

Kishore Bingi · Rosdiazli Ibrahim ·
Mohd Noh Karsiti · Sabo Miya Hassan ·
Vivekananda Rajah Harindran

Fractional-order Systems and PID Controllers

Using Scilab and Curve Fitting Based
Approximation Techniques

Studies in Systems, Decision and Control

Volume 264

Series Editor

Janusz Kacprzyk, Systems Research Institute, Polish Academy of Sciences,
Warsaw, Poland

The series “Studies in Systems, Decision and Control” (SSDC) covers both new developments and advances, as well as the state of the art, in the various areas of broadly perceived systems, decision making and control—quickly, up to date and with a high quality. The intent is to cover the theory, applications, and perspectives on the state of the art and future developments relevant to systems, decision making, control, complex processes and related areas, as embedded in the fields of engineering, computer science, physics, economics, social and life sciences, as well as the paradigms and methodologies behind them. The series contains monographs, textbooks, lecture notes and edited volumes in systems, decision making and control spanning the areas of Cyber-Physical Systems, Autonomous Systems, Sensor Networks, Control Systems, Energy Systems, Automotive Systems, Biological Systems, Vehicular Networking and Connected Vehicles, Aerospace Systems, Automation, Manufacturing, Smart Grids, Nonlinear Systems, Power Systems, Robotics, Social Systems, Economic Systems and other. Of particular value to both the contributors and the readership are the short publication timeframe and the world-wide distribution and exposure which enable both a wide and rapid dissemination of research output.

** Indexing: The books of this series are submitted to ISI, SCOPUS, DBLP, Ulrichs, MathSciNet, Current Mathematical Publications, Mathematical Reviews, Zentralblatt Math: MetaPress and Springerlink.

More information about this series at <http://www.springer.com/series/13304>

Kishore Bingi · Rosdiazli Ibrahim ·
Mohd Noh Karsiti · Sabo Miya Hassan ·
Vivekananda Rajah Harindran

Fractional-order Systems and PID Controllers

Using Scilab and Curve Fitting Based
Approximation Techniques



Springer

Kishore Bingi
Institute of Autonomous Systems
Universiti Teknologi PETRONAS
Perak, Malaysia

Mohd Noh Karsiti
Department of Electrical
and Electronic Engineering
Universiti Teknologi PETRONAS
Perak, Malaysia

Vivekananda Rajah Harindran
Instrumentation and Control
PETRONAS Group Technical Solutions
Petaling Jaya, Malaysia

Rosdiazli Ibrahim
Department of Electrical
and Electronic Engineering
Universiti Teknologi PETRONAS
Perak, Malaysia

Sabo Miya Hassan
Department of Electrical
and Electronics Engineering
Abubakar Tafawa Balewa University
Bauchi, Nigeria

ISSN 2198-4182 ISSN 2198-4190 (electronic)
Studies in Systems, Decision and Control
ISBN 978-3-030-33933-3 ISBN 978-3-030-33934-0 (eBook)
<https://doi.org/10.1007/978-3-030-33934-0>

© Springer Nature Switzerland AG 2020

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

This work is dedicated to:

*My parents Narasimharaao and Sudharani,
wife Sai Ramya Sri and brother Subbu for
their encouragement, care, love and support.*
—Kishore Bingi

*My beloved wife and best friend, Lidia,
and my princesses Azra, Auni, Ahna and
Ayla, my love will always be with you.*
—Rosdiazli Ibrahim

*My much-loved family, Wan Fatimah, Nuar
Noh, Redzwan Noh and Amirah Farhanah.*
—Mohd Noh Karsiti

*My beloved wife, Amina and daughters
Amatullah and Aishatu for their support
and love.*
—Sabo Miya Hassan

*My parents, wife and friendly colleagues
for their unfailing support.*
—Vivekananda Rajah Harindran

Preface

This work aims to design and develop an improved PID control strategies based on set-point weighting, structural modifications and fractional ordering for achieving adequate set-point tracking and disturbance rejection through smoother control action. In line with this, two fractional-order PID controllers, i.e., fractional-order set-point weighted PID and fractional-order PI-PD controller for process control to achieve adequate set-point tracking and disturbance rejection through smoother control action has been developed. Furthermore, the performance of the proposed fractional-order control strategies has been evaluated on real-time pH neutralization and pressure process plants over set-point tracking and disturbance rejection for smoother control action.

For practical realization or equivalent circuit implementation of the fractional-order systems and controllers, a novel curve fitting-based integer-order approximation techniques for fractional-order controllers using exact frequency response data has been developed. On the other hand, for effective implementation of fractional-order PI/PID controllers, the Scilab-based toolbox has been developed. The developed Scilab-based toolbox is the first toolbox for fractional-order systems developed in open-source software. The toolbox includes the definitions of fractional-order parameters, approximation techniques, fractional-order differentiator/integrator, fractional-order based systems, and PI/PID controllers. The toolboxes also allow time and frequency domain as well as stability analysis of the fractional-order systems and controllers. The toolbox allows the analysis of the dynamic behavior of these chaotic systems for various commensurate and non-commensurate orders using Scilab.

Therefore, with a total of five chapters, the book is structured in such a way that the sequential flow is maintained. Thus, after the introduction in Chaps. 1 and 2 gives the review on the PID controller and its modifications. Then, a review of these modified PID control strategies such as SWPID, PI-PD and FOPID controllers is provided. The chapter then provides the design of fractional-order set-point weighted control strategies in standard, industrial, ideal and parallel configurations. The simulation study on control of real-time pH neutralization process and pressure control using the developed fractional-order control strategies has also been

presented in the chapter. Chapter 3 presents the development of novel curve fitting-based approximation algorithms for fractional-order parameters. The chapter also presents the simulation study on various fractional order-based systems and controllers on the performance of the proposed curve-fitting approximation techniques. Chapter 4 develops the Scilab-based toolbox for fractional-order systems and PID controllers. The toolbox includes the definitions of fractional-order parameters, approximation techniques, fractional-order differentiator/integrator, fractional-order based systems, and PI/PID controllers. Furthermore, the toolbox also provides the time and frequency domain as well as stability analysis of the fractional-order systems. The chapter then presents the simulation study on various fractional-order based systems and controllers. Chapter 5 presents the toolbox for fractional-order chaotic systems. The toolbox includes fractional-order Van der Pol and duffing oscillators and fractional-order Lorenz, Chen and Rössler's systems. Then chapter analyzes the dynamic behavior of these chaotic systems for various commensurate and non-commensurate orders using Scilab. In the Appendix, the tuning of fractional-order set-point weighted controller parameters using the accelerated particle swarm optimization algorithm is presented.

We would like to thank Dr. Thomas Ditzinger, Editorial Director of Interdisciplinary and Applied Sciences Engineering and Prof. Dr. Janusz Kacprzyk, Series Editor of Studies in Systems, Decision and Control for acceptance of the publication of this book with Springer. We also want to thank the production team of Springer, in particular, the project coordinator Ms. Saranya Kalidoss for her professional and kind assistance.

The book was supported by Universiti Teknologi PETRONAS through the Award of Yayasan Universiti Teknologi PETRONAS Fundamental Research Grants with Nos 0153AA-H16 and 015LC0-053.

Last but not the least, the authors would like to express our gratitude to Dr. Irraivan Elamvazuthi, Ms. Saranya, Mr. Shihab, Mr. Wasif Umar, Dr. Nurlidia, Dr. Tran Duc, Mr. Sohail, Mr. Moslem, Mr. Emad, Mr. Seshu, Mr. Erman, Ms. Anika, Mr. Fuad, Mr. Atef, Mr. Rama Krishna, Mr. Arun, Ms. Manasa, Ms. Niharika, Mr. Malla Konda Reddy, Ms. Madiah, Ms. Fatihah, Ms. Swathi and Mr. Arun Kumar Kakumanu for their encouragement in the research.

Seri Iskandar, Malaysia
Seri Iskandar, Malaysia
Seri Iskandar, Malaysia
Kabale, Uganda
Petaling Jaya, Malaysia
September 2019

Kishore Bingi
Rosdiazli Ibrahim
Mohd Noh Karsiti
Sabo Miya Hassan
Vivekananda Rajah Harindran

Acknowledgements

This monograph is based on series of conference proceedings and articles that we have been published during my Ph.D. study. Therefore, it has been necessary at times to reuse some works that we previously published in various papers and publications. Even though, in many cases, the work has been modified and rewritten for this monograph, copyright permission from several publishers is acknowledged as follows.

Acknowledgement is given to the Springer Science & Business Media to reproduce material from the following papers:

- Bingi, K., Ibrahim, R., Karsiti, M. N., Hassan, S. M., Harindran, V. R.: Real-time control of pressure plant using 2DOF fractional-order PID controller. *Arab. J. Sci. Eng.* **44**(3), 2091–2102 (2019)
- Bingi, K., Ibrahim, R., Karsiti, M. N., Hassan, S. M.: Fractional order set-point weighted PID controller for pH neutralization process using accelerated PSO algorithm. *Arab. J. Sci. Eng.* **43**(6), 2687–2701 (2018)

Acknowledgement is given to the Polish Academy of Sciences to reproduce material from the following paper:

- Bingi, K., Ibrahim, R., Karsiti, M. N., Hassan, S. M., Harindran, V. R.: A comparative study of 2DOF PID and 2DOF fractional order PID controllers on a class of unstable systems. *Arch. Control Sci.* **28**(4), pp. 635–682 (2018)

Acknowledgement is given to the Walter de Gruyter to reproduce material from the following paper:

- Bingi, K., Ibrahim, R., Karsiti, M. N., Hassan, S. M., Harindran, V. R.: Frequency Response Based Curve Fitting Approximation of Fractional-order PID Controllers. *Int. J. Appl. Math. Comput. Sci.*, **29**(2), 311–326 (2019)

Seri Iskandar, Malaysia
Seri Iskandar, Malaysia
Seri Iskandar, Malaysia
Kabale, Uganda
Petaling Jaya, Malaysia
September 2019

Kishore Bingi
Rosdiazli Ibrahim
Mohd Noh Karsiti
Sabo Miya Hassan
Vivekananda Rajah Harindran

Contents

1	Introduction	1
1.1	Introduction	1
1.2	Summary	4
References		4
 Part I Fractional-order Set-Point Weighted Controllers and Approximation Techniques		
2	Fractional-order Set-Point Weighted Controllers	9
2.1	Introduction	9
2.1.1	Review on Modified PID Control Strategies	13
2.1.2	Review on Application of Fractional-order PID Controllers	18
2.1.3	Summary	19
2.2	Fractional-order Set-Point Weighted PID Controller	20
2.2.1	Standard Structure	20
2.2.2	Ideal Structure	23
2.2.3	Parallel Structure	24
2.2.4	Industrial Structure	25
2.2.5	Equivalent Configurations	27
2.2.6	Parameters of Various Controllers	31
2.2.7	Conversion of Controller Parameters	31
2.3	Fractional-order PI-PD Controller	32
2.3.1	PI-PD Controller	33
2.3.2	$\text{PI}^\lambda\text{-PD}^\mu$ Controller	35
2.3.3	Parameters of Various Controller Configurations	37
2.3.4	Conversion of Controller Parameters	38
2.4	Simulation Study on pH Neutralization Process	39
2.4.1	P&ID of pH Neutralization Process Plant	39
2.4.2	Modeling of pH Neutralization Process	42

2.4.3	Schematics of Experimental Setup	44
2.4.4	Performance of SWPI $^{\lambda}$ D $^{\mu}$ Controller	44
2.4.5	Performance of PI $^{\lambda}$ -PD $^{\mu}$ Controller	51
2.4.6	Comparison of SWPI $^{\lambda}$ D $^{\mu}$ and PI $^{\lambda}$ -PD $^{\mu}$ Controllers	59
2.5	Simulation Study on Pressure Process Plant	62
2.5.1	PI&D of Pressure Process Plant	63
2.5.2	Schematics of Experimental Setup	64
2.5.3	Performance of SWPI $^{\lambda}$ D $^{\mu}$ Controller	65
2.5.4	Performance of PI $^{\lambda}$ -PD $^{\mu}$ Controller	78
2.5.5	Comparison of SWPI $^{\lambda}$ D $^{\mu}$ and PI $^{\lambda}$ -PD $^{\mu}$ Controllers	87
2.6	Summary	94
	References	95
3	Approximation Techniques	101
3.1	Introduction	101
3.2	Curve Fitting Based Approximation Algorithm	102
3.3	Transfer Function Estimation Algorithm	105
3.4	Numerical and Stability Analysis	105
3.5	Performance Analysis of Approximation Algorithms	106
3.5.1	Curve Fitting Based Approximation	107
3.5.2	Transfer Function Estimation Algorithm	110
3.6	Approximation Table for Fractional-order Differentiator	114
3.7	Simulation Study	118
3.7.1	Fractional-order Differentiator	119
3.7.2	Fractional-order Integrator	123
3.7.3	Fractional-order PID Controller	126
3.7.4	Fractional-order Filter	128
3.7.5	Fractional-order Transfer Function	130
3.8	Summary	133
	References	133

Part II Scilab Based Toolbox for Fractional-order Systems and PID Controllers

4	Scilab Based Toolbox for Fractional-order Systems and PID Controllers	137
4.1	Introduction	137
4.2	Fractional-order Differintegral Operator	139
4.3	Approximation of Fractional-order Operator	140
4.3.1	Oustaloup Approximation	140
4.3.2	Refined Oustaloup Approximation	141
4.3.3	Matsuda Approximation	142
4.3.4	Curve Fitting Approximation Algorithm	143

4.3.5	Transfer Function Estimation	143
4.3.6	Power Series Expansion Techniques	144
4.4	Fractional-order Parameters	145
4.4.1	Fractional-order Differentiator	145
4.4.2	Fractional-order Integrator	146
4.5	Fractional-order Transfer Function	147
4.6	Fractional-order Filters	148
4.6.1	Fractional-order Low-Pass Filter	149
4.6.2	Fractional-order High-Pass Filter	150
4.6.3	Fractional-order Band-Pass Filter	151
4.6.4	Fractional-order All-Pass Filter	152
4.7	Fractional-order PID Controllers	153
4.7.1	PID Controller	153
4.7.2	I. Podlubny's Fractional-order PID Controller	153
4.7.3	M. Tenoutit's Fractional-order PID Controller	154
4.7.4	Luo's Fractional-order PI Controller	155
4.7.5	IMC Based Fractional-order PID Controller	156
4.7.6	Modified Fractional-order PI Controller	157
4.7.7	Modified Fractional-order PID Controller	158
4.8	Numerical and Stability Analysis	159
4.8.1	Time and Frequency Domain Analysis	159
4.8.2	Stability Analysis	162
4.9	Simulation Study	166
4.9.1	Fractional-order Operators and Transfer Functions	166
4.9.2	Fractional-order PID Controllers	177
4.9.3	Fractional-order Filters	195
4.10	Summary	209
	References	210
5	Scilab Based Toolbox for Fractional-order Chaotic Systems	213
5.1	Introduction	213
5.2	Fractional-order Chaotic Oscillators	214
5.2.1	Fractional-order Van der Pol Oscillator	214
5.2.2	Fractional-order Duffing Oscillator	219
5.3	Fractional-order Lorenz's Chaotic System	227
5.3.1	Commensurate Orders	231
5.3.2	Non-commensurate Orders	233
5.4	Fractional-order Chen's Chaotic System	237
5.4.1	Commensurate Orders	238
5.4.2	Non-commensurate Orders	240
5.5	Fractional-order Rössler's Chaotic System	241
5.5.1	Commensurate Orders	242
5.5.2	Non-commensurate Orders	242

5.6 Summary	244
References	245
Appendix: Tuning of Controller Parameters Using Accelerated Particle Swarm Optimization.....	247
Index	253

Acronyms

AG	Agitator
AI	Analog Input
AIC	pH Internal Controller
AO	Analog Output
APSO	Accelerated Particle Swarm Optimisation
AT	pH Sensor
CFE	Continued Fraction Expansions
CRONE	Commande Robuste d'Ordre Non Entier
CSTR	Continuous Stirred-tank Reactor
CT	Conductivity Transmitter
DISO	Dual-input Single-output
EnPID	Enhanced PID
FCV	Flow Control Valve
FIR	Finite Impulse Response
FIS	Fuzzy Inference System
FIT	Fractional Integration Tool
FOC	Fractional-order Controller
FOD	Fractional-order Differentiator
FOI	Fractional-order Integrator
FOMCON	Fractional Order Modeling and Control
FOPDT	First-order Plus Dead Time
FOS	Fractional-order Systems
FOTF	Fractional-order Transfer Function
FPGA	Field Programmable Gate Array
FSST	Fractional State Space Tool
FT	Flow Transmitter
GL	Grünwal-Letnikov
HV	Hand Valve
IAE	Integral Absolute Error
IIR	Infinite Impulse Response

IMC	Internal Model Control
ISE	Integral Squared Error
ITAE	Integral Time Absolute Error
ITSE	Integral Time Squared Error
LS	Level Sensor
MAE	Mean Absolute Error
MATLAB	Matrix Laboratory
MPC	Model Predictive Control
MSE	Mean Squared Error
Ninteger	Non-Integer
NPID	Non-linear PID
P&ID	Process and Instrumentation Diagram
PCI	Peripheral Component Interconnect
PCV	Process Control Valve
PIC	Pressure Indicating Controller
$\text{PI}^{\lambda}\text{D}^{\mu}$	Fractional-order PID
PID	Proportional Integral Derivative
$\text{PID}\tau_d$	PID-deadtime
$\text{PI}^{\lambda}\text{-PD}^{\mu}$	Fractional-order PI-PD
PI-PD	Proportional Integral—Proportional Derivative
PPI	Predictive PI
PSE	Power Series Expansion
PSO	Particle Swarm Optimisation
PT	Pressure Transmitter
RL	Riemann-Liouville
RMSE	Root Mean Square Error
Scilab	Scientific Laboratory
SISO	Single-input Single-output
SK	Sanathanan-Koerner
$\text{SWPI}^{\lambda}\text{D}^{\mu}$	Fractional-order Set-point Weighted PID
SWPID	Set-point Weighted PID
TID	Tilt-Integral-Derivative
TIOO	Two-input One-output
VE	Vessel Tank
ZN	Ziegler-Nichols

Chapter 1

Introduction



1.1 Introduction

Proportional-Integral-Derivative (PID) controllers are the most widely used controller in industry [1]. The PID controller is a three action controller that has a long history in the field of process and automation [2]. The three actions Proportional (P), Integral (I) and Derivative (D) are based on present (P), past (I) and future (D) control errors. These actions are defined as follows [3]:

- **P**—Proportional action is proportional to the current/present control error,
- **I**—Integral action is proportional to the integral of past values of control error and
- **D**—Derivative action is proportional to the derivative of future values of control error.

The conventional PID controllers are applicable to many control problems, and often achieve adequate performance without any improvement and can perform poorly in some control applications [1]. The basic problem with PID control is that with constant parameters and no direct knowledge of the overall process and thus performance is compromised [4]. Another problem with the PID controllers is that they are linear and in particular symmetric. Thus, the performance of PID in non-linear systems is variable [3]. In the PID controller, the proportional and derivative actions in the forward paths cause rapid alteration of the control signal during the set-point change. These effects are called proportional and derivative kick effects [5]. Furthermore, the control action of these conventional PID controllers is quite oscillatory which degrades the actuators faster. Therefore, these controllers need to be modified for process control applications.

On the other hand, the fractional calculus is an emerging field in mathematics with deep applications in the areas of engineering, physics, finance, chemistry, and social sciences [6]. In the last two decades, applications of fractional calculus in control of engineering problems has attracted by many researchers [7]. This is because fractional-order controllers (FOCs) based on fractional calculus methods are less sensitive to parameter changes in the controller and the controlled systems [8, 9].

At the very beginning of differential calculus, in 1695 Leibniz raised a question to L'Hôpital as “Can the meaning of derivatives with integer-order be generalized

to derivatives with non-integer-orders?" L'Hôpital was replied by another question to Leibniz: "What if the order will be $\frac{1}{2}$?" then Leibniz replied as "It will lead to a paradox, from which one-day useful consequences will be drawn." The question raised by Leibniz for a non-integer-order derivative was an ongoing topic for more than 300 years, and now it is known as *fractional calculus* [10, 11]. Fractional calculus is a generalization of ordinary differentiation and integration to arbitrary non-integer-order. In differential calculus, the n th-order differentiation of $f(x)$ is denoted as $\frac{d^n f(x)}{dx^n}$ where n is a positive integer. Thus, the extension of this theory to fractional calculus where α th-order differentiation of $f(x)$ is denoted as $\frac{d^\alpha f(x)}{dx^\alpha}$. Here, α could be rational or irrational, positive or negative, real or complex [12, 13].

Until recent times, fractional calculus was considered as a mathematical theory without any applications, but in the last few decades, there has been an explosion of research activities on the application of fractional calculus to very diverse scientific fields [6, 8, 13]. These applications include fractional-order control of engineering systems, optimal control of fractional-order dynamic systems, fundamental-order modeling of mechanical, electrical, thermal, biomedical, signal and image processing and other engineering and scientific applications [7, 9].

The fractional-order PID ($PI^{\lambda}D^{\mu}$) controller is an extension of conventional PID which has been achieved by fractional-ordering the integral and derivative actions. Thus, the controller has two more parameters than the conventional PID controller. Therefore, two more specifications to be met, which improves the performance of the overall system can be improved. The controller has been extensively used in many engineering control applications to achieve more robust and stable performance [8, 14–16]. This is because fractional-order controllers are less sensitive to parameter changes and can easily attain the property of iso-damping; a desirable property of the system referring to a state where the open-loop phase Bode plot is flat [8, 11].

The $PI^{\lambda}D^{\mu}$ controller achieves better performance compared to conventional PID in the following situations/applications:

- Performs better and achieves adequate tracking response for systems like
 - Higher-order systems [17],
 - Highly nonlinear systems [18],
 - High time-delay processes [19] and
 - Non-minimum phase systems [8].
- Achieves more robust and stable performance [20, 21].
- Attain the property of iso-damping easily.

Over the years many other modifications of PID has been proposed by researchers. Among these variants, SWPID, PI-PD and $PI^{\lambda}D^{\mu}$ received the most attention in the area of process control [3, 10, 22–24]. This is because, compared to conventional PIDs, the SWPIDs and PI-PDs have the advantage of handling both set-point tracking and load regulation separately while the fractional-order controllers provide more robust and stable control performance. However, a key issue with the practical realization or equivalent circuit implementation of fractional-order controllers

in a finite-dimensional integer-order system is the approximation of fractional-order parameters for the desired frequency range and the order of approximation [13, 25].

For an effective approximation of fractional-order integrator/differentiator, researchers have proposed several frequency-domain approximation techniques such as Oustaloup, Matsuda, Carlson, continued fractional and power series expansion techniques [11, 14, 26]. Among the frequency domain approximation techniques, Outstaloup's approximation is the most popular and widely used. However, in certain situations, the Oustaloup's approximation cannot fit around the frequency range of interest. Moreover, the modified Oustaloup produces higher-order transfer function. Similarly, the power series expansion techniques produces higher-order transfer function [10, 27]. On the other hand, the fitting range of continued fractional expansion techniques are very small and the quality of fitting is unsatisfactory [27]. Other approximation algorithms based on stability boundary locus, vector fitting method, time moments approach and frequency distribution mode are quite complex hence difficult to implement. Thus, a novel curve fitting approximation is needed that can be easy to implement and fit the entire desired frequency range.

For the implementation or practical realization of fractional-order controllers, researchers have developed few toolboxes such as CRONE, Ninteger, FOMCON [6, 8, 9, 13, 25, 28]. A common feature of these toolboxes is that they are based on MATLAB/Simulink software. A key disadvantage of these toolboxes is that there is no direct option to approximate higher-order systems. Moreover, the available techniques are mostly limited to Oustaloup and refined Oustaloup approximation approaches. Another improvement needed on the toolboxes is in the graphical, numerical and stability analysis. It should be noted that MATLAB is a powerful software that provides many numerical computations. However, despite many advantages, the software is expensive and generally have limited licenses in terms of time and number of installations [29]. On the other hand, Scilab is extremely effective and available as a free open-source software package for scientific computations. It includes hundreds of general and specialized functions for numerical computations, 2D and 3D visualization and data analysis [30–33].

Therefore, research into fractional-order PID control strategies, approximation techniques and open-source software toolboxes for real-time process control, with the view to mitigating the proportional and derivative effects for achieving adequate set-point point tracking and disturbance rejection through smoother control action is inevitable.

Thus, this book will focus on the development of fractional-order based control strategies for real-time process control to achieve adequate set-point tracking and disturbance rejection performance through smoother control action. The control methodologies to be developed will be based on set-point weighting, structural modification, and fractional-ordering of PID controllers. In this book, an attempt has been made for the real-time implementation of proposed fractional-order controllers for the control of pH neutralization and pressure process plants. The book will further concentrate on developing the integer-order approximation techniques based on curve fitting approaches using frequency response data. This book will also concentrate on developing Scilab based toolboxes for the implementation of

the fractional-order PID controllers and systems. Lastly, a Scilab based toolbox for fractional-order oscillators and chaotic systems has also been developed.

1.2 Summary

In this chapter, to simulate the interest to the readers, a background on conventional PID controllers, fractional calculus, fractional-order PID controllers, approximation techniques and toolboxes for implementation of fractional-order controllers has been presented. Finally, the scope of this book has been provided.

References

1. Ang, K.H., Chong, G., Li, Y.: PID control system analysis, design, and technology. *IEEE Trans. Control Syst. Technol.* **13**(4), 559–576 (2005)
2. Åström, K.J., Hägglund, T.: Advanced PID control. ISA-The Instrumentation, Systems, and Automation Society (2006)
3. Vilanova, R., Visioli, A.: PID Control in the Third Millennium. Springer (2012)
4. Visioli, A.: Practical PID Control. Springer Science & Business Media (2006)
5. Alfaró, V.M., Vilanova, R.: Model-reference robust tuning of 2DoF PI controllers for first-and second-order plus dead-time controlled processes. *J. Process Control* **22**(2), 359–374 (2012)
6. Machado, J.T., Kiryakova, V., Mainardi, F.: Recent history of fractional calculus. *Commun. Nonlinear Sci. Numer. Simul.* **16**(3), 1140–1153 (2011)
7. Ortigueira, M.D.: Fractional Calculus for Scientists and Engineers, vol. 84. Springer Science & Business Media (2011)
8. Shah, P., Agashe, S.: Review of fractional PID controller. *Mechatronics* **38**, 29–41 (2016)
9. De Oliveira, E.C., Tenreiro Machado, J.A.: A review of definitions for fractional derivatives and integral. *Math. Probl. Eng.* (2014)
10. Caponetto, R.: Fractional Order Systems: Modeling and Control Applications. World Scientific (2010)
11. Monje, C.A., Chen, Y., Vinagre, B.M., Xue, D., Feliu-Batlle, V.: Fractional-Order Systems and Controls: Fundamentals and Applications. Springer Science & Business Media (2010)
12. Xue, D., Chen, Y., Atherton, D.P.: Linear Feedback Control: Analysis and Design with MATLAB. Siam (2007)
13. Krishna, B.T.: Studies on fractional order differentiators and integrators: a survey. *Signal Process.* **91**(3), 386–426 (2011)
14. Li, H., Luo, Y., Chen, Y.: A fractional order proportional and derivative (FOPD) motion controller: tuning rule and experiments. *IEEE Trans. Control Syst. Technol.* **18**(2), 516–520 (2009)
15. Padula, F., Visioli, A.: Tuning rules for optimal PID and fractional-order PID controllers. *J. Process Control.* **21**(1), 69–81 (2011)
16. Sharma, R., Rana, K.P.S., Kumar, V.: Performance analysis of fractional order fuzzy PID controllers applied to a robotic manipulator. *Expert Syst. Appl.* **41**(9), 4274–4289 (2014)
17. Das, S., Saha, S., Das, S., Gupta, A.: On the selection of tuning methodology of FOPID controllers for the control of higher order processes. *ISA Trans.* **50**(3), 376–388 (2011)
18. Petráš, I.: Fractional-Order Nonlinear Systems: Modeling, Analysis and Simulation. Springer Science & Business Media (2011)
19. Luo, Y., Chen, Y.: Stabilizing and robust fractional order PI controller synthesis for first order plus time delay systems. *Automatica* **48**(9), 2159–2167 (2012)

20. Margarita, R., Sergei, V.R., José, A.T.M., Juan, J.T.: Stability of fractional order systems. *Math. Probl. Eng.* (2013)
21. Tavazoei, M.S., Haeri, M.: A note on the stability of fractional order systems. *Math. Comput. Simul.* **79**(5), 1566–1576 (2009)
22. Mudi, R.K., Dey, C.: Performance improvement of PI controllers through dynamic set-point weighting. *ISA Trans.* **50**(2), 220–230 (2011)
23. Sahu, R.K., Panda, S., Rout, U.K., Sahoo, D.K.: Teaching learning based optimization algorithm for automatic generation control of power system using 2-DOF PID controller. *Int. J. Electr. Power Energy Syst.* **77**, 287–301 (2016)
24. Zou, H., Li, H.: Improved PI-PD control design using predictive functional optimization for temperature model of a fluidized catalytic cracking unit. *ISA Trans.* **67**, 215–221 (2017)
25. Li, Z., Liu, L., Dehghan, S., Chen, Y., Xue, D.: A review and evaluation of numerical tools for fractional calculus and fractional order controls. *Int. J. Control.* **90**(6), 1165–1181 (2017)
26. Tepljakov, A., Petlenkov, E., Belikov, J.: Application of Newton's method to analog and digital realization of fractional-order controllers. *Int. J. Microelectron. Comput. Sci.* **3**(2), 45–52 (2012)
27. Valério, D., Trujillo, J.J., Rivero, M., Machado, J.T., Baleanu, D.: Fractional calculus: a survey of useful formulas. *Eur. Phys. J. Spec. Top.* **222**(8), 1827–1846 (2013)
28. Freeborn, T.J.: A survey of fractional-order circuit models for biology and biomedicine. *IEEE J. Emerg. Sel. Top. Circuits* **3**(3), 416–424 (2013)
29. Sohal, J.S.: Improvement of artificial neural network based character recognition system, using SciLab. *Optik* **127**(22), 10510–10518 (2016)
30. Campbell, S.L., Chancelier, J.P., Nikoukhah, R.: *Modeling and Simulation in SCILAB*. Springer, New York (2006)
31. Bunks, C., Chancelier, J.P., Delebecque, F., Goursat, M., Nikoukhah, R., Steer, S.: *Engineering and Scientific Computing with Scilab*. Springer Science & Business Media (2012)
32. Magyar, Z., Žáková, K.: Scilab based remote control of experiments. *IFAC Proc. Vol.* **45**(11), 206–211 (2012)
33. Rohit, M.T., Ashish, M.K.: *Digital Image Processing Using SCILAB*. Springer, Cham (2018)

Part I

**Fractional-order Set-Point Weighted
Controllers and Approximation Techniques**

Chapter 2

Fractional-order Set-Point Weighted Controllers



2.1 Introduction

The advantage of the PID controller is that it can deal with important practical issues such as actuator saturation, integrator windup. The controller is the most widely used in the process industry for the control of temperature, flow, pressure, force, position, speed, concentration and pH [1–5]. In fact, based on the recent survey published in IEEE control system magazine [6], it is clear that the PID controllers are among the top choice of control technologies in terms of industrial impact as compared to model predictive, nonlinear, adaptive, robust, hybrid, decentralized, discrete and intelligent control strategies. The summary of the survey is shown in Fig. 2.1. The survey results from the figure show that the PID controller has the highest 100% industrial-impact compared to all other control strategies.

This is because of its advantages of simplicity in design, ease of tuning and implementation [7–10]. However, the instantaneous step change in the set-point with a PID controller produces an excessive change in the controller output. This is caused by the derivative action since it takes the derivative of the error, which is very large during step change. Hence, the overall error will be very high. Furthermore, for disturbance rejection, a fast response will be achieved with a high-gain PID controller but with an oscillatory response [11–14]. On the other hand, the controller also produces undesired oscillations and rapid changes in the control signal. Therefore, reducing these undesired oscillations of a control signal is another common control objective [15]. This is because the undesired oscillations in the control signals degrade the actuators faster.

Over time, many modifications of PID has been proposed by various researchers. This is due to the failure of conventional PID structure to achieve robust performance under conditions such as the change in process dynamics, variation in set-point, high external disturbance, long deadtime. Among the various modifications of PID are Set-point Weighted PID (SWPID), Predictive PI (PPI), Fractional-order PID (FOPID or $\text{PI}^\lambda \text{D}^\mu$), Non-linear PID (NPID), PI-PD, Enhanced PID (EnPID) and PID-deadtime

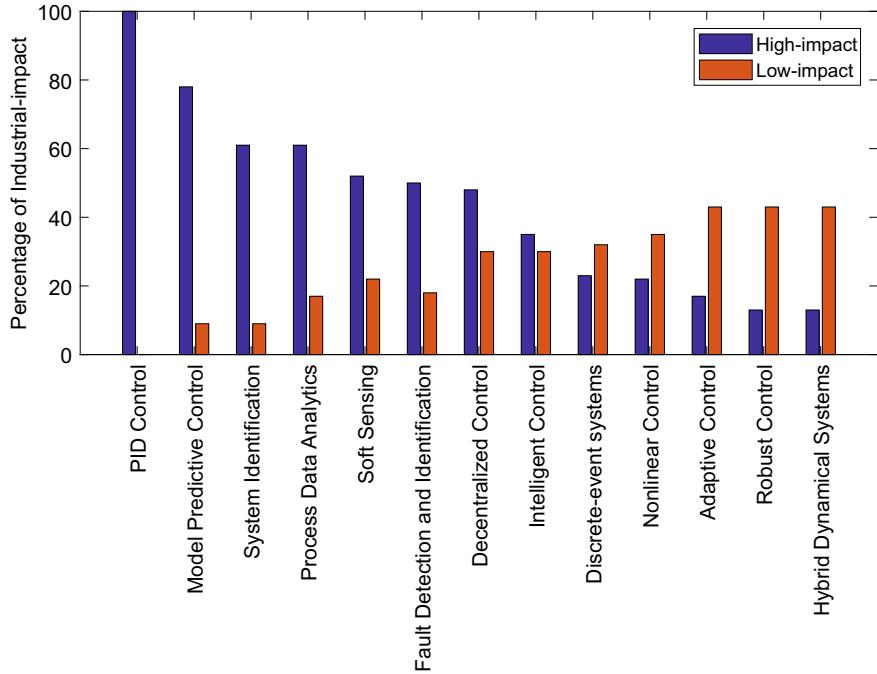


Fig. 2.1 Industrial impact of low-level and advanced control strategies

($\text{PID}\tau_d$) controller. The features and limitations of these modified PID controllers are given in Table 2.1.

Of these PID variants from the table, the SWPID, PI-PD and $\text{PI}^\lambda\text{D}^\mu$ received the most attention recently in the area of process control. This is because, compared to conventional PIDs, the SWPIDs and PI-PDs have the advantage of handling both set-point tracking and load regulation separately. On the other hand, the fractional-order controllers provide more robust and stable control performance. This is because fractional-order PID is more flexible and less sensitive to parameter changes in the controller and the systems to be controlled [7, 16]. For example, consider the conventional integrator and differentiator in PID as $\frac{1}{s^n}$ and s^n , $n \in \mathbb{R}^+$ respectively, the positive effects of these control actions in a closed-loop system are defined as follows:

- the integral action eliminates the steady-state errors which can be deduced by an infinite gain at zero frequency and
- the derivative action increases the relative stability which is achieved by introducing $\frac{\pi}{2}$ phase lead.

On the other hand, the negative effects of these control actions are as follows:

- the integral action decreases the relative stability by introducing $\frac{\pi}{2}$ phase lag and
- the derivative action increases the sensitivity to high-frequency noise by increasing gain with the slope of 20 dB/dec.

Table 2.1 Features and limitations of several modified PID controllers

Controller	Features	Limitations	References
SWPID	Capable of achieving fast disturbance rejection without significant increase in overshoot during set-point tracking. Also, reduces both proportional and derivative kick effects	Produces undesired oscillations and rapid changes in control signal also increases the number of controller parameters to the conventional PID	[10, 11, 17–27]
PPI	Capable of time delay/deadtime compensation even for long time delay process	Unable to achieve robust performance in the presence of high external disturbance and noise	[14, 28–30]
PI-PD	Capable of avoiding derivative kick effect during set-point change and has the same number of controller parameters as that of conventional PID	Produces oscillatory control signal, high overshoot during set-point tracking and slow disturbance rejection	[31–41]
NPID	High robustness against uncertainties, delays and noise. Thus, achieving fast set-point tracking response and prevent oscillatory behavior	Causes the problems of integral windup, stability and high noise rejection. Also, doubles the number of controller parameters as that of conventional PID	[42–45]
PID_{τ_d}	Capable of having deadtime compensation with functionality similar to model based controllers. Also, achieves satisfactory performance over disturbance rejection	Failed to achieve robust performance in the presence of high external disturbance	[28, 29]
$\text{PI}^{\lambda}\text{D}^{\mu}$	More robust and stable performance even for higher-order systems characterized by high non-linearities and delays. Also, attains the property of iso-damping ^a easily	Produces high overshoot during set-point tracking and change in set-point which is caused by kick effects. Also, difficulties may raise on approximation and implementation of controllers	[7, 46, 59, 68]

^aA desirable property of the system referring to a state where the open-loop phase Bode plot is flat [7]

From these effects, it can be concluded that satisfactory performance can be achieved by combining and compromising between these effects. Furthermore, it can also be noted that the gain and phase of integrator/differentiator are only in the orders of 20 dB/dec and $\frac{\pi}{2}$. Thus, more powerful and flexible designs than the conventional can be achieved with the use of fractional-order integrator and differentiator. With fractional-order integrator ($\frac{1}{s^\alpha}$) and differentiator (s^α), $\alpha \in \mathbb{R}$, the gain and phase can be achieved in the orders of $\alpha \times 20 \text{ dB/dec}$ and $\frac{\alpha\pi}{2}$ respectively. An example of

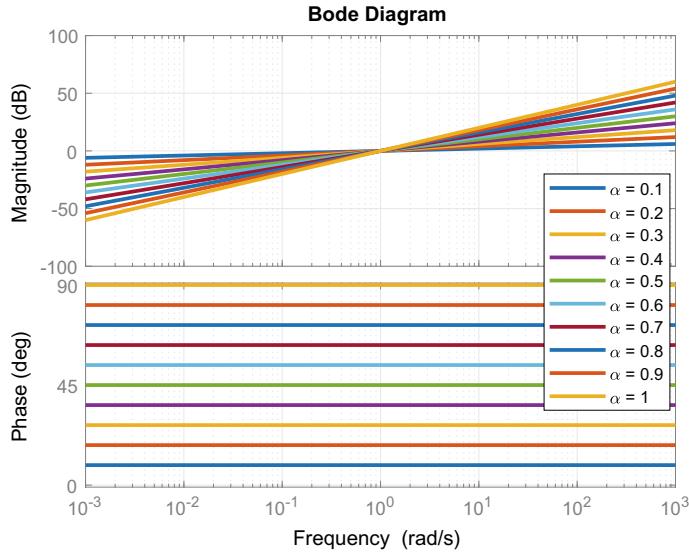


Fig. 2.2 Bode plot of fractional-order differentiator for orders $0 < \alpha < 1$

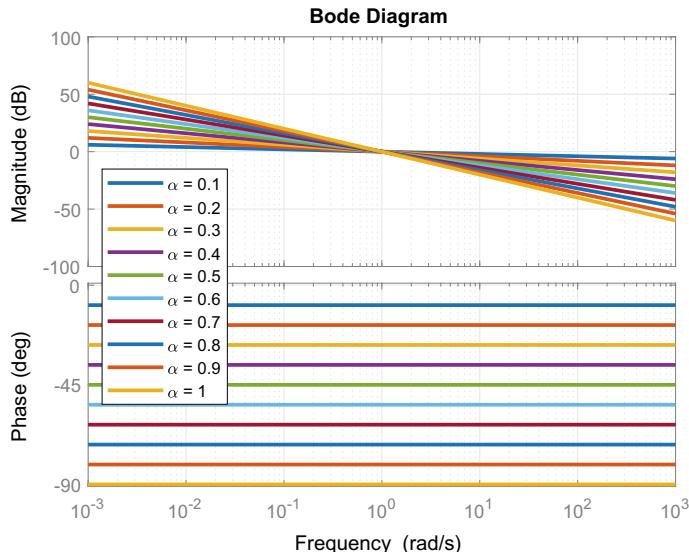


Fig. 2.3 Bode plot of fractional-order integrator for orders $0 < \alpha < 1$

such flexibility with fractional-order differentiator and integrator for $0 < \alpha < 1$ are shown in Figs. 2.2 and 2.3 respectively. From the figures, it can also be observed that for $\alpha = 1$, the characteristics of the fractional-order differentiator and integrator is the same as the conventional differentiator and integrator.

2.1.1 Review on Modified PID Control Strategies

This section reviews three modified or improved configurations of the conventional PID controller. This section is divided into three subsections. In the first subsection, the review on SWPID control strategy is presented. Then, the review on the PI-PD controller followed by the review on FOPID controller is presented.

2.1.1.1 Set-Point Weighted PID Controller

The set-point weighing strategy for PID controller has been attempted by many researchers [10, 17, 21, 23, 25, 47, 48]. Generally, these strategies focus on achieving an adequate set-point tracking performance while maintaining the simplicity of the PID controller. The other feature of this control strategy is that it can reduce the derivative kick effect during the set-point change. The derivative kick effect is the spike in the control signal due to an abrupt change of the set-point.

On the other hand, the implementation of the control strategy can also be done in two ways i.e., single-input-single-output (SISO) and dual-input-single-output (DISO) configurations. Thus, maintaining the advantage of PIDs easy of implementation. Furthermore, based on the set-point weighting parameters (b and c) of various controller structures can be achieved as given in Table 2.2. On the set-point weighing strategies, for instance, the strategy reported in [21] is a component separated type SISO configuration. The developed control strategy has been implemented for a class of unstable time-delay processes including real-time processes like isothermal continuous stirred-tank reactor (CSTR) and CSTR with non-ideal mixing. Furthermore, the controller parameters including the set-point weighting parameters were tuned using heuristic optimization algorithms. The problem with this method is that it does not solve the problem of derivative kick effect during the set-point change. Furthermore, the researcher has not evaluated the performance over variation in set-point. Another disadvantage of this method is that the control signal rapidly changes during

Table 2.2 Parameters of controllers derived from SWPID controller

Controller		Set-point weighting parameters	
Category	Type	b	c
PID	P	1	0
	PD	1	1
	PI	1	0
	PID	1	1
SWPID	SWPD	1	$0 < c < 1$
	SWPI	$0 < b < 1$	1
	SWPID	$0 < b < 1$	$0 < c < 1$
	SWPI-D	$0 < b < 1$	0

transient response which causes the high overshoot and oscillatory response in the behavior.

To avoid the problem of derivative kick effect, the authors of [13] has proposed the selection of derivative set-point weighting (c) as either 0 or 1. Thus, to avoid the extreme instantaneous change in the controller output during set-point change c is usually set to zero. With the use of this method, the same authors in [48–51] has developed robust tuning algorithms based on maximum sensitivity function and model-reference optimization procedures. The developed tuning procedures have been validated for a simulation study on stable and unstable time-delay processes and integrating processes. However, for practical use, the control action has quite oscillatory behavior.

In [11, 18], the use of dynamic set-point weighting for PI controller has been proposed based on instantaneous process trend and immersion invariance framework respectively. The effectiveness of the proposed techniques was evaluated in the simulation environment for high-order processes. However, both the techniques are applicable only for the PI controller. Thus, neglecting the issue of derivative kick effect. On the other hand, the authors of [8, 10] proposed the conversion of controller parameters between various controller structures. Furthermore, the use of the fuzzy algorithm for tuning of controller parameters including set-point weighting has been proposed in [22].

For the implementation of SWPID controller, various equivalent configurations such as pre-filter, feedforward, feedback type have been proposed in [52]. A completely different type of structure based on feedforward configuration has been proposed for magnetic levitation system in [19]. However, the proposed configuration has failed to achieve adequate disturbance rejection performance.

2.1.1.2 PI-PD Controller

The PI-PD controller has been used for process control due to excellent control performance and robustness [33, 34, 36]. The advantage of this control strategy is that the inner PD feedback controller converts the integrating or unstable plant to a stable open-loop system. Then, the cascaded PI controller controls the overall stable process [14]. Another advantage of the controller is that it has the same number of parameters as that of PID. Furthermore, the controller has the capability of reducing derivative kick effect without producing high overshoot during the set-point change.

Research on improvement of PI-PD controller performance has been going on for some time. For instance, in [32], the author proposed a model-based design of PI-PD controller for the control of unstable and integrating processes. Moreover, the controller parameters were obtained using the relay-based auto-tuning approach reported by the same author in [31]. The extension of the developed control strategy for stable processes has been reported in [33]. A similar type of relay-based PI-PD controller structure is also reported in [35, 36]. However, the proposed methods are only applicable to model-based approaches. Furthermore, in all the works the

implementation is done only in the simulation environment and also not evaluated for variation in set-point.

In [34], an improved version of the PI-PD controller has been proposed using a predictive functional optimization algorithm. The control strategy achieved better performance over PID and conventional PI-PD. However, the major industrial constraints on the algorithm like output range must be within the limits and the inputs boundary range should be based on the output demand are not considered. In a related development, the authors of [39] implemented the cuckoo optimization algorithm based PI-PD controller for the control of stable and unstable systems.

Similarly, the authors of [37, 38] implemented the cascaded PI-PD controller for automatic generation control of power systems. In both, the parameters of the controller were tuned using fractal search, pattern search, and grey wolf optimization techniques. In all these reported works, the PI-PD controller achieved better performance over conventional PID but produces an oscillatory control action. Furthermore, there is no reported work on robust performance analysis of the developed control strategy in the real-time environment.

2.1.1.3 Fractional-order PID Controller

In last two decades, there is a continuous growth in the design and development of fractional calculus for feedback control problems as shown in Fig. 2.4 [7, 16, 53–55]. From the figure, it can be seen that over the years, researchers are developing several fractional-order forms of conventional systems and controllers. For example, Igor Podlubny was the first person reported on fractional-order systems and fractional-order PID controller (FOPID or $\text{PI}^\lambda \text{D}^\mu$). Here, the fractional-order form of the conventional PID has achieved by fractional-ordering the integral and derivative actions with λ and μ respectively.

Thus, the control action of fractional-order PID ($\text{PI}^\lambda \text{D}^\mu$) from conventional PID is obtained by fractional ordering the integral and derivative actions with λ and μ as follows:

$$U(s) = \left(K_p + \frac{K_i}{s^\lambda} + \frac{K_d s^\mu}{\alpha K_d s^\mu + 1} \right) E(s), \quad 0 < \lambda, \mu < 1 \quad (2.1)$$

where

- λ is the order of integration and
- μ is the order of differentiation.

Similarly, the control actions of standard and industrial configuration of $\text{PI}^\lambda \text{D}^\mu$ controller with fractional-order integral and derivative actions is given as

$$U(s) = K_p \left(1 + \frac{1}{T_i s^\lambda} + \frac{T_d s^\mu}{T_d s^\mu + 1} \right), \quad 0 < \lambda, \mu < 1 \quad (2.2)$$

2017	Identification Algorithms	Vector Curve Fitting	HOL4
2016	LMS Filter, Model reference adaptive control	stability boundary locus	-
2015	Caputo Fractional-order derivatives	Time-domain Oustaloup	forlocus()
2014	Fractional-order Filters, Fractional-order Sliding Mode Control	Frequency Distribution Mode	gldfdiff()
2013	Modified FOPIID	Time-moments Approach	FOPID, FIT
2012	Observer based Robust Stability	-	dfod3()
2011	IMC-FOPID, $(\text{PI})^n$, $(\text{PID})^n$, Nonlinear FOPIID	State space Approach	FOMCON, INV LAP
2010	FO[PI], FO[PD], Fractional-order MPC	-	DFOC
2009	Fractional Optimal Control	-	Sysquake FOPID, @fotf
2008	Smith-predictor based fractional-order integral	-	irid_fod(), ml_fun(), gm1_fun()
2007	Controllability and Observability	H_2 pseudo-rational Approximation	fderiv()
2006	Discrete Fractional-order State space, Kalman Filter	Modified Oustaloup	ml_func()
2005	Fractional Robust System Control	-	mlf()
2004	Fractional-order Lead Compensator	-	NINTEGER
2003	Norms of fractional-order Controllers	Oustaloup, Tustin	dfod1(), dfod2(), ora_foc(), FOSS
2002	H-infinity Norms, Analog Realization	-	-
2001	Fractional-order digital FIR differentiator	-	-
2000	Frequency domain Approach for FOPIID	continued fractional expansions	CRONE, ban()
1999	Fractional-order Systems, Fractional-order PID	-	Num Inverse of Laplace
1993-1998	CRONE Control, TID Controller, Fourier Transforms	Charef, Matsuda, Fractional Taylor Series	-
	Controllers & Others	Apprximation Techniques	Scripts & Toolboxes

Fig. 2.4 Evolution of fractional-order based controllers and other related techniques

$$U(s) = K_p \left(1 + \frac{1}{T_i s^\lambda} \right) \left(\frac{T_d s^\mu + 1}{\alpha T_d s^\mu + 1} \right), \quad 0 < \lambda, \mu < 1 \quad (2.3)$$

Observing (2.1), (2.2) and (2.3), it can be noted that the $\text{PI}^\lambda \text{D}^\mu$ controller has two more parameters (i.e., λ and μ) than the conventional PID. Therefore, optimal selection of K_p , K_i , K_d , λ and μ in (2.1) will improve the overall system performance. This is because the fractional-order controllers are less sensitive to parameter changes of the controller and the system to be controlled. Furthermore, the controller also attain the property of iso-damping easily.

The fractional-order form of several advanced controllers such as sliding mode control, optimal control, model reference adaptive control and model predictive control has been proposed by researchers. Another important component in the feedback loop is the noise filter. The performance of conventional filters can also be improved with the use of fractional-order filters. Observing all the fractional-order based controllers and filters, it can be noted that the development of these fractional-order controllers (FOC) is only happened because of the numerical solutions for fractional-order derivatives defined by Grünwald-Letnikov, Riemann-Liouville, and Caputo [56, 57]. However, a key issue with the practical realization or equivalent circuit implementation of FOCs in a finite-dimensional integer-order system is the approximation of the fractional-order operators (fractional-order differentiator and integrator) for the desired frequency range. Over the years, researchers proposed several approximation techniques including continued fraction, power series expansions and curve fitting approaches [57–61, 68].

The researchers also proposed several MATLAB and Simulink based toolboxes for the implementation of fractional-order controllers. These toolboxes include CRONE, Non-Integer (NINTEGER), fractional state space tool (FSST) and Fractional Order Modeling and Control (FOMCON) [53]. Apart from these toolboxes, several fractional-order functions also available from MATLAB central file exchange as given in Fig. 2.4. Researchers also developed the fractional-order numerical functions in other software packages like Wolfram Mathworld and Sysquake [54]. These functions also include time and frequency domain analysis as well as numerical and stability analysis of the controllers [55]. Thus, it can be concluded that there is a continuous growth in the application of fractional-order controllers for feedback control problems.

The fractional-order PID ($PI^{\lambda}D^{\mu}$) controller has been extensively used in many engineering control applications to achieve more robust and stable performance [7, 62–67]. This is due to the fact that fractional-order controllers are less sensitive to parameter changes and can easily attain the property of iso-damping [7, 68]. With the fractional-order PID controller, various specifications of the controller can be achieved as given in Table 2.3.

The generalized transfer function of $PI^{\lambda}D^{\mu}$ controller proposed by Podlubny has to meet two more specifications which improve or enhance the performance of the overall systems. Over the years, many forms of fractional-order PI and PID controllers have been proposed by researchers. The $PI^{\lambda}D^{\mu}$ class of controllers are referred to as a generalized or standard form of the controller derived from the conventional PID. However, as argued by Luo et al. in [69–72], the structure is not appropriate in ensuring the robust performance of the closed-loop system. Thus, the authors proposed a novel structure called FO[PI] (or $[PI]^{\lambda}$). The proposed structure was implemented on a class of fractional-order time-delay systems, motion control systems, and fixed-wing unmanned aerial vehicle. In all these cases, the control parameters were obtained using frequency domain analysis and the application is based on the simulated environment.

