aligned} \tag{3.36}$$

$$H_2(z) = \sum_{m=0}^{N} a_2(m) z^{-m} \tag{3.37}$$

where

$$a_2(m) = \frac{1}{d^{\alpha}} \left(\sum_{k=0}^{L} a[k] g(m, I_k + f_k) \right) \tag{3.38}$$

Thus, $a_2(m)$ represents the RBF interpolation-based FIR filter coefficients of the *proposed* RL-FODD. After establishing the closed-form analytical formulation for the design of RBF-RL-FODD, the next aim is to validate the efficiency of the *proposed* RBF-RL-FODD against that of well-established RBF-interpolation based design of GL-FODD (*herein, referred to as RBF-GL-FODD*) [54].

3.4.2.2 RBF-RL-FODD and RBF-GL-FODD – A Comparative Analysis

The comparison between the *proposed* RBF-RL-FODD and the conventional RBF-GL-FODD [54] is illustrated in Figure 3.9. The design parameters are selected as $N = 100, L = 620, d = 0.05, D = 30$ and $\alpha = 0.8$ for both cases, where N represents the length of the *proposed* RBF-RL-FODD. Figure 3.9a and Figure 3.9b depict the normalized magnitude $|H_2(e^{j\omega})|$ and phase responses $\angle H_2(e^{j\omega})$ (solid lines) of *proposed* RBF-RL-FODD. The dashed lines represent the ideal frequency response. Figure 3.9c and 3.9d depict the corresponding magnitude and phase responses of the conventional RBF-GL-FODD design, presented in [54].

Comparing Figure 3.9a and 3.9b with that of Figure 3.9c and 3.9d, respectively, it can be seen that the *proposed* RBF-RL-FODD design has a better magnitude response at low-frequency range than the method presented in [54]. However, in the mid- and high-frequency regions, the conventional RBF-GL-FODD provides better approximation to the ideal FODD in comparison to the *proposed* RBF-RL-FODD. The magnitude response errors for the

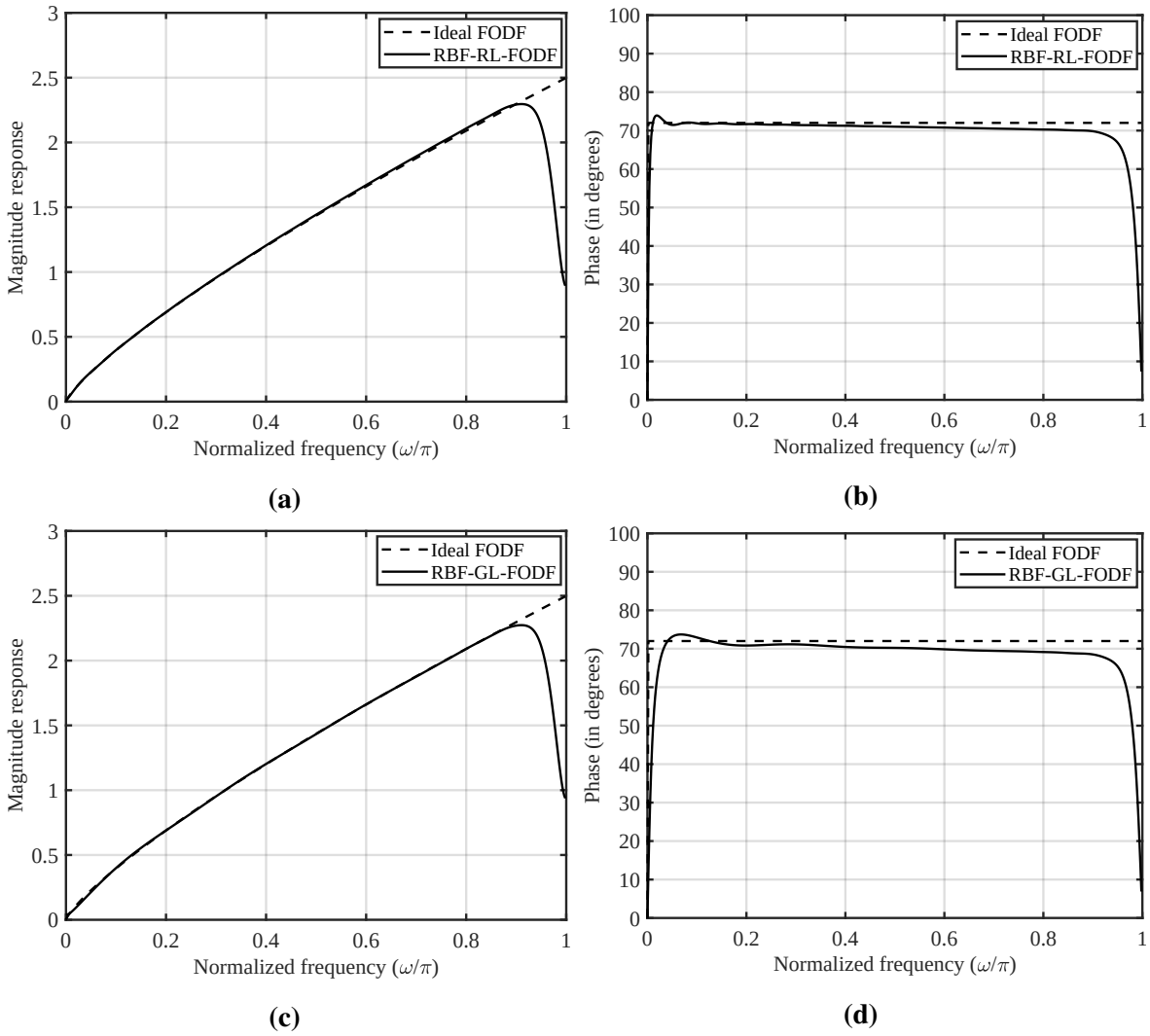


Figure 3.9: (a) Magnitude response and (b) phase response (solid lines) of the *proposed* RBF-RL-FODD using parameters $N = 100, L = 620, D = 30, d = 0.05, \alpha = 0.8$; (c) magnitude response and (d) phase response of the conventional RBF-GL-FODD. The dashed lines represent the ideal response of FODD.

proposed RBF-RL-FODD and the conventional RBF-GL-FODD [54] are reported to be $E = 0.852 \times 10^{-4}$ and $E = 0.306 \times 10^{-4}$ respectively. However, the *proposed* RBF-RL-FODD outperforms the conventional RBF-GL-FODD in terms of phase response approximation over the entire frequency range, as can be visualized from Figure 3.9b and 3.9d. The corresponding phase response errors of the *proposed* RBF-RL-FODD and conventional RBF-GL-FODD [54] are found to be $E = 4.6598$ and $E = 27.214$, respectively.

The magnitude and phase response errors of the *proposed* RBF-RL-FODD for various fractional orders are further reported in Section 3.5.

Similarly, to indicate the effectiveness of the *proposed* method, the magnitude and phase

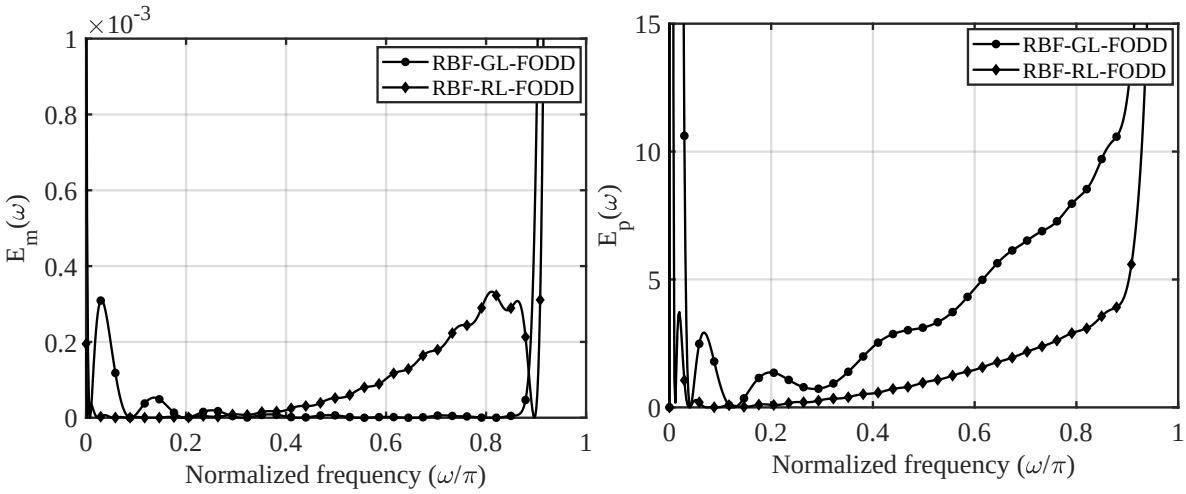


Figure 3.10: (a) Magnitude response and (b) Phase response errors of the conventional RBF-GL-FODD and the *proposed* RBF-RL-FODD with respect to normalized frequencies (ω/π).

response errors of the *proposed* RBF-RL-FODD and the conventional RBF-GL-FODD [54] have been plotted against the range of normalized frequencies, as presented in Figure 3.10a and 3.10b, respectively. From Figure 3.10a, it can be observed that the *proposed* RBF-RL-FODD outperforms the conventional RBF-GL-FODD [54] in magnitude response approximation in the frequency ranges $0 < \omega < 0.3\pi$ and $0.85\pi < \omega < 0.95\pi$. However, the magnitude response error of the *proposed* RBF-RL-FODD is higher than that of RBF-GL-FODD [54] in the range $0.3\pi < \omega < 0.85\pi$.

However, the corresponding phase response error plot, presented in Figure 3.10b indicates that the *proposed* RBF-RL-FODD outperforms the conventional RBF-GL-FODD [54] over the entire frequency range. The phase response error of the *proposed* RBF-RL-FODD is found to be $E = 4.6598$ which is significantly less than that of RBF-GL-FODD, where MSE is reported to be $E = 27.214$. Therefore, it can be said that the *proposed* RBF-RL-FODD shows a significant improvement in frequency response approximation of the ideal differentiator as compared to that of the conventional RBF-GL-FODD [54].

Therefore, it can be said that incorporating fractional delay interpolation in the design process of FODD significantly enhances the design accuracy and hence, provides better approximation to the ideal FODD. Furthermore, it can be said that the RL-based designs of FODD yield a superior performance than the conventional well-established GL-based design meth-

ods presented in the literature.

Once the filter coefficients are obtained, the *proposed* filters (i.e. L-RL-FODD, RBF-RL-FODD) can be implemented by using time-domain convolution, which requires $O(N^2)$ operations. However, the *proposed* filters can also be implemented in frequency domain, which reduces the total order in complexity to $O(NlogN)$. Since the same design parameters have been utilized for the design of RL-based FODD so that fair comparison can be made between conventional GL-based and the *proposed* RL-based methods, the implementation complexity of both methods is same.

3.5 FDI based RL-FODD and GL-FODD – A Comparative Analysis

From the analytical and simulation studies conducted, it has been proved that incorporating fractional sample delay in the design of FODD significantly improves the design accuracy in the high-frequency region. Now, this section provides the detailed comparison between FDI based design techniques of *proposed* RL-FODD and GL-FODD.

Figure 3.11 represents the comparison among the magnitude and phase responses of the *proposed* L-RL-FODD and RBF-RL-FODD with that of L-GL-FODD and RBF-GL-FODD, presented in [53] and [54], respectively. The results are plotted for fractional-order $\alpha = 0.5$ using the same design parameters as discussed in Section 3.4. On comparing the magnitude responses of the *proposed* L-RL-FODD and L-GL-FODD [53] presented in Figure 3.11a, it can be seen that the magnitude responses are almost the same for both the cases. However, the MSE value $E_m = 0.00167$ for the case of L-GL-FODD is slightly higher than the value $E_m = 0.00132$ for L-RL-FODD over the frequency band $B = [0, 0.9\pi]$. The approximated magnitude response of the *proposed* L-RL-FODD has shown an average percentage improvement of 20.95% over that of conventional L-GL-FODD [53]. Comparing the phase responses of the *proposed* L-RL-FODD and L-GL-FODD [53] presented in Figure 3.11b, a drastic improvement of 50.65% has been observed.

Furthermore, comparing the designs of *proposed* RBF-RL-FODD with that of the conventional RBF-GL-FODD in Figure 3.11c, it can be depicted that the magnitude responses of the *proposed* RBF-RL-FODD and RBF-GL-FODD [54] are almost similar except in the low-

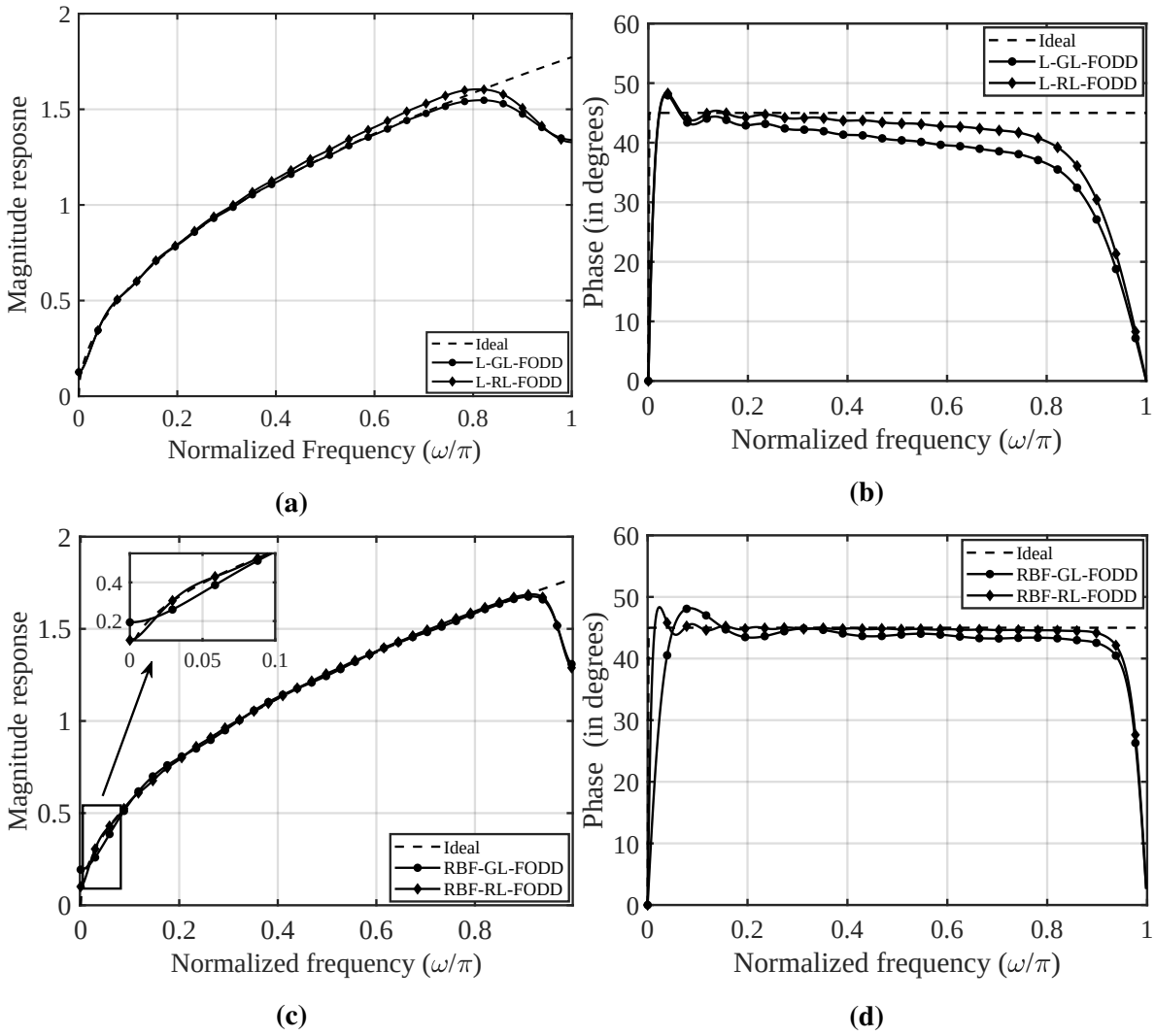


Figure 3.11: (a) Comparison among magnitude responses and (b) phase responses (in degrees) of L-GL-FODD with that of the *proposed* L-RL-FODD for $\alpha = 0.5$; (c) comparison among magnitude responses and (d) phase responses (in degrees) of RBF-GL-FODD with that of the *proposed* RBF-RL-FODD for $\alpha = 0.5$.

frequency region. In a very-low-frequency range, the *proposed* RBF-RL-FODD outperforms RBF-GL-FODD in [54]. The MSE for the RBF-RL-FODD is $E_m = 0.633 \times 10^{-4}$, which is significantly less than the error $E_m = 3.13 \times 10^{-4}$ obtained by utilizing RBF-GL-FODD [54].

In addition, from the results presented in Figure 3.11d, it can be inferred that the *proposed* RBF-RL-FODD has shown a drastic improvement in approximating the ideal phase response of FODD. The percentage phase response improvement over that of conventional RBF-GL-FODD [54] is approximately 79.5%, which indicates the significant overall improvement of the design method. This accurate representation of phase response can play a crucial role in

Table 3.4: MSE comparison between magnitude and phase responses of L-GL-FODD and the *proposed* L-RL-FODD

Fractional order α	$\lvert H_1(e^{j\omega})\rvert$		$\angle H_1(e^{j\omega})$	
	L-GL-FODD	L-RL-FODD	L-GL-FODD	L-RL-FODD
0.1	**0.00111**	0.00112	2.1745	**1.0801**
0.2	**0.00063**	0.00065	8.2399	**4.0171**
0.3	0.00051	0.00051	17.485	**8.4358**
0.4	0.00079	**0.00069**	29.120	**14.076**
0.5	0.00167	**0.00132**	42.254	**20.856**
0.6	0.00350	**0.00263**	55.897	**28.967**
0.7	0.00677	**0.00500**	68.996	**38.997**
0.8	0.01200	**0.00911**	80.509	**52.072**
0.9	0.01966	**0.01637**	89.783	**70.259**

Bold values indicate the lowest values of MSE.

modelling and processing of time–frequency representation of signals, for example in computing instantaneous frequency, which is defined as time derivative of instantaneous phase.

For quantitative comparisons, the magnitude response and phase response errors have been calculated for various fractional orders. The design errors for various fractional orders are reported in Table 3.4 over the frequency band of interest, i.e. $B = [0, 0.9\pi]$.

From the results presented in Table 3.4, it can be depicted that the magnitude response errors of the conventional L-GL-FODD [53] and L-RL-FODD are quite similar for $\alpha \leq 0.3$. However, for $\alpha > 0.3$, the *proposed* L-RL-FODD shows a significant improvement over that of L-GL-FODD [53]. On the other hand, the *proposed* L-RL-FODD provides a remarkable improvement in approximating the phase response of ideal FODD for the entire range of fractional orders α.

Similarly, comparing the MSE of RBF-GL-FODD [54] with that of the *proposed* RBF-RL-FODD presented in Table 3.5, it can be surveyed that for fractional orders $\alpha < 0.7$, RBF-RL-FODD shows a significant improvement over that of RBF-GL-FODD. However, for fractional orders $\alpha \geq 0.7$, the variation between the magnitude responses of the *proposed* RBF-RL-FODD and conventional RBF-GL-FODD [54] is not large. Furthermore, comparing the phase response errors presented in Table 3.5, it can be depicted that *proposed* RBF-RL-FODD outperforms RBF-GL-FODD, presented in [54], in approximating phase re-

sponse effectively for the entire range of fractional orders α.

On the other hand, comparing the results presented in Table 3.4 and 3.5, it can be seen that *RBF interpolation yields better accuracy than that of Lagrange interpolation.* This is because for Lagrange interpolation, the approximation error is often unnecessarily small at low frequencies, at the cost of performance at higher frequencies [44]. Therefore, comparing the design results of RL-based FIR FODD, L-RL-FODD, and RBF-RL-FODD, it has been proved that the *proposed* RBF-RL-FODD outperforms the other two designs.

Table 3.5: MSE comparison between magnitude and phase responses of RBF-GL-FODD and the *proposed* RBF-RL-FODD

Fractional order α	$\lvert H_1(e^{j\omega})\rvert$		$\angle H_1(e^{j\omega})$	
	RBF-GL-FODD	RBF-RL-FODD	RBF-GL-FODD	RBF-RL-FODD
0.1	14.53×10^{-4}	$\mathbf{9.879\times10^{-4}}$	1.8226	**0.4295**
0.2	10.93×10^{-4}	$\mathbf{4.567\times10^{-4}}$	6.3916	**1.4653**
0.3	7.922×10^{-4}	$\mathbf{2.115\times10^{-4}}$	12.503	**2.7802**
0.4	5.251×10^{-4}	$\mathbf{1.023\times10^{-4}}$	19.055	**4.0866**
0.5	3.130×10^{-4}	$\mathbf{0.633\times10^{-4}}$	24.970	**5.1212**
0.6	1.651×10^{-4}	$\mathbf{0.629\times10^{-4}}$	29.138	**5.6498**
0.7	$\mathbf{0.756\times10^{-4}}$	0.796×10^{-4}	30.356	**5.4980**
0.8	$\mathbf{0.306\times10^{-4}}$	0.852×10^{-4}	27.214	**4.6598**
0.9	$\mathbf{0.146\times10^{-4}}$	0.473×10^{-4}	17.931	**3.8029**

Bold values indicate the lowest values of MSE.

The *proposed* design based on RL fractional-order derivative bridges the gap in the field of FODD, where the design methods are mostly considered by taking GL fractional-order derivative into account. Therefore, the *proposed* RL-based methods serve as an effective alternative to the already existing design techniques of FODD based upon GL fractional-order differential operator. Hence, it can be said that the existing designs of FODD can be improved by utilizing the definition of RL fractional-order derivative and thus, could be beneficial for various signal processing applications.

In a nutshell, the significance of the *proposed* work can be concluded as follows:

1. The improvement obtained by utilizing the *proposed* method is highly significant in obtaining exact fractional-order derivatives of the signals. For fractional-orders $0 <$

$\alpha < 1$, the conventional GL-based formula can only achieve an accuracy of order 1. However, the *proposed* RL-based numerical scheme can achieve an improved accuracy of order $(2-\alpha)$ [106, 83, 108]. This is in conformity to the simulation results obtained in Section 3.2.2 for various signals, where it has been observed that the *proposed* RL-FODD yields a better estimation of the ideal fractional derivative than the conventional GL-FODD.

2. On the other hand, it has been observed that the *proposed* method significantly outperforms the conventional design methods in approximating the phase response of ideal differentiator. This significant improvement in phase response of ideal FODD is depicted in Figure 3.8 and Figure 3.10. This improvement in phase response approximation can play a crucial role in modelling and processing of time-frequency representation of signals; for example, in computing instantaneous frequency which is defined as time-derivatives of instanteanous phase [75]. Furthermore, the improvement in phase response can be highly beneficial in speech signal processing where reduction in phase distortion plays a significant role.

3. The better the method is to compute fractional-order derivatives of the signals, the better the fractal process can be generated [54]. This can be highly significant in biomedical signal analysis where many biomedical signals have been shown to exhibit $1/(j\omega)^{\alpha}$-power spectra over several decades [8]. Examples of these signals include QRS complex of electrocardiogram (ECG) signals, magnetoencephalography (MEG) and electroencephalography (EEG) signals of spontaneous neural activity in humans, loudness fluctuations in speech and music, to name a few [112]. Therefore, it can be said that the improvement shown by the *proposed* method can prove to be beneficial in many real applications.

3.6 Summary

This chapter establishes a novel analytical approach based on Riemann-Liouville fractional differential operator in order to design fractional-order digital differentiator (FODD). Since GL-based methods have been extensively investigated by the research community for the design of FODD, this chapter bridges the gap in the field of FODD where there seems to

be an improvement window for designing FIR filters based upon RL fractional differential operator.

Therefore, a closed-form analytical expression is formulated for computing FIR filter coefficients of RL-FODD and its performance is compared against well-established design methods of GL-FODD, based upon their ability to yield ideal frequency response of FODD. Two fractional delay interpolation techniques known as Lagrange interpolation and radial basis function interpolation have been incorporated in the design of RL-FODD in order to improve its design accuracy in the high-frequency region.

From the analytical and simulation studies conducted, it can be ascertained that the *proposed* method not only can efficiently approximate the ideal frequency response of FODD but can also provide much better results than the conventional well-established designs of GL-FODD in estimating fractional-order derivatives of the signals.

Chapter 4

Atangana-Baleanu Fractional Differential-Based Design of Fractional-Order Digital Filter (FODF)

ATANGANA AND BALEANU, in 2016, tried to address a challenge that whether it is possible to construct another type of fractional-order differential operator which can better describe the dynamics of non-local phenomena and has non-singular kernel. As a result, Atangana and Baleanu (AB) proposed a novel fractional-order operator incorporating a kernel which is non-singular as well as non-local, namely Mittag-Leffler function.

As mentioned earlier, Mittag-Leffler function possesses significant advantages in fractional calculus theory. Therefore, AB fractional-order operator has been explored in various fields of science and engineering. However, applications of AB fractional-order operator in signal and image processing area have not yet been fully explored, and hence, this chapter aims to add to its applications by establishing AB-based design of fractional-order digital filter (FODF) for signal and image processing applications.

4.1 Atangana-Baleanu Fractional-Order Differential Operator

Owing to the limitation of singularity in conventional RL and Caputo fractional-order differential operators, Atangana and Baleanu proposed two definitions of non-local and non-singular fractional-order differential operators [31]. One is termed as ABR fractional-order differential operator (where R denotes Riemann-Liouville type) and the other is ABC

fractional-order differential operator (where C denotes Caputo type). The definitions of ABR and ABC fractional-order differential operators from signal processing viewpoint are given as follows:

4.1.1 Atangana-Baleanu Fractional-Order Differential in RL sense

From the viewpoint of signal processing literature, the ABR fractional-order differential operator can be defined as the rate of change of convolution between the signal and the Mittag-Leffler kernel (denoted by $E_\alpha(.)$) [92], represented by

$$D^{\alpha}_{ABR}\, p(t) = \frac{M(\alpha)}{1-\alpha}\frac{d}{dt}\left[p(t) * E_\alpha\left(-\frac{\alpha}{1-\alpha}t^\alpha\right)\right] \tag{4.1}$$

which can further be expressed as

$$D^{\alpha}_{ABR}\, p(t) = \frac{M(\alpha)}{1-\alpha}\frac{d}{dt}\int_0^t p(\zeta)E_\alpha\left(-\frac{\alpha}{1-\alpha}(t-\zeta)^\alpha\right)d\zeta \tag{4.2}$$

where "*" represents the convolution operator and $M(\alpha)$ represents the normalization constant defined as $M(\alpha) = 1-\alpha+\alpha/\Gamma(\alpha)$, such that $M(0) = M(1) = 1$. The normalization constant $M(\alpha)$ allows some values of α to contribute more than others and hence, provides the freedom to weight different values of α as desired [40].

The ABR fractional-order differential operator is in correspondance with the conventional RL (denoted by D^{α}_{RL}) fractional-order differential operator which incorporate power-law kernel and is defined as [107]

$$D^{\alpha}_{RL}\, p(t) = \frac{d}{dt}\left(p(t) * \frac{t^{-\alpha}}{\Gamma(1-\alpha)}\right) \tag{4.3}$$

4.1.2 Atangana-Baleanu Fractional-Order Differential in Caputo sense

In a similar sense, the ABC fractional-order differential operator can be viewed as convolution between derivative of the signal and Mittag-Leffler kernel [92], defined as

$$D^{\alpha}_{ABC}\, p(t) = \frac{M(\alpha)}{1-\alpha}\left[\frac{d}{dt}p(t) * E_\alpha\left(-\frac{\alpha}{1-\alpha}t^\alpha\right)\right] \tag{4.4}$$

which can further be written as

$$D^{\alpha}_{ABC}\, p(t) = \frac{M(\alpha)}{1-\alpha} \int_0^t p'(\zeta) E_{\alpha}\left(-\frac{\alpha}{1-\alpha}(t-\zeta)^{\alpha}\right) d\zeta \tag{4.5}$$

This is in correspondance with the conventional Caputo (denoted by D^{α}_{C}) fractional-order differential operator which incorporate power-law kernel and is defined as [107]

$$D^{\alpha}_{C}\, p(t) = \frac{d}{dt} p(t) * \frac{t^{-\alpha}}{\Gamma(1-\alpha)} \tag{4.6}$$

4.1.3 ABR and ABC Fractional-Order Differential Operators

Although RL fractional derivative has been applied successfully in many areas of engineering, unfortunately, it leads to initial conditions involving fractional exponents when solving fractional differential equations. There is no physical interpretation for such type of initial conditions. Therefore, solution to this conflict was proposed by M. Caputo in terms of Caputo fractional derivative [2]. Caputo fractional derivative is suitable for real-world problems due to the fact that it allows usual initial conditions [85] when playing with integral transforms for instance, Laplace transform.

The Caputo fractional derivative is more suited than the RL derivative because the differentiation appears inside the integral instead of outside as in RL derivative, and hence, it alleviates the effects of noise and numerical integration, as is elaborated by eminent researchers in [3, 7, 107]. Also, there has been a lot of debate about the usage and the practicality between the RL and Caputo based fractional derivative definitions in the research community.

However, as already mentioned, it has been pointed out recently that the power kernel associated with these fractional-differential operators (i.e., RL and Caputo) exhibits an artificial singularity at its initial point, due to which many real-life phenomena cannot be implemented exactly [9]. Therefore, Atangana and Baleanu (AB) suggested non-local and non-singular generalized Mittag-Leffler function as a new kernel, which possesses wide range of advantages in the field of fractional differential theory itself.

Hence, owing to conventional well-established RL and Caputo fractional-order derivatives, Atangana and Baleanu replaced power-law kernel with Mittag-Leffler kernel and hence, proposed two definitions: Atangana-Baleanu in Reimann-Liouville (ABR) sense and Atangana-

Baleanu in Caputo (ABC) sense. ABC definition is considered to be useful for real world problems [31, 43, 92], as it has better relations with the Laplace transform and because the differentiation appears inside instead outside the integral, so to alleviate the effects of noise and numerical differentiation.

4.1.4 Properties of AB Fractional-Order Differential Operator

On the other hand, some of the useful properties pertaining to Atangana-Baleanu fractional-order differential operators are further reviewed briefly as follows:

Let $0 < \alpha < 1$, then the corresponding Laplace transforms of ABR and ABC fractional-order differential operators are defined as [40]

$$\mathscr{L}\{D^{\alpha}_{ABR}\, p(t)\} = \frac{M(\alpha)}{1-\alpha}\frac{s^{\alpha}\mathscr{L}\{p(t)\}}{s^{\alpha}+\frac{\alpha}{1-\alpha}} \tag{4.7}$$

$$\mathscr{L}\{D^{\alpha}_{ABC}\, p(t)\} = \frac{M(\alpha)}{1-\alpha}\frac{s^{\alpha}\mathscr{L}\{p(t)\} - s^{\alpha-1}p(0)}{s^{\alpha}+\frac{\alpha}{1-\alpha}} \tag{4.8}$$

where $\mathscr{L}$ represents the Laplace transform operator. The relation between ABC and ABR fractional differential operators can be established by using Laplace transforms of both the definitions, which is represented by [40]

$$D^{\alpha}_{ABC}\, p(t) = D^{\alpha}_{ABR}\, p(t) - \frac{M(\alpha)}{1-\alpha}\, p(0) E_{\alpha}\left(-\frac{\alpha}{1-\alpha}t^{\alpha}\right) \tag{4.9}$$

The various other properties and integral transforms such as Sumudu transform, Fourier transform, Mellin transform associated with AB fractional-order differential operators can be referred in detail from [113].

On the other hand, the transfer function of ABC fractional-order differential operator by assuming zero initial conditions can be computed from (4.8) and is given by [42]

$$H(s) = \frac{M(\alpha)}{1-\alpha}\frac{s^{\alpha}}{s^{\alpha}+\frac{\alpha}{1-\alpha}} \tag{4.10}$$

By substituting $s = j\omega$, where $\omega \in \mathbb{R}$ in (4.10), the ideal frequency response of ABC fractional-order differential can be obtained. Hence, by assuming initial conditions to be zero, the ideal frequency response of ABC fractional-order differential can be written as

[42, 92]

$$H_d(j\omega) = \frac{M(\alpha)}{1-\alpha}\frac{(j\omega)^\alpha}{(j\omega)^\alpha + \frac{\alpha}{1-\alpha}} \tag{4.11}$$

On the other hand, Figure 4.1a depicts the comparison between conventional power-law kernel and Mittag-Leffler kernel for various values of α. From Figure 4.1a, the singularity of the conventional power-law kernel at $t = 0$ for different values of α can be observed. Therefore, it has been stated that the memory with power-law kernel is not complete and has no beginning [43]. On the contrary, the proposed ABC-FODF is based upon Mittag-Leffler kernel, which eliminates the singularity of power-law kernel and hence, possesses an excellent memory [34, 43]. Figure 4.1b shows the numerical representation of Mittag-Leffler function for various values of fractional-order α.

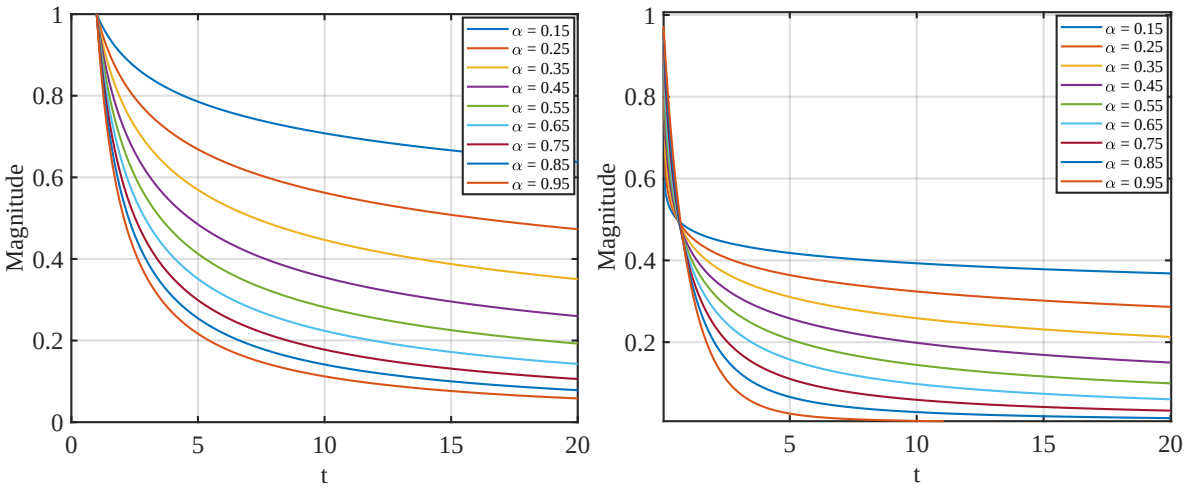


Figure 4.1: Comparison between (a) power-law and (b) Mittag-Leffler kernels for various values of α.

As the domain of FOC expanded, various definitions of fractional-order operators came into existence. Apart from power-law kernel based fractional-order derivatives, Mittag-Leffler kernel based fractional-order derivatives gained significant popularity over the past few years. Therefore, the idea is to explore Mittag-Leffler function as a kernel function for filtering applications. For this, initially the design of ABC-based fractional-order digital filter (FODF) is put forward. Then, the utility of the *proposed* filter is discussed for signal and image processing applications.

4.2 Design of ABC-FODF using Finite Difference Method

Before delving into the design details of ABC-based FODF, the ABC fractional-order differential operator, presented in (4.4) is first numerically approximated by using backward finite difference method (FDM). After that, a closed-form analytical formulation is proposed to find FIR filter coefficients of the proposed ABC-FODF.