In a similar way, a number of authors proposed a different structure called $(PI)^n$ for a class of stable time-delay systems [73–75]. Here, to determine the controller param-

Table 2.3 Parameters of controllers derived from $\text{PI}^\lambda \text{D}^\mu$ controller

Controller		Fractional-order parameters	
Category	Type	λ	μ
PID	P	0	0
	PI	1	0
	PD	0	1
	PID	1	1
$\text{PI}^\lambda \text{D}^\mu$	PI^λ	$0 < \lambda < 1$	0
	PD^μ	0	$0 < \mu < 1$
	$\text{PI}^\lambda \text{D}^\mu$	$0 < \lambda < 1$	$0 < \mu < 1$

eters, output feedback approach was adopted [73] using time moments approach in [75] and steepest descent method in [74]. These approaches indicated that the performance of the controller is more robust to gain variants. Similarly, to the frequency domain approach application here is also limited to the simulation environment. On the other hand, the nonlinear form of the fractional-order PID controller is a combination of NPID in [42–45] and FOPID controllers [76]. This type of controller can improve the set-point tracking properties of the given closed-loop system with a smaller design effort. However, it is very difficult to tune the eight controller parameters.

In a related development, the internal model controller (IMC) of fractional-order PID (IMC- $\text{PI}^\alpha \text{D}^\alpha$) reported in [4, 77, 78] is an extension of $\text{PI}^\lambda \text{D}^\mu$ in which $\gamma s^\alpha + 1$ is added to make proper transfer function and derived with an assumption of $\lambda = \mu = \alpha$. The one of the objective of this class of $\text{PI}^\lambda \text{D}^\mu$ controller is to reduce the controller parameters. Thus, a modified version of FOPID controller reported in [79–81] has only three controller parameters. These proposed structures exhibits a flat phase band-limited lead contribution, and improves the robustness of the system.

2.1.2 Review on Application of Fractional-order PID Controllers

The industrial perspective of exploiting the expertise of systems and control engineering has a significant wide spectrum. The objective of the industrial process control is to develop high-performance feedback systems, adaptive and self-tuning controllers taking care of the changes in process dynamics, nonlinear, robust and optimal control policies to meet a predefined set of performance criteria, fault-tolerant schemes, and safety-critical applications. Example of such sectors fall in this field are sensor and transducer manufacturing processes, chemical process industries, robot manufacturing industries, energy industries processing all forms of resources and energy transmission industries, marine and automotive sectors and metal forming industries [56, 82, 83]. The application of fractional-order controllers in some of these sec-

Table 2.4 List of patents related to fractional-order systems and controllers

Year	Patent number	Title	Reference
1994	US5371670A	Three-parameter tunable tilt-integral-derivative (TID) controller	[97]
2004	US6678670B2	Non-integer order dynamic systems	[98]
2006	WO2006112976A2	Electrical component with fractional-order impedance	[99]
2009	US7599752B2	Tuning methods for fractional-order controllers	[100]
2016	US9305706B2	Fractional-order capacitor	[101]

tors has been studied by various researchers. An example of such applications are fractional-order control of bioreactor temperature [4, 84, 85], robotic manipulator [15, 86, 87], load frequency [88, 89], power control [90–92], DC motor [93], automatic voltage regulator [64, 94], wind turbine generator [64, 89], level control [95], twin-rotor system [15, 67, 87], rail vehicles, electro-hydraulic systems, electrical drives and position control [62, 96].

In all these applications, the FOPIID controller achieved better robust and stable performance compared to conventional PID. However, in the area of process control, like in the case of PID, the controller has both proportional and derivative kick effects. Furthermore, the key challenges in the implementation of the fractional-order controller involve the approximation of fractional-order operators and computational tools for analysis, simulation and implementation.

It should be noted that the patents and hardware prototypes can be seen as important factors in industrial applications and automation. Thus, the list of available patents related to fractional-order controllers are given in Table 2.4. From the table, it can be seen that there are only a few patents related to fractional-order systems and controllers. However, these patents have also been useful in the sense of achieving an ideal implementation of fractional-order controllers.

On the other hand, for hardware prototypes, the authors of [102–104] developed an analog and digital realizations of fractional-order integrator and differentiator using field-programmable gate array (FPGA). Furthermore, Tepljakov developed a hardware prototype for FOPIID controller using 8-bit Atmel AVR and 32-bit STMicroelectronics STM32F407 microcontroller for the digital implementation of the controller [105]. However, the prototype is designed mainly for laboratory experiments. For industrial use, a suitable form-factor and a suitable choice of components like manual controls and display are necessary to meet industrial standards.

2.1.3 Summary

Looking at the available literature, it is confirmed that the PID controllers are the most widely used and has the highest industrial impact compared to all other robust,

advanced and intelligent control strategies. This is because of its advantages of simplicity in design, ease of tuning and implementation. However, during the instantaneous step change in the set-point with a PID controller produces an excessive change in the controller output. Furthermore, for disturbance rejection, a fast response will be achieved with a high gain PID controller but with an oscillatory response. On the other hand, reducing undesired oscillations of a control signal is another common control objective. This is because the undesired oscillations in the control signals degrade the actuators faster. Thus, the overall objective is now to achieve adequate set-point tracking and effective load regulation through smooth control action. Therefore, to mitigate these issues, this work proposes the fractional-ordering of SWPID and PI-PD control strategies. The proposed control strategy is expected to achieve good set-point tracking and load regulation performance through smooth control action over conventional controllers.

2.2 Fractional-order Set-Point Weighted PID Controller

In the first part of this section, the design of fractional-order set-point weighted PID (SWPI ^{λ} D ^{μ}) controller in standard, industrial, parallel and ideal configurations of PID are presented [106]. Then, the implementation of the controller in dual-input single-output and single-input single-output configurations are presented. Lastly, the conversion of controller parameters from one form to other inducing the effect of variation on these controller parameters is presented.

2.2.1 Standard Structure

Consider the closed-loop control system of Fig. 2.5, the control signal ($U(s)$) of a standard configuration of a conventional PID controller is defined as follows:

$$U(s) = K_p \left(1 + \frac{1}{T_i s} + \frac{T_d s}{\alpha T_d s + 1} \right) E(s) \quad (2.4)$$

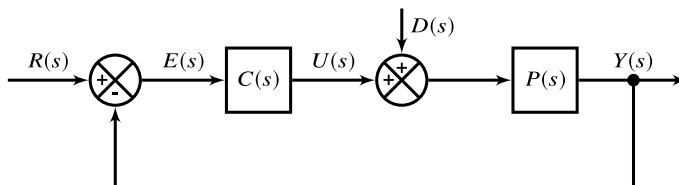


Fig. 2.5 Representation of simple closed-loop process control system

where

- K_p is the proportional gain,
- T_i is the integral time constant,
- T_d is the derivative time constant and
- α is the derivative filter constant.

Thus, from $U(s)$ in Eq.(2.4) it can be seen that the three control actions are functions of error signal ($E(s)$) defined as

$$E(s) = R(s) - Y(s) \quad (2.5)$$

where $R(s)$ is the reference signal and $Y(s)$ is the process output.

The control signal of set-point weighted PID (SWPID) is obtained by treating the set-point $R(s)$ and the process output $Y(s)$ separately. In order to achieve this, the control signal of PID in Eq. (2.4) can be written with error terms associated with each of the proportional, integral and derivative actions as follows:

$$U(s) = K_p \left(E_p(s) + \frac{1}{T_i s} E_i(s) + \frac{T_d s}{\alpha T_d s + 1} E_d(s) \right) \quad (2.6)$$

In the above equation, $E_p(s)$, $E_i(s)$ and $E_d(s)$ are the error terms associated with each of the proportional, integral and derivative actions. Therefore, the control action of set-point weighted PID controller is obtained by weighting the reference signal of the proportional and derivative actions of $U(s)$ in Eq. (2.6) with set-point weights b and c respectively as:

$$E_p(s) = bR(s) - Y(s), \quad 0 < b < 1 \quad (2.7)$$

$$E_i(s) = R(s) - Y(s) \quad (2.8)$$

$$E_d(s) = cR(s) - Y(s), \quad 0 < c < 1 \quad (2.9)$$

where b is the set-point weight on the proportional action and c is the set-point weight on the derivative action.

From $E_i(s)$ given in Eq. (2.8), it should be noted that to avoid steady-state control error, the error associated with the integral action is not weighted. Thus, by substituting $E_p(s)$, $E_i(s)$ and $E_d(s)$ from Eqs. (2.7), (2.8) and (2.9) respectively in $U(s)$ of Eq. (2.6), the control action can be written as

$$\begin{aligned} U(s) &= K_p \left(b + \frac{1}{T_i s} + \frac{c T_d s}{\alpha T_d s + 1} \right) R(s) \\ &\quad - K_p \left(1 + \frac{1}{T_i s} + \frac{T_d s}{\alpha T_d s + 1} \right) Y(s), \quad 0 < b, c < 1 \end{aligned} \quad (2.10)$$

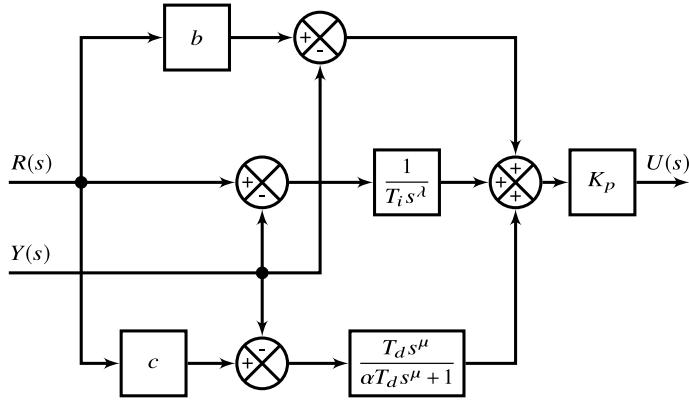


Fig. 2.6 Block diagram of standard configuration of SWPI $^\lambda$ D $^\mu$ controller

From the above control action, the control signal of fractional-order set-point weighted PID (SWPI $^\lambda$ D $^\mu$) is obtained by fractional-ordering the integral and derivative actions. Thus, the control action of SWPI $^\lambda$ D $^\mu$ controller from the control signal of SWPID in Eq. (2.10) with fractional-order integral and derivative actions of orders λ and μ respectively as follows:

$$U(s) = K_p \left(b + \frac{1}{T_i s^\lambda} + \frac{c T_d s^\mu}{\alpha T_d s^\mu + 1} \right) R(s) - K_p \left(1 + \frac{1}{T_i s^\lambda} + \frac{T_d s^\mu}{\alpha T_d s^\mu + 1} \right) Y(s), \quad 0 < \lambda, \mu < 1 \quad (2.11)$$

where λ is the order of integration and μ is the order of differentiation.

From the control signal of SWPI $^\lambda$ D $^\mu$ controller in Eq. (2.11), it can be seen that the controller is synonymous to having two fractional-order PID (PI $^\lambda$ D $^\mu$) controllers; one for set-point tracking and the other for disturbance rejection. The block diagram implementation of Eq. (2.11) is given in Fig. 2.6. Here, for small values of weight b , reduces overshoot but results in a slower response to set-point changes. Furthermore, the weight c is usually set to zero in order to avoid derivative kick effect in the control action. In this case Eq. (2.11) reduces to

$$U(s) = K_p \left(b + \frac{1}{T_i s^\lambda} \right) R(s) - K_p \left(1 + \frac{1}{T_i s^\lambda} + \frac{T_d s^\mu}{\alpha T_d s^\mu + 1} \right) Y(s) \quad (2.12)$$

Therefore, Eq. (2.12) is implemented in a closed loop control structure of Fig. 2.7. In the figure, $C_{sp}(s)$ and $C_y(s)$ are the respective controllers applied to $R(s)$ for set-point tracking and $Y(s)$ for disturbance rejection are given in Eqs. (2.13) and (2.14) respectively.

$$C_{sp}(s) = K_p \left(b + \frac{1}{T_i s^\lambda} \right) \quad (2.13)$$

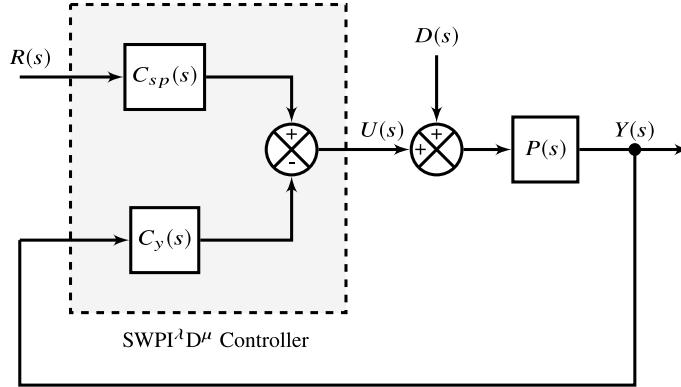


Fig. 2.7 Implementation of SWPI $^\lambda$ D $^\mu$ controller

$$C_y(s) = K_p \left(1 + \frac{1}{T_i s^\lambda} + \frac{T_d s^\mu}{\alpha T_d s^\mu + 1} \right) \quad (2.14)$$

Thus, the designed SWPI $^\lambda$ D $^\mu$ controller will respond to set-point changes and load disturbances separately. Furthermore, the fractional-order control action of the controller is smoother and avoids undesired oscillations.

2.2.2 Ideal Structure

The standard SWPI $^\lambda$ D $^\mu$ controller in Eq. (2.12) reduces to ideal controller for $\alpha = 0$. Therefore, the control signal of ideal SWPI $^\lambda$ D $^\mu$ is given as

$$U(s) = K_p \left(b + \frac{1}{T_i s^\lambda} \right) R(s) - K_p \left(1 + \frac{1}{T_i s^\lambda} + T_d s^\mu \right) Y(s) \quad (2.15)$$

Thus, the ideal SWPI $^\lambda$ D $^\mu$ controller with external first-order filter with time constant T_f is given as

$$U(s) = \left[K_p \left(b + \frac{1}{T_i s^\lambda} \right) R(s) - K_p \left(1 + \frac{1}{T_i s^\lambda} + T_d s^\mu \right) Y(s) \right] \left(\frac{1}{T_f s + 1} \right) \quad (2.16)$$

The block diagram implementation of Eq. (2.16) is given in Fig. 2.8. Figure 2.7 can also be used for implementation in a similar way to standard controller. Thus, the respective set-point weighted and disturbance rejection controllers $C_{sp}(s)$ and $C_y(s)$ are given as

$$C_{sp}(s) = K_p \left(b + \frac{1}{T_i s^\lambda} \right) \left(\frac{1}{T_f s + 1} \right) \quad (2.17)$$

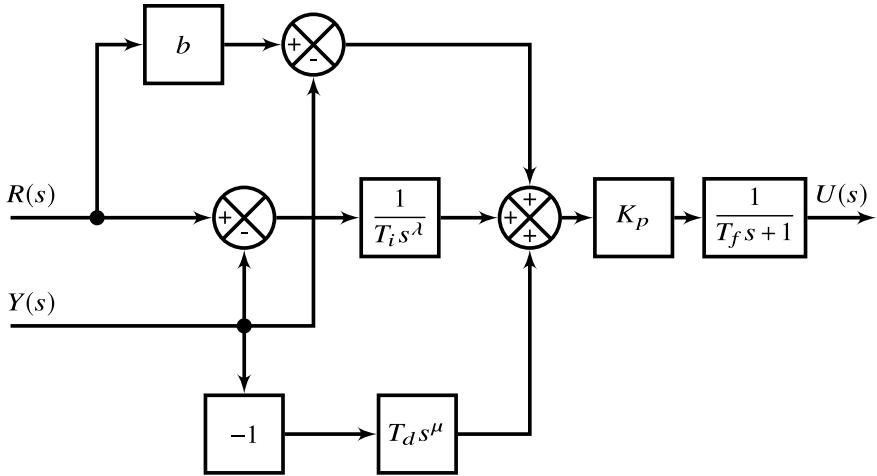


Fig. 2.8 Block diagram of ideal SWPI $^\lambda$ D $^\mu$ controller with external filter

$$C_y(s) = K_p \left(1 + \frac{1}{T_i s^\lambda} + T_d s^\mu \right) \left(\frac{1}{T_f s + 1} \right) \quad (2.18)$$

2.2.3 Parallel Structure

Similarly, from Eq. (2.12), the control signal of parallel SWPI $^\lambda$ D $^\mu$ controller is given as follows:

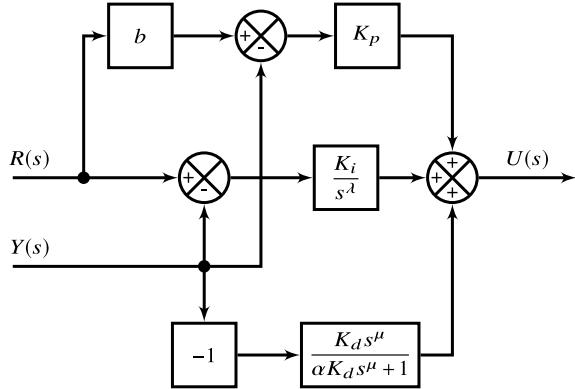
$$U(s) = \left(bK_p + \frac{K_i}{s^\lambda} \right) R(s) - \left(K_p + \frac{K_i}{s^\lambda} + \frac{K_d s^\mu}{\alpha K_d s^\mu + 1} \right) Y(s) \quad (2.19)$$

where

- K_p is the proportional gain,
- K_i is the integral gain,
- K_d is the derivative gain and
- α_p is the derivative filter constant.

It should be noted that, unlike the standard controller where the K_p appears in all actions, here, K_p is independent of integral and derivative actions. The block diagram implementation of Eq. (2.19) is given in Fig. 2.9. The closed-loop implementation of the controller is done using the block diagram in Fig. 2.7. Thus, the respective controllers $C_{sp}(s)$ and $C_y(s)$ for $R(s)$ and $Y(s)$ are given as

Fig. 2.9 Block diagram of parallel SWPI $^\lambda$ D $^\mu$ controller



$$C_{sp}(s) = bK_p + \frac{K_i}{s^\lambda} \quad (2.20)$$

$$C_y(s) = K_p + \frac{K_i}{s^\lambda} + \frac{K_d s^\mu}{\alpha K_d s^\mu + 1} \quad (2.21)$$

2.2.4 Industrial Structure

From the closed-loop control system of Fig. 2.5, the control signal ($U(s)$) for industrial configuration of PID controller is defined as follows:

$$U(s) = K'_p \left(1 + \frac{1}{T'_i s} \right) \left(\frac{T'_d s + 1}{\alpha T'_d s + 1} \right) E(s) \quad (2.22)$$

From the equation, it can be seen that this type of controller is synonymous to having PI and PD controllers in series for easy implementation in industry. In the above control signal, the ' notation is used to differentiate with the standard parameters of PID and it is for analysis purpose only.

Therefore, in a similar way to standard PID, the control signal of industrial configuration of SWPID controller is derived as

$$U(s) = K'_p \left(b' + \frac{1}{T'_i s} \right) \left(\frac{c' T'_d s + 1}{\alpha' c' T'_d s + 1} \right) R(s) - \\ K'_p \left(1 + \frac{1}{T'_i s} \right) \left(\frac{T'_d s + 1}{\alpha' T'_d s + 1} \right) Y(s), \quad 0 < b', c' < 1 \quad (2.23)$$

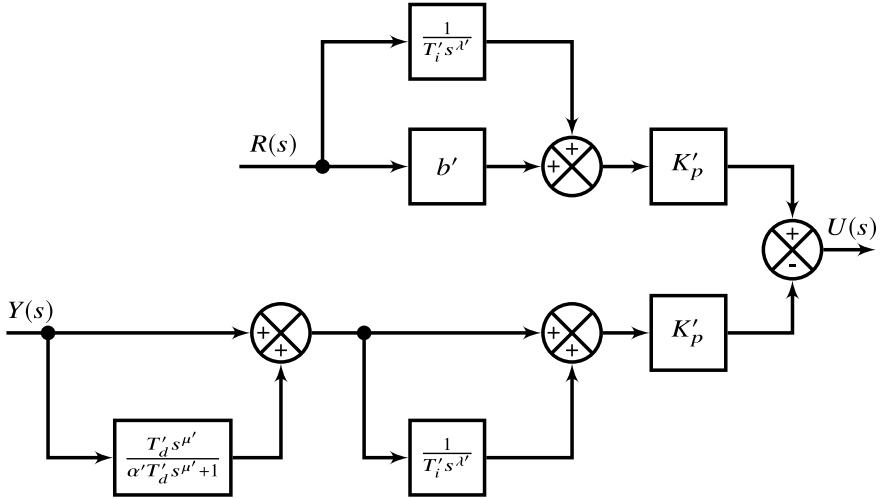


Fig. 2.10 Block diagram of industrial configuration of $\text{SWPI}^\lambda \text{D}^\mu$ controller

The same assumption of $c' = 0$ for avoiding the derivative kick effect is done here. Therefore, the simplified form of the controller is given as

$$U(s) = K'_p \left(b' + \frac{1}{T'_i s^{\lambda'}} \right) R(s) - K'_p \left(1 + \frac{1}{T'_i s^{\lambda'}} \right) \left(\frac{T'_d s + 1}{\alpha' T'_d s^{\mu'} + 1} \right) Y(s) \quad (2.24)$$

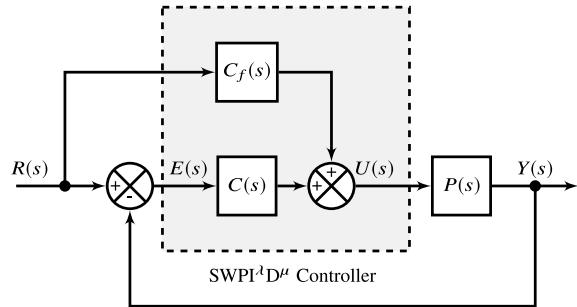
From the above control signal, the fractional-order form of the set-point weighted PID controller is obtained by fractional ordering the integral and derivative actions of the controller with λ' and μ' as follows:

$$\begin{aligned} U(s) &= K'_p \left(b' + \frac{1}{T'_i s^{\lambda'}} \right) R(s) \\ &\quad - K'_p \left(1 + \frac{1}{T'_i s^{\lambda'}} \right) \left(\frac{T'_d s^{\mu'} + 1}{\alpha' T'_d s^{\mu'} + 1} \right) Y(s), \quad 0 < \lambda', \mu' < 1 \end{aligned} \quad (2.25)$$

The block diagram implementation of industrial $\text{SWPI}^\lambda \text{D}^\mu$ controller structure based on Eq. (2.25) is shown in Fig. 2.10. Equation (2.25) is implemented in a similar way to Eq. (2.11) using Fig. 2.7. The first terms of standard $\text{SWPI}^\lambda \text{D}^\mu$ in Eq. (2.12) and industrial $\text{SWPI}^\lambda \text{D}^\mu$ in Eq. (2.25) clearly shows that both configurations have similar $C_{sp}(s)$ controller. The difference between the two configurations is in the second terms of the equations i.e., $C_y(s)$, which for the industrial configuration is an interactive PID given in Eq. (2.26).

$$C_y(s) = K'_p \left(1 + \frac{1}{T'_i s^{\lambda'}} \right) \left(\frac{T'_d s^{\mu'} + 1}{\alpha' T'_d s^{\mu'} + 1} \right) \quad (2.26)$$

Fig. 2.11 Feedforward type of SWPI $^\lambda$ D $^\mu$ controller



2.2.5 Equivalent Configurations

The standard implementation of standard, ideal, parallel and industrial configurations of SWPI $^\lambda$ D $^\mu$ controller shown in Fig. 2.7 is a dual-input and single-output (DISO) controller which can be decomposed into $C_{sp}(s)$ and $C_y(s)$ controllers. The controller can also be in other configurations such as feedforward, feedback, pre-filter and component separated type. These additional configurations take the form of single-input-single-output (SISO) controllers as explained in the following subsections.

2.2.5.1 Feedforward Configuration

In feedforward configuration, the SWPI $^\lambda$ D $^\mu$ controller is decomposed into a fractional-order forward controller $C(s)$ and a fractional-order feedforward controller $C_f(s)$ as shown in Fig. 2.11.

From the figure, the control signal ($U(s)$) is given as

$$U(s) = C_f(s)R(s) + C(s)E(s) \quad (2.27)$$

By substituting $E(s) = R(s) - Y(s)$ in Eq. (2.27), we get

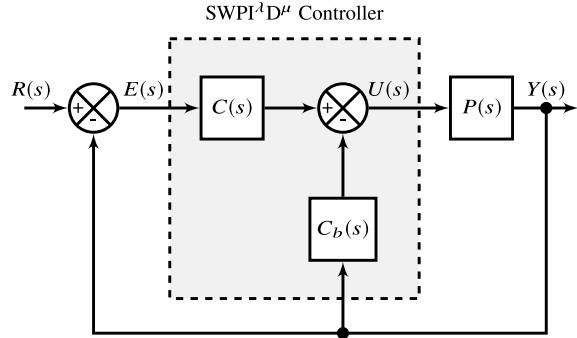
$$\begin{aligned} U(s) &= C_f(s)R(s) + C(s)(R(s) - Y(s)) \\ &= (C_f(s) + C(s))R(s) - C(s)Y(s) \end{aligned} \quad (2.28)$$

Therefore, by comparing Eqs. (2.12) and (2.28), the controllers transfer functions can be deduced to

$$C_f(s) + C(s) = K_p \left(b + \frac{1}{T_i s^\lambda} \right) \quad (2.29)$$

$$C(s) = K_p \left(1 + \frac{1}{T_i s^\lambda} + \frac{T_d s^\mu}{\alpha T_d s^\mu + 1} \right) \quad (2.30)$$

Fig. 2.12 Feedback type of SWPI $^\lambda$ D $^\mu$ controller



Thus, by substituting Eqs. (2.30) in (2.29), the transfer function of $C_f(s)$ is derived as

$$C_f(s) = K_p \left((b - 1) - \frac{T_d s^\mu}{\alpha T_d s^\mu + 1} \right) \quad (2.31)$$

2.2.5.2 Feedback Configuration

Similar to feedforward configuration, the feedback configuration of controller can be decomposed into a fractional-order forward controller $C(s)$ and a fractional-order feedback controller $C_b(s)$ as shown in Fig. 2.12.

From the figure, the control signal ($U(s)$) can be written as

$$\begin{aligned} U(s) &= C(s)E(s) - C_b(s)Y(s) \\ &= C(s)(R(s) - Y(s)) - C_b(s)Y(s) \\ &= C(s)R(s) - (C(s) + C_b(s))Y(s) \end{aligned} \quad (2.32)$$

Therefore, by comparing Eqs. (2.12) and (2.32), the controllers transfer functions can be deduced to

$$C(s) = K_p \left(b + \frac{1}{T_i s^\lambda} \right) \quad (2.33)$$

$$C(s) + C_b(s) = K_p \left(1 + \frac{1}{T_i s^\lambda} + \frac{T_d s^\mu}{\alpha T_d s^\mu + 1} \right) \quad (2.34)$$

Substituting Eq. (2.33) in Eq. (2.34), the transfer function of $C_b(s)$ is derived as

$$C_b(s) = K_p \left((1 - b) + \frac{T_d s^\mu}{\alpha T_d s^\mu + 1} \right) \quad (2.35)$$

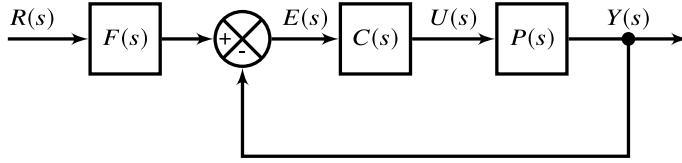


Fig. 2.13 Pre-filter type of $\text{SWPI}^\lambda \text{D}^\mu$ controller

2.2.5.3 Pre-filter Configuration

In this configuration, the controller is decomposed into a fractional-order forward controller $C(s)$ and a fractional-order pre-filter $F(s)$ on the reference signal as shown in Fig. 2.13. From the figure, the control signal is given as

$$\begin{aligned} U(s) &= C(s)(F(s)R(s) - Y(s)) \\ &= C(s)F(s)R(s) - C(s)Y(s) \end{aligned} \quad (2.36)$$

By comparing Eqs. (2.12) and (2.36), the controllers transfer functions can be simplified as

$$C(s) = K_p \left(1 + \frac{1}{T_i s^\lambda} + \frac{T_d s^\mu}{\alpha T_d s^\mu + 1} \right) \quad (2.37)$$

$$C(s)F(s) = K_p \left(b + \frac{1}{T_i s^\lambda} \right) \quad (2.38)$$

Thus, the transfer function of filter ($F(s)$) is given as

$$F(s) = \frac{b\alpha T_i T_d s^{\lambda+\mu} + b T_i s^\lambda + \alpha T_d s^\mu + 1}{(\alpha + 1) T_i T_d s^{\lambda+\mu} + T_i s^\lambda + \alpha T_d s^\mu + 1} \quad (2.39)$$

2.2.5.4 Component Separated Type

In component separated type, the proportional, integral and derivative actions of the controller are built separately as shown in Fig. 2.14. Therefore, from the figure, the control signal is given as

$$\begin{aligned} U(s) &= K_p \left((1 - \alpha)(R(s) - Y(s)) + \frac{1}{T_i s^\lambda} (R(s) - Y(s)) \right. \\ &\quad \left. + (1 - \beta) \frac{T_d s^\mu}{T_f s^\mu + 1} (R(s) - Y(s)) \right), \quad 0 < \alpha, \beta < 1 \end{aligned} \quad (2.40)$$

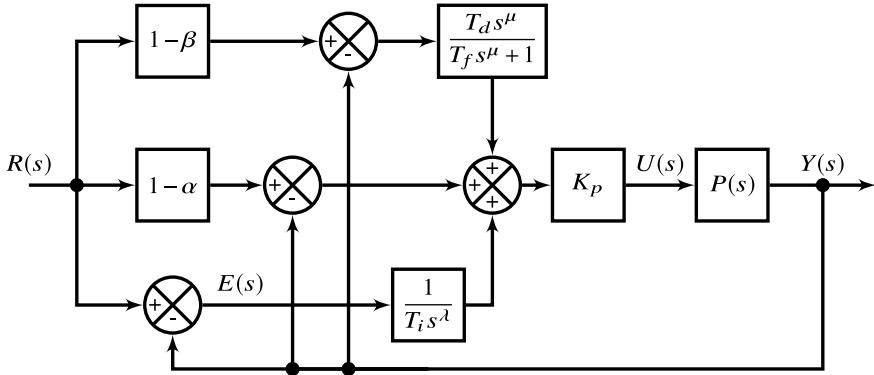


Fig. 2.14 Component-separated type of $\text{SWPI}^\lambda \text{D}^\mu$ controller

where

- α and β are the set-point weight on the proportional derivative actions and
- T_f is the derivative filter time constant.

The control signal ($U(s)$) given in Eq. (2.40) can be written in the form of controllers applied to $R(s)$ for set-point tracking and $Y(s)$ for disturbance rejection as follows:

$$U(s) = K_p \left((1 - \alpha) + \frac{1}{T_i s^\lambda} + (1 - \beta) \frac{T_d s^\mu}{T_f s^\mu + 1} \right) R(s) - K_p \left(1 + \frac{1}{T_i s^\lambda} + \frac{T_d s^\mu}{T_f s^\mu + 1} \right) Y(s) \quad (2.41)$$

Therefore, the respective controllers $C_{sp}(s)$ and $C_y(s)$ as defined in Fig. 2.7 applied to $R(s)$ for set-point tracking and $Y(s)$ for disturbance rejection are given as

$$C_{sp}(s) = K_p \left((1 - \alpha) + \frac{1}{T_i s^\lambda} + (1 - \beta) \frac{T_d s^\mu}{T_f s^\mu + 1} \right) \quad (2.42)$$

$$C_y(s) = K_p \left(1 + \frac{1}{T_i s^\lambda} + \frac{T_d s^\mu}{T_f s^\mu + 1} \right) \quad (2.43)$$

In order to avoid steady state error, β is set to one. In this case, $C_{sp}(s)$ is reduced to

$$C_{sp}(s) = K_p \left((1 - \alpha) + \frac{1}{T_i s^\lambda} \right) \quad (2.44)$$

Table 2.5 Parameters of controllers derived from standard SWPI $^{\lambda}$ D $^{\mu}$ controller

Controller		Set-point weighting parameters		Fractional-order parameters	
Category	Type	b	c	λ	μ
PID	P	1	0	0	0
	PD	1	1	0	1
	PI	1	0	1	0
	PID	1	1	1	1
SWPID	SWPD	$0 < b < 1$	$0 < c < 1$	0	1
	SWPI	$0 < b < 1$	$0 < c < 1$	1	0
	SWPID	$0 < b < 1$	$0 < c < 1$	1	1
	SWPI-D	$0 < b < 1$	0	1	1
PI $^{\lambda}$ D $^{\mu}$	I $^{\lambda}$ -PD $^{\mu}$	0	0	$0 < \lambda < 1$	$0 < \mu < 1$
	I $^{\lambda}$ D $^{\mu}$ -P	0	1	$0 < \lambda < 1$	$0 < \mu < 1$
	PI $^{\lambda}$ -D $^{\mu}$	1	0	$0 < \lambda < 1$	$0 < \mu < 1$
	PI $^{\lambda}$ D $^{\mu}$	1	1	$0 < \lambda < 1$	$0 < \mu < 1$
SWPI $^{\lambda}$ D $^{\mu}$	SWPI $^{\lambda}$ -D $^{\mu}$	$0 < b < 1$	0	$0 < \lambda < 1$	$0 < \mu < 1$
	SWI $^{\lambda}$ D $^{\mu}$ -P	0	$0 < c < 1$	$0 < \lambda < 1$	$0 < \mu < 1$
	SWPI $^{\lambda}$ D $^{\mu}$	$0 < b < 1$	$0 < c < 1$	$0 < \lambda < 1$	$0 < \mu < 1$

2.2.6 Parameters of Various Controllers

In this section, the parameters of various controllers that are derived from the SWPI $^{\lambda}$ D $^{\mu}$ controller and the effect of changing these parameters on the steady-state and transient response of the system is presented. Therefore, the tunable set-point weighting parameter (b and c) and fractional-order parameters (λ and μ) of various controllers derived from SWPI $^{\lambda}$ D $^{\mu}$ are given in Table 2.5. From the table, respective planes for the PI $^{\lambda}$ D $^{\mu}$ and SWPI $^{\lambda}$ D $^{\mu}$ for various configurations are shown in Fig. 2.15.

The PI $^{\lambda}$ D $^{\mu}$ plane in Fig. 2.15a is formed by four control points P, PI, PD, and PID. In this plane, there are two tuning knobs λ and μ to enhance the performance. This performance can be further be improved by adding two more additional tuning knobs b and c in the SWPI $^{\lambda}$ D $^{\mu}$ plane with boundaries I $^{\lambda}$ -PD $^{\mu}$, PI $^{\lambda}$ -D $^{\mu}$, I $^{\lambda}$ D $^{\mu}$ -P and PI $^{\lambda}$ D $^{\mu}$ as shown in Fig. 2.15b. Effectively, this makes the tunable knobs to four i.e., λ , μ , b and c . The effect of changing these parameters on the steady-state and transient response of the system is summarized in Table 2.6.

2.2.7 Conversion of Controller Parameters

The conversion of controller parameters from standard SWPI $^{\lambda}$ -D $^{\mu}$ controller given in Sect. 2.2.1 to industrial one presented in Sect. 2.2.4 can be deduced by comparing both the control signals as shown in Fig. 2.16. In the figure, F and F' denotes the

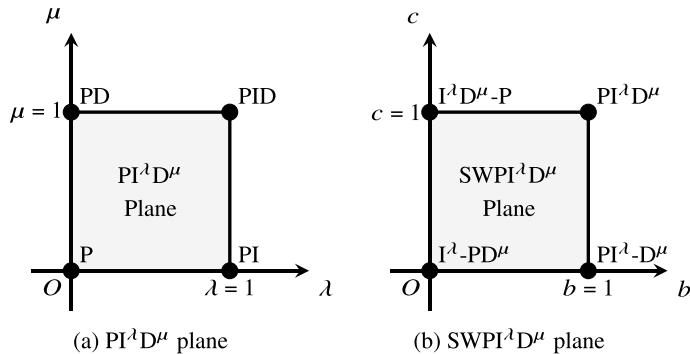


Fig. 2.15 Graphical representation of controller structures

Table 2.6 Effect of variation in parameters of standard $\text{SWPI}^\lambda\text{-D}^\mu$ controller

Parameter	Variation	Rise time	Settling time	Overshoot	Steady-state error	Stability
K_p	Increasing	Reduces	Minor change	Increases	Reduces	Degrades
T_i	Decreasing	Reduces	Increases	Increases	Eliminates	Degrades
T_d	Increasing	Minor change	Reduces	Reduces	Minor change	Improves
b	Decreasing	Increases	Reduces	Reduces	Eliminates	Improves
λ	Decreasing	Minor change	Reduces	Reduces	Increases	Degrades
μ	Decreasing	Minor change	Reduces	Reduces	Minor change	Improves

conversion factors. From the relations, it should be noted that the conversion from standard to industrial and vice versa is only possible if $1 - \frac{(4+2\alpha)T_d}{T_i} + \frac{\alpha^2 T_d^2}{T_i^2} > 0$ and $\alpha' < 1 + \frac{T'_i}{T'_d}$ respectively.

2.3 Fractional-order PI-PD Controller

In this section, the design of the fractional-order PI-PD ($\text{PI}^\lambda\text{-PD}^\mu$) controller in two single-loop control configurations is presented. The advantage of this controller is the integer-order part of the controller has the ability to achieve good set-point tracking and load regulation performance while the fractional-order part reduces the oscillations in the control signal. Next, the parameters of the various controller configurations are deduced. Finally, the conversion of these controller parameters from one configuration to another is presented.

Fig. 2.16 Conversion of SWPI $^{\lambda}$ -D $^{\mu}$ controller parameters

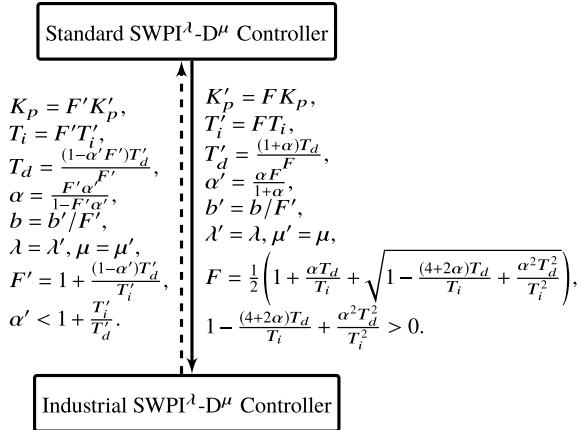
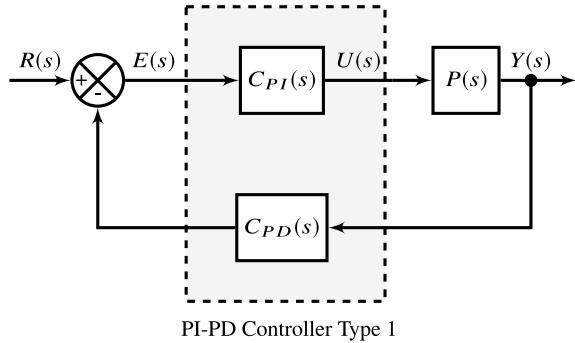


Fig. 2.17 PI-PD controller Type 1



2.3.1 PI-PD Controller

Consider the controller transfer functions of PI and PD controllers of PI-PD as defined as follows:

$$C_{PI}(s) = k_p + \frac{k_i}{s} \quad (2.45)$$

$$C_{PD}(s) = 1 + k_d s \quad (2.46)$$

where, k_p , k_i and k_d are the gains of PI-PD controller.

From Eq. (2.46), it can be noted that the proportional gain of the PD controller must be one in order to have a zero steady-state error. This type of controller can be implemented in two single-loop control configurations as shown in Figs. 2.17 and 2.18.

Observing the implementation block diagram of PI-PD controller Type 1 which is referred to as PI-PD1 in Fig. 2.17, the control signal ($U(s)$) is derived as

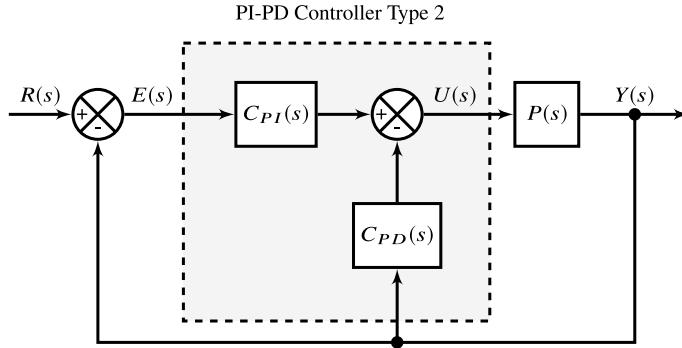


Fig. 2.18 PI-PD controller Type 2

$$\begin{aligned} U(s) &= C_{PI}(s)E(s) \\ &= C_{PI}(s)[R(s) - C_{PD}(s)Y(s)] \end{aligned} \quad (2.47)$$

where $C_{PI}(s)$ and $C_{PD}(s)$ are the PI and PD controllers defined in Eqs. (2.45) and (2.46) respectively. Thus, by substituting both the control actions in Eq. (2.47), we get

$$\begin{aligned} U(s) &= \left(k_p + \frac{k_i}{s} \right) \left[R(s) - (1 + k_d s)Y(s) \right] \\ &= \left(k_p + \frac{k_i}{s} \right) R(s) - \left((k_p + k_i k_d) + \frac{k_i}{s} + k_p k_d s \right) Y(s) \end{aligned} \quad (2.48)$$

From the above control signal, it should be noted that the controller is equivalent to SWPID controller. Therefore, the transfer functions of $C_{sp}(s)$ and $C_y(s)$ controllers applied to $R(s)$ for set-point tracking and $Y(s)$ for disturbance rejection are given as

$$C_{sp}(s) = k_p + \frac{k_i}{s} \quad (2.49)$$

$$C_y(s) = (k_p + k_i k_d) + \frac{k_i}{s} + k_p k_d s \quad (2.50)$$

Thus, the PI-PD controller has the advantage of achieving set-point weighted behavior and load disturbance rejection with same number of parameters as that of PID. Similarly, from Fig. 2.18, the control signal ($U(s)$) of Type 2 PI-PD (PI-PD2) controller is given as follows:

$$\begin{aligned} U(s) &= C_{PI}(s)E(s) - C_{PD}(s)Y(s) \\ &= C_{PI}(s)[R(s) - Y(s)] - C_{PD}(s)Y(s) \end{aligned} \quad (2.51)$$

After substituting $C_{PI}(s)$ and $C_{PD}(s)$ from Eqs. (2.45) and (2.46) in Eq. (2.51), we get

$$\begin{aligned} U(s) &= \left(k_p + \frac{k_i}{s} \right) [R(s) - Y(s)] - (1 + k_d s) Y(s) \\ &= \left(k_p + \frac{k_i}{s} \right) R(s) - \left((k_p + 1) + \frac{k_i}{s} + k_d s \right) Y(s) \end{aligned} \quad (2.52)$$

Likewise, the control signal in Eq. (2.52) is similar to SWPID controller. Here, the controller $C_{sp}(s)$ with respective to $R(s)$ is for set-point tracking and the controller $C_y(s)$ with respect to $Y(s)$ is for disturbance rejection are given as follows:

$$C_{sp}(s) = k_p + \frac{k_i}{s} \quad (2.53)$$

$$C_y(s) = (k_p + 1) + \frac{k_i}{s} + k_d s \quad (2.54)$$

Therefore, in a similar way to SWPID, the PI-PD controller responds to set-point changes and load disturbances separately. As mentioned earlier, the PI-PD retains the simplicity of PID by maintaining the same number of parameters of the PID controller. However, to achieve smoother control action with the PI-PD controller, the proposed design of fractional-order PI-PD (PI^λ - PD^μ) is presented in the following subsection.

2.3.2 PI^λ - PD^μ Controller

The controller transfer functions for fractional-order PI (PI^λ) and fractional-order PD (PD^μ) controllers of PI^λ - PD^μ controller are obtained by fractional-ordering the conventional PI and PD controllers given in Eqs. (2.45) and (2.46) with parameters λ and μ respectively as follows

$$C_{PI^\lambda}(s) = k_p + \frac{k_i}{s^\lambda}, \quad 0 < \lambda < 1 \quad (2.55)$$

$$C_{PD^\mu}(s) = 1 + k_d s^\mu, \quad 0 < \mu < 1 \quad (2.56)$$

The controller will be implemented in a similar way to the integer-order PI-PD controller as given in Figs. 2.19 and 2.20. Therefore, the control signal of Type 1 PI^λ - PD^μ (PI^λ - PD^μ 1) controller based on Fig. 2.17 is given as

$$\begin{aligned} U(s) &= C_{PI^\lambda}(s) E(s) \\ &= C_{PI^\lambda}(s) [R(s) - C_{PD^\mu}(s) Y(s)] \end{aligned} \quad (2.57)$$

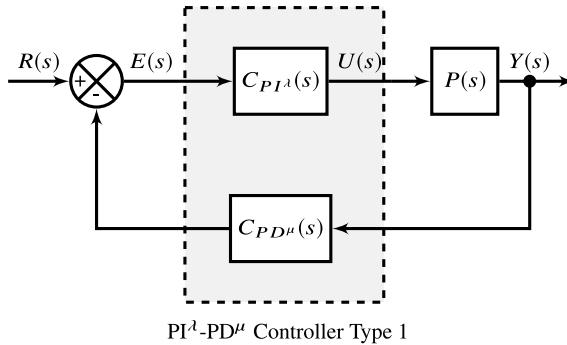


Fig. 2.19 PI $^{\lambda}$ -PD $^{\mu}$ controller Type 1

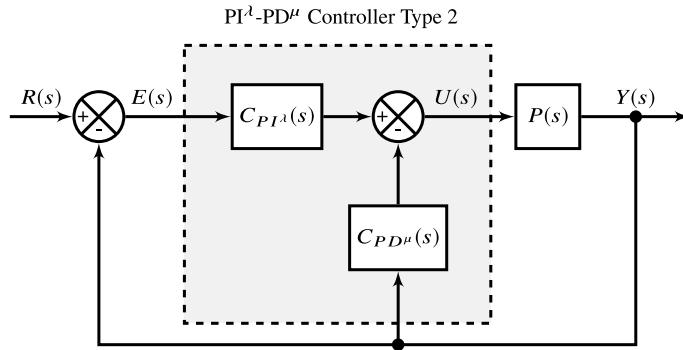


Fig. 2.20 PI $^{\lambda}$ -PD $^{\mu}$ controller Type 2

Hence, by substituting $C_{PI^\lambda}(s)$ and $C_{PD^\mu}(s)$ from Eqs. (2.55) and (2.56) in Eq. (2.57), we get

$$\begin{aligned} U(s) &= \left(k_p + \frac{k_i}{s^\lambda} \right) \left[R(s) - (1 + k_d s^\mu) Y(s) \right] \\ &= \left(k_p + \frac{k_i}{s^\lambda} \right) R(s) - \left((k_p + k_i k_d) + \frac{k_i}{s^\lambda} + k_p k_d s^\mu \right) Y(s) \end{aligned} \quad (2.58)$$

The transfer functions of $C_{sp}(s)$ and $C_y(s)$ controllers applied to $R(s)$ for set-point tracking and $Y(s)$ for disturbance rejection are given as

$$C_{sp}(s) = k_p + \frac{k_i}{s^\lambda} \quad (2.59)$$

$$C_y(s) = (k_p + k_i k_d) + \frac{k_i}{s^\lambda} + k_p k_d s^\mu \quad (2.60)$$

Thus, the controller responds to set-point changes and load disturbances separately. Furthermore, the fractional-order part of the control action is smoother and avoids undesired oscillations. It can also be noted that the controller is simple and has the same number of parameters of fractional-order PID. On the other hand, the control signal of Type 2 $\text{PI}^\lambda\text{-PD}^\mu$ ($\text{PI}^\lambda\text{-PD}^\mu 2$) controller based on Fig. 2.18 is given as

$$\begin{aligned} U(s) &= C_{PI^\lambda}(s)E(s) - C_{PD^\mu}(s)Y(s) \\ &= C_{PI^\lambda}(s)[R(s) - Y(s)] - C_{PD^\mu}(s)Y(s) \end{aligned} \quad (2.61)$$

Similarly, by substituting $C_{PI^\lambda}(s)$ and $C_{PD^\mu}(s)$ from Eqs. (2.55) and (2.56) in Eq. (2.61), we get

$$\begin{aligned} U(s) &= \left(k_p + \frac{k_i}{s^\lambda}\right)[R(s) - Y(s)] - (1 + k_d s^\mu)Y(s) \\ &= \left(k_p + \frac{k_i}{s^\lambda}\right)R(s) - \left((k_p + 1) + \frac{k_i}{s^\lambda} + k_d s^\mu\right)Y(s) \end{aligned} \quad (2.62)$$

Thus, the controller transfer functions of $C_{sp}(s)$ and $C_y(s)$ controllers applied to $R(s)$ for set-point tracking and $Y(s)$ for disturbance rejection are given in Eqs. (2.63) and (2.64) respectively.

$$C_{sp}(s) = k_p + \frac{k_i}{s^\lambda} \quad (2.63)$$

$$C_y(s) = (k_p + 1) + \frac{k_i}{s^\lambda} + k_d s^\mu \quad (2.64)$$

Observing the control actions, it can be noted that the developed $\text{PI}^\lambda\text{-PD}^\mu$ controllers have the capability of handling set-point changes and load disturbances separately through smoother control action in both the cases. Furthermore, the controller maintains the same number of parameters of the $\text{PI}^\lambda\text{D}^\mu$ controller.

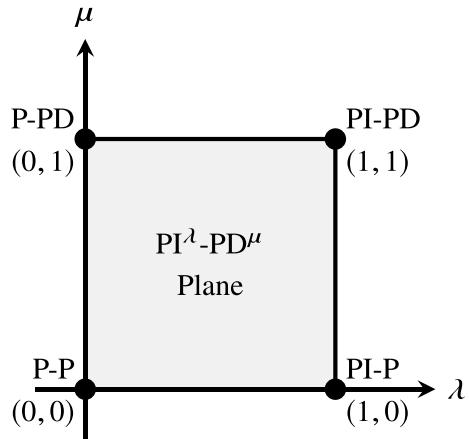
2.3.3 Parameters of Various Controller Configurations

In a similar way to the parameters of various controller configurations derived from SW $\text{PI}^\lambda\text{D}^\mu$ controller in Sect. 2.2.7, here also, the parameters of various controllers will be derived from the developed $\text{PI}^\lambda\text{-PD}^\mu$ controller. Thus, the tunable fractional-order parameters λ and μ of various controllers derived from $\text{PI}^\lambda\text{-PD}^\mu$ controller are given in Table 2.7.

The graphical representation of various controllers derived from $\text{PI}^\lambda\text{-PD}^\mu$ on a $\lambda\text{-}\mu$ plane is given in Fig. 2.21. The $\text{PI}^\lambda\text{-PD}^\mu$ plane in the figure is formed by four control points P-P, PI-P, P-PD, and PI-PD. In this plane, there are two tuning knobs λ and μ to enhance the control performance of the PI-PD controller.