Let the signal $p(t)$ is discretized into L equal parts, i.e., $t_0, t_1, ..., t_L$ in the interval $[0, t]$, as described in Figure 3.2. Then, the ABC fractional-order differential operator is numerically approximated (i.e., the interval $[0, t]$ is uniformly partitioned into L sub-intervals $[p_k, p_{k+1}]$ and the length of each sub-interval is $h = (t - 0)/L$) as given by:

$$\begin{aligned} D^{\alpha}_{ABC}p(t) &= \frac{M(\alpha)}{1-\alpha}\int_0^t \frac{d}{d\zeta}p(t-\zeta)E_{\alpha}\left(-\frac{\alpha}{1-\alpha}\zeta^{\alpha}\right)d\zeta \\ &= \frac{M(\alpha)}{1-\alpha}\left[\sum_{k=0}^{L-1}\int_{kh}^{(k+1)h}\frac{d}{d\zeta}p(t-\zeta)E_{\alpha}\left(-\frac{\alpha}{1-\alpha}\zeta^{\alpha}\right)d\zeta\right] \end{aligned} \tag{4.12}$$

By using backward FDM, the first-order differential inside the integral in (4.12) can be expressed as

$$\begin{aligned} &\int_{kh}^{(k+1)h}\frac{d}{d\zeta}p(t-\zeta)E_{\alpha}\left(-\frac{\alpha}{1-\alpha}\zeta^{\alpha}\right)d\zeta \\ &\cong \frac{p(t-kh)-p(t-(k+1)h)}{h}\int_{kh}^{(k+1)h}E_{\alpha}\left(-\frac{\alpha}{1-\alpha}\zeta^{\alpha}\right)d\zeta \\ &= \frac{p_k - p_{k+1}}{h}\int_{kh}^{(k+1)h}E_{\alpha}\left(-\frac{\alpha}{1-\alpha}\zeta^{\alpha}\right)d\zeta \\ &= (p_k - p_{k+1})\left\{(k+1)E_{\alpha,2}\left[-\lambda(k+1)^{\alpha}h^{\alpha}\right] - kE_{\alpha,2}\left[-\lambda k^{\alpha}h^{\alpha}\right]\right\} \end{aligned} \tag{4.13}$$

where $\lambda = \alpha/(1-\alpha)$ and $E_{\alpha,\beta}$ represents a two-parameter Mittag-leffler function defined as $E_{\alpha,\beta}(z) = \sum_{k=0}^{\infty}\frac{z^k}{\Gamma(\alpha k+\beta)}$ with $\beta = 2$. Substituting (4.13) into (4.12), implies

$$\begin{aligned} D^{\alpha}_{ABC}p(t) = \frac{M(\alpha)}{1-\alpha}&\left\{\sum_{k=0}^{L-1}(p_k - p_{k+1})\right. \\ &\left.\left\{(k+1)E_{\alpha,2}\left[-\lambda(k+1)^{\alpha}h^{\alpha}\right] - kE_{\alpha,2}\left[-\lambda k^{\alpha}h^{\alpha}\right]\right\}\right\} \end{aligned} \tag{4.14}$$

Now, the $L+1$ approximated backward differences of ABC fractional-order differential operator can be written as

$$\begin{aligned} D^{\alpha}_{ABC}p(t) = &\frac{M(\alpha)}{1-\alpha}E_{\alpha,2}(-\lambda h^{\alpha})p(t) \\ &+ \frac{M(\alpha)}{1-\alpha}\left\{\sum_{k=1}^{L-1}\left[(k+1)E_{\alpha,2}\left(-\lambda(k+1)^{\alpha}h^{\alpha}\right)\right.\right. \\ &-2kE_{\alpha,2}\left(-\lambda k^{\alpha}h^{\alpha}\right) -(k-1)E_{\alpha,2}\left(-\lambda(k-1)^{\alpha}h^{\alpha}\right)\Big]p(t-kh) \\ &+\left.\left[(L-1)E_{\alpha,2}\left(-\lambda(L-1)^{\alpha}h^{\alpha}\right) -LE_{\alpha,2}\left(-\lambda L^{\alpha}h^{\alpha}\right)\right]p(t-Lh)\right\} \end{aligned} \tag{4.15}$$

Finally, (4.15) can be written in compact form as

$$D^{\alpha}_{ABC}p(t) = \frac{M(\alpha)}{1-\alpha}\sum_{k=0}^{L}c(k)p(t-kh) \tag{4.16}$$

where $c(k)$ represents the *proposed* FIR filter coefficients of ABC-fractional differential-based FODF.

$$c(k) = \begin{cases} E_{\alpha,2}(-\lambda h^{\alpha}) & ;\, k=0 \\ \Big[(k+1)E_{\alpha,2}(-\lambda(k+1)^{\alpha}h^{\alpha}) - 2kE_{\alpha,2}(-\lambda k^{\alpha}h^{\alpha}) & \\ -\,(k-1)E_{\alpha,2}(-\lambda(k-1)^{\alpha}h^{\alpha})\Big] & ;\, 1\le k\le L-1 \\ (L-1)E_{\alpha,2}(-\lambda(L-1)^{\alpha}h^{\alpha}) - LE_{\alpha,2}(-\lambda L^{\alpha}h^{\alpha}) & ;\, k=L \end{cases} \tag{4.17}$$

Therefore, (4.16) represents the *proposed* framework for ABC-FODF, where FIR filter coefficients can be computed from (4.17). Here, the parameter L represents the prescribed filter length.

On the other hand, the FIR filter coefficients of ABR-based FODF can be derived similarly

by utilizing (4.2) and can be presented as

$$c_1(k) = \begin{cases} E_{\alpha,2}(-\lambda h^\alpha) & ; k = 0 \\ \big[(k+1)E_{\alpha,2}(-\lambda(k+1)^\alpha h^\alpha) - 2kE_{\alpha,2}(-\lambda k^\alpha h^\alpha) & \\ -\ (k-1)E_{\alpha,2}(-\lambda(k-1)^\alpha h^\alpha)\big] & ; 1 \le k \le L-1 \\ (L-1)E_{\alpha,2}(-\lambda(L-1)^\alpha h^\alpha) - LE_{\alpha,2}(-\lambda L^\alpha h^\alpha) & \\ +E_{\alpha,2}(-\lambda L^\alpha h^\alpha) & ; k = L \end{cases} \tag{4.18}$$

Hence, (4.17) and (4.18) represent the *proposed* ABC and ABR based FIR coefficients of fractional-order digital filter.

The *proposed* framework, as presented in (4.17) is then utilized to obtain the transfer function of ABC-FODF. By considering $t = n - D$ in (4.16), where D represents the integer delay of the system, the discrete-time ABC fractional-order differential can be written as

$$D^\alpha_{ABC} p[n-D] = \frac{M(\alpha)}{1-\alpha} \sum_{k=0}^{L} c[k]\, p[n-D-kh] \tag{4.19}$$

Now, considering z-transform on both sides of (4.19), yields

$$Y(z) = \left(\frac{M(\alpha)}{1-\alpha} \sum_{k=0}^{L} c[k] z^{-(D+kh)} \right) P(z) \tag{4.20}$$

where $Y(z)$ and $P(z)$ represent the z-transforms of $D^\alpha p[n-D]$ and $p[n]$, respectively. Now, (4.20) can further be written in compact form as

$$Y(z) = H(z)P(z) \tag{4.21}$$

where

$$H(z) = \frac{M(\alpha)}{1-\alpha} \sum_{k=0}^{L} c[k] z^{-(D+kh)} \tag{4.22}$$

Thus, $H(z)$ represents the transfer function of the *proposed* ABC-FODF for fractional orders $0 < \alpha < 1$, whose frequency response $H(e^{j\omega})$ approximates the ideal frequency response $H_d(j\omega)e^{-j\omega D}$ of ABC-FODF, defined in (4.11).

Now, the efficieny of the *proposed* design depends on the parameter h. As already mentioned

in Section 3.3, smaller the value of h, better is the approximation. Therefore, assuming uniform sampling of the signal with sampling period T, h is chosen such that it is non-integer multiple of the sampling interval i.e., $h = dT$, where $0 < d < 1$. However, when h is non-integer multiple of sampling interval T, (4.19) can be written as

$$D^{\alpha}_{ABC} p[n-D] = \frac{M(\alpha)}{1-\alpha} \sum_{k=0}^{L} c[k] p[n-D-kd] \tag{4.23}$$

Now, $p[n-D-kd]$ represents the non-integer delay samples of the signal $p[n]$ that would lie somewhere between the existing samples of $p[n]$. The appropriate values of these samples are hence, found via interpolation, most commonly known as fractional delay interpolation [44]. Taking z-transform on both sides of (4.23) gives

$$Y(z) = \left(\frac{M(\alpha)}{1-\alpha} \sum_{k=0}^{L} c[k] z^{-(D+kd)} \right) P(z) \tag{4.24}$$

where

$$H(z) = \frac{M(\alpha)}{1-\alpha} \sum_{k=0}^{L} c[k] z^{-(D+kd)} \tag{4.25}$$

In general, the z-transform holds only for integral powers of z and hence, if $0 < d < 1$ is utilized, $z^{-(D+kd)}$ cannot be implemented directly. Therefore, to solve this problem, the well-documented design techiques of fractional delay filters needs to be utilized. Two fractional delay interpolation techniques known as Lagrange interpolation and radial basis function interpolation have been utilized here to design the *proposed* ABC-FODF.

4.3 Design of ABC-FODF using Fractional Delay Interpolation

To illustrate the efficiency of the *proposed* design method, the closed-form analytical formulation, presented in (4.25) must be implemented digitally. For this, the delay value $D + kd$, presented in (4.25) can be written as:

$$D + kd = I_k + f_k \tag{4.26}$$

where $I_k = \lfloor D + kd \rfloor$ denotes the integer part (here $\lfloor . \rfloor$ represents the *floor* operator) and $f_k = D + kd - I_k$ denotes the fractional part of the delay value. Substituting (4.26) into (4.25), implies

$$H(z) = \frac{M(\alpha)}{1-\alpha} \sum_{k=0}^{L} c[k] z^{-(I_k + f_k)} \tag{4.27}$$

Now, to implement $z^{-(I_k+f_k)}$, fractional sample delay interpolation techniques such as FIR Lagrange and radial basis function (RBF) are incorporated into the design process to approximate the ideal frequency response of proposed ABC-FODF. The corresponding design details are as follows:

4.3.1 Lagrange Interpolation Based Design of ABC-FODF (L-ABC-FODF)

In this case, Lagrange interpolation has been utilized to design a FIR filter for estimating fractional sample delays $z^{-(I_k+f_k)}$, presented in (4.27) [44].

Using Lagrange interpolation, the fractional sample delays are estimated by

$$z^{-(I_k+f_k)} \cong \sum_{m=0}^{2I_k} q_k(m) z^{-m} \tag{4.28}$$

where $q_k(m)$ represents FIR filter coefficients, defined as

$$q_k(m) = \prod_{l=0, l\neq m}^{2I_k} \frac{I_k + f_k - l}{m - l} \tag{4.29}$$

Now, substituting (4.28) and (4.29) into (4.27), yields

$$\begin{aligned} H_1(z) &= \frac{M(\alpha)}{1-\alpha} \sum_{k=0}^{L} c[k] \sum_{m=0}^{2I_k} q_k(m) z^{-m} \\ &= \frac{M(\alpha)}{1-\alpha} \sum_{k=0}^{L} c[k] \left[\sum_{m=0}^{2I_k} \left(\prod_{l=0, l\neq m}^{2I_k} \frac{I_k + f_k - l}{m - l} \right) z^{-m} \right] \\ &= \frac{M(\alpha)}{1-\alpha} \sum_{m=0}^{2I_k} \left(\sum_{k=0}^{L} c[k] \prod_{l=0, l\neq m}^{2I_k} \frac{I_k + f_k - l}{m - l} \right) z^{-m} \\ &= \frac{M(\alpha)}{1-\alpha} \sum_{m=0}^{2I_k} a_1(m) z^{-m} \end{aligned} \tag{4.30}$$

where

$$a_1(m) = \sum_{k=0}^{L} c[k] \left(\prod_{l=0, l\neq m}^{2I_k} \frac{I_k + f_k - l}{m - l} \right) \tag{4.31}$$

Therefore, $a_1(m)$ represents Lagrange interpolation-based FIR filter coefficients of the *proposed* ABC-FODF (herein, referred to as *L-ABC-FODF*).

Given the input parameters, the procedure to design and implement the *proposed L-ABC-FODF* i.e., $H_1(z)$ is summarized as

1. Design fractional sample delay filter using Lagrange interpolation and compute its filter coefficients $q_k(m)$, as given in (4.29).

2. Determine coefficients $c[k]$ of ABC-FODF by using (4.17).

3. Use (4.31) to calculate filter coefficients $a_1(m)$ of the *proposed L-ABC-FODF*.

4. The transfer function of the *proposed L-ABC-FODF* is thus written as $H_1(z) = \frac{M(\alpha)}{1-\alpha} \sum_{m=0}^{2I_k} a_1(m) z^{-m}$, as depicted in (4.30).

To investigate the effectiveness of the *proposed* method, a design example is presented. The design parameters are chosen to be $L = 300, d = 0.2, D = 20$, and $\alpha = 0.5$, where d is a small positive number that represents fractional sample delay, α is the fractional-order, and D represents the delay of the system.

Figure 4.2 illustrates the magnitude and phase response approximations (solid lines) of the *proposed L-ABC-FODF* for $\alpha = 0.5$. The dashed lines represent the ideal response. From Figure 4.2, it can be observed that the *proposed L-ABC-FODF* significantly approximates the ideal frequency response of ABC-FODF. The mean square error (MSE) corresponding to magnitude response approximation is found to be 0.0005, for $\alpha = 0.5$ over the frequency band of interest $\omega = [0, 0.9\pi]$. The MSE for various fractional orders is further reported in the next section. Therefore, from the results depicted in Figure 4.2, it can be observed that the *proposed L-ABC-FODF* design can effectively approximate the ideal frequency response of ABC fractional-order differential operator.

Furthermore, it can be seen that the performance of the *proposed* closed-form expression, presented in (4.30), significantly depends upon the choice of fractional delay parameter d. Therefore, the relation between MSE and the parameter d by using the same design

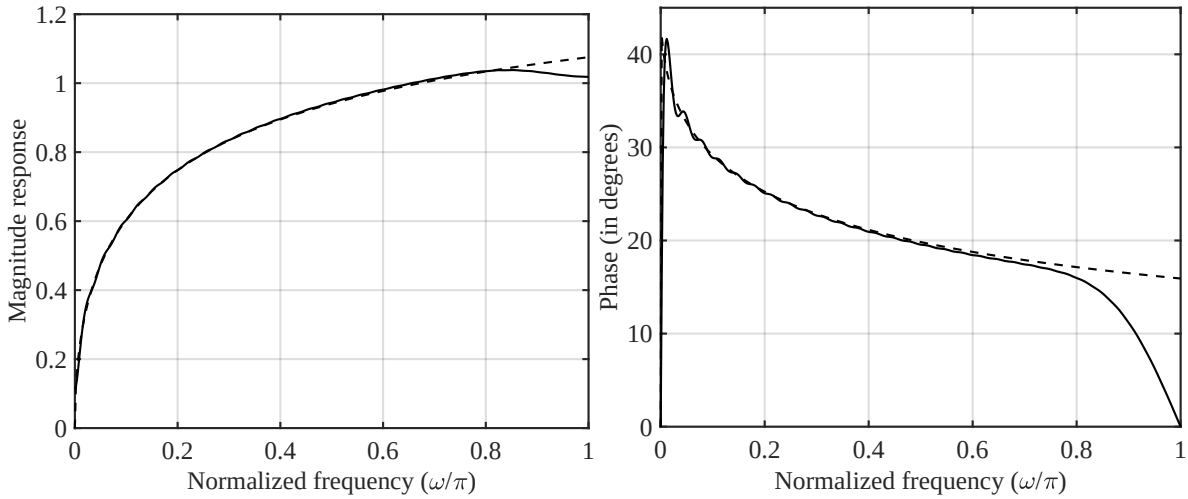


Figure 4.2: (a) Magnitude and (b) Phase response approximations of the *proposed L-ABC-FODF* for $\alpha = 0.5$; the dashed lines represent the ideal magnitude and phase response respectively.

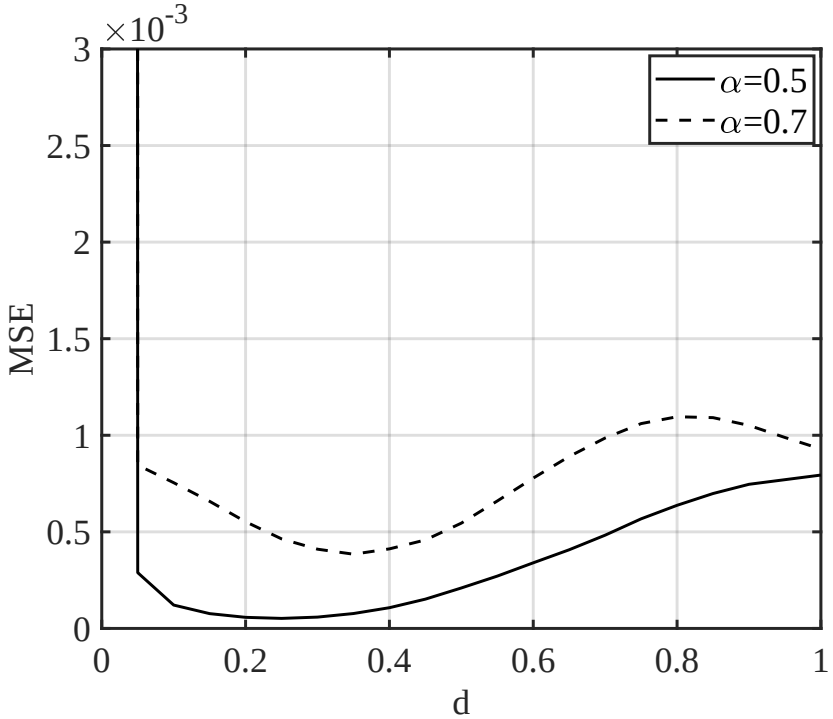


Figure 4.3: MSE of the *proposed L-ABC-FODF* design for parameter d and $\alpha = 0.5, 0.7$.

parameters, as illustrated above, is studied. Figure 4.3 illustrates the MSE error plot for $d \in [0.01, 1]$, and for $\alpha = 0.5, 0.7$. From the results, it has been observed that MSE is an increasing function of d when $d > 0.4$. However, it has the lowest MSE in the range $d \in [0.1, 0.4]$ for both $\alpha = 0.5$ and 0.7. Therefore, the parameter d is chosen to be $d = 0.2$ for the *proposed L-ABC-FODF* method.

4.3.2 Radial Basis Function Based Design of ABC-FODF (RBF-ABC-FODF)

RBF interpolation has found utmost importance in a variety of applications such as, multivariate interpolation, neural networks, control of nonlinear systems, etc. [54]. Therefore, the performance of RBF interpolation in estimating fractional sample delays, presented in (4.19) for the design of proposed ABC-FODF is tested here in this section. The fractional delay samples using RBF interpolation can be approximated by

$$p[n - I_k - f_k] = \sum_{m=0}^{N} g(m, I_k + f_k) p[n - m] \tag{4.32}$$

where the weights $g(m, I_k + f_k)$ are computed by using a method, presented in [54]. Taking z-transform on both sides of (4.32), implies

$$z^{-(I_k+f_k)} = \sum_{m=0}^{N} g(m, I_k + f_k) z^{-m} \tag{4.33}$$

Now, substituting (4.33) into (4.27), the transfer function of the *proposed* RBF-based ABC-FODF (herein, referred to as *RBF-ABC-FODF*) can be represented as

$$\begin{aligned} H_2(z) &= \frac{M(\alpha)}{1-\alpha} \sum_{k=0}^{L} c[k] \sum_{m=0}^{N} g(m, I_k + f_k) z^{-m} \\ &= \frac{M(\alpha)}{1-\alpha} \sum_{m=0}^{N} \left(\sum_{k=0}^{L} c[k] g(m, I_k + f_k) \right) z^{-m} \\ &= \frac{M(\alpha)}{1-\alpha} \sum_{m=0}^{N} a_2(m) z^{-m} \end{aligned} \tag{4.34}$$

where

$$a_2(m) = \sum_{k=0}^{L} c[k] g(m, I_k + f_k) \tag{4.35}$$

Thus, $a_2(m)$ represents the *proposed* RBF-interpolation based FIR filter coefficients of ABC-FODF. The procedure to design and implement the proposed *RBF-ABC-FODF*, i.e., $H_2(z)$ is summarized as follows:

1. Compute coefficients of fractional delay filter by utilizing RBF interpolation, as presented by $g(m, I_k + f_k)$ in (4.33).

2. Calculate coefficients $c[k]$ of ABC-FODF by using (4.17).

3. Use (4.35) to calculate filter coefficients $a_2(m)$ of the *proposed RBF-ABC-FODF*.

4. The transfer function of the *proposed RBF-ABC-FODF* is thus expressed as $H_2(z) = \frac{M(\alpha)}{1-\alpha} \sum_{m=0}^{N} a_2(m) z^{-m}$, as depicted in (4.34).

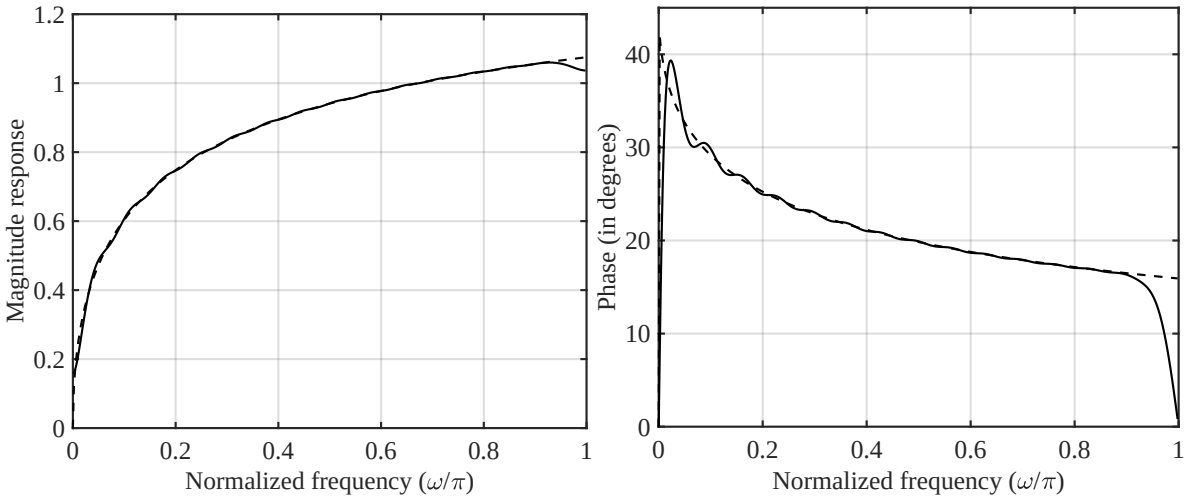


Figure 4.4: (a) Magnitude and (b) Phase response approximations of the *proposed RBF-ABC-FODF* for $\alpha = 0.5$; the dashed lines represent the ideal magnitude and phase response.

As an illustration, a design example is presented to reveal the effectiveness of the *proposed* method. The design parameters are chosen to be $N = 300, L = 620, D = 30, d = 0.05$, and $\alpha = 0.5$. Figure 4.4 represents the magnitude and phase response approximations of the *proposed RBF-ABC-FODF* for $\alpha = 0.5$. From the figure, it can be seen that the approximation of the *proposed RBF-ABC-FODF* design has significantly improved the design accuracy in the high-frequency region as compared to the previously *proposed* design method.

Comparing Figure 4.4 with Figure 4.2, it can be seen that the *proposed RBF-ABC-FODF* outperforms the *L-ABC-FODF* design. The corresponding MSE for magnitude response approximation of the *proposed RBF-ABC-FODF* is reported to be 0.00009 for $\alpha = 0.5$. This implies that the *proposed* RBF-ABC-FODF has shown a percentage improvement of 84.25% over that of *L-ABC-FODF*, for $\alpha = 0.5$, over the frequency band of interest $B = [0, 0.9\pi]$.

Furthermore, to illustrate the effect on the *proposed* approach for the choice of d, the relation between MSE and the parameter d is depicted in Figure 4.5, for $\alpha = 0.5, 0.7$. The design

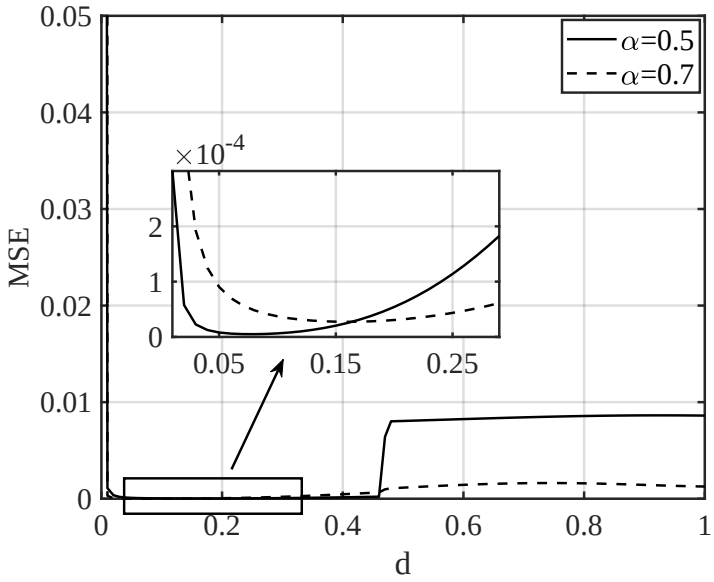


Figure 4.5: MSE of the proposed *RBF-ABC-FODF* design for parameter d and $\alpha = 0.5, 0.7$.

parameters are selected as $N = 300, L = 620$, and $D = 30$. Figure 4.5 depicts the error plot for $d = [0.01, 1]$ and $\alpha = 0.5, 0.7$. From the figure, it is observed that MSE increases when $d > 0.4$. However, the MSE is lowest in the range $d = [0.05, 0.3]$. Therefore, for the *proposed* design approach the fractional delay parameter d is chosen to be 0.05, for $\alpha = 0.5$.

Thus, it can be concluded that the *proposed* design methods significantly approximate the ideal frequency response of ABC fractional-order differential operator. Furthermore, it can be stated that the closed-form analytical formulations (4.30) and (4.34) are obtained for the design of ABC-based FIR FODF, which makes *the proposed design approach simple and easy to implement.*

4.4 Simulation Results

In the preceding section, various design methods for implementing Mittag-Leffler kernel-based ABC-FODF have been presented. The quantitative comparison between *proposed* design techniques such as *L-ABC-FODF* and *RBF-ABC-FODF* is illustrated in this section for various fractional orders.

The *proposed* methods are compared by considering three statistical measures, namely, MSE, normalized root mean square error (NRMSE), and correlation coefficient known as R-value (RV). The smaller values of MSE and NRMSE indicate better performance of the design method. Furthermore, the closer the RV is to one, the better is the design method.

The detailed statistical comparison in terms of MSE, NRMSE, RV is reported in Table 4.1 for the design methods *proposed* in Section 4.3 at various fractional orders. Using the same design complexity, as illustrated in Section 4.3 for both design methods, the fractional sample delay parameter d is chosen to be 0.2 for *L-ABC-FODF* and 0.05 for *RBF-ABC-FODF*. Furthermore, various fractional-orders in the range of $\alpha = [0.1, 1]$, with a step size of 0.1, are chosen to validate the robustness of the proposed design methods.

Table 4.1: Quantitative comparison among magnitude response approximations of the *proposed L-ABC-FODF* and *RBF-ABC-FODF* design methods for various fractional orders.

α	MSE		NRMSE		RV	
	L-ABC-FODF	*RBF-ABC-FODF*	*L-ABC-FODF*	*RBF-ABC-FODF*	*L-ABC-FODF*	*RBF-ABC-FODF*
0.1	0.00160	0.001600	**0.0435**	0.04400	**0.9981**	0.9980
0.2	**0.00091**	0.001000	**0.0343**	0.03630	**0.9987**	0.9986
0.3	**0.00039**	0.000530	**0.0223**	0.02610	**0.9994**	0.9992
0.4	**0.00013**	0.000200	**0.0123**	0.01650	**0.9998**	0.9996
0.5	0.00057	**0.000900**	**0.0072**	0.00900	**0.9999**	**0.9999**
0.6	0.00011	**0.000028**	0.0085	**0.00430**	0.9999	**1**
0.7	0.00055	**0.000077**	0.0152	**0.00180**	0.9996	**1**
0.8	0.00290	**0.000027**	0.0274	**0.00084**	0.9983	**1**
0.9	0.01250	**0.000046**	0.0453	**0.00087**	0.9945	**1**
1	0.02990	**0.000014**	0.0611	**0.00130**	0.9888	**1**

Bold *designates lowest values of MSE and NRMSE; and highest values of RV (max=1) for corresponding design methods.*

Comparing *L-ABC-FODF* and *RBF-ABC-FODF* from the results depicted in Table 4.1, it can be seen that the *RBF-ABC-FODF* outperforms *L-ABC-FODF* for fractional orders $\alpha > 0.4$. However, for $\alpha = 0.4$, difference between two methods is negligible. Therefore, it can be said that out of the two design methods *proposed* in this chapter, the performance of *RBF-ABC-FODF* is superior to *L-ABC-FODF* design, for $\alpha > 0.4$.

Furthermore, the comparison between *L-ABC-FODF* and *RBF-ABC-FODF* design methods is illustrated in Figure 4.6, where MSE values are plotted for various fractional orders. From the figure, it can be depicted that the *proposed RBF-ABC-FODF* performs better than the *L-ABC-FODF* method presented in this chapter.

Therefore, this chapter establishes a novel and efficient framework for the design of non-singular Mittag-Leffler kernel-based Atangana-Baleanu-Caputo fractional-order digital filter and opens up a new paradigm of research for various signal filtering and image processing applications.

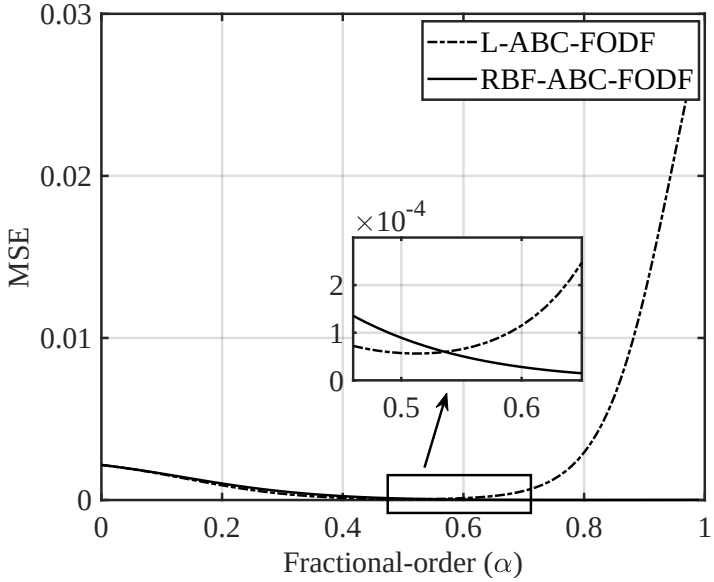


Figure 4.6: MSE comparison between magnitude response approximation of proposed *L-ABC-FODF* and *RBF-ABC-FODF* for various fractional orders.

4.5 Applications of Proposed ABC-FODF

Since the design of ABC-based fractional-order digital filter (ABC-FODF) has been proposed for the first time, it is important to discuss the applicability of the proposed method for different signal and image processing applications. To establish the horizon of the application area of the proposed fractional-order digital filter (i.e., ABC-FODF), one-dimensional (1D) and two-dimensional (2D) applications of the proposed FODF are considered. One is the detection of R-peaks in electrocardiogram (ECG) signals, and other is digital image sharpening. The details are described as follows:

4.5.1 R-peak Detection in Electrocardiogram (ECG) Signals

ECG is an electrical signal which is used to evaluate cardiac activity in humans and is composed of several waveforms of different bandwidths such as P-wave, QRS complex and T-wave [57]. In real-world scenarios, various artifacts such as power line interference (PLI), baseline wander (BW), instrumentation noise, etc. severely affect the morphology of electrocardiogram signals, and hence makes its analysis a complex task [57, 58]. Therefore, for the detection of several cardiovascular diseases, denoising of ECG signals as well as accurate delineation of R-peaks become a crucial job to investigate.

Hence, for accurate detection of R-peaks in ECG signals and for establishing the usefulness of ABC fractional-order derivative in 1-D signal processing applications, the proposed RBF-ABC-FODF is utilized for detecting R-peaks in ECG signals. The ECG signals are extracted from MIT-BIH arrhythmia database [114] and the proposed algorithm for the detection of R-peaks in ECG signal is presented in Figure 4.7. The step-by-step procedure for denoising ECG signals and detecting R-peaks, as presented in Figure 4.7, is described as follows:

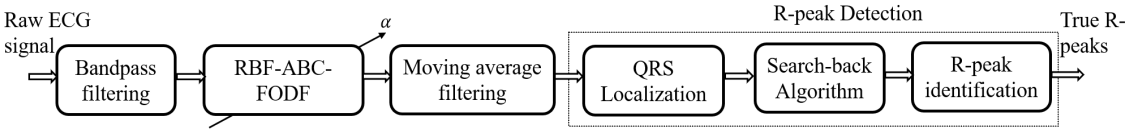


Figure 4.7: Block diagram for ECG R-peak detection using the proposed ABC-FODF.

Step 1 Bandpass filtering: Since the raw ECG signal extracted from MIT-BIH arrhythmia database has various high-frequency and low-frequency noises, bandpass filtering (BPF) is utilized as a first step to eliminate these noises, whose impulse response is expressed as $h_{bp}(n) = sin(n\omega_{c2})/\pi n - sin(n\omega_{c1})/\pi n$, where ω_{c1} and ω_{c2} represent the cut-off frequencies of the ideal BPF [15]. The lower and higher cut-off frequencies of BPF are set to be 5 Hz and 25 Hz, respectively (as frequency band of QRS complex is 5–25 Hz [57, 58]). This significantly eliminates the artifacts such as power line interference (50 Hz) and baseline wander (nearly 1 Hz) from the raw ECG signal.

Step 2 Proposed *RBF-ABC-FODF*: The *proposed RBF-ABC-FODF* is utilized as the next step in order to enhance the high-frequency QRS complex in ECG signals and to suppress the low-frequency P and T waves. This is because fractional differential operators are known to enhance high-frequency contents of the signal effectively [115]. The fractional-order of the *proposed RBF-ABC-FODF* is chosen in the range $[0.5, 0.9]$ as lower fractional orders $[0.1, 0.4]$ are less efficient in attenuating low-frequency noises in comparison to $\alpha \in [0.5, 0.9]$. Also, as $\alpha \rightarrow 1$, the peculiar details tend to be lost from the signal.

Step 3 Moving average filtering: During step 2, some high frequency noise also gets amplified. This is due to the fact that these noises contain same spectral contents as that of QRS complex. Hence, to lower the effect of these high-frequency noises, a moving window average filter [58] is employed after differentiating ECG signal.

Step 4 R-peak detection: Thereafter, QRS localization is performed to identify R-peaks in ECG signal. This is obtained by applying thresholding technique [116] on the smooth ECG waveform obtained after step 3. The threshold is set in the range of (0.3-0.4) times the maximum amplitude of output waveform. Furthermore, if multiple peaks are detected within the time frame of 100 ms, one of them is rejected to avoid the detection of pseudo R-peaks [115]. This is done because the QRS sample has a duration of 100 ms or 36 samples. On the other hand, if no peak is detected within the time frame of the QRS complex, the search back algorithm is applied using a new threshold which is 0.25 times the previous threshold, until all the missed peaks are localized. The details regarding QRS localization, search back algorithm, and identification of R-peaks can be referred from [116, 117].

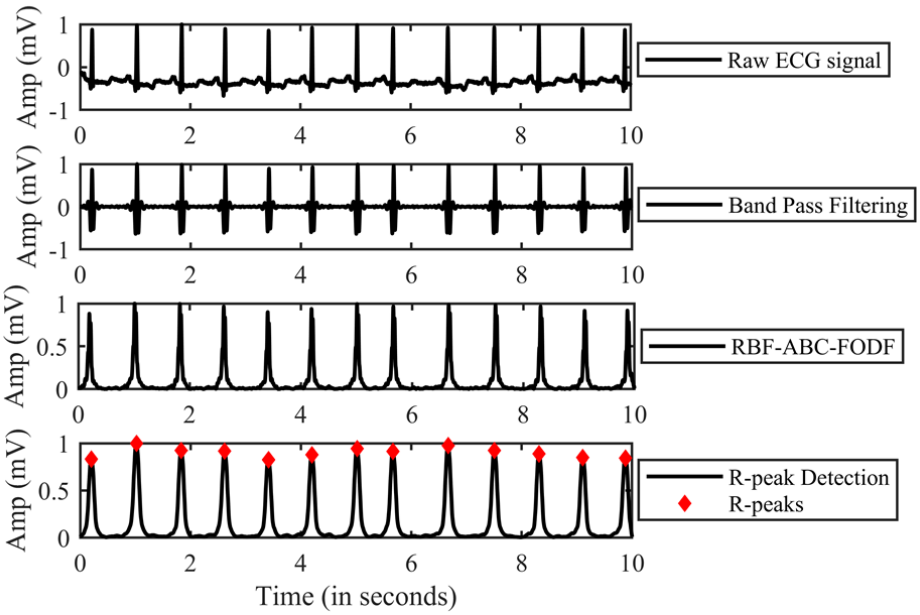


Figure 4.8: ECG R-peak detection for segment of record no. 100 from MIT-BIH arrhythmia database.

To show the performance of the *proposed RBF-ABC-FODF*, a small segment of MIT-BIH arrhythmia record no. 100 from 0 to 10 s is selected, as shown in Figure 4.8. Figure 4.8 il-

lustrates the output waveforms after each step of the *proposed* algorithm, presented in Figure 4.7.