Table 2.7 Parameters of controllers derived from $\text{PI}^\lambda\text{-PD}^\mu$ controller

Controller		λ	μ
Category	Type		
PI-PD	P-P	0	0
	P-PD	0	1
	PI-P	1	0
	PI-PD	1	1
$\text{PI}^\lambda\text{-PD}^\mu$	$\text{PI}^\lambda\text{-P}$	$0 < \lambda < 1$	0
	$\text{PI}^\lambda\text{-PD}$	$0 < \lambda < 1$	1
	P-PD^μ	0	$0 < \mu < 1$
	PI-PD^μ	1	$0 < \mu < 1$
	$\text{PI}^\lambda\text{-PD}^\mu$	$0 < \lambda < 1$	$0 < \mu < 1$

Fig. 2.21 Graphical representation of controller structures

2.3.4 Conversion of Controller Parameters

In this subsection, the conversion relations are developed to convert the $\text{SWPI}^\lambda\text{D}^\mu$ controller parameters presented in Sect. 2.2.3 to $\text{PI}^\lambda\text{-PD}^\mu$ controller parameters. Therefore, Eq. (2.58) can be rewritten by multiplying and dividing the term $(k_p + k_i k_d)$ as follows:

$$\begin{aligned} U(s) = & \left(\left(\frac{k_p}{k_p + k_i k_d} \right) (k_p + k_i k_d) + \frac{k_i}{s^\lambda} \right) R(s) \\ & - \left((k_p + k_i k_d) + \frac{k_i}{s^\lambda} + k_p k_d s^\mu \right) Y(s) \end{aligned} \quad (2.65)$$

Therefore, by comparing the above control signal and parallel configuration of $\text{SWPI}^\lambda\text{D}^\mu$ controller in Sect. 2.2.1, the relations are given in Fig. 2.22. The left-hand

Fig. 2.22 Conversion of controller parameters

SWPI $^{\lambda}$ -D $^{\mu}$ Controller

$$\begin{aligned} K_p &= k_p + k_i k_d, & k_p &= 0.5(K_p + \sqrt{K_p^2 - 4K_i K_d}), \\ K_i &= k_i, & k_i &= K_i, \\ K_d &= k_p k_d, & k_d &= \frac{2K_d}{(K_p + \sqrt{K_p^2 - 4K_i K_d})}, \\ b &= \frac{k_p}{k_p + k_i k_d}, & \lambda &= \lambda, \\ \lambda &= \lambda, & \mu &= \mu, \\ \mu &= \mu. & K_p^2 &> 4K_i K_d. \end{aligned}$$

PI $^{\lambda}$ -PD $^{\mu}$ Controller

side of equations is SWPI $^{\lambda}$ -D $^{\mu}$ parameters while the right-hand side is PI $^{\lambda}$ -PD $^{\mu}$ parameters. Observing the figure, it can be seen that the conversion from SWPI $^{\lambda}$ -D $^{\mu}$ to PI $^{\lambda}$ -PD $^{\mu}$ is only possible if $K_p^2 > 4K_i K_d$.

2.4 Simulation Study on pH Neutralization Process

In the first part of the section, the description of pH neutralization process plant is given with the help of Process and Instrumentation Diagram (P&ID). In the second section, the fundamental modeling of neutralization process for the titration of acetic acid with sodium hydroxide is presented. Then, the schematics of the complete entire experimental setup is presented. The subsequent three subsections present the performance evaluation of the designed SWPI $^{\lambda}$ D $^{\mu}$ and PI $^{\lambda}$ -PD $^{\mu}$ controllers on the pH neutralization process plant.

2.4.1 P&ID of pH Neutralization Process Plant

The P&ID of the pH neutralization process representing the flow is shown in Fig. 2.23 [107, 108]. From the figures, it can be seen that the process is performed in a continuous stirred-tank reactor (CSTR) where a neutralizing reagent (acid) is added to the base to regulate its pH to an acceptable value. In the process, the desired amount of acid and base streams from the source tanks VE100 and VE110 are pumped into chemical mixing tank VE120 by using the pumps P100 and P110 respectively. To regulate the flow rate of acid and base indicated by the flow transmitters FT120 and FT121, two flow control valves FCV120 and FCV121 are used.

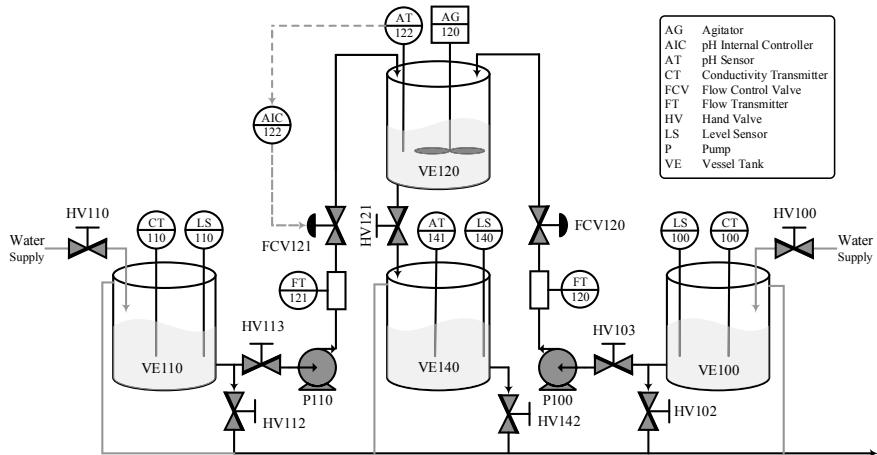


Fig. 2.23 P&ID of pH neutralization process plant



Fig. 2.24 Experimental setup of pH neutralization process plant (Rear View)

Furthermore, the acid and base concentrations are monitored continuously using two conductivity transmitters CT110 and CT110 respectively. The pH transmitter AT122 is used to measure the pH value of the mixed solution. Whereas hand valve HV121 is used to convey the neutralized solution to the storage tank and HV102, HV112 and HV142 are used to drain out the acid, storage, and base tanks respectively. The level sensors in acid, storage, and base tanks are LS100, LS110 and LS140 respectively are used to measure the level of solution tanks. The agitator AG120 in chemical mixing tank is used for mixing the solution homogeneously. On the other hand, the front and rear views of the real-time pH neutralization process plant used in the experiment are given in Figs. 2.25 and 2.24 respectively.



Fig. 2.25 Experimental setup of pH neutralization process plant (Front View)

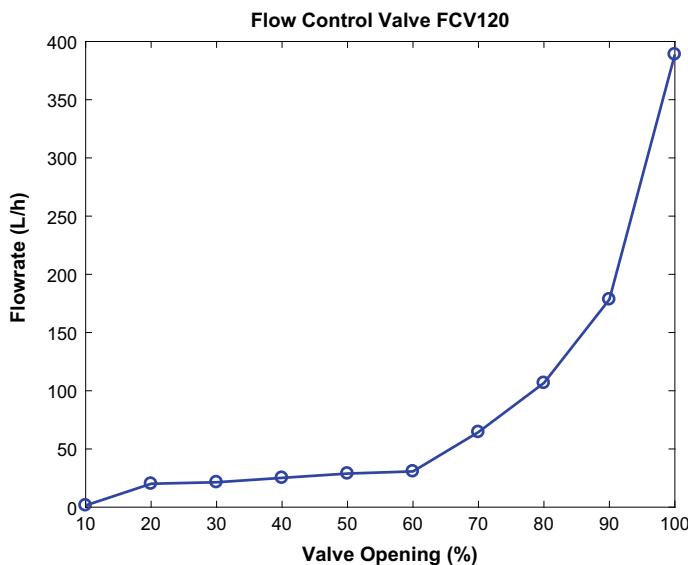


Fig. 2.26 Characteristics of electro-pneumatic control valve FCV 120

The characteristics of both the flow control valves FCV120 and FCV121 electro-pneumatic control valves are shown in Figs. 2.26 and 2.27 respectively. From the characteristics, it can be observed that the type of behavior is a nonlinear increasing sensitivity type and the maximum input flowrates are around 390 and 230 L/h.

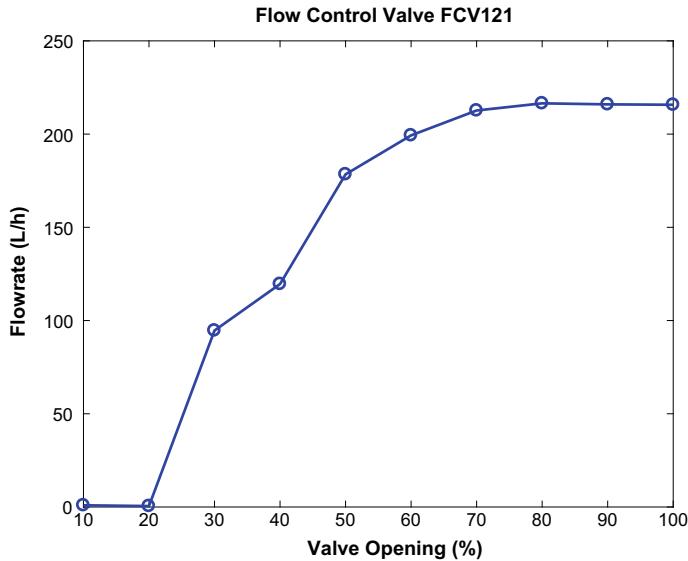


Fig. 2.27 Characteristics of electro-pneumatic control valve FCV 121

2.4.2 Modeling of pH Neutralization Process

The dynamical behavior of the CSTR is governed by the incoming flow-rates (F_a , F_b) and outgoing flow-rate ($F_a + F_b$). Therefore, the dynamical model is given by mass balance equations as:

$$V \frac{d}{dt} C[H^+] = F_a C_a - (F_a + F_b)C[H^+] \quad (2.66)$$

$$V \frac{d}{dt} C[OH^-] = F_b C_b - (F_a + F_b)C[OH^-] \quad (2.67)$$

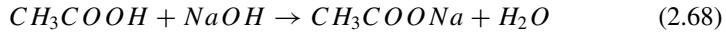
where

- V is the volume of tank VE120,
- C_a and C_b are the concentrations of acid and base and
- $C[H^+]$ and $C[OH^-]$ are the concentrations of non-reactant components H^+ and OH^- .

Consider the neutralization process of acetic acid (CH_3COOH) titrated with sodium hydroxide ($NaOH$). In this process, the control objective is to regulate the pH by controlling F_b and at a constant F_a . Therefore, the titration process model is developed based on the following assumptions:

1. all ions are completely soluble,
2. there is perfect mixing in the chemical mixing tank and
3. the reactions in the process are instantaneous and isothermal.

The first stage is to adequately model the titration process by observing the following chemical reactions occurring in the mixing tank:



From Eqs. (2.69) and (2.70) the ionic product of water (K_w) and the dissociation constant of acetic acid (K_a) are given in Eqs. (2.72) and (2.73) respectively.

$$K_w = [H^+][OH^-] = 1 \times 10^{-14} \text{ (at } 25^\circ\text{C}) \quad (2.72)$$

$$K_a = \frac{[H^+][CH_3COO^-]}{[CH_3COOH]} \quad (2.73)$$

The next stage is to derive the mass balance equations of non-reactant components in the system that relates the concentrations of dissociation products as follows:

$$C[H^+] = [CH_3COOH] + [CH_3COO^-] \quad (2.74)$$

$$C[OH^-] = [Na^+] \quad (2.75)$$

The final stage is the derivation of the electroneutrality principle. According to the principle, the sum of all positively charged ions must be equal to the sum of all negatively charged ions in aqueous solutions. Therefore, the charge balance is given as follows:

$$[H^+] + [Na^+] = [OH^-] + [CH_3COO^-] \quad (2.76)$$

In summary, by substituting Eqs. (2.72) through (2.75) in Eq. (2.76) a polynomial equation of cubic order in $[H^+]$ is obtained as follows:

$$\begin{aligned} & [H^+]^3 + [H^+]^2(K_a + C[OH^-]) + \\ & [H^+](K_a C[OH^-] - K_w - K_a C[H^+]) - K_a K_w = 0 \end{aligned} \quad (2.77)$$

The pH value, which indicates the level of concentration of hydrogen ion $[H^+]$ is calculated as

$$pH = -\log_{10}[H^+] \quad (2.78)$$

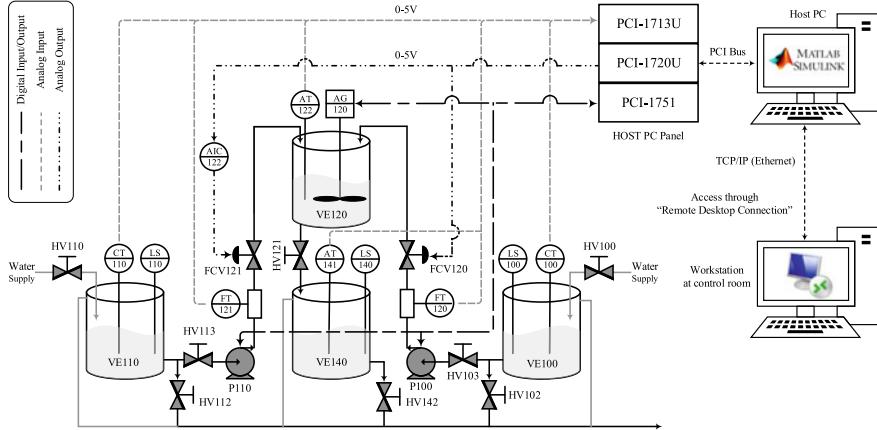


Fig. 2.28 Schematic diagram of the complete experimental setup

2.4.3 Schematics of Experimental Setup

The schematic diagram of the complete experimental setup is shown in Fig. 2.28. The developed fractional-order control strategies will be implemented in the host PC using MATLAB/Simulink software which is accessed through ‘Remote Desktop Connection’ in the control room. The host PC is interfaced with the plant through Peripheral Component Interconnect (PCI) cards 1713U, 1720U and 1751 as shown in Table 2.8. The PCI-1713U is a 100 kS/s, 12-bit, 32-ch isolated analog input (AI) card used to access signal from all the transmitters. On the other hand, the PCI-1720U is a 12-bit, 4-channel isolated analog output (AO) card used to send the control signal to flow control valves. The last card PCI-1751 is a 48-channel digital I/O and 3-channel counter card used to operate the pumps, agitator and the process plant either Remotely or Locally (R/L).

2.4.4 Performance of SWPI $^\lambda$ D $^\mu$ Controller

In this subsection, the design of SWPI $^\lambda$ D $^\mu$ controller parameters will be presented. Then, the results of the proposed standard and industrial configuration of the designed controller in comparison with PID, PI $^\lambda$ D $^\mu$ and SWPI-D for set-point tracking, variation in set-point and disturbance rejection will be presented. In all these cases, as mentioned earlier, the numerical analysis will be done in terms of rise time (t_r), settling time before and after disturbance (t_{s_1} and t_{s_2}) and percentage overshoot (%OS). Furthermore, it should be noted that a disturbance ($D(s)$) of 25% is injected at 250 s for the purpose of disturbance rejection evaluation. Moreover, the ability of the controllers to track variation in set point will also be analyzed.

Table 2.8 Channel connections of PCI cards to the devices of pH neutralization plant

PCI card	Channel	Device
PCI-1713U (Analog Input)	0	CT100
	1	FT120
	2	CT110
	3	FT121
	4	AT122
PCI-1720 (Analog Output)	0	FCV120
	1	FCV121
PCI-1751 (Digital Input/Output)	0	P100
	1	P110
	2	AG120
	3	WS-PNL

2.4.4.1 Controller Parameters

The parameters of the pH model developed earlier in Sect. 2.4.2, are given in Table 2.9, while the simulation result on the variation of titration curve for the pH process due to different concentrations of solutions is shown in Fig. 2.29. From the curve, it can be seen that the behavior of the process is highly nonlinear. The S-shaped curve is very sensitive around the equilibrium range (i.e., 6–12 pH), and thus a small perturbation of the input at this point causes a significant change of the output. Another observation made on the curve is that change in concentrations of the acid and base results in the dynamic behavior of the process.

On the other hand, for the tuning of developed $\text{SWPI}^\lambda\text{D}^\mu$ controller using APSO algorithm as discussed in Appendix A, the optimization parameters for the algorithm are also given in Table 2.9. Therefore, the tuned parameters of the controllers compared with performance index ITAE in Eq. (A.8) are given in Table 2.10.

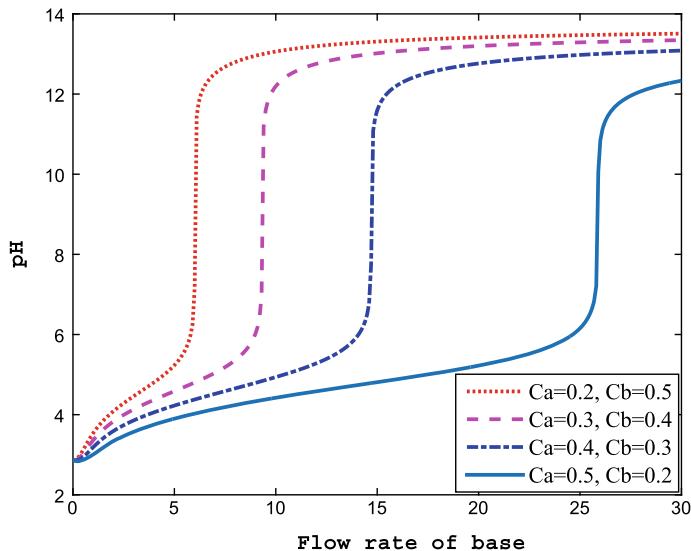
2.4.4.2 Standard $\text{SWPI}^\lambda\text{D}^\mu$ Controller

The comparison of closed-loop response of the system for set-point tracking and disturbance rejection with standard configuration of PID in Eq. (2.4), $\text{PI}^\lambda\text{D}^\mu$ in Eq. (2.2), SWPID in Eq. (2.10) and $\text{SWPI}^\lambda\text{D}^\mu$ in Eq. (2.12) is shown in Fig. 2.30.

From the figure, it can be seen that the set-point tracking ability of $\text{SWPI}^\lambda\text{-D}^\mu$ is better when compared to those of PID, $\text{PI}^\lambda\text{D}^\mu$ and SWPI-D . The figure also reveals that the action of $\text{SWPI}^\lambda\text{-D}^\mu$ gives a smoother control signal while other controllers produced undesired oscillations. This is more visible in Fig. 2.31 of the zoomed plot of regions of interest A, B, C and D. To numerical observation of Table 2.11 shows that the $\text{SWPI}^\lambda\text{-D}^\mu$ has less overshoot of 0.4802% compared to SWPI-D , $\text{PI}^\lambda\text{D}^\mu$ and PID with overshoots of 0.9536, 4.4836 and 4.9182% respectively. From the table, it can

Table 2.9 Model and optimization parameters

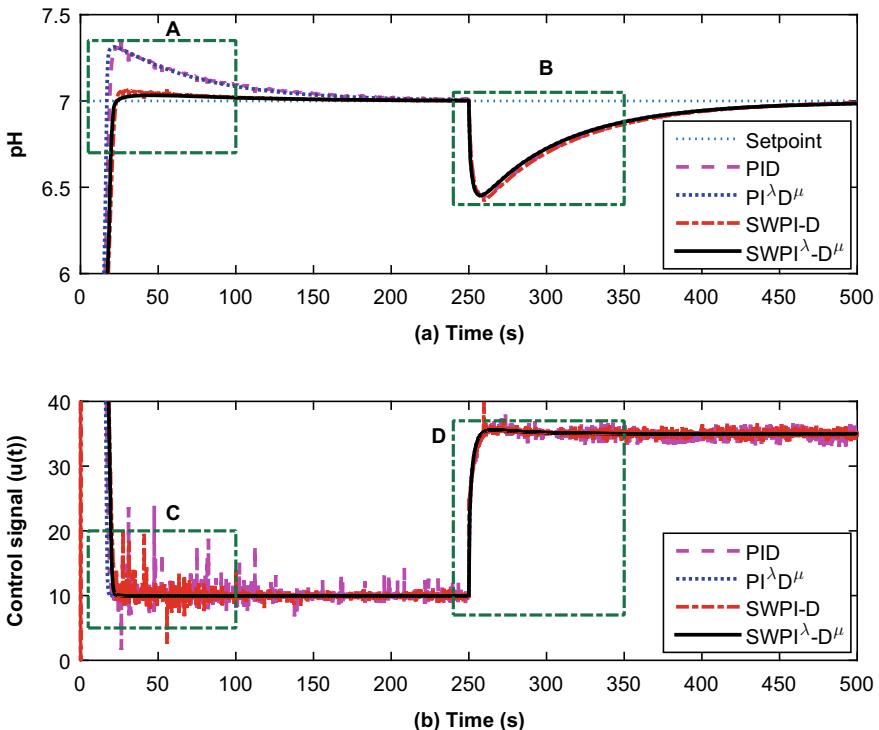
Category	Parameter	Value
Model parameters	V	50 L
	F_a	$10 \text{ m}^3/\text{h}$
	C_a	0.5 mol/L
	C_b	0.5 mol/L
	K_a	1.86×10^{-5}
	K_w	10^{-14}
Optimization parameters	D_1	$3(K_p, T_i, T_d)$
	D_2	$2(\lambda, \mu)$
	D_3	$1(b)$
	N	50
	<i>birdStep</i>	50
	α_o	0.5
	γ	0.7
	β	0.7

**Fig. 2.29** Titration curve variation under various concentrations

be seen that the $\text{SWPI}^\lambda\text{-D}^\mu$ settled faster compared to the other controllers at 20.7780 and 363.5095 s before and after disturbance respectively. However, $\text{PI}^\lambda\text{D}^\mu$ responds faster with a rise time of 15.8972 s as against 17.0750, 18.8494 and 19.5630 s of PID, $\text{SWPI}^\lambda\text{-D}^\mu$ and SWPI-D controllers. This is because the weighting factor b which is less than one for the two set-point weighted controllers SWPI-D and $\text{SWPI}^\lambda\text{-D}^\mu$.

Table 2.10 Tuned controller parameters for standard form of controllers

Controller	K_p	T_i	T_d	λ	μ	b
PID	39.7059	1.2836	0.0171	—	—	—
$PI^{\lambda}D^{\mu}$	39.7059	1.2836	0.0171	0.9767	0.5811	—
SWPI-D	39.7059	1.2836	0.0171	—	—	0.9516
SWPI $^{\lambda}$ -D $^{\mu}$	39.7059	1.2836	0.0171	0.9767	0.5811	0.9516

**Fig. 2.30** Performance comparison of standard $SWPI^{\lambda}D^{\mu}$ controller for set-point tracking and disturbance rejection

The faster settling times 363.5675 and 363.5095 s of the $PI^{\lambda}D^{\mu}$ and $SWPI^{\lambda}-D^{\mu}$ controllers after the disturbance compared to 370.9695 and 370.9194 s of PID and SWPI-D is an indication of the good load regulation ability of the fractional-order controllers. It is worth noting that, the disturbance rejection controller $C_y(s)$ for PID and SWPI-D is the same while that of the $PI^{\lambda}D^{\mu}$ and $SWPI^{\lambda}-D^{\mu}$ is the same. Hence, the closeness of the t_{s_2} of PID and SWPI-D in one hand and $PI^{\lambda}D^{\mu}$ and $SWPI^{\lambda}-D^{\mu}$ on the other hand.

To observe the derivative kick effect of the compared controllers, the plant is simulated to a variable set-point signal and the result is shown in Fig. 2.32. From

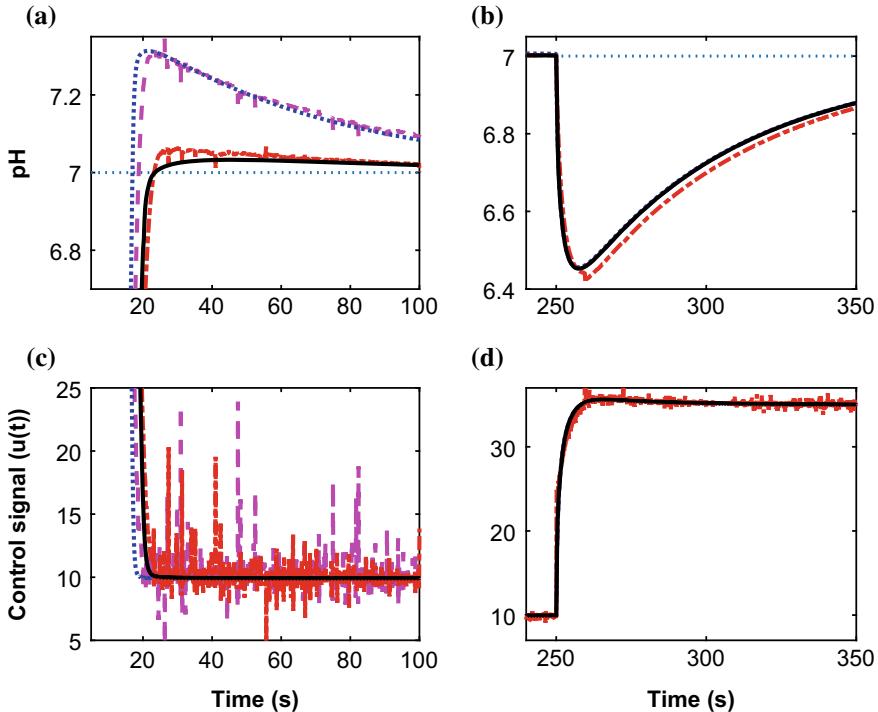


Fig. 2.31 Regions of interest A, B, C and D of Fig. 2.30

Table 2.11 Performance analysis of standard form of controllers

Controller	t_r	t_{s_1}	t_{s_2}	%OS	J_{ITAE}
PID	17.0750	110.1329	370.9695	4.9182	0.8131
$PI^{\lambda}D^{\mu}$	15.8972	101.4693	363.5675	4.4836	0.6930
SWPI-D	19.5630	22.1078	370.9194	0.9536	0.2903
SWPI $^{\lambda}$ -D $^{\mu}$	18.8494	20.7780	363.5095	0.4802	0.2624

the responses, it can be seen that during the set-point change, the performance of SWPI $^{\lambda}$ -D $^{\mu}$ is better when compared to the other controllers. This is clearly seen in the zoomed plot of regions of interest A and B in Fig. 2.33.

Additionally, during the set-point change, the derivative kick effect observed in the control signals of both PID and PI $^{\lambda}D^{\mu}$ is reduced in SWPI-D and SWPI $^{\lambda}$ -D $^{\mu}$. This is visible from the zoomed plot of regions C and D also in Fig. 2.33. As seen in the zoomed figure, the control signals of PID and PI $^{\lambda}D^{\mu}$ are both very narrow, sharp and reached a peak of 100% and around 40% respectively. On the other hand, the peak of the control signals of SWPI-D and SWPI $^{\lambda}$ -D $^{\mu}$ is both around 25%.

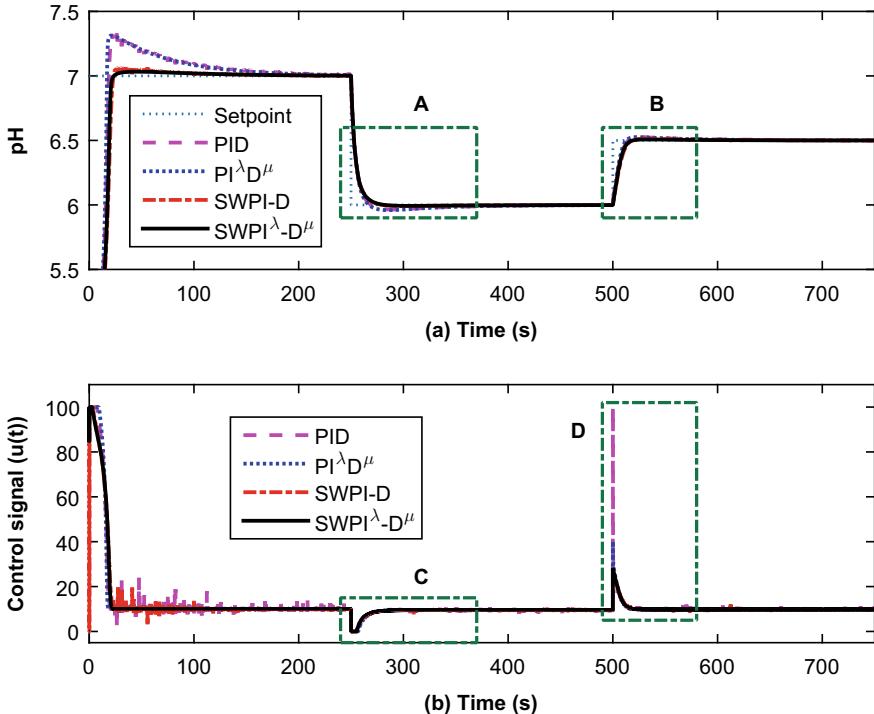


Fig. 2.32 Performance comparison of standard $\text{SWPI}^\lambda\text{D}^\mu$ controller for variable set-point tracking

2.4.4.3 Industrial $\text{SWPI}^\lambda\text{D}^\mu$ Controller

Following the similar fashion to the standard PIDs in Sect. 2.4.4.2, the tuned controller parameters of the industrial PIDs compared based on the tuning procedure given in Appendix A and the APSO algorithm with ITAE in Eq. (A.8) as error function are given in Table 2.12.

The comparison of closed-loop response of the system for set-point tracking and disturbance rejection with industrial configuration of PID in Eq. (2.22), $\text{PI}^\lambda\text{D}^\mu$ in Eq. (2.3), SWPID in Eq. (2.23) and $\text{SWPI}^\lambda\text{D}^\mu$ in Eq. (2.25) is shown in Fig. 2.34.

From the figure, just as observed in the case of standard PID forms, the set-point tracking ability of $\text{SWPI}^\lambda\text{D}^\mu$ is better when compared to those of PID, $\text{PI}^\lambda\text{D}^\mu$ and SWPI-D. The smoother control signal is also observed with $\text{SWPI}^\lambda\text{D}^\mu$ compared to the oscillatory response of the other controllers. This becomes more clear if regions of interests A, B, C and D of Fig. 2.35 are observed.

The numerical observation of Table 2.13 further confirms the advantages of the proposed approach over the compared controllers. From the table, $\text{SWPI}^\lambda\text{D}^\mu$ has less overshoot of 0.4395% compared to SWPI-D, $\text{PI}^\lambda\text{D}^\mu$ and PID with overshoots of 0.9732, 2.8758 and 2.9329% respectively.

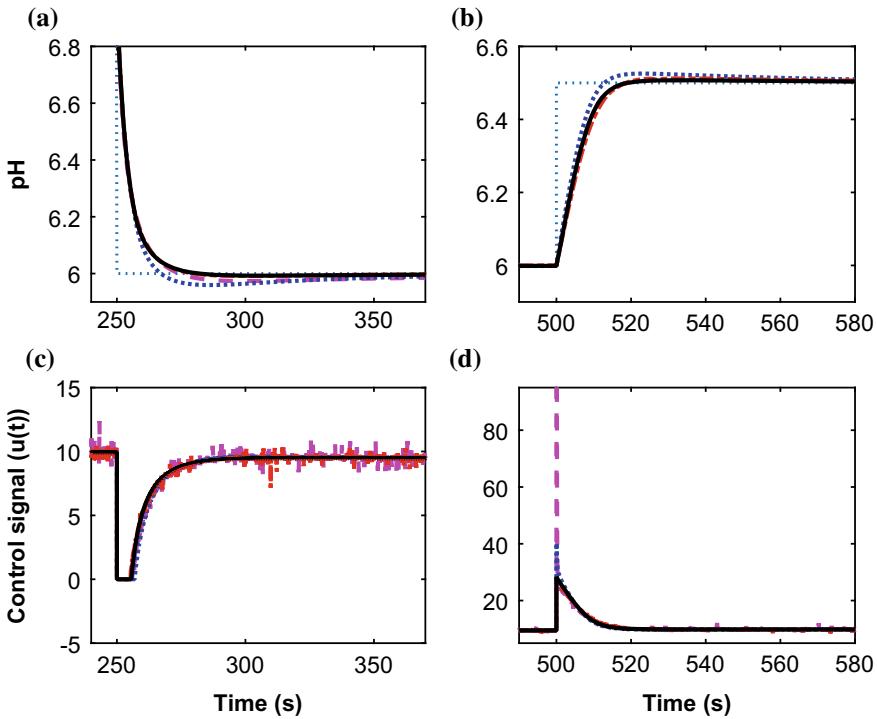


Fig. 2.33 Regions of interest A, B, C and D of Fig. 2.32

Table 2.12 Tuned controller parameters for industrial form of controllers

Controller	K_p	T_i	T_d	λ	μ	b
PID	35.9386	1.7955	0.0390	—	—	—
$PI^{\lambda}D^{\mu}$	35.9386	1.7955	0.0390	0.9164	0.9570	—
SWPI-D	35.9386	1.7955	0.0390	—	—	0.9835
SWPI $^{\lambda}$ -D $^{\mu}$	35.9386	1.7955	0.0390	0.9164	0.9570	0.9835

The SWPI $^{\lambda}$ -D $^{\mu}$ also settled faster than other controllers at 26.3520 and 395.4278 s before and after disturbance respectively. However, PI $^{\lambda}D^{\mu}$ responds faster with a rise time of 19.3370 s as against 20.8046, 22.4501 and 23.8224 s of PID, SWPI $^{\lambda}$ -D $^{\mu}$ and SWPI-D respectively. This is due to the reason explained earlier of the weighting factor b being less than unity for the set-point weighted controllers.

As observed in the case of standard configurations, faster settling times 397.1517 and 395.4278 s of the PI $^{\lambda}D^{\mu}$ and SWPI $^{\lambda}$ -D $^{\mu}$ controllers after the disturbance compared to 405.7697 and 404.9420 s of PID and SWPI-D indicates good disturbance rejection capability of the fractional-order configuration. The slightly different behavior of control signal of the PI $^{\lambda}D^{\mu}$ is observed here compared to the standard

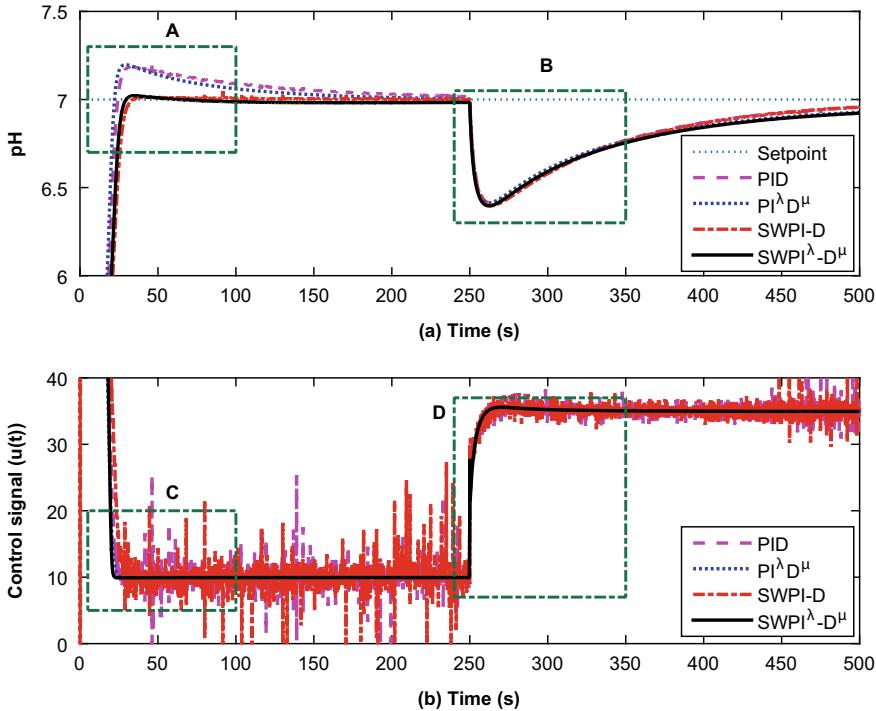


Fig. 2.34 Performance comparison of industrial $\text{SWPI}^\lambda\text{-D}^\mu$ controller for set-point tracking and disturbance rejection

form. The derivative kick effect of the compared controllers is shown in Fig. 2.36. From the responses, it can be seen that for the set-point change $\text{SWPI}^\lambda\text{-D}^\mu$ performed better than the other controllers. This is clearly seen in the zoomed plot of regions of interest A and B in Fig. 2.37.

The derivative kick effect observed in the control signals of both PID and $\text{PI}^\lambda\text{D}^\mu$ at set-point change is reduced in SWPI-D and $\text{SWPI}^\lambda\text{-D}^\mu$. This is further highlighted in the zoomed plot of regions of interest C and D in Fig. 2.37. From this figure, the control signals of PID and $\text{PI}^\lambda\text{D}^\mu$ are very narrow, sharp and both reached a peak of 100%. On the other hand, the control signals of both SWPI-D and $\text{SWPI}^\lambda\text{-D}^\mu$ peaked at around 25%.

2.4.5 Performance of $\text{PI}^\lambda\text{-PD}^\mu$ Controller

This subsection presents the performance evaluation of the proposed $\text{PI}^\lambda\text{-PD}^\mu$ controller developed in Sect. 2.3 on real-time pH neutralization process given in Sect. 2.4.2. First, the design of $\text{PI}^\lambda\text{-PD}^\mu$ controller parameters will be presented.

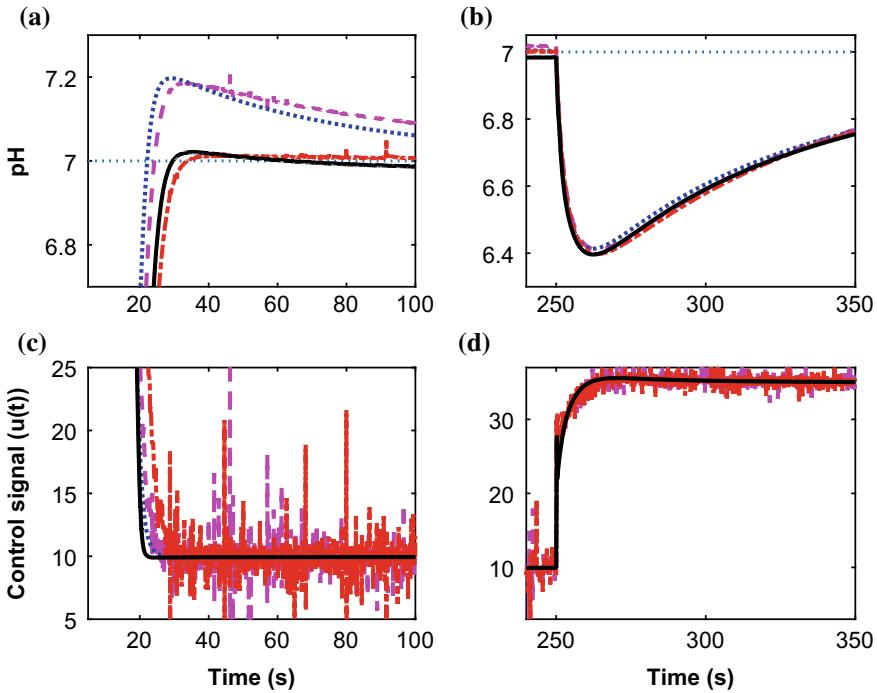


Fig. 2.35 Regions of interest A, B, C and D of Fig. 2.34

Table 2.13 Performance analysis of industrial form of controllers

Controller	t_r	t_{s1}	t_{s2}	%OS	J_{ITAE}
PID	20.8046	119.3300	405.7697	2.9329	0.9797
$PI^{\lambda}D^{\mu}$	19.3370	83.9811	397.1517	2.8758	0.9294
SWPI-D	23.8224	28.6699	404.9420	0.9732	0.9126
SWPI $^{\lambda}$ -D $^{\mu}$	22.4501	26.3520	395.4278	0.4395	0.8322

Then, the results of the proposed controllers will be presented in three phases. In the first phase, the performance of the integer-order PI-PD controller on the real-time pressure process will be given. Then in the second phase, the performance of the fractional-order PI-PD controller will be presented. Lastly, a comparison between the integer and fractional-order PI-PD controllers will be presented.

In all these cases, as mentioned earlier, the performance has been evaluated for set-point tracking, variation in set-point and disturbance rejection will be presented. Furthermore, the numerical analysis will be done in terms of rise time (t_r), settling time before and after disturbance (t_{s1} and t_{s2}) and percentage overshoot (%OS). Furthermore, it should be noted that a disturbance ($D(s)$) of 25% is injected at

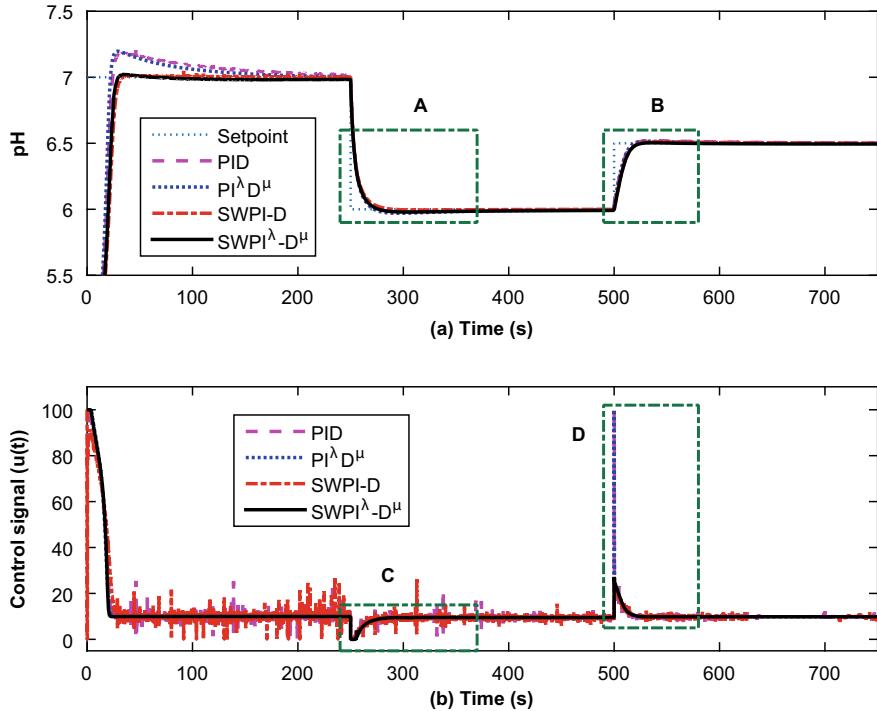


Fig. 2.36 Performance comparison of industrial $\text{SWPI}^\lambda \text{D}^\mu$ controller for variable set-point tracking

250 s for the purpose of disturbance rejection evaluation. Moreover, the ability of the controllers to track variation in set point will also be analyzed.

2.4.5.1 Controller Parameters

The tuned controller parameters of $\text{PI}^\lambda\text{-PD}^\mu$ given in Sect. 2.3 can be obtained from the designed $\text{SWPI}^\lambda \text{D}^\mu$ controller in Sect. 2.4.4 using the conversion relations given in Appendix A and Fig. A.1. Therefore, the parameters of PID, PI-PD and $\text{PI}^\lambda\text{-PD}^\mu$ controllers are given in Table 2.14.

2.4.5.2 PI-PD Controllers

The comparison of the response of the system with various integer-order controllers as given in Sect. 2.3.1 will be presented here. Thus, the performance of PID, PI-PD1 and PI-PD2 in Eqs. (2.4), (2.48) and (2.52) respectively for set-point tracking and disturbance rejection is given in Fig. 2.38. The zoomed-in view of regions of interest

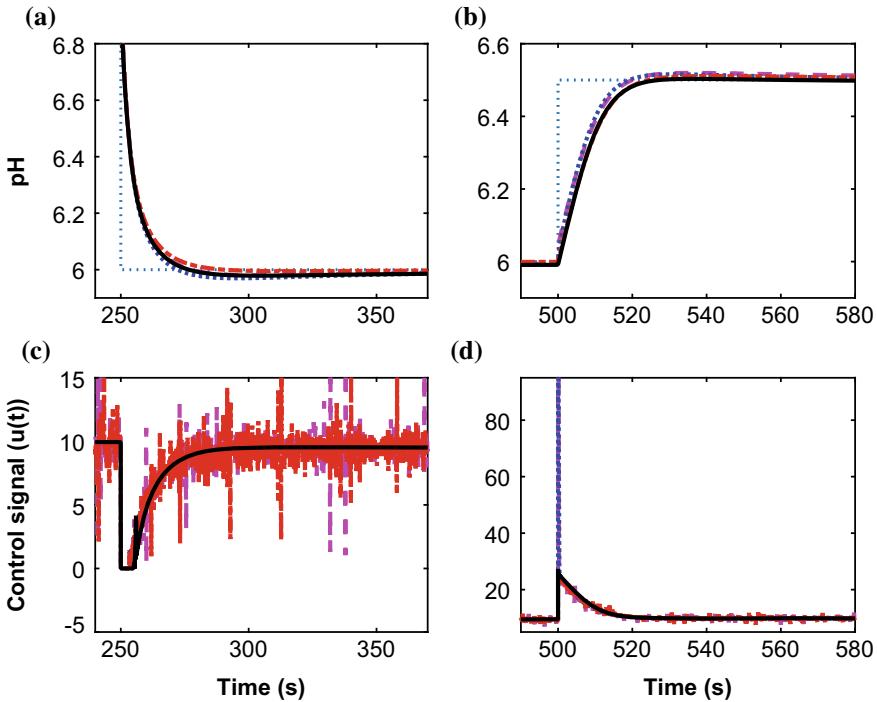


Fig. 2.37 Regions of interest A, B, C and D of Fig. 2.36

Table 2.14 Controller parameters of PI-PD controllers

Controller	P	I	D	λ	μ
PID	39.7059	30.9332	0.6790	—	—
PI-PD	39.1697	30.9332	0.0173	—	—
PI $^\lambda$ -PD $^\mu$	39.1697	30.9332	0.0173	0.9767	0.5811

A, B, C and D of the figure is highlighted in Fig. 2.39 while the numerical assessment of the figure is given in Table 2.15.

Observing the Figs. 2.38 and 2.39 and the Table 2.15, it can be seen that the set-point tracking ability of PI-PD2 controller is outperformed those of PID and PI-PD1. The PI-PD1 and PI-PD2 controllers have the fastest rise time and settling time of 16.8577 and 57.0507 s respectively. Furthermore, the controller produced the least overshoot of 2.2464%. However, for disturbance rejection, the performance of all the compared controllers are satisfactory with PID has the fastest settling time of 370.9301 s.

Observing the control signals of all the compared controllers, it can be seen that the PI-PD1 produced smoother action as compared to the oscillatory actions of both PID and PI-PD2. The smooth control action is due to the structure of the PI-PD.

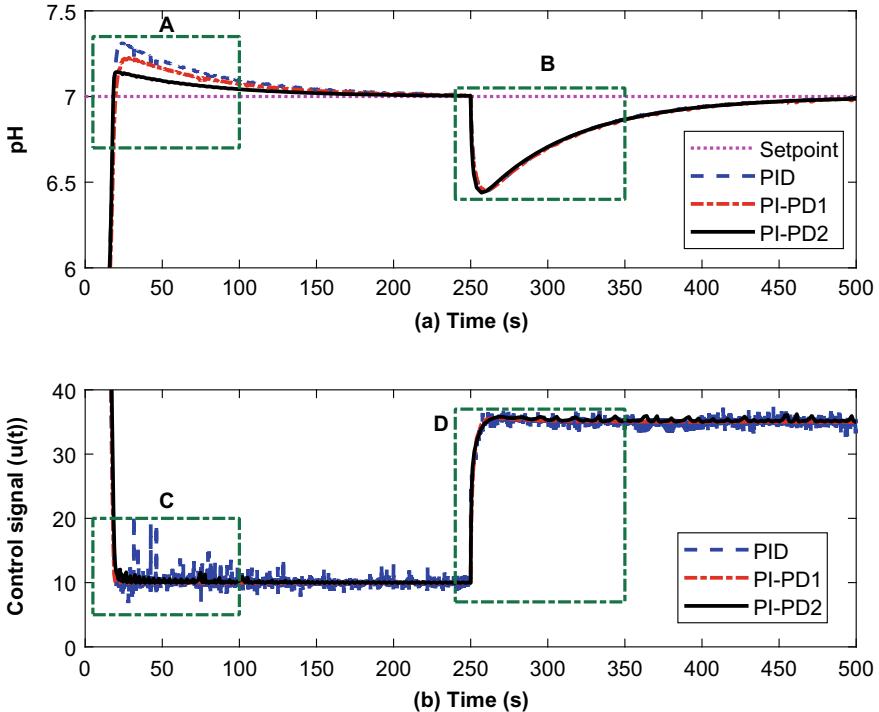


Fig. 2.38 Performance comparison of PI-PD controllers for set-point tracking and disturbance rejection

Similarly, the performance comparison of the PID, PI-PD1 and PI-PD2 controllers for variable set-point tracking is given in Fig. 2.40 while the regions of interest are further highlighted in Fig. 2.41. From the figures, it can be observed that, during step change, PID controller experiences a very narrow, sharp and peak of 100% while both the PI-PD controllers reached a peak of around 30%. Therefore, it can be concluded that the derivative kick effect of the PID controller is avoided with the use of PI-PD controllers.

2.4.5.3 $\text{PI}^\lambda\text{-PD}^\mu$ Controllers

In a similar way to the performance of integer-order controllers presented in Sect. 2.4.5.2, the comparison of the response of the system with fractional-order controllers as given in Sect. 2.3.2 will be analyzed. Therefore, the performance of $\text{PI}^\lambda\text{D}^\mu$, $\text{PI}^\lambda\text{-PD}^\mu$ 1 and $\text{PI}^\lambda\text{-PD}^\mu$ 2 given in Eqs. (2.1), (2.58) and (2.62) respectively for set-point tracking and disturbance rejection is given in Fig. 2.42. The zoomed-in view of regions of interest A, B, C and D of the figure is highlighted in Fig. 2.43 while the numerical observations are given in Table 2.16.

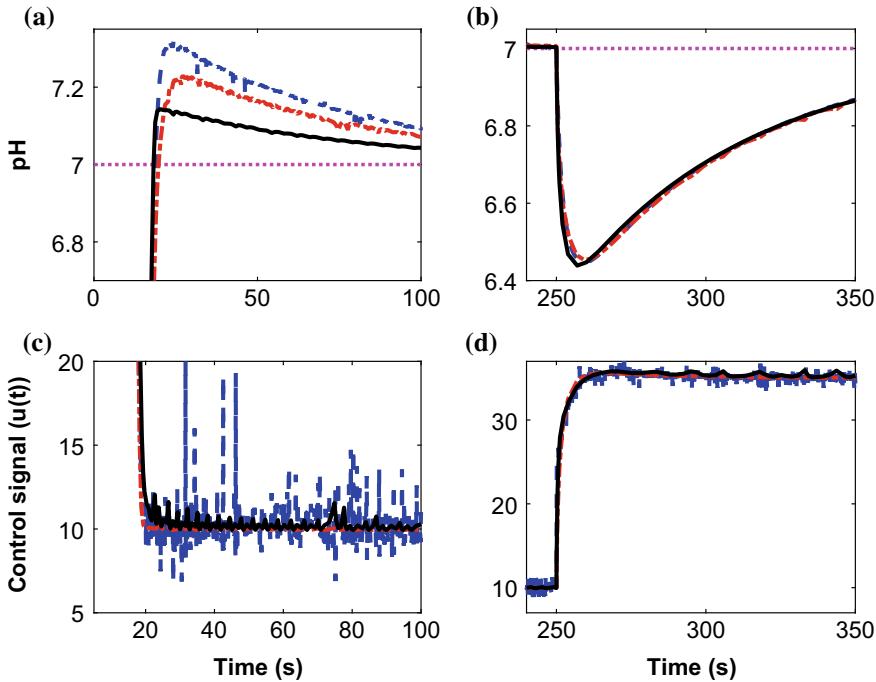


Fig. 2.39 Regions of interest A, B, C and D of Fig. 2.38

Table 2.15 Performance analysis of integer-order controllers

Controller	t_r	t_{s_1}	t_{s_2}	%OS
PID	17.0742	111.5825	370.9301	4.5570
PI-PD1	17.5341	95.5005	372.1216	3.3269
PI-PD2	16.8577	57.0507	371.8033	2.2464

From the figures and the table, it can be seen that unlike in the case of integer-order where PI-PD2 controller outperformed the other controllers, the performance of the two fractional PI-PD controllers ($\text{PI}^\lambda\text{-PD}^\mu 1$ and $\text{PI}^\lambda\text{-PD}^\mu 2$) outperformed that of $\text{PI}^\lambda\text{D}^\mu$. Observing the table, the two $\text{PI}^\lambda\text{-PD}^\mu$ controllers have each produced %OS of around 2.2% while that of the $\text{PI}^\lambda\text{D}^\mu$ is 4.4836%.

Furthermore, the $\text{PI}^\lambda\text{-PD}^\mu 2$ controller settled faster at 52.1932 s when compared to 71.4198 and 101.4690 s of $\text{PI}^\lambda\text{-PD}^\mu 1$ and $\text{PI}^\lambda\text{D}^\mu$ controllers respectively. However, in terms of rise time, the $\text{PI}^\lambda\text{D}^\mu$ controller has the fastest rise time of 15.8972 s. Furthermore, the controller also settled faster after a disturbance with a settling time of 363.2637 s.