From Figure 4.8, it can be observed that the band-pass filter retains the 5–25 Hz frequency components and the *proposed RBF-ABC-FODF* significantly enhances the QRS complex of the ECG waveform. Finally, the average smoothing filter smoothens the ECG signal for accurate R-peak detection, as shown in Figure 4.8.

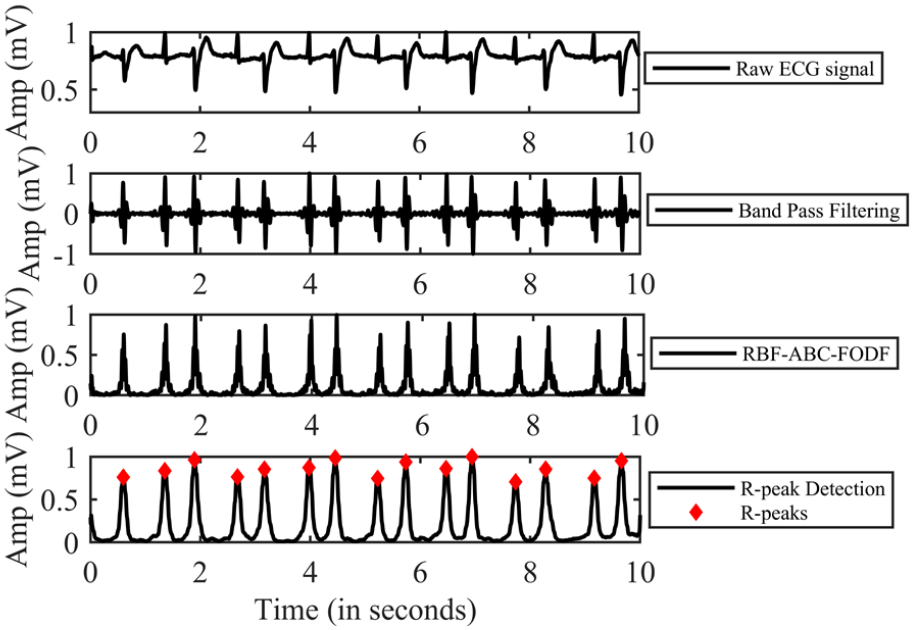


Figure 4.9: ECG R-peak detection for segment of record no. 200 from MIT-BIH arrhythmia database.

Furthermore, a segment of record no. 200 consisting of various bursts of muscle noise is considered for evaluating the performance of the *proposed* method. Figure 4.9 illustrates the raw ECG signal extracted from MIT-BIH arrhythmia database. The output waveforms after bandpass filtering, *proposed* RBF-ABC-FODF and smoothing operations are depicted in Figure 4.9. From the results described in Figure 4.9, the effectiveness of *proposed* method in accurate delineation of R-peaks and suppression of the burst noise can be observed.

Therefore, it can be stated that preliminary results of the *proposed RBF-ABC-FODF* in R-peak detection of ECG signals are encouraging and hence, can be beneficial for ECG signal processing applications.

4.5.1.1 Performance Metrics

To evaluate the performance of the *proposed RBF-ABC-FODF* for R-peak detection, the well-known quantitative parameters such as positive predictivity (PP), sensitivity (S) and error rate (ER) are taken into account, which are defined as

$$\begin{aligned} S &= \frac{TP}{FN+TP} \\ PP &= \frac{TP}{FP+TP} \\ ER &= \frac{FN+FP}{Total\ number\ of\ beats\ detected} \end{aligned} \tag{4.36}$$

where TP denotes true positives (i.e., true/actual R-peak detected as R-peak), FP denotes false positives (i.e., false peak detected as R-peak by the algorithm), and FN denotes false negatives (i.e., missed true R-peak).

4.5.1.2 Simulation Results and Comparative Analysis

To illustrate the effectiveness of the *proposed* method against well-established state-of-the-art methods, five full-length records from MIT-BIH arrhythmia database (i.e., record no. 100, 101, 102, 103 and 104) are taken into consideration. All the ECG records of MIT-BIH database are of half-an-hour fragments and are digitized at 360 Hz with 11-bit resolution over a 10-mV range [114]. The quantitative analysis in terms of various parameters such as, positive predictivity (PP), sensitivity (S) and error rate (ER) is done and the comparison is made against various methods *proposed* in the literature.

Table 4.2: Experimental results of the *proposed RBF-ABC-FODF* based R-peak detection algorithm

Record No.	Actual beats	Detected beats	TP	FP	FN	S (%)	PP (%)	ER (%)
100	2273	2273	2273	0	0	100	100	0
101	1865	1868	1865	3	0	100	99.83	0.16
102	2187	2187	2187	0	0	100	100	0
103	2084	2084	2084	0	0	100	100	0
104	2229	2232	2229	3	2	99.91	99.86	0.22
Total	**10638**	**10644**	**10638**	**6**	**2**	**99.98**	**99.94**	**0.075**

Table 4.2 represents the experimental results achieved by the *proposed* method in terms of S,

PP, and ER. The results of these five full-length records indicated that the *proposed* method has attained S = 99.98%, PP = 99.94% and ER = 0.075%, which validates the usefulness of the *proposed* method in ECG R-peak detection.

Table 4.3: Comparative analysis of the *proposed* method against well-established methods from the literature

Methods	Sensitivity (%)	Positive Predictivity (%)	Error Rate (%)
Manikandan [118]	99.98	99.83	0.188
Nayak *et al.* [119]	99.96	99.85	0.178
Sharma *et al.* [117]	99.92	99.88	0.188
Kaur *et al.* [58]	99.80	99.92	0.272
Nayak *et al.* [120]	99.99	99.80	0.206
Proposed method	**99.98**	**99.94**	**0.075**

Furthermore, for comparative analysis, the quantitative parameters (i.e., S, PP, and ER) for the methods presented in [58, 117, 118, 119, 120] have been calculated explicitly for same five records (i.e., record no. 100, 101, 102, 103, and 104). The experimental results from the literature [58, 117, 118, 119, 120] for the same records are presented in Table 4.3 and comparison has been made with the results achieved by the *proposed* method.

From the results tabulated in Table 4.3, it can be observed that the *proposed* method outperforms conventional algorithms in terms of increased sensitivity and decreased error rate. Therefore, it can be said that the *proposed* method exhibits potential applications in the field of ECG R-peak detection.

To illustrate the superiority of the *proposed* fractional-order filter against that of the conventional first-order differentiation filter, Figure 4.10 is presented which indicates the magnitude response of the filter for $\alpha = 0.6$ and $\alpha = 1$. In the normalized frequency, with sampling frequency 360 Hz, the 5–25 Hz frequency band falls in the range of 0.027–0.138 [120]. From Figure 4.10, it can be seen that in this frequency range, the magnitude response of the *proposed* fractional-order filter for $\alpha = 0.6$ is better as compared to integer-order counterpart, i.e., when $\alpha = 1$.

This improved differentiation capability of the *proposed* filter assists in generating feature signal efficiently for wide duration and tiny amplitude QRS complexes and hence, improves the R-peak detection rate. This is because the slope information generated by tiny and wide duration QRS complexes is so small that most of the derivative-based methods fail to detect

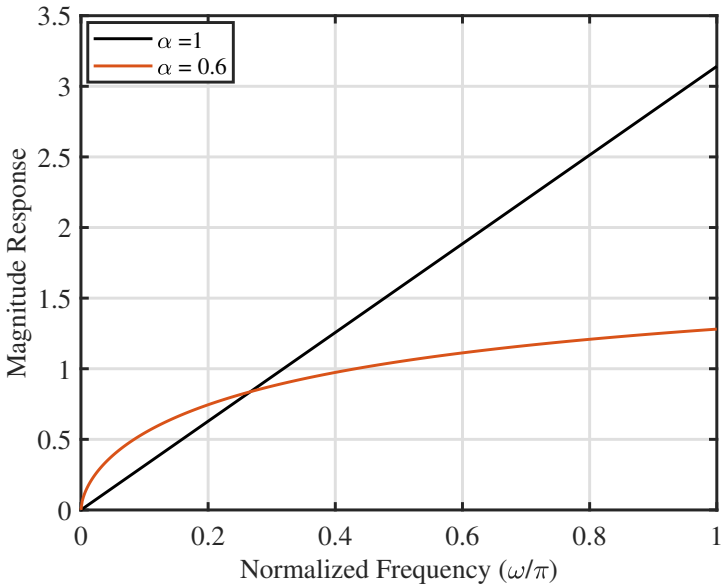


Figure 4.10: Magnitude response of the *proposed* filter at $\alpha = 0.6$ and $\alpha = 1$.

these beats [121, 115]. This is where the advantage of the *proposed* fractional-order filter exists. The better performance of the *proposed* method can also be validated from the values of error rate obtained by the *proposed* method, presented in Table 4.3, which indicates that *proposed* method has less number of FNs and FPs in comparison to the conventional methods.

Hence, from the preliminary study conducted, it can be said that the *proposed* approach has the ability to accurately delineate R-peaks in ECG signals and therefore, could be beneficial for various biomedical applications. Furthermore, it can be stated that the results of the *proposed* RBF-ABC-FODF methods in the detection of R-peak in ECG signals are encouraging and hence opens up a new paradigm of research for ECG signal processing applications.

4.5.2 Digital Image Sharpening

Digital image sharpening is extensively applied in various applications such as medical imaging, industrial inspection, electronic printing and autonomous guidance in military systems, to name a few [63]. In these applications, digital image sharpening is used to enhance the texture details of the images. This is because during image acquisition, some of the details in the images get blurred. Therefore, spatial filtering methods based on second-order derivative (*a.k.a.* Laplacian) have been used in the literature from the past few decades to enhance these

details. A conventional Laplacian image sharpening technique is given by [63, 122]:

$$p(x, y) = u(x, y) - \lambda \nabla^2 u(x, y) \tag{4.37}$$

where $u(x, y)$ and $p(x, y)$ represents the input image and sharpened image, respectively. The sharpening strength is controlled by the parameter λ. The Laplacian operator ∇^2 presented in (4.37) is given by:

$$\nabla^2 u(x, y) = \frac{\partial^2 u(x, y)}{\partial x^2} + \frac{\partial^2 u(x, y)}{\partial y^2} \tag{4.38}$$

Substituting (4.38) in (4.37), implies:

$$p(x, y) = u(x, y) - \lambda \left(\frac{\partial^2 u(x, y)}{\partial x^2} + \frac{\partial^2 u(x, y)}{\partial y^2} \right) \tag{4.39}$$

Taking Fourier transform on both sides of (4.39), the frequency domain expression can be obtained as:

$$\begin{aligned} P(\omega_1, \omega_2) &= U(\omega_1, \omega_2) - \lambda((j\omega_1)^2 U(\omega_1, \omega_2) + (j\omega_2)^2 U(\omega_1, \omega_2)) \\ &= (1 + \lambda\omega_1^2 + \lambda\omega_2^2) U(\omega_1, \omega_2) \end{aligned} \tag{4.40}$$

where $P(\omega_1, \omega_2)$ and $U(\omega_1, \omega_2)$ represent the Fourier transforms of sharpened and input images, respectively. The block diagram for implementing image sharpening is presented in Figure 4.11, where the input is $u(x, y)$ and the output is sharpened image $p(x, y)$. The transfer function for implementing the 2-D filter on an input image $u(x, y)$ can be written as:

$$S(z_1, z_2) = 1 - \lambda H(z_1) - \lambda H(z_2) \tag{4.41}$$

where $p(x, y)$ denotes the sharpened image. The variables z_1 and z_2 denote the variables of 2-D z-transform and $H(z)$ represents the conventional Laplacian filter.

However, in recent years, fractional-order filters have been proved to outperform the conventional spatial filters in extracting peculiar details from the data [64]. Therefore, this section discusses the applicability of the *proposed* RBF-ABC-FODF method for digital image sharpening and compares it with widely popular power-law kernel based Grünwald-Letnikov (GL) and Riemann-Liouville (RL) fractional digital image sharpening approaches [13].

In order to provide flexibility in digital image sharpening, the second-order derivative in

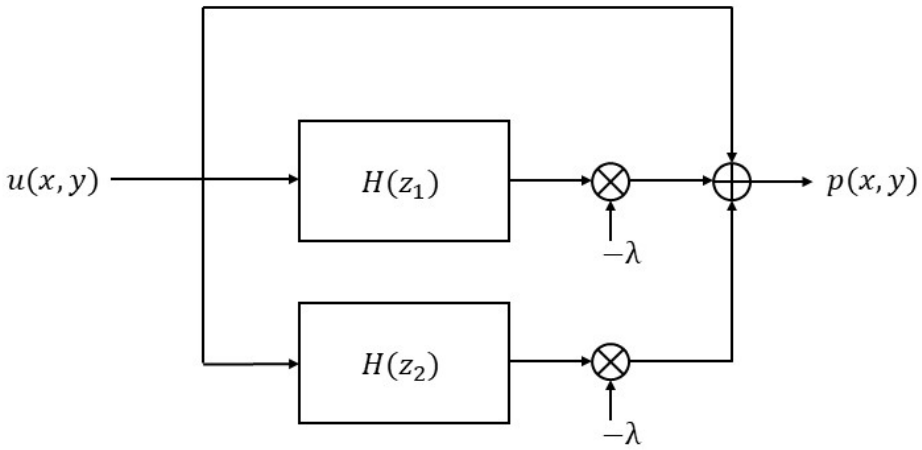


Figure 4.11: Digital image sharpening approach.

(4.39) is replaced with fractional-order derivative, which is implemented by utilizing the *proposed* RBF-ABC-FODF method. With this generalization, the sharpness of the image can be adjusted by varying the fractional-order α of the *proposed* filter. Here, the conventional Laplacian filter $H(z)$ is replaced by the *proposed* RBF-ABC-FODF, defined in (4.34). The *proposed* $H(z)$ is constrained to have exact zero gain at $\omega = 0$ so that the brightness of background should remain same. Therefore, the coefficient $a_2(0)$ of the *proposed* RBF-ABC-FODF is modified to $-\sum_{m=1}^{N} a_2(m)$ [54]. Finally, the method presented in Figure 4.11 is implemented by utilizing the *proposed* filter in order to attain the sharpened images.

4.5.2.1 Performance Metrics

To evaluate the performance of the *proposed* image sharpening method, several quantitative parameters such as information entropy (IE), measure of enhancement (EME), and gray level co-occurrence matrix (GLCM) parameters are considered.

Information Entropy (IE) is the measure of randomness which characterizes the texture of an image. It is utilized to compute the extent of information in the enhanced image. It is calculated over entire image by utilizing histogram of the image. The IE of an image is computed as [123]:

$$IE = -\sum_{i=0}^{255} \rho_i log_2 \rho_i \tag{4.42}$$

where ρ_i represents the probability associated with gray level i. IE is approximately inversely correlated with uniformity and hence, images that have larger number of gray levels will have

larger entropy.

Measure of enhancement (EME) is one of the basic contrast measures in which the image is divided into blocks [123]. It is highly sensitive to randomness, texture, uniform background, and noise in the image. Higher value of EME indicates more image contrast and information clarity. It is computed by classifying the image $p(x, y)$ into b_1 and b_2 blocks of $W_{b,c}(x, y)$ as:

$$EME = \frac{1}{b_1 b_2} \sum_{c=1}^{b1} \sum_{b=1}^{b2} \left[20 log_e \left(\frac{p_{max;b,c}}{p_{min;b,c}} \right) \right] \tag{4.43}$$

where $p_{max;b,c}$ and $p_{min;b,c}$ indicate the maximum and minimum of the image inside the block $W_{b,c}(x, y)$.

Gray level co-occurrence matrix (GLCM) characterizes the texture of an image by calculating how often pairs of pixel with specific values and in a specified spatial relationship occur in an image. GLCM matrix is created by varying the distance among the pixels in a pair at various rotation offsets such as 0°, 45°, 90° and 135°. After creating a GLCM matrix, several statistical measures from this matrix are extracted [13].

The GLCM parameters utilized for image enhancement are contrast, correlation, energy and homogeneity [13]. The contrast gives the measure of intensity contrast between the pixel of interest and its neighbor. It basically provides the amount of local gray level variation in an image. A higher value of contrast indicates presence of edges and enhancement of texture details in an image [124].

Correlation provides the measure by which one pixel is correlated to its neighboring pixel. Lower value of correlation between the original and enhanced image indicates more clarity. Energy depicts the uniformity of the gray level distribution of an image. Images with larger number of gray levels have lesser energy or uniformity. Homogeneity, on the other hand, indicates the closeness in the spatial distribution (i.e., smoothness) of gray levels over the entire image. Homogeneity is approximately inversely correlated with contrast, i.e., if contrast is large, homogeneity is small [124].

4.5.2.2 Simulation Results and Comparative Analysis

In order to illustrate the performance of the *proposed* image sharpening approach, several standard images are taken into account. Furthermore, two cases are considered - one based

upon grayscale images and other based on color images.

Case 1: In this case, the standard grayscale "Lena" image is taken to perform gray-level image sharpening.

For a grayscale image, the data is a 2-D matrix with size $(K_1 \times K_2)$. The *proposed* image sharpening approach is then applied to the image directly. Figure 4.12 depicts the sharpened grayscale images obtained by utilizing the *proposed* method. The fractional-orders are chosen as $\alpha = 0.3, 0.6, 0.9$.

(a) $\alpha = 0$ **(b)** $\alpha = 0.3$ **(c)** $\alpha = 0.6$ **(d)** $\alpha = 0.9$

Figure 4.12: (a) Grayscale "Lena" image (512×512) ; Sharpened grayscale images at (b) $\alpha = 0.3$ (c) $\alpha = 0.6$ (d) $\alpha = 0.9$ by utilizing the *proposed* approach.

From the results depicted in Figure 4.12, it can be observed that the contrast of the sharpened image when $\alpha = 0.3, 0.6$ and 0.9, looks better than that of the original image. Therefore, it can be said that the *proposed* method has the potential to significantly enhance the image quality in order to represent fine details present in the image. Additionally, it can be observed that sharpness of the image can be adjusted by varying the fractional order in a relevant manner.

For quantitative analysis, the assessment parameters known as IE, EME, contrast, energy, correlation and homogeneity are considered at various fractional orders. Table 4.4 demonstrates the assessment parameters obtained by utilizing the *proposed* method at various fractional orders with respect to the original image.

From the results depicted in Table 4.4, it can be seen that when $\alpha = 0$, the original image is obtained. Furthermore, as the fractional order α is increased, IE, EME and contrast of the image increases significantly. However, when $\alpha \rightarrow 1$, it can be seen that smaller values of EME, entropy and contrast are attained in comparison to the *proposed* method at $\alpha = 0.9$. This is because fractional differential based methods could non-linearly preserve the low-

Table 4.4: Image assessment parameters for grayscale "Lena" image at various fractional orders

α	IE	EME	Contrast	Energy	Correlation	Homogeneity
0	7.1914	8.3158	0.3799	0.1890	0.9128	0.8753
0.1	7.1988	8.5206	0.3997	0.1858	0.9091	0.8712
0.2	7.2267	9.0225	0.4580	0.1741	0.8975	0.8578
0.3	7.2594	9.8093	0.5559	0.1577	0.8784	0.8368
0.4	7.3053	10.9858	0.7071	0.1366	0.8504	0.8063
0.5	7.3635	12.2495	0.9465	0.1119	0.8089	0.7646
0.6	7.4401	13.9891	1.3225	0.0868	0.7497	0.7134
0.7	7.5256	16.0970	1.8664	0.0665	0.6736	0.6603
0.8	7.5986	17.6838	2.5255	0.0521	0.5940	0.6142
0.9	**7.6389**	**18.7821**	**3.0314**	**0.0447**	**0.5394**	**0.5861**
1	7.5938	17.5898	2.5836	0.0522	0.5950	0.6194

frequency features in smooth areas to the furthest degree, and non-linearly enhance texture details in areas where gray scale does not change evidently [13].

Additionally, it can be observed that as α increases, the correlation of GLCM matrix decreases and hence, texture details become more visible. However, as $\alpha \rightarrow 1$, the correlation is increased. This indicates that the image becomes white or black in certain areas and hence, the texture details are less visible. Similar pattern can be analysed from the values obtained for energy and homogeneity, i.e., with the increase in α, the texture details in different sections of the image change more dramatically and hence, there is less uniformity and homogeneity. However, as $\alpha \rightarrow 1$, the energy and homogeneity is increased. This indicates that when $\alpha \rightarrow 1$, there is decrease in the contrast of the image. Therefore, it can be said that when $\alpha \in [0.8, 0.9]$, the texture details of the image are more clearer and hence, better enhancement can be attained by using the *proposed* fractional differential approach as compared to the integer-order filter.

Case 2: In this case, the colored "Lena" image, which contains RGB (red, green, blue) components is considered, to illustrate the usefulness of the *proposed* designed approach. Since the input image contains three components, each component is sharpened individually by utilizing the same procedure as described in Algorithm 1. Then, the three sharpened RGB components are combined to obtain the final sharpened image.

Figure 4.13 represents the original image and the sharpened images obtained by using *pro-*

posed RBF-ABC-FODF for $\alpha = 0.3, 0.6$ and 0.9, respectively. Comparing Figure 4.13a with that of Figure 4.13b-4.13d, it can be observed that the original image has been enhanced significantly. Furthermore, it can be seen that the quality of the sharpened image depends upon the choice of fractional-order α. Therefore, unlike conventional spatial filters that provide fixed quality enhancement, the *proposed* fractional-order method provides flexibility based upon α to adjust the sharpness of the image.

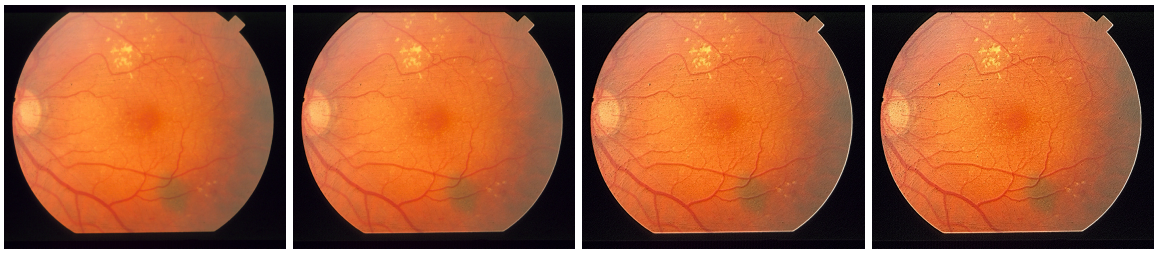
(a) Original Image **(b)** $\alpha = 0.3$ **(c)** $\alpha = 0.6$ **(d)** $\alpha = 0.9$

Figure 4.13: (a) Colored "Lena" image ($512 \times 512 \times 3$) ; Sharpened colored images at (b) $\alpha = 0.3$ (c) $\alpha = 0.6$ (d) $\alpha = 0.9$ by utilizing the *proposed* approach.

On the other hand, the *proposed* approach is also implemented on a medical image, extracted from STARE (structured analysis of the retina) [125] dataset. Here, the main aim is to show that by applying the *proposed* method, significant texture details from medical image can be extracted. Hence, enhanced image information could prove to be advantageous for the diagnosis of serious disorders with better criteria.

Figure 4.14a represent the proliferative diabetic retinopathy image, which specifies one of the serious issues endured by diabetic patients. This ailment leads to loss of vision because the tiny blood vessels of the retina gets destroyed. Hence, for proper diagnosis, it is highly important to detect these changes at early stages itself.

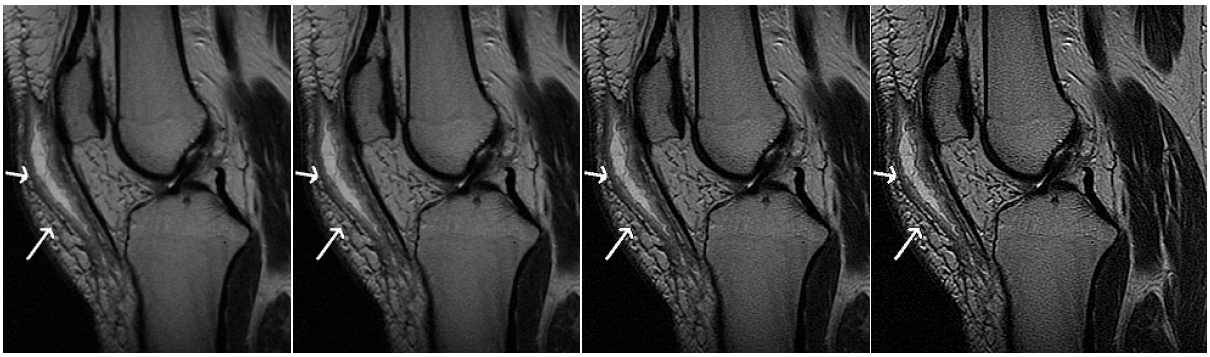
Image sharpening can assist in early detection of diabetes due to the fact that it has the ability to highlight small blood vessels in the images. Figure 4.14b-4.14d present the original and sharpened diabetic retinopath images obtained by the *proposed* method at $\alpha = 0.3, 0.6$, and 0.9, respectively. The efficiency of the *proposed* method can be noticed from the fact that small blood vessels of the image are enhanced considerably and hence, makes the method beneficial for medical image sharpening and analysis.



(a) Original Image **(b)** $\alpha = 0.3$ **(c)** $\alpha = 0.6$ **(d)** $\alpha = 0.9$

Figure 4.14: Diabetic retinopathy (a) Original image ($605 \times 700 \times 3$); Sharpened images at (b) $\alpha = 0.3$ (c) $\alpha = 0.6$ (d) $\alpha = 0.9$ by utilizing the *proposed* method.

Furthermore, a medical "MRI" image is taken to perform gray-level image sharpening. Figure 4.15 depicts the sharpened grayscale images obtained by utilizing the *proposed* method. The fractional-orders are chosen as $\alpha = 0.3, 0.6, 0.9$. From the results depicted in Figure 4.15, it can be seen that certain details of the original image have been enhanced significantly at various fractional orders. Therefore, it can be said that the *proposed* method has significantly enhanced the image quality to represent fine details present in the image.



(a) MRI Image **(b)** $\alpha = 0.3$ **(c)** $\alpha = 0.6$ **(d)** $\alpha = 0.9$

Figure 4.15: MRI image (a) Original image (286×276); Sharpened images at (b) $\alpha = 0.3$ (c) $\alpha = 0.6$ (d) $\alpha = 0.9$ by utilizing the *proposed* method.

Hence, from the simulation study conducted, it can be visualised that the *proposed* method has the capability to provide better image enhancement for various types of images. Hence, it exhibits a great potential to enhance tiny details in images and hence, can be employed for image sharpening when it gets difficult to retain the peculiar details from the original image itself.

Furthermore, in order to illustrate the robustness of the *proposed* method, several images

with different characteristics from Set12 [126] dataset are taken into account. Set12 dataset contains a set of 12 standard images that are widely utilized for image processing applications. Figure 4.16 represents the randomly selected test images utilized for simulation study.

(a) House (512 × 512) **(b)** Lena (512 × 512) **(c)** Boat (512 × 512) **(d)** Pirate (512 × 512) **(e)** Room (512 × 512)

Figure 4.16: Test images used for simulation from Set12 dataset.

Furthermore, the results attained by the *proposed* method are then compared against well-established conventional methods, which include conventional Laplacian mask [63], GL-based fractional differential mask [13], RL-based fractional differential mask [13]. The comparison is done on the basis of several quantitative metrics such as EME, entropy and GLCM parameters considered so far.

Table 4.5 presents the quantitative results attained by various methods on different images. From the results tabulated in Table 4.5, it can be seen that the *proposed* method attains the highest values of IE, EME and contrast and minimum values for energy, correlation and homogeneity in comparison to state-of-the-art methods. This is because the *proposed* fractional-order derivative approach is able to provide better image enhancement due to its ability to enhance and preserve texture details in various parts of the image. On the other hand, it can be seen that RL fractional-order mask provides better quantitative results in comparison to GL fractional-order mask which is in line with the analytical and simulation study presented in Chapter 3. Furthermore, it can be observed that Mittag-Leffler kernel based AB fractional-order mask provides better enhancement metrics in comparison to both GL and RL fractional-order masks. The difference between the conventional GL, RL and recently introduced AB fractional-order derivatives lies in the kernel function being utilized. However, all these fractional-order based methods possess flexibility in terms of fractional order α and hence, provide significant advantages over integer order counterparts.

The superiority of the fractional-order mask lies in the fact that it can non-linearly preserve

Table 4.5: Comparative analysis of the *proposed* image sharpening method against well-established methods from the literature

Images	Metrics/ Methods	Entropy	EME	Contrast	Energy	Correlation	Homogeneity
House	Original	*6.4971*	*5.3243*	*0.1863*	*0.2029*	*0.9497*	*0.9251*
	Laplacian	7.2813	14.0090	0.8148	0.0857	0.8478	0.7803
	GL	7.2890	13.9866	0.8402	0.0828	0.8416	0.7702
	RL	7.3261	14.2750	0.8993	0.0802	0.8341	0.7618
	AB	**7.3274**	**14.3365**	**0.9049**	**0.0799**	**0.8332**	**0.7608**
Lena	Original	*7.1914*	*8.3158*	*0.3799*	*0.1890*	*0.9128*	*0.8753*
	Laplacian	7.5938	17.5898	2.5836	0.0522	0.5950	0.6194
	GL	7.6175	18.2545	2.7623	0.0485	0.5680	0.6008
	RL	7.6378	18.7014	3.0201	0.0449	0.5406	0.5867
	AB	**7.6389**	**18.7821**	**3.0314**	**0.0447**	**0.5394**	**0.5861**
Boat	Original	*7.4474*	*6.5333*	*0.2293*	*0.1316*	*0.9502*	*0.9056*
	Laplacian	7.6945	14.5283	1.1709	0.0611	0.7979	0.7347
	GL	7.6965	14.8981	1.2297	0.0582	0.7874	0.7218
	RL	7.7073	15.1489	1.3419	0.0551	0.7718	0.7091
	AB	**7.7080**	**15.1627**	**1.3464**	**0.0550**	**0.7711**	**0.7087**
Pirate	Original	*7.1926*	*7.8367*	*0.2757*	*0.1304*	*0.9377*	*0.8947*
	Laplacian	7.6581	16.7210	1.8200	0.0588	0.7082	0.7159
	GL	7.6579	16.8127	1.9097	0.0573	0.6930	0.7067
	RL	7.6689	17.0056	2.0705	0.0541	0.6750	0.6947
	AB	**7.6694**	**17.0664**	**2.0772**	**0.0540**	**0.6742**	**0.6943**
Room	Original	*7.0572*	*9.9101*	*0.3207*	*0.1346*	*0.9196*	*0.8791*
	Laplacian	7.6036	15.4261	1.9119	0.0525	0.6731	0.6871
	GL	7.6044	15.6286	1.9306	0.0508	0.6652	0.6764
	RL	7.6088	15.7258	2.1052	0.0476	0.6456	0.6631
	AB	**7.6092**	**15.7520**	**2.1121**	**0.0475**	**0.6449**	**0.6625**

the texture features in smooth regions of the image. This is in contrast to first-order filters such as Sobel operator and second-order filters such as Gauss-Laplace operator, where texture features in smooth regions of an image may be greatly attenuating. This is because the gray scale values of image's smooth area does not change intensively and hence, the integer order differential of constant is zero. Therefore, for preserving and enhancing texture features in smooth areas, fractional order filters proves to be superior to conventional

integer-order differential based methods.

Therefore, from the simulation study conducted, it can be observed that the enhanced images obtained by using the *proposed* ABC-FODF contains more information and clarity than that of the original image. Also, it can be seen that the quality of enhancement can be adjusted by varying the order α appropriately. Hence, fractional-order filtering provides an extra degree of freedom in terms of fractional-order α, which could be beneficial for signal and image processing applications.

4.6 Summary

This chapter proposes the design of novel fractional-order digital filter (FODF) based upon Mittag-Leffler kernel based Atangana-Baleanu (AB) fractional-order differential operator. In the pursuit of establishing a closed-form analytical expression for the design of AB-based FIR FODF, the definition of Atangana–Baleanu fractional differential in Caputo sense (ABC) is considered. Furthermore, fractional-sample delay interpolation techniques such as Lagrange interpolation and radial basis function (RBF) interpolation are considered to improve the design accuracy of the *proposed* method. Several design examples are investigated, discussed, and compared to reveal the efficiency of the *proposed* design methods (i.e., L-ABC-FODF and RBF-ABC-FODF). From the analytical and simulation analysis, it has been observed that the *proposed* RBF-ABC-FODF outperforms the L-ABC-FODF method *proposed* in this chapter.

Furthermore, in order to discuss the applicability of the *proposed* method, the preliminary applications in signal and image processing are considered. The 1-D application of the *proposed* method focuses on the accurate delineation of R-peaks in ECG signals and 2-D application emphasis on sharpening of digital images. From the simulation studies conducted, it has been proved that the *proposed* method has the ability to accurately delineate R-peaks in ECG signals and also, possesses significant advantages over well-established state-of-the-art methods from the literature. Furthermore, the *proposed* image sharpening method not only enhances the quality of digital images significantly, but also provides the flexibility in enhancement of images in contrast to conventional integer order counterparts. Also, the comparison among GL, RL and the *proposed* AB fractional-order masks is done which indicates the superiority of the Mittag-Leffler kernel in contrast to conventional power-law kernel as a

filter function.

After establishing the preliminary applications of the *proposed* fractional-order differential filter, the next aim is to discuss and compare the conventional and AB-based fractional-order integral operators for image denoising applications. Therefore, as a major contribution of the research work carried out in this thesis, a novel generalized fractional integral and fractional differential based adaptive algorithm is introduced in the subsequent chapters. Also, a rigorous comparison among various fractional-order operators is done for image denoising applications.

Chapter 5

Generalization of Fractional Integrals: From Riemann-Liouville to Atangana-Baleanu for Image Denoising

AFTER ESTABLISHING CLOSED-FORM ANALYTICAL FORMULATIONS for the design of fractional-order digital filters based on various fractional-order differential operators, the next aim is to study various definitions of fractional-order integrals for image denoising applications. Therefore, a generalized fractional integral (GFI) mask is proposed in this chapter which incorporates both Riemann-Liouville and Atangana-Baleanu fractional-order integral operators. The proposed GFI mask provide a more flexible and appropriate tool for image denoising applications and hence, avoids the construction of different masks for each definition separately. Furthermore, a GFI-based adaptive mask (GFI-AM) is put forward to select the fractional-order of the masks based upon local features of an image. Then, the performance of various fractional-order integral operators is validated for images affected by Gaussian noise and salt and pepper noise.

5.1 Concept of Image Denoising: Two-Dimensional Signal Processing

The task of image denoising (i.e., removing noise and recovering the original image) is extremely important because quality of a digital image gets inevitably degraded during the process of acquisition and transmission [63]. Among various types of noises, for instance,

additive, multiplicative, impluse, and periodic, the impulse noise, which originates due to sensor vibrations is the most prominent one. One such type of impluse noise is salt and pepper (SP) noise, which greatly impacts the performance of various image processing techniques [63].

Several traditional methods which include median filter (MF) [127] and its variants such as non-local mean filter [128], weighted mean filter [129], adaptive switching based filters [130, 131], based on pixel density filter (BPDF) [104], different applied median filter (DAMF) [100] etc., have been applied to deal with SP noise effectively. Although these methods can remove noise efficiently and possess several advantages, the preservation of edge and texture details of an image still remains a challenging issue. On the other hand, most of the traditional methods adopt integer-order integrals directly or indirectly during the construction of a denoising model and hence, loses texture and edge details of the image.

On the contrary, fractional-order filters have been proved to be advantagenous because of their ability to preserve the information while denoising the image [64]. The flexibility in the choice of fractional-order parameter in fractional-order calculus (FOC) promises to yield better denoising effect and preserve the edge and texture information of the image simultaneously. In recent years, the applications of FOC in the domain of digital image processing have gained a significant importance due to its ability to preserve the texture and edge information of image in the process of image denoising [13, 122]. Pu *et al.* [13] introduced six fractional differential masks based upon GL and RL fractional-order operators for enhancement of texture-rich images. In addition, various other masks such as improved GL (IGL) [132], Alexander fractional differential (AFD) mask and Alexander fractional integral (AFI) mask [133] have been introduced in rich literature for image enhancement and image denoising applications [134, 135].

In the literature, various fractional-order masks corresponding to different fractional-order operators exist but in isolation i.e., for each definition of fractional-order operator, fractional-order masks have been constructed separately [13]. A generalized fractional K-operator has been utilized in [136] and [137], which reduces to conventional RL and Caputo fractional-order operators in a special case. However, the research on the potential of recently introduced fractional-order operators for image denoising applications is still limited [91, 96]. Therefore, this chapter puts forward a framework for generalized fractional integral (GFI)

mask for the first time which incorporates conventional (i.e., power-law kernel based) as well as recently introduced (i.e., Mittag-Leffler kernel based) definitions of fractional-order integral operators.