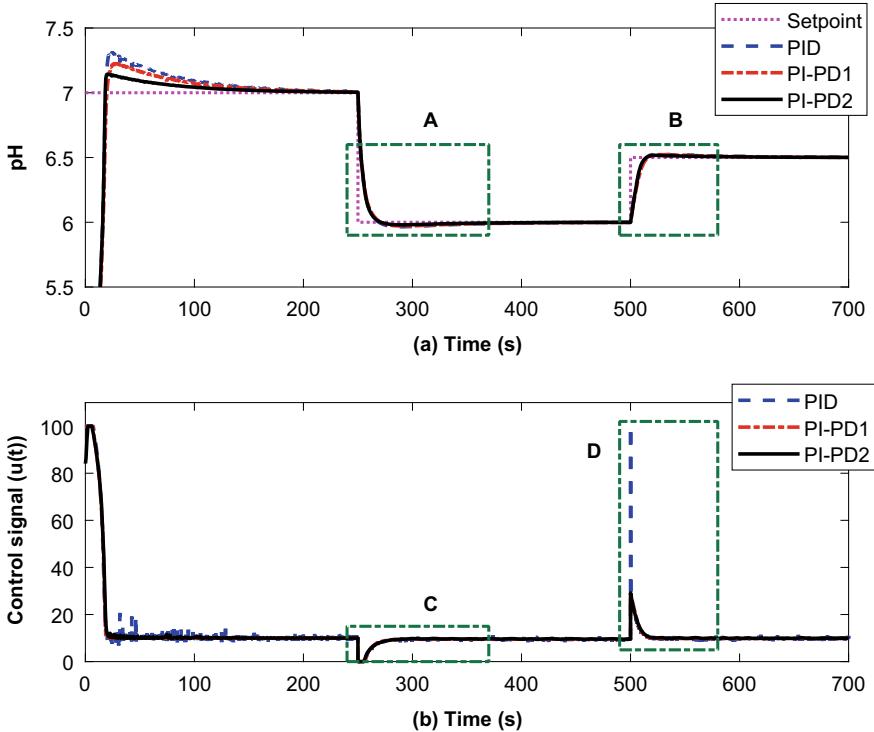


Fig. 2.40 Performance comparison of PI-PD controllers for variable set-point tracking

Similarly, the performance comparison of the various controller for variable set-point tracking is given in Fig. 2.44 while the regions of interest are further highlighted in Fig. 2.45. From the figure, it can be observed that, during step change, $\text{PI}^\lambda \text{D}^\mu$ controller experiences a sharp peak of 40% while the $\text{PI}^\lambda\text{-PD}^\mu$ controllers reached a peak of around 30% as in the PI-PD controllers.

2.4.5.4 Comparison of PI-PD and $\text{PI}^\lambda\text{-PD}^\mu$ Controllers

The comparison of the response of the system with various integer-order controllers (PID, PI-PD1 and PI-PD2) as given in Sect. 2.3.1 and fractional-order controllers ($\text{PI}^\lambda \text{D}^\mu$, $\text{PI}^\lambda\text{-PD}^\mu 1$ and $\text{PI}^\lambda\text{-PD}^\mu 2$) as given in Sect. 2.3.2 for set-point tracking and disturbance rejection is given in Fig. 2.46. The regions of interest A, B, C and D of the figure are further highlighted in Fig. 2.47. The numerical assessment of the figure in terms of step response characteristics is given in Table 2.17.

Observing the figures, it can be seen that the proposed fractional-order PI-PD controllers performed better compared to all other integer-order controllers. From the numerical analysis, it can be seen that both the proposed $\text{PI}^\lambda\text{-PD}^\mu$ controllers have

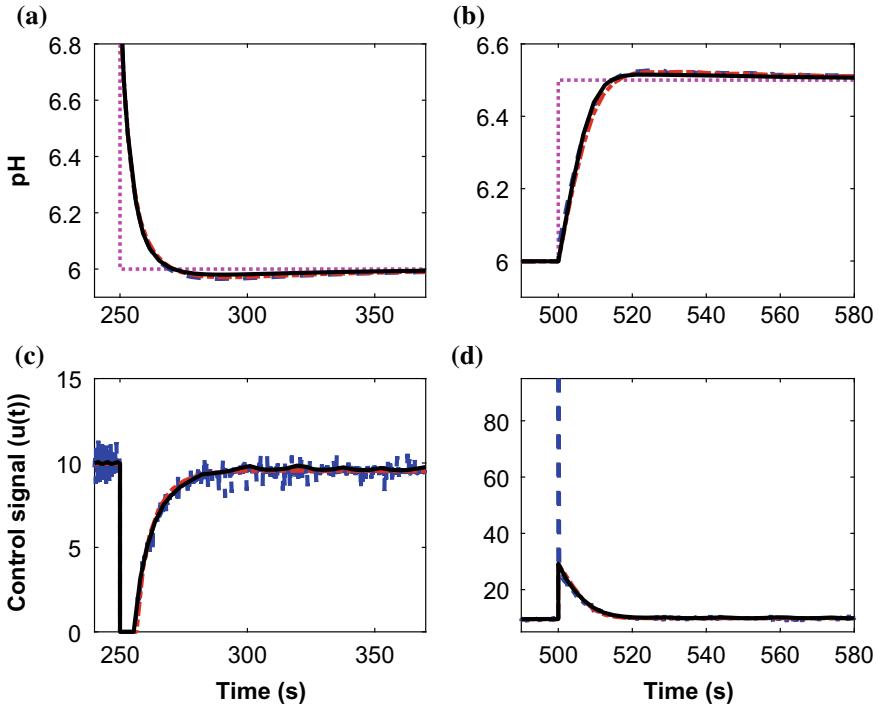


Fig. 2.41 Regions of interest A, B, C and D of Fig. 2.40

produced the least overshoots of around 2.2%. Furthermore, the $\text{PI}^\lambda\text{-PD}^\mu$ controller settled faster at 52.1932 s. However, in terms of rise time, the $\text{PI}^\lambda\text{D}^\mu$ controller has the fastest t_r of 15.8972 s.

Furthermore, the comparison of the response for disturbance rejection shows that all the compared controllers are satisfactory with $\text{PI}^\lambda\text{D}^\mu$ controller has the fastest settling time of 363.2637 s. Observing the control signals, it can be seen that the proposed controllers have a smoother control action compared to the oscillatory actions of integer-order controllers.

Similarly, the comparison of the response of the system for variable set-point tracking is given in Fig. 2.48 while the regions of interest A, B, C and D of the figure are further highlighted in Fig. 2.49. From the figures, it can be seen that during set-point change both the integer-order and fractional-order PI-PD controllers are free from derivative kick effects. On the other hand, the PID and $\text{PI}^\lambda\text{D}^\mu$ controllers experience very sharp narrow peaks of 100% and 40% respectively.

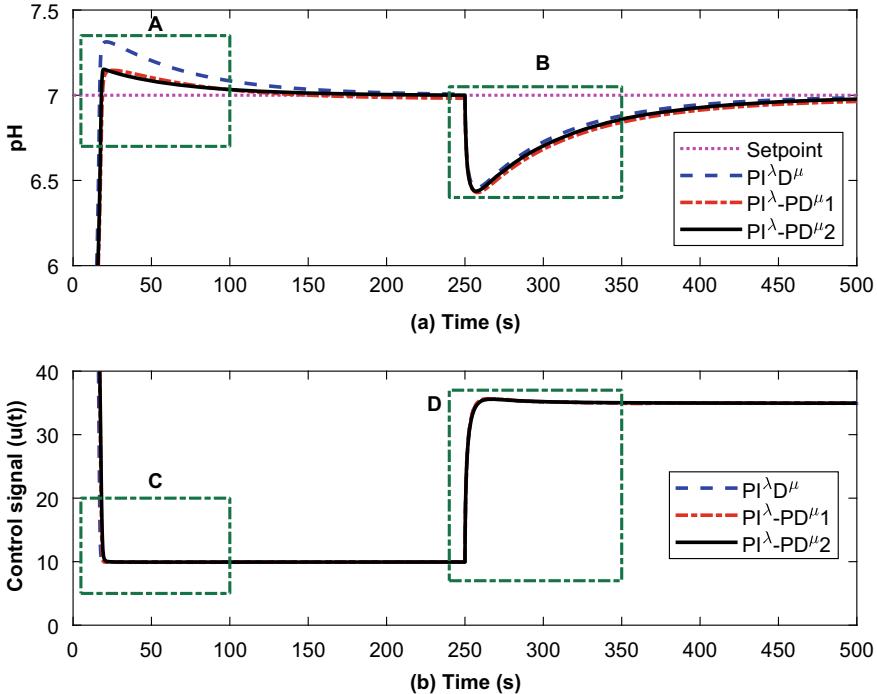


Fig. 2.42 Performance comparison of $\text{PI}^\lambda\text{-PD}^\mu$ controllers for set-point tracking and disturbance rejection

2.4.6 Comparison of SW $\text{PI}^\lambda\text{D}^\mu$ and $\text{PI}^\lambda\text{-PD}^\mu$ Controllers

This section presents the comparison of both the proposed SW $\text{PI}^\lambda\text{D}^\mu$ and $\text{PI}^\lambda\text{-PD}^\mu$ controllers as given in Sects. 2.2.3 and 2.3 on the pH neutralization process plant given in Sect. 2.4.2 is presented. First, the performance of all compared controllers will be evaluated for set-point tracking and disturbance rejection. Then, the performance will be evaluated for all the controllers to track variation in set point.

2.4.6.1 Set-Point Tracking and Disturbance Rejection

The controller parameters of various compared controllers is shown in Table 2.18. Thus, the set-point tracking and disturbance rejection performance of the system with various controllers including PID, SWPI-D, PI-PD, $\text{PI}^\lambda\text{D}^\lambda$, SW $\text{PI}^\lambda\text{D}^\mu$ and $\text{PI}^\lambda\text{-PD}^\mu$ is shown in Fig. 2.50. The regions of interest A, B, C and D of the figure are further highlighted in Fig. 2.51. Observing the figures, it can be seen that the proposed fractional-order controllers (SW $\text{PI}^\lambda\text{D}^\mu$, $\text{PI}^\lambda\text{-PD}^\mu$ 1 and $\text{PI}^\lambda\text{-PD}^\mu$ 2) performed better compared to all other controllers.

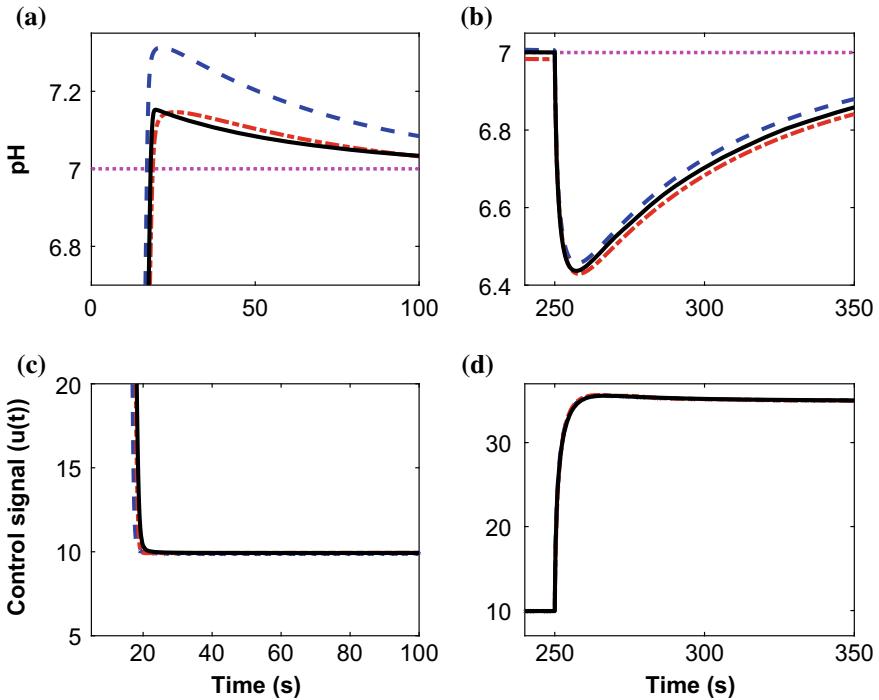


Fig. 2.43 Regions of interest A, B, C and D of Fig. 2.42

Table 2.16 Performance analysis of fractional-order controllers

Controller	t_r	t_{s1}	t_{s2}	%OS
$\text{PI}^\lambda \text{D}^\mu$	15.8972	101.4690	363.2637	4.4836
$\text{PI}^\lambda\text{-PD}^\mu$ 1	17.1534	71.4198	373.4480	2.2243
$\text{PI}^\lambda\text{-PD}^\mu$ 2	16.8406	52.1932	372.2233	2.2045

The numerical assessment of the figure for step-response characteristics is given in Table 2.19. From the table, it can be noted that the proposed controllers $\text{SWPI}^\lambda\text{-D}^\mu$, $\text{PI}^\lambda\text{-PD}^\mu$ 1 and $\text{PI}^\lambda\text{-PD}^\mu$ 2 has the least overshoots of 0.4805, 2.2243 and 2.2045% respectively. Furthermore, it can also be observed that the controller $\text{SWPI}^\lambda\text{-D}^\mu$ and $\text{PI}^\lambda\text{-PD}^\mu$ 2 also has the fastest settling times of 20.7780 and 52.1932 s respectively. However, in terms of rise time, the fractional-order controllers $\text{PI}^\lambda\text{D}^\mu$ and $\text{PI}^\lambda\text{-PD}^\mu$ 2 has the fastest rise time of 15.8972 and 16.8406 s respectively.

On the other hand, the disturbance rejection performance of all the compared controllers is satisfactory. The shorter settling times 363.5675 and 363.5095 s of the $\text{PI}^\lambda\text{D}^\mu$ and $\text{SWPI}^\lambda\text{-D}^\mu$ controllers after the disturbance compared to 370.9695 and 370.9194 s of PID and SWPI-D is an indication of the good load regulation ability of the fractional-order controllers.

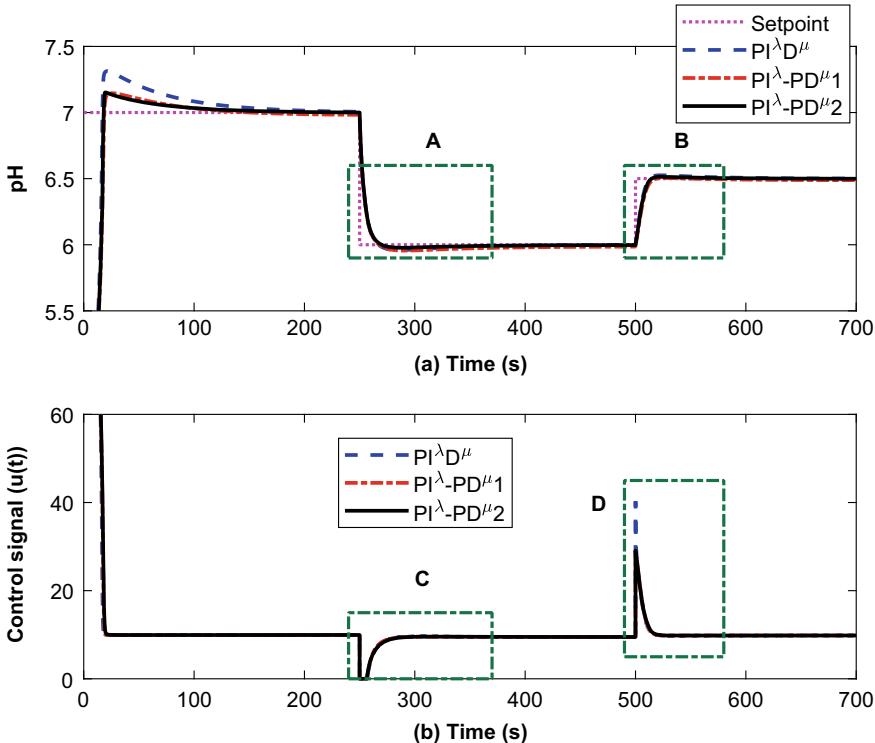


Fig. 2.44 Performance comparison of $\text{PI}^\lambda\text{-PD}^{\mu}$ controllers for variable set-point tracking

It is worth noting that the disturbance rejection controller for PID and SWPI-D is the same while that of the $\text{PI}^\lambda \text{D}^\mu$ and SWPI $^\lambda\text{-D}^\mu$ is the same. Hence, the closeness of t_{s_2} of around 370 s for PID and SWPI-D on one hand and $\text{PI}^\lambda \text{D}^\mu$ and SWPI $^\lambda\text{-D}^\mu$ of around 363 s on the other hand. Furthermore, from the zoomed figures C and D, it can be seen that the control actions of the proposed fractional-order controllers SWPI $^\lambda\text{-D}^\mu$, $\text{PI}^\lambda\text{-PD}^{\mu 1}$ and $\text{PI}^\lambda\text{-PD}^{\mu 2}$ is smooth compared to the oscillatory signals of PID, PI-PD1, PI-PD2, and SWPI-D.

2.4.6.2 Variable Set-Point Tracking

To observe the derivative kick effect of all the compared controllers including PID, SWPI-D, PI-PD, $\text{PI}^\lambda \text{D}^\lambda$, SWPI $^\lambda\text{-D}^\mu$ and $\text{PI}^\lambda\text{-PD}^{\mu}$, the plant is simulated to a variable set-point signal and the result is shown in Fig. 2.52. The zoomed-in view for regions of interest A, B, C and D of the figure is given in Fig. 2.53. From the responses, it can be seen that during the set-point change, the performance of proposed fractional-order controllers is better when compared to the other controllers. This is clearly seen in the zoomed plot A and B in Fig. 2.53.

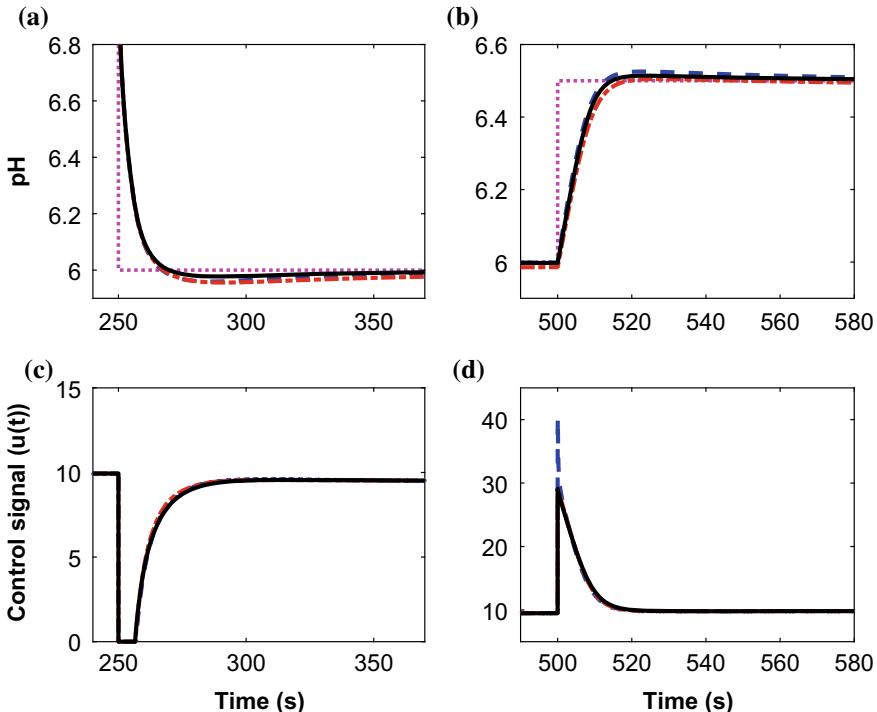


Fig. 2.45 Regions of interest A, B, C and D of Fig. 2.44

Additionally, during the set-point change, the derivative kick effect observed in the control signals of both PID and $\text{PI}^\lambda\text{D}^\mu$ controllers is reduced in SWPI-D, PI-PD, $\text{SWPI}^\lambda\text{-D}^\mu$ and $\text{PI}^\lambda\text{-PD}^\mu$ controllers. This is visible from the zoomed plot of regions C and D in Fig. 2.53. As seen in the zoomed figure, the control signals of PID and $\text{PI}^\lambda\text{D}^\mu$ are both very narrow and sharp and reached a peak of 100% and around 40%, respectively. On the other hand, the peak of the control signals of SWPI-D, PI-PD and $\text{SWPI}^\lambda\text{-D}^\mu$ is about 25%.

2.5 Simulation Study on Pressure Process Plant

In the first part of this section, a description of the pressure process plant is given with the help of P&ID while the second part gives the schematics of the entire experimental setup. Then, the subsequent three subsections present the performance evaluation of the designed $\text{SWPI}^\lambda\text{D}^\mu$ and $\text{PI}^\lambda\text{-PD}^\mu$ controllers on the pressure process plant.

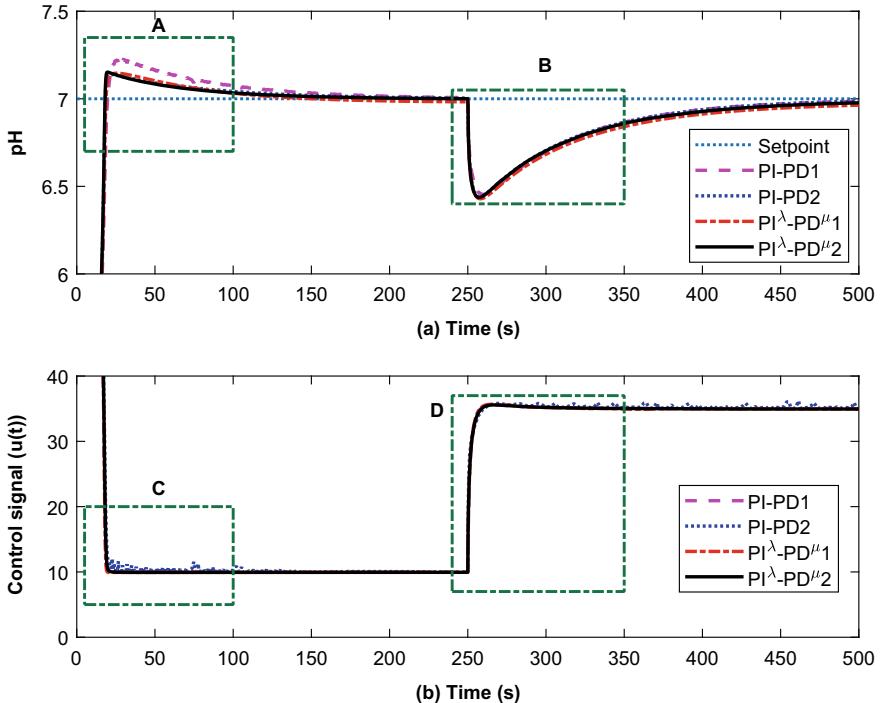


Fig. 2.46 Performance comparison of PI-PD and $\text{PI}^\lambda\text{-PD}^\mu$ controllers for set-point tracking and disturbance rejection

2.5.1 PI&D of Pressure Process Plant

The real-time pressure process plant used in the experiment is shown in Fig. 2.54 while the P&ID of process representing the flow is shown in Fig. 2.55 [109]. The plant consists of a buffer tank VL 202 which is connected to a centralized air compressor through a hand valve HV 202 and a process control valve PCV 202. The characteristics of the electro-pneumatic control valve PCV 202 is shown in Fig. 2.56. From the characteristics, it can be observed that the type of behavior is a nonlinear increasing sensitivity type and the maximum input air pressure is around 6 bar. Thus, this will guide in the choice of the set point.

Furthermore, a pressure transmitter PT 202 is used to measure the pressure in the tank. The pressure readings from PT 202 are fed into pressure indicating controller PIC 202 in terms of voltage in the range of 0–5 V. The control signal is used to control the opening of PCV 202 to adjust the pressure in accordance with the set-point. On the other side of the tank, the downstream process valve is set at a fixed opening to allow a constant flow from the pressure tank during the experiment.

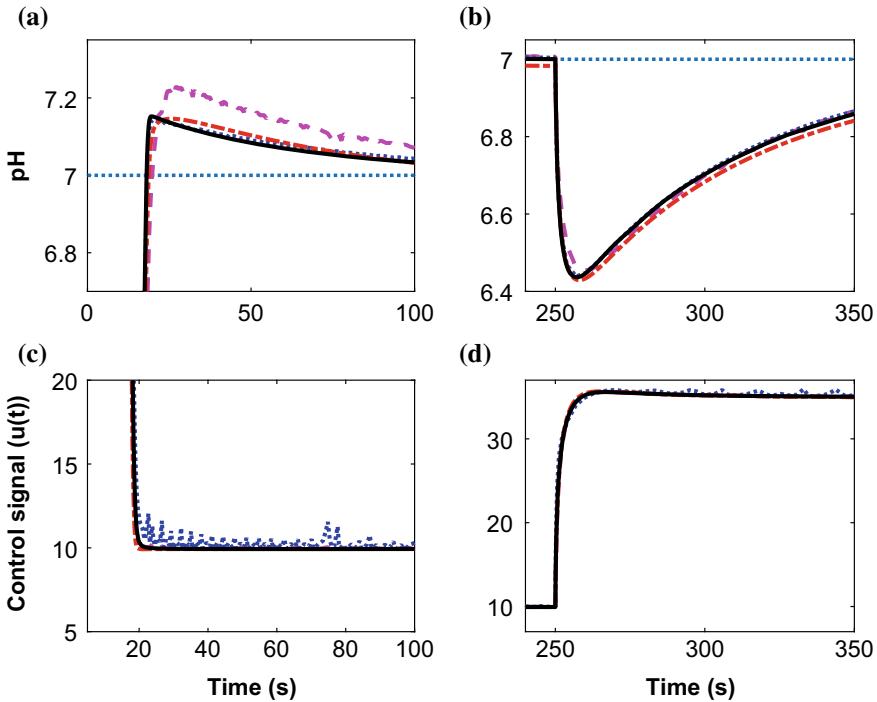


Fig. 2.47 Regions of interest A, B, C and D of Fig. 2.46

Table 2.17 Performance analysis of PI-PD and $\text{PI}^\lambda\text{-PD}^\mu$ controllers

Category	Controller	t_r	t_{s1}	t_{s2}	%OS
Integer-order controllers	PID	17.0742	111.5825	370.9301	4.5570
	PI-PD1	17.5341	95.5005	372.1216	3.3269
	PI-PD2	16.8577	57.0507	371.8033	2.2464
Fractional-order controllers	$\text{PI}^\lambda\text{D}^\mu$	15.8972	101.4690	363.2637	4.4836
	$\text{PI}^\lambda\text{-PD}^\mu$ 1	17.1534	71.4198	373.4480	2.2243
	$\text{PI}^\lambda\text{-PD}^\mu$ 2	16.8406	52.1932	372.2233	2.2045

2.5.2 Schematics of Experimental Setup

The schematic diagram of the complete experimental setup is shown in Fig. 2.57. The proposed control strategy will be implemented in the host PC using MATLAB/Simulink software which is accessed through ‘Remote Desktop Connection’ in the control room. As in the case of the pH neutralization process plant, here also the host PC is interfaced with the pilot plant through Peripheral Component Interconnect (PCI) cards 1713U, 1720U and 1751.

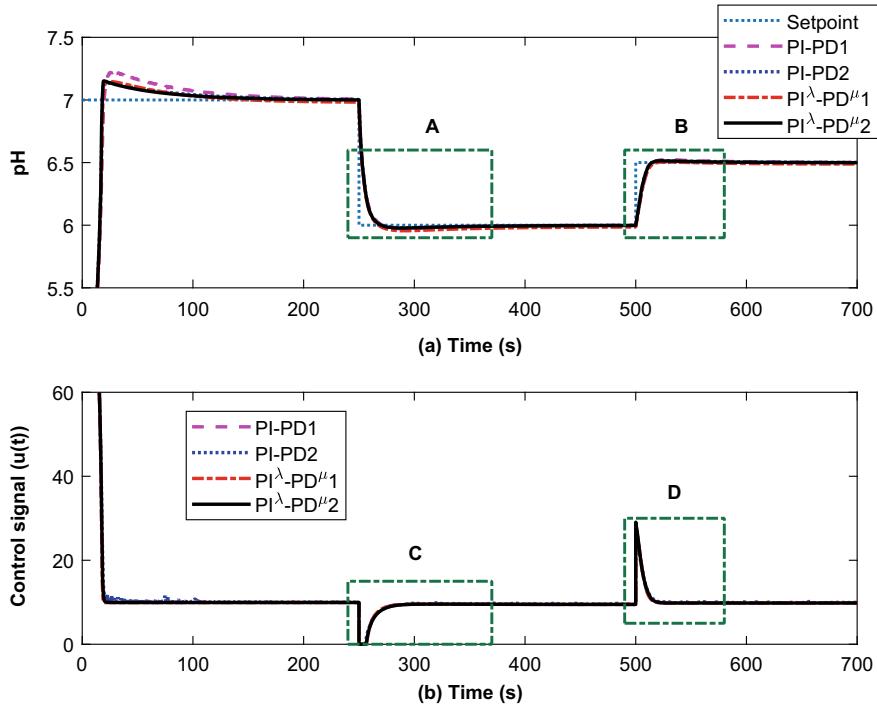


Fig. 2.48 Performance comparison of PI-PD and $\text{PI}^\lambda\text{-PD}^\mu$ controllers for variable set-point tracking

The PCI-1713U is a 100 kS/s, 12-bit, 32-ch isolated AI card used to access signal from PT 202. On the other hand, the PCI-1720U; a 12-bit, 4-channel isolated AO card, is used to send the control signal to PCV 202. The last card PCI-1751 is a 48-channel digital I/O and 3-channel counter card used to operate the process plant either Remotely or Locally (R/L).

2.5.3 Performance of SW $\text{PI}^\lambda\text{D}^\mu$ Controller

This subsection presents the performance evaluation of the proposed SW $\text{PI}^\lambda\text{D}^\mu$ controller presented in Sect. 2.2.3 on pressure process plant in Sect. 2.5. First, the design of SW $\text{PI}^\lambda\text{D}^\mu$ controller parameters will be presented. Then, the results of proposed parallel and series configuration of the designed SW $\text{PI}^\lambda\text{D}^\mu$ controller in comparison with PID, $\text{PI}^\lambda\text{D}^\mu$ and SW PI-D will be presented.

In all these cases, the performance has been evaluated for set-point tracking, variation in set-point and disturbance rejection will be presented. Furthermore, the numerical analysis will be done in terms of rise time (t_r), settling time before and

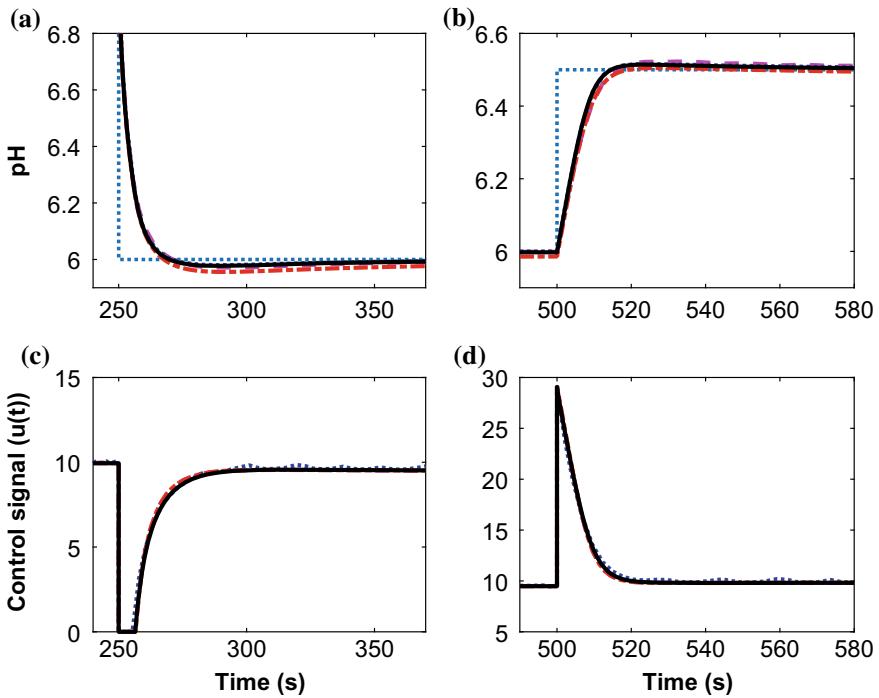


Fig. 2.49 Regions of interest A, B, C and D of Fig. 2.48

Table 2.18 Controller parameters of proposed controllers on pH neutralization process

Category	Controller	K_p	K_i	K_d	b	λ	μ
Integer-order controllers	PID	39.7059	30.9332	0.6790	–	–	–
	PI-PD1	39.1697	30.9332	0.0173	–	–	–
	PI-PD2	39.1697	30.9332	0.0173	–	–	–
	SWPI-D	39.7059	30.9332	0.6790	0.9516	–	–
Fractional-order controllers	$PI^{\lambda}D^{\mu}$	39.7059	30.9332	0.6790	–	0.9767	0.5811
	$PI^{\lambda}-PD^{\mu}1$	39.1697	30.9332	0.0173	–	0.9767	0.5811
	$PI^{\lambda}-PD^{\mu}2$	39.1697	30.9332	0.0173	–	0.9767	0.5811
	$SWPI^{\lambda}-D^{\mu}$	39.7059	30.9332	0.6790	0.9516	0.9767	0.5811

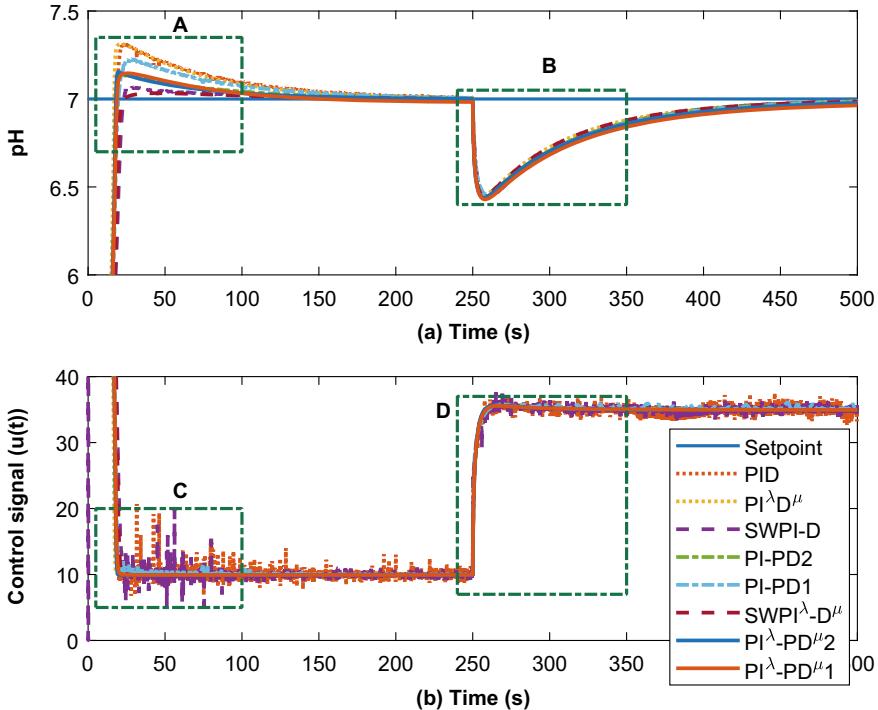


Fig. 2.50 Performance comparison of $\text{SWPI}^{\lambda}\text{D}^{\mu}$ and $\text{PI}^{\lambda}\text{-PD}^{\mu}$ controllers for set-point tracking and disturbance rejection

after disturbance (t_{s_1} and t_{s_2}) and percentage overshoot (%OS). Furthermore, it should be noted that a disturbance ($D(s)$) of 25% is injected at 500 s for the purpose of disturbance rejection evaluation. Moreover, the ability of the controllers to track variation in set point will also be analyzed.

2.5.3.1 Controller Parameters

The conventional PID controller parameters (K_p , K_i , and K_d) are tuned using the Ziegler-Nichols (ZN) tuning approach. Therefore, the sustained uniform oscillations of PT202 for an ultimate gain of $K_u = 51$ is shown in Fig. 2.58. From the response, the ultimate period (P_u) is calculated as 20 s.

In order to select the best values of set-point weighting and fractional-order parameters (b , λ and μ), several values of these parameters within the range 0 to 1 are used. Therefore, the effect on variation of b on the plant performance given in Fig. 2.59 shows that as the value of b decreases the %OS decreases but increases in t_r . This effect is shown numerically in Table 2.20. Furthermore, it can be seen from the con-

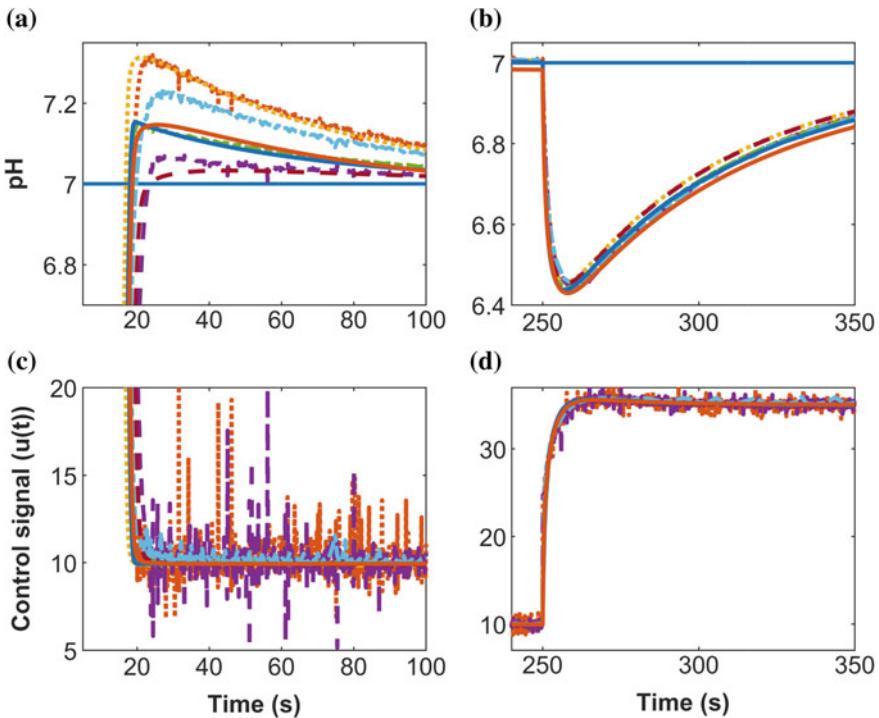


Fig. 2.51 Regions of interest A, B, C and D of Fig. 2.50

Table 2.19 Performance analysis of proposed controllers on pH neutralization process

Category	Controller	t_r	t_{s_1}	t_{s_2}	%OS
Integer-order controllers	PID	17.0750	110.1329	370.9695	4.9182
	PI-PD1	17.5341	95.5005	372.1216	3.3269
	PI-PD2	16.8577	57.0507	371.8033	2.2464
	SWPI-D	19.5630	22.1078	370.9194	0.9536
Fractional-order controllers	$\text{PI}^\lambda \text{D}^\mu$	15.8972	101.4693	363.5675	4.4836
	$\text{PI}^\lambda \text{-PD}^\mu 1$	17.1534	71.4198	373.4480	2.2243
	$\text{PI}^\lambda \text{-PD}^\mu 2$	16.8406	52.1932	372.2233	2.2045
	$\text{SWPI}^\lambda \text{-D}^\mu$	18.8494	20.7780	363.5095	0.4805

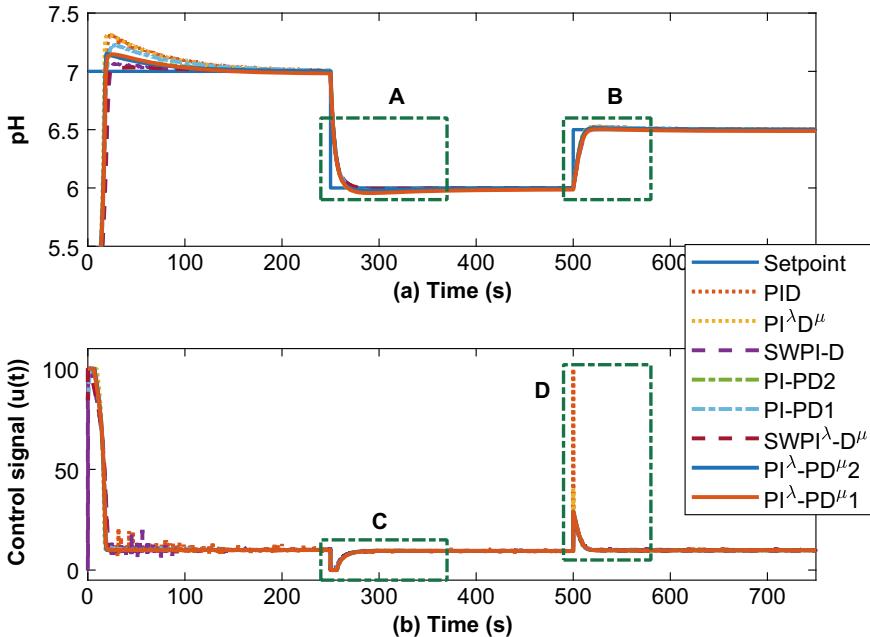


Fig. 2.52 Performance comparison of $\text{SWPI}^\lambda \text{D}^\mu$ and $\text{PI}^\lambda \text{-PD}^\mu$ controllers for variable set-point tracking

trol signal that as b decreases the proportional kick effect decreases. Hence, the best value of b is selected as 0.8.

Observing the effect on the variation of λ on the plant performance as given in Fig. 2.60 and Table 2.21, it can be seen that the maximum value of λ for the steady-state is 1.0 while the minimum value is 0.8. Values lower than the minimum leads to high steady-state error. Similarly, from Fig. 2.61, it can be clearly seen that as the values of μ drifts away from unity, the performance of the system in terms of t_r , t_s and %OS improves. This also leads to reduced oscillation in the control signal. Hence, the best values for λ and μ are chosen as 0.95 and 0.5 respectively.

2.5.3.2 Parallel $\text{SWPI}^\lambda \text{D}^\mu$ Controller

The parameters of compared parallel configurations of PID, $\text{PI}^\lambda \text{D}^\mu$, SWPID and $\text{SWPI}^\lambda \text{D}^\mu$ controllers obtained from the previous Sect. 2.5.3.1 are given in Table 2.22.

The comparison of the responses of the system with these PID, $\text{PI}^\lambda \text{D}^\mu$, SWPID and $\text{SWPI}^\lambda \text{D}^\mu$ controllers for set-point tracking ability, disturbance rejection capability, and smoother control action are given in Fig. 2.62. For more clarity regions A, B, C, and D of the figure are highlighted in Fig. 2.63. Furthermore, the numerical analysis of the controller performance is also given in Table 2.23.

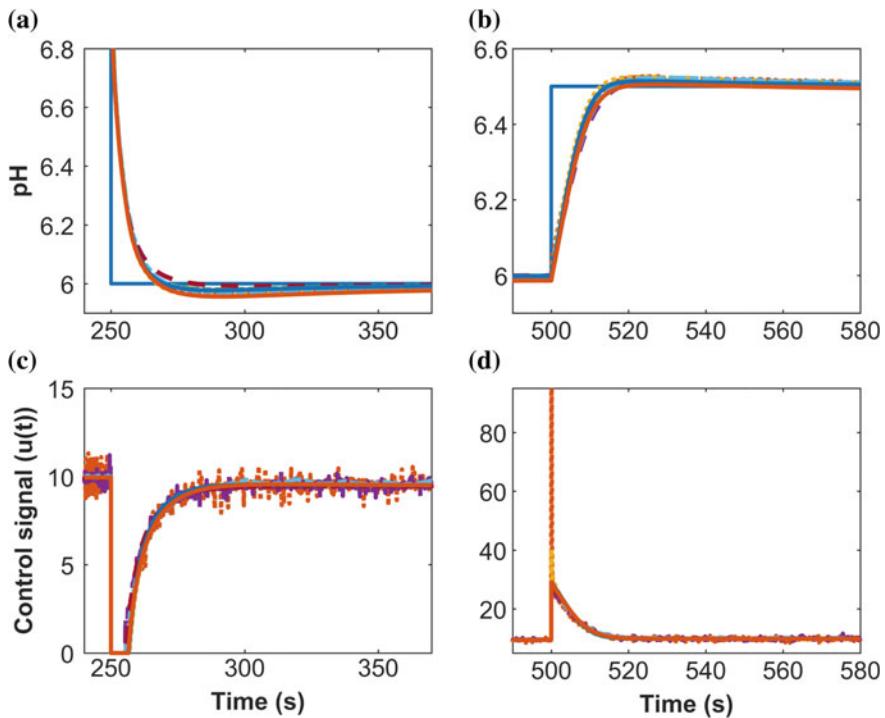


Fig. 2.53 Regions of interest A, B, C and D of Fig. 2.52

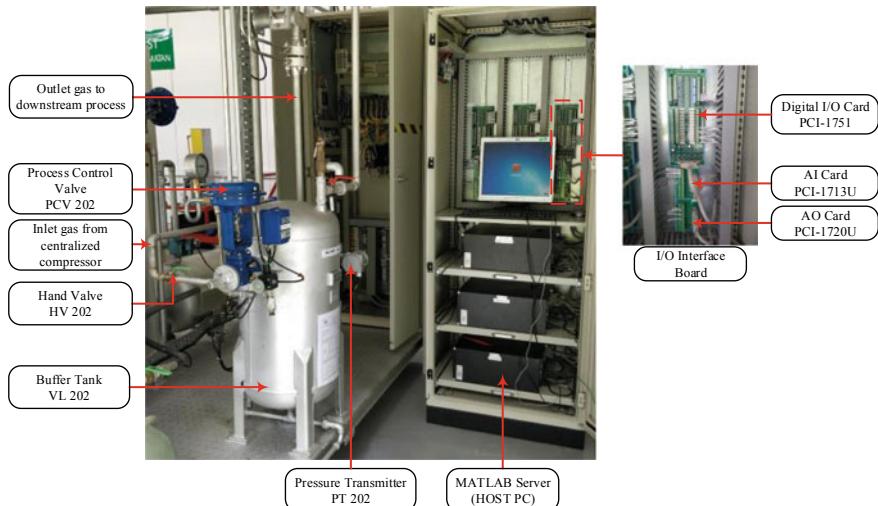
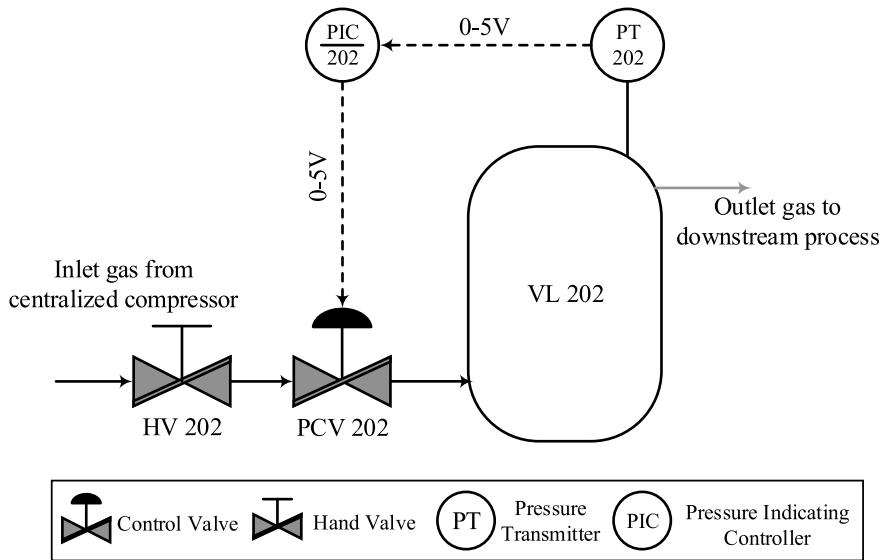
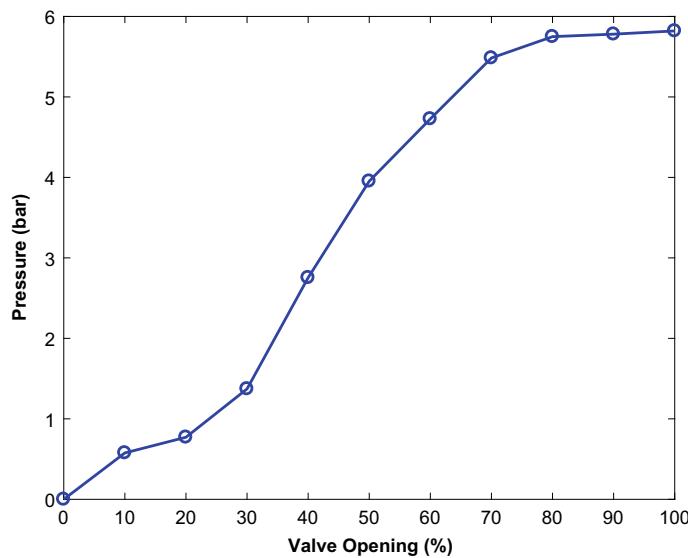


Fig. 2.54 Experimental setup of real-time pressure process plant

**Fig. 2.55** P&ID of pressure process plant**Fig. 2.56** Characteristics of electro-pneumatic control valve PCV 202

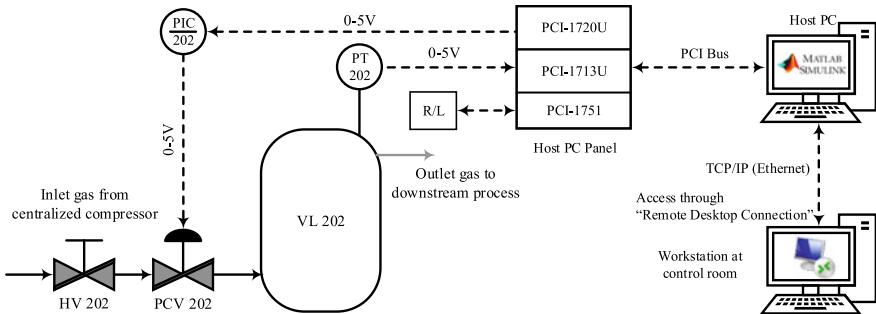


Fig. 2.57 Schematic diagram of the complete experimental setup

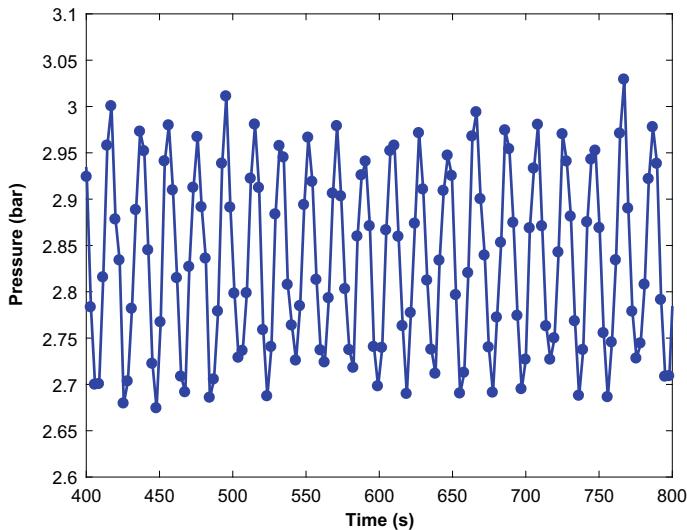


Fig. 2.58 Sustained uniformed oscillation of PT202 for ZN closed-loop tuning

From both figures and table, it can be seen that the $\text{SWPI}^{\lambda}\text{D}^{\mu}$ controller produces the least %OS of 3.6283% as compared to 8.2935, 5.1853, 3.9360% of the PID, SWPID, and $\text{PI}^{\lambda}\text{D}^{\mu}$ respectively. From the settling time (t_{s_1}) of the controllers, it can be seen that at 83.2167 s the proposed approach settles faster than the other controllers. However, in terms of rise-time (t_r), the proposed controller came third with a value of 7.3698 s. This indicated that there is a trade-off between good overshoot and settling time in one hand and speed of response (rise time) on the other.

The disturbance rejection of all controllers PID, $\text{PI}^{\lambda}\text{D}^{\mu}$, SWPID, and $\text{SWPI}^{\lambda}\text{D}^{\mu}$ controllers were found to be satisfactory with the SWPID having the fastest recovery with t_{s_2} of 537.9967 s and $\text{PI}^{\lambda}\text{D}^{\mu}$ having the slowest recovery of 582.2311 s. Observing the control signals of all controllers compared, it can be seen that the proposed $\text{SWPI}^{\lambda}\text{D}^{\mu}$ generates smoother signal compared to the oscillatory signals of others.

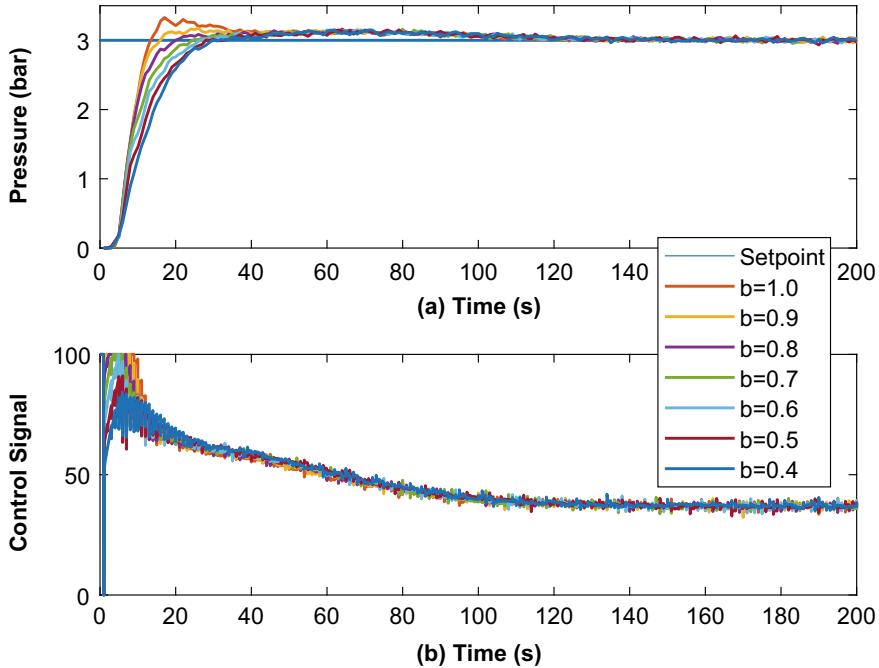


Fig. 2.59 Effect of parameter b on the plant performance

Table 2.20 Numerical assessment on effect of variation of b

Parameter	Value	t_r	t_s	%OS
b	1.0	6.4044	95.8871	10.9402
	0.8	8.8206	104.2408	5.4508
	0.6	12.9751	108.0727	5.0218
	0.4	16.2399	104.5702	5.0060

To evaluate the ability of various controllers for variable set-point tracking, the plant with various controllers is made to track variable reference signal as shown in Fig. 2.64. For more clarity, regions A, B, C and D of the figure are further highlighted in Fig. 2.65.

From both figures, it can be observed during set-point change (i.e., at 500 s), the two fractional-order controllers (i.e., $\text{PI}^\lambda \text{D}^\mu$ and $\text{SWPI}^\lambda \text{D}^\mu$) produced smaller overshoot as against those of two integer-order controllers (i.e., PID and SWPID). Observing the control signals of the controllers, it can be seen that $\text{PI}^\lambda \text{D}^\mu$ and PID have higher derivative kick effect making the signal to reach 80% of the valve opening. However, this effect is significantly reduced in the SWPID and $\text{SWPI}^\lambda \text{D}^\mu$.

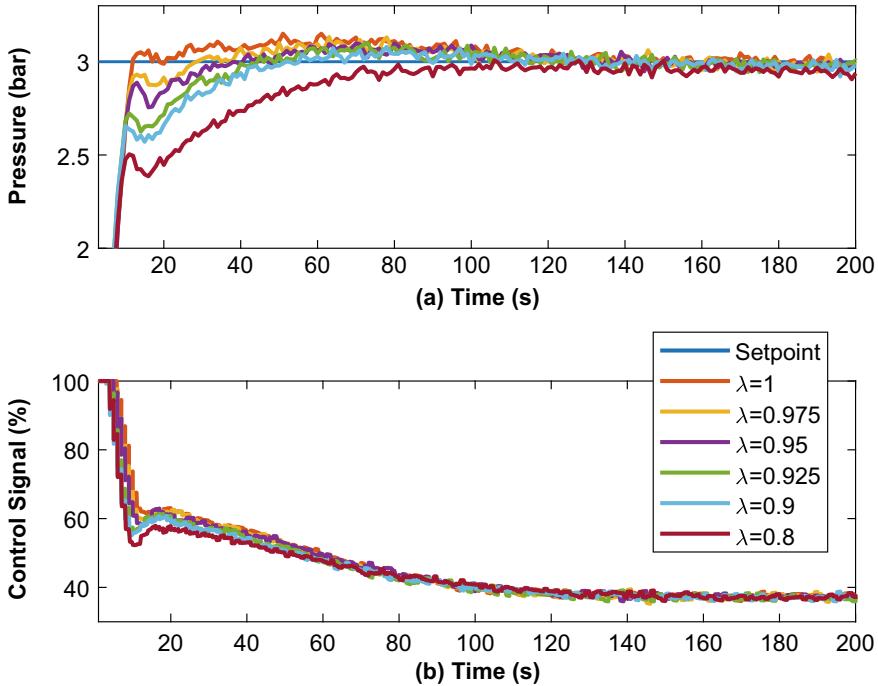


Fig. 2.60 Effect of parameter λ on the plant performance

Table 2.21 Numerical assessment on effect of variation of fractional-order controller parameters λ and μ

Parameter	Value	t_r	t_s	%OS
λ	1.0	6.4911	95.3112	5.0151
	0.975	6.6421	77.9867	4.4215
	0.95	6.6913	79.5911	3.2994
	0.925	6.9599	72.2900	3.4077
	0.9	17.8056	70.0560	2.6667
	0.8	32.7617	198.1350	0.2777
μ	0.9	7.0383	88.4318	5.4255
	0.7	7.1283	99.3004	5.0350
	0.5	6.9372	93.6639	4.6373
	0.3	6.7716	85.7019	4.7881

2.5.3.3 Series SWPI $^\lambda$ D $^\mu$ Controller

The controller parameters of compared series configuration of PID in Eq. (2.22), PI $^\lambda$ D $^\mu$ in Eq. (2.3), SWPID in Eq. (2.24) and SWPI $^\lambda$ D $^\mu$ in Eq. (2.25) controllers

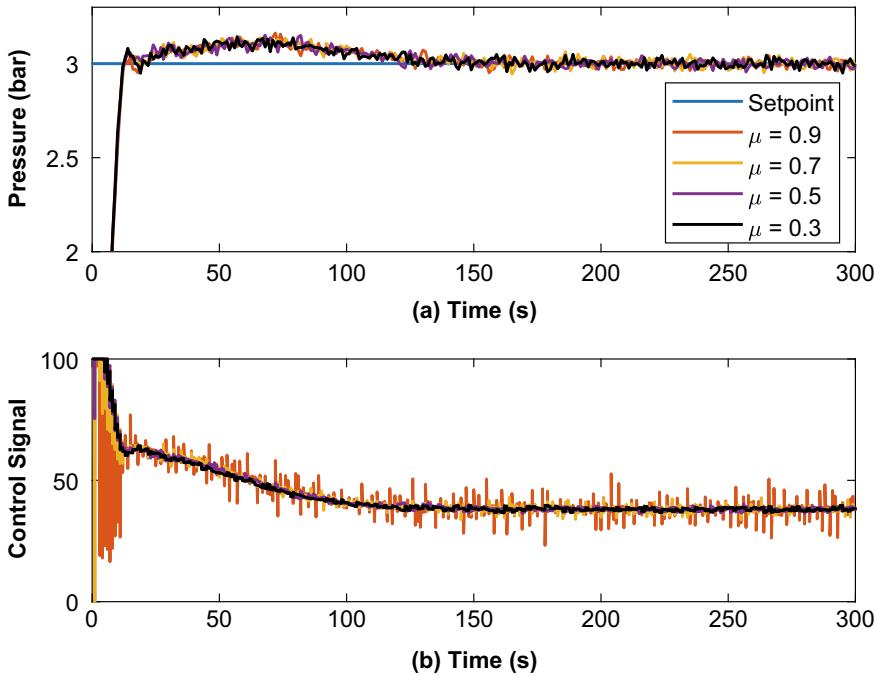


Fig. 2.61 Effect of parameter μ on the plant performance

Table 2.22 Controller parameters for parallel form of controllers

Controller	K_p	K_i	K_d	α	b	λ	μ
PID	30.0	3.0	1.0	0.025	—	—	—
$\text{PI}^\lambda \text{D}^\mu$	30.0	3.0	1.0	0.025	—	0.95	0.50
SWPI-D	30.0	3.0	1.0	0.025	0.80	—	—
SWPI $^\lambda$ -D $^\mu$	30.0	3.0	1.0	0.025	0.80	0.95	0.50

will be obtained using the parallel configuration of $\text{SWPI}^\lambda \text{D}^\mu$ previous Sect. 2.5.3.2. Therefore, based on the tuning procedure reported in Appendix A and Fig. A.1, the parameters of the series PID's are given in Table 2.24.

The responses of the system with the compared controllers for set-point tracking ability and disturbance rejection capability are given in Fig. 2.66. The regions of interest A, B, C, and D are further highlighted in Fig. 2.67 while the numerical results are given in Table 2.25.

From both figures and table, it can be seen that the $\text{SWPI}^\lambda \text{D}^\mu$ controller produces the least %OS of 2.9943% as compared to 12.5007, 9.1338, 4.1832% of the PID, SWPID and $\text{PI}^\lambda \text{D}^\mu$ respectively. Observing the settling times t_{s_1} and t_{s_2} of all the compared controllers, it can be seen that at 33.4194 s the proposed approach settled

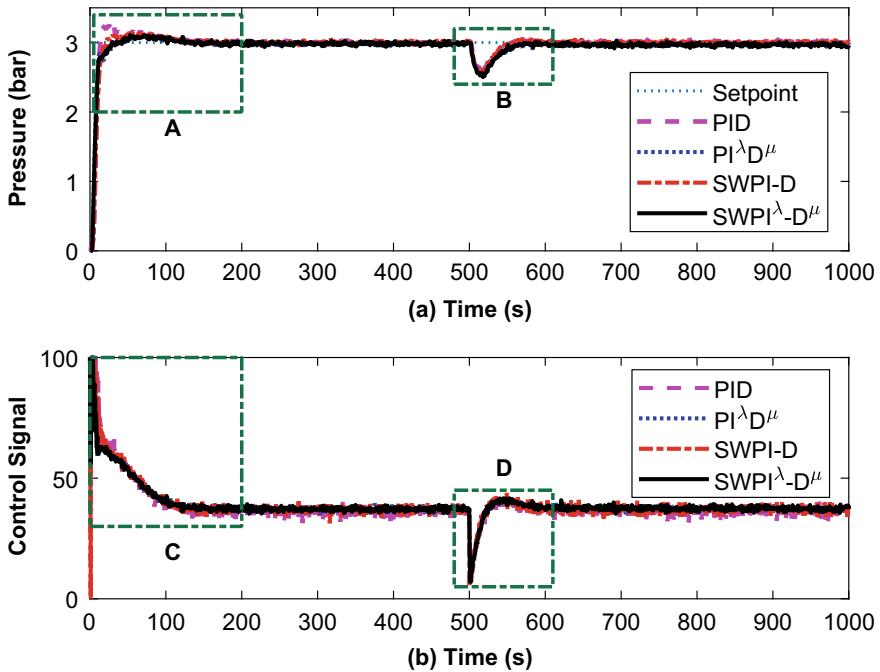


Fig. 2.62 Performance comparison of parallel $\text{SWPI}^{\lambda}\text{D}^{\mu}$ controller for set-point tracking and disturbance rejection

much faster than the others. However, in terms of t_r , the proposed $\text{SWPI}^{\lambda}\text{D}^{\mu}$ controller came fourth with a value of 17.2004 s. The results here followed the similar pattern to the parallel configuration of $\text{SWPI}^{\lambda}\text{D}^{\mu}$ controller in Sect. 2.5.3.2.

A noticeable improvement in the performance of the series configuration of $\text{SWPI}^{\lambda}\text{D}^{\mu}$ controller compared to the parallel is that of the faster settling time of 33.4194 s and less overshoot of 2.9943% as against of 83.2167 s and 3.6283% of the parallel configuration of $\text{SWPI}^{\lambda}\text{D}^{\mu}$ controller.

The SWPID has the fastest recovery from disturbance effect settling at 541.8869 s while $\text{PI}^{\lambda}\text{D}^{\mu}$ has the slowest recovery settling at 841.0198 s. However, the disturbance rejection ability of all controllers was found to be satisfactory. Observing the control signals of all controllers, it can be seen that the proposed $\text{SWPI}^{\lambda}\text{D}^{\mu}$ generates smoother signal compared to the oscillatory signals of others.

The performance comparison of various controllers PID, $\text{PI}^{\lambda}\text{D}^{\mu}$, SWPID and $\text{SWPI}^{\lambda}\text{D}^{\mu}$ for variable set-point tracking has been evaluated in the same as to the previous section. The plant with various controllers is made to track the variable reference signal as shown in Fig. 2.68. The regions A, B, C and D of the figure are highlighted in Fig. 2.69.

From the two figures, it can be observed that, during the set-point change, the two fractional-order controllers (i.e., $\text{PI}^{\lambda}\text{D}^{\mu}$ and $\text{SWPI}^{\lambda}\text{D}^{\mu}$) produced smaller overshoot

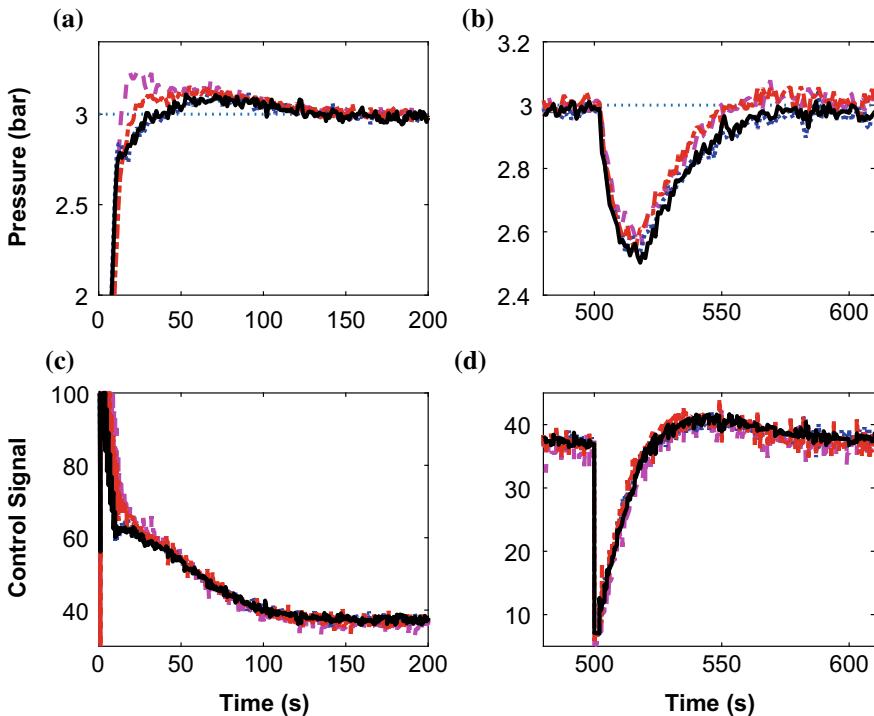


Fig. 2.63 Regions of interest A, B, C and D of Fig. 2.62

Table 2.23 Performance analysis for parallel form of controllers

Controller	t_r	t_{s1}	t_{s2}	%OS
PID	6.7511	98.1610	544.3894	8.2935
$\text{PI}^\lambda \text{D}^\mu$	6.9613	90.5565	582.2311	3.9360
SWPID	8.4787	94.5839	537.9967	5.1853
$\text{SWPI}^\lambda \text{-D}^\mu$	7.3698	83.2167	553.4759	3.6283

as against those of PID and SWPID as in the case of the parallel configuration. Observing the control signals of the controllers, it can be seen that the PID has a higher derivative kick effect reaching 100% of the valve opening. On the other hand, this effect is significantly reduced in the $\text{PI}^\lambda \text{D}^\mu$, SWPID and $\text{SWPI}^\lambda \text{D}^\mu$.

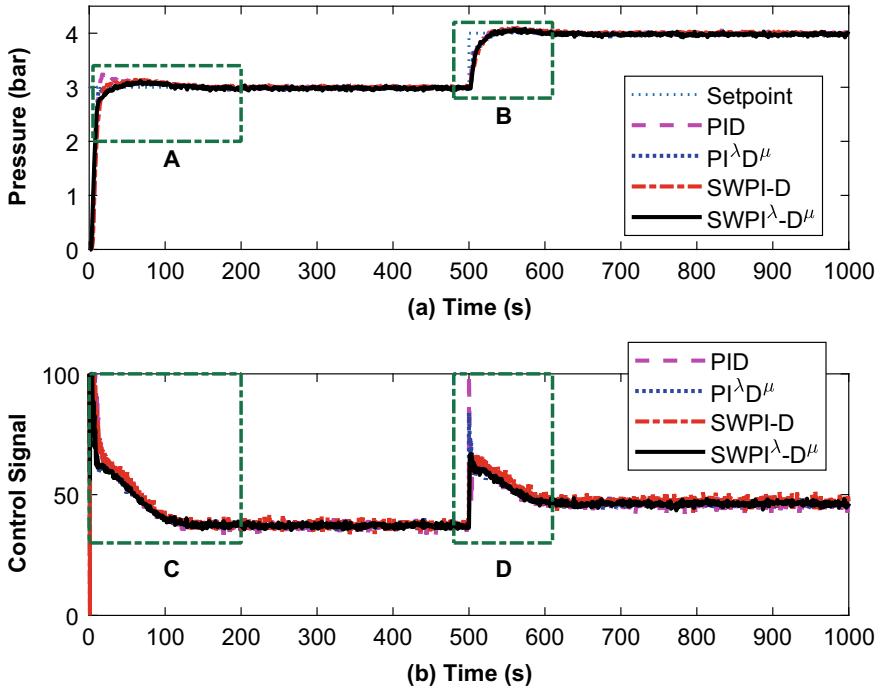


Fig. 2.64 Performance comparison of parallel $\text{SWPI}^{\lambda}\text{-D}^{\mu}$ controller for variable set-point tracking

2.5.4 Performance of $\text{PI}^{\lambda}\text{-PD}^{\mu}$ Controller

In this subsection, first, the design of $\text{PI}^{\lambda}\text{-PD}^{\mu}$ controller parameters based will be presented on the tuning procedure given in Appendix A. Then, the results of the proposed controllers will be presented in three phases. In the first phase, the performance of the integer-order PI-PD controller given in Sect. 2.3.1 on the real-time pressure process will be given. Then in the second phase, the performance of the fractional-order PI-PD controller given in Sect. 2.3.2 will be presented. Lastly, a comparison between the integer and fractional-order PI-PD controllers will be presented.

In all these cases, as mentioned earlier, the performance has been evaluated for set-point tracking, variation in set-point and disturbance rejection will be presented. Furthermore, the numerical analysis will be done in terms of rise time (t_r), settling time before and after disturbance (t_{s_1} and t_{s_2}) and percentage overshoot (%OS). Furthermore, it should be noted that a disturbance ($D(s)$) of 25% is injected at 500s for the purpose of disturbance rejection evaluation. Moreover, the ability of the controllers to track variation in set point will also be analyzed.

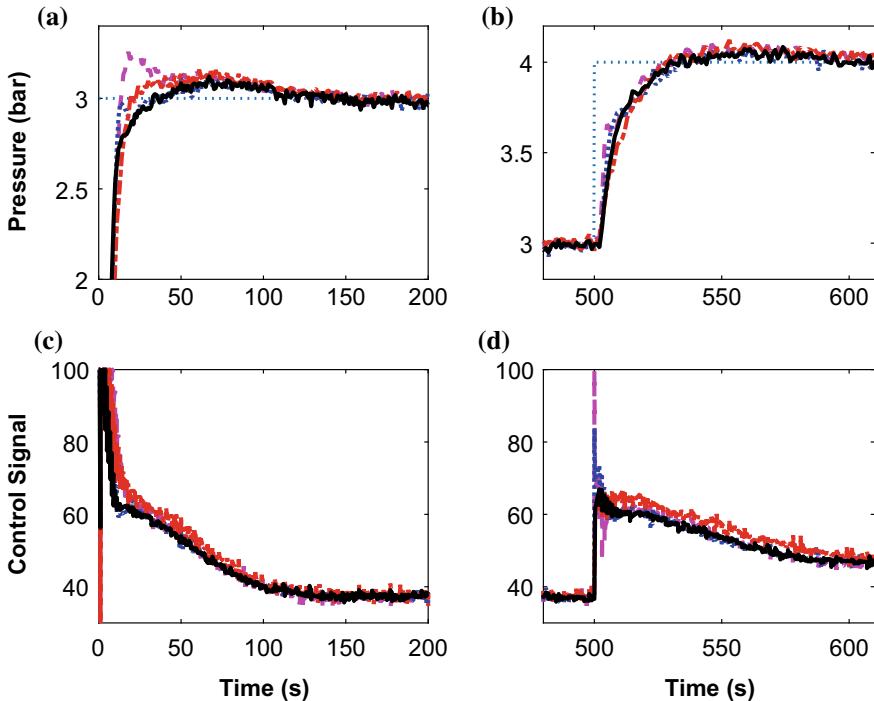


Fig. 2.65 Regions of interest A, B, C and D of Fig. 2.64

Table 2.24 Controller parameters for series form of controllers

Controller	K_p	T_i	T_d	α	b	λ	μ
PID	29.8995	9.9665	0.0585	0.4271	—	—	—
$PI^{\lambda}D^{\mu}$	29.8995	9.9665	0.0585	0.4271	—	0.95	0.50
SWPI-D	29.8995	9.9665	0.0585	0.4271	0.8028	—	—
SWPI $^{\lambda}$ -D $^{\mu}$	29.8995	9.9665	0.0585	0.4271	0.8028	0.95	0.50

2.5.4.1 Controller Parameters

The parameters of the proposed $PI^{\lambda}\text{-}PD^{\mu}$ controller can be obtained from the designed $SWPI^{\lambda}D^{\mu}$ controller parameters as given in Sect. 2.5.3.1 using the conversion relations given in Fig. 2.22. Therefore, the parameters of PI-PD and $PI^{\lambda}\text{-}PD^{\mu}$ controllers are given in Table 2.26.

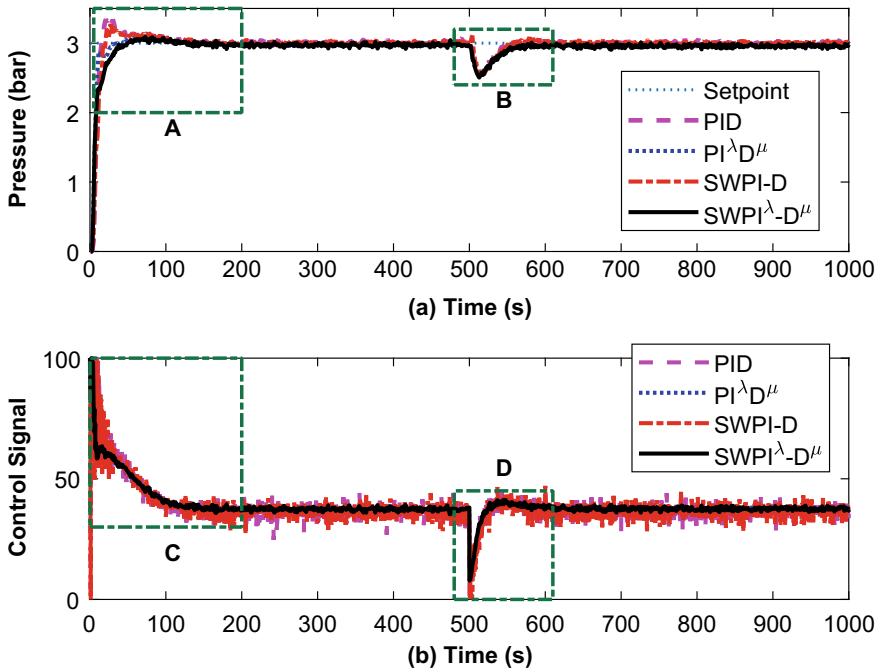


Fig. 2.66 Performance comparison of series $\text{SWPI}^{\lambda}\text{D}^{\mu}$ controller for set-point tracking and disturbance rejection

2.5.4.2 PI-PD Controllers

The comparison of the response of the system with various integer-order controllers (PID, PI-PD1, and PI-PD2) for set-point tracking and disturbance rejection is given in Fig. 2.70 while the zoomed-in view of regions of interest A, B, C and D of the figure is given in Fig. 2.71. The numerical assessment of the figure is given in Table 2.27.

From the figure and the table, it can be observed that the set-point tracking ability of PI-PD1 controller is outperformed those of PID and PI-PD2 in terms of all the parameters. Comparing the control signals of the controllers, it can be seen that the PI-PD1 produced smoother action as compared to the oscillatory actions of both PID and PI-PD2. The smooth control action is due to the structure of the PI-PD.

Similarly, the performance comparison of the various controller for variable set-point tracking is given in Fig. 2.72 while the regions of interest are further highlighted in Fig. 2.73. From the figure, it can be observed that, during step change, PID controller experiences derivative kick effect while both the PI-PD1 and PI-PD2 controllers experienced none of this effect.

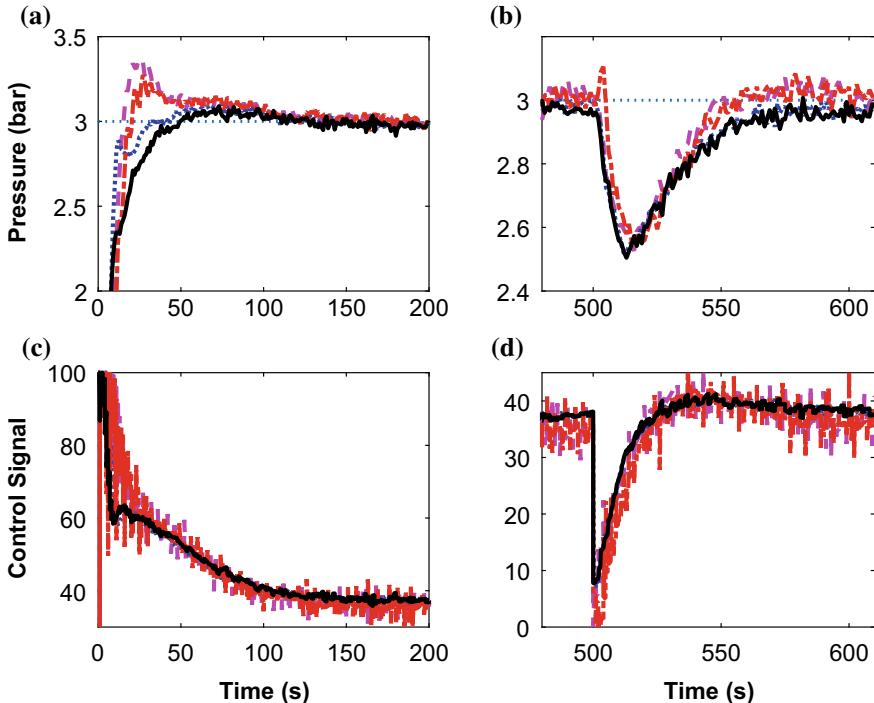


Fig. 2.67 Regions of interest A, B, C and D of Fig. 2.66

Table 2.25 Performance analysis for series form of controllers

Controller	t_r	t_{s_1}	t_{s_2}	%OS
PID	7.6653	95.1112	543.0661	12.5007
$\text{PI}^\lambda \text{D}^\mu$	6.7738	93.1063	841.0198	4.1832
SWPI-D	10.9104	100.4979	541.8869	9.1338
SWPI $^\lambda$ -D $^\mu$	17.2004	33.4194	686.2195	2.9943

2.5.4.3 $\text{PI}^\lambda\text{-PD}^\mu$ Controllers

In a similar way to the integer-order controllers given in previous section, the comparison of the response of the system with various fractional-order controllers $\text{PI}^\lambda \text{D}^\mu$, $\text{PI}^\lambda\text{-PD}^\mu$ 1 and $\text{PI}^\lambda\text{-PD}^\mu$ 2 given in Eqs. (2.1), (2.58) and (2.62) for set-point tracking and disturbance rejection is given in Fig. 2.74 while the zoomed-in view of regions of interest A, B, C and D of the figure is given in Fig. 2.75. The numerical observations are given in Table 2.28.

From the figure and the table, it can be seen that unlike in the case of integer-order where PI-PD1 controller outperformed the other controllers, the performance

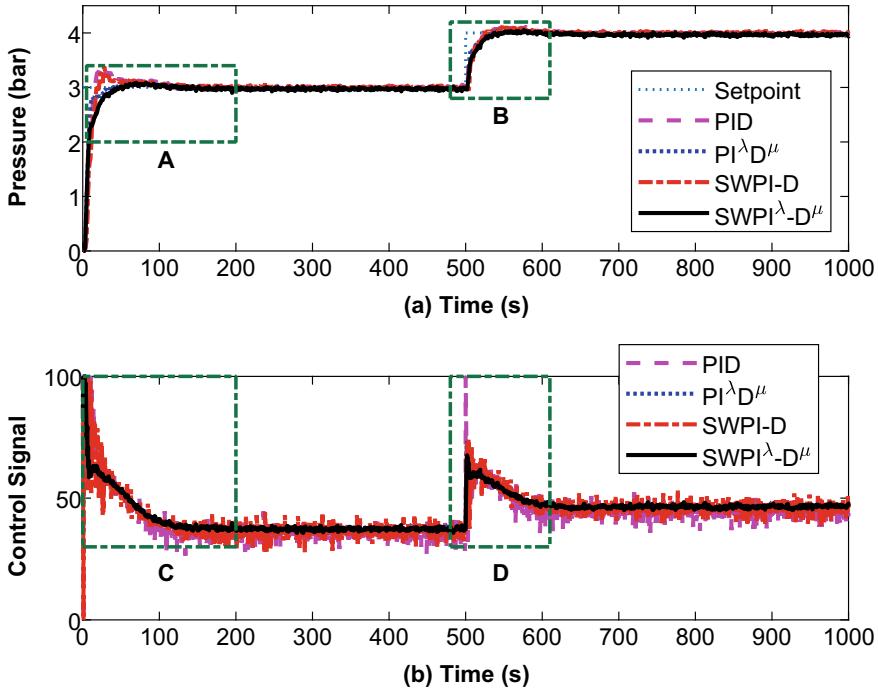


Fig. 2.68 Performance comparison of series $\text{SWPI}^\lambda \text{D}^\mu$ controller for variable set-point tracking

of the two fractional PI-PD controllers ($\text{PI}^\lambda\text{-PD}^\mu 1$ and $\text{PI}^\lambda\text{-PD}^\mu 2$) outperformed that of $\text{PI}^\lambda\text{D}^\mu$.

The two $\text{PI}^\lambda\text{-PD}^\mu$ controllers have each produced an overshoot of around 3.5% while that of the $\text{PI}^\lambda\text{D}^\mu$ is 4.0737%. In terms of rise time, each of the $\text{PI}^\lambda\text{-PD}^\mu 1$ and $\text{PI}^\lambda\text{-PD}^\mu 2$ controllers produced around 6.6 s. This is faster compared to 7.17 s of $\text{PI}^\lambda\text{D}^\mu$. Similar pattern is also observed for both settling times as given in the table.

Similarly, the performance comparison of the various controller for variable set-point tracking is given in Fig. 2.76 while the zoomed-in view of regions of interest A, B, C and D of the figure is given in Fig. 2.77. From the figure, it can be observed that, during step change, the $\text{PI}^\lambda\text{D}^\mu$ experienced derivative kick effect while both the $\text{PI}^\lambda\text{-PD}^\mu 1$ and $\text{PI}^\lambda\text{-PD}^\mu 2$ controllers experienced none of this effect.

2.5.4.4 Comparison of PI-PD and $\text{PI}^\lambda\text{-PD}^\mu$ Controllers

The responses of the system with type 1 ($\text{PI-PD}1$ and $\text{PI}^\lambda\text{-PD}^\mu 1$) and type 2 ($\text{PI-PD}2$ and $\text{PI}^\lambda\text{-PD}^\mu 2$) is given in Fig. 2.78 while the regions of interest of the figure is highlighted in Fig. 2.79. For the numerical results of these figures see Table 2.29.

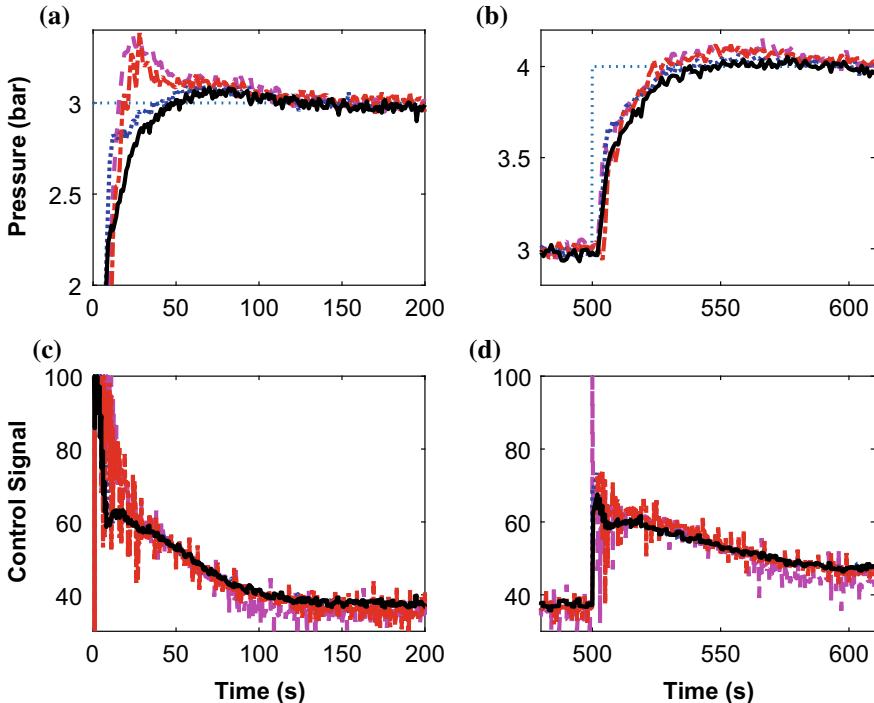


Fig. 2.69 Regions of interest A, B, C and D of Fig. 2.68

Table 2.26 Controller parameters of PI-PD controllers

Controller	P	I	D	λ	μ
PID	30.0	3.0	1.0	—	—
PI-PD	29.8997	3.0000	0.0334	—	—
PI^λ -PD $^\mu$	29.8997	3.0000	0.0334	0.95	0.5

From the figures, considering the type 1 controllers, it can be seen that the PI^λ -PD $^\mu$ 1 controller performed better compared to PI-PD1. The numerical comparison shows that the PI^λ -PD $^\mu$ 1 controller produced the least overshoot of 3.5303% while PI-PD1 has the overshoot of 5.4441%. Furthermore, the PI^λ -PD $^\mu$ 1 controller settled faster at 86.1941 s while PI-PD1 at 91.7818 s. However, in terms of t_r , both the controller has the same rise time of around 6.6 s. The disturbance rejection of both the controllers (PI-PD1 and PI^λ -PD $^\mu$ 1) are similar and satisfactory with PI-PD1 has the fastest settling time of around 240 s.

On the other hand, observing the performance of PI-PD2 and PI^λ -PD $^\mu$ 2, it can be seen that the PI^λ -PD $^\mu$ 2 performed much better when compared to PI-PD2. The numerical comparison also shows that PI^λ -PD $^\mu$ 2 controller has the fastest rise and settling times and very less overshoot of 6.6958 s, 78.1124 s and 3.5257% respectively

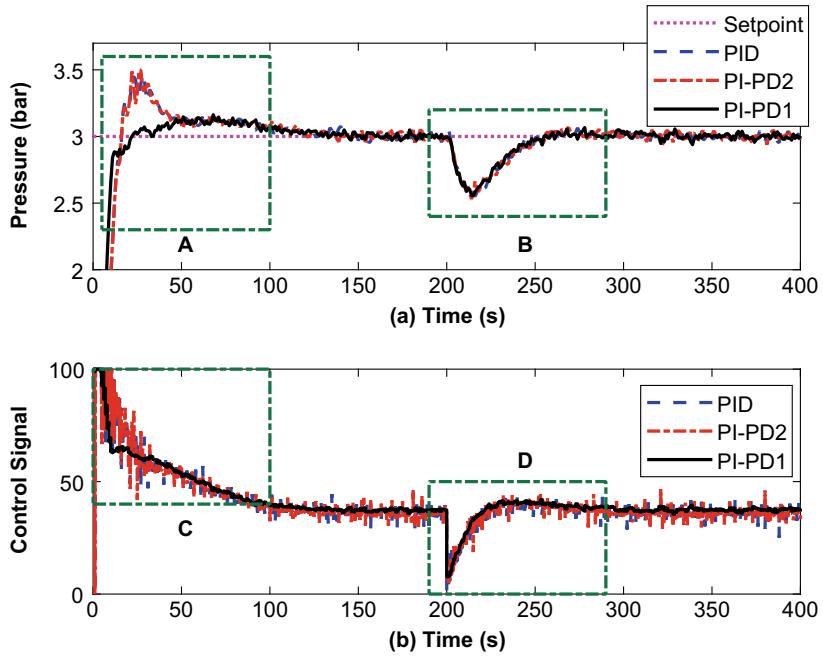


Fig. 2.70 Performance comparison of PI-PD controller for set-point tracking and disturbance rejection

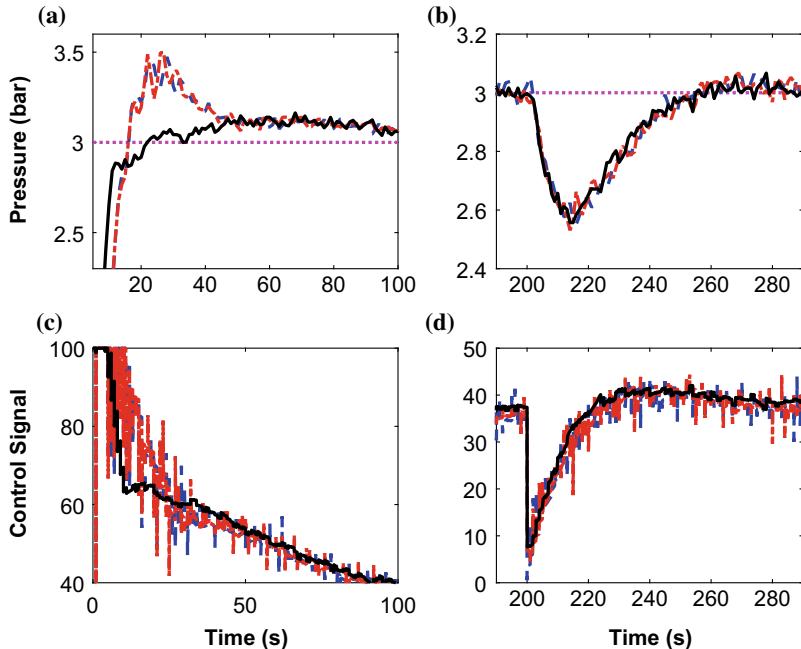
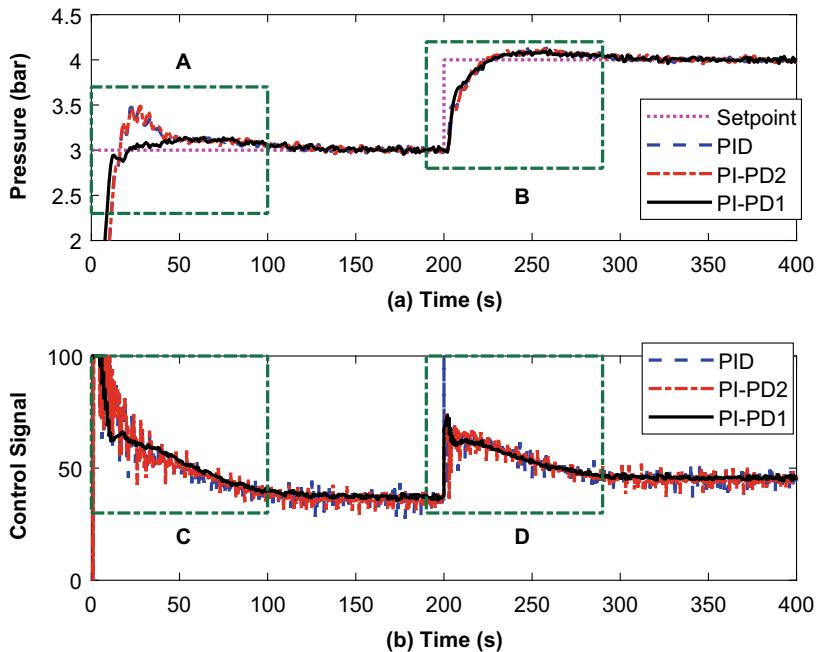


Fig. 2.71 Regions of interest A, B, C and D of Fig. 2.70

Table 2.27 Performance analysis of integer-order controllers

Controller	t_r	t_{s1}	t_{s2}	%OS
PID	7.7953	103.9629	240.9393	15.7821
PI-PD2	8.0101	105.0095	239.8613	16.6024
PI-PD1	6.6580	91.7818	239.3725	5.4441

**Fig. 2.72** Performance comparison of PI-PD controller for variable set-point tracking

when compared to PI-PD2 controller with a rise time of 8.0101 s, settling time of 105.0095 s and overshoot of 16.6024%.

The performance of PI-PD2 controller has been significantly improved by fractional ordering. From the control signals, it can be observed that the $\text{PI}^\lambda\text{-PD}^\mu$ controller produced smoother control action compared to the oscillatory signal of PI-PD2. The undesired oscillations may increase the deterioration rate of the valve.

Similarly, the responses of the system with type 1 and type 2 for variation in set-point is given in Fig. 2.80 and the regions of interest of the figure are highlighted in Fig. 2.81. From the figures it can be seen the both $\text{PI}^\lambda\text{-PD}^\mu$ controllers produced smoother control action while the PI-PD controllers produced undesired oscillations. Furthermore, it can be seen that all the PI-PD controllers are free from derivative kick effects with fractional-order controllers having smoother control action.

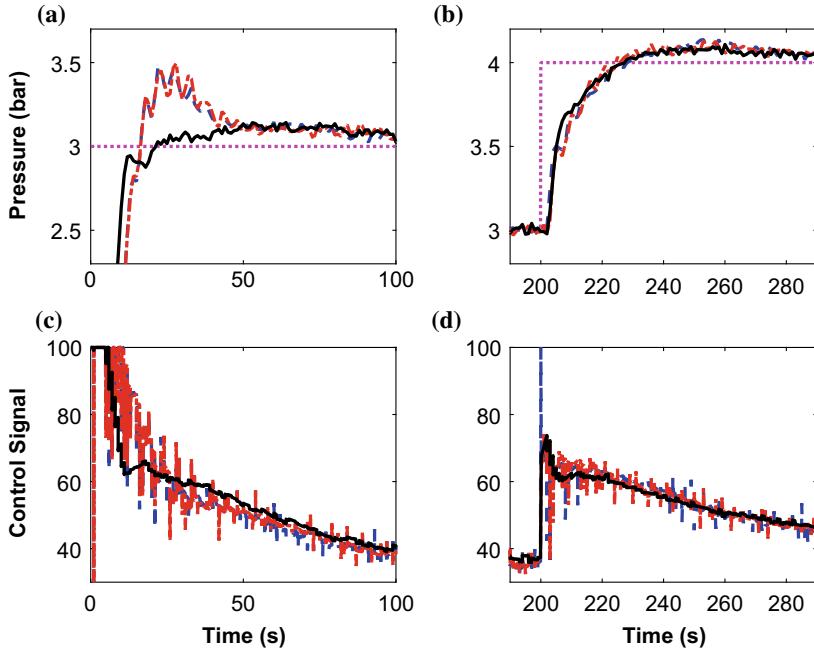


Fig. 2.73 Regions of interest A, B, C and D of Fig. 2.72

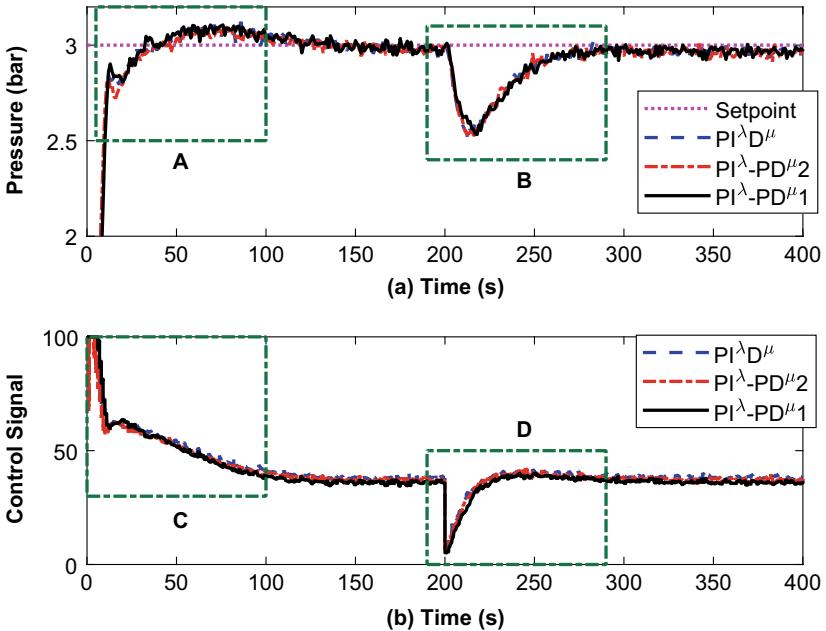


Fig. 2.74 Performance comparison of $\text{PI}^\lambda \text{-PD}^\mu$ controller for set-point tracking and disturbance rejection

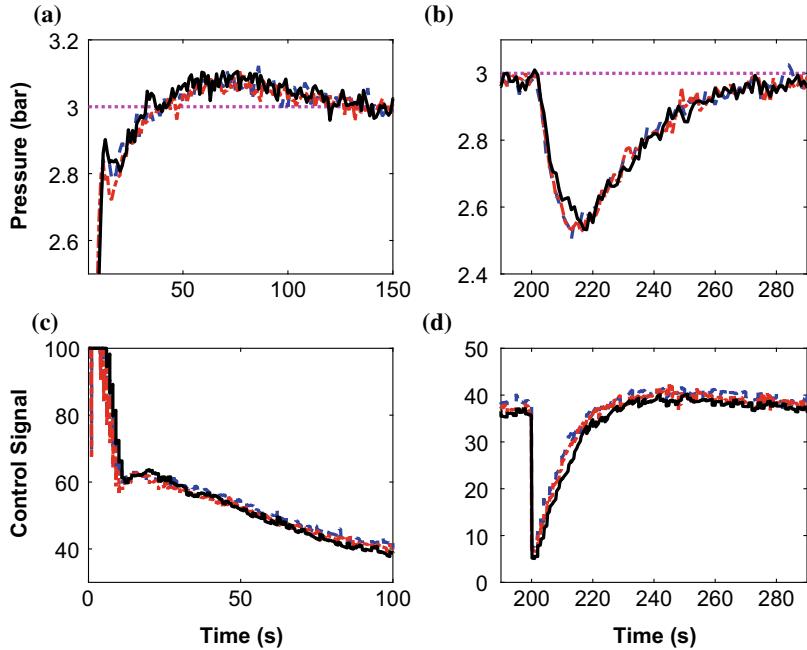


Fig. 2.75 Regions of interest A, B, C and D of Fig. 2.74

Table 2.28 Performance analysis of fractional-order controllers

Controller	t_r	t_{s_1}	t_{s_2}	%OS
$PI^{\lambda}D^{\mu}$	7.1764	86.5618	260.1200	4.0737
$PI^{\lambda}-PD^{\mu}2$	6.6958	78.1124	256.1254	3.5257
$PI^{\lambda}-PD^{\mu}1$	6.6820	86.1941	252.2575	3.5303

2.5.5 Comparison of $SWPI^{\lambda}D^{\mu}$ and $PI^{\lambda}-PD^{\mu}$ Controllers

In this subsection, the comparison of both the proposed controllers on the pressure process plant is presented. First, the performance of all compared controllers will be evaluated for set-point tracking and disturbance rejection. Then, the ability of all the controllers to track variation in set point will be analyzed. In both the cases, as mentioned earlier, a disturbance ($D(s)$) of 30% is injected at 200 s for the purpose of disturbance rejection evaluation. Furthermore, the performance of the controllers to track variation in set point will also be analyzed.