Furthermore, most of the traditional fractional image denoising masks (for example, GL [13], RL [13], IGL [132], AFD [133], AFI [133], fractional K-operator [136], and recently proposed ABC mask [91]) employ a fixed fractional-order to deal with the noise present in the images. However, the characteristics of an image varies locally. Hence, it is difficult to obtain better image denoising effect by using a fixed fractional-order to process noisy pixels in distinct regions. This is due to the fact that if higher fractional-order is chosen, the texture details are lost. On the other hand, if lower fractional-order is chosen, denoising performance is degraded.

Therefore, after establishing a framework for the design of GFI mask, an adaptive function for the selection of fractional-order is presented. Then, the performance of the proposed GFI and GFI-based adaptive mask (GFI-AM) is tested for images affected by Gaussian noise and SP noise, respectively. Additionally, the effectiveness of various fractional-order operators is tested and compared against each other for image denoising applications.

5.2 Analytical Foundation of Generalized Fractional Integral (GFI) Mask

To develop an analytical foundation of generalized fractional integral mask, the definitions of fractional-order integral operators are reviewed briefly and are presented as defined below:

Definition 5.2.1: The fractional-order integral associated with power-law kernel based RL fractional-order differential operator (denoted by I^{α}_{RL}) is stated as [8]:

$$I^{\alpha}_{RL}\, p(t) = \frac{1}{\Gamma(-\alpha)} \int_0^t p(\zeta)(t-\zeta)^{(-\alpha-1)} d\zeta \tag{5.1}$$

where $\Gamma(x)$ represents the Gamma function of x and the fractional-order $\alpha < 0$. The above equation can also be written as [13]:

$$I^{\alpha}_{RL}\, p(t) = \frac{1}{\Gamma(\alpha)} \int_0^t p(\zeta)(t-\zeta)^{(\alpha-1)} d\zeta \tag{5.2}$$

for $\alpha > 0$.

Definition 5.2.2: The fractional-order integral corresponding to Mittag-Leffler kernel based Atangana-Baleanu fractional-order differential operator (denoted by I_{AB}^{α}) is stated as [31]:

$$I_{AB}^{\alpha}\, p(t) = \frac{1-\alpha}{M(\alpha)} p(t) + \frac{\alpha}{\Gamma(\alpha) M(\alpha)} \int_0^t p(\zeta)(t-\zeta)^{(\alpha-1)} d\zeta \tag{5.3}$$

where $M(\alpha)$ is the normalization constant and is defined as $M(\alpha) = 1 - \alpha + \frac{\alpha}{\Gamma(\alpha)}$ and fractional integral order $\alpha > 0$.

From (5.2) and (5.3), the AB fractional-order integral can be written in terms of RL fractional-order integral as follows [40]:

$$I_{AB}^{\alpha}\, p(t) = \frac{1-\alpha}{M(\alpha)} p(t) + \frac{\alpha}{M(\alpha)} I_{RL}^{\alpha}\, p(t) \tag{5.4}$$

Hence, the *proposed* generalized fractional-order integral incorporating both RL and AB fractional-order integral operators can be put forward as:

$$I_{GFI}^{\alpha}\, p(t) = c_1 p(t) + c_2 \int_0^t p(\zeta)(t-\zeta)^{(\alpha-1)} d\zeta \tag{5.5}$$

where the subscript GFI represents the generalized definition of fractional integral operator. The corresponding constants c_1 and c_2 for GFI incorporating RL (GFI-RL) and AB (GFI-AB) fractional-order integrals are listed in Table 5.1.

Table 5.1: Fractional-order integral operators and their corresponding constants

Fractional Integral Operators	c_1	c_2
Reimann-Liouville (GFI-RL)	0	$\frac{1}{\Gamma(\alpha)}$
Atangana-Baleanu (GFI-AB)	$\frac{1-\alpha}{M(\alpha)}$	$\frac{\alpha}{M(\alpha)\Gamma(\alpha)}$

Now, to design GFI mask, (5.5) is numerically approximated by first dividing the interval $[0, t]$ into L equal parts. Then, it is analytically formulated to yield the coefficients of the proposed GFI mask, the details of which are as follows:

$$\begin{aligned} I_{GFI}^{\alpha}\, p(t) &= c_1 p(t) + c_2 \int_0^t p(t-\zeta)\zeta^{\alpha-1} d\zeta \\ &= c_1 p(t) + c_2 \sum_{k=0}^{L-1} \int_{kt/L}^{(k+1)t/L} p(t-\zeta)\zeta^{\alpha-1} d\zeta \end{aligned} \tag{5.6}$$

Approximating p in (5.6) by average value on each sub-interval, the integral inside (5.6) can be represented as

$$\begin{aligned}\int_{kt/L}^{(k+1)t/L} p(t-\zeta)\zeta^{\alpha-1}d\zeta &\cong \frac{p(t-kt/L)+p(t-(k+1)t/L)}{2}\int_{kt/L}^{(k+1)t/L}\zeta^{\alpha-1}d\zeta \\ &= \frac{p_k+p_{k+1}}{2}\int_{kt/L}^{(k+1)t/L}\zeta^{\alpha-1}d\zeta \\ &= \frac{p_k+p_{k+1}}{2\alpha}\left[(k+1)^\alpha\left(\frac{t}{L}\right)^\alpha - k^\alpha\left(\frac{t}{L}\right)^\alpha\right] \\ &= \frac{p_k+p_{k+1}}{2\alpha}\left(\frac{t}{L}\right)^\alpha\left[(k+1)^\alpha - k^\alpha\right]\end{aligned} \tag{5.7}$$

Substituting (5.7) in (5.6), implies

$$\begin{aligned}I^\alpha_{GFI}\,p(t) &= c_1p_0 + c_2\sum_{k=0}^{L-1}\frac{p_k+p_{k+1}}{2\alpha}\left(\frac{t}{L}\right)^\alpha\left[(k+1)^\alpha-k^\alpha\right] \\ &= c_1p_0 + \frac{c_2h^\alpha}{2\alpha}\left[\sum_{k=0}^{L-1}(p_k+p_{k+1})\left((k+1)^\alpha-k^\alpha\right)\right]\end{aligned} \tag{5.8}$$

where $t/L = h$. The expression for the *proposed* GFI mask can thus be attained as

$$I^\alpha_{GFI}\,p(t) = \left(c_1+\frac{c_2h^\alpha}{2\alpha}\right)p_0 + \frac{c_2h^\alpha}{2\alpha}\left[\sum_{k=1}^{L-1}\left[\left((k+1)^\alpha-k^\alpha\right)p_k\right] + \left(L^\alpha-(L-1)^\alpha\right)p_L\right] \tag{5.9}$$

Hence, from (5.9), the coefficients d_k of *proposed* GFI mask can be described by

$$d_k = \begin{cases} c_1+\frac{c_2h^\alpha}{2\alpha} & ;\, k=0 \\ \frac{c_2h^\alpha}{2\alpha}\left[(k+1)^\alpha-k^\alpha\right] & ;\, 1\le k\le L-1 \\ \frac{c_2h^\alpha}{2\alpha}\left[L^\alpha-(L-1)^\alpha\right] & ;\, k=L \end{cases} \tag{5.10}$$

where $h = 1$ (As in digital images, the length of image is finite and minimum step-size is one pixel. Therefore, L can be large enough so that $t/L = 1$ [13]) and the corresponding constants c_1 and c_2 for GFI-RL and GFI-AB fractional-order integrals, respectively are listed in Table 5.1. Therefore, (5.10) represents the coefficients of the *proposed* GFI mask which are further utilized for the purpose of image denoising. The *proposed* GFI mask is applied in all possible eight directions by constructing a single superimposed mask, as presented in Table 5.2 for the task of image denoising. Furthermore, the *proposed* mask is normalized by dividing each coefficient by the summation of all the coefficients of the mask.

Table 5.2: Superposition of GFI mask in all possible eight directions

d_L	0	$\cdots$	0	d_L	0	$\cdots$	0	d_L
0	$\ddots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$.^{\cdot}$	$.^{\cdot}$	0
$\vdots$	$\ddots$	d_2	0	d_2	0	d_2	$.^{\cdot}$	$\vdots$
0	$\cdots$	0	d_1	d_1	d_1	0	$\cdots$	0
d_L	$\cdots$	d_2	d_1	$8d_0$	d_1	d_2	$\cdots$	d_L
0	$\cdots$	0	d_1	d_1	d_1	0	$\cdots$	0
$\vdots$	$.^{\cdot}$	d_2	0	d_2	0	d_2	$\ddots$	$\vdots$
0	$.^{\cdot}$	$.^{\cdot}$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\ddots$	0
d_L	0	$\cdots$	0	d_L	0	$\cdots$	0	d_L

5.3 Comparative Analysis between GFI-RL and GFI-AB Fractional-Order Masks

After establishing the design of *proposed* GFI mask, the performance of the *proposed* mask is demonstrated by taking various digital images into account. These images are contaminated with various noises such as, Gaussian noise and salt and pepper noise. Then, the denoising performance of various fractional-order integral operators is compared.

Therefore, this section is segregated into two parts: In the first part, the performance of the *proposed* GFI mask is tested on images affected by Gaussian noise. Then, the comparsion among GL, RL, and AB fractional-order integral operators is done. In the second part, the performance of each of these fractional-order integral operators is tested for images affected by salt and pepper noise. The experiments are performed on images extracted from a dataset named Set12 [126] which contains a set of 12 images. Apart from visual interpretation, different performance metrics are considered to quantify the performance of the *proposed* method as defined below.

5.3.1 Performance Metrics

For quantitative analysis, the performance metrics utilized are peak signal to noise ratio (PSNR) and structural similarity index measure (SSIM) of the image.

PSNR is defined as the ratio of maximum power of the signal to the noise power [63] and is measured in decibel (dB). It is defined in terms of mean square error (MSE) as:

$$PSNR = 20\log_{10}\left(\frac{2^8-1}{\sqrt{MSE}}\right) \tag{5.11}$$

where

$$MSE = \frac{1}{AB}\sum_{i=0}^{A-1}\sum_{j=0}^{B-1}[p(i,j)-q(i,j)]^2 \tag{5.12}$$

here $p(i,j)$ and $q(i,j)$ represent the pixels of original and denoised images, respectively. For an image denoising algorithm, the higher value of PSNR indicates better performance.

SSIM is utilized to measure similarity between two images [138]. The closer the SSIM is to one, similar the contents of image are. It is defined as

$$SSIM(p,q) = [l(p,q)][c(p,q)][s(p,q)] \tag{5.13}$$

where l, c, and s represent the luminance, contrast, and structure, respectively [138] defined as:

$$l(p,q) = \frac{2\mu_p\mu_q + K_1}{\mu_p^2 + \mu_q^2 + K_1}$$
$$c(p,q) = \frac{2\sigma_p\sigma_q + K_2}{\sigma_p^2 + \sigma_q^2 + K_2}$$
$$s(p,q) = \frac{\sigma_{pq} + K_3}{\sigma_p\sigma_q + K3}$$

here $\mu_p, \mu_q, \sigma_p, \sigma_q, \sigma_{pq}$ represent local means, standard deviations and cross-covariance for images $p(i,j)$ and $q(i,j)$, respectively, whereas K_1, K_2, and K_3 represent the constants [138].

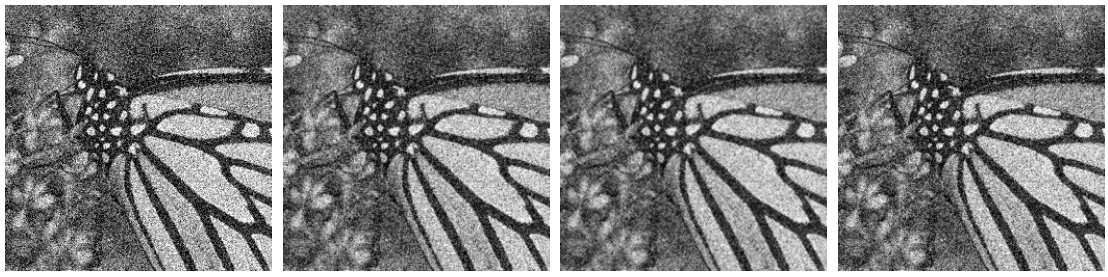
5.3.2 Simulation Results Considering Gaussian Noise

To illustrate the effectiveness of the *proposed* GFI mask, standard images from Set12 dataset [126] are considered. These images are then affected by Gaussian noise, a type of noise which affects every pixel of an image by random values from a normal distribution [63, 101]. The *proposed* GFI mask is applied directly to the images affected by Gaussian noise and the performance of the *proposed* mask is analysed. It should be noted that the *proposed* GFI

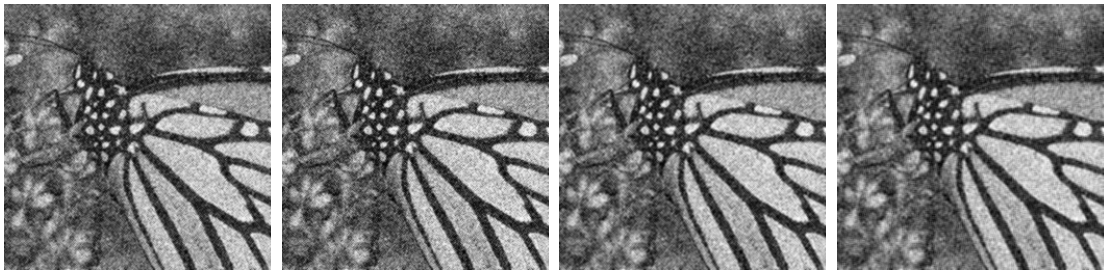
mask exhibits an advantage that by utilizing only one mask, the performance of both RL and AB fractional-order operators can be analyzed.

Furthermore, performance of the *proposed* GFI mask is compared against well-established methods such as, Gaussian filter (GF) [63], fractional K-operator (K-opr) [136], improved GL (IGL) [132], Alexander fractional integral (AFI) [133], and conventional GL mask [13]. The results are compared on the basis of quantitative parameters, which include PSNR and SSIM.

Figure 5.1 illustrates the performance of various methods on an image (extracted from Set12 dataset [126]) corrupted by Gaussian noise with zero mean and 0.03 variance. Figure 5.1a depicts the noisy image. The corresponding denoised images attained by utilizing various state-of-the-art methods are presented in Figures 5.1b-5.1f. The results of the proposed GFI-based RL and AB masks are presented in Figures 5.1g-5.1h, respectively. The PSNR and SSIM values attained by these methods are also mentioned along with corresponding denoised images.



(a) Noisy Image PSNR=15.646 SSIM=0.2829 | **(b)** GF [63] PSNR=19.337 SSIM=0.3984 | **(c)** K-opr [136] PSNR=20.759 SSIM=0.4501 | **(d)** IGL [132] PSNR=18.368 SSIM=0.3644

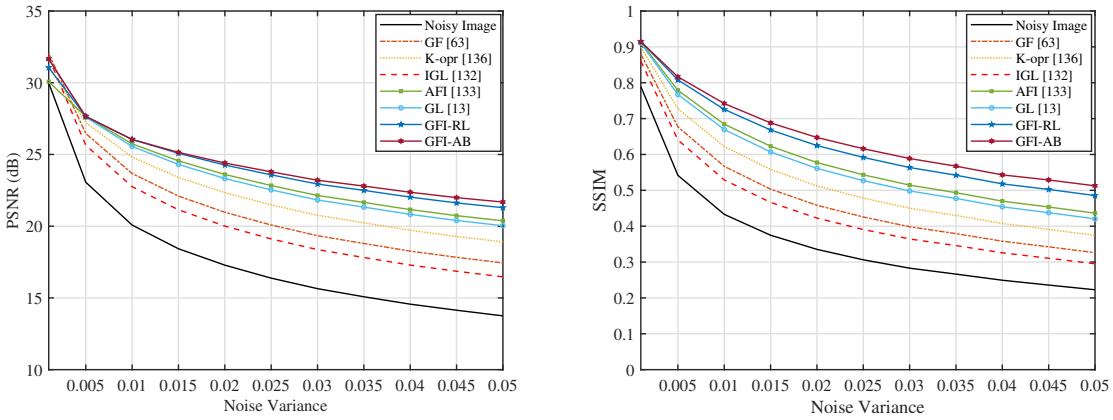


(e) AFI [133] PSNR=22.140 SSIM=0.5145 | **(f)** GL [13] PSNR=21.826 SSIM=0.4983 | **(g)** GFI-RL PSNR=22.937 SSIM=0.5636 | **(h) GFI-AB** **PSNR=23.202** **SSIM=0.5886**

Figure 5.1: Denoising performance of various methods on Image 5 of Set12 dataset corrupted by 0.03 variance Gaussian noise.

From quantitative results obtained, it can be depicted that the values of PSNR and SSIM attained by the *proposed* GFI mask incorporating AB fractional operator (GFI-AB) are higher than that of GL and GFI mask incorporating RL fractional operator (GFI-RL). Furthermore, it can be observed that the *proposed* GFI-AB mask outperforms the conventional Gaussian filter (GF) [63] and other conventional methods based on fractional-order operators in terms of highest values of PSNR and SSIM.

To illustrate the effectiveness of the *proposed* GFI mask for an image corrupted by Gaussian noise at various levels of noise variance, noise variance versus PSNR, and SSIM plots are presented in Figure 5.2. From the results, it can be depicted that the values of PSNR and SSIM attained by the *proposed* GFI-AB mask are consistently higher than the other discussed methods even when the noise variance is increased.



(a) **(b)**

Figure 5.2: Noise variance versus (a) PSNR and (b) SSIM plots for comparison among various methods on Image 5 of Set12 dataset corrupted by Gaussian noise.

Additionally, to show the applicability of the *proposed* method on images with different characteristics, all images of Set12 dataset are considered. Table 5.3 depicts the PSNR and SSIM values attained by various methods for all images of Set12 dataset that have been corrupted by zero mean Gaussian noise of 0.03 variance.

From the values obtained, it can be observed that the *proposed* GFI-AB method attains highest values of PSNR and SSIM for all images and hence, it can be said that the *proposed* GFI-AM mask outperforms the conventional fractional-order masks significantly.

Therefore, from the simulation study conducted on images affected by various levels of Gaussian noise as well as on images of different characteristics, it can be concluded that

the *proposed* GFI masks incorporating AB fractional operator outperforms the conventional GL and RL-based fractional-order masks and hence, exhibits a great potential for image denoising applications.

Table 5.3: Denoising performance of various methods on Set12 dataset corrupted by zero mean Gaussian noise of variance 0.03.

Image No.	Parameters	Noisy Image	GF [63]	K-opr [136]	IGL [132]	AFI [133]	GL [13]	GFI-RL	GFI-AB
1	PSNR	15.858	19.446	20.725	18.523	21.867	21.621	22.411	**22.498**
	SSIM	0.2116	0.2965	0.3346	0.2704	0.3861	0.3730	0.4256	**0.4401**
2	PSNR	15.558	19.348	20.913	18.337	22.556	22.162	23.661	**24.260**
	SSIM	0.1477	0.2379	0.2838	0.2092	0.3486	0.3315	0.4033	**0.4341**
3	PSNR	15.627	19.311	20.724	18.348	22.096	21.784	22.884	**23.141**
	SSIM	0.2192	0.3285	0.3808	0.2954	0.4482	0.4311	0.5001	**0.5253**
4	PSNR	15.848	19.514	20.940	18.552	22.333	22.013	23.155	**23.457**
	SSIM	0.2753	0.4080	0.4658	0.3687	0.5355	0.5184	0.5844	**0.6051**
5	PSNR	15.646	19.337	20.759	18.368	22.140	21.826	22.937	**23.202**
	SSIM	0.2829	0.3984	0.4501	0.3644	0.5145	0.4983	0.5636	**0.5886**
6	PSNR	16.021	19.650	20.952	18.708	22.144	21.885	22.739	**22.854**
	SSIM	0.2488	0.3501	0.3934	0.3197	0.4502	0.4358	0.4935	**0.5117**
7	PSNR	16.005	19.519	20.773	18.619	21.896	21.652	22.449	**22.556**
	SSIM	0.2365	0.3291	0.3730	0.3012	0.4294	0.4150	0.4726	**0.4917**
8	PSNR	15.598	19.393	20.990	18.382	22.669	22.266	23.819	**24.466**
	SSIM	0.1381	0.2345	0.2876	0.2037	0.3598	0.3409	0.4203	**0.4550**
9	PSNR	15.727	19.354	20.644	18.421	21.7926	21.546	22.325	**22.400**
	SSIM	0.2359	0.3416	0.3832	0.3104	0.4337	0.4216	0.4656	**0.4722**
10	PSNR	15.565	19.315	20.814	18.323	22.325	21.973	23.266	**23.660**
	SSIM	0.1859	0.2928	0.3442	0.2596	0.4108	0.3939	0.4603	**0.4804**
11	PSNR	15.606	19.379	20.925	18.378	22.510	22.136	23.538	**24.026**
	SSIM	0.1716	0.2807	0.3353	0.2464	0.4068	0.3886	0.4620	**0.4877**
12	PSNR	15.533	19.287	20.780	18.293	22.288	21.937	23.224	**23.613**
	SSIM	0.1987	0.3128	0.3633	0.2773	0.4295	0.4130	0.4775	**0.4962**

Bold values indicate highest values of PSNR and SSIM.

5.3.3 Simulation Results Considering Salt and Pepper Noise

Now, the performance of the *proposed* GFI mask is tested on images affected by salt and pepper (SP) noise, a kind of impulse noise in which particular amount of pixels have the intensity values 0 or 255. For images affected by SP noise, the denoising performance of the *proposed* GFI mask is analyzed so as to compare the performance of different fractional-order operators. The experiments are performed on images extracted from a dataset named Set12 [126].

(a) Original Image	**(b)** Noisy Image	**(c)** GL	**(d)** GFI-RL	**(e) GFI-AB**
	PSNR=20.732	PSNR=27.031	PSNR=28.228	**PSNR=28.542**
	SSIM=0.4828	SSIM=0.6668	SSIM=0.7055	**SSIM=0.7143**

Figure 5.3: Denoising performance of various fractional-order masks on Lena image corrupted by salt and pepper noise at noise density 0.03.

Figure 5.3 presents the denoising performance of the *proposed* GFI mask on a randomly selected image from this dataset. Figure 5.3a represents the original image and Figure 5.3b represents the noisy image affected by SP noise at density 0.03. Figure 5.3c, 5.3d, and 5.3e represent the denoising results of the conventional GL, RL (herein, referred to as GFI-RL) and the *proposed* GFI-AB masks along with the values of PSNR and SSIM. From the results depicted in Figure 5.3, it can be seen that the *proposed* GFI-AB mask not only removes the noise significantly but also outperforms the conventional GL and RL-based masks in terms of highest values of PSNR and SSIM.

(a) Original Image	**(b)** Noisy Image	**(c)** GL	**(d)** GFI-RL	**(e) GFI-AB**
	PSNR=18.595	PSNR=24.775	PSNR=25.941	**PSNR=26.265**
	SSIM=0.4662	SSIM=0.6412	SSIM=0.6762	**SSIM=0.6836**

Figure 5.4: Denoising performance of various fractional-order masks on MRI image corrupted by SP noise at density 0.04.

To broaden the scope of the *proposed* work, an MRI medical image corrupted with salt and pepper (SP) noise is also taken into account and the denoising performance of various methods is studied. The denoising performance of several methods on MRI image corrupted by 0.04 density SP noise is illustrated in Figure 5.4 along with the values of PSNR and SSIM.

From the results, it can be observed the *proposed* GFI-AB performs better than GL and RL-based masks in terms of visual perception as well as highest values of PSNR and SSIM.

Furthermore, it can be observed that although the conventional and *proposed* fractional-order masks reduce the noise significantly, the performance of these masks is highly dependent on the choice of fractional-order α. Hence, while denoising the images, fractional-order needs to be optimized manually according to the noise variance and image features. Therefore, it becomes challenging to choose fractional-order for each image independently. On the other hand, it is difficult to obtain better image denoising effect by utilizing a fixed fractional-order to process noisy pixels in distinct regions.

5.4 Adaptive Fractional-Order for Image Denoising

An image affected by salt and pepper (SP) noise consists of noisy pixels which may be located in the regions with different characteristics (such as textures, edges and smooth regions) of an image. Since characteristics of an image varies locally, it is difficult to obtain better denoising effect by using same fractional order to process noisy pixels in distinct regions. Therefore, it is neccessary to select the fractional-order of the masks in an adaptive manner based upon the local features of an image. To address this challenge, an adaptive function for the selection of fractional order for GFI mask is presented.

5.4.1 Mathematical Investigation of GFI-based Adaptive Mask (GFI-AM)

This section presents a generalized fractional integral based adaptive mask (GFI-AM), where fractional-order is decided adaptively based upon average local gradient of the image. As the gradient of noisy pixel is larger as compared to that of edge and texture regions, image gradient helps to analyze intensity variations of the image. Therefore, it can be said that the noisy pixels with significantly larger average gradient will require more attenuation than the ones with small average gradient, and hence, fractional-order needs to be chosen accordingly in adaptive manner. Therefore, image gradient plays a crucial role in deciding the fractional-order of the *proposed* GFI mask.

The mean of local absolute gradients of each non-corner pixel of an image $p(i, j)$ in all

possible eight directions is given by

$$\begin{aligned} M(i,j) = \frac{1}{8}\,(&|p(i,j) - p(i-1,j-1)| + |p(i,j) - p(i-1,j+1)| \\ &+|p(i,j) - p(i,j+1)| + |p(i,j) - p(i-1,j)| + |p(i,j) - p(i+1,j)| \\ &+|p(i,j) - p(i,j-1)| + |p(i,j) - p(i+1,j-1)| + |p(i,j) - p(i+1,j+1)|) \end{aligned} \tag{5.14}$$

If the maximum of $M(i,j)$ in its 3×3 neighborhood is X and the minimum of $M(i,j)$ is Y, then the adaptive fractional-order α_{GFI-AM} of GFI mask is given by

$$\alpha_{GFI-AM} = \alpha_1 + \frac{(M(i,j) - X)(\alpha_2 - \alpha_1)}{Y - X} \tag{5.15}$$

where the range of fractional-order is given by $\alpha_{GFI-AM} \in [\alpha_1, \alpha_2]$. Here, the range of fractional integral order is chosen as $[0.5, 1.5]$ in accordance with the well-established literature [93, 94]. As the noisy pixels with large average gradient need more attenuation than the ones with small average gradient, the fractional-order related to each noisy pixel is selected by utilizing the formula, presented in (5.15). Then, the $(2r+1) \times (2r+1)$ mask is convolved with the image to get the final denoised image. Here, $(2r+1)$ represents the size of the mask for image denoising, where $r = 1, 2, 3$ corresponds to 3×3, 5×5 and 7×7 masks, respectively.

5.4.2 Comparative Analysis between GFI-AM-RL and GFI-AM-AB

To illustrate the effectiveness of the *proposed* GFI-AM, the performance of RL-based GFI-AM (herein, referred to as GFI-AM-RL) and AB-based GFI-AM (herein, referred to as GFI-AM-AB) are tested on same images as presented in Section 5.3.3. Furthermore, the performance of the *proposed* GFI-AM is compared against well-established GL-based global adaptive fractional integral algorithm (GAFIA), presented in [93].

Figure 5.5 presents the original, noisy and denoised images obtained by various fractional-order operators where fractional-order has been selected by utilizing the formula presented in (5.15). Figure 5.5c, 5.5d, and 5.5e illustrates the denoised images attained by utilizing GAFIA [93], GFI-AM-RL, and GFI-AM-AB, respectively. From the results presented in Figure 5.5, it can be seen that the values of PSNR and SSIM attained by each method is higher than that of the noisy image. However, the *proposed* GFI-AM-AB attains the highest values of PSNR and SSIM as compared to well-estabilised GAFIA [93] and GFI-AM-RL

methods.

(a) Original Image	**(b)** Noisy Image	**(c)** GAFIA [93]	**(d)** GFI-AM-RL	**(e) GFI-AM-AB**
	PSNR=20.732	PSNR=27.663	PSNR=28.963	**PSNR=32.850**
	SSIM=0.4828	SSIM=0.6941	SSIM=0.7352	**SSIM=0.8708**

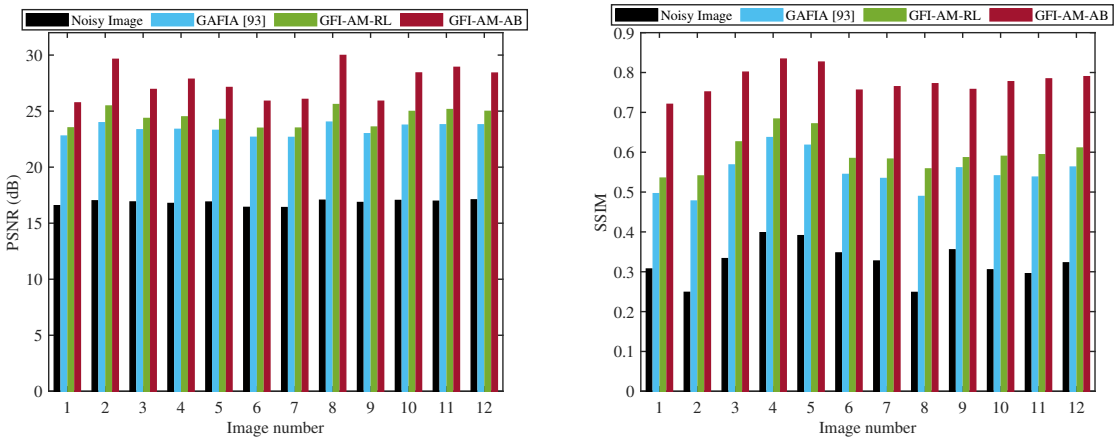
Figure 5.5: Denoising performance of various adaptive fractional-order masks on Lena image corrupted by salt and pepper noise at noise density 0.03.

Similarly, the results have been obtained for MRI image affected by SP noise at density 0.04. The corresponding results are presented in Figure 5.6, where it can be seen that the proposed GFI-AM-AB significantly outperforms the conventional methods presented in Figure 5.6c and 5.6d, respectively.

(a) Original Image	**(b)** Noisy Image	**(c)** GAFIA [93]	**(d)** GFI-AM-RL	**(e) GFI-AM-AB**
	PSNR=18.595	PSNR=25.396	PSNR=26.659	**PSNR=30.525**
	SSIM=0.4662	SSIM=0.6719	SSIM=0.7083	**SSIM=0.8537**

Figure 5.6: Denoising performance of various fractional-order masks on MRI image corrupted by SP noise at density 0.04.

To further validate the robustness of the proposed method, all images from Set12 dataset are considered and are affected by SP noise at density 0.07. The comparison among GAFIA [93], GFI-AM-RL, and GFI-AM-AB in terms of PSNR and SSIM are presented in Figure 5.7. Figure 5.7a and 5.7b present the PSNR and SSIM values of all images numbered 1 to 12.



(a) **(b)**

Figure 5.7: (a) PSNR and (b) SSIM comparison among various methods on all images of Set12 dataset corrupted by salt and pepper noise at density 0.07.

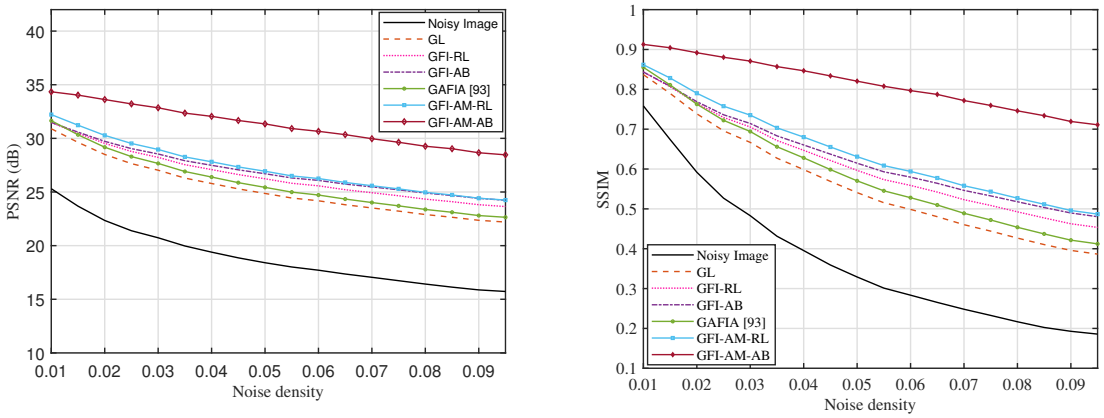
From the results, it can be depicted that the PSNR and SSIM values of the images denoised by the *proposed* GFI-AM-AB are higher than the conventional GAFIA [93] and GFI-AM-RL methods for all images. This validates the effectiveness of proposed GFI-AM-AB approach for images of different sizes and characteristics.

5.5 Comparative Analysis between GFI and GFI-AM Fractional-Order Masks

To elucidate the advantage of utilizing adaptive fractional-order, the comparison between non-adaptive GFI mask and GFI-AM is done rigorously. Comparing the results presented in Figure 5.3 and Figure 5.5, it is evident that the performance of adaptive masks (i.e., GAFIA, GFI-AM-RL, GFI-AM-AB) in each case is better than the non-adaptive masks (i.e., GL, GFI-RL, GFI-AB), respectively. Similar pattern can be observed by comparing Figure 5.4 and Figure 5.6, respectively. However, it can be noticed that in both these cases, the proposed GFI-AM-AB outperforms the non-adaptive and adaptive methods discussed so far.

Furthermore, to show the effectiveness of the proposed GFI and GFI-AM at various levels of noise densities, Figure 5.8a and Figure 5.8b present the noise density versus PSNR and noise density versus SSIM plots, respectively. Here, the same image from Set12 dataset is contaminated with noise at various densities ranging from 0.01-0.09. The results depict that the proposed GFI-AM-AB mask outperforms the conventional GL, GFI-RL, GAFIA [93],

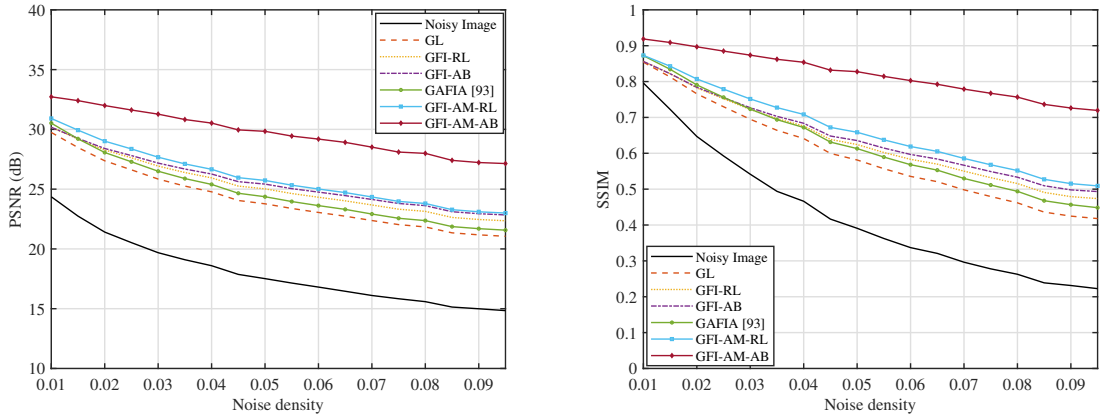
and GFI-AM-RL masks even when high density noise is taken into account.



(a) **(b)**

Figure 5.8: (a) PSNR and (b) SSIM comparisons among various methods on Lena Image affected by salt and pepper noise at various noise densities.

Additionally, it can be observed from Figure 5.8 that the proposed GFI-AM-AB attains highest values of PSNR and SSIM in comparison to other methods at various levels of noise density, which validates the effectiveness of the proposed GFI based adaptive mask incorporating AB fractional operator.



(a) **(b)**

Figure 5.9: (a) PSNR and (b) SSIM comparison among several methods on MRI image affected by salt and pepper noise at various noise densities.

This indicates that out of the three fractional-order operators considered, the AB-based fractional-order integral mask outperforms the conventional GL and RL based fractional-

order masks significantly.

Similar pattern of results have been obtained for MRI image, where MRI image is contaminated with noise at various densities ranging from 0.01-0.09. The quantitative results of various fractional operators on MRI image at different noise densities is illustrated in Figure 5.9. The corresponding values of PSNR and SSIM attained by utilizing different fractional operators at various noise densities are also detailed in Table 5.4. From the results, it is evident that the *proposed* GFI-AM-AB outperforms other methods and hence, has the potential to denoise medical images affected by SP noise at various densities in an effective manner.

Table 5.4: Quantitative comparison among different methods for an MRI image corrupted by salt and pepper noise.

Noise density	Parameters	Noisy	GL [13]	GFI-RL	**GFI-AB**	GAFIA [93]	GFI-AM-RL	**GFI-AM-AB**
0.01	PSNR	24.362	29.742	30.176	**30.176**	30.520	30.913	**32.723**
	SSIM	0.795	0.853	0.856	**0.856**	0.872	0.872	**0.918**
0.02	PSNR	21.409	27.371	28.271	**28.400**	28.066	29.015	**31.992**
	SSIM	0.646	0.765	0.782	**0.783**	0.7906	0.807	**0.8965**
0.03	PSNR	19.683	25.843	26.931	**27.178**	26.499	27.674	**31.273**
	SSIM	0.541	0.694	0.722	**0.7268**	0.723	0.751	**0.873**
0.04	PSNR	18.595	24.775	25.941	**26.265**	25.396	26.659	**30.525**
	SSIM	0.466	0.641	0.676	**0.683**	0.671	0.708	**0.853**
0.05	PSNR	17.514	23.772	25.023	**25.422**	24.366	25.729	**29.837**
	SSIM	0.3908	0.581	0.6249	**0.635**	0.613	0.658	**0.827**
0.06	PSNR	16.807	23.051	24.322	**24.753**	23.622	25.010	**29.188**
	SSIM	0.336	0.535	0.583	**0.596**	0.568	0.618	**0.802**
0.07	PSNR	16.104	22.375	23.675	**24.138**	22.918	24.345	**28.514**
	SSIM	0.296	0.498	0.550	**0.566**	0.529	0.585	**0.778**
0.08	PSNR	15.588	21.836	23.141	**23.620**	22.368	23.800	**27.992**
	SSIM	0.262	0.461	0.515	**0.533**	0.493	0.551	**0.756**
0.09	PSNR	14.993	21.176	22.461	**22.946**	21.694	23.105	**27.226**
	SSIM	0.231	0.424	0.479	**0.497**	0.456	0.515	**0.726**

Bold values indicate highest values of PSNR and SSIM attained by the proposed methods.

Based upon the experimental studies conducted, it is seen that the performance of GFI mask incorporating AB fractional integral operator is better than that of conventional GL and RL fractional-order masks. Furthermore, it can be seen that the RL fractional-order mask performs better than that of GL-based fractional-order mask, which further validates the analytical study presented in Chapter 3. On the other hand, it has been seen that it is necessary to choose the fractional-order adaptively to obtain a better denoising effect. However, among various adaptive fractional-order masks, it is validated that the proposed GFI-AM incorporat-

ing AB fractional operator (GFI-AM-AB) outperforms the well-established global adaptive fractional integral algorithm (GAFIA) method based upon GL fractional-order operator [93].

5.6 Summary

This chapter puts forward a generalized framework for the design of fractional-order integral mask which incorporates Riemann-Liouville and Atangana-Baleanu definitions of fractional-order integrals for image denoising applications. The main advantage of the proposed GFI mask is that by utilizing only one mask, the performance of both fractional-order operators for image denoising applications can be analyzed. On the other hand, since there is difficulty in the selection of fractional-order manually, the order selection technique for the proposed GFI mask is presented. Several standard and medical images of different characteristics are examined to evaluate the performance of the proposed mask for denoising applications. From the simulation studies conducted, it has been validated that the proposed AB based fractional-order adaptive mask outperforms the conventional GL and RL based non-adaptive and adaptive masks significantly.

However, in the process of image denoising, some of the features such as texture and edge details of the image get blurred. Therefore, it is important to preserve the texture and edge details while denoising the image simultaneously. Also, a noise of higher density may exist in digital images. Hence, to address these challenges, a novel generalized fractional integral and fractional differential based adaptive algorithm (GFIFD-AA) is proposed in next chapter for images affected by salt and pepper noise. The proposed algorithm not only preserves the texture and edge information of the image, but also extends the range of noise density considered in this chapter (i.e., from 1%-10% to 1%-95%).

Chapter 6

Generalized Fractional Integral and Fractional Differential based Adaptive Algorithm for Image Denoising

In the previous chapter, a generalized fractional integral based adaptive mask (GFI-AM) incorporating RL and AB fractional operators was presented, where the fractional order was chosen based upon the average local gradient of the image. In the process of image denoising, it has been observed from the results that some of the image features, for example, edges, textures, etc., get blurred. Therefore, to enhance and preserve the image edge and texture features while denoising the image, a novel image denoising algorithm, referred to as generalized fractional integral and fractional differential based adaptive algorithm (GFIFD-AA) is proposed in this chapter.

The proposed GFIFD-AA is segregated into two parts: In the first part, a novel noise detection method (NDM) for the detection of salt and pepper noise is introduced. In the second part, the noisy pixels detected by proposed NDM are processed by utilizing the proposed GFI-AM introduced in the previous chapter. In order to preserve and enhance the texture and edge details of the image, the non-noisy pixels are further updated by utilizing fractional differential mask. For this, a generalized framework for the design of fractional differential mask is introduced which incorporates various definitions of fractional-differential operators, such as, Riemann-Liouville (RL)/Caputo, Caputo-Fabrizio (CF), Atangana-Baleanu (AB), etc. Then, a generalized fractional differential based adaptive mask (GFD-AM) is put forward. The proposed GFI-AM and GFD-AM are then utilized together in the framework

of proposed GFIFD-AA for the effective removal of SP noise and preservation of details in images.

6.1 Generalization of Fractional-Order Differential Operators

This chapter proposes a generalized fractional differential (GFD) mask which incorporates various fractional-order kernels such as power-law kernel, exponentional kernel, and Mittag-Leffler kernel, corresponding to Riemann-Liouville (RL)/Caputo, Caputo-Fabrizio (CF), and Atangana-Baleanu (AB) fractional-order differentials, respectively.

To develop an analytical foundation of GFD mask, various definitions of fractional-order differential operators are first reviewed briefly. Let α be the fractional order (where $\alpha > 0$), $\kappa(t, \alpha)$ be the kernel function according to the definition of fractional-order differential being utilized and $\mathcal{N}(\alpha)$ be the normalization constant [29]. Then, the generalized fractional-order derivative in Caputo and Riemann-Liouville sense can be described as follows:

Definition 6.1.1: The generalized fractional-order differential of order α in Caputo sense is defined as:

$$D^{\alpha}_{GFD-C}\, p(t) = \mathcal{N}(\alpha) \int_0^t \frac{d}{dt} p(\zeta)\kappa(t-\zeta, \alpha) d\zeta \tag{6.1}$$

where the subscript GFD and C denote generalized fractional derivative and Caputo, respectively. From signal processing prespective, fractional-order derivative in Caputo sense can be seen as convolution of the kernel function with derivative of the signal, i.e.,

$$D^{\alpha}_{GFD-C}\, p(t) = \mathcal{N}(\alpha) \left[\frac{d}{dt} p(t) * \kappa(t, \alpha) \right] \tag{6.2}$$

where $*$ represents the convolution operator.

Definition 6.1.2: Similarly, the generalized fractional-order differential of order α in Riemann-Liouville sense is defined as:

$$D^{\alpha}_{GFD-RL}\, p(t) = \mathcal{N}(\alpha) \frac{d}{dt} \int_0^t p(\zeta)\kappa(t-\zeta, \alpha) d\zeta \tag{6.3}$$

where the subscript GFD and RL denote generalized fractional derivative and Riemann-Liouville, respectively. A fractional-order derivative in RL sense can be viewed as the rate of change of convolution between the signal and the kernel [139], i.e.,

$$D^{\alpha}_{GFD-RL}\, p(t) = \mathcal{N}(\alpha)\frac{d}{dt}\left[p(t) * \kappa(t,\alpha)\right] \tag{6.4}$$

The kernel functions $\kappa(t,\alpha)$ and normalization constants $\mathcal{N}(\alpha)$ for each definition are listed in Table 6.1, where $\Gamma(x)$ represents the Gamma function of x, $E_{\alpha}(x) = \sum_{k=0}^{\infty} \frac{x^k}{\Gamma(\alpha k+1)}$ represents one-parameter Mittag-Leffler function, and $M(\alpha) = B(\alpha) = 1-\alpha+\alpha/\Gamma(\alpha)$ is the normalization constant. The kernel function $\kappa(t,\alpha)$ includes power-law kernel corresponding to RL/Caputo fractional derivatives, exponentional kernel corresponding to Caputo-Fabrizio (CF) fractional derivative, and Mittag-Leffler kernel corresponding to Atangana-Baleanu (AB) fractional derivative.

Table 6.1: Fractional-differential operators and their corresponding parameters

Fractional Derivative Operators	$\mathcal{N}(\alpha)$	$\kappa(\zeta,\alpha)$	$g(\zeta,\alpha)$
Riemann-Liouville/Caputo	$\frac{1}{\Gamma(\alpha)}$	$\zeta^{-\alpha}$	$\frac{\zeta^{1-\alpha}}{1-\alpha}$
Caputo-Fabrizio (CF)	$\frac{M(\alpha)}{1-\alpha}$	$exp\left(-\frac{\alpha}{1-\alpha}\zeta\right)$	$-\frac{1-\alpha}{\alpha}exp\left(-\frac{\alpha}{1-\alpha}\zeta\right)$
Atangana-Baleanu (AB)	$\frac{B(\alpha)}{1-\alpha}$	$E_{\alpha}(-\frac{\alpha}{1-\alpha}\zeta^{\alpha})$	$\zeta E_{\alpha,2}(\frac{-\alpha}{1-\alpha}\zeta^{\alpha})$

6.1.1 Analytical Foundation of Generalized Fractional Differential (GFD) Mask

Before detailing the design procedure, let the interval $[0,t]$ in (6.1) is divided into L equal parts. Then, the $L+1$ causal pixels can be expressed as

$$\begin{aligned}
&p_L \equiv p(0)\\
&p_{L-1} \equiv p(\tfrac{t}{L})\\
&\vdots\\
&p_k \equiv p(t-\tfrac{kt}{L})\\
&\vdots\\
&p_0 \equiv p(t)
\end{aligned}$$

To design GFD mask of order α, the generalized fractional-order differential in Caputo sense, presented in (6.1) can be written as:

$$\begin{aligned} D^{\alpha}_{GFD-C}\, p(t) &= \mathcal{N}(\alpha) \int_0^t \frac{d}{d\zeta} p(t-\zeta)\kappa(\zeta,\alpha)d\zeta \\ &= \mathcal{N}(\alpha) \sum_{k=0}^{L-1} \int_{kt/L}^{(k+1)t/L} \frac{d}{d\zeta} p(t-\zeta)\kappa(\zeta,\alpha)d\zeta \end{aligned} \tag{6.5}$$

Then, the first-order differential inside (6.5) is approximated numerically by utilizing finite difference method, described as

$$\begin{aligned} &\int_{kt/L}^{(k+1)t/L} \frac{d}{d\zeta} p(t-\zeta)\kappa(\zeta,\alpha)d\zeta \\ &\qquad \approx \frac{p(t-kt/L) - p(t-(k+1)t/L)}{t/L} \int_{kt/L}^{(k+1)t/L} \kappa(\zeta,\alpha)d\zeta \end{aligned} \tag{6.6}$$

Let $\int \kappa(\zeta,\alpha) = g(\zeta,\alpha)$, then (6.6) can be written as

$$\begin{aligned} \int_{kt/L}^{(k+1)t/L} \frac{d}{d\zeta} p(t-\zeta)\kappa(\zeta,\alpha)d\zeta &\approx \frac{p_k - p_{k+1}}{h} \left[g\left(\frac{(k+1)t}{L},\alpha\right) - g\left(\frac{kt}{L},\alpha\right)\right] \\ &\approx \frac{p_k - p_{k+1}}{h} \left[g((k+1)h,\alpha) - g(kh,\alpha)\right] \end{aligned} \tag{6.7}$$

where $h = t/L$ (As in digital images, the length of image is finite and minimum step-size is one pixel. Therefore, L can be large enough so that $t/L = 1$ [13]) Substituting (6.7) into (6.5), implies

$$\begin{aligned} D^{\alpha}_{GFD-C}\, p(t) &= \mathcal{N}(\alpha) \sum_{k=0}^{L-1} (p_k - p_{k+1}) \left[g((k+1),\alpha) - g(k,\alpha)\right] \\ &= \mathcal{N}(\alpha) \Bigg\{ \Big[g(1,\alpha) - g(0,\alpha) \Big] p_0 \\ &\quad + \sum_{k=1}^{L-1} \Big[g((k+1),\alpha) - 2g(k,\alpha) + g((k-1),\alpha) \Big] p_k \\ &\quad + \Big[g((L-1),\alpha) - g(L,\alpha) \Big] p_L \Bigg\} \end{aligned} \tag{6.8}$$

As a result, the coefficients a_k of the *proposed* GFD mask in Caputo (GFD-C) sense are given as:

$$a_k = \begin{cases} g(1,\alpha) - g(0,\alpha) & ; k = 0 \\ g((k+1),\alpha) - 2g(k,\alpha) + g((k-1),\alpha) & ; 1 \leq k \leq L-1 \\ g((L-1),\alpha) - g(L,\alpha) & ; k = L \end{cases} \tag{6.9}$$

where $\kappa(\zeta)$ and $g(\zeta)$ represent the kernel and its corresponding integral, presented in Table 6.1, based upon the type of definition being utilized.

Similarly, the coefficients b_k of the *proposed* generalized fractional differential mask in RL (GFD-RL) sense can be derived and are as follows:

$$b_k = \begin{cases} g(1,\alpha) - g(0,\alpha) & ; k = 0 \\ g((k+1),\alpha) - 2g(k,\alpha) + g((k-1),\alpha) & ; 1 \leq k \leq L-1 \\ g((L-1),\alpha) - g(L,\alpha) + \kappa(L) & ; k = L \end{cases} \tag{6.10}$$

Therefore, (6.9) and (6.10) present the *proposed* coefficients of generalized fractional differential masks in Caputo and RL sense, respectively which incorporate power-law kernel, exponential kernel, and Mittag-Leffler kernel, etc. For instance, substituting the integral of power-law kernel denoted by $g(k,\alpha)$ in (6.10), the coefficients of RL fractional-order differential can be obtained, similar to the one attained in the operator YiFeiPU-5 [13].

Hence, it can be said that the *proposed* GFD mask provides the flexibility in obtaining various types of fractional-order masks without going through the numerical approximation for each definition separately.

6.1.2 Mathematical Investigation of GFD-based Adaptive Mask (GFD-AM)

As mentioned earlier, it is difficult to choose fractional order manually and hence, an adaptive function for the selection of fractional-order for GFD mask is presented. This section puts forward a GFD-based adaptive mask (GFD-AM) in order to select the fractional-order based upon the characteristics of the given image. Since the properties of an image vary locally, an

image can be classified into four distinct regions, i.e., smooth region, object region, texture region, and edge region. Then, the fractional differential order of GFD-AM is selected based upon the characteristics of the image. The adaptive function for selecting fractional-order for GFD mask is described by

$$\alpha_{GFD-AM} = \begin{cases} \beta^s & ; M(i,j) < \tau_s \\ \left(\frac{\beta_2^t - \beta_1^t}{\tau_s - \tau_t}\right) M(i,j) + \frac{\beta_1^t \tau_s - \beta_2^t \tau_t}{\tau_s - \tau_t} & ; \tau_s \leq M(i,j) < \tau_t \\ \left(\frac{\beta_2^o - \beta_1^o}{\tau_t - \tau_e}\right) M(i,j) + \frac{\beta_1^o \tau_t - \beta_2^o \tau_e}{\tau_t - \tau_e} & ; \tau_t \leq M(i,j) < \tau_e \\ \beta^e & ; M(i,j) \geq \tau_e \end{cases} \tag{6.11}$$

where τ_s, τ_t, τ_e represent the thresholds to identify smooth regions, texture regions and edge regions, respectively [95] and $M(i,j)$ represents the local average gradient, presented in (5.14).

The pixels with very low value of local average gradient are assumed to be the part of smooth regions. This is because smooth regions are the regions where the pixel intensities do not change much in a neighbourhood, i.e., the difference in intensity values of pixels in a region of say 3×3 is very less. Hence, the value of local average gradient (denoted by $M(i,j)$) is also less. Therefore, the threshold τ_s for smooth regions is chosen to be 0.02 (approx. close to 0). That is, the pixels with $M(i,j) \leq 0.02$ are presumed to be the part of smooth regions [95] and hence, the intensity value of the pixels in smooth areas should remain unchanged. Therefore, the fractional differential order α_{GFD-AM} corresponding to smooth region is chosen as $\beta_s = 0.01$. The threshold for texture region τ_t is selected by utilizing improved Otsu method which uses 2-D histogram of gray image and object region to segment an image [140]. As discussed earlier, the local average gradients of edge pixels are relatively larger, therefore, the edge threshold τ_e is computed as an average of highest fifty values of $M(i,j)$ of pixels. Furthermore, the parameters β_1^t, β_2^t are utilized for texture regions, β_1^o, β_2^o are used for object regions and β^e is used for edges. These parameters are utilized to select the generalized fractional differential order for the pixels present in the image, as discussed further in Section 6.2.2.

6.1.3 Image Enhancement using GFD-based Adaptive Mask

The main aim of image enhancement is to make edges clearer and preserve smooth areas and weak textures in the image. Therefore, to achieve the purpose of image enhancement, an adaptive fractional differential mask is considered where high fractional order is chosen for edge pixels and a relatively small order is chosen for weak texture pixels. The criteria to select fractional order based on characteristics of the image is presented in (6.11).

To illustrate the effectiveness of the adaptive mask for image enhancement applications, different images from Set12 [126] dataset are considered. Table 6.2 presents the assessment parameters obtained by utilizing the proposed AB fractional differential mask at $\alpha = 0.5, 0.8, 0.9$ and when fractional-order is chosen adaptively.

From the results presented in Table 6.2, it can be seen that the entropy, EME, and contrast values for $\alpha = 0.8$ and $\alpha = 0.9$ are the highest. This is also evident from the results presented in Table 4.4. This is because the same fractional-order is applied on the entire image where different regions of the image such as smooth, texture and edge pixels are enhanced by the same fractional order. This is in contrast to the approach where fractional order is chosen adaptively. For smooth and texture-rich regions, the pixel densities do not change much in the neighborhood. Therefore, a relatively small fractional-order is chosen in order to preserve the smooth and texture details of the image. However, for edge pixels, the fractional-order chosen is large so as to enhance the details significantly. Therefore, it can be said that choosing fractional-order adaptively provides better performance in preserving the texture details, enhancing the edge details and hence, improving the overall image clarity and contrast in comparison to the case where fixed fractional order is chosen.

Comparing the results presented in Table 4.4 and Table 6.2, although the values of these parameters attained at $\alpha = 0.8$ and $\alpha = 0.9$ order are larger than that of adaptive approach, smooth and texture regions in these images get over enhanced. On the other hand, the weak textures and smooth areas of an image are nearly eliminated in images processed by the integer order method. Its global information is not complete and lacks some textures and smooth areas that were present in the original image [98, 99]. Similar pattern of results can be seen for all the images and hence, it can be concluded that choosing fractional-order adaptively possess significant advantages in preserving the details of the image while enhancing it si-

multaneously.

Table 6.2: Comparative analysis between non-adaptive and adaptive fractional-order differential masks

Images	Order α	Entropy	EME	Contrast	Energy	Correlation	Homogeneity
Lena (512×512)	0	7.1914	8.3158	0.3799	0.1890	0.9128	0.8753
	0.5	7.3635	12.249	0.9465	0.1119	0.8089	0.7646
	0.8	7.5986	17.838	2.5255	0.0521	0.5940	0.5861
	0.9	7.6289	18.782	3.0314	0.0447	0.5394	0.5861
	Adaptive	**7.5451**	**15.856**	**2.1484**	**0.0608**	**0.6356**	**0.6425**
Starfish (512×512)	0	7.7506	9.9454	0.40	0.0827	0.9481	0.8647
	0.5	7.7956	14.6002	0.8765	0.0660	0.8964	0.8089
	0.8	7.7856	19.9859	2.0431	0.0477	0.7939	0.7265
	0.9	7.7522	19.0155	2.3424	0.0449	0.7719	0.7093
	Adaptive	**7.7986**	**18.2639**	**1.9008**	**0.0507**	**0.8039**	**0.7425**
Plane (512×512)	0	6.7131	7.3526	0.4624	0.3526	0.8905	0.8937
	0.5	7.0052	10.487	1.2737	0.2720	0.7609	0.8247
	0.8	7.0360	11.761	2.7493	0.1727	0.6169	0.7350
	0.9	7.0269	11.149	3.0859	0.1490	0.5927	0.7139
	Adaptive	**7.0268**	**10.969**	**2.4292**	**0.1901**	**0.6420**	**0.7524**
Peppers (512×512)	0	7.5327	9.5925	0.3850	0.1096	0.9312	0.8881
	0.5	7.6339	10.9539	0.8076	0.0804	0.8678	0.8190
	0.8	7.6699	12.5275	1.7391	0.0515	0.7540	0.7163
	0.9	7.6640	12.423	2.0064	0.0467	0.7258	0.6929
	Adaptive	**7.6732**	**11.542**	**1.5172**	**0.0558**	**0.7783**	**0.7349**

6.2 Proposed GFIFD-based Adaptive Algorithm (GFIFD-AA)

An image affected with SP noise consists of noisy pixels which may be located in the regions with different characteristics (such as textures, edges, and smooth regions) of an image. Since it is difficult to preserve the texture and edge details of the image while denoising the image simutaneously, a generalized fractional integral and fractional differential based adaptive algorithm (GFIFD-AA) is proposed for efficient removal of SP noise as well as

preservation of details in images.

The proposed GFIFD-AA is segregated into two parts. In the first part, a novel noise detection method (NDM) is proposed which detects the noisy pixels of an image by utilizing gradient information of the image. In the second part, these noisy pixels are processed by utilizing the proposed GFI-based adaptive mask (GFI-AM). The rest of the pixels, i.e., the noise-free pixels are then processed by utilizing the proposed GFD-based adaptive mask (GFD-AM) so as to enhance the texture and edge details of an image. Therefore, it can be said that the proposed GFIFD-AA not only removes the noise, but also preserves and enhances the texture and edge information of the image simultaneously. Hence, GFI-AM mask is utilized for image denoising and GFD-AM mask is utilized for enhancement, with the aim that texture and edge details are preserved and enhanced simultaneously while removing the noise.

6.2.1 Noise Detection Method (NDM) for Salt and Pepper Noise

The main aim of the *proposed* noise detection method (NDM) is to initially calculate the expected number of noisy pixels in an image and then to identify the location of those noisy pixels. Therefore, in the *proposed* NDM, initially the expected total number of noise points in an image are obtained based on probability strategy. Then, the location of these noisy pixels is identified by using gradient information of the image.

Salt and pepper (SP) noise is a kind of impulse noise in which a particular amount of pixels have the intensity values 0 or 255. Therefore, the main aim of the *proposed* NDM is to distinguish original pixels having intensity values 0 or 255 against that of the noisy pixels. The details of the proposed NDM are presented as follows:

6.2.1.1 Probability Strategy for the Prediction of Noisy pixels

In the proposed NDM, the appearance of noise points in the image corrupted with SP noise is considered as a probability event. Assume that the occurrence of noise point in an image is a random event ξ and the probability of occurrence of ξ is P_n (i.e., the predicted density of noise). Then, for an image of size $A \times B$, the expected number of noise points and the locations of those noise points in the image are calculated by the *proposed* NDM, elaborated as follows:

Consider 3×3 regions of an image. The probability that only center pixel is corrupted in 3×3 regions of an image is calculated as:

$$P_1 = \frac{(A-2)(B-2)}{AB} P_n(1-P_n)^8 + \frac{2(A+B)-8}{AB} P_n(1-P_n)^5 + \frac{4}{AB} P_n(1-P_n)^3 \tag{6.12}$$

where A and B represent the number of rows and columns of an image, P_n is the density of noise, which is calculated as $P_n = \frac{card(S)+card(P)}{AB}$. Here, S and P represent the set of salt and pepper pixels, respectively and $card(.)$ denotes cardinality of a set. That is, $card(S)$ and $card(P)$ represent the number of salt and pepper pixels in an image, i.e., the pixels with intensity values 255 and 0, respectively. AB represents the total number of pixels in an image. Since in natural images the number of noise-free pixels acquiring the boundary values (0 or 255) is very small [102, 104, 105], the noise density for natural images is calculated by using this formula.

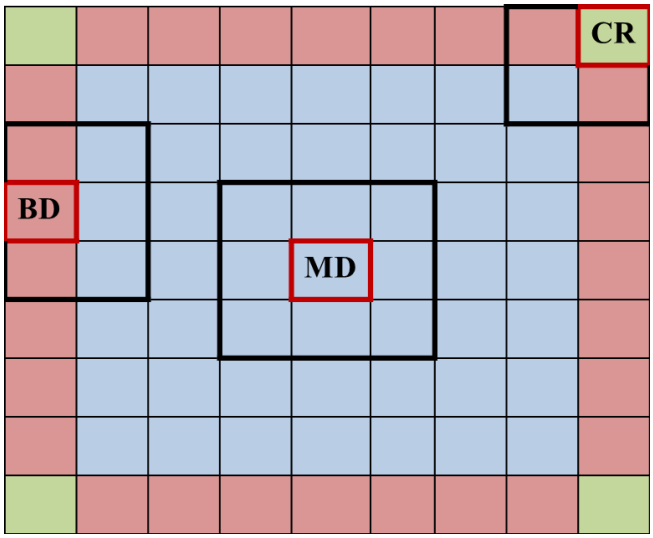


Figure 6.1: Middle (MD), Boundary (BD) and Corner (CR) pixels in an image and their neighboring pixels

To get a more intuitive idea about the probability strategy utilized in (6.12), Figure 6.1 represents the middle (MD), boundary (BD) and corner (CR) pixels of an image depicted in blue, red and green colors respectively. The black region represents the 3×3 neighborhood around these corresponding pixels. Therefore, it can be seen that whenever center of the 3×3 region approaches boundary of an image, only six pixels exist in its neighborhood. Similarly, for corner pixels, there exist only four pixels in its neighborhood. Based on that, the terms presented in (6.12) are explained as follows:

The first term in (6.12) represents the probability that the corrupted center pixel lies in the

non-boundary and non-corner regions of the image, where $(A-2)(B-2)$ represents the total number of non-boundary and non-corner pixels in an image of size $A \times B$ and $P_n(1-P_n)^8$ represents that out of total 9 pixels, one center pixel is affected in 3×3 region of an image. The second term in (6.12) denotes the probability that the corrupted center pixel is among the boundary pixels of the image, where $(2(A+B)-8)$ represents the total number of boundary pixels in an image and $P_n(1-P_n)^5$ represents that out of total 6 pixels in the vicinity of the boundary pixel, one center pixel is affected. The third term represents the probability that the corner pixel is affected by noise, where 4 represents the number of corner pixels in an image and $P_n(1-P_n)^3$ represents that out of total 4 pixels in the vicinity of corner pixel, one center pixel is affected.

Similarly, the probability that one or more than one pixel in the neighborhood is also affected along with the center pixel in 3×3 region of an image is computed as:

$$P_{l+1} = \frac{(A-2)(B-2)}{AB} P_n \left[\binom{8}{l} P_n^l (1-P_n)^{(8-l)} \right] + \frac{2(A+B)-8}{AB} P_n \left[\binom{5}{l} P_n^l (1-P_n)^{(5-l)} \right] + \frac{4}{AB} P_n \left[\binom{3}{l} P_n^l (1-P_n)^{(3-l)} \right] \tag{6.13}$$

where l represents number of pixels affected in the neighborhood along with the corrupted center pixel ($l = 1, 2, ..., 8$) and $\binom{8}{l} P_n^l (1-P_n)^{(8-l)}$ in the first term of (6.13) denotes that out of 8 pixels in the neighborhood of the affected center pixel, l pixels are also affected. In the similar manner, the rest of the terms can be depicted.

Therefore, it can be said that (6.12) calculates the probability that in 3×3 region of an image, only center pixel is affected and all the neighboring pixels are non-noisy. On the other hand, (6.13) is a generalized form of (6.12) which calculates the probability that in 3×3 region of an image, one or more than one neighboring pixels are also affected along with the corrupted center pixel. These formulae are then utilized to calculate the expected number of noisy pixels in an image corresponding to number of pixels (zero, one, or more than one) affected in its neighbourhood. Based on these number of pixels, the location of noisy pixels is identified by utilizing the proposed NDM.

Therefore, the expected quantity of noisy center pixels where no neighboring pixel is affected can be calculated as:

$$N_{P_1} = A \times B \times P_1 \tag{6.14}$$

In the similar manner, the expected number of noisy center pixels where l number of neighboring pixels are also affected along with the center pixel is given as

$$N_{P_{l+1}} = A \times B \times P_{l+1} \tag{6.15}$$

Hence, the expected total number of noise points in an image can be obtained by

$$N = A \times B \times \sum_{l=0}^{8} P_{l+1} \tag{6.16}$$

Therefore, after calculating that how many neighboring pixels are expected to be affected in the vicinity of the noisy center pixel, the next aim is to identify the location of those pixels.

6.2.1.2 Detection of Noisy Pixels using Gradient Information

It is well-known from rich literature that the gradient amplitudes of edge pixels and noisy pixels are relatively larger [93, 94, 95]. However, the edge pixels are continuous i.e., one can definitely find at least one edge pixel surrounding that edge pixel which has similar gradient magnitude [93]. On the other hand, the noise points are so random that it is difficult to find another pixel with similar gradient magnitude in its neighborhood. Therefore, the minimum absolute gradient amplitude of noisy pixel in eight directions is generally larger. The criteria based on minimum absolute gradient amplitudes is, thus utilized for detecting the locations of these noise points in an image.

For the pixel $p(i, j)$ in an image of size $A \times B$, the minimum absolute gradient amplitude of each pixel in eight directions is given as:

$$\begin{aligned} D_{min}(i,j) = min[&|(p(i,j) - p(i-1,j-1))|, |(p(i,j) - p(i-1,j))|, \\ &|(p(i,j) - p(i-1,j+1))|, |(p(i,j) - p(i,j-1))|, \\ &|(p(i,j) - p(i,j+1))|, |(p(i,j) - p(i+1,j-1))|, \\ &|(p(i,j) - p(i+1,j))|, |(p(i,j) - p(i+1,j+1))|] \end{aligned} \tag{6.17}$$

Therefore, to identify the location of noisy pixels, the image is classified based upon the number of pixels affected in 3×3 neighborhood of the noisy center pixel. For example, the regions where only center pixel is affected and no neighboring pixel is affected is classified as class one, the regions where l pixels are affected along with the center pixel are classified

into various other classes. Then, for each class, the minimum gradient amplitude $D_{min}(i, j)$ of each pixel in 3×3 neighborhood is calculated. It should be noted that while calculating minimum gradient amplitudes in 3×3 region of an image, the neighboring pixels with intensity values 0 and 255 are not taken into account. Subsequently, the noisy pixels are identified based upon the criteria that the corresponding $D_{min}(i, j)$ for each noisy pixel will be relatively larger.

In the next step, the minimum gradient amplitudes $D_{min}(i, j)$ of each pixel are arranged in the descending order and a sequence $C_{l+1}(m)$ is obtained, where $l = 0, 1, 2..., 8$. As it has already been computed that $N_{P_{l+1}}$ denotes the number of noisy center pixels where l neighboring pixels along with the center pixel are also affected, the corresponding points in $C_{l+1}(m)$ when $m \in [1, N_{P_{l+1}}]$ are termed as noisy pixels i.e., when $D_{min}(i, j) \geq C(N_{P_{l+1}})$. On that account, the *proposed* NDM can be summarized in algorithmic form as presented in Algorithm 1.

Special Case: (a) If the pixels in 3×3 regions are all 0 or all 255, then the probability that all the pixels can be 0 or 255 is calculated. If expected number of noise points corresponding to this probability are less than one, then such center pixels are termed as non-noisy. (b) If expected number of noise points corresponding to this probability are greater than one, the similar procedure is repeated by considering 5×5 regions into account. This criteria eliminates the misclassification of non-noisy pixels as noisy pixels, and hence, proves to be advantageous in images with pixels having intensity values 0 or 255, which is detailed in Section 4.2.2. (c) If all pixel values of an image are 0 or 255, i.e., if synthetic images are taken into account, the average gradient amplitudes (denoted by $D_{avg}(i, j)$) of each pixel in 3×3 region of an image (instead of minimum gradient amplitudes) are calculated. Then, $D_{avg}(i, j)$ of each pixel is arranged in descending order to form a sequence $C_s(m)$. The number of noise points for synthetic image are calculated by using the formula $N_s = A \times B \times P_n$, where P_n is the density of added noise. Therefore, the corresponding points in $C_s(m)$ when $m \in [1, N_s]$ are termed as noisy pixels i.e., when $D_{avg}(i, j) \geq C(N_s)$. This procedure helps in the detection of noisy pixels in synthetic images effectively.

Since the *proposed* NDM is based on minimum gradient amplitudes, it eliminates the need of prerequisite threshold to be determined by the user for noise detection, in contrast to the method presented in [95]. Furthermore, it avoids the misclassification of original pixels as

Algorithm 1: Proposed Noise Detection Method (NDM)

1: Take a zero matrix $z(i, j)$ of the same size as noisy image $p(i, j)$.
2: Calculate number of pixels affected in 3×3 regions of an image by computing P_{l+1} by using (6.13) and the corresponding $N_{p_{l+1}}$ (where $l = 0, 1, 2, ..., 8$) by using (6.15).
3: Calculate $D_{min}(i, j)$ of each pixel in 3×3 neighborhood of an image.
4: Arrange $D_{min}(i, j)$ in descending order and obtain $C_{l+1}(m)$ for each class.
5: If $D_{min}(i, j) \geq C(N_{P_{l+1}})$, mark the corresponding center pixel as noisy pixel i.e., $z(i, j) = 1$.
6: If all the pixels in 3×3 gets corrupted and the probability P_9 is high, repeat steps 2-5 by considering 5×5 region of an image into account.
7: Mark rest of the pixels as noise-free pixels i.e., $z(i, j) = 0$.

noisy pixels when the original image itself contains some pixels with intensity values 0 or 255. Also, the *proposed* NDM detects the noise points effectively even when high density noise is taken into account. This is in contrast to the methods presented in [93] and [94], where the authors considered the appearance of noisy pixels as small probability event, i.e., in 3×3 region of an image, at most one pixel is affected. The methods proposed in [93] and [94] are applicable only when the noise density is small i.e., $0.01 - 0.05$. However, the *proposed* method can also be applied when noise density lies in the range $10\% - 95\%$.

When cardinalities of salt and pepper pixels is unknown, an iterative approach can be followed to determine the appropriate probability P_n of noise: Initial guess of P_n is made based on number of salt and pepper pixels in the image. Using this as initial guess, denoised image is obtained. If denoised image contains sparsely populated noisy pixels, value of predicted noise density is reduced based on amount of noisy pixels This is done because if the predicted noise density is greater than the actual noise density, then the predicted number of cases where only one pixel is affected in 3×3 regions of an image will be less than the actual. Hence, these sparsely populated noisy pixels will not be denoised and will remain as it is in the output image.

Alternatively, if the denoised image contains noisy pixels which are densely populated, then the value of predicted noise density is increased based on the number of noisy pixels. This is due to the fact that if the predicted noise density is lesser than the actual noise density, then the predicted number of cases where large number of pixels in 3×3 region are affected will be less than the actual. Hence, these densely populated noisy pixels will not be processed and will remain as it is in the output image.

On iteratively following this approach, we will arrive at the noise density that is close to the actual noise density.

6.2.2 GFI-AM and GFD-AM based Image Denoising

After detecting the noise points effectively, the noisy and noise-free pixels are then processed and updated by the *proposed* GFI-AM and GFD-AM, respectively. To process the detected noisy pixels by the proposed GFI-AM, the $(2r + 1) \times (2r + 1)$ regions of the image where atleast one pixel is non-noisy are identified and the convolution is performed η number of times by utilizing the *proposed* GFI-AM-AB to get the final denoised image. Here, $(2r + 1)$ represents the size of the mask and η represents the iteration number for image denoising. In this study, $r = 1$ is chosen and the value of η depends on noise density of the image.

The rest of the pixels, i.e., the noise-free pixels are then updated by utilizing the *proposed* GFD-AM. As fractional-order differential mask is utilized only to recover the lost information due to noise and to maintain correlation, the fractional differential orders are chosen to be very small. Since the *proposed* GFD-AM is applied only to the noise-free pixels of an image, the sensitivity of parameters $\beta_1^t, \beta_2^t, \beta_1^o, \beta_2^o$ and β^e, presented in (6.11) depends on the noise-free pixels lying in the corresponding regions. Therefore, the values of these parameters can be varied based upon image characteristics and density of noise. Here, the values of the parameters are chosen to be $\beta_1^t = 0.06, \beta_2^t = 0.12, \beta_1^o = 0.13, \beta_2^o = 0.18, \beta^e = 0.22$ for the experiments. The variation in these values result in the variation of ± 1 dB in PSNR and $\pm 1.0 \times 10^{-3}$ in SSIM. These parameter values are in correspondence to the values presented in [95] and may vary based upon characteristics of the image and noise density.