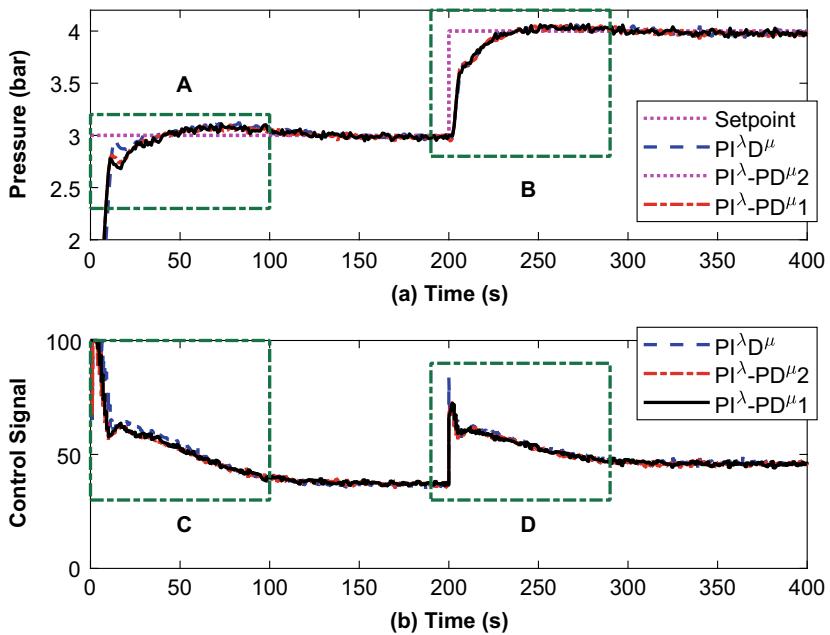


Fig. 2.76 Performance comparison of $\text{PI}^\lambda\text{-PD}^{\mu}$ controller for variable set-point tracking

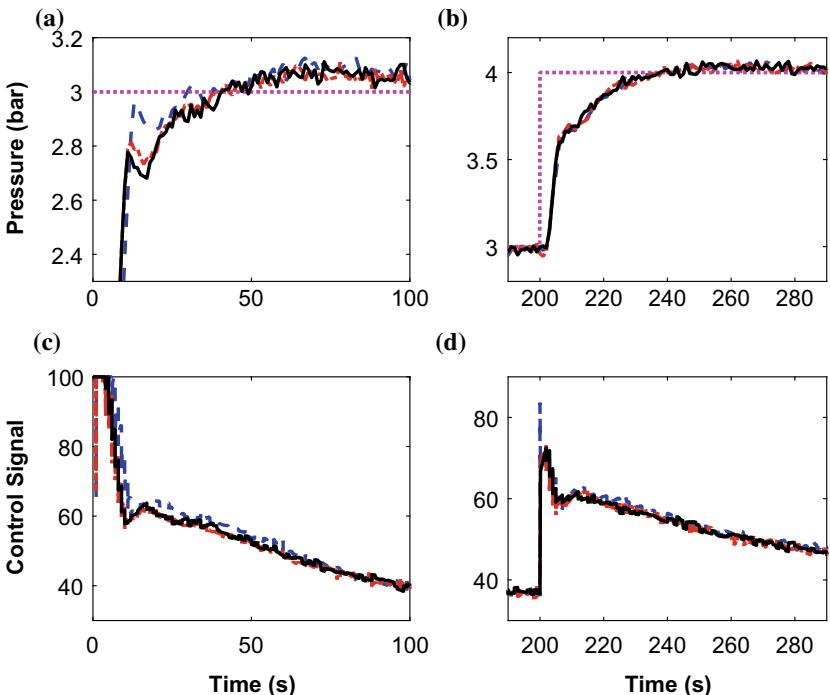


Fig. 2.77 Regions of interest A, B, C and D of Fig. 2.76

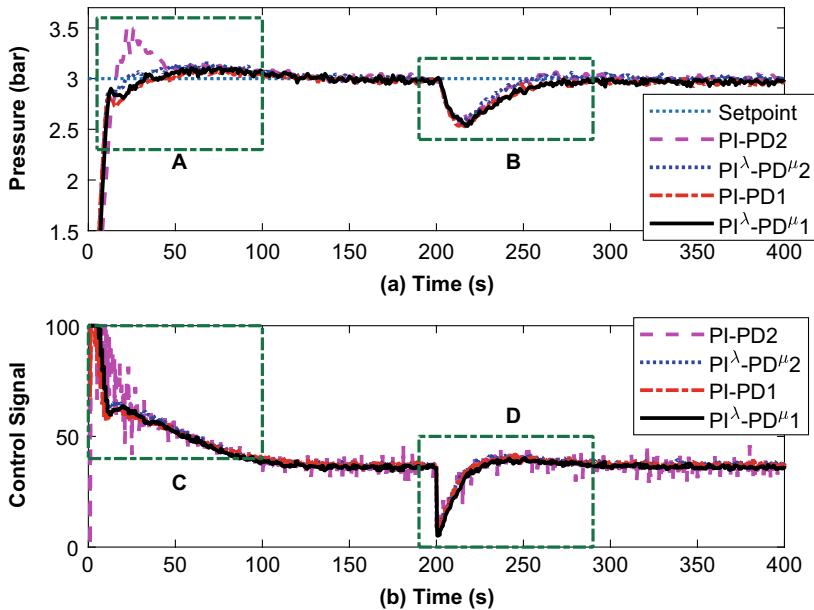


Fig. 2.78 Performance comparison of PI-PD and $\text{PI}^\lambda\text{-PD}^\mu$ for set-point tracking and disturbance rejection

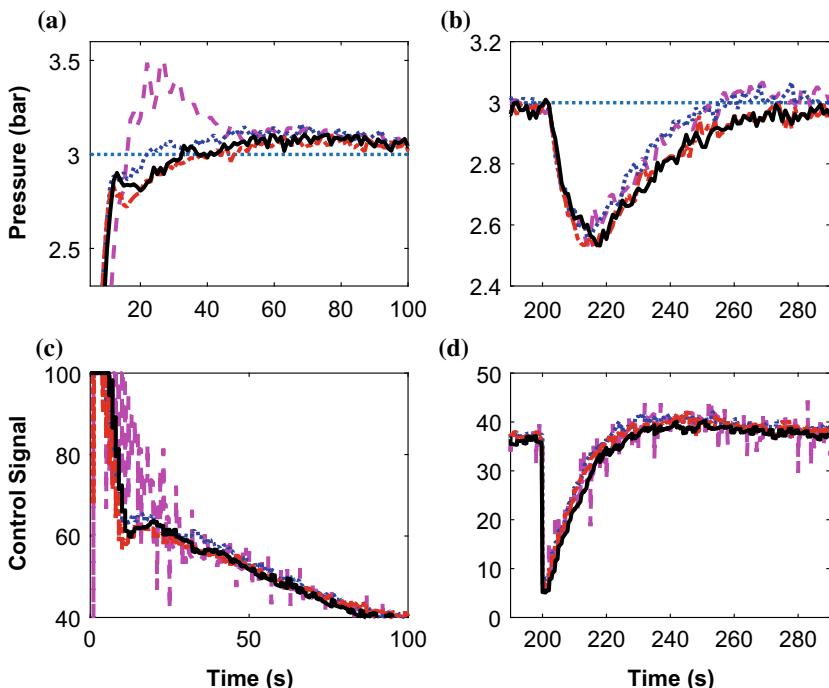


Fig. 2.79 Regions of interest A, B, C and D of Fig. 2.78

Table 2.29 Performance analysis of PI-PD and $\text{PI}^\lambda\text{-PD}^\mu$ controllers

Category	Controller	t_r	t_{s_1}	t_{s_2}	%OS
Integer-order controllers	PID	7.7953	103.9629	240.9393	15.7821
	PI-PD2	8.0101	105.0095	239.8613	16.6024
	PI-PD1	6.6580	91.7818	239.3725	5.4441
Fractional-order controllers	$\text{PI}^\lambda\text{D}^\mu$	7.1764	86.5618	260.1200	4.0737
	$\text{PI}^\lambda\text{-PD}^\mu 2$	6.6958	78.1124	256.1254	3.5257
	$\text{PI}^\lambda\text{-PD}^\mu 1$	6.6820	86.1941	252.2575	3.5303

Table 2.30 Controller parameters of proposed controllers

Category	Controller	K_p	K_i	K_d	b	λ	μ
Integer-order controllers	PID	30.00	3.00	1.00	—	—	—
	SWPI-D	30.00	3.00	1.00	0.80	—	—
	PI-PD2	29.8997	3.00	0.0334	—	—	—
	PI-PD1	29.8997	3.00	0.0334	—	—	—
Fractional-order controllers	$\text{PI}^\lambda\text{D}^\mu$	30.00	3.00	1.00	—	0.95	0.50
	$\text{SWPI}^\lambda\text{-D}^\mu$	30.00	3.00	1.00	0.80	0.95	0.50
	$\text{PI}^\lambda\text{-PD}^\mu 2$	29.8997	3.00	0.0334	—	0.95	0.50
	$\text{PI}^\lambda\text{-PD}^\mu 1$	29.8997	3.00	0.0334	—	0.95	0.50

Table 2.31 Performance analysis of proposed controllers

Category	Controller	t_r	t_{s_1}	t_{s_2}	%OS
Integer-order controllers	PID	6.7511	98.1610	244.3894	8.2935
	SWPI-D	8.4787	94.5839	237.9967	5.1853
	PI-PD2	8.0101	105.0095	239.8613	16.6024
	PI-PD1	6.6580	91.7818	239.3725	5.4441
Fractional-order controllers	$\text{PI}^\lambda\text{D}^\mu$	6.9613	90.5565	282.2311	3.9360
	$\text{SWPI}^\lambda\text{-D}^\mu$	7.3698	83.2167	253.4759	3.1089
	$\text{PI}^\lambda\text{-PD}^\mu 2$	6.6958	78.1124	256.1254	3.5257
	$\text{PI}^\lambda\text{-PD}^\mu 1$	6.6820	86.1941	252.2575	3.5303

2.5.5.1 Set-Point Tracking and Disturbance Rejection

The parameters of various compared integer-order and fractional-order controllers are given in Table 2.30. The comparison of the response of the system with all the compared controllers for set-point tracking and disturbance rejection is given in Fig. 2.82. The regions A, B, C and D of the figure are highlighted in Fig. 2.83.

Observing the figures, it can be seen that the proposed fractional-order controllers are performed better compared to all other integer-order controllers. The numerical assessment of the figure given in Table 2.31 shows that the proposed controllers have

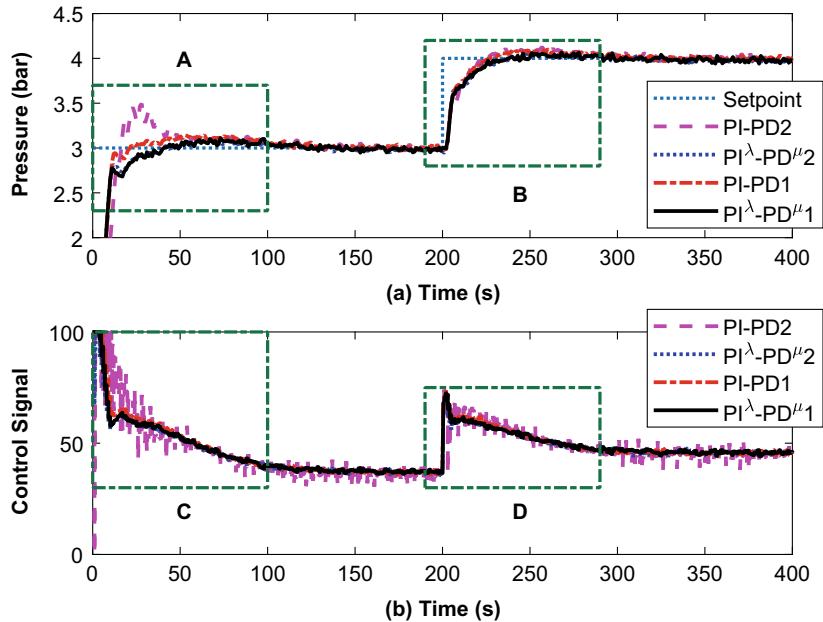


Fig. 2.80 Performance comparison of PI-PD and $\text{PI}^\lambda\text{-PD}^\mu$ for variable set-point tracking

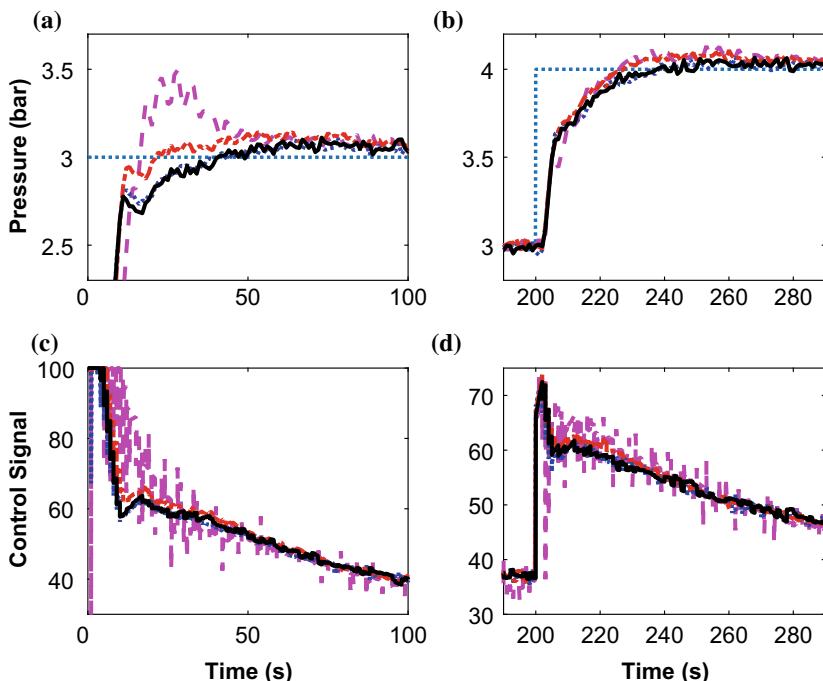


Fig. 2.81 Regions of interest A, B, C and D of Fig. 2.80

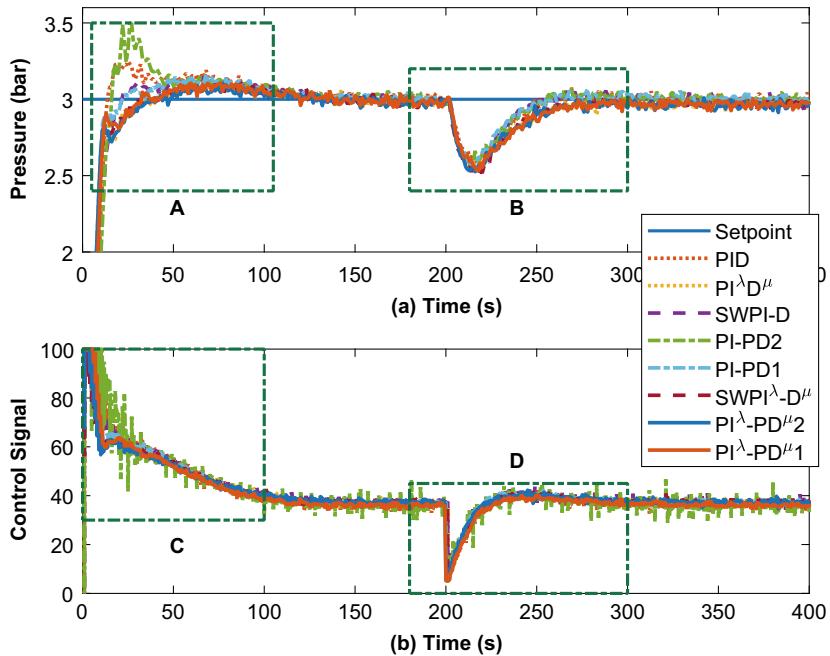


Fig. 2.82 Performance comparison of $\text{SWPI}^\lambda\text{D}^\mu$ and $\text{PI}^\lambda\text{-PD}^\mu$ controllers for set-point tracking and disturbance rejection

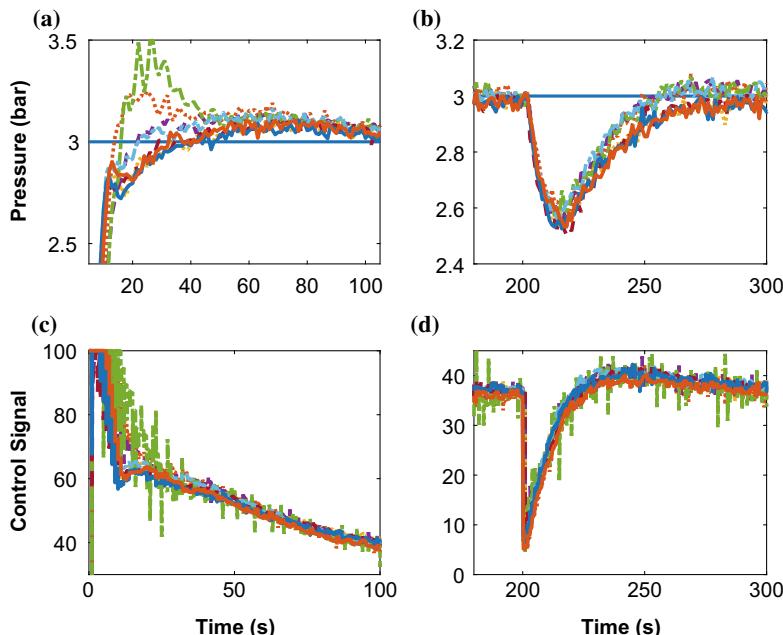


Fig. 2.83 Regions of interest A, B, C and D of Fig. 2.82

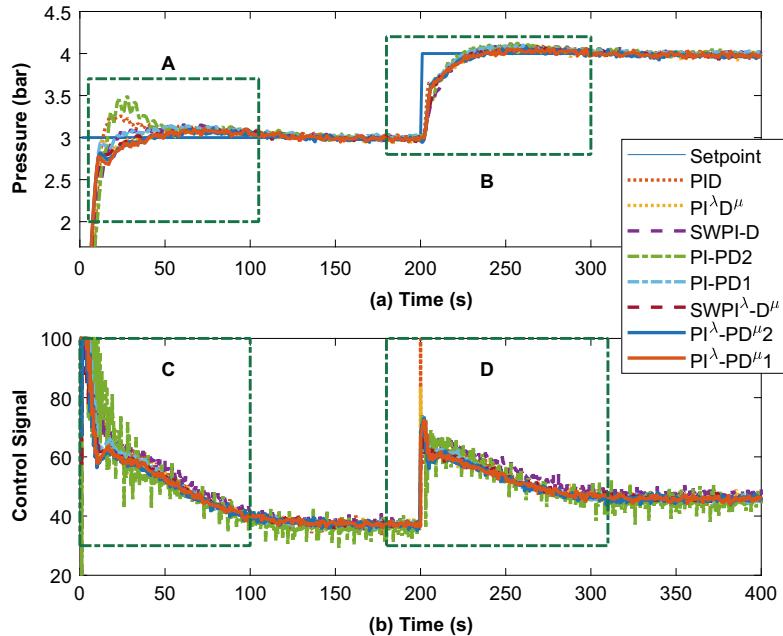


Fig. 2.84 Performance comparison of $\text{SWPI}^\lambda\text{D}^\mu$ and $\text{PI}^\lambda\text{-PD}^\mu$ controllers for variable set-point tracking

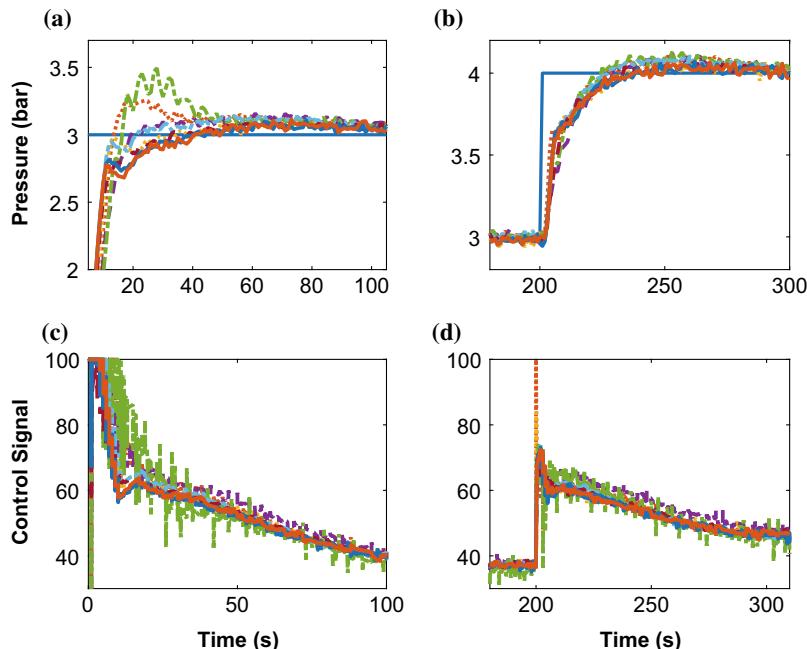


Fig. 2.85 Regions of interest A, B, C and D of Fig. 2.84

the least overshoot of around 3.5%. Furthermore, the controllers also performed better with least t_r and t_{s_1} of around 6.7 s and 82.5 s respectively.

On the other hand, the disturbance rejection performance from the figure shows that all the compared controllers are similar and satisfactory with the integer-order controller have the fastest settling time of around 240 s. This is an indication of the good load regulation ability of the fractional-order controllers. Furthermore, from the control signals, it can be observed that the proposed fractional-order controllers ($\text{PI}^\lambda\text{-PD}^\mu 1$, $\text{PI}^\lambda\text{-PD}^\mu 2$ and $\text{SWPI}^\lambda\text{-D}^\mu$) produced smoother control signal compared to the oscillatory signal of integer-order controllers (PID, PI-PD1, PI-PD2, and SWPI-D). The undesired oscillations and rapid changes in the control signal may increase the deterioration rate of the control valve.

2.5.5.2 Variable Set-Point Tracking

The comparison of the response of the system with all the compared controllers for variable set-point tracking is shown in Fig. 2.84. The regions A, B, C and D of the figure are highlighted in Fig. 2.85.

From the figure, it can be seen that during set-point change, the proposed fractional-order controllers ($\text{PI}^\lambda\text{-PD}^\mu 1$, $\text{PI}^\lambda\text{-PD}^\mu 2$ and $\text{SWPI}^\lambda\text{-D}^\mu$) performed better when compared to the other controllers. This is clearly seen in the zoomed plot of regions of interest B of Fig. 2.85.

Additionally, observing the control signals, it can be seen that during the set-point change the control signals of PID and $\text{PI}^\lambda\text{D}^\mu$ are very narrow and sharp and both reached a peak of 100%. On the other hand, the control actions of both integer-order and fractional-order SWPI-D and PI-PD controllers are peaked at 75%. Furthermore, smoother control actions of proposed fractional-order controllers are observed compared to the oscillatory signal of integer-order controllers. Thus, the reduction in derivative kick effect of the proposed controllers thereby reducing the deterioration rate of the valve. This is due to the setting of derivative set-point weight c which is zero.

2.6 Summary

This chapter has reviewed the PID control strategy and its modifications. The fractional-order PID is an extension of PID which has been extensively used in many control engineering applications to achieve more robust and stable performance. Furthermore, the key challenge in achieving the objective of adequate set-point tracking and disturbance rejection through smoother control action with the conventional PID has been highlighted. Therefore, the design of two fractional-order control strategies namely $\text{SWPI}^\lambda\text{D}^\mu$ and $\text{PI}^\lambda\text{-PD}^\mu$ for the control of highly non-linear, sensitive and dynamic processes is presented. The designed $\text{SWPI}^\lambda\text{D}^\mu$ controller has been developed for standard, industrial, ideal and parallel configurations and also implemented

in single-input-single-output and dual-input-single-output configurations. Similarly, the $\text{PI}^\lambda\text{-PD}^\mu$ controller has been developed in two single-loop feedback configurations. Furthermore, in both the control strategies, the parameters of various control configurations, the effect of changing these parameters and conversion relations for controller parameters have been developed.

On the other hand, the results on real-time pH neutralization and pressure process plants show the proposed controllers have adequate set-point tracking and disturbance rejection through smoother control action compared to PID, PI-PD, set-point weighted PID, and fractional-order PID controllers. Furthermore, during a set-point change, the control action of all compared controllers is sharp and reached a peak of 100% while the proposed controllers around 40%. Thus, the reduction in derivative kick effect of the proposed controllers thereby reducing the deterioration rate of the valve. This is due to the setting of derivative set-point weight c which is zero. The step-response characteristics also show that the proposed controllers have the least overshoot and fastest setting time. However, the proposed controllers have the slowest rise-time. This is because of the proportional set-point weighting factor b which is less than one. In terms of disturbance rejection, the proposed controller shows quick recovery from the effect of disturbance. This is an indication of the good load regulation ability of the fractional-order controllers.

References

1. Wang, Q.G., Ye, Z., Cai, W.J., Hang, C.C.: PID Control for Multivariable Processes. Springer (2008)
2. Grimholt, C., Skogestad, S.: Optimal PI and PID control of first-order plus delay processes and evaluation of the original and improved SIMC rules. *J. Process Control* **70**, 36–46 (2018)
3. Haugen, F.: Comparing PI tuning methods in a real benchmark temperature control system. *Model. Identif. Control* **31**(3), 79–91 (2010)
4. Pachauri, N., Rani, A., Singh, V.: Bioreactor temperature control using modified fractional-order IMC-PID for ethanol production. *Chem. Eng. Res. Des.* **122**, 97–112 (2017)
5. Hermansson, A.W., Syafie, S.: Model predictive control of pH neutralization processes: a review. *Control Eng. Pract.* **45**, 98–109 (2015)
6. Samad, T.: A survey on industry impact and challenges thereof [technical activities]. *IEEE Control Syst. Mag.* **37**(1), 17–18 (2017)
7. Shah, P., Agashe, S.: Review of fractional PID controller. *Mechatronics* **38**, 29–41 (2016)
8. Alfaro, V.M., Vilanova, R.: Conversion formulae and performance capabilities of two-degree-of-freedom PID control algorithms. In: Proceedings of 2012 IEEE 17th International Conference on Emerging Technologies & Factory Automation, Kraków, Poland, 17–21 Sept 2012
9. Ang, K.H., Chong, G., Li, Y.: PID control system analysis, design, and technology. *IEEE Trans. Control Syst. Technol.* **13**(4), 559–576 (2005)
10. Alfaro, V.M., Vilanova, R.: Model-Reference Robust Tuning of PID Controllers. Springer (2016)
11. Mudi, R.K., Dey, C.: Performance improvement of PI controllers through dynamic set-point weighting. *ISA Trans.* **50**(2), 220–230 (2011)
12. Visioli, A.: Practical PID Control. Springer (2006)
13. Alfaro, V.M., Vilanova, R.: Model-reference robust tuning of 2DoF PI controllers for first-and second-order plus dead-time controlled processes. *J. Process Control* **22**(2), 359–374 (2012)

14. Åström, K.J., Hägglund, T.: Advanced PID Control. ISA-The Instrumentation, Systems, and Automation Society (2006)
15. Azarmi, R., Tavakoli-Kakhki, M., Sedigh, A.K., Fatehi, A.: Analytical design of fractional order PID controllers based on the fractional set-point weighted structure: case study in twin rotor helicopter. *Mechatronics* **31**, 222–233 (2015)
16. De Oliveira, E.C., Tenreiro Machado, J.A.: A review of definitions for fractional derivatives and integral. *Math. Probl. Eng.* (2014)
17. Sahu, R.K., Panda, S., Rout, U.K.: DE optimized parallel 2-DOF PID controller for load frequency control of power system with governor dead-band nonlinearity. *Int. J. Electr. Power Energy Syst.* **49**, 19–33 (2013)
18. Mantz, R.J.: A PI controller with dynamic set-point weighting for nonlinear processes. *IFAC Proc. Vol.* **45**(3), 512–517 (2012)
19. Ghosh, A., Krishnan, T.R., Tejaswy, P., Mandal, A., Pradhan, J.K., Ranasingh, S.: Design and implementation of a 2-DOF PID compensation for magnetic levitation systems. *ISA Trans.* **53**(4), 1216–1222 (2014)
20. Jin, Q.B., Liu, Q.: Analytical IMC-PID design in terms of performance/robustness tradeoff for integrating processes: from 2-Dof to 1-Dof. *J. Process Control* **24**(3), 22–32 (2014)
21. Rajinikanth, V., Latha, K.: Setpoint weighted PID controller tuning for unstable system using heuristic algorithm. *Arch. Control Sci.* **22**(4), 481–505 (2012)
22. Rodriguez-Martinez, A., Garduno-Ramirez, R.: 2 DOF fuzzy gain-scheduling PI for combustion turbogenerator speed control. *IFAC Proc. Vol.* **45**(3), 276–281 (2012)
23. Pachauri, N., Singh, V., Rani, A.: Two degree of freedom PID based inferential control of continuous bioreactor for ethanol production. *ISA Trans.* **68**, 235–250 (2017)
24. Bianchi, F.D., Mantz, R.J., Christiansen, C.F.: Multivariable PID control with set-point weighting via BMI optimisation. *Automatica* **44**(2), 472–478 (2008)
25. Sahu, R.K., Panda, S., Rout, U.K., Sahoo, D.K.: Teaching learning based optimization algorithm for automatic generation control of power system using 2-DOF PID controller. *Int. J. Electr. Power Energy Syst.* **77**, 287–301 (2016)
26. Vilanova, R., Visioli, A.: PID Control in the Third Millennium. Springer (2012)
27. Papadopoulos, K.G.: PID Controller Tuning Using the Magnitude Optimum Criterion. Springer (2015)
28. Ingimundarson, A., Hägglund, T.: Performance comparison between PID and dead-time compensating controllers. *J. Process Control* **12**(8), 887–895 (2002)
29. Tan, K.K., Tang, K.Z., Su, Y., Lee, T.H., Hang, C.C.: Deadtime compensation via setpoint variation. *J. Process Control* **20**(7), 848–859 (2010)
30. Larsson, P., Hägglund, T.: Comparison between robust PID and predictive PI controllers with constrained control signal noise sensitivity. *IFAC Proc. Vol.* **45**(3), 175–180 (2012)
31. Kaya, I.: Obtaining controller parameters for a new PI-PD Smith predictor using autotuning. *J. Process Control* **13**(5), 465–472 (2003)
32. Kaya, I.: A PI-PD controller design for control of unstable and integrating processes. *ISA Trans.* **42**(1), 111–121 (2003)
33. Kaya, I.: PI-PD controllers for controlling stable processes with inverse response and dead time. *Electr. Eng.* **98**(1), 55–65 (2016)
34. Zou, H., Li, H.: Improved PI-PD control design using predictive functional optimization for temperature model of a fluidized catalytic cracking unit. *ISA Trans.* **67**, 215–221 (2017)
35. Tsai, K.I., Tsai, C.C.: Design and experimental evaluation of robust PID and PI-PD temperature controllers for oil-cooling machines. In: 9th World Congress on Intelligent Control and Automation, Taipei, Taiwan 21–25 June 2011
36. Padhy, P.K., Majhi, S.: Relay based PI-PD design for stable and unstable FOPDT processes. *Comput. Chem. Eng.* **30**(5), 790–796 (2006)
37. Padhy, S., Panda, S.: A hybrid stochastic fractal search and pattern search technique based cascade PI-PD controller for automatic generation control of multi-source power systems in presence of plug in electric vehicles. *CAAI Trans. Intell. Technol.* **2**(1), 12–25 (2017)

38. Padhy, S., Panda, S., Mahapatra, S.: A modified GWO technique based cascade PI-PD controller for AGC of power systems in presence of plug in electric vehicles. *Int. J. Eng. Sci. Technol.* **20**(2), 427–442 (2017)
39. Nema, S., Kumar Padhy, P.: Identification and cuckoo PI-PD controller design for stable and unstable processes. *Trans. Inst. Meas. Control* **37**(6), 708–720 (2015)
40. Majhi, M.P.V.S., Mahanta, C.: Fuzzy proportional integral-proportional derivative (PI-PD) controller. In: Proceedings of the 2004 American Control Conference, vol. 5, pp. 4028–4033, IEEE (2004)
41. Silva, G.J., Datta, A., Bhattacharyya, S.P.: PID Controllers for Time-Delay Systems. Springer (2007)
42. Liu, G.P., Daley, S.: Optimal-tuning nonlinear PID control of hydraulic systems. *Control Eng. Pract.* **8**(9), 1045–1053 (2000)
43. Su, Y.X., Sun, D., Duan, B.Y.: Design of an enhanced nonlinear PID controller. *Mechatronics* **15**(8), 1005–1024 (2005)
44. Ye, J.: Adaptive control of nonlinear PID-based analog neural networks for a nonholonomic mobile robot. *Neurocomputing* **71**(7–9), 1561–1565 (2008)
45. Prakash, J., Srinivasan, K.: Design of nonlinear PID controller and nonlinear model predictive controller for a continuous stirred tank reactor. *ISA Trans.* **48**(3), 273–282 (2009)
46. Xue, D., Chen, Y., Atherton, D.P.: Linear Feedback Control: Analysis and Design with MATLAB. Siam (2007)
47. Adar, N.G., Kozan, R.: Comparison between real time PID and 2-DOF PID controller for 6-DOF robot arm. *Acta Phys. Pol. A* **130**(1), 269–271 (2016)
48. Alfaro, V.M., Vilanova, R.: Robust tuning of 2DoF five-parameter PID controllers for inverse response controlled processes. *J. Process Control* **23**(4), 453–462 (2013)
49. Alfaro, V.M., Vilanova, R.: Robust tuning and performance analysis of 2DoF PI controllers for integrating controlled processes. *Ind. Eng. Chem. Res.* **51**(40), 13182–13194 (2012)
50. Alfaro, V.M., Vilanova, R.: Simple robust tuning of 2DoF PID controllers from a performance/robustness trade-off analysis. *Asian J. Control* **15**(6), 1700–1713 (2013)
51. Vilanova, R., Alfaro, V.M., Arrieta, O.: Simple robust autotuning rules for 2-DoF PI controllers. *ISA Trans.* **51**(1), 30–41 (2012)
52. Araki, M., Taguchi, H.: Two-degree-of-freedom PID controllers. *Int. J. Control. Autom. Syst.* **1**(4), 401–411 (2003)
53. Li, Z., Liu, L., Dehghan, S., Chen, Y., Xue, D.: A review and evaluation of numerical tools for fractional calculus and fractional order controls. *Int. J. Control* **90**(6), 1165–1181 (2017)
54. Machado, J.T., Kiryakova, V., Mainardi, F.: Recent history of fractional calculus. *Commun. Nonlinear Sci. Numer. Simul.* **16**(3), 1140–1153 (2011)
55. Margarita, R., Sergei, V.R., José, A.T.M., Juan, J.T.: Stability of fractional order systems. *Math. Probl. Eng.* (2013)
56. Krishna, B.T.: Studies on fractional order differentiators and integrators: a survey. *Signal Process.* **91**(3), 386–426 (2011)
57. Valério, D., Trujillo, J.J., Rivero, M., Machado, J.T., Baleanu, D.: Fractional calculus: a survey of useful formulas. *Eur. Phys. J. Spec. Top.* **222**(8), 1827–1846 (2013)
58. Petráš, I.: Fractional-Order Nonlinear Systems: Modeling, Analysis and Simulation. Springer Science & Business Media (2011)
59. Caponetto, R.: Fractional Order Systems: Modeling and Control Applications. World Scientific (2010)
60. Vinagre, B.M., Podlubny, I., Hernandez, A., Feliu, V.: Some approximations of fractional order operators used in control theory and applications. *Fract. Calc. Appl. Anal.* **3**(3), 231–248 (2000)
61. Petráš, I.: Fractional derivatives, fractional integrals, and fractional differential equations in Matlab. Engineering education and research using MATLAB, IntechOpen (2011)
62. Monje, C.A., Vinagre, B.M., Feliu, V., Chen, Y.: Tuning and auto-tuning of fractional order controllers for industry applications. *Control Eng. Pract.* **16**(7), 798–812 (2008)

63. Chen, Y., Bhaskaran, T., Xue, D.: Practical tuning rule development for fractional order proportional and integral controllers. *J. Comput. Nonlinear Dyn.* **3**(2), 021403 (2008)
64. Zamani, M., Karimi-Ghartemani, M., Sadati, N., Parniani, M.: Design of a fractional order PID controller for an AVR using particle swarm optimization. *Control Eng. Pract.* **17**(12), 1380–1387 (2009)
65. Li, H., Luo, Y., Chen, Y.: A fractional order proportional and derivative (FOPD) motion controller: tuning rule and experiments. *IEEE Trans. Control Syst. Technol.* **18**(2), 516–520 (2009)
66. Padula, F., Visioli, A.: Tuning rules for optimal PID and fractional-order PID controllers. *J. Process Control* **21**(1), 69–81 (2011)
67. Sharma, R., Rana, K.P.S., Kumar, V.: Performance analysis of fractional order fuzzy PID controllers applied to a robotic manipulator. *Expert Syst. Appl.* **41**(9), 4274–4289 (2014)
68. Monje, C.A., Chen, Y., Vinagre, B.M., Xue, D., Feliu-Batlle, V.: *Fractional-Order Systems and Controls: Fundamentals and Applications*. Springer Science & Business Media (2010)
69. Luo, Y., Chen, Y.: Fractional order [proportional derivative] controller for a class of fractional order systems. *Automatica* **45**(10), 2446–2450 (2009)
70. Luo, Y., Chao, H., Di, L., Chen, Y.: Lateral directional fractional order (PI) α control of a small fixed-wing unmanned aerial vehicles: controller designs and flight tests. *IET Control Theory Appl.* **5**(18), 2156–2167 (2011)
71. Luo, Y., Chen, Y.: Fractional-order [proportional derivative] controller for robust motion control: tuning procedure and validation. In: American Control Conference, St. Louis, Missouri, USA, 10–12 June 2009
72. Luo, Y., Chen, Y.Q., Wang, C.Y., Pi, Y.G.: Tuning fractional order proportional integral controllers for fractional order systems. *J. Process Control* **20**(7), 823–831 (2010)
73. Tenoutit, M., Maamri, N., Trigeassou, J.C.: An output feedback approach to the design of robust fractional PI and PID controllers. *IFAC Proc. Vol.* **44**(1), 12568–12574 (2011)
74. Lachhab, N., Svaricek, F., Wobbe, F., Rabba, H.: Fractional order PID controller (FOPID)-Toolbox. In: European Control Conference. Zurich, Switzerland, 17–19 July 2013
75. Tenoutit, M., Maamri, N., Trigeassou, J.C.: A time moments approach to the design of robust fractional PID controllers. In: 8th International Multi-conference on Systems, Signals & Devices, Sousse-Tunisia, 22–25 March 2011
76. Merrikh-Bayat, F., Mirebrahimi, N.: Introduction to the nonlinear $PI^\lambda D^\mu$ control. In: 2011 IEEE International Conference on Control System, Computing and Engineering, Penang, Malaysia, 25–27 Nov 2011
77. Tavakoli-Kakhki, M., Haeri, M.: Fractional order model reduction approach based on retention of the dominant dynamics: application in IMC based tuning of FOPI and FOPID controllers. *ISA Trans.* **50**(3), 432–442 (2011)
78. Bettayeb, M., Mansouri, R.: Fractional IMC-PID-filter controllers design for non integer order systems. *J. Process Control* **24**(4), 261–271 (2014)
79. Feliu-Batlle, V., Perez, R.R., Rodriguez, L.S.: Fractional robust control of main irrigation canals with variable dynamic parameters. *Control Eng. Pract.* **15**(6), 673–686 (2007)
80. Feliu-Batlle, V., Rivas-Perez, R., Castillo-Garcia, F.J.: Fractional order controller robust to time delay variations for water distribution in an irrigation main canal pool. *Comput. Electron. Agric.* **69**(2), 185–197 (2009)
81. El-Khazali, R.: Fractional-order $PI^\lambda D^\mu$ controller design. *Comput. Math. Appl.* **66**(5), 639–646 (2013)
82. Freeborn, T.J.: A survey of fractional-order circuit models for biology and biomedicine. *IEEE J. Emerg. Sel. Top. Circuits* **3**(3), 416–424 (2013)
83. Freeborn, T.J., Maundy, B., Elwakil, A.S.: Fractional-order models of supercapacitors, batteries and fuel cells: a survey. *Mater. Renew. Sustain. Energy* **4**(3), 1–9 (2015)
84. Jain, M., Rani, A., Pachauri, N., Singh, V., Mittal, A.P.: Design of fractional order 2-DOF PI controller for real-time control of heat flow experiment. *Eng. Sci. Technol. Int. J.* **22**(1), 215–228 (2019)

85. Pachauri, N., Singh, V., Rani, A.: Two degrees-of-freedom fractional-order proportional-integral-derivative-based temperature control of fermentation process. *J. Dyn. Syst. Meas. Control-Trans. ASME* **140**(7), 071006 (2018)
86. Angel, L., Viola, J.: Fractional order PID for tracking control of a parallel robotic manipulator type delta. *ISA Trans.* **79**, 172–188 (2018)
87. Sharma, R., Gaur, P., Mittal, A.P.: Performance analysis of two-degree of freedom fractional order PID controllers for robotic manipulator with payload. *ISA Trans.* **58**, 279–291 (2015)
88. Sondhi, S., Hote, Y.V.: Fractional order PID controller for load frequency control. *Energy Conv. Manag.* **85**, 343–353 (2014)
89. Zamani, A., Barakati, S.M., Yousofi-Darmian, S.: Design of a fractional order PID controller using GBMO algorithm for load-frequency control with governor saturation consideration. *ISA Trans.* **64**, 56–66 (2016)
90. Lamba, R., Singla, S.K., Sondhi, S.: Fractional order PID controller for power control in perturbed pressurized heavy water reactor. *Nucl. Eng. Des.* **323**, 84–94 (2017)
91. Kumar, N., Tyagi, B., Kumar, V.: Deregulated multiarea AGC scheme using BBBC-FOPID controller. *Arab. J. Sci. Eng.* **42**(7), 2641–2649 (2017)
92. Debbarma, S., Saikia, L.C., Sinha, N.: Automatic generation control using two degree of freedom fractional order PID controller. *Int. J. Electr. Power Energy Syst.* **58**, 120–129 (2014)
93. Tepljakov, A., Gonzalez, E.A., Petlenkov, E., Belikov, J., Monje, C.A., Petráš, I.: Incorporation of fractional-order dynamics into an existing PI/PID DC motor control loop. *ISA Trans.* **60**, 262–273 (2016)
94. Pan, I., Das, S.: Frequency domain design of fractional order PID controller for AVR system using chaotic multi-objective optimization. *Int. J. Electr. Power Energy Syst.* **51**, 106–118 (2013)
95. Kumar, G., Arunshankar, J.: Control of nonlinear two-tank hybrid system using sliding mode controller with fractional-order PI-D sliding surface. *Comput. Electr. Eng.* **71**, 953–965 (2018)
96. Tepljakov, A., Alagoz, B.B., Yeroglu, C., Gonzalez, E., HosseiniNia, S.H., Petlenkov, E.: FOPID controllers and their industrial applications: a survey of recent results. *IFAC-PapersOnLine* **51**(4), 25–30 (2018)
97. Lurie, B.J.: Three-parameter tunable tilt-integral-derivative (TID) controller. US Patent 5,371,670, 6 Dec 1994
98. Abbisso, S., Caponetto, R., Diamante, O., Porto, D., Di Cola, E., Fortuna, L.: Non-integer order dynamic systems. US Patent 6,678,670, 13 Jan 2004
99. Bohannan, G., Hurst, S., Spangler, L.: Electrical component with fractional order impedance. US Patent 11/372,232, 30 Nov 2006
100. Chen, Y.: Tuning methods for fractional-order controllers. US Patent 7,599,752, 6 Oct 2009
101. Almadhoun, M.N., Elshurafa, A., Salama, K., Alshareef, H.: Fractional order capacitor. US Patent 9,305,706, 6 Apr 2016
102. Rana, K.P.S., Kumar, V., Mittra, N., Pramanik, N.: Implementation of fractional order integrator/differentiator on field programmable gate array. *Alex. Eng. J.* **55**(2), 1765–1773 (2016)
103. Muñiz-Montero, C., García-Jiménez, L.V., Sánchez-Gaspariano, L.A., Sánchez-López, C., González-Díaz, V.R., Tlelo-Cuautle, E.: New alternatives for analog implementation of fractional-order integrators, differentiators and PID controllers based on integer-order integrators. *Nonlinear Dyn.* **90**(1), 241–256 (2017)
104. Tolba, M.F., AboAlNaga, B.M., Said, L.A., Madian, A.H., Radwan, A.G.: Fractional order integrator/differentiator: FPGA implementation and FOPID controller application. *AEU-Int. J. Electron. Commun.* **98**, 220–229 (2019)
105. Tepljakov, A.: Fractional-Order Modeling and Control of Dynamic Systems. Springer (2017)
106. Bingi, K., Ibrahim, R., Karsiti, M.N., Hassan, S.M., Harindran, V.R.: A comparative study of 2DOF PID and 2DOF fractional order PID controllers on a class of unstable systems. *Arch. Control Sci.* **28**(4), 635–682 (2018)
107. Bingi, K., Ibrahim, R., Karsiti, M.N., Hassan, S.M.: Fractional order set-point weighted PID controller for pH neutralization process using accelerated PSO algorithm. *Arab. J. Sci. Eng.* **43**(6), 2687–2701 (2018)

108. Bingi, K., Ibrahim, R., Karsiti, M.N., Chung, T.D., Hassan, S.M.: Optimal PID control of pH neutralization plant. In: 2nd IEEE International Symposium on Robotics and Manufacturing Automation. Universiti Teknologi PETRONAS, Malaysia 25–27 Sept 2016
109. Bingi, K., Ibrahim, R., Karsiti, M.N., Hassan, S.M., Harindran, V.R.: Real-time control of pressure plant using 2DOF fractional-order PID controller. *Arab. J. Sci. Eng.* **44**(3), 2091–2102 (2019)

Chapter 3

Approximation Techniques



3.1 Introduction

A key issue with the practical realization or equivalent circuit implementation of fractional-order PID controllers in a finite-dimensional integer-order system is the approximation of the fractional-order parameters for the desired frequency range and the order of approximation. Thus, for the effective approximation of fractional-order integrator and differentiator in the $\text{PI}^\lambda \text{D}^\mu$ controller, researchers have proposed several frequency-domain approximation techniques [1–3]. The summary of these approximation techniques including the MATLAB commands for implementing these algorithms is given in Table 3.1. However, in certain situations, these methods cannot fit around the desired frequency range and also have limitations on order of approximation, the accuracy of frequency and time responses [4, 5].

Among the frequency domain approximation techniques available in the literature, Oustaloup's approximation is the most popular and widely used. This approximation is based on a recursive distribution of poles and zeros in the desired frequency range using frequency response fitting [6]. In certain situations, the Oustaloup's approximation cannot fit around the frequency range of interest [16]. Thus, a modified or refined version has been proposed [6, 13, 17–20]. However, the modified approximation technique produces a higher-order approximated model. Thus, to reduce the order of approximation, [21] proposed a fixed-pole approximation technique.

Similarly, Carlson [8] proposed an approximation technique using Newton iterative method for continued fractional expansion (CFE) of the fractional-order differentiator. However, the technique is limited to fewer values of fractional-order parameter [8, 22]. Furthermore, Matsuda proposed an approximation technique using the gain of the fractional-order transfer function [9]. However, in this method, if the order of the approximation is chosen as an odd number, the approximated transfer function will be improper i.e., there will be one more zero than poles [2, 22, 23]. Researchers also proposed various power series expansion (PSE) techniques based on Taylor series, Maclaurin Series, finite impulse response (FIR) and infinite impulse response

Table 3.1 Summary of approximation techniques for fractional-order parameters

Category	Technique	Syntax ^a	Reference
Continued fractional expansion	CFE Low, CFE High	nid()	[7]
	Carlson		[8]
	Matsuda		[9]
Power series expansion	FIR Filter	dfod1(), dfod2()	[10]
	IIR Filter	dfod3()	[11]
Curve Fitting	Oustaloup	ousta_fod()	[12]
	Refined Oustaloup	new_fod()	[13]
	Charef	-	[14]
	Vector Fitting	-	[15]
Others	Impulse Response	irid_fod(), irid_folpf()	[12]
	Step Response	srid_fod()	

^aThe ‘Syntax’ column denotes the command for the approximation technique available from MATLAB file exchange

(IIR) [9–11]. However, it is proven that the CFE methods are frequently converging more rapidly than PSE methods [2].

In the related development, Charef proposed an approximation technique [14] where accuracy is determined by properly selecting the maximal permissible error. However, the order of approximation involves a significant amount of trials and errors [24, 25]. Thus, an extension of this method was proposed by [17]. This extended method is focused on improving the accuracy of the original proposal. Other approximation algorithms based on stability boundary locus [5], vector fitting method [15], time moments approach [26], state-space approach [27, 28] and frequency distribution mode [18] have been proposed too. A key issue with these methods is that they are quite complex hence difficult to implement.

Therefore, here a simple curve fitting based approximation techniques using exact frequency response data of fractional-order operators (differentiator/integrator) has been proposed. The approach is expected to achieve better approximation compared with the commonly used Oustaloup, refined Oustaloup and Matsuda [12]. To demonstrate the performance of the proposed approach, a simulation study will be conducted on a class of fractional-order systems and controllers.

3.2 Curve Fitting Based Approximation Algorithm

The proposed curve fitting based approximation algorithm for fractional-order differentiator s^α will be done in three stages. The first stage of the approximation is obtaining the frequency response data (frd) of s^γ . This is computed by substitut-

ing $s = j\omega$ and then evaluating the function for different value of $\omega \in (\omega_l, \omega_h)$ as follows:

$$s^\gamma = (j\omega)^\gamma |_{\omega=(\omega_l, \dots, \omega_h)} \quad (3.1)$$

The next stage is to derive the integer-order transfer function of the fractional-order differentiator $G_d(s) = s^\alpha$ from the data obtained in Eq. (3.1) using Sanathanan-Koerner (SK) least square iterative method [29]. Thus, the integer-order transfer function of fractional-order differentiator ($G_d(s)$) is defined as follows:

$$G_d(s) = \frac{P(s)}{Q(s)} \approx \frac{\sum_{n=0}^N p_n s^n}{1 + \sum_{n=1}^N q_n s^n} \approx \frac{P\psi(s)}{1 + Q\phi(s)} \quad (3.2)$$

where

- P and Q are the coefficients and
- $\psi(s)$ and $\phi(s)$ are the monomial functions.

Thus, the coefficients and monomial functions are defined as follows:

$$P = \begin{bmatrix} p_0 \\ p_1 \\ \vdots \\ p_N \end{bmatrix}; \quad Q = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_N \end{bmatrix} \quad (3.3)$$

$$\psi(s) = [1, s, \dots, s^N]; \quad \phi(s) = [s, s^2, \dots, s^N] \quad (3.4)$$

The objective is to identify P and Q of the integer-order transfer function $G_d(s)$ in the desired frequency range $\omega \in (\omega_l, \omega_h)$. This is achieved by minimizing the difference between data samples ($H(s)$) obtained from Eq. (3.1) and $G(s)$ using the following Levy's linearized cost function with the SK's least square iteration method.

$$\arg \min_{P, Q} \sum_{k=l}^h \left| \frac{P(j\omega_k)}{Q^{\tau-1}(j\omega_k)} - \frac{Q^\tau(j\omega_k)}{Q^{\tau-1}(j\omega_k)} H(j\omega_k) \right|^2 \quad (3.5)$$

where $\tau = 1, \dots, \mathcal{T}$ is the iteration step.

From the above equation, it should be noted that the unbiased curve fitting is achieved, as $Q^{\tau-1}(j\omega_k)$ approaches to $Q^\tau(j\omega_k)$. Furthermore, to obtain the minimal realization of $G_d(s)$, the matching poles and zeros will be canceled. The final stage is to convert the obtained state-space model as given in Eq. (3.6) after pole zero cancellation into a transfer function model in the form of:

$$\begin{aligned} sX(s) &= AX(s) + BU(s) \\ Y(s) &= CX(s) + DU(s) \end{aligned} \quad (3.6)$$

$$G_d(s) \approx \frac{Y(s)}{U(s)} = C(sI_N - A)^{-1}B + D \quad (3.7)$$

where

- $X(s)$, $Y(s)$ and $U(s)$ are the state, output and control vectors respectively,
- A , B , C and D are the state, input, output, feedforward matrices respectively.

Therefore, the above procedure for the proposed integer-order approximation of fractional-order differentiator s^α based on the curve fitting of frequency response data with the use of MATLAB inbuilt commands will be implemented as follows:

1. Obtain the frequency response data for integer-order part of s^α within the desired frequency range $\omega \in (\omega_l, \omega_h)$ using MATLAB inbuilt function `frd()`.
2. Obtain the exact frd of s^α by powering the data obtained in previous step with α .
3. Choose the order of the approximate N for the integer-order model.
4. Obtain the state space model of exact function response data based on SK's least square iteration method by using MATLAB inbuilt function `fitfrd()`.
5. Convert the state space model to transfer function using MATLAB inbuilt command `ss2tf()`.

The MATLAB commands for implementing the proposed algorithm is given as follows:

```
function Gp = curveFitting(alpha,N,wl,wh)
w = logspace(log10(wl),log10(wh));
s = tf('s'); FRD = frd(s,w);
FRD.ResponseData = FRD.ResponseData.^alpha;
Gp = fitfrd(FRD,N);
[num,den] = ss2tf(Gp.A,Gp.B,Gp.C,Gp.D);
Gp = tf(num,den);
end
```

This proposed curve fitting approximation is simple and expected to fit the entire frequency range of interest. Furthermore, for effective approximation, the order of fractional-order differentiator s^α is limited to $[-1, 1]$ range [9]. Thus, s^α is divided as follows:

$$s^\alpha = s^{\lfloor \alpha \rfloor} s^{\alpha - \lfloor \alpha \rfloor} \quad (3.8)$$

For example, the 3.2th order of fractional-order differentiator can be written using the above equation as $s^{3.2} = s^3 \times s^{0.2}$. Therefore, the integer-order approximation of fractional-order differentiator $s^{0.2}$ will be obtained using the proposed curve fitting based approximation technique while the integer-order differentiator s^3 will be implemented directly.

3.3 Transfer Function Estimation Algorithm

Similarly, the proposed transfer function estimation algorithm using the Sanathanan-Koerner (SK) least square iterative method presented in the previous section can also be implemented in MATLAB using inbuilt function `t fest()` [30]. Thus, for obtaining the integer-order transfer of fractional-order differentiator s^α using the exact frequency response data is as follows:

1. Obtain the frequency response data for integer-order part of s^α within the desired frequency range $\omega \in (\omega_l, \omega_h)$ using the function `frd()` or `idfrd()`.
2. Obtain the exact frequency response data of s^α by powering the data obtained in the previous step with α .
3. Choose the order of the approximate N for the integer-order model.
4. Obtain the transfer function model of frequency response data using the `t fest()` function.

Thus, the MATLAB commands for estimating the integer-order transfer function using the above-proposed algorithm is given as follows:

```
function Gp = functionEstimation(gam,N,wb,wh)
w = logspace(log10(wb),log10(wh),10000);
s=tf('s'); FRD = idfrd(s,w);
FRD.ResponseData = FRD.ResponseData.^gam;
Gp = tfest(FRD,N,N);
end
```

3.4 Numerical and Stability Analysis

The performance evaluation for the accuracy of approximation techniques will be done in both frequency and time domains. The frequency domain analysis include Bode plot response while for the numerical assessment of the response, the mean absolute error (MAE) defined in Eq. (3.9) is chosen. This type of error function is the most suitable for non-time based problems.

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - x_i| \quad (3.9)$$

where y_i is the approximated frequency response data, x_i is the exact frequency response data and n is the total size of the data.

Furthermore, to evaluate the effectiveness of the proposed approaches, a time domain comparison will also be done. Therefore, the exact step response of the

fractional-order integrator $1/s^\alpha$ is obtained from the inverse Laplace transform of integer-order integrator $1/s^n$ as:

$$L^{-1}\left[\frac{1}{s^n}\right] = \frac{t^{n-1}}{(n-1)!}, \quad n \in \mathbb{N} \quad (3.10)$$

As in the case of Eq. (3.10), the inverse Laplace transform of fractional integrator $1/s^\alpha$ is defined as:

$$L^{-1}\left[\frac{1}{s^\alpha}\right] = \frac{t^{\alpha-1}}{\Gamma(\alpha)}, \quad 0 < \alpha < 1 \quad (3.11)$$

where $\Gamma(\alpha) = (\alpha - 1)!$. From Eq. (3.11), the step response is computed as

$$L^{-1}\left[\frac{1}{s^{\alpha+1}}\right] = \frac{t^\alpha}{\Gamma(\alpha + 1)} = \frac{t^\alpha}{\alpha\Gamma(\alpha)} \quad (3.12)$$

Furthermore, the stability analysis of the approximated transfer function ($G(s)$) will be done using H_2 and H_∞ -norms as defined in Eqs. (3.13) and (3.14) respectively.

$$\|G(s)\|_2 = \sqrt{\frac{1}{2\pi} \int_{-\infty}^{\infty} \text{trace}[G(j\omega)^* G(j\omega)] d\omega} \quad (3.13)$$

$$\|G(s)\|_\infty = \sup_{\omega \in R} |G(j\omega)| \quad (3.14)$$

From the equations, it can be noted that for a stable system the H_2 -norm is the average system gain over all frequencies while H_∞ -norm is the peak gain of the frequency response. However, for an unstable system, these norms will be infinite.

3.5 Performance Analysis of Approximation Algorithms

This section evaluates the performance of both the developed curve fitting based approximation techniques developed. In the first part of this section, the performance of curve fitting approximation has been evaluated on a fractional-order differentiator of order $\alpha = 0.1$. Then, for the same system, the performance of the transfer function estimation algorithm has been evaluated. Next, an approximated transfer functions for orders ($\alpha = 0.1, 0.2, \dots, 0.9$) has been developed using both the proposed approaches. Lastly, a simulation study will be conducted on fractional-order based controllers and systems. The selected systems and controller are differentiator, integrator, PID controller and higher-order transfer function. In all the cases, the obtained results from the proposed approaches will be compared with Oustaloup, refined Oustaloup and Matsuda approximations for MAE's of frequency and step responses. Furthermore, the stability analysis of the approximated transfer functions

will also be done using H_2 and H_∞ -norms as defined in Eqs. (3.13) and (3.14) respectively.

3.5.1 Curve Fitting Based Approximation

To study the proposed curve fitting approach given in Sect. 3.2, the fractional-order differentiator, $s^{0.1}$ is used. For this term, the desired frequency range ω is chosen as $(10^{-2}, 10^2)$. However, to study the effect of variation of N , the order of approximation N is chosen as 4, 5 and 6. Thus, the approximated transfer functions using the proposed approach for orders 4, 5 and 6 are given as follows:

$$s^{0.1} \approx \frac{1.714s^4 + 75.22s^3 + 248.1s^2 + 83.38s + 1.935}{s^4 + 58.57s^3 + 244.1s^2 + 103.5s + 3.237} \quad (3.15)$$

$$s^{0.1} \approx \frac{1.777s^5 + 123.9s^4 + 873.4s^3 + 909.9s^2 + 137.7s + 1.914}{s^5 + 90.81s^4 + 785.4s^3 + 985s^2 + 182.9s + 3.335} \quad (3.16)$$

$$s^{0.1} \approx \frac{1.831s^6 + 183.7s^5 + 2279s^4 + 5199s^3 + 2427s^2 + 203s + 1.9}{s^6 + 128.7s^5 + 1919s^4 + 5142s^3 + 2818s^2 + 283.7s + 3.424} \quad (3.17)$$

The Bode plots of the proposed approach in comparison to Oustaloup, refined Oustaloup and Matsuda for orders 4, 5 and 6 are presented in Figs. 3.1, 3.2 and 3.3 respectively. From the figures, it can be observed that the proposed approach is more accurate within the desired frequency range than the other methods for the order of approximations 5 and 6. The MAE of various techniques is compared for magnitude and phase responses of $s^{0.1}$ in Table 3.2.

From the table and for the order of approximations 5 and 6, the proposed technique has the least MAE. However, for the 4th order approximation, Oustaloup, and refined Oustaloup approximation techniques are better with the least errors. Thus, for the higher order of approximations, the proposed technique gives a better approximation than the other approaches.

The step responses of the proposed approach in comparison to Oustaloup, refined Oustaloup and Matsuda for the orders 4, 5 and 6 are presented in Figs. 3.4, 3.5 and 3.6 respectively. From the figure, similar observation of better accuracy of the proposed approach as in the case of frequency plot is made. Furthermore, the numerical assessment of the step response in Table 3.3 also confirmed that the proposed approach has produced less error when compared to other approaches. On the other hand, the stability analysis from the table also shows that all the approximated transfer functions are stable with a finite value of H_∞ -norm.

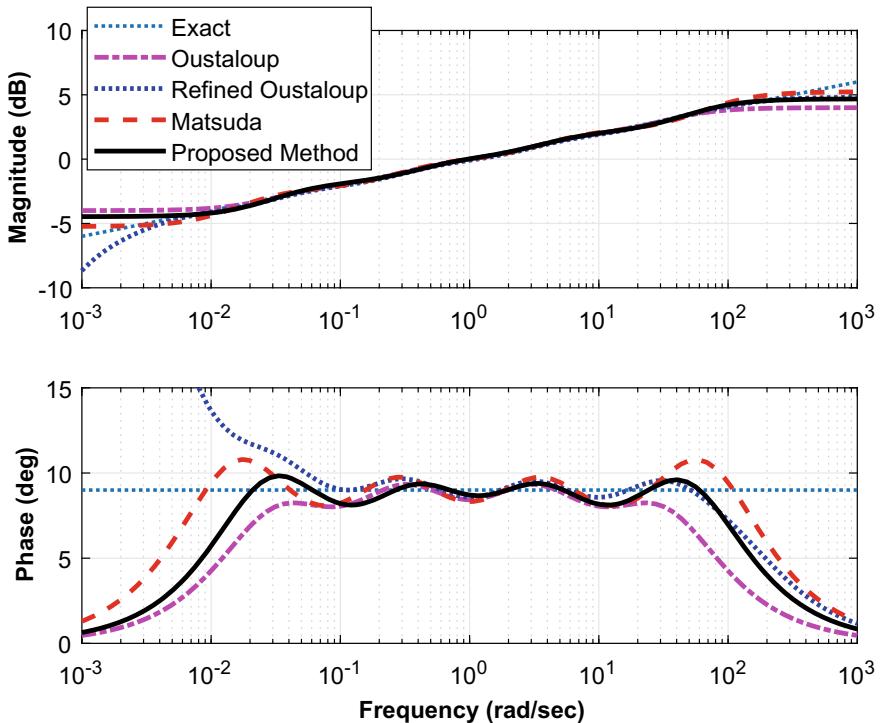


Fig. 3.1 Bode response of $s^{0.1}$ using curve fitting for $N = 4$

Table 3.2 Comparison of MAE's for Bode response of $s^{0.1}$

Technique	Order	$\omega_1 \in (10^{-1}, 10)$		$\omega_2 \in (10^{-2}, 10^2)$	
		Magnitude	Phase	Magnitude	Phase
Oustaloup	4	0.0053	0.4936	0.0091	2.3204
Refined Oustaloup		0.0084	0.2879	0.0154	0.6566
Matsuda		0.0107	0.4887	0.0290	1.0742
Proposed		0.0068	0.3982	0.0227	0.6822
Oustaloup	5	0.0027	0.2530	0.0131	2.3867
Refined Oustaloup		0.0116	0.1664	0.0096	0.5042
Matsuda		0.0035	0.2407	0.0112	0.6956
Proposed		0.0020	0.1433	0.0124	0.2614
Oustaloup	6	0.0007	0.3016	0.0154	2.4373
Refined Oustaloup		0.0119	0.0887	0.0075	0.5241
Matsuda		0.0018	0.0795	0.0067	0.3352
Proposed		0.0011	0.0541	0.0058	0.1415

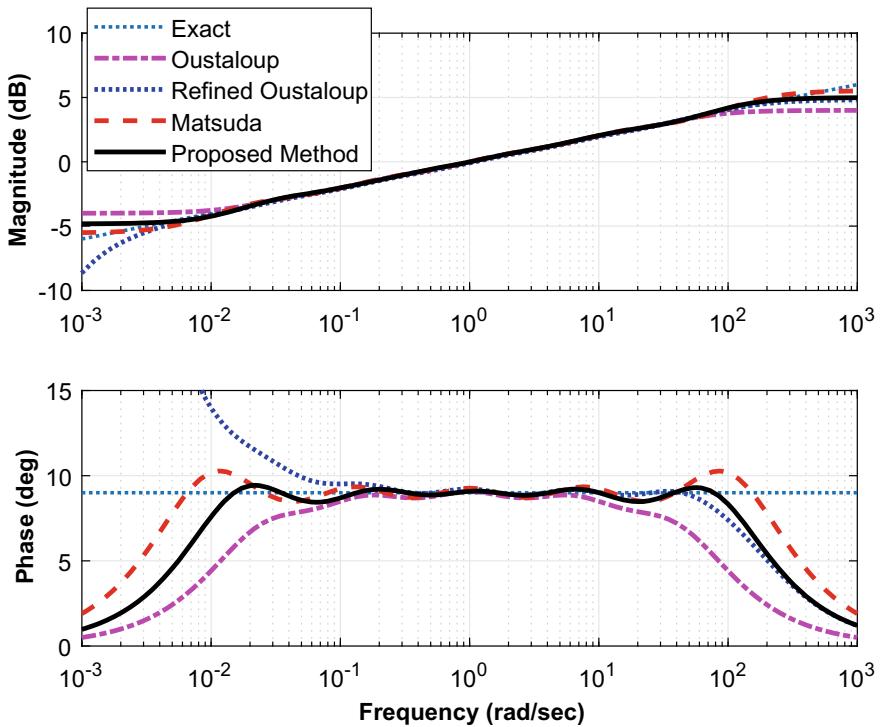


Fig. 3.2 Bode response of $s^{0.1}$ using curve fitting for $N = 5$

Table 3.3 Comparison of MAE's for step response of $s^{0.1}$

Technique	Order	$t_1 \in (0, 125)$	$t_2 \in (0, 250)$	H_∞ -norm	Stability
Oustaloup	4	0.0215	0.0441	1.5849	Stable
Refined Oustaloup		0.0547	0.0809	1.7425	Stable
Matsuda		0.0144	0.0122	1.8280	Stable
Proposed		0.0061	0.0199	1.7145	Stable
Oustaloup	5	0.0237	0.0455	1.5849	Stable
Refined Oustaloup		0.0540	0.0813	1.7425	Stable
Matsuda		0.0110	0.0120	1.8905	Stable
Proposed		0.0053	0.0097	1.7773	Stable
Oustaloup	6	0.0247	0.0463	1.5849	Stable
Refined Oustaloup		0.0538	0.0814	1.7425	Stable
Matsuda		0.0103	0.0110	1.9462	Stable
Proposed		0.0071	0.0059	1.8315	Stable

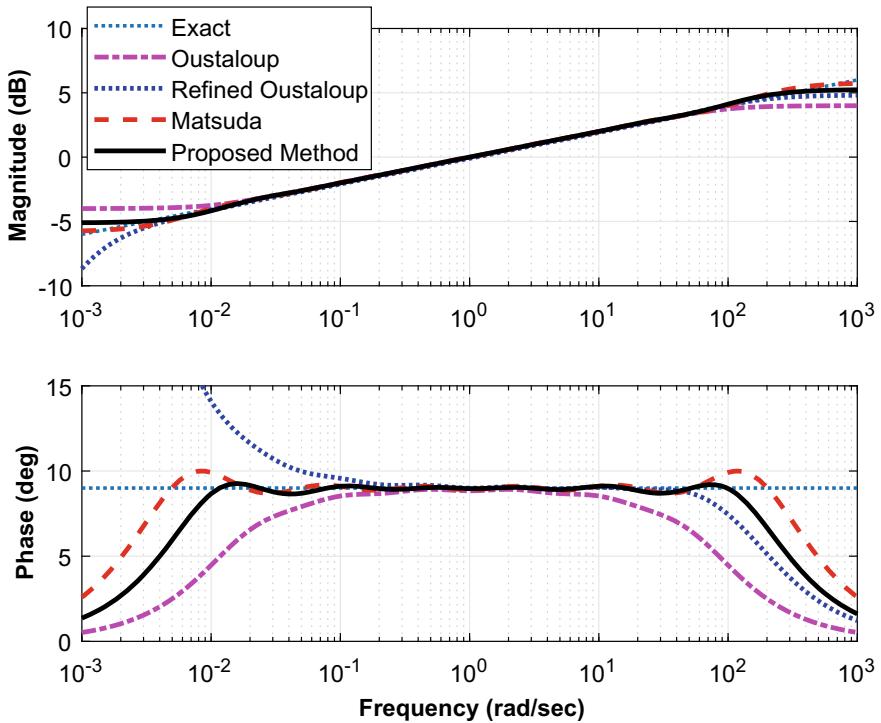


Fig. 3.3 Bode response of $s^{0.1}$ using curve fitting for $N = 6$

3.5.2 Transfer Function Estimation Algorithm

In a similar way, to study the transfer function estimation algorithm given in Sect. 3.3, the fractional-order differentiator of order $\alpha = 0.1$ is used. For the integer-order transfer function estimation of this term, the desired frequency range ω is chosen as $(10^{-2}, 10^2)$ and order of approximation N is chosen as 4, 5 and 6. Thus, the approximated transfer function using the proposed approach for orders 4, 5 and 6 are given as follows:

$$s^{0.1} \approx \frac{1.758s^4 + 109.7s^3 + 527.2s^2 + 194.7s + 4.272}{s^4 + 81.96s^3 + 504.9s^2 + 242.2s + 7.147} \quad (3.18)$$

$$s^{0.1} \approx \frac{1.824s^5 + 177s^4 + 1839s^3 + 2326s^2 + 330.3s + 4.043}{s^5 + 124.8s^4 + 1600s^3 + 2501s^2 + 444.4s + 7.103} \quad (3.19)$$

$$s^{0.1} \approx \frac{1.881s^6 + 259.7s^5 + 4740s^4 + 14170s^3 + 6852s^2 + 492.8s + 3.919}{s^6 + 175.3s^5 + 3850s^4 + 13770s^3 + 8010s^2 + 701.2s + 7.152} \quad (3.20)$$

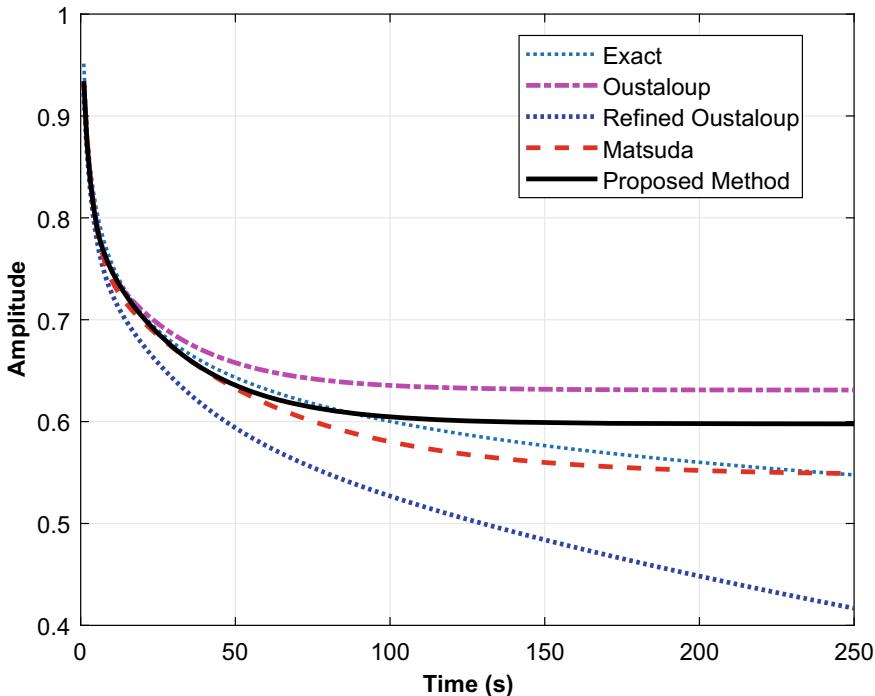


Fig. 3.4 Step response of $s^{0.1}$ using curve fitting for $N = 4$

The Bode plots of the proposed approach in comparison to Oustaloup, refined Oustaloup and Matsuda for orders 4, 5 and 6 are presented in Figs. 3.7, 3.8 and 3.9 respectively. From the figures, a similar observation of the better performance of the proposed approach as in the case of curve fitting algorithm is made.

The MAE of various techniques is compared for magnitude and phase responses of $s^{0.1}$ in Table 3.4. From the table and for the desired frequency of ω_2 , the proposed approach has the least MAE. However, for the desired frequency ω_1 , Oustaloup and refined Oustaloup approximation techniques are better with the least error. Thus, for the high-frequency ranges, the proposed technique gives a better approximation than the other approaches.

Furthermore, the step responses of the proposed approach in comparison to Oustaloup, refined Oustaloup and Matsuda for the orders 4, 5 and 6 are presented in Figs. 3.10, 3.11 and 3.12 respectively. From the figure, similar observation of better accuracy of the proposed approach as in the case of frequency plot is made. The numerical assessment of the step response in Table 3.5 also confirmed that for longer time periods, the proposed approach gives a better approximation than the other approaches. On the other hand, the stability analysis from the table also shows that all the approximated transfer functions are stable with a finite value of H_∞ -norm.

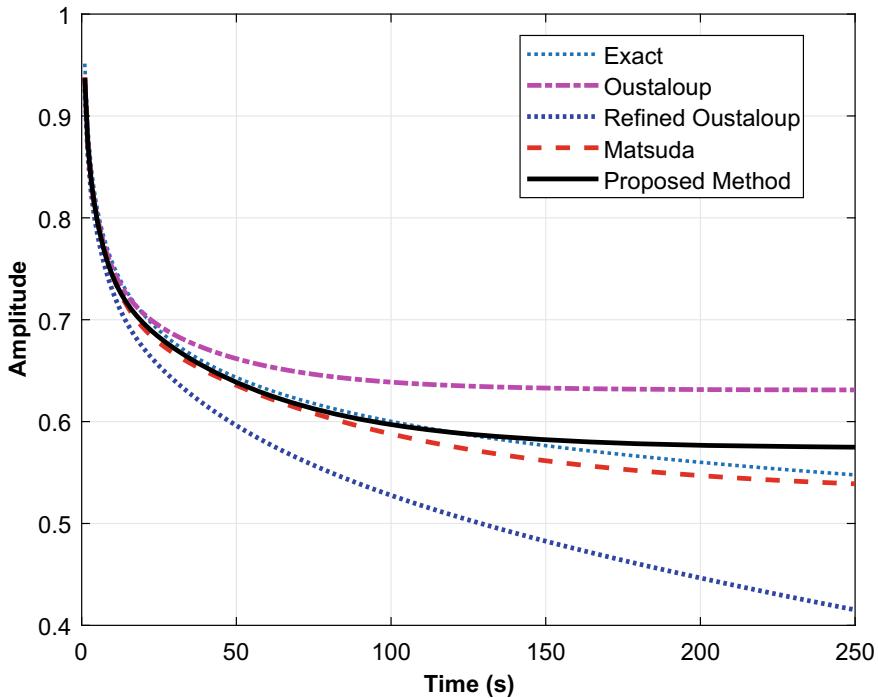


Fig. 3.5 Step response of $s^{0.1}$ using curve fitting for $N = 5$

Table 3.4 Comparison of MAE's for Bode response of $s^{0.1}$

Technique	Order	$\omega_1 \in (10^{-1}, 10)$		$\omega_2 \in (10^{-2}, 10^2)$	
		Magnitude	Phase	Magnitude	Phase
Oustaloup	4	0.0053	0.4936	0.0091	2.3204
Refined Oustaloup		0.0084	0.2879	0.0154	0.6566
Matsuda		0.0107	0.4887	0.0290	1.0742
Proposed		0.0081	0.4439	0.0134	0.3664
Oustaloup	5	0.0027	0.2530	0.0131	2.3867
Refined Oustaloup		0.0116	0.1664	0.0096	0.5042
Matsuda		0.0035	0.2407	0.0112	0.6956
Proposed		0.0047	0.1613	0.0059	0.1491
Oustaloup	6	0.0007	0.3016	0.0154	2.4373
Refined Oustaloup		0.0119	0.0887	0.0075	0.5241
Matsuda		0.0018	0.0795	0.0067	0.3352
Proposed		0.0016	0.0980	0.0022	0.0871

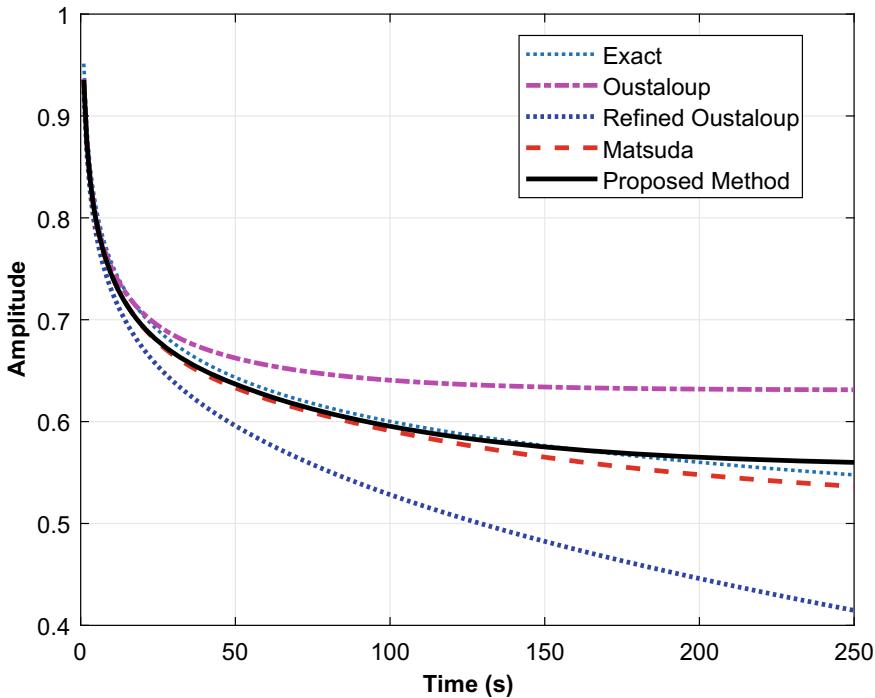


Fig. 3.6 Step response of $s^{0.1}$ using curve fitting for $N = 6$

Table 3.5 Comparison of MAE's for step response of $s^{0.1}$

Technique	Order	$t_1 \in (0, 125)$	$t_2 \in (0, 250)$	H_∞ -norm	Stability
Oustaloup	4	0.0215	0.0441	1.5849	Stable
Refined Oustaloup		0.0547	0.0809	1.7425	Stable
Matsuda		0.0144	0.0122	1.8280	Stable
Proposed		0.0048	0.0194	1.7576	Stable
Oustaloup	5	0.0237	0.0455	1.5849	Stable
Refined Oustaloup		0.0540	0.0813	1.7425	Stable
Matsuda		0.0110	0.0120	1.8905	Stable
Proposed		0.0043	0.0082	1.8239	Stable
Oustaloup	6	0.0247	0.0463	1.5849	Stable
Refined Oustaloup		0.0538	0.0814	1.7425	Stable
Matsuda		0.0103	0.0110	1.9462	Stable
Proposed		0.0071	0.0052	1.8809	Stable

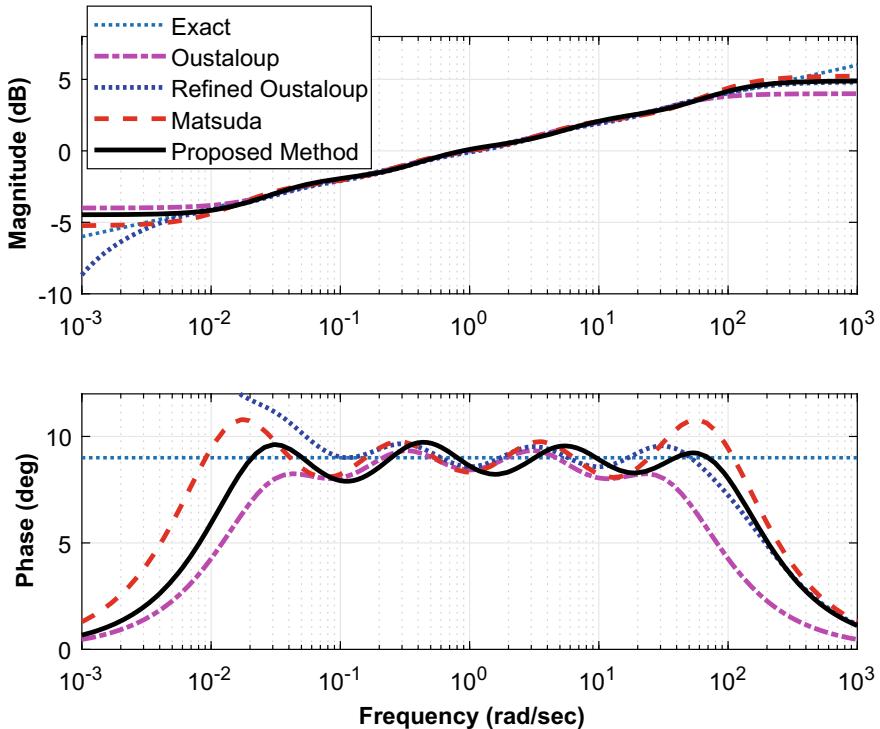


Fig. 3.7 Bode response of $s^{0.1}$ using transfer function estimation algorithm for $N = 4$

3.6 Approximation Table for Fractional-order Differentiator

The approximated transfer functions for s^α ($\alpha = 0.1, 0.2, \dots, 0.9$) using both the curve fitting approximation and transfer function estimation algorithm are given in Tables 3.6 and 3.7 respectively [31].

These approximated transfer functions can be used directly to obtain the approximation of fractional-order based systems and controllers. The comparison of the accuracy of the proposed approaches with Oustaloup, refined Oustaloup and Matsuda is shown in Table 3.8. From the table and for the desired frequency of ω_2 , both the proposed approaches have the least MAE in all the values of α for both magnitude and phase.

Furthermore, for the desired frequency of ω_1 , the proposed approach outperformed all the others in phase response. However, for magnitude, the Oustaloup produced better approximation in all but two cases ($\alpha = 0.6, 0.9$) where the proposed approaches are better. It should be noted despite the Oustaloup being the best for ω_1 magnitude, the proposed approaches were very close in all the cases. On the other hand, from the time domain comparison, it can be noted that for longer time periods,

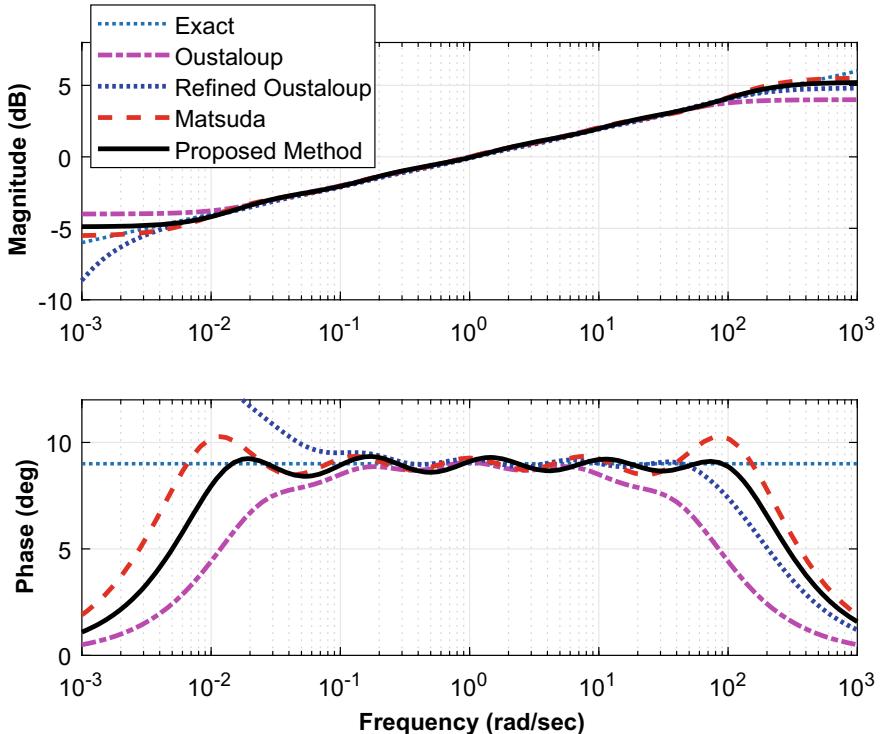


Fig. 3.8 Bode response of $s^{0.1}$ using transfer function estimation algorithm for $N = 5$

Table 3.6 Approximation table for s^α using curve fitting based approximation

s^α	Approximated transfer function using curve fitting
$s^{0.1}$	$\frac{1.777s^5 + 123.9s^4 + 873.4s^3 + 909.9s^2 + 137.7s + 1.914}{s^5 + 90.81s^4 + 785.4s^3 + 985s^2 + 182.9s + 3.335}$
$s^{0.2}$	$\frac{3.233s^5 + 223s^4 + 1624s^3 + 1762s^2 + 275.3s + 3.725}{s^5 + 116.8s^4 + 1279s^3 + 2011s^2 + 473s + 11.14}$
$s^{0.3}$	$\frac{6.048s^5 + 413.7s^4 + 3111s^3 + 3513s^2 + 565.4s + 7.36}{s^5 + 151s^4 + 2089s^3 + 4115s^2 + 1224s + 37.33}$
$s^{0.4}$	$\frac{11.7s^5 + 794.7s^4 + 6166s^3 + 7236s^2 + 1198s + 14.71}{s^5 + 197.2s^4 + 3438s^3 + 8478s^2 + 3181s + 125.9}$
$s^{0.5}$	$\frac{23.59s^5 + 1594s^4 + 12740s^3 + 15520s^2 + 2632s + 29.63}{s^5 + 262.6s^4 + 5753s^3 + 17720s^2 + 8368s + 429.6}$
$s^{0.6}$	$\frac{50.26s^5 + 3381s^4 + 27790s^3 + 35050s^2 + 6070s + 59.93}{s^5 + 361.4s^4 + 9918s^3 + 38090s^2 + 22550s + 1498}$
$s^{0.7}$	$\frac{116s^5 + 7769s^4 + 65570s^3 + 85390s^2 + 15010s + 121.2}{s^5 + 526.7s^4 + 18060s^3 + 86240s^2 + 63740s + 5453}$
$s^{0.8}$	$\frac{305.7s^5 + 20410s^4 + 176400s^3 + 236400s^2 + 41890s + 244.3}{s^5 + 857.9s^4 + 36660s^3 + 217000s^2 + 199500s + 21830}$
$s^{0.9}$	$\frac{1092s^5 + 72700s^4 + 642000s^3 + 881700s^2 + 156000s + 489.3}{s^5 + 1852s^4 + 98540s^3 + 720500s^2 + 819500s + 113800}$

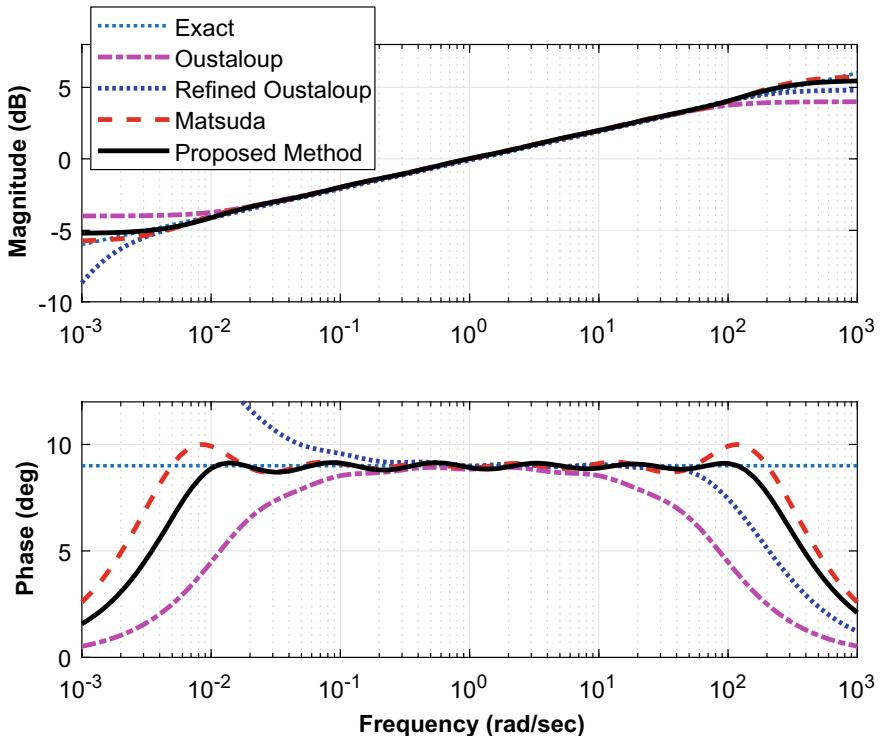


Fig. 3.9 Bode response of $s^{0.1}$ using transfer function estimation algorithm for $N = 6$

Table 3.7 Approximation table for s^α using transfer function estimation algorithm

s^α	Approximated transfer function using transfer function estimation
$s^{0.1}$	$\frac{1.824s^5 + 177s^4 + 1839s^3 + 2326s^2 + 330.3s + 4.043}{s^5 + 124.8s^4 + 1600s^3 + 2501s^2 + 444.4s + 7.103}$
$s^{0.2}$	$\frac{3.455s^5 + 361.7s^4 + 4476s^3 + 7173s^2 + 1276s + 17.44}{s^5 + 171.8s^4 + 3183s^3 + 7732s^2 + 2170s + 51.73}$
$s^{0.3}$	$\frac{6.807s^5 + 762.6s^4 + 11040s^3 + 22190s^2 + 4988s + 77.03}{s^5 + 234.5s^4 + 6075s^3 + 22370s^2 + 1.001 \times 10^4 s + 370.7}$
$s^{0.4}$	$\frac{13.99s^5 + 1668s^4 + 2.782e04s^3 + 69160s^2 + 19720s + 343.9}{s^5 + 320.5s^4 + 1.129 \times 10^4 s^3 + 61150s^2 + 43410s + 2584}$
$s^{0.5}$	$\frac{30.19s^5 + 3800s^4 + 71770s^3 + 216200s^2 + 77260s + 1479}{s^5 + 442.9s^4 + 20660s^3 + 1.588 \times 10^5 s^2 + 1.742 \times 10^5 s + 16870}$
$s^{0.6}$	$\frac{69.07s^5 + 9111s^4 + 191400s^3 + 681900s^2 + 298900s + 5913}{s^5 + 627.5s^4 + 37900s^3 + 398300s^2 + 651000s + 101400}$
$s^{0.7}$	$\frac{171.2s^5 + 23420s^4 + 536800s^3 + 2.2 \times 10^6 s^2 + 1.1 \times 10^6 s + 21150}{s^5 + 933.1s^4 + 71490s^3 + 990800s^2 + 2.3 \times 10^6 s + 557300}$
$s^{0.8}$	$\frac{484.9s^5 + 68230s^4 + 1.6 \times 10^6 s^3 + 7.7 \times 10^6 s^2 + 4.6 \times 10^6 s + 68530}{s^5 + 1541s^4 + 147600s^3 + 2.6 \times 10^6 s^2 + 8.3 \times 10^6 s + 3.0 \times 10^6}$
$s^{0.9}$	$\frac{1855s^5 + 2.6 \times 10^5 s^4 + 6.9 \times 10^6 s^3 + 3.5 \times 10^7 s^2 + 2.3 \times 10^7 s + 1.9 \times 10^5}{s^5 + 3349s^4 + 3.9 \times 10^5 s^3 + 8.7 \times 10^6 s^2 + 3.7 \times 10^7 s + 1.9 \times 10^7}$

Table 3.8 Numerical and stability analysis of approximated transfer functions using both the proposed algorithms

s^α	Technique	$\omega_1 \in (10^{-1}, 10)$		$\omega_2 \in (10^{-2}, 10^2)$		$t_1 \in (0, 125)$	$t_2 \in (0, 250)$	H_∞ -norm	Stability
		Mag	Phase	Mag.	Phase				
$s^{0.1}$	Oustaloup	0.0027	0.2530	0.0131	2.3867	0.0237	0.0455	1.5849	Stable
	Refined Oustaloup	0.0116	0.1664	0.0096	0.5042	0.0540	0.0813	1.7425	Stable
	Matsuda	0.0035	0.2407	0.0112	0.6956	0.0110	0.0120	1.8905	Stable
	Proposed	0.0047	0.1613	0.0059	0.1491	0.0043	0.0082	1.8239	Stable
$s^{0.2}$	Oustaloup	0.0043	0.5102	0.0400	4.7832	0.0265	0.0510	2.5119	Stable
	Refined Oustaloup	0.0261	0.3213	0.0288	0.9244	0.0832	0.1065	3.0744	Stable
	Matsuda	0.0075	0.4588	0.0313	1.3285	0.0286	0.0268	3.5911	Stable
	Curve Fitting	0.0048	0.3045	0.0257	0.4257	0.0208	0.0175	2.6816	Stable
	TF Estimate	0.0076	0.4074	0.0102	0.2617	0.0181	0.0172	3.4554	Stable
$s^{0.3}$	Oustaloup	0.0068	0.7757	0.0931	7.1992	0.0262	0.0450	3.9811	Stable
	Refined Oustaloup	0.0445	0.4539	0.0498	1.2581	0.0943	0.1045	5.5103	Stable
	Matsuda	0.0117	0.6336	0.0636	1.8410	0.0427	0.0363	6.8910	Stable
	Curve Fitting	0.0069	0.4015	0.0479	0.5379	0.0345	0.0233	4.4710	Stable
	TF Estimate	0.0072	0.6095	0.0125	0.3349	0.0300	0.0251	6.8073	Stable
$s^{0.4}$	Oustaloup	0.0091	1.0534	0.1952	9.6440	0.0272	0.0378	6.3096	Stable
	Refined Oustaloup	0.0683	0.5553	0.0825	1.5059	0.0942	0.0916	10.0820	Stable
	Matsuda	0.0158	0.7484	0.1104	2.1852	0.0511	0.0400	13.4464	Stable
	Curve Fitting	0.0111	0.4452	0.0675	0.6272	0.0433	0.0263	7.4974	Stable
	TF Estimate	0.0124	0.4709	0.0131	0.3715	0.0365	0.0296	13.9939	Stable
$s^{0.5}$	Oustaloup	0.0111	1.3481	0.3877	12.1269	0.0293	0.0319	10.0000	Stable
	Refined Oustaloup	0.0991	0.6192	0.1268	1.6761	0.0881	0.0761	18.9737	Stable
	Matsuda	0.0194	0.7919	0.1721	2.3275	0.0546	0.0398	26.9262	Stable
	Curve Fitting	0.0151	0.4306	0.0845	0.6768	0.0476	0.0270	12.7730	Stable
	TF Estimate	0.0124	0.4625	0.0124	0.4062	0.0393	0.0307	30.1873	Stable
$s^{0.6}$	Oustaloup	0.0121	1.6874	0.7450	14.6585	0.0314	0.0276	15.8489	Stable
	Refined Oustaloup	0.1393	0.6424	0.2471	1.7973	0.0794	0.0617	37.1951	Stable
	Matsuda	0.0216	0.7591	0.2433	2.2501	0.0548	0.0374	56.1234	Stable
	Curve Fitting	0.0162	0.3525	0.0949	0.6501	0.0488	0.0265	22.0811	Stable
	TF Estimate	0.0101	0.7294	0.0115	0.4299	0.0398	0.0294	69.0747	Stable
$s^{0.7}$	Oustaloup	0.0115	2.0579	1.3991	17.2459	0.0330	0.0256	25.1189	Stable
	Refined Oustaloup	0.1916	0.6247	0.6104	1.9275	0.0703	0.0496	77.7765	Stable
	Matsuda	0.0217	0.6520	0.3089	1.9530	0.0529	0.0340	124.8754	Stable
	Curve Fitting	0.0149	0.2691	0.0950	0.5576	0.0483	0.0245	38.8116	Stable
	TF Estimate	0.0125	0.6856	0.0105	0.4451	0.0393	0.0267	171.2372	Stable
$s^{0.8}$	Oustaloup	0.0088	2.4601	2.5821	19.8955	0.0338	0.0251	39.8107	Stable
	Refined Oustaloup	0.2596	0.5840	1.4006	2.0564	0.0618	0.0402	182.9633	Stable
	Matsuda	0.0187	0.4792	0.3369	1.4538	0.0499	0.0305	313.3221	Stable
	Curve Fitting	0.0109	0.2380	0.0809	0.4085	0.0468	0.0222	74.6020	Stable
	TF Estimate	0.0105	0.4763	0.0089	0.4404	0.0387	0.0235	484.8823	Stable
$s^{0.9}$	Oustaloup	0.0059	2.8934	4.6972	22.6126	0.0340	0.0205	63.0957	Stable
	Refined Oustaloup	0.3478	0.5762	3.0330	2.2127	0.0543	0.0330	573.8763	Stable
	Matsuda	0.0117	0.2553	0.2680	0.7866	0.0466	0.0273	1052.5	Stable
	Curve Fitting	0.0055	0.1535	0.0499	0.2176	0.0451	0.0249	150.9561	Stable
	TF Estimate	0.0057	0.2575	0.0059	0.3492	0.0401	0.0210	1855.5	Stable

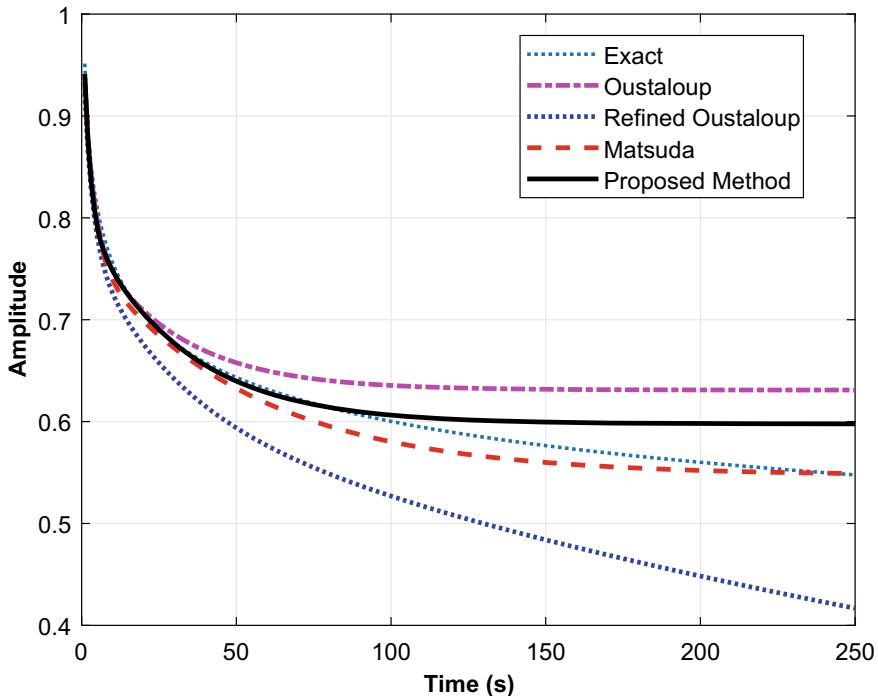


Fig. 3.10 Step response of $s^{0.1}$ using transfer function estimation algorithm for $N = 4$

the proposed approaches gives a better approximation than the other approaches. Furthermore, it can also be noted that all the approximated transfer functions using various techniques are stable with a finite value of H_∞ -norm.

3.7 Simulation Study

In this section, a simulation study will be conducted on five examples of fractional-order based controllers and systems. The selected systems are differentiator, integrator, PID controller, filter and higher-order transfer function. The obtained results from the proposed approaches will be compared with Oustaloup, refined Oustaloup and Matsuda approximations in terms of MAE's of frequency and step responses.

Furthermore, the comparison will also be done for the stability analysis of approximated transfer functions. In all the cases, the selected desired frequency range ω and the order of approximation N are chosen as $(10^{-3}, 10^3)$ and 5 respectively. Furthermore, for effective approximation, as mentioned earlier in Sect. 3.2, the order of s^α is limited to $[-1, 1]$ range as defined in Eq. (3.8).

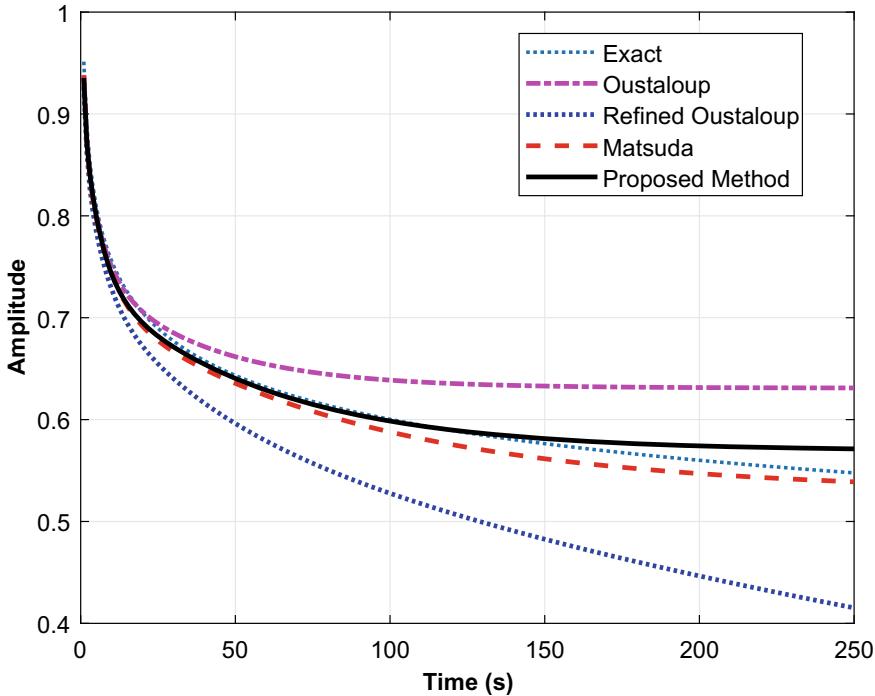


Fig. 3.11 Step response of $s^{0.1}$ using transfer function estimation algorithm for $N = 5$

3.7.1 Fractional-order Differentiator

This example studies the performance of the proposed approach on a fractional-order differentiator ($G(s)$) given in Eq. (3.21) and compares it with exact solution and other approximation methods. The approximated transfer function using the proposed curve fitting and transfer function estimation approaches are given in Eqs. (3.22) and (3.23) respectively.

$$G(s) = s^{0.26} \quad (3.21)$$

$$G(s) \approx \frac{7.147s^5 + 1836s^4 + 24930s^3 + 33930s^2 + 4811s + 29.74}{s^5 + 597.2s^4 + 16080s^3 + 38500s^2 + 10220s + 163.4} \quad (3.22)$$

$$G(s) \approx \frac{8.808s^5 + 7141s^4 + 584900s^3 + 4540000s^2 + 2255000s + 32640}{s^5 + 1585s^4 + 231800s^3 + 3409000s^2 + 3655000s + 149300} \quad (3.23)$$

Similarly, the approximated transfer function of fractional-order differentiator given in Sect. 3.7.1 using Oustaloup $G_O(s)$, refined Oustaloup $G_R(s)$ and Matsuda $G_M(s)$ approximation techniques are given in Eqs. (3.24), (3.25) and (3.26) respectively.

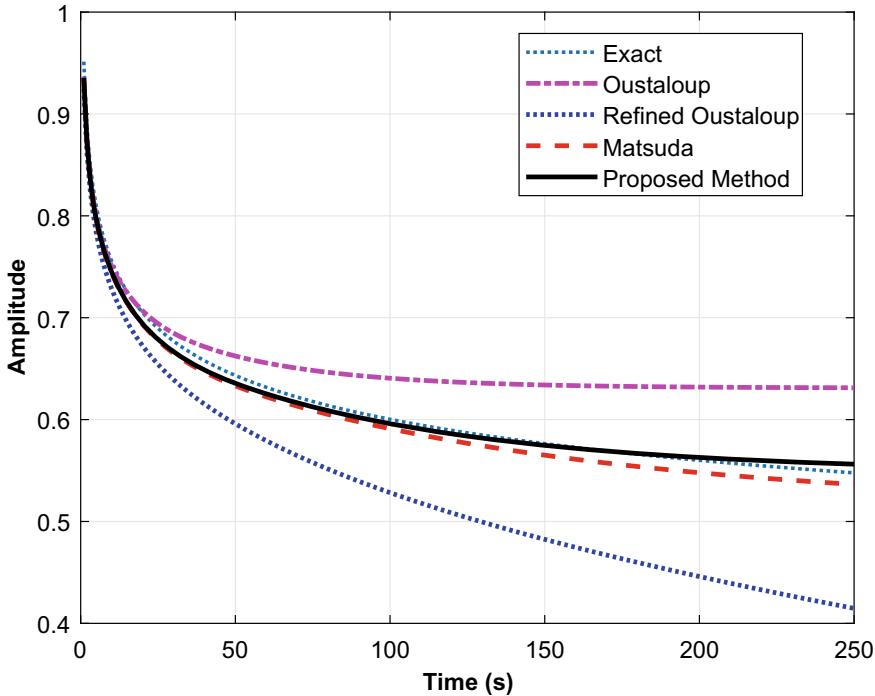


Fig. 3.12 Step response of $s^{0.1}$ using transfer function estimation algorithm for $N = 6$

$$G_O(s) \approx \frac{6.026s^5 + 1128s^4 + 1.253 \times 10^4 s^3 + 8750s^2 + 384s + 1}{s^5 + 384s^4 + 8750s^3 + 1.253 \times 10^4 s^2 + 1128s + 6.026} \quad (3.24)$$

$$\begin{aligned} & [7.923s^{13} + 1.029 \times 10^4 s^{12} + 1.664 \times 10^6 s^{11} + 1.832 \times 10^7 s^{10} \\ & + 1.279 \times 10^7 s^9 + 5.634 \times 10^5 s^8 + 1566s^7 + 0.2747s^6 \\ & + 3.039 \times 10^{-6}s^5 + 2.122 \times 10^{-12}s^4 + 9.347 \times 10^{-20}s^3 \\ & + 2.588 \times 10^{-28}s^2 + 4.252 \times 10^{-38}s] \end{aligned}$$

$$\begin{aligned} G_R(s) \approx & \frac{[s^{13} + 1885s^{12} + 5.853 \times 10^5 s^{11} + 1.315 \times 10^7 s^{10}]}{\\ & + 1.883 \times 10^7 s^9 + 1.705 \times 10^6 s^8 + 1.01 \times 10^4 s^7 + 5.758s^6 \\ & + 0.0008956s^5 + 1.864 \times 10^{-8}s^4 + 2.655 \times 10^{-14}s^3 \\ & + 2.398 \times 10^{-21}s^2 + 1.362 \times 10^{-29}s + 4.589 \times 10^{-39}] \end{aligned} \quad (3.25)$$

$$G_M(s) \approx \frac{8.283s^5 + 2347s^4 + 2.872 \times 10^4 s^3 + 1.982 \times 10^4 s^2 + 749.1s + 1}{s^5 + 749.1s^4 + 1.982 \times 10^4 s^3 + 2.872 \times 10^4 s^2 + 2347s + 8.283} \quad (3.26)$$

The Bode plots of the proposed approach in comparison to Oustaloup, refined Oustaloup and Matsuda are presented in Fig. 3.13 while the numerical assessment of

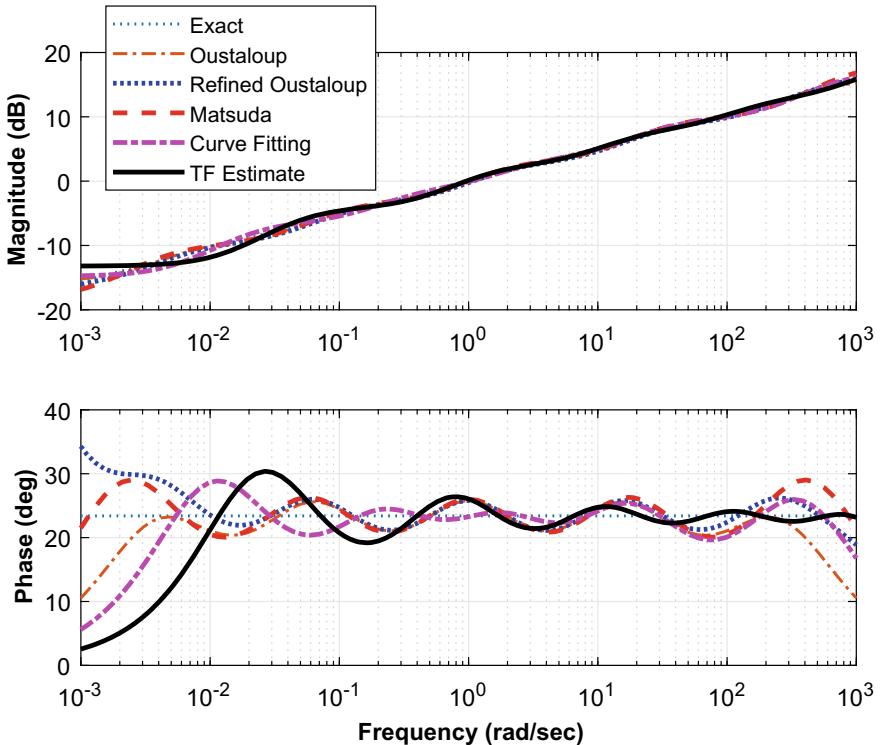


Fig. 3.13 Bode plot of fractional-order differentiator ($G(s)$) for different methods

the figure is given in Table 3.9. From the figure, it can be observed that the proposed approaches are more accurate within the desired frequency range than the other methods. Also, from the table, it can be seen that for the frequency range ω_1 the curve fitting approximation has the least MAE for both magnitude and phase of 0.0219 and 0.6364 respectively. On the other hand, for the frequency range ω_2 and ω_3 , the transfer function estimation has the least errors for both magnitude and phase.