In a nutshell, the *proposed* GFIFD based adaptive algorithm (**GFIFD-AA**) can be summarized as follows: Initially, the noisy pixels are identified with the help of *proposed* noise detection method. Then, the identified noisy pixels are processed and updated by utilizing the *proposed* GFI-AM. The rest of the pixels are then processed by the proposed GFD-AM based upon their respective characteristics. Hence, GFI mask is utilized for denoising of image and GFD mask is utilized for enhancement of an image, so that image features are preserved and enhanced simultaneously while noise is removed.

A step-by-step implementation of the *proposed* **GFIFD-AA** is illustrated in Algorithm 2 and the details are presented as follows:

Let $Z = [z(i,j)]_{A\times B}$ be the noisy matrix where $z(i,j) = 1$ indicates that the pixel is noisy and $z(i,j) = 0$ indicates that the pixel is non-noisy, as obtained by the *proposed* NDM detailed in Algorithm 1. Furthermore, let $W_{(2r+1)}(i,j)$ denotes a $(2r+1)\times(2r+1)$ window of an image $P = [p(i,j)]_{A\times B}$ with center (i,j) and $N_{NP}(i,j)$ represents the number of non-noisy pixels in $W_{(2r+1)}(i,j)$. For each pixel (i,j) of an image P at iteration $\eta = 0$, if

Algorithm 2: Proposed GFIFD-based Adaptive Algorithm

Input: A noisy image $P := [p(i,j)]_{A\times B}$ and noise detected matrix $Z := [z(i,j)]_{A\times B}$
Output: A denoised and enhanced image $Q := [q(i,j)]_{A\times B}$

Initialize: $\eta \leftarrow 0$; $Q^{[0]} \leftarrow P$; $r \leftarrow 1$; $\epsilon \leftarrow 0$;
for *each pixel* (i,j) *of image* $Q^{[\eta]}$ *at iteration* η **do**
  if $z(i,j) = 1$ **then**
    if $N_{NP}^{[\eta]}(i,j) > 0$ **then**
      Define: GFI mask $M_{2r+1}^{GFI-AM}(i,j)$ using (5.10)
      Update: $q^{[\eta+1]}(i,j) = conv(W_{(2r+1)}^{[\eta]}(i,j), M_{2r+1}^{GFI-AM}(i,j))$
    else
      Assign: $q^{[\eta+1]}(i,j) = q^{[\eta]}(i,j)$;
    end
  end
  Until: $|Q^{[\eta+1]} - Q^{[\eta]}| \le \epsilon$;
end
Denoised Image: Q
for *each pixel* (i,j) *of denoised image* Q **do**
  if $z(i,j) = 0$ **then**
    Define: GFD mask $M_{2r+1}^{GFD-AM}(i,j)$ using (6.9)
    Update: $q(i,j) = conv(W_{(2r+1)}(i,j), M_{2r+1}^{GFD-AM}(i,j))$
  end
end
Denoised and Enhanced Image: Q

$z(i,j) = 1$, then number of non-noisy pixels in its $(2r+1)\times(2r+1)$ neighborhood are calculated. If $N_{NP}(i,j) > 0$, the value of noisy pixel is updated by the convolution ($conv(\cdot,\cdot)$) of $W_{(2r+1)}(i,j)$ with GFI-AM mask (herein, referred to as $M_{(2r+1)}^{GFI-AM}(i,j)$). If $N_{NP}(i,j) = 0$, then value of that noisy pixel is kept unchanged and is processed in the next iteration. Subsequently, in the next iteration, the noisy pixels are updated by utilizing the image $P^{[\eta]}$ obtained in the previous iteration. This procedure is repeated until all the noisy pixels are processed and the condition $|P^{[\eta+1]} - P^{[\eta]}| \le \epsilon$ is met. Here, the value of $\epsilon = 0$ is set if integer values of pixels are considered. If gray values of the pixels are taken to be real numbers, ϵ can be set as small as 10^{-5}. After that, the denoised image is obtained.

The non-noisy pixels (initially detected by NDM) of the denoised image are then processed

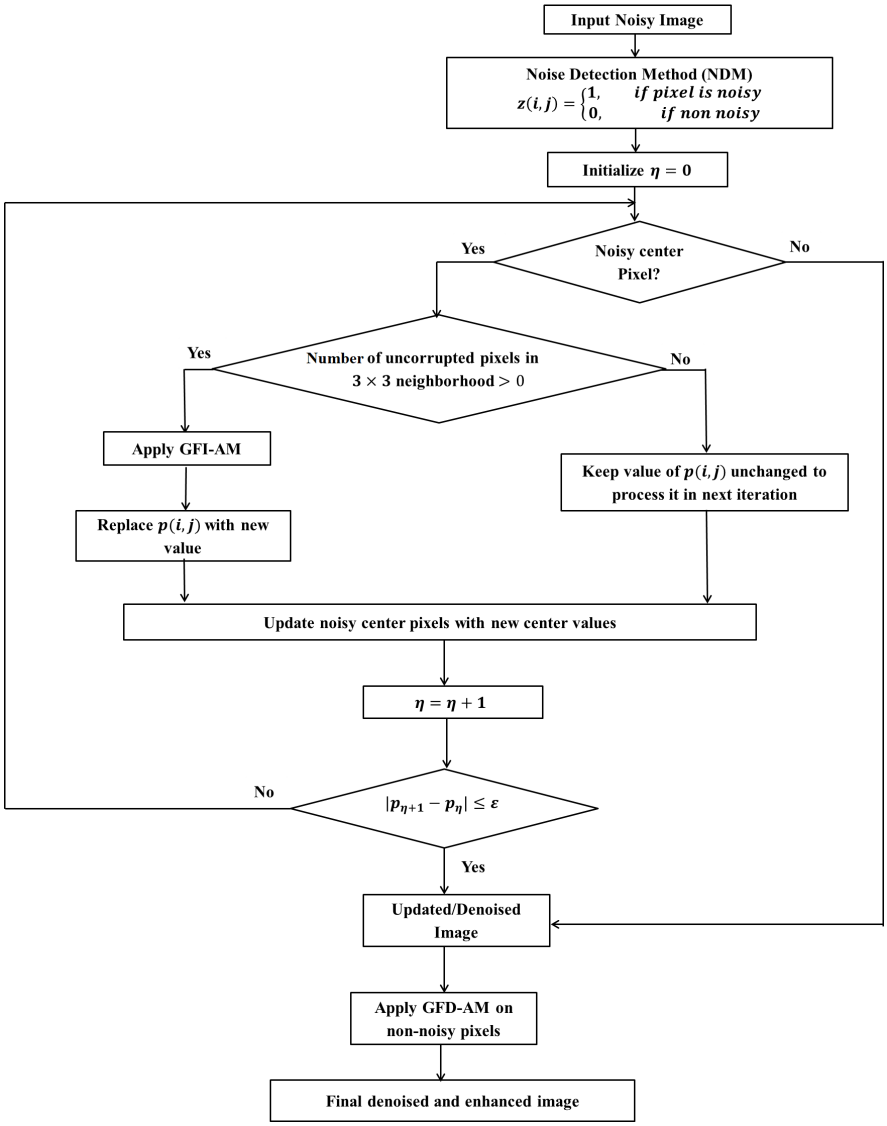


Figure 6.2: Flowchart of the proposed GFIFD-AA

by the proposed GFD-AM (herein, referred to as $M_{(2r+1)}^{GFD-AM}(i,j)$) so as to enhance the texture and edge details of the image. Figure 6.2 presents the flowchart of the *proposed* method.

6.3 Performance Analysis of the Proposed GFIFD-AA

After discussing the flowchart of the proposed GFIFD-AA, the next aim is to test the performance of the proposed method on images affected by salt and pepper (SP) noise of various densities and then to validate its performance against well-established state-of-the-art methods. In the literature, several traditional methods which include median filter (MF) [127] and

its variants such as non-local mean filter [128], weighted mean filter [129], adaptive switching based filters [130, 131], based on pixel density filter (BPDF) [104], different applied median filter (DAMF) [100] etc., have been applied to deal with SP noise effectively.

BPDF [104] method is highly effective in removing low-density noise. However, its effectiveness reduces as medium and high-density noise is taken into account. It causes raindrop effect and destroys the image structure when high density noise is present in images. On the other hand, DAMF [100] performs competently for all noise densities. But when images with boundary values are taken into account, its effectiveness reduces due to lack of accuracy in detecting noisy pixels. Furthermore, some other methods which include adaptive frequency median filter (AFMF) [141], two-stage filter (TSF) [105], adaptive Riesz mean filter (ARmF) [101], iterative mean filter (IMF) [102], noise accumulation and harmonic analysis technique (NAHAT) [103] have been proposed in the literature to deal with high-density salt and pepper noise effectively.

Although these methods can remove noise efficiently and possess several advantages, the preservation of edge and texture details of an image still remains a challenging issue. On the other hand, most of the methods in the literature assume all pixels with intensity values 0 and 255 as noisy [100, 101, 102, 103]. Therefore, these methods misclassify non-noisy pixels (of values 0 or 255) as noisy pixels. As a result, the performance of these methods get degraded when images containing pixels with intensity values 0 or 255 are taken into account. On the other hand, some of the traditional methods adopt integer-order integrals directly or indirectly during the construction of a denoising model. On the contrary, fractional-order filters have been proved to be advantagenous because of their ability to preserve the information while denoising the image [64].

Most of the traditional fractional image denoising masks (for example, GL [13], RL [13], IGL [132], AFD [133], AFI [133], fractional K-operator [136], and recently proposed ABC mask [91]) employ a fixed fractional-order to deal with the noise present in the images. However, the characteristics of an image varies locally. Hence, as stated previously, it is difficult to obtain better image denoising effect by using a fixed fractional-order to process noisy pixels in distinct regions. This is due to the fact that if higher fractional-order is chosen, the texture details are lost. On the other hand, if lower fractional-order is chosen, denoising performance is degraded. Therefore, an attempt has been made in [93] to design

a global adaptive fractional integral algorithm (GAFIA) and adaptive fractional calculus of small probability strategy (AFC-SPS) to deal with the noise adaptively. Furthermore, an adaptive algorithm based on entropy and gradient features (EN-AFC) has been proposed in [94] for the removal of salt and pepper noise in images. An adaptive algorithm based on fractional K-operator has also been designed in [95]. Although these methods can remove noise adaptively, the methods presented in [93] and [94] are applicable only when the noise density is small i.e., 0.01-0.05. Furthermore, the methods presented in [93, 94, 95] are based upon conventional fractional-order operators. On the other hand, the recently introduced ABC fractional integral based adaptive mask [91] has only been applied on images affected by Gaussian noise.

Therefore, GFIFD-AA has been put forward for images affected with salt and pepper noise. The *proposed* method not only extends the range of noise density considered in [93] and [94] to 10%-95%, but also incorporates recently introduced definitions of fractional-order operators in generalized framework in contrast to the ones presented in [93, 94, 95].

To validate the performance of the *proposed* method, the datasets named Set12 and BSD68 [126] consisting of 12 and 68 images, respectively and biomedical images of different characteristics are taken into account. Initially, the image affected by various levels of noise density is taken into account. Then, different images from the standard datasets are considered to show the robustness of the proposed method. Furthermore, the performance of the *proposed* GFIFD-AA method is validated and compared against well-established methods in the literature, presented in [127, 93, 94, 104, 100, 141, 105, 101, 102] and [103].

6.3.1 Based on Various Levels of Noise Density

To show the effectiveness of the *proposed* method, an MRI medical image corrupted with SP noise at various levels of noise density (i.e., 10%-90%) is considered. Figure 6.3 depicts the denoising performance of the *proposed* GFIFD-AA along with the values of PSNR and SSIM. Figures 6.3a, 6.3b, 6.3c, 6.3d and 6.3e represent the images affected by SP noise at $10\%, 30\%, 50\%, 70\%, 90\%$ noise densities, respectively. The corresponding denoised images attained by the proposed GFIFD-AA are presented in Figures 6.3f, 6.3g, 6.3h, 6.3i and 6.3j, respectively.

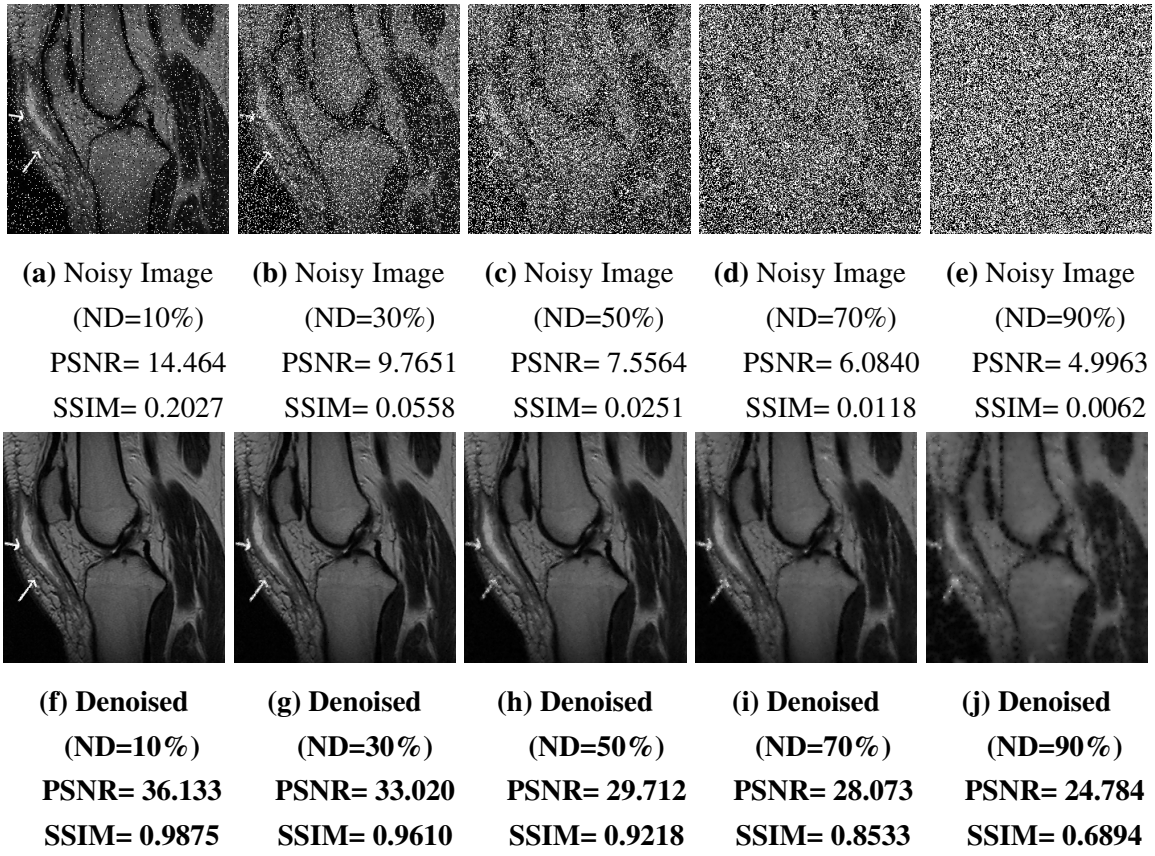


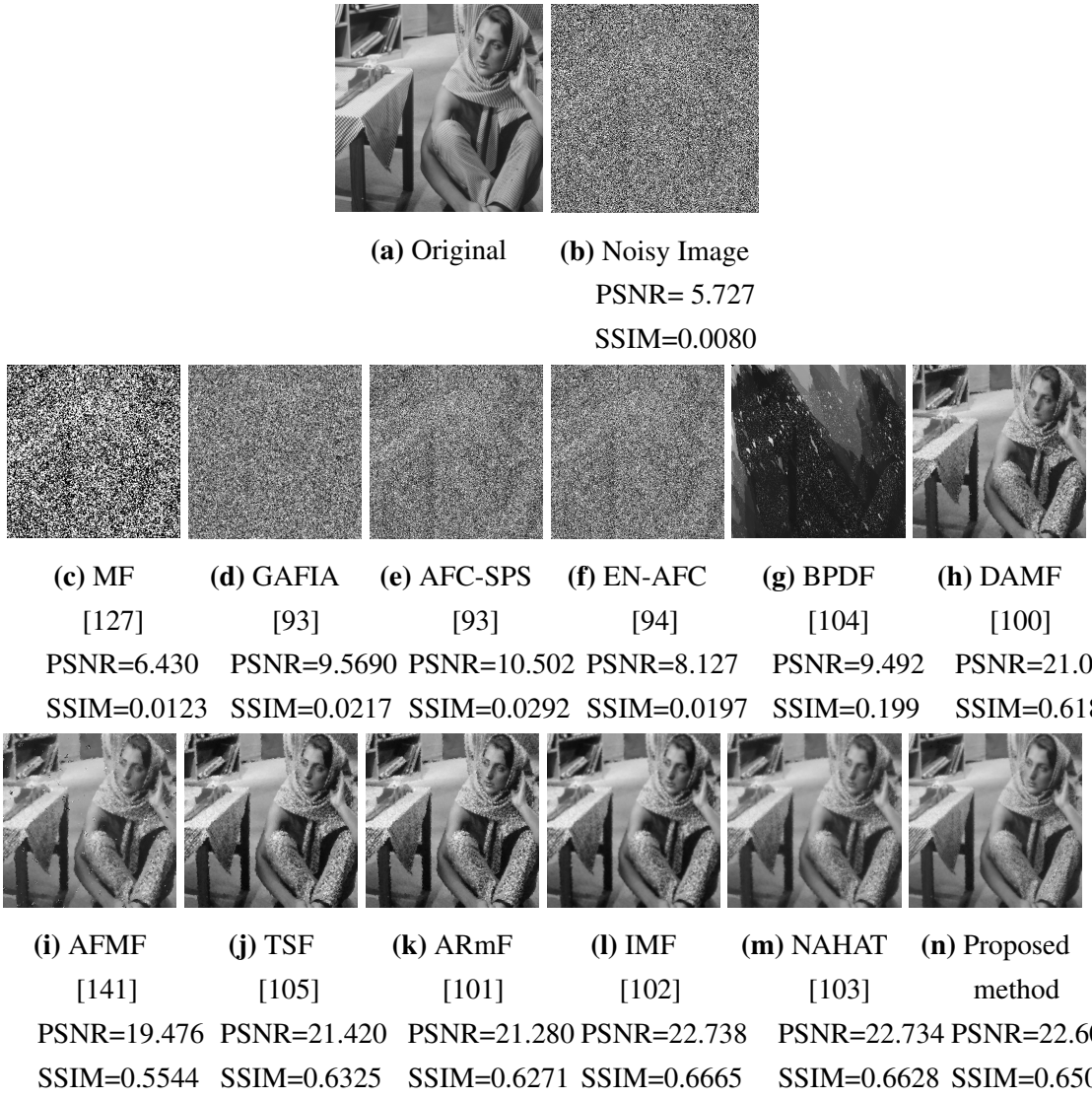
Figure 6.3: (a-e) Noisy MRI images corrupted by SP noise at various noise densities (ND) (f-j) Corresponding denoised images obtained by utilizing proposed GFIFD-AA along with the values of PSNR and SSIM.

From Figure 6.3, it can be seen that the *proposed* GFIFD-AA not only removes the low-density SP noise but also removes the affect of the noise effectively even when the image is corrupted with noise at density 90%. This illustrates the effectiveness of the *proposed* **GFIFD-AA** in removing SP noise at various levels of noise density.

6.3.2 On Images with Different Characteristics

In this section, a standard image from Set12 dataset is considered to show the performance of the proposed method against well-established state-of-the-art methods. Figure 6.4 presents the denoised images obtained by utilizing various methods on Image 9 of Set12 dataset corrupted by SP noise at 90% noise density.

From the results, it can be seen that the median filter [127], GAFIA [93], AFC-SPS [93], EN-AFC [94] methods are unable to deal with the high density noise. However, BPDF [104]



(a) Original **(b)** Noisy Image PSNR= 5.727 SSIM=0.0080

(c) MF [127] PSNR=6.430 SSIM=0.0123 **(d)** GAFIA [93] PSNR=9.5690 SSIM=0.0217 **(e)** AFC-SPS [93] PSNR=10.502 SSIM=0.0292 **(f)** EN-AFC [94] PSNR=8.127 SSIM=0.0197 **(g)** BPDF [104] PSNR=9.492 SSIM=0.199 **(h)** DAMF [100] PSNR=21.075 SSIM=0.6184

(i) AFMF [141] PSNR=19.476 SSIM=0.5544 **(j)** TSF [105] PSNR=21.420 SSIM=0.6325 **(k)** ARmF [101] PSNR=21.280 SSIM=0.6271 **(l)** IMF [102] PSNR=22.738 SSIM=0.6665 **(m)** NAHAT [103] PSNR=22.734 SSIM=0.6628 **(n)** Proposed method PSNR=22.606 SSIM=0.6502

Figure 6.4: Denoising performance of various techniques on Image 9 of Set12 dataset corrupted by SP noise at 90% noise density.

method removes the noise to some extent but the information is lost to a great extent. It causes a raindrop effect and hence, proves to be ineffective in dealing with high-density SP noise.

Furthermore, DAMF [100] and AFMF [141] methods eliminate the noise to a greater extent but the denoised image obtained by AFMF [141] still contains some noise points, as can be visualised from Figure 6.4i. Furthermore, they fail to preserve the texture details of the image properly. Furthermore, TSF [105] and ARmF [101] proves to be effective in removing high-density SP noise. However, the denoising results contain some distortions, especially in the bottom-right region of the image.

On the other hand, IMF [102], NAHAT [103], and the *proposed* method are able to remove the noise effectively. The PSNR and SSIM values obtained by the *proposed* method are better than MF [127], GAFIA [93], AFC-SPS [93], EN-AFC [94], BPDF [104], DAMF [100], AFMF [141], TSF [105], ARmF [101] and are comparable to the values obtained by IMF [102] and NAHAT [103] methods. Therefore, it can be said that the *proposed* method performs better than most of the discussed methods. This is evident from the values of PSNR and SSIM obtained by the *proposed* method. However, the results of the *proposed* method are comparable to that of IMF [102] and NAHAT [103] methods.

6.4 Comparative Analysis

To further illustrate the effectiveness of the proposed method against well-established state-of-the-art methods, various images such as images from standard datasets, black and white images, and synthetic images are taken into account. These images are affected by SP noise at various noise densities and performance of different methods is analyzed based upon several quantitative metrics.

6.4.1 Performance Metrics

Apart from PSNR and SSIM, various other well-known performance metrics such as, mean absolute error (MAE), mean structural similarity index measure (MSSIM), and entropy are taken into account to illustrate the effectiveness of the proposed method.

Mean absolute error between the original image p and denoised image q is calculated mathematically as

$$MAE = \frac{1}{AB} \sum_{i=0}^{A-1} \sum_{j=0}^{B-1} |p(i,j) - q(i,j)| \tag{6.18}$$

A mean structural similarity index measure (MSSIM) is utilized to evaluate the overall image quality [138]. It is defined as:

$$MSSIM(p,q) = \frac{1}{H} \sum_{i=1}^{H} SSIM(p_i, q_i) \tag{6.19}$$

where p and q represent the original and denoised images, respectively. p_i and q_i are the image contents at the i^{th} local window, and H represents the number of local windows of the image [138].

Entropy is defined as a statistical measure of randomness, which characterizes the texture of an image. It is calculated over entire image by utilizing histogram of the image [97, 63]. Entropy of an image is defined as

$$Entropy = -\sum_{i=0}^{255} \rho_i log_2 \rho_i \tag{6.20}$$

where ρ_i represents the probability associated with gray level i. The closer the value of entropy to the original image, the better the preservation of details will be.

6.4.2 Simulation Results and Comparison

To show the effectiveness of the proposed GFIFD-AA method, several images from standard datasets as well as synthetic images are taken into account. The results are validated in terms of various quantitative parameters, which include PSNR, SSIM, MAE, MSSIM, and entropy.

6.4.2.1 On Images from Standard Datasets

To validate the performance of the proposed method, Image 9 from Set12 dataset is corrupted with SP noise at various levels of noise density ranging from 10% to 95%. The quantitative comparison in terms of PSNR, SSIM, MAE, and MSSIM and entropy among various methods and the *proposed* **GFIFD-AA** is detailed in Table 6.3.

Table 6.3: Quantitative comparison for Image 9 of Set12 dataset corrupted by SP noise at various noise densities.

ND	Parameters	Input	MF [127]	GAFIA [93]	AFC-SPS [93]	EN-AFC [94]	BPDF [104]	DAMF [100]	AFMF [141]	TSF [105]	ARmF [101]	IMF [102]	NAHAT [103]	Proposed
10 %	PSNR	15.275	24.703	21.205	22.85	23.417	32.615	33.804	28.835	33.804	34.792	34.262	35.392	34.211
	SSIM	0.2704	0.7967	0.4678	0.5408	0.6086	0.9736	0.9811	0.9323	0.9811	0.9839	0.9822	0.9854	0.982
	MAE	5.8779	3.5942	4.1706	3.9561	4.6149	0.6218	0.4397	1.0817	0.4397	0.4116	0.4367	0.3794	0.4393
	MSSIM	0.4744	0.5064	0.5112	0.5212	0.5519	0.9367	0.9541	0.8107	0.9539	0.9582	0.9548	0.9598	0.9551
	Entropy*	7.6321	7.593	7.6944	7.6593	7.7685	7.6291	7.6288	7.6163	7.6288	7.628	7.6279	7.6282	7.6334
20 %	PSNR	12.267	23.461	17.999	19.634	20.411	29.3	30.51	28.1	30.51	31.542	31.07	31.954	30.971
	SSIM	0.1485	0.7516	0.3031	0.3662	0.4539	0.942	0.9584	0.9226	0.9584	0.965	0.9616	0.9673	0.9597
	MAE	11.609	4.09	7.4601	6.7281	7.0639	1.2927	0.9083	1.2542	0.9083	0.8388	0.8892	0.7923	0.905
	MSSIM	0.229	0.4719	0.2831	0.303	0.3791	0.8655	0.9007	0.8015	0.9002	0.9112	0.9047	0.9138	0.9042
	Entropy	7.6321	7.6064	7.7401	7.6801	7.8089	7.6248	7.6251	7.6194	7.6251	7.6239	7.6229	7.6237	7.6386
30 %	PSNR	10.496	20.835	15.894	17.416	18.298	27.257	28.422	27.092	28.422	29.515	29.11	29.795	29.022
	SSIM	0.095	0.6246	0.2093	0.2584	0.3539	0.9047	0.9319	0.905	0.9319	0.9431	0.9379	0.9455	0.9375
	MAE	17.647	5.1478	11.005	9.757	9.5966	2.0301	1.413	1.5207	1.413	1.2926	1.3688	1.247	1.3725
	MSSIM	0.1161	0.4142	0.1691	0.1897	0.2795	0.7859	0.8407	0.7706	0.8399	0.8591	0.8495	0.8616	0.8511
	Entropy	7.6321	7.6169	7.7709	7.6884	7.8333	7.6207	7.6214	7.6196	7.6214	7.6196	7.6178	7.6192	7.6365
40 %	PSNR	9.25	17.614	14.354	15.764	16.716	25.597	26.919	26.123	26.92	27.997	27.728	28.254	27.63
	SSIM	0.0663	0.427	0.1522	0.1899	0.2835	0.8589	0.9019	0.8813	0.9019	0.9174	0.9111	0.9202	0.9085
	MAE	23.385	7.0066	14.213	12.46	11.951	2.9242	1.9767	1.8704	1.977	1.8042	1.8934	1.7616	1.9222
	MSSIM	0.0648	0.3376	0.1092	0.1265	0.2144	0.6959	0.7728	0.7245	0.772	0.7983	0.7877	0.802	0.7874
	Entropy	7.6321	7.5843	7.7955	7.6952	7.8413	7.6169	7.6199	7.6211	7.6199	7.6168	7.6127	7.6146	7.6343
50 %	PSNR	8.279	14.587	13.028	14.325	15.349	24.125	25.687	25.079	25.695	26.654	26.599	26.989	26.526
	SSIM	0.0458	0.2381	0.1071	0.135	0.224	0.8031	0.8671	0.8478	0.8673	0.8858	0.8804	0.8897	0.8772
	MAE	29.499	10.732	17.872	15.643	14.418	3.9244	2.5045	2.2419	2.5035	2.2901	2.3627	2.2425	2.4043
	MSSIM	0.039	0.2396	0.0715	0.0841	0.1642	0.5982	0.6975	0.6584	0.6967	0.7286	0.7203	0.7342	0.7187
	Entropy	7.6321	7.4233	7.81	7.7038	7.83	7.6097	7.6187	7.6215	7.6186	7.6153	7.608	7.6097	7.6377
60 %	PSNR	7.468	11.904	11.963	13.154	13.993	22.478	24.616	24.085	24.64	25.4	25.622	25.884	25.579
	SSIM	0.0324	0.1174	0.0778	0.0991	0.1714	0.7269	0.8278	0.8067	0.8283	0.8468	0.844	0.8532	0.8415
	MAE	35.214	16.191	20.879	18.221	16.981	5.6105	3.1836	2.7935	3.1776	2.9528	2.9778	2.8709	3.022
	MSSIM	0.0242	0.1404	0.047	0.0558	0.1212	0.483	0.6152	0.5785	0.615	0.6465	0.6444	0.6576	0.6435
	Entropy	7.6321	7.0275	7.8076	7.7048	7.7905	7.5984	7.6155	7.6193	7.6154	7.6128	7.6005	7.6027	7.639
70 %	PSNR	6.821	9.727	11.097	12.196	12.537	20.474	23.527	23.12	23.599	24.053	24.664	24.843	24.648
	SSIM	0.0222	0.0578	0.0545	0.07	0.1177	0.6236	0.7773	0.7538	0.7795	0.7942	0.7991	0.8065	0.796
	MAE	41.091	24.142	24.022	20.976	20.136	8.372	3.9719	3.4481	3.9487	3.7782	3.6562	3.6031	3.7411
	MSSIM	0.0153	0.064	0.0303	0.0361	0.0816	0.349	0.5155	0.4781	0.5171	0.5442	0.5564	0.566	0.5521
	Entropy	7.6321	6.2615	7.8108	7.7284	7.6847	7.5785	7.6157	7.6194	7.6154	7.6131	7.5946	7.5947	7.6327
80 %	PSNR	6.245	7.934	10.305	11.319	10.591	16.283	22.483	22.208	22.668	22.8	23.837	23.93	23.798
	SSIM	0.015	0.0301	0.0378	0.0492	0.0652	0.4443	0.7158	0.6876	0.7224	0.7285	0.7467	0.7502	0.7353
	MAE	46.941	34.339	27.003	23.598	25.245	18.038	4.8244	4.2083	4.7662	4.6813	4.3499	4.3859	4.5604
	MSSIM	0.009	0.0232	0.018	0.0213	0.0444	0.1873	0.4008	0.3609	0.4076	0.4221	0.4567	0.4598	0.4436
	Entropy	7.6321	5.0782	7.7922	7.7446	7.3131	7.4905	7.6193	7.6187	7.6172	7.6177	7.5896	7.5865	7.623
90 %	PSNR	5.727	6.43	9.569	10.502	8.127	9.492	21.075	19.476	21.42	21.28	22.738	22.734	22.606
	SSIM	0.008	0.0123	0.0217	0.0292	0.0197	0.199	0.6184	0.5544	0.6325	0.6271	0.6665	0.6628	0.6502
	MAE	53.043	46.842	30.237	26.533	36.386	63.525	6.1421	5.9484	5.9899	6.0612	5.4307	5.5775	5.7533
	MSSIM	0.0036	0.0048	0.0073	0.0087	0.0112	0.0312	0.2535	0.2182	0.2691	0.2647	0.3304	0.3254	0.3112
	Entropy	7.6321	3.3861	7.718	7.713	5.9942	5.346	7.6129	7.6127	7.6046	7.6112	7.5668	7.5549	7.5677
95 %	PSNR	5.487	5.815	9.22	10.113	6.734	9.082	18.362	14.201	20.209	20.165	21.682	21.709	21.203
	SSIM	0.0057	0.0072	0.0155	0.0213	0.0086	0.2027	0.4884	0.317	0.5462	0.5455	0.6059	0.5917	0.5619
	MAE	55.85	52.931	31.564	27.711	45.406	68.639	8.0933	11.502	7.4005	7.2786	6.3803	6.7714	7.2791
	MSSIM	0.0019	0.0011	0.0035	0.0039	0.0031	0.0134	0.1507	0.1118	0.167	0.162	0.2512	0.2343	0.2079
	Entropy	7.6321	2.3466	7.4739	7.4751	4.3809	4.355	7.6257	7.415	7.6082	7.6137	7.5762	7.5389	7.5472

* Input presents entropy values of original image instead of noisy image.

From the results tabulated in Table 6.3, it can be observed that the *proposed* method outperforms the conventional methods such as MF [127], GAFIA [93], AFC-SPS [93], EN-AFC [94], BPDF [104], DAMF [100], AFMF [141], and TSF [105] in terms of highest values of PSNR, SSIM, MSSIM and lowest values of MAE obtained by the *proposed* method. On the other hand, the values of PSNR, SSIM, and MSSIM obtained by ARmF [101], IMF [102], NAHAT [103] methods are slightly higher than that of the *proposed* method. However, the results for ARmF [101] method are only better for low and medium density of SP noise. As the noise density is increased beyond 60%, the *proposed* method outperforms the ARmF [101] method.

Furthermore, the performance of various methods is also analysed on the basis of entropy values obtained by them. From entropy values tabulated in Table 6.3, it can be seen that the entropy values obtained by the *proposed* method are close to that of the original image. This is where the advantage of utilizing GFD-AM for non-noisy pixels lies. GFD-AM is applied on non-noisy pixels in the *proposed* method so as to enhance the information present in the image. On the other hand, the entropy values obtained by other methods such as MF [127], GAFIA [93], AFC-SPS [93], EN-AFC [94] are highly inconsistent with the noise density. Furthermore, the entropy values attained by BPDF [104], DAMF [100], AFMF [141], TSF [105], ARmF [101], IMF [105] and NAHAT [103] are less than that of original image. Therefore, it can be said that the proposed method is effective in preserving the details of the image while removing the noise simultaneaously. However, if the noise density is increased beyond 80%, it is difficult to preserve the details of the image effectively due to very high level of noise. This is valid for all the methods discussed in this chapter.

To further illustrate the robustness of the *proposed* method on images with different characteristics, all images of Set12 dataset are considered and are affected by SP noise at density 90%. Table 6.4 illustrates the PSNR, SSIM, and MSSIM values of all the denoised images obtained by utilizing different methods. The entropy values are not taken into account because the level of noise considered is very high.

From the results, it can be seen that the *proposed* method attains highest values of PSNR, SSIM, and MSSIM for all the images affected by SP noise at density 90% against that of well-established methods such as MF [127], GAFIA [93], AFC-SPS [93], EN-AFC [94], BPDF [104], DAMF [100], AFMF [141], TSF [105] and ARmF [101]. However, the results

Table 6.4: Quantitative comparison among different methods for all images of Set12 dataset corrupted by SP noise at 90% noise density.

No.	Parameters	Noisy	MF [127]	GAFIA [93]	AFC-SPS [93]	EN-AFC [94]	BPDF [104]	DAMF [100]	AFMF [141]	TSF [105]	ARmF [101]	IMF [102]	NAHAT [103]	Proposed
1	PSNR	5.5556	6.2944	9.2066	7.534	7.8573	11.134	20.438	18.593	20.62	20.701	21.704	21.626	21.369
	SSIM	0.0097	0.0169	0.0241	0.0188	0.019	0.449	0.6966	0.626	0.705	0.707	0.721	0.713	0.6969
	MSSIM	0.0047	0.0064	0.0087	0.0088	0.0086	0.054	0.1867	0.155	0.188	0.197	0.208	0.2	0.1861
2	PSNR	5.9428	6.6889	10.0426	8.0785	8.4644	14.409	24.729	20.522	24.85	25.125	26.211	25.583	25.508
	SSIM	0.0075	0.0133	0.0224	0.0169	0.0186	0.4597	0.7459	0.657	0.757	0.758	0.789	0.776	0.7652
	MSSIM	0.0022	0.0041	0.0054	0.0057	0.0065	0.0549	0.1843	0.153	0.197	0.199	0.228	0.217	0.2019
3	PSNR	5.7802	6.5187	9.687	7.8318	8.2013	9.8789	21.268	18.773	21.335	21.445	22.284	21.998	21.791
	SSIM	0.0082	0.0216	0.0241	0.0191	0.0203	0.2091	0.7095	0.63	0.715	0.718	0.761	0.743	0.7298
	MSSIM	0.0018	0.0058	0.0048	0.0058	0.0063	0.0337	0.3134	0.259	0.32	0.325	0.401	0.376	0.3577
4	PSNR	5.5426	6.2306	9.1487	7.5352	7.8658	10.914	21.252	18.349	21.385	21.505	22.616	22.296	22.000
	SSIM	0.0113	0.0246	0.0319	0.0261	0.0277	0.2424	0.66	0.553	0.664	0.672	0.705	0.686	0.664
	MSSIM	0.0049	0.0118	0.0124	0.0126	0.0148	0.0465	0.3527	0.274	0.356	0.369	0.439	0.413	0.3865
5	PSNR	5.7362	6.4505	9.576	7.7947	8.1495	10.224	20.029	17.331	20.112	20.271	21.312	20.729	20.32
	SSIM	0.0131	0.0289	0.0346	0.0316	0.0331	0.2626	0.7085	0.609	0.715	0.719	0.751	0.724	0.7174
	MSSIM	0.0053	0.0125	0.0119	0.0142	0.0159	0.0453	0.3282	0.264	0.336	0.341	0.413	0.383	0.3585
6	PSNR	5.3848	6.1647	8.9026	7.367	7.6683	9.1951	20.188	18.491	20.329	20.406	21.424	21.439	21.207
	SSIM	0.0106	0.0282	0.0282	0.0229	0.0209	0.1683	0.6797	0.606	0.685	0.688	0.71	0.703	0.6863
	MSSIM	0.0044	0.0111	0.0096	0.0102	0.0109	0.0288	0.2291	0.189	0.234	0.238	0.271	0.26	0.2407
7	PSNR	5.3843	6.0765	8.8384	7.3074	7.6159	9.1218	19.467	17.039	19.505	19.75	20.676	20.494	20.196
	SSIM	0.0098	0.0183	0.024	0.0212	0.0209	0.2528	0.7206	0.618	0.726	0.73	0.746	0.734	0.7174
	MSSIM	0.0043	0.0079	0.0082	0.0096	0.0093	0.0373	0.3295	0.255	0.33	0.343	0.394	0.368	0.3478
8	PSNR	5.902	6.645	9.9695	8.0424	8.4202	10.827	25.858	21.722	26.083	26.123	27.342	26.945	26.568
	SSIM	0.0058	0.0115	0.0188	0.0135	0.0149	0.2888	0.7642	0.683	0.771	0.773	0.809	0.798	0.7869
	MSSIM	0.0011	0.0034	0.0031	0.0036	0.0041	0.037	0.2696	0.23	0.275	0.281	0.349	0.334	0.3182
9	PSNR	5.7271	6.4304	9.5692	7.7699	8.1271	9.4921	21.075	19.476	21.42	21.28	22.738	22.734	22.606
	SSIM	0.008	0.0123	0.0217	0.0179	0.0197	0.199	0.6184	0.554	0.632	0.627	0.666	0.663	0.6496
	MSSIM	0.0036	0.0048	0.0073	0.0094	0.0112	0.0312	0.2535	0.218	0.269	0.265	0.33	0.325	0.3112
10	PSNR	5.9296	6.6585	10.028	8.0331	8.4327	11.59	23.241	20.889	23.408	23.538	24.557	24.479	24.188
	SSIM	0.0078	0.0159	0.0229	0.0177	0.0196	0.2788	0.6429	0.567	0.647	0.654	0.683	0.679	0.6634
	MSSIM	0.0026	0.0062	0.0061	0.007	0.0082	0.0354	0.2305	0.189	0.229	0.242	0.278	0.273	0.2586
11	PSNR	5.8451	6.561	9.8344	7.9474	8.3141	10.755	24.344	21.298	24.568	24.576	25.72	25.522	25.164
	SSIM	0.0062	0.0114	0.0191	0.0148	0.0163	0.2383	0.6831	0.599	0.69	0.692	0.72	0.712	0.692
	MSSIM	0.0016	0.0036	0.004	0.0052	0.0062	0.0287	0.2749	0.216	0.279	0.287	0.32	0.311	0.2892
12	PSNR	5.9795	6.7043	10.121	8.0817	8.4862	10.667	23.126	20.406	23.299	23.389	24.436	24.366	24.075
	SSIM	0.0078	0.0141	0.0227	0.0174	0.0192	0.193	0.6227	0.535	0.629	0.637	0.662	0.661	0.6416
	MSSIM	0.0027	0.0056	0.0063	0.0071	0.0084	0.0328	0.2375	0.183	0.242	0.254	0.277	0.278	0.2575

obtained by IMF [102] and NAHAT [103] are slightly better than the *proposed* method but are comparable to that of the *proposed* method. This validates the effectiveness and robustness of the *proposed* method on images with different characteristics and also in comparison to most of the well-established state-of-the-art methods.

The reason why existing IMF [102] and NAHAT [103] methods perform better than the *proposed* method can be explained as follows: The existing IMF approach utilizes the mean of gray values of noise-free pixels to attain the denoising image in an iterative manner. On the other hand, in NAHAT approach, the image is considered as a harmonic function and the

smoothness of the harmonic function is obtained by satisfying the Laplace equation. Therefore, the approach utilizes Laplace equation along with boundary conditions to obtain the gray values of the noisy pixels from the non-noisy pixels.

In conclusion, both of these approaches exclude neighboring noisy pixels while computing the gray values of the noisy pixels, i.e., only the gray values of non-noisy pixels are utilized to denoise the image. On the other hand, in the proposed approach, the generalized fractional integral mask is applied on all the pixels in the vicinity of the noisy center pixel in the 3×3 neighborhood region, irrespective of the pixels in 3×3 region being noisy or non-noisy to obtain the gray value of the noisy pixel.

Therefore, these methods perform slightly better than the proposed method for natural images. However, these methods lack robust noise detection algorithm and hence, their performance degrades when black and white images (i.e., images with original intensity values of 0 or 255) or synthetic images are taken into account, as detailed in the subsequent subsections.

6.4.2.2 On Black and White Images

Furthermore, to authenticate the superiority of the *proposed* NDM as well as the *proposed* GFIFD-AA, one image from BSD68 [126] dataset, where some of the pixels in the original image itself have the intensity value 255 is considered. The original image, the corresponding noisy image and the denoised images obtained by utilizing various methods are presented in Figure 6.5 along with the values of PSNR and SSIM.

From the results, it can be seen that the conventional BPDF [104], DAMF [100], AFMF [141], ARmF [101] methods remove the noise only up to certain degree. The output of these methods get distorted in the regions where original image contains the pixels with intensity value 255. The noise points in the images obtained by these methods are clearly visible, as depicted in Figures 6.5c, 6.5d, 6.5e, and 6.5g. On the other hand, the methods such as TSF [105], IMF [102], NAHAT [103], and *proposed* GFIFD-AA prove to be effective in restoring original pixels with intensity value 255 and removing the noise simultaneously. Furthermore, from the results, it can be inferred that the *proposed* method outperforms TSF [105], ARmF [101], and IMF [102] methods in terms of highest values of PSNR and SSIM. However, the results of the *proposed* method are comparable to that of the NAHAT [103]. Therefore, it can be said that the *proposed* method exhibits significant advantages over that of conventional

(a) Original Image

(b) Noisy Image PSNR= 7.2260 SSIM= 0.0667

(c) BPDF [104] PSNR= 21.428 SSIM= 0.7812

(d) DAMF [100] PSNR= 17.276 SSIM= 0.7736

(e) AFMF [141] PSNR= 15.957 SSIM= 0.7240

(f) TSF [105] PSNR= 23.649 SSIM= 0.8448

(g) ARmF [101] PSNR= 21.451 SSIM= 0.8405

(h) IMF [102] PSNR= 24.203 SSIM= 0.8524

(i) NAHAT [103] PSNR= 24.736 SSIM= 0.8691

(j) Proposed PSNR= 24.290 SSIM= 0.8579

Figure 6.5: Denoising performance of various techniques on Image 68 of BSD68 dataset corrupted by SP at 50% noise density.

methods for removing SP noise in images containing original pixels with intensity values 0 or 255.

Furthermore, a chessboard image is also taken into consideration to validate the robustness of the *proposed* GFIFD-AA. The chessboard image contains some pixels of intensity values 0 and 255 alongwith sharp edges. The image is then corrupted with SP noise at density 50%. Then, the results are compared against best performing conventional methods from the previous study to validate the effectiveness of the *proposed* method. Figure 6.6 illustrates the denoising performance of various methods on chessboard image affected with SP noise.

From the results, it can be seen that all the conventional methods prove to be ineffective in this case. Although BPDF [104], DAMF [100], AFMF [141], TSF [105], ARmF [101] methods are able to reduce noise to some extent, but they are not able to retain the information present in the image. BPDF causes a raindrop effect and highly distorts the image. DAMF [100], AFMF [141], and ARmF [101] are unable to remove all the noisy pixels present in the image and hence, prove to be ineffective for images containing white and black regions along with sharp edges. On the other hand, TSF [105], IMF [102] and NAHAT [103] methods remove the noise to a great extent. However, from the results presented in Figure 6.6f, 6.6h and 6.6i, it can be seen that the black region gets spread out in the white region. This is because some of the white regions in the image have almost all the pixels with intensity

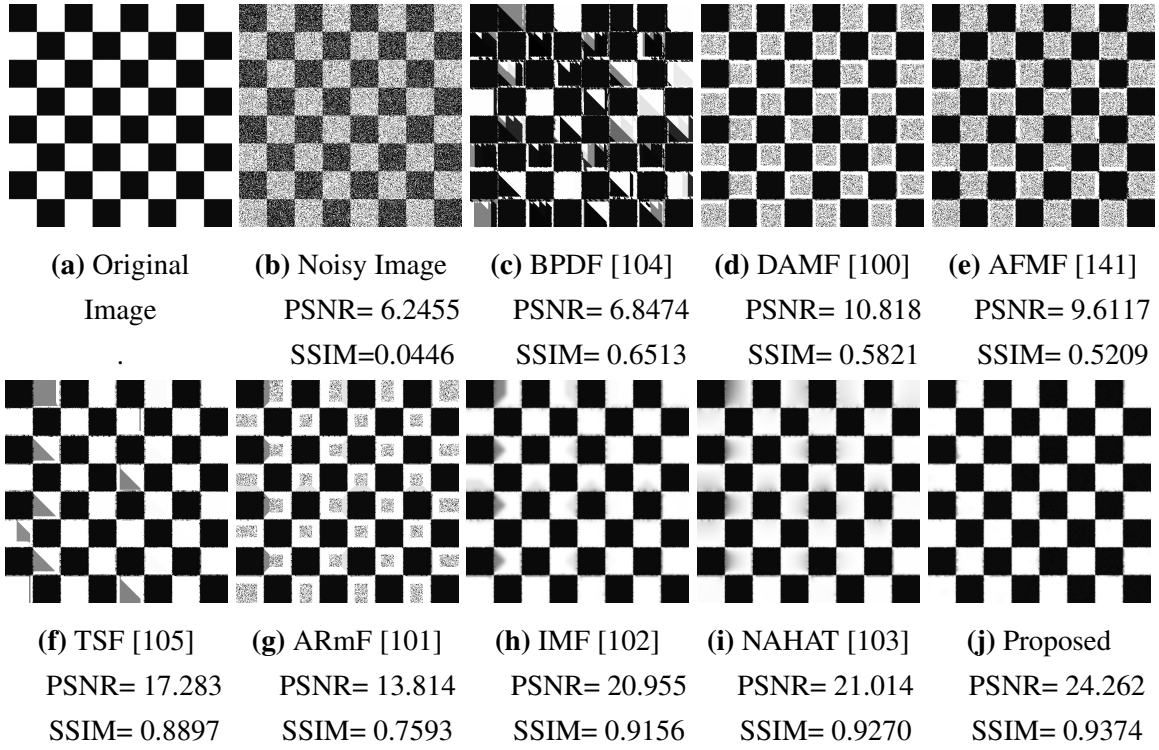


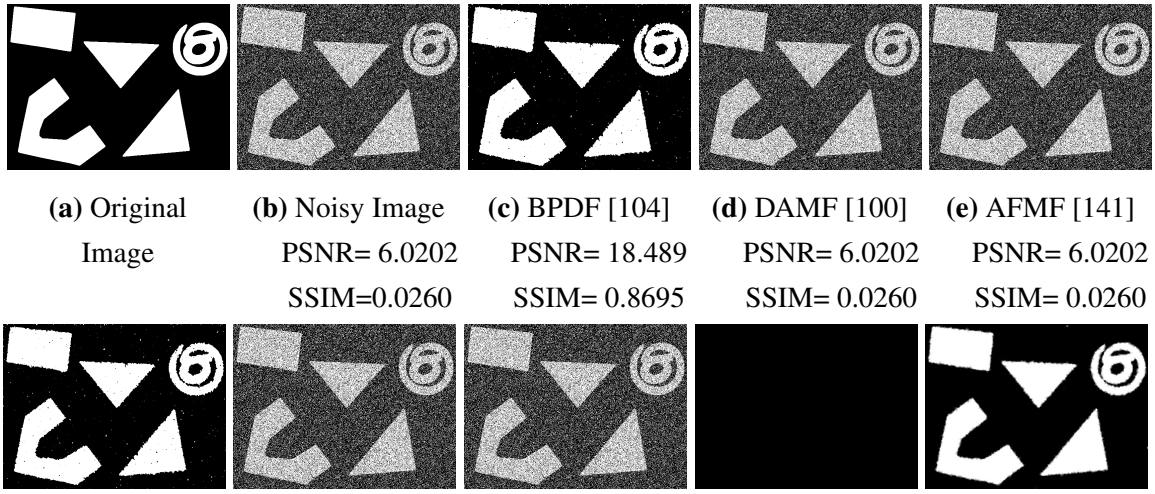
Figure 6.6: Denoising performance of various methods on chessboard image corrupted by SP noise at 50% noise density.

value 255. All these white pixels are thus, considered as noisy pixels by these methods. This causes the spreading of black pixels present in the neighborhood over such white regions. Hence, these methods prove to be ineffective in such cases.

However, the *proposed* method overcomes all these disadvantages and provides an efficient image denoising. This is because of the robust NDM method *proposed* in this chapter. The *proposed* method avoids the misclassification of original pixels as noisy pixels in such cases and hence, outperforms all the conventional methods. This is also evident from the highest values of PSNR and SSIM obtained by the *proposed* method in comparison to the conventional methods.

6.4.2.3 On Synthetic Images

To further validate the advantages of the *proposed* method, a synthetic image containing *all* the pixels with intensity values 0 and 255 is taken into account. The synthetic image is then corrupted with SP noise of density 50%. Figure 6.7 illustrates the denoising performance of various methods on the synthetic image.



(a) Original Image | **(b)** Noisy Image PSNR= 6.0202 SSIM=0.0260 | **(c)** BPDF [104] PSNR= 18.489 SSIM= 0.8695 | **(d)** DAMF [100] PSNR= 6.0202 SSIM= 0.0260 | **(e)** AFMF [141] PSNR= 6.0202 SSIM= 0.0260

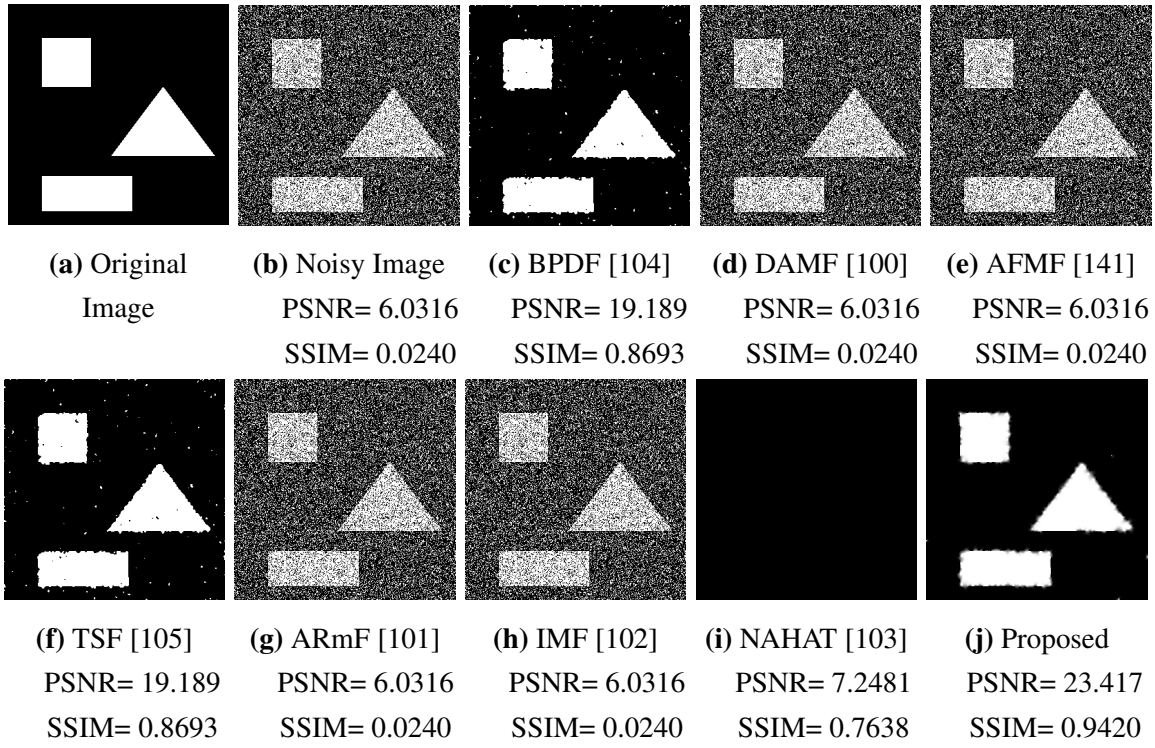
(f) TSF [105] PSNR= 18.489 SSIM= 0.8695 | **(g)** ARmF [101] PSNR= 6.0202 SSIM= 0.0260 | **(h)** IMF [102] PSNR= 6.0202 SSIM= 0.0260 | **(i)** NAHAT [103] PSNR= 5.0995 SSIM= 0.6344 | **(j)** Proposed PSNR= 21.778 SSIM= 0.9115

Figure 6.7: Denoising performance of various methods on synthetic image 1 corrupted by SP noise at 50% noise density.

From the results, it can be inferred that most of the methods such as DAMF [100], AFMF [141], ARmF [101], IMF [102] entirely fail in this case and hence, output the noisy image as it is. The methods such as BPDF [104] and TSF [105] are able to remove noise to a great extent. However, these methods are unable to remove the noise completely. The noisy pixels can be seen in the results presented in Figure 6.7c and 6.7f. Furthermore, NAHAT [103] produces a result consisting of all zeros and hence, proves to be ineffective for synthetic images.

On the other hand, the *proposed* method outperforms the conventional methods in detecting noisy pixels as well as in removing them. This is due to the robust noise detection method that has been utilized in the *proposed* GFIFD-AA. The performance of the *proposed* method can also be validated from the highest values of PSNR and SSIM presented in Figure 6.7.

To further illustrate the performance of the *proposed* method, a wide variety of black and white images affected by SP noise are taken into account and the performance is compared against well-established state-of-the-art methods. Figure 6.8 presents the denoising performance of various methods on an image affected by SP of density 50%. From the results, it can be seen that most of the methods such as DAMF [100], AFMF [141], ARmF [101], IMF [102] entirely fail in this case as well and provides the noisy image itself as an output. Alternatively, methods such as BPDF [104] and TSF [105] are able to remove noise to a great extent, but not significantly. Furthermore, NAHAT [103] provides a black image as an



(a) Original Image | **(b)** Noisy Image PSNR= 6.0316 SSIM= 0.0240 | **(c)** BPDF [104] PSNR= 19.189 SSIM= 0.8693 | **(d)** DAMF [100] PSNR= 6.0316 SSIM= 0.0240 | **(e)** AFMF [141] PSNR= 6.0316 SSIM= 0.0240

(f) TSF [105] PSNR= 19.189 SSIM= 0.8693 | **(g)** ARmF [101] PSNR= 6.0316 SSIM= 0.0240 | **(h)** IMF [102] PSNR= 6.0316 SSIM= 0.0240 | **(i)** NAHAT [103] PSNR= 7.2481 SSIM= 0.7638 | **(j)** Proposed PSNR= 23.417 SSIM= 0.9420

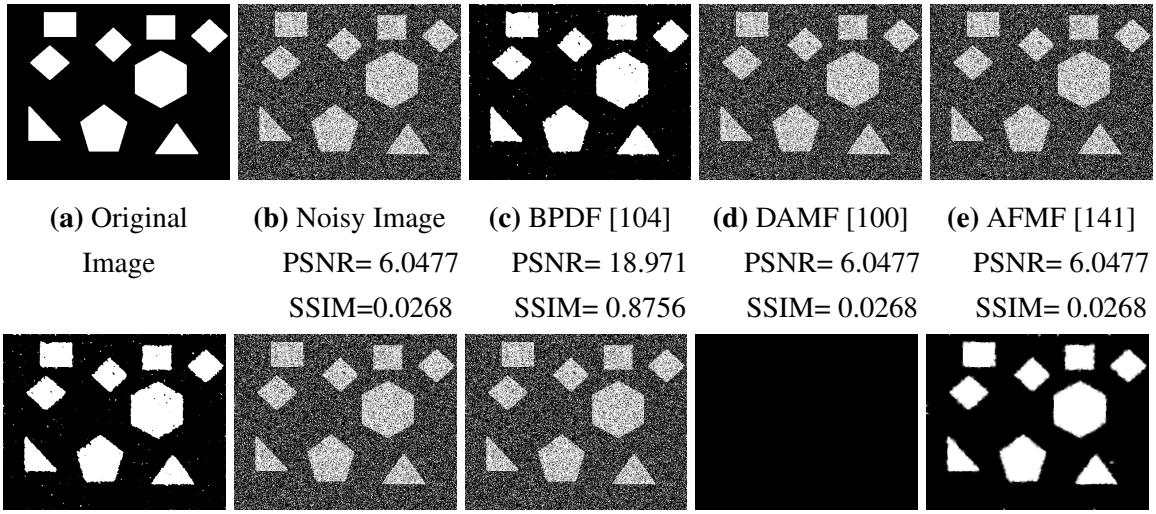
Figure 6.8: Denoising performance of various methods on synthetic image 2 corrupted by SP noise at 50% noise density.

output and hence, proves to be inefficient for synthetic images. The performance of these methods can also be observed by the PSNR and SSIM values attained by them.

On the other hand, it can be seen from Figure 6.8j that the *proposed* method is able to detect and remove the noise significantly and hence, proves to be effective for synthetic images affected by SP noise. The performance of the *proposed* method can also be validated from the highest values of PSNR and SSIM presented in Figure 6.8.

Similar results have been obtained for another image presented in Figure 6.9, where DAMF [100], AFMF [141], ARmF [101], IMF [102], NAHAT[103] fail entirely, BPDF [104] and TSF [105] are able to remove noise to a great extent and the *proposed* method outperforms all these methods in terms of visualization as well as highest values of PSNR and SSIM.

On the other hand, a black and white image with white background has been taken into account and is further affected by SP noise of density 50%. Then, the performance of all these methods is tested and compared against the *proposed* method, as shown in Figure 6.10. From the results presented in Figure 6.10, it is observed that the methods such as DAMF [100], AFMF [141] and ARmF [101] are not able to remove noise. The methods such as BPDF [104] and IMF [102] provide an entire white image as an output. However, the methods such



(a) Original Image | **(b)** Noisy Image PSNR= 6.0477 SSIM=0.0268 | **(c)** BPDF [104] PSNR= 18.971 SSIM= 0.8756 | **(d)** DAMF [100] PSNR= 6.0477 SSIM= 0.0268 | **(e)** AFMF [141] PSNR= 6.0477 SSIM= 0.0268

(f) TSF [105] PSNR= 18.971 SSIM= 0.8756 | **(g)** ARmF [101] PSNR= 6.0477 SSIM= 0.0268 | **(h)** IMF [102] PSNR= 6.0477 SSIM= 0.0268 | **(i)** NAHAT [103] PSNR= 6.5674 SSIM= 0.7209 | **(j)** Proposed PSNR= 22.058 SSIM= 0.9182

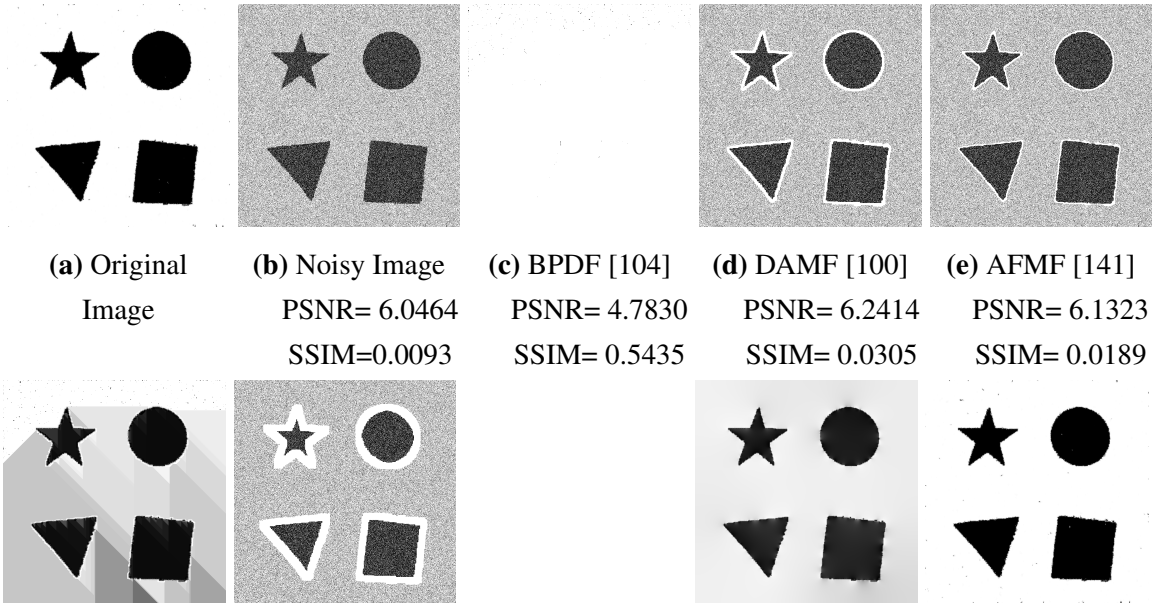
Figure 6.9: Denoising performance of various methods on synthetic image 3 corrupted by SP noise at 50% noise density.

as TSF [105] and NAHAT [103] provide blur outputs with gray background. Also, it can be seen that the methods such as BPDF [104] and TSF [105], which removed noise to a great extent in synthetic images with black background, as presented in Figure 6.7, Figure 6.8, Figure 6.9, are not able to perform for synthetic images with white background.

On the other hand, the *proposed* method outperforms all these methods for this image as well, as can be seen from the result presented in Figure 6.10j. The efficiency of *proposed* method can also be seen from the highest values of PSNR and SSIM attained by the *proposed* method.

To further validate the denoising performance of all methods, an another image with different characteristics affected with SP noise of density 50% is taken into account, as presented in Figure 6.11. From the results presented in Figure 6.11, it can be seen that the methods such as DAMF [100], AFMF [141], and ARmf [101] outputs the noisy image with enhanced edges and hence, prove to be inefficient. On the other hand, the methods such as BPDF [104], TSF [105] and IMF [102] provide white image as an output. However, NAHAT [103] outputs a distorted image with blur effects where the black region spreads out in the white region.

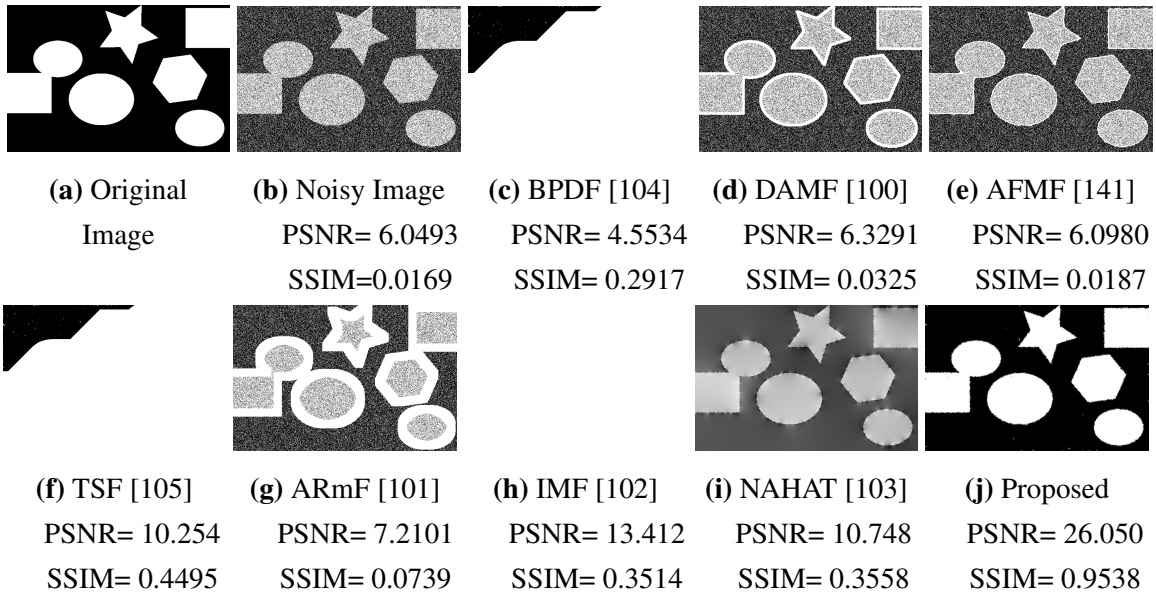
On the other hand, the *proposed* method is effective in detecting as well as removing noise,



(a) Original Image | **(b)** Noisy Image PSNR= 6.0464 SSIM=0.0093 | **(c)** BPDF [104] PSNR= 4.7830 SSIM= 0.5435 | **(d)** DAMF [100] PSNR= 6.2414 SSIM= 0.0305 | **(e)** AFMF [141] PSNR= 6.1323 SSIM= 0.0189

(f) TSF [105] PSNR= 15.523 SSIM= 0.8636 | **(g)** ARmF [101] PSNR= 6.6251 SSIM= 0.0729 | **(h)** IMF [102] PSNR= 26.207 SSIM= 0.8282 | **(i)** NAHAT [103] PSNR= 20.490 SSIM= 0.8129 | **(j)** Proposed PSNR= 31.594 SSIM= 0.9855

Figure 6.10: Denoising performance of various methods on synthetic image 4 corrupted by SP noise at 50% noise density.



(a) Original Image | **(b)** Noisy Image PSNR= 6.0493 SSIM=0.0169 | **(c)** BPDF [104] PSNR= 4.5534 SSIM= 0.2917 | **(d)** DAMF [100] PSNR= 6.3291 SSIM= 0.0325 | **(e)** AFMF [141] PSNR= 6.0980 SSIM= 0.0187

(f) TSF [105] PSNR= 10.254 SSIM= 0.4495 | **(g)** ARmF [101] PSNR= 7.2101 SSIM= 0.0739 | **(h)** IMF [102] PSNR= 13.412 SSIM= 0.3514 | **(i)** NAHAT [103] PSNR= 10.748 SSIM= 0.3558 | **(j)** Proposed PSNR= 26.050 SSIM= 0.9538

Figure 6.11: Denoising performance of various methods on synthetic image 5 corrupted by SP noise at 50% noise density.

as can be seen from the results presented in Figure 6.11j. Therefore, it can be said that the proposed method is not only effective for images containing some pixels with intensity values 0 or 255, but also for synthetic images where all the pixel values are 0 and 255.

Hence, it can be concluded that the performance of the *proposed* method is better than the well-established state-of-the-art methods such as MF [127], GAFIA [93], AFC-SPS [93], EN-AFC [94], BPDF [104], DAMF [100], AFMF [141], TSF [105] and ARmF [101] and is comparable to that of IMF [102] and NAHAT [103] methods for natural images. However, the *proposed* **GFIFD-AA** is able to preserve and enhance the texture and edge details of the original image in comparison to the conventional methods. On the other hand, for the case of images containing some pixels with intensity values 0 and 255, the *proposed* method possesses significant advantages over that of conventional methods. For the case of synthetic images, the *proposed* method outperforms all the conventional methods discussed in this chapter. Therefore, it can be said that the *proposed* method not only detects noise effectively in natural as well as synthetic images, but also proves to be efficient in removing the noise and enhancing the details by utilizing the *proposed* **GFI-AM** and **GFD-AM**, respectively. Therefore, the *proposed* method exhibits a great potential in image denoising applications when images are corrupted with high density SP noise.

6.4.3 Computational Time

Figure 6.12 presents the execution time of the *proposed* method in comparison to that of conventional methods. From the results, it can be seen that NAHAT [103] method is fastest among all. The *proposed* method is faster than AFMF [141] method. The execution time of the *proposed* method is comparable to that of BPDF [104] and IMF [102] methods. The comparatively high processing time of the *proposed* method against that of DAMF [100], ARmF [101], TSF [105], NAHAT [103] methods is due to the iterative nature of the *proposed* method. However, it can be seen that the *proposed* method takes on an average 2 seconds only to process an image of Set12 dataset.

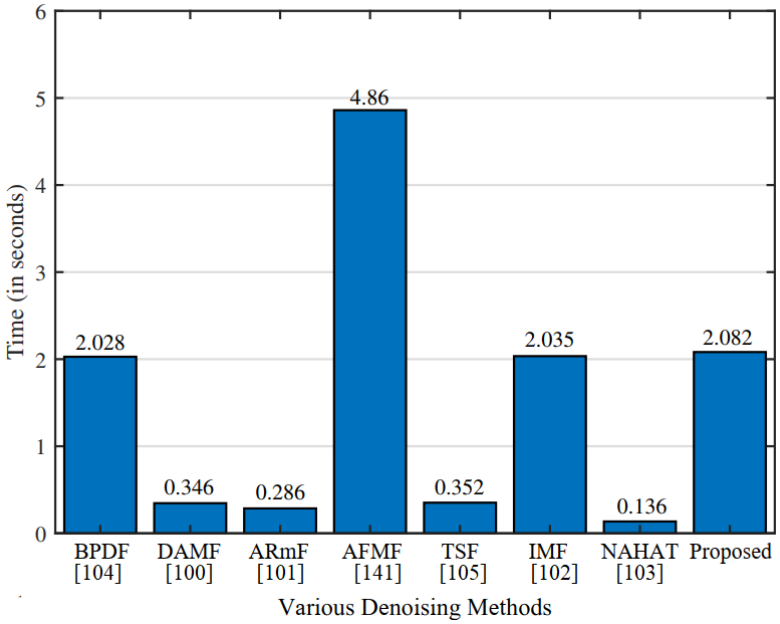


Figure 6.12: Average processing time of various methods for all the images of Set12 dataset with all noise levels from 10% to 90%

6.5 Summary

This chapter puts forward a generalized framework for the design of fractional differential mask which incorporates various definitions of fractional-differential operators, such as, Riemann-Liouville (RL)/Caputo, Caputo-Fabrizio (CF), Atangana-Baleanu (AB), etc. The main advantage of the *proposed* generalized fractional differential mask (GFD) is that by utilizing only one mask, the performance of various fractional-order kernels for image processing applications can be analyzed. This, in turn, avoids the need to construct fractional differential mask for each definition separately.

On the other hand, to obtain better denoising effect, a generalized fractional integral and fractional differential based adaptive algorithm (**GFIFD-AA**) is *proposed*. In the *proposed* **GFIFD-AA**, a novel noise detection method (NDM) is put forward to detect noisy pixels in the image corrupted by salt and pepper noise. The noisy pixels are then processed by utilizing GFI-based adaptive mask (GFI-AM). The noise-free pixels are then updated by utilizing the *proposed* GFD-based adaptive mask (GFD-AM) so as to preserve and enhance the texture and edge details of an image simultaneously.

To show the effectiveness of the *proposed* **GFIFD-AA** method, several standard and medical images are taken into account. The results are validated in terms of quantitative parameters,

which include, PSNR, SSIM, MAE, MSSIM, and entropy. Furthermore, the *proposed* noise detection method detects the noise points in the images affected by SP noise effectively and hence, the *proposed* **GFIFD-AA** outperforms most of the well-established methods discussed in this study.

Furthermore, the *proposed* method exhibits a profound advantage in avoiding misclassification of original pixels with intensity values 0 or 255 as noisy pixels and is effective in removing salt and pepper noise at various densities, i.e., 10%-95%. Also, the *proposed* NDM as well the image denoising method works well for synthetic images affected by SP noise. However, when the noise density is very high (i.e., greater than 80%), it is very hard to preserve the details of the image completely. This is also the limitation of conventional methods proposed in the literature. Therefore, there is still a scope of improvement for preserving the details in images affected by SP noise of density higher than 80%.

Chapter 7

Conclusions and Future Work

Since the existence of fractional-order calculus (FOC), several definitions of fractional-order operators based upon various kernel functions have been proposed in the research community. The comprehensive study conducted elucidated the fact that these definitions of fractional-order operators have found numerous applications in several areas of science and engineering. The thesis investigated the performance of these fractional-order operators for various signal and image processing applications. In particular, the thesis focused on the design of fractional-order digital filters based on various definitions of fractional-order operators. The main contributions of the thesis are summarized in Section 7.1 and a few tracks for future research are delineated in Section 7.2.

7.1 Conclusion

The main contribution of the thesis is the analytical formulation of fractional-order digital filters based on various fractional-order kernels in order to evaluate their performance for various signal and image processing applications. The contribution of the research work reported in this thesis can be concluded as follows:

- Development of power-law kernel based Riemann-Liouville fractional-order digital differentiator (RL-FODD) in contrast to conventional Grünwald-Letnikov based FODD (GL-FODD).

- Development of non-local and non-singular Mittag-Leffler kernel based Atangana-Baleanu-Caputo fractional-order digital filter (ABC-FODF) and its applications in

ECG R-peak detection and image enhancement.

- Design of generalized fractional integral and generalized fractional differential based non-adaptive and adaptive masks for image processing applications.
- Development of novel noise detection method for the detection of salt and pepper noise in images and
- Establishment of novel generalized fractional integral and fractional differential based adaptive algorithm (GFIFD-AA) for the detection and removal of salt and pepper noise in standard as well as synthetic images.

The initial work in this thesis focused on the development of closed-form analytical framework for the design of power-law kernel based RL-FODD and compared its performance against well-established designs of GL-FODD, based upon their ability to yield ideal frequency response of FODD. Since GL-based methods have been extensively investigated by the research community for the design of FODD, this work closed the gap in the field of FODD where existed an improvement window for designing FIR filters based upon RL fractional differential operator. From the analytical and simulation studies conducted, it is concluded that the *proposed* method not only can efficiently approximate the ideal frequency response of FODD but can also provide much better results than the conventional well-established designs of GL-FODD in estimating fractional-order derivatives of the signals.

Furthermore, considering Mittag-Leffler kernel based fractional-order operator, a detailed study has been presented in the literature, where it has been pointed out that the transfer function associated to AB fractional differential operator is yet to be explored and analyzed in-depth. Therefore, further research work carried out in the thesis extended the domain of recently introduced fractional-order operator by establishing the design of ABC-based fractional-order digital filter for signal and image processing applications. Furthermore, in order to discuss the applicability of the *proposed* method, the preliminary applications of the *proposed* ABC-FODF in ECG R-peak detection and digital image sharpening are considered. From the simulation studies conducted, it is proved that the *proposed* method has the ability to accurately delineate R-peaks in ECG signals and also, possesses significant advantages over well-established state-of-the-art methods from the literature. Furthermore, the *proposed*

image sharpening method not only enhanced the quality of digital images significantly, but also provided the flexibility in enhancement of images in contrast to conventional integer order counterparts. Also, the comparison among GL, RL and the *proposed* AB fractional-order masks indicated the superiority of the Mittag-Leffler kernel in contrast to conventional power-law kernel for image enhancement applications.

After establishing closed-form analytical formulations for the design of fractional-order digital filters based on different fractional-order differential operators, further research work focused on the analysis of various fractional-order integral operators for denoising applications. For that, a generalized fractional integral (GFI) mask is *proposed* which incorporated both Riemann-Liouville and Atangana-Baleanu fractional-order integral operators. The *proposed* GFI mask provided a more flexible and appropriate tool for image denoising applications and hence, avoided the construction of different masks for each definition separately. On the other hand, since there is difficulty in the selection of fractional-order manually, the order selection technique for the *proposed* GFI mask is presented. Therefore, a GFI-based adaptive mask (GFI-AM) is put forward to select the fractional-order of the masks based upon local features of an image. Several standard and medical images of different characteristics are examined to evaluate the performance of the *proposed* mask for denoising applications. From the simulation studies conducted, it is seen that the *proposed* AB based fractional-order adaptive mask outperforms the conventional GL and RL based non-adaptive and adaptive masks significantly.

However, in the process of image denoising, some of the features such as texture and edge details of the image get blurred. Therefore, it is important to preserve the texture and edge details while denoising the image simultaneously. Also, a noise of higher density may exist in digital images. Hence, to address these challenges and as a final contribution of this thesis, a novel generalized fractional integral and fractional differential based adaptive algorithm (GFIFD-AA) is *proposed* for images affected by salt and pepper noise.

For that, initially, a generalized framework for the design of fractional differential mask which incorporates various definitions of fractional-differential operators, such as, Riemann-Liouville (RL)/Caputo, Caputo-Fabrizio (CF), Atangana-Baleanu (AB), etc. is *proposed*. The main advantage of the *proposed* generalized fractional differential (GFD) mask is that by utilizing only one mask, the performance of various fractional-order kernels for image

processing applications can be analyzed. This, in turn, also avoided the need to construct fractional differential mask for each definition separately. Similarly, in order to select the fractional order based upon local features of the image, GFD-based adaptive mask (GFD-AM) is presented.

Then, a generalized fractional integral and fractional differential based adaptive algorithm (GFIFD-AA) is put forward. In the *proposed* GFIFD-AA, a novel noise detection method (NDM) based on probability strategy is introduced to detect noisy pixels in the images corrupted by salt and pepper noise. The detected noisy pixels are then processed by utilizing the *proposed* GFI-based adaptive mask (GFI-AM). The noise-free pixels are then updated by utilizing the *proposed* GFD-based adaptive mask (GFD-AM) so as to preserve and enhance the texture and edge details of an image simultaneously. To show the effectiveness of the *proposed* GFIFD-AA method, several standard and medical images are taken into account and the results are validated in terms of several quantitative parameters.

From the simulation studies conducted, it is validated that the *proposed* noise detection method detected the noise points in the images affected by SP noise effectively and hence, the *proposed* GFIFD-AA outperformed most of the well-established methods discussed in this study. Furthermore, the *proposed* method exhibited a profound advantage in avoiding misclassification of original pixels with intensity values 0 or 255 as noisy pixels and is effective in removing salt and pepper noise at various densities, i.e., 10%-95%. Also, the *proposed* NDM as well as the image denoising method worked well for synthetic images affected by SP noise in contrast to conventional methods. Therefore, it can be said that the *proposed* method not only detected the noise effectively in natural as well as synthetic images, but also proved to be efficient in removing the noise and enhancing the details by utilizing the *proposed* GFI-AM and GFD-AM, respectively.

7.2 Future Work

Despite of comprehensive research in the area of FOC over the last few decades, several challenges still remain open. A few tracks for future research out of the present work are outlined below:

- The preliminary results obtained by the *proposed* ABC-FODF for the detection of R-

peaks in ECG signals and digital image sharpening are encouraging, and hence, it can be said that the *proposed* method could be beneficial for various signal filtering, image enhancement, and biomedical signal/image processing applications. Therefore, the potential applications of ABC-based FODF for various signal and image processing applications and its detailed comparative analysis with the conventional methods remain open for further future investigation.

- The performance of the *proposed* ABC-FODF method in real-time applications depends on the choice of fractional order α, which have been selected based upon the experimental studies conducted. Therefore, future work can be extended based upon the design and implementation of an adaptive ABC-fractional differential algorithm that can adjust fractional-order according to the dynamic gradient features of real-time ECG signals and images.

- Furthermore, the *proposed* denoising algorithm can be extended to denoise other noises in images such as random-value impulse noise. In addition, the application of *proposed* fractional-order adaptive masks combined with deep learning techniques for object detection in biomedical images can be an interesting problem to explore.

- However, when the noise density is very high (i.e., greater than 80%), it is very hard to preserve the details of the image completely. This is also the limitation of conventional methods *proposed* in the literature. Therefore, there is still a scope of improvement for preserving the details in images affected by SP noise of density higher than 80%. Hence, as a future scope of this work, this challenge can be taken into account.

BIBLIOGRAPHY

[1] C. A. Monje, Y. Chen, B. M. Vinagre, D. Xue, and V. Feliu Batlle, *Fractional-order Systems and Controls:Fundamentals and Applications*. Springer Science & Business Media, 2010.

[2] H. Sheng, Y. Chen, and T. Qiu, *Fractional Processes and Fractional-order Signal Processing:Techniques and Applications*. Springer Science & Business Media, 2011.

[3] R. Magin, M. D. Ortigueira, I. Podlubny, and J. Trujillo, "On the fractional signals and systems," *Signal Processing*, vol. 91, no. 3, pp. 350–371, 2011.

[4] C. Nayak, S. K. Saha, R. Kar, and D. Mandal, "An efficient and robust digital fractional order differentiator based ECG pre-processor design for QRS detection," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 13, no. 4, pp. 682–696, 2019.

[5] D. Baleanu, J. A. T. Machado, and A. C. Luo, *Fractional Dynamics and Control*. Springer Science & Business Media, 2011.

[6] B. Ross, "A brief history and exposition of the fundamental theory of fractional calculus," *Fractional Calculus and Its Applications*, pp. 1–36, 1975.

[7] M. D. Ortigueira, *Fractional Calculus for Scientists and Engineers*. Springer Science & Business Media, 2011.

[8] K. Oldham and J. Spanier, *The Fractional Calculus Theory and Applications of Differentiation and Integration to Arbitrary Order*. New York : Academic, 1974.

[9] J. T. Machado, V. Kiryakova, and F. Mainardi, "Recent history of fractional calculus," *Communications in Nonlinear Science and Numerical Simulation*, vol. 16, no. 3, pp. 1140–1153, 2011.

[10] H. Sun, Y. Zhang, D. Baleanu, W. Chen, and Y. Chen, "A new collection of real world applications of fractional calculus in science and engineering," *Communications in Nonlinear Science and Numerical Simulation*, vol. 64, pp. 213–231, 2018.

[11] S. Das, *Kindergarten of Fractional Calculus*. Cambridge Scholars Publishing, 2020.

[12] B. T. Krishna, "Studies on fractional order differentiators and integrators: A survey," *Signal Processing*, vol. 91, no. 3, pp. 386–426, 2011.

[13] Y. Pu, J. Zhou, and X. Yuan, "Fractional differential mask: A fractional differential-based approach for multiscale texture enhancement," *IEEE Transactions on Image Processing*, vol. 19, no. 2, pp. 491–511, 2010.

[14] C. M. Rader and B. Gold, "Digital filter design techniques in the frequency domain," *Proceedings of the IEEE*, vol. 55, no. 2, pp. 149–171, 1967.

[15] J. G. Proakis, *Digital Signal Processing: Principles Algorithms and Applications*. Pearson Education India, 2001.

[16] G. Jajoo, Y. Kumar, A. Kumar, and S. K. Yadav, "Blind signal modulation recognition through density spread of constellation signature," *Wireless Personal Communications*, vol. 114, no. 4, pp. 3137–3156, 2020.

[17] G. Jajoo, Y. Kumar, and S. K. Yadav, "Blind signal PSK/QAM recognition using clustering analysis of constellation signature in flat fading channel," *IEEE Communications Letters*, vol. 23, no. 10, pp. 1853–1856, 2019.

[18] A. Bhardwaj and S. Agnihotri, "Energy- and spectral-efficiency trade-off for D2D-multicasts in underlay cellular networks," *IEEE Wireless Communications Letters*, vol. 7, no. 4, pp. 546–549, 2018.

[19] Y. F. Pu, P. Siarry, A. Chatterjee, Z. N. Wang, Z. Yi, Y. G. Liu, J. L. Zhou, and Y. Wang, "A fractional-order variational framework for retinex: Fractional-order partial differential equation-based formulation for multi-scale nonlocal contrast enhancement with

texture preserving," *IEEE Transactions on Image Processing*, vol. 27, no. 3, pp. 1214–1229, 2017.

[20] Y. Ferdi, "Some applications of fractional order calculus to design digital filters for biomedical signal processing," *Journal of Mechanics in Medicine and Biology*, vol. 12, no. 2, p. 1240008, 2012.

[21] B. Mathieu, P. Melchior, A. Oustaloup, and C. Ceyral, "Fractional differentiation for edge detection," *Signal Processing*, vol. 83, no. 11, pp. 2421–2432, 2003.

[22] M. Sharma, B. S. Rajpurohit, S. Agnihotri, and A. K. Rathore, "Development of fractional order modeling of voltage source converters," *IEEE Access*, vol. 8, pp. 131 750–131 759, 2020.

[23] H. Sheng, H. Sun, Y. Chen, and T. Qiu, "A variable-order fractional operator based synthesis method for multifractional Gaussian noise," in *Proceedings of IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications*, 2010, pp. 474–479.

[24] P. C. Shrivastava, P. Kumar, M. Tiwari, and A. Dhawan, "Efficient architecture for the realization of 2-D adaptive FIR filter using distributed arithmetic," *Circuits, Systems, and Signal Processing*, vol. 40, no. 3, pp. 1458–1478, 2021.

[25] P. Kumar, P. C. Shrivastava, M. Tiwari, and G. R. Mishra, "High-throughput, area-efficient architecture of 2-D block FIR filter using distributed arithmetic algorithm," *Circuits, Systems, and Signal Processing*, vol. 38, no. 3, pp. 1099–1113, 2019.

[26] R. R. Nigmatullin and S. Osokin, "Signal processing and recognition of true kinetic equations containing non-integer derivatives from raw dielectric data," *Signal Processing*, vol. 83, no. 11, pp. 2433–2453, 2003.

[27] A. Goutas, Y. Ferdi, J. P. Herbeuval, M. Boudraa, and B. Boucheham, "Digital fractional order differentiation-based algorithm for P and T-waves detection and delineation," *ITBM-RBM*, vol. 26, no. 2, pp. 127–132, 2005.

[28] M. D. Ortigueira and J. T. Machado, "Fractional calculus applications in signals and systems," *Signal Processing*, vol. 86, no. 10, pp. 2503–2504, 2006.

[29] D. Zhao and M. Luo, "Representations of acting processes and memory effects: General fractional derivative and its application to theory of heat conduction with finite wave speeds," *Applied Mathematics and Computation*, vol. 346, pp. 531–544, 2019.

[30] M. Caputo and M. Fabrizio, "A new definition of fractional derivative without singular kernel," *Progress in Fractional Differentiation and Applications*, vol. 1, no. 2, pp. 1–13, 2015.

[31] A. Atangana and D. Baleanu, "New fractional derivatives with nonlocal and non-singular kernel: Theory and application to heat transfer model," *Thermal Science*, vol. 20, no. 2, pp. 763–769, 2016.

[32] D. Baleanu and A. Fernandez, "On fractional operators and their classifications," *Mathematics*, vol. 7, no. 9, p. 830, 2019.

[33] E. C. De Oliveira and J. A. Tenreiro Machado, "A review of definitions for fractional derivatives and integral," *Mathematical Problems in Engineering*, vol. 2014, 2014.

[34] B. S. T. Alkahtani, "Chua's circuit model with Atangana-Baleanu derivative with fractional order," *Chaos, Solitons & Fractals*, vol. 89, pp. 547–551, 2016.

[35] T. Abdeljawad and D. Baleanu, "Integration by parts and its applications of a new nonlocal fractional derivative with Mittag-Leffler nonsingular kernel," *Journal of Nonlinear Sciences and Applications*, vol. 10, pp. 1098–1107, 2017.

[36] S. Yadav, R. K. Pandey, and A. K. Shukla, "Numerical approximations of Atangana-Baleanu-Caputo derivative and its application," *Chaos, Solitons & Fractals*, vol. 118, pp. 58–64, 2019.

[37] J. Solís-Pérez, J. F. Gómez-Aguilar, R. F. Escobar-Jiménez, and J. Reyes-Reyes, "Blood vessel detection based on fractional Hessian matrix with non-singular Mittag-Leffler Gaussian kernel," *Biomedical Signal Processing and Control*, vol. 54, p. 101584, 2019.

[38] H. J. Haubold, A. M. Mathai, and R. K. Saxena, "Mittag-Leffler functions and their applications," *Journal of Applied Mathematics*, vol. 2011, 2011.

[39] R. Gorenflo, A. A. Kilbas, F. Mainardi, and S. V. Rogosin, *Mittag-Leffler Functions, Related Topics and Applications*. Springer, 2020.

[40] D. Baleanu and A. Fernandez, "On some new properties of fractional derivatives with Mittag-Leffler kernel," *Communications in Nonlinear Science and Numerical Simulation*, vol. 59, pp. 444–462, 2018.

[41] A. Giusti, "A comment on some new definitions of fractional derivative," *Nonlinear Dynamics*, vol. 93, no. 3, pp. 1757–1763, 2018.

[42] M. D. Ortigueira, V. Martynyuk, M. Fedula, and J. T. Machado, "The failure of certain fractional calculus operators in two physical models," *Fractional Calculus and Applied Analysis*, vol. 22, no. 2, pp. 255–270, 2019.

[43] A. Atangana and J. Gómez-Aguilar, "Decolonisation of fractional calculus rules: Breaking commutativity and associativity to capture more natural phenomena," *The European Physical Journal Plus*, vol. 133, no. 4, pp. 1–22, 2018.

[44] T. I. Laakso, V. Valimaki, M. Karjalainen, and U. K. Laine, "Splitting the unit delay [FIR/all pass filters design]," *IEEE Signal Processing Magazine*, vol. 13, no. 1, pp. 30–60, 1996.

[45] S. Haykin, *Neural Networks and Learning Machines, 3/E*. Pearson Education India, 2010.

[46] J. Vesma and T. Saramaki, "Interpolation filters with arbitrary frequency response for all-digital receivers," in *IEEE International Symposium on Circuits and Systems*, 1996, pp. 568–571.

[47] M. Olsson, H. Johansson, and P. Lowenborg, "Delay estimation using adjustable fractional delay all-pass filters," in *Proceedings of the 7th Nordic Signal Processing Symposium*, 2006, pp. 346–349.

[48] P. Murphy, A. Krukowski, and A. Tarczynski, "An efficient fractional sample delayer for digital beam steering," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, 1997, pp. 2245–2248.

[49] V. Välimäki, H.-M. Lehtonen, and T. I. Laakso, "Musical signal analysis using fractional delay inverse comb filters," in *Proceedings of the International Conference on Digital Audio Effects*, 2007, pp. 261–268.

[50] T. Bensouici and A. Charef, "Fractional Euler analog-to-digital transform," *AEU-International Journal of Electronics and Communications*, vol. 69, no. 4, pp. 730–735, 2015.

[51] O. P. Goswami, T. K. Rawat, and D. K. Upadhyay, "Fractional interpolation and multirate technique based design of optimum IIR integrators and differentiators," *International Journal of Electronics*, vol. 108, no. 9, pp. 1482–1496, 2021.

[52] T. Bensouici, A. Charef, and I. Assadi, "A new approach for the design of fractional delay by an FIR filter," *ISA Transactions*, vol. 82, pp. 73–78, 2018.

[53] C. C. Tseng, "Improved design of digital fractional-order differentiators using fractional sample delay," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 53, no. 1, pp. 193–203, 2006.

[54] C. C. Tseng and S. Lee, "Design of fractional order digital differentiator using radial basis function," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 7, pp. 1708–1718, 2010.

[55] S. D. Roy and B. Kumar, "6 digital differentiators," *Handbook of Statistics*, vol. 10, pp. 159–205, 1993.

[56] C. C. Tseng, "Design of fractional order digital FIR differentiators," *IEEE Signal Processing Letters*, vol. 8, no. 3, pp. 77–79, 2001.

[57] P. Sabherwal, M. Agrawal, and L. Singh, "Automatic detection of the R-peaks in single-lead ECG signal," *Circuits, Systems, and Signal Processing*, vol. 36, no. 11, pp. 4637–4652, 2017.

[58] A. Kaur, A. Agarwal, R. Agarwal, and S. Kumar, "A novel approach to ECG R-peak detection," *Arabian Journal of Science and Engineering*, vol. 44, no. 8, pp. 6679–6691, 2019.

[59] M. Benmalek and A. Charef, "Digital fractional order operators for R-wave detection in electrocardiogram signal," *IET Signal Processing*, vol. 3, no. 5, pp. 381–391, 2009.

[60] A. Kaur, S. Kumar, A. Agarwal, and R. Agarwal, "An efficient R-peak detection using Riesz fractional-order digital differentiator," *Circuits, Systems, and Signal Processing*, vol. 39, no. 4, pp. 1965–1987, 2020.

[61] M. Deriche and A. H. Tewfik, "Signal modeling with filtered discrete fractional noise processes," *IEEE Transactions on Signal Processing*, vol. 41, no. 9, pp. 2839–2849, 1993.

[62] D. Chen, Y. Chen, and D. Xue, "Digital fractional order Savitzky-Golay differentiator," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 58, no. 11, pp. 758–762, 2011.

[63] R. C. Gonzalez, R. E. Woods, and S. L. Eddins, *Digital Image Processing using MATLAB*. Pearson Education India, 2004.

[64] Q. Yang, D. Chen, T. Zhao, and Y. Chen, "Fractional calculus in image processing: A review," *Fractional Calculus and Applied Analysis*, vol. 19, no. 5, pp. 1222–1249, 2016.

[65] A. Oustaloup, F. Levron, B. Mathieu, and F. M. Nanot, "Frequency-band complex noninteger differentiator: Characterization and synthesis," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 47, no. 1, pp. 25–39, 2000.

[66] C. C. Tseng, S. C. Pei, and S. C. Hsia, "Computation of fractional derivatives using Fourier transform and digital FIR differentiator," *Signal Processing*, vol. 80, no. 1, pp. 151–159, 2000.

[67] G. Carlson and C. Halijak, "Approximation of fractional capacitors (1/s)ˆ(1/n) by a regular Newton process," *IEEE Transactions on Circuit Theory*, vol. 11, no. 2, pp. 210–213, 1964.

[68] S. Roy, "On the realization of a constant-argument immittance or fractional operator," *IEEE Transactions on Circuit Theory*, vol. 14, no. 3, pp. 264–274, 1967.

[69] A. Charef, H. Sun, Y. Tsao, and B. Onaral, "Fractal system as represented by singularity function," *IEEE Transactions on Automatic Control*, vol. 37, no. 9, pp. 1465–1470, 1992.

[70] K. Matsuda and H. Fujii, "H∞ optimized wave-absorbing control-Analytical and experimental results," *Journal of Guidance, Control, and Dynamics*, vol. 16, no. 6, pp. 1146–1153, 1993.

[71] M. D. Ortigueira, "A coherent approach to non-integer order derivatives," *Signal Processing*, vol. 86, no. 10, pp. 2505–2515, 2006.

[72] B. M. Vinagre, Y. Q. Chen, and I. Petráš, "Two direct Tustin discretization methods for fractional-order differentiator/integrator," *Journal of the Franklin Institute*, vol. 340, no. 5, pp. 349–362, 2003.

[73] A. Tepljakov, E. Petlenkov, and J. Belikov, "Application of Newton's method to analog and digital realization of fractional-order controllers," *International Journal of Microelectronics and Computer Science*, vol. 3, no. 2, pp. 45–52, 2012.

[74] J. T. Machado, "Analysis and design of fractional-order digital control systems," *Systems Analysis Modelling Simulation*, vol. 27, pp. 107–122, 1997.

[75] R. S. Barbosa, J. T. Machado, and M. F. Silva, "Time domain design of fractional differintegrators using least-squares," *Signal Processing*, vol. 86, no. 10, pp. 2567–2581, 2006.

[76] C. C. Tseng and S. L. Lee, "Design of fractional order differentiator using discrete Fourier transform interpolation," in *10th International Symposium on Communications and Information Technologies*, 2010, pp. 306–311.

[77] ——, "Design of fractional order differentiator using discrete Hartley transform," in *18th European Signal Processing Conference*, 2010, pp. 50–54.

[78] ——, "Closed-form design of fractional order differentiator using discrete cosine transform," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2013, pp. 2609–2612.

[79] ——, "Design of digital fractional order differentiator using discrete sine transform," in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, 2013, pp. 1–9.

[80] ——, "Designs of matrix fractional order differentiators," *Signal Processing*, vol. 111, pp. 73–88, 2015.

[81] A. Garg and O. P. Sahu, "Enhancement of speech signal using diminished empirical mean curve decomposition-based adaptive Wiener filtering," *Pattern Analysis and Applications*, vol. 23, no. 1, pp. 179–198, 2020.

[82] M. F. Tolba, L. A. Said, A. H. Madian, and A. G. Radwan, "FPGA implementation of the fractional order integrator/differentiator: Two approaches and applications," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 4, pp. 1484–1495, 2018.

[83] R. Garrappa, E. Kaslik, and M. Popolizio, "Evaluation of fractional integrals and derivatives of elementary functions: Overview and tutorial," *Mathematics*, vol. 7, no. 5, p. 407, 2019.

[84] S. Samadi, M. O. Ahmad, and M. Swamy, "Exact fractional-order differentiators for polynomial signals," *IEEE Signal Processing Letters*, vol. 11, no. 6, pp. 529–532, 2004.

[85] D. Y. Liu, O. Gibaru, W. Perruquetti, and T. M. Laleg-Kirati, "Fractional order differentiation by integration and error analysis in noisy environment," *IEEE Transactions on Automatic Control*, vol. 60, no. 11, pp. 2945–2960, 2015.

[86] P. Amoako-Yirenkyi, J. K. Appati, and I. K. Dontwi, "A new construction of a fractional derivative mask for image edge analysis based on Riemann-Liouville fractional derivative," *Advances in Difference Equations*, vol. 2016, no. 1, pp. 1–23, 2016.

[87] A. Nandal, H. Gamboa-Rosales, A. Dhaka, J. M. Celaya-Padilla, J. I. Galvan-Tejada, C. E. Galvan-Tejada, F. J. Martinez-Ruiz, and C. Guzman-Valdivia, "Image edge detection using fractional calculus with feature and contrast enhancement," *Circuits, Systems, and Signal Processing*, vol. 37, no. 9, pp. 3946–3972, 2018.

[88] J. M. Cruz-Duarte, J. Rosales-Garcia, C. R. Correa-Cely, A. Garcia-Perez, and J. G. Avina-Cervantes, "A closed form expression for the Gaussian-based Caputo-Fabrizio fractional derivative for signal processing applications," *Communications in Nonlinear Science and Numerical Simulation*, vol. 61, pp. 138–148, 2018.

[89] G. Zhang, J. Liu, J. Wang, Z. Tang, Y. Xie, J. He, T. Ma, and J. P. Niyoyita, "Fogdbed: Fractional-order Gaussian derivatives-based edge-relevant structure detection using Caputo-Fabrizio definition," *Digital Signal Processing*, vol. 98, p. 102639, 2020.

[90] J. Lavín-Delgado, J. Solís-Pérez, J. F. Gómez-Aguilar, and R. F. Escobar-Jiménez, "A new fractional-order mask for image edge detection based on Caputo–Fabrizio fractional-order derivative without singular kernel," *Circuits, Systems, and Signal Processing*, vol. 39, pp. 1419–1448, 2020.

[91] B. Ghanbari and A. Atangana, "A new application of fractional Atangana-Baleanu derivatives: Designing ABC-fractional masks in image processing," *Physica A: Statistical Mechanics and its Applications*, vol. 542, p. 123516, 2020.

[92] A. Atangana, "Blind in a commutative world: Simple illustrations with functions and chaotic attractors," *Chaos, Solitons & Fractals*, vol. 114, pp. 347–363, 2018.

[93] B. Li and W. Xie, "Image denoising and enhancement based on adaptive fractional calculus of small probability strategy," *Neurocomputing*, vol. 175, pp. 704–714, 2016.

[94] J. Yu, L. Tan, S. Zhou, L. Wang, and M. A. Siddique, "Image denoising algorithm based on entropy and adaptive fractional order calculus operator," *IEEE Access*, vol. 5, pp. 12 275–12 285, 2017.

[95] A. K. Shukla, R. K. Pandey, and P. K. Reddy, "Generalized fractional derivative based adaptive algorithm for image denoising," *Multimedia Tools and Applications*, vol. 79, pp. 14 201–14 224, 2020.

[96] X. Lin, Y. Wang, G. Wu, and J. Hao, "Image denoising of adaptive fractional operator based on Atangana-Baleanu derivatives," *Journal of Mathematics*, vol. 2021, p. 5581944, 2021.

[97] A. K. Shukla, R. K. Pandey, and S. Yadav, "Adaptive fractional masks and super resolution based approach for image enhancement," *Multimedia Tools and Applications*, vol. 80, no. 20, pp. 30 213–30 236, 2021.

[98] B. Li and W. Xie, "Adaptive fractional differential approach and its application to medical image enhancement," *Computers & Electrical Engineering*, vol. 45, pp. 324–335, 2015.

[99] F. Hu, S. Si, H. San Wong, B. Fu, M. Si, and H. Luo, "An adaptive approach for texture enhancement based on a fractional differential operator with non-integer step and order," *Neurocomputing*, vol. 158, pp. 295–306, 2015.

[100] U. Erkan, L. Gökrem, and S. Enginoğlu, "Different applied median filter in salt and pepper noise," *Computers & Electrical Engineering*, vol. 70, pp. 789–798, 2018.

[101] S. Enginoğlu, U. Erkan, and S. Memiş, "Pixel similarity-based adaptive Riesz mean filter for salt-and-pepper noise removal," *Multimedia Tools and Applications*, vol. 78, no. 24, pp. 35 401–35 418, 2019.

[102] D. N. H. Thanh and S. Engínoğlu, "An iterative mean filter for image denoising," *IEEE Access*, vol. 7, pp. 167 847–167 859, 2019.

[103] D. N. H. Thanh, V. S. Prasath, T. K. Phung, and N. Q. Hung, "Impulse denoising based on noise accumulation and harmonic analysis techniques," *Optik*, vol. 241, p. 166163, 2021.

[104] U. Erkan and L. Gökrem, "A new method based on pixel density in salt and pepper noise removal," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 26, no. 1, pp. 162–171, 2018.

[105] D. N. H. Thanh, N. H. Hai, V. S. Prasath, and J. M. R. Tavares, "A two-stage filter for high density salt and pepper denoising," *Multimedia Tools and Applications*, vol. 79, no. 29, pp. 21 013–21 035, 2020.

[106] D. Baleanu, K. Diethelm, E. Scalas, and J. J. Trujillo, *Fractional Calculus: Models and Numerical Methods*. World Scientific, 2012.

[107] S. Kumar, K. Singh, and R. Saxena, "Closed-form analytical expression of fractional order differentiation in fractional Fourier transform domain," *Circuits, Systems, and Signal Processing*, vol. 32, no. 4, pp. 1875–1889, 2013.

[108] E. Sousa, "How to approximate the fractional derivative of order $1< \alpha \leq 2$," *International Journal of Bifurcation and Chaos*, vol. 22, no. 4, p. 1250075, 2012.

[109] R. Garrappa, "Numerical solution of fractional differential equations: A survey and a software tutorial," *Mathematics*, vol. 6, no. 2, p. 16, 2018.

[110] D. W. Brzeziński, "Fractional order derivative and integral computation with a small number of discrete input values using Grünwald–Letnikov formula," *International Journal of Computational Methods*, vol. 17, no. 5, p. 1940006, 2020.

[111] X. L. Li, Y. M. Chen, D. Y. Liu, Y. Q. Wei, and D. Boutat, "Model-free fractional order differentiator based on fractional order Jacobi orthonormal functions," *Digital Signal Processing*, vol. 71, pp. 69–82, 2017.

[112] J. Wang, Y. Ye, X. Pan, X. Gao, and C. Zhuang, "Fractional zero-phase filtering based on the Riemann-Liouville integral," *Signal Processing*, vol. 98, pp. 150–157, 2014.

[113] A. Atangana and I. Koca, "Chaos in a simple nonlinear system with Atangana-Baleanu derivatives with fractional order," *Chaos, Solitons & Fractals*, vol. 89, pp. 447–454, 2016.

[114] G. B. Moody and R. G. Mark, "The impact of the MIT-BIH arrhythmia database," *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 3, pp. 45–50, 2001.

[115] N. Arzeno, Z. Deng, and C. Poon, "Analysis of first-derivative based QRS detection algorithms," *IEEE Transactions on Biomedical Engineering*, vol. 55, no. 2, pp. 478–484, 2008.

[116] Z. Zidelmal, A. Amirou, M. Adnane, and A. Belouchrani, "QRS detection based on wavelet coefficients," *Computer Methods and Programs in Biomedicine*, vol. 107, no. 3, pp. 490–496, 2012.

[117] T. Sharma and K. K. Sharma, "QRS complex detection in ECG signals using locally adaptive weighted total variation denoising," *Computers in Biology and Medicine*, vol. 87, pp. 187–199, 2017.

[118] M. S. Manikandan and K. Soman, "A novel method for detecting R-peaks in electrocardiogram (ECG) signal," *Biomedical Signal Processing and Control*, vol. 7, no. 2, pp. 118–128, 2012.

[119] C. Nayak, S. K. Saha, R. Kar, and D. Mandal, "Automated QRS complex detection using MFO-based DFOD," *IET Signal Processing*, vol. 12, no. 9, pp. 1172–1184, 2018.

[120] ——, "An optimally designed digital differentiator based preprocessor for R-peak detection in electrocardiogram signal," *Biomedical Signal Processing and Control*, vol. 49, pp. 440–464, 2019.

[121] J. Pan and W. J. Tompkins, "A real-time QRS detection algorithm," *IEEE Transactions on Biomedical Engineering*, no. 3, pp. 230–236, 1985.

[122] B. Mathieu, P. Melchior, A. Oustaloup, and C. Ceyral, "Fractional differentiation for edge detection," *Signal Processing*, vol. 83, no. 11, pp. 2421–2432, 2003.

[123] E. A. Silva, K. Panetta, and S. S. Agaian, "Quantifying image similarity using measure of enhancement by entropy," in *Mobile Multimedia/Image Processing for Military and Security Applications*, vol. 6579, 2007, pp. 219–230.

[124] S. Hemalatha and S. Margret Anouncia, "G-L fractional differential operator modified using auto-correlation function:Texture enhancement in images," *Ain Shams Engineering Journal*, vol. 9, no. 4, pp. 1689–1704, 2018.

[125] M. Goldbaum, "STARE (STructured Analysis of the Retina) Project," 2004, http://cecas.clemson.edu/~ahoover/stare/,(2004).

[126] https://github.com/cszn/DnCNN/tree/master/testsets, Accessed: 2021-03-20.

[127] B. I. Justusson, *Median Filtering: Statistical Properties*, 1981, pp. 161–196.

[128] X. Wang, S. Shen, G. Shi, Y. Xu, and P. Zhang, "Iterative non-local means filter for salt and pepper noise removal," *Journal of Visual Communication and Image Represention*, vol. 38, pp. 440–450, 2016.

[129] P. Zhang and F. Li, "A new adaptive weighted mean filter for removing salt-and-pepper noise," *IEEE Signal Processing Letters*, vol. 21, no. 10, pp. 1280–1283, 2014.

[130] Y. Wang, J. Wang, X. Song, and L. Han, "An efficient adaptive fuzzy switching weighted mean filter for salt-and-pepper noise removal," *IEEE Signal Processing Letters*, vol. 23, no. 11, pp. 1582–1586, 2016.

[131] K. K. V. Toh and N. A. M. Isa, "Noise adaptive fuzzy switching median filter for salt-and-pepper noise reduction," *IEEE Signal Processing Letters*, vol. 17, no. 3, pp. 281–284, 2010.

[132] N. He, J. B. Wang, L. L. Zhang, and K. Lu, "An improved fractional-order differentiation model for image denoising," *Signal Processing*, vol. 112, pp. 180–188, 2015.

[133] H. A. Jalab and R. W. Ibrahim, "Fractional Alexander polynomials for image denoising," *Signal Processing*, vol. 107, pp. 340–354, 2015.

[134] H. Guo, X. Li, C. Qing-li, and W. Ming-rong, "Image denoising using fractional integral," in *IEEE International Conference on Computer Science and Automation Engineering (CSAE)*, vol. 2, 2012, pp. 107–112.

[135] P. Amoako-Yirenkyi, J. K. Appati, and I. K. Dontwi, "A new construction of a fractional derivative mask for image edge analysis based on Riemann-Liouville fractional derivative," *Advances in Difference Equations*, vol. 2016, pp. 1–23, 2016.

[136] A. K. Shukla, R. K. Pandey, and S. Yadav, "Salt and pepper noise removal algorithm based on generalized fractional operator," *International Journal of Pure and Applied Mathematics*, vol. 119, no. 16, pp. 5147–5158, 2018.

[137] A. K. Shukla, R. K. Pandey, S. Yadav, and R. B. Pachori, "Generalized fractional filter-based algorithm for image denoising," *Circuits, Systems, and Signal Processing*, vol. 39, pp. 363–390, 2020.

[138] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[139] S. Kumar, R. Saxena, and K. Singh, "Fractional Fourier transform and fractional-order calculus-based image edge detection," *Circuits, Systems, and Signal Processing*, vol. 36, no. 4, pp. 1493–1513, 2017.

[140] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[141] U. Erkan, S. Enginoğlu, and D. N. H. Thanh, "Adaptive frequency median filter for the salt and pepper denoising problem," *IET Image Processing*, vol. 14, no. 7, pp. 1291–1302, 2020.