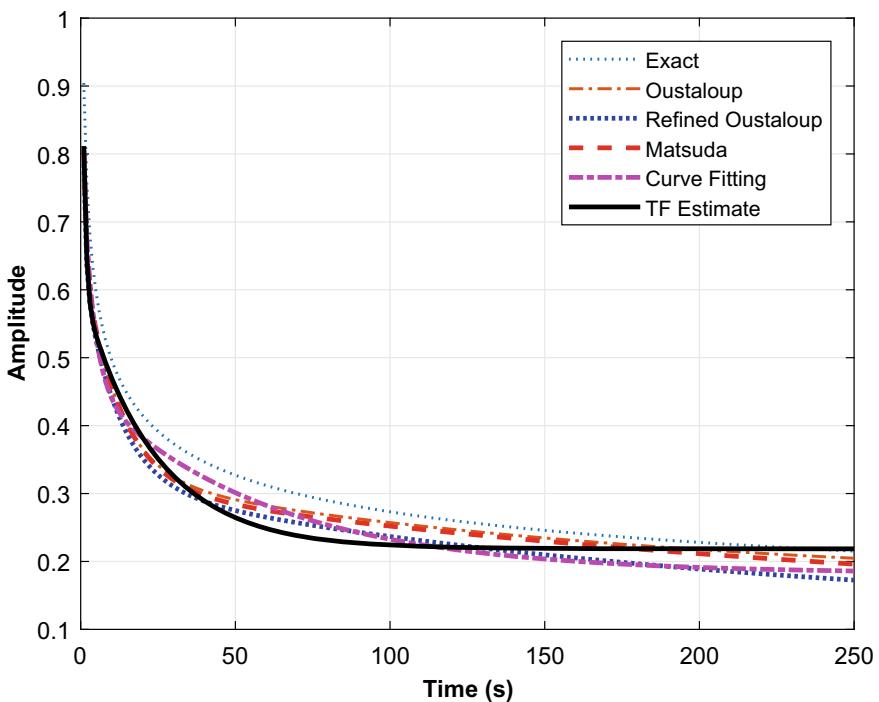
To further evaluate the performance of the proposed approach, the step responses of all the compared techniques is shown in Fig. 3.14 while numerical assessment of the figure is given in the same Table 3.10. In the figure, the exact time response of the system $G(s)$ will be obtained using Eq. (3.12) is given in Eq. (3.27).

$$g(t) = \frac{0.26 \times \Gamma(0.26)}{t^{0.26}} \quad (3.27)$$

From both figure and table, it can be observed that for the time period between 0 and 125 s, the Oustaloup has the least MAE of 0.0315 while for the time period from 0 to 250 s, the curve fitting approach has the least MAE of 0.0212. This is an indication that for longer time periods, the proposed approach gives a better approximation

Table 3.9 Comparison of MAE's for Bode response of $G(s)$

Technique	$\omega_1 \in (10^{-1}, 10)$		$\omega_2 \in (10^{-2}, 10^2)$		$\omega_3 \in (10^{-3}, 10^3)$	
	Magnitude	Phase	Magnitude	Phase	Magnitude	Phase
Oustaloup	0.0399	1.2785	0.0598	2.1321	0.1031	5.8981
Refined Oustaloup	0.0593	1.2832	0.0774	1.5645	0.1890	2.0613
Matsuda	0.0442	1.4149	0.0722	2.3276	0.4328	2.9122
Curve Fitting	0.0219	0.6364	0.0662	1.4370	0.2470	2.5384
TF Estimate	0.0360	1.3301	0.0377	0.6927	0.0472	0.3968

**Fig. 3.14** Step response of fractional-order differentiator ($G(s)$) for different methods

than the other approaches. Furthermore, the stability analysis of $G(s)$ for different methods given in table shows that all the approximation techniques are stable with a finite value of H_∞ -norm.

Table 3.10 Comparison of MAE's for step response of $G(s)$

Technique	$t_1 \in (0, 125)$	$t_2 \in (0, 250)$	H_∞ -norm	Stability
Oustaloup	0.0315	0.0213	6.0256	Stable
Refined Oustaloup	0.0493	0.0439	7.9227	Stable
Matsuda	0.0363	0.0266	8.2832	Stable
Curve Fitting	0.0368	0.0212	7.1467	Stable
TF Estimate	0.0515	0.0331	8.8083	Stable

3.7.2 Fractional-order Integrator

In this example, the proposed approaches are studied for a fractional-order integrator ($P(s)$) given in Eq. (3.28). Here, a similar comparison is made as to the differentiator example of Sect. 3.7.1. Therefore, the approximated transfer function model using the proposed curve fitting and transfer function estimation techniques are given in Eqs. (3.29) and (3.30) respectively.

$$P(s) = \frac{1}{s^{0.6}} \quad (3.28)$$

$$P(s) \approx \frac{s^5 + 2134s^4 + 209400s^3 + 1.6 \times 10^6s^2 + 1.3 \times 10^6s + 88840}{148.8s^5 + 46460s^4 + 929400s^3 + 1.8 \times 10^6s^2 + 373100s + 2353} \quad (3.29)$$

$$P(s) \approx \frac{s^5 + 5870s^4 + 3.2 \times 10^6s^3 + 2.9 \times 10^8s^2 + 3.7 \times 10^9s + 3.4 \times 10^9}{264.3s^5 + 3.2 \times 10^5s^4 + 6.0 \times 10^7s^3 + 1.7 \times 10^9s^2 + 5.2 \times 10^9s + 3.0 \times 10^8} \quad (3.30)$$

Similarly, the approximated transfer functions using Oustaloup $P_O(s)$, refined Oustaloup $P_R(s)$ and Matsuda $P_M(s)$ techniques are given in Eqs. (3.31), (3.32) and (3.33) respectively.

$$P_O(s) \approx \frac{s^5 + 614.2s^4 + 2.239 \times 10^4s^3 + 5.129e04s^2 + 7384s + 63.1}{63.1s^5 + 7384s^4 + 5.129 \times 10^4s^3 + 2.239e04s^2 + 614.2s + 1} \quad (3.31)$$

$$\begin{aligned} P_R(s) \approx & \left[s^{13} + 3392s^{12} + 1.728 \times 10^6s^{11} + 6.224 \times 10^7s^{10} \right. \\ & + 1.426 \times 10^8s^9 + 2.068 \times 10^7s^8 + 1.99 \times 10^5s^7 + 209.6s^6 \\ & + 0.0623s^5 + 2.129 \times 10^{-6}s^4 + 4.857 \times 10^{-12}s^3 \\ & + 7.017 \times 10^{-19}s^2 + 6.374 \times 10^{-27}s + 3.436 \times 10^{-36} \Big] \\ & \left[148.1s^{13} + 1.819 \times 10^5s^{12} + 1.938 \times 10^7s^{11} + 1.338 \times 10^8s^{10} \right. \\ & + 5.84 \times 10^7s^9 + 1.608 \times 10^6s^8 + 2795s^7 + 0.3065s^6 \\ & + 2.12 \times 10^{-6}s^5 + 9.255 \times 10^{-13}s^4 + 2.548 \times 10^{-20}s^3 \\ & \left. + 4.411 \times 10^{-29}s^2 + 4.532 \times 10^{-39}s \right] \end{aligned} \quad (3.32)$$

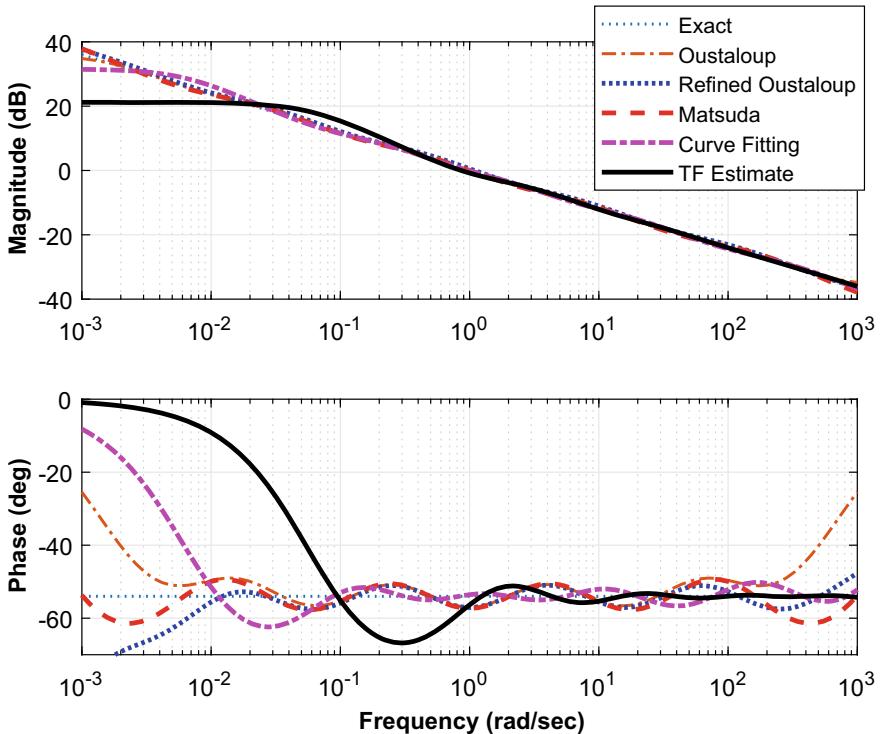


Fig. 3.15 Bode plots of fractional-order integrator ($P(s)$) for different methods

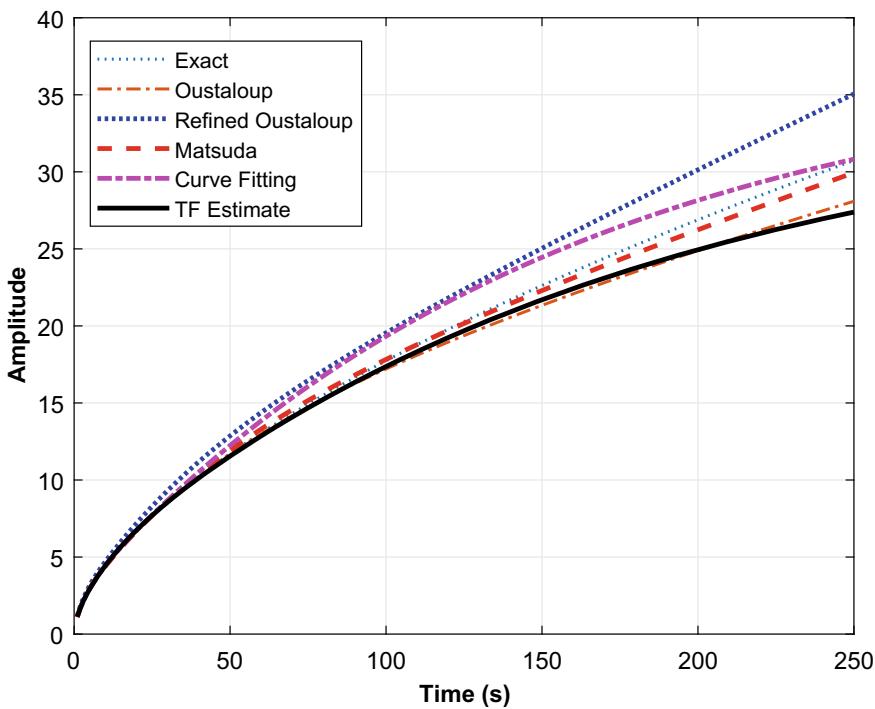
$$P_M(s) \approx \frac{s^5 + 1807s^4 + 8.141 \times 10^4 s^3 + 1.922 \times 10^5 s^2 + 2.565 \times 10^4 s + 156.4}{156.4s^5 + 2.565 \times 10^4 s^4 + 1.922 \times 10^5 s^3 + 8.141 \times 10^4 s^2 + 1807s + 1} \quad (3.33)$$

The frequency plots of the proposed approach in comparison to Oustaloup, Refined Oustaloup and Matsuda are presented in Fig. 3.15 while the numerical assessment of the figure is given in Table 3.11. From both figure and table, it can be observed that compared to the other three techniques the proposed methods are more accurate for the phase responses with the least error. However, for the magnitude response, the curve fitting algorithm, refined Oustaloup and Oustaloup approximations are better with MAE of 0.0282, 0.0112 and 0.0094 for ω_1 , ω_2 and ω_3 respectively.

In the time domain, the step response of all the compared approaches is shown in Fig. 3.16 while the numerical assessment is given in the same Table 3.12. It can be observed from both response and table, that for the longest time range $t \in (0, 250)$, the proposed approach has the least MAE of 0.1146 while for the shorter time range $t \in (0, 125)$, Matsuda has the least MAE of 0.1280. This indicated that the proposed approach achieves better approximation for longer time periods than the other approaches. This is in agreement with the case of fractional-order differentiator given

Table 3.11 Comparison of MAE's for Bode response of $P(s)$

Technique	$\omega_1 \in (10^{-1}, 10)$		$\omega_2 \in (10^{-2}, 10^2)$		$\omega_3 \in (10^{-3}, 10^3)$	
	Magnitude	Phase	Magnitude	Phase	Magnitude	Phase
Oustaloup	0.0311	1.6822	0.0117	3.3804	0.0094	14.0237
Refined Oustaloup	0.0353	1.6965	0.0112	2.0836	0.0153	2.7880
Matsuda	0.0369	1.8590	0.0126	3.1571	0.0177	4.3198
Curve Fitting	0.0282	0.8503	0.0529	1.3945	0.0270	1.3961
TF Estimate	0.2056	1.4566	0.0478	0.9195	0.0519	0.2173

**Fig. 3.16** Step responses of fractional-order integrator ($P(s)$) for different methods

in Sect. 3.7.1. Furthermore, the stability analysis of various methods given in the table shows that the approximation transfer functions of Oustaloup, Matsuda and proposed approach are stable with proposed technique has the least H_∞ -norm of 37.7568. From the table, it can also be seen that the approximated transfer function using refined Oustaloup is unstable.

Table 3.12 Comparison of MAE's for step response of $P(s)$

Technique	$t_1 \in (0, 125)$	$t_2 \in (0, 250)$	H_∞ -norm	Stability
Oustaloup	0.2531	1.0345	63.0957	Stable
Refined Oustaloup	1.2426	2.1651	inf	Unstable
Matsuda	0.1280	0.3347	156.4489	Stable
Curve Fitting	0.8510	0.1146	37.7568	Stable
TF Estimate	1.1916	5.1254	11.4671	Stable

3.7.3 Fractional-order PID Controller

In this example, the proposed approaches are studied for the $PI^\lambda D^\mu$ controller which consists of both fractional-order differentiator and integrator given in Sects. 3.7.1 and 3.7.2 respectively. The transfer function of $PI^\lambda D^\mu$ controller used for the study is given as follows:

$$C(s) = 5 + \frac{1}{s^{0.8}} + 2s^{0.5} \quad (3.34)$$

For this example, the approximation will be done in two stages. First, the fractional-order differentiator term ($s^{0.5}$) and the fractional-order integrator term ($1/s^{0.8}$) will be approximated using the approximations given in Sect. 3.6. Then, using the approximated transfer functions, the overall approximation of $C(s)$ is done by combining this integral and differential terms which will lead to the transfer function of $C(s)$ given in Eqs. (3.35) and (3.36) for curve fitting and transfer function estimation approaches respectively.

$$C(s) \approx \frac{\left[1.64 \times 10^5 s^{10} + 1.122 \times 10^8 s^9 + 2.207 \times 10^{10} s^8 + 9.906 \times 10^{11} s^7 \right.}{\left[1379 s^{10} + 2.47 \times 10^6 s^9 + 8.25 \times 10^8 s^8 + 6.329 \times 10^{10} s^7 \right]} \\ + 1.546 \times 10^{13} s^6 + 7.506 \times 10^{13} s^5 + 1.384 \times 10^{14} s^4 + 9.075 \times 10^{13} s^3 \\ + 2.321 \times 10^{13} s^2 + 1.953 \times 10^{12} s + 5.153 \times 10^{10} \\ \left. + 1.405 \times 10^{12} s^6 + 8.811 \times 10^{12} s^5 + 1.842 \times 10^{13} s^4 + 1.202 \times 10^{13} s^3 \right. \\ \left. + 2.418 \times 10^{12} s^2 + 1.009 \times 10^{11} s + 3.871 \times 10^8 \right] \quad (3.35)$$

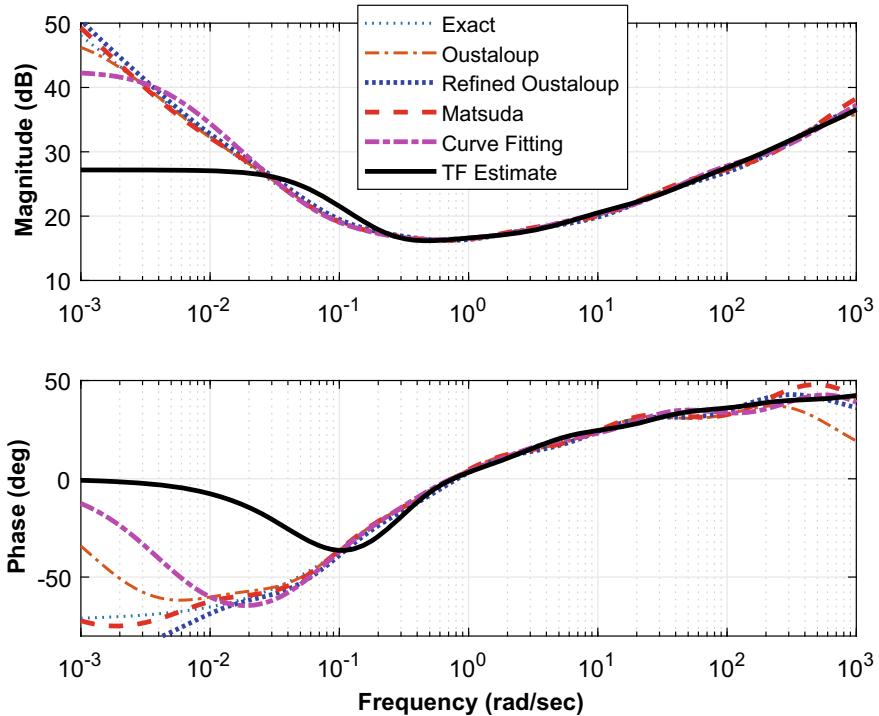


Fig. 3.17 Bode plots of fractional-order PID controller ($C(s)$) for different methods

$$C(s) \approx \frac{[5.598 \times 10^5 s^{10} + 1.431 \times 10^9 s^9 + 1.212 \times 10^{12} s^8 + 3.887 \times 10^{14} s^7 + 5.233 \times 10^{16} s^6 + 2.829 \times 10^{18} s^5 + 6.148 \times 10^{19} s^4 + 4.541 \times 10^{20} s^3 + 1.164 \times 10^{21} s^2 + 5.637 \times 10^{20} s + 8.381 \times 10^{19}]}{[2977 s^{10} + 1.606 \times 10^7 s^9 + 2.213 \times 10^{10} s^8 + 1.076 \times 10^{13} s^7 + 2.109 \times 10^{15} s^6 + 1.626 \times 10^{17} s^5 + 4.955 \times 10^{18} s^4 + 4.893 \times 10^{19} s^3 + 1.582 \times 10^{20} s^2 + 7.402 \times 10^{19} s + 3.667 \times 10^{18}]} \quad (3.36)$$

The frequency plot of the approximated $C(s)$ using the proposed approaches in comparison to Oustaloup, refined Oustaloup and Matsuda are presented in Fig. 3.17 while the numerical assessment of the figure is given in Table 3.13. From both figure and table, it can be observed that compared to the other three techniques the proposed methods are more accurate for both magnitude and phase response. This is true for all the desired frequency ranges ω_1 , ω_2 and ω_3 . This can be seen more clearly from the table.

For the magnitude response, the proposed approaches has the least MAE of 0.0805, 0.2932 and 0.3419 for ω_1 , ω_2 and ω_3 respectively. Similarly, for the phase response, the proposed approaches have the least MAE of 0.4521, 0.9308 and 0.2766 for ω_1 , ω_2

Table 3.13 Numerical analysis of $C(s)$ for various methods

Technique	$\omega_1 \in (10^{-1}, 10)$		$\omega_2 \in (10^{-2}, 10^2)$		$\omega_3 \in (10^{-3}, 10^3)$	
	Magnitude	Phase	Magnitude	Phase	Magnitude	Phase
Oustaloup	0.1700	0.9766	0.4032	2.5189	1.4721	10.6800
Refined Oustaloup	0.2688	1.5433	0.6408	1.8742	2.2084	2.5266
Matsuda	0.1916	1.1085	0.5167	2.4684	5.6817	4.0801
Curve Fitting	0.0805	0.4521	0.3908	1.3567	1.2892	1.5815
TF Estimate	0.3555	0.7057	0.2932	0.9308	0.3419	0.2766

Table 3.14 Stability analysis of $C(s)$ for various methods

Technique	H_∞ -norm	Stability
Oustaloup	256.2519	Stable
Refined Oustaloup	inf	Unstable
Matsuda	1.2269×10^3	Stable
Curve Fitting	133.1226	Stable
TF Estimate	187.5695	Stable

and ω_3 respectively. Furthermore, the stability analysis shows that the approximation transfer functions of Oustaloup, Matsuda and proposed approach are stable and that of refined Oustaloup is unstable. From the stability analysis given in Table 3.14, it can also be seen that the proposed curve fitting and transfer function estimation methods having the least H_∞ -norms of 133.1226 and 187.5695 respectively.

The result obtained here shows an improvement to the separate integrator and differentiator cases. This indicates that combining the two using the proposed approaches yields an overall better result. Hence, the overall improvement in system performance.

3.7.4 Fractional-order Filter

Apart from the fractional-order controllers, the proposed approaches can also be used to approximate the other fractional-order systems. Example of such is the fractional-order filter given in Eq. (3.37).

$$F(s) = \frac{1}{0.5s^{1.15} + 1} \quad (3.37)$$

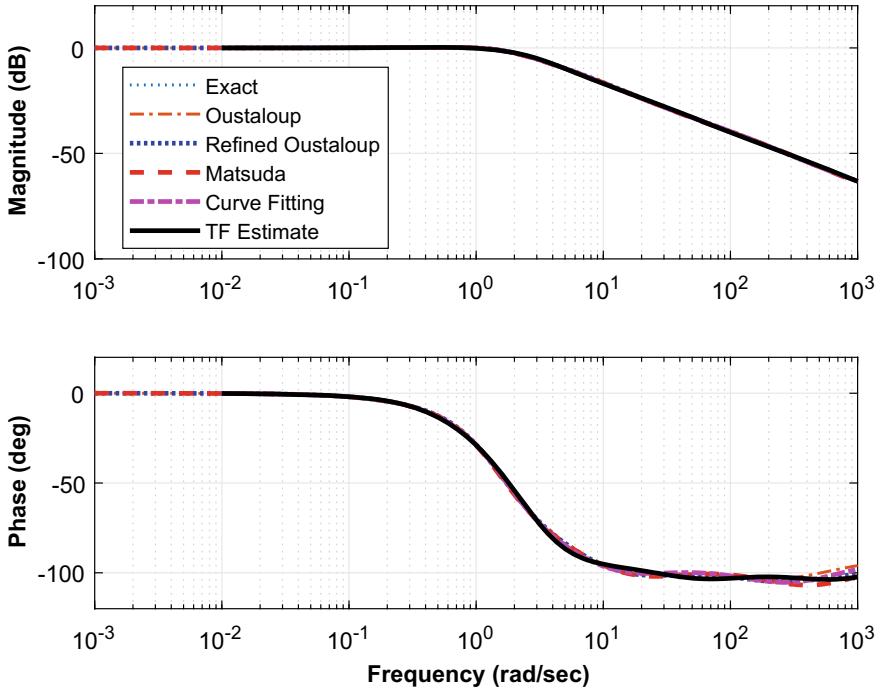


Fig. 3.18 Bode plots of fractional-order filter ($F(s)$) for different methods

Like in the previous case of the $\text{PI}^\lambda\text{D}^\mu$ controller, the approximation of $F(s)$ will be done in three stages. First, based on Eq. (3.8), the fractional-order derivative $s^{1.15}$ is divided into $s \times s^{0.15}$. Then $s^{0.15}$ will be approximated using the proposed approach. Finally, by substituting this in Eq. (3.37), the overall approximated transfer function of $F(s)$ is determined. Thus, the approximated transfer function using both the proposed approaches are given in Eqs. (3.38) and (3.39).

$$F(s) \approx \frac{s^5 + 387.7s^4 + 6767s^3 + 11190s^2 + 2004s + 19.27}{1.504s^6 + 355.3s^5 + 4579s^4 + 11780s^3 + 11810s^2 + 2008s + 19.27} \quad (3.38)$$

$$F(s) \approx \frac{s^5 + 938s^4 + 66610s^3 + 356600s^2 + 109400s + 1224}{1.661s^6 + 1039s^5 + 51610s^4 + 247200s^3 + 391600s^2 + 109700s + 1224} \quad (3.39)$$

The frequency plots of the approximated filter using the proposed approaches in comparison to Oustaloup, refined Oustaloup and Matsuda are presented in Fig. 3.18 while the numerical comparison of the figure is given in Table 3.15. From the results, it can be seen that, just as in the case of a $\text{PI}^\lambda\text{D}^\mu$ controller, the proposed methods are more accurate for both magnitude and phase response compared to the other three techniques. Moreover, Oustaloup approximation is also performed better for the magnitude response of $F(s)$.

Table 3.15 Numerical analysis of $F(s)$ for various methods

Technique	$\omega_1 \in (10^{-1}, 10)$		$\omega_2 \in (10^{-2}, 10^2)$		$\omega_3 \in (10^{-3}, 10^3)$	
	Magnitude	Phase	Magnitude	Phase	Magnitude	Phase
Oustaloup	0.0061	0.7071	0.0010	1.2853	0.0001	1.3831
Refined Oustaloup	0.0067	0.7490	0.0011	0.9792	0.0001	1.2997
Matsuda	0.0066	0.7896	0.0012	1.4194	0.0002	1.7922
Curve Fitting	0.0025	0.3451	0.0007	1.2658	0.0001	1.2313
TF Estimate	0.0058	0.6640	0.0008	0.6186	0.0001	0.4495

Table 3.16 Stability analysis of $F(s)$ for various methods

Technique	H_2 -norm	H_∞ -norm	Stability
Oustaloup	0.9773	1.0273	Stable
Refined Oustaloup	0.9841	1.0276	Stable
Matsuda	0.9772	1.0275	Stable
Curve Fitting	0.9784	1.0267	Stable
TF Estimate	0.9796	1.0289	Stable

On the other hand, the stability analysis is given in Table 3.16 also shows that all the compared techniques are better and stable with H_2 and H_∞ -norms of around 0.98 and 1.02 respectively.

3.7.5 Fractional-order Transfer Function

To further study the performance of the proposed approach, a fractional-order transfer function (FOTF) given in Eq. (3.40) is considered [12, 26]. The FOTF in polynomial form with fractional powers is given as

$$R(s) = \frac{s + 1}{10s^{3.2} + 185s^{2.5} + 288s^{0.7} + 1} \quad (3.40)$$

Like in the previous two cases of $P\!I^\lambda D^\mu$ and $F(s)$, the approximation will be done in three stages. First, based on Eq. (3.8), the fractional-order derivatives $s^{3.2}$ and $s^{2.5}$ will be divided as $s^3 \times s^{0.2}$ and $s^2 \times s^{0.5}$ respectively. Then $s^{0.2}$, $s^{0.5}$ and $s^{0.7}$ will be approximated using the approximations given in Sect. 3.6. Finally, by substituting all these in Eq. (3.40), the overall approximated transfer function of $R(s)$ is determined. Thus, the approximated transfer functions of $R(s)$ using both the proposed approaches are given in Eqs. (3.41) and (3.42).

$$\begin{aligned}
& [s^{16} + 5206s^{15} + 7.573 \times 10^6 s^{14} + 3.565 \times 10^9 s^{13} \\
& + 5.807 \times 10^{11} s^{12} + 3.69 \times 10^{13} s^{11} + 9.463 \times 10^{14} s^{10} \\
& + 1.085 \times 10^{16} s^9 + 5.936 \times 10^{16} s^8 + 1.626 \times 10^{17} s^7 \\
& + 2.361 \times 10^{17} s^6 + 1.854 \times 10^{17} s^5 + 7.647 \times 10^{16} s^4 \\
& + 1.53 \times 10^{16} s^3 + 1.359 \times 10^{15} s^2 + 4.618 \times 10^{13} s \\
R(s) \approx & \frac{+ 4.001 \times 10^{11}}{[44.31s^{18} + 2.31 \times 10^5 s^{17} + 3.299 \times 10^8 s^{16} + 1.355 \times 10^{11} s^{15}} \quad (3.41) \\
& + 2.312 \times 10^{13} s^{14} + 1.738 \times 10^{15} s^{13} + 5.26 \times 10^{16} s^{12} \\
& + 7.555 \times 10^{17} s^{11} + 5.299 \times 10^{18} s^{10} + 1.788 \times 10^{19} s^9 \\
& + 3.458 \times 10^{19} s^8 + 4.593 \times 10^{19} s^7 + 4.372 \times 10^{19} s^6 \\
& + 2.484 \times 10^{19} s^5 + 6.997 \times 10^{18} s^4 + 8.796 \times 10^{17} s^3 \\
& + 3.93 \times 10^{16} s^2 + 5.396 \times 10^{14} s + 2.125 \times 10^{12}] \\
& [s^{16} + 14170s^{15} + 5.995 \times 10^7 s^{14} + 9.665 \times 10^{10} s^{13} \\
& + 7.069 \times 10^{13} s^{12} + 2.522 \times 10^{16} s^{11} + 4.501 \times 10^{18} s^{10} \\
& + 4.14 \times 10^{20} s^9 + 1.942 \times 10^{22} s^8 + 4.43 \times 10^{23} s^7 \\
& + 4.868 \times 10^{24} s^6 + 2.555 \times 10^{25} s^5 + 6.026 \times 10^{25} s^4 \\
& + 6.579 \times 10^{25} s^3 + 3.278 \times 10^{25} s^2 + 6.25 \times 10^{24} s \\
R(s) \approx & \frac{+ 1.166 \times 10^{23}}{[51.32s^{18} + 7.189 \times 10^5 s^{17} + 2.933 \times 10^9 s^{16} + 4.274 \times 10^{12} s^{15}} \quad (3.42) \\
& + 2.914 \times 10^{15} s^{14} + 1.015 \times 10^{18} s^{13} + 1.835 \times 10^{20} s^{12} \\
& + 1.785 \times 10^{22} s^{11} + 9.375 \times 10^{23} s^{10} + 2.536 \times 10^{25} s^9 \\
& + 3.471 \times 10^{26} s^8 + 2.294 \times 10^{27} s^7 + 6.909 \times 10^{27} s^6 \\
& + 1.133 \times 10^{28} s^5 + 1.343 \times 10^{28} s^4 + 8.991 \times 10^{27} s^3 \\
& + 2.554 \times 10^{27} s^2 + 1.728 \times 10^{26} s + 2.571 \times 10^{24}]
\end{aligned}$$

The frequency plots of the approximated $R(s)$ using the proposed approaches in comparison to Oustaloup, refined Oustaloup and Matsuda are presented in Fig. 3.19 while the numerical comparison of the figure is given in Table 3.17. From the results, it can be seen that for phase response the curve fitting approach has the least MAE of 1.2067 for ω_1 while for ω_2 and ω_3 transfer function estimation algorithm has the least MAE of 0.9718 and 0.3777 respectively. For magnitude response, all the compared techniques have provided the best approximation with almost zero error. Furthermore, the stability analysis from the Table 3.18 also shows that all the approximated transfer functions are stable with the proposed approaches has the least H_2 and H_∞ -norms.

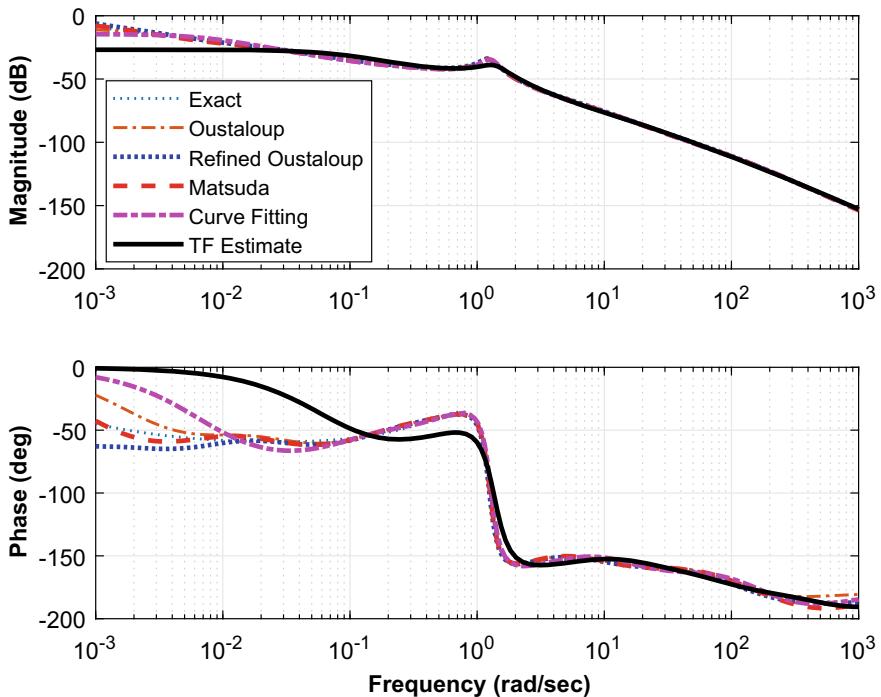


Fig. 3.19 Bode plots of FOTF ($R(s)$) for different methods

Table 3.17 Numerical analysis of $R(s)$ for various methods

Technique	$\omega_1 \in (10^{-1}, 10)$		$\omega_2 \in (10^{-2}, 10^2)$		$\omega_3 \in (10^{-3}, 10^3)$	
	Magnitude	Phase	Magnitude	Phase	Magnitude	Phase
Oustaloup	0.0001	1.5838	0.0000	2.5255	0.0000	5.1760
Refined Oustaloup	0.0002	1.7686	0.0000	1.5011	0.0001	1.9960
Matsuda	0.0001	1.7556	0.0000	2.6874	0.0000	2.6449
Curve Fitting	0.0001	1.2067	0.0002	2.4408	0.0001	2.6240
TF Estimate	0.0013	3.4518	0.0004	0.9718	0.0003	0.3777

Table 3.18 Stability analysis of $R(s)$ for various methods

Technique	H ₂ -norm	H _∞ -norm	Stability
Oustaloup	0.0141	0.3042	Stable
Refined Oustaloup	0.0225	1.2395	Stable
Matsuda	0.0164	0.5885	Stable
Curve Fitting	0.0138	0.1883	Stable
TF Estimate	0.0103	0.0453	Stable

3.8 Summary

In this chapter, the curve fitting based approximation has been developed for the practical realization of fractional-order parameters using SK least-square iterative method with Levy's cost function. Furthermore, the approximation table for fractional-order differentiator has also been developed. The results from the simulation study on various fractional-order based systems and controllers show that the proposed approximation technique performed better compared to Oustaloup, refined Oustaloup and Matsuda approximations. This is true for both time and frequency-domain analysis using Step and Bode responses respectively. Furthermore, the numerical analysis in terms of MAE shows that the proposed approach shows better approximation for high-frequency range and longer time periods. The stability analysis in terms of H-norms also confirms that the proposed approach is better and stable.

References

1. Krishna, B.T.: Studies on fractional order differentiators and integrators: a survey. *Signal Process.* **91**(3), 386–426 (2011)
2. Vinagre, B.M., Podlubny, I., Hernandez, A., Feliu, V.: Some approximations of fractional order operators used in control theory and applications. *Fract. Calc. Appl. Anal.* **3**(3), 231–248 (2000)
3. Li, Z., Liu, L., Dehghan, S., Chen, Y., Xue, D.: A review and evaluation of numerical tools for fractional calculus and fractional order controls. *Int. J. Control.* **90**(6), 1165–1181 (2017)
4. Djouambi, A., Charef, A., BesançOn, A.: Optimal approximation, simulation and analog realization of the fundamental fractional-order transfer function. *Int. J. Appl. Math. Comput. Sci.* **17**(4), 455–462 (2007)
5. Deniz, F.N., Alagoz, B.B., Tan, N., Atherton, D.P.: An integer order approximation method based on stability boundary locus for fractional-order derivative/integrator operators. *ISA Trans.* **62**, 154–163 (2016)
6. Merrikh-Bayat, F.: Rules for selecting the parameters of Oustaloup recursive approximation for the simulation of linear feedback systems containing $\text{PI}^\lambda \text{D}^\mu$ controller. *Commun. Nonlinear Sci. Numer. Simul.* **17**(4), 1852–1861 (2012)
7. Valério, D., Da Costa, J.S.: Ninteger: a non-integer control toolbox for MatLab. In: Proceedings of Fractional Differentiation and Its Applications, Bordeaux, July 2004
8. Tepljakov, A., Petlenkov, E., Belikov, J.: Application of Newton's method to analog and digital realization of fractional-order controllers. *Int. J. Microelectron. Comput. Sci.* **3**(2), 45–52 (2012)
9. Valério, D., Trujillo, J.J., Rivero, M., Machado, J.T., Baleanu, D.: Fractional calculus: a survey of useful formulas. *Eur. Phys. J. Spec. Top.* **222**(8), 1827–1846 (2013)
10. Petráš, I.: Fractional-Order Nonlinear Systems: Modeling, Analysis and Simulation. Springer Science & Business Media (2011)
11. Caponetto, R.: Fractional Order Systems: Modeling and Control Applications. World Scientific (2010)
12. Monje, C.A., Chen, Y., Vinagre, B.M., Xue, D., Feliu-Batlle, V.: Fractional-Order Systems and Controls: Fundamentals and Applications. Springer Science & Business Media (2010)
13. Xue, D., Zhao, C., Chen, Y.: A modified approximation method of fractional order system. In: Proceedings of the 2006 IEEE International Conference on Mechatronics and Automation, Luoyang, Henan, China, 25–28 June 2006
14. Das, S.: Functional Fractional Calculus. Springer Science & Business Media (2011)

15. Du, B., Wei, Y., Liang, S., Wang, Y.: Rational approximation of fractional order systems by vector fitting method. *Int. J. Control Autom. Syst.* **15**(1), 186–195 (2017)
16. Oustaloup, A., Levron, F., Mathieu, B., Nanot, F.M.: Frequency-band complex noninteger differentiator: characterization and synthesis. *IEEE Trans. Circuits Syst. I-Fundam. Theor. Appl.* **47**(1), 25–39 (2000)
17. Meng, L., Xue, D.: A new approximation algorithm of fractional order system models based optimization. *J. Dyn. Syst. Meas. Control* **134**(4), 044504 (2012)
18. Wei, Y., Gao, Q., Peng, C., Wang, Y.: A rational approximate method to fractional order systems. *Int. J. Control Autom. Syst.* **12**(6), 1180–1186 (2014)
19. Krajewski, W., Viaro, U.: A method for the integer-order approximation of fractional-order systems. *J. Frankl. Inst.-Eng. Appl. Math.* **351**(1), 555–564 (2014)
20. Atherton, D.P., Tan, N., Yüce, A.: Methods for computing the time response of fractional-order systems. *IET Control Theory Appl.* **9**(6), 817–830 (2014)
21. Liang, S., Peng, C., Liao, Z., Wang, Y.: State space approximation for general fractional order dynamic systems. *Int. J. Syst. Sci.* **45**(10), 2203–2212 (2014)
22. Sheng, H., Chen, Y., Qiu, T.: Fractional Processes and Fractional-Order Signal Processing: Techniques and Applications. Springer Science & Business Media (2011)
23. Yüce, A., Deniz, F.N., Tan, N.: A new integer order approximation table for fractional order derivative operators. *IFAC-PapersOnLine* **50**(1), 9736–9741 (2017)
24. Mitkowski, W., Oprzedkiewicz, K.: An estimation of accuracy of Charef approximation. In: Domek, S., Dworak, P. (eds.) Theoretical Developments and Applications of Non-integer Order Systems: 7th Conference on Non-integer Order Calculus and Its Applications. Lecture Notes in Electrical Engineering, vol. 357, pp. 71–80. Springer (2016)
25. Oprzedkiewicz, K.: Approximation method for a fractional order transfer function with zero and pole. *Arch. Control Sci.* **24**(4), 447–463 (2014)
26. Khanra, M., Pal, J., Biswas, K.: Rational approximation and analog realization of fractional order transfer function with multiple fractional powered terms. *Asian J. Control* **15**(3), 723–735 (2013)
27. Poinot, T., Trigeassou, J.C.: A method for modelling and simulation of fractional systems. *Signal Process.* **83**(11), 2319–2333 (2003)
28. Krajewski, W., Viaro, U.: On the rational approximation of fractional order systems. In: 2011 16th International Conference on Methods and Models in Automation and Robotics, Miedzyzdroje, Poland, 22–25 Aug 2011
29. Shi, G.: On the nonconvergence of the vector fitting algorithm. *IEEE Trans. Circuits Syst. II-Express Briefs* **63**(8), 718–722 (2016)
30. Ozdemir, A.A., Gumussoy, S.: Transfer function estimation in system identification toolbox via vector fitting. *IFAC-PapersOnLine* **50**(1), 6232–6237 (2017)
31. Bingi, K., Ibrahim, R., Karsiti, M.N., Hassan, S.M., Harindran, V.R.: Frequency response based curve fitting approximation of fractional-order PID controllers. *Int. J. Appl. Math. Comput. Sci.* **29**(2), 311–326 (2019)

Part II

**Scilab Based Toolbox for Fractional-order
Systems and PID Controllers**

Chapter 4

Scilab Based Toolbox for Fractional-order Systems and PID Controllers



4.1 Introduction

Fractional-order systems have been extensively used in many engineering applications [1, 2]. This is due to more accurate modeling and controlling possibilities they offer over integer-order systems [3, 4]. Fractional-order systems are useful in studying the dynamical behavior of the systems in the area of physics, biology and electrical engineering. They are also used to study chaotic systems and many others. Moreover, fractional-order PID (FOPID) controllers which are derived based on fractional-ordering of the parameters, allow for robust and stable control performance [5–8]. This is because they are less sensitive to parameter changes [9]. The controllers can also easily attain the iso-damping; a desirable property referring to a state where the open-loop phase Bode plot of the system is flat [10].

A key issue with the implementation of fractional-order systems and controllers is the numerical tools. Researchers have developed few toolboxes such as Commande Robuste d'Ordre Non Entier (CRONE) [11–14], Non-integer (NINTEGER) [15, 16], Fractional Order Modeling and Control (FOMCON) [17–19], FOPIID [20], fractional-order transfer function (@fotf) [21], fractional state space tool (FSST) [22, 23], Sysquake FOPIID [24–26], Discrete FOPIID [4, 27], fractional integration tool (FIT) [28] for fractional order modeling and control applications. A summary of these toolboxes explaining the features and limitations is given in Table 4.1.

The CRONE toolbox was developed by CRONE research team and is dedicated to the application of fractional-order derivatives in science and engineering. The toolbox allows for mathematical modeling, system identification and CRONE control design [11, 12, 14, 29]. On the other hand, the NINTEGER toolbox developed by Valério presents the implementation of non-integer PID controllers and CRONE controllers [15]. The toolbox also allows for the approximation of non-integer order derivatives, functions for model development and frequency response plots. Unlike the toolboxes mentioned earlier, the FOMCON toolbox incorporates advanced features such as system identification, fractional-order PID control, and real-time implementation. The toolbox is an extension of @fotf toolbox developed in [21] for fractional-order system identification and PID controller design. Similarly, the authors in [20] developed a fractional-order PID (FOPIID) toolbox for designing and tuning the controller using

Table 4.1 Summary of numerical toolboxes for fractional-order systems and PID controller

Name	Features	Software	Syntax ^a	Download ^b	Reference
CRONE	Non-integer differentiation, differential equations, system identifications, inverse Laplace transform and CRONE control	Matlab/ Simulink	–	✓	[29, 30]
NINTEGER	Fractional-order approximation techniques, system identification function and analysis, GUI for controller design	Matlab/ Simulink	nipid()	✓	[15]
FOPID	FOPID design, Approximation techniques, Step, Nyquist, Bode, Nichols plots	Matlab	–	✗	[20]
@fotf	Overload functions were developed for fractional transfer function, time and frequency domain, stability analysis	Matlab	fotf()	✓	[21]
FSST	Simulink blocks for stochastic and non-stochastic fractional-order systems and fractional Kalman filter.	Simulink	–	✓	[22, 23]
FOMCON	System identification in both time and frequency domains, FOPID design, tuning and implementation, Step, Bode, Nyquist and Nichols plots	Matlab/ Simulink	fpid()	✓	[17, 31]
Sysquake FOPID	Integrative tool for FOPID design and analysis	Sysquake	GUI	✗	[24–26]
DFOC	Discrete version of FOPID controller	Matlab	DFOC()	✓	[4, 27]
FIT	Riemann-Liouville type of differentiation and integration	C++, Matlab	–	✗	[28]

^aThe ‘Syntax’ column denotes the MATLAB command for FOPID controller

^bThe ‘Download’ column denotes whether the toolbox is publicly available for download either from MATLAB file exchange or other websites/forums

the steepest descent method. On the other hand, a MATrix LABoratory (MATLAB) script defining the discrete form of FOPID controller is given in [4]. A common feature of these toolboxes is that they are based on MATLAB/Simulink software.

Apart from these toolboxes, researchers also developed MATLAB scripts for several fractional-order based functions. These functions include fractional-order definitions, differentiator, integrator, inverse Laplace transforms, root locus and approximation techniques. The list of these function retrieved from file exchange central of MATLAB with the syntax is shown in Table 4.2.

MATLAB is a powerful software that provides many numerical computations. However, despite many advantages, the software is expensive and generally have limited licenses in terms of time and number of installations [32]. On the other hand,

Table 4.2 List of fractional-order functions retrieved from MATLAB central

Category	Method	Script/Function name
Fractional-order Definitions	Grünwald-Letnikov	fderiv() glfdiff()
	Mittag-Leffler	mlf()
Fractional-order Differentiator	Fourier series	fourier_diffint()
	Power series	dfod1(), dfod2()
Approximations Techniques	Refined Oustaloup	ora_foc()
	Impulse response	irid_fod()
Others	Inverse Laplace	invlap()
	Root locus	forlocus()

the Scientific laboratory (Scilab) is extremely effective and available as a free open-source software package for scientific computations. It includes hundreds of general and specialized functions for numerical computations, 2D and 3D visualization and data analysis [33, 34]. The software is organized in libraries called toolboxes, which cover areas such as simulation, optimization, control systems, and signal processing. It also provides its own dynamic systems modeler and simulator called Xcos. Recently, researchers have applied Scilab to solve ordinary and partial differential equations [35]. Furthermore, it has also been used for digital image processing [36], control and optimization [37–40]. Thus, there is a scope to develop an open-source toolbox in this open-source software for fractional-order systems and PID controllers.

4.2 Fractional-order Differintegral Operator

The fractional-order differintegral operator; a combined differentiator/integrator, \mathcal{D}_t^α of order $\alpha \in \mathbb{R}$ that generalizes the notations for integrals ($\alpha < 0$) and derivatives ($\alpha > 0$) is defined as

$$\mathcal{D}_t^\alpha f(t) = \begin{cases} \int_0^t f(\tau) d\tau^\alpha & \alpha < 0, \\ f(t) & \alpha = 0, \\ \frac{d^\alpha f(t)}{dt^\alpha} & \alpha > 0. \end{cases} \quad (4.1)$$

The fractional-order differentiator in Eq. (4.1) can be defined in many forms such as Grünwald-Letnikov (GL), Riemann-Liouville (RL), Caputo etc. Among these forms, GL definition is the most straightforward and widely used and is defined as follows:

$$\mathcal{D}_t^\alpha f(t) \approx \frac{1}{h^\alpha} \sum_{j=0}^{t/h} \omega_j^{(\alpha)} f(t - jh) \quad (4.2)$$

where h is the step size, ω_j are the coefficients which can be calculated as follows:

$$\omega_0^{(\alpha)} = 1; \quad \omega_j^{(\alpha)} = \left(1 - \frac{\alpha + 1}{j}\right) \omega_{j-1}^{(\alpha)}, \quad j = 1, 2, 3, \dots \quad (4.3)$$

Based on the above definition, the α th derivative of $f(t)$ can be evaluated in Scilab using the following function with syntax `df=grunwald(f,t,alpha,h)`.

```
function df=grunwald(f,t,alpha,h)
df(1)=0; w=1; f=f';
for j=2:length(t)
    w(j)=w(j-1)*(1-((alpha+1)/(j-1)));
end
w=w';
for i=2:length(t)
    df(i)=w(1:i)*[f(i:-1:1)];
end
df=df/h^alpha;
endfunction
```

Also, the Laplace transform of fractional derivative in Eq. (4.2) at zero initial conditions is given as

$$\mathcal{L}\{\mathcal{D}_t^\alpha f(t); s\} = s^\alpha F(s) \quad (4.4)$$

where $F(s)$ is the Laplace transform of $f(t)$.

4.3 Approximation of Fractional-order Operator

The fractional-order operator s^α in Eq. (4.4) can be approximated using any of the approximation technique such as Oustaloup, refined Oustaloup, Matsuda, Curve Fitting, Transfer Function Estimation, finite impulse, and infinite impulse responses. The techniques will be briefly explained subsequently.

4.3.1 Oustaloup Approximation

The Oustaloup approximation of s^α in the desired frequency range (ω_l, ω_h) is defined as:

$$s^\alpha \approx \omega_h^\alpha \prod_{k=1}^N \frac{s + \omega'_k}{s + \omega_k}, \quad 0 < \alpha < 1 \quad (4.5)$$

where

- α is the order of fractional-order derivative,
- N is the order of approximation,
- (ω_l, ω_h) is the frequency range of interest and
- ω'_k and ω_k are the zeros and poles.

The ω'_k and ω_k of Eq. (4.5) are computed as:

$$\omega'_k = \omega_l \left(\frac{\omega_h}{\omega_l} \right)^{\frac{2k-1-\alpha}{2N}} \quad (4.6)$$

$$\omega_k = \omega_l \left(\frac{\omega_h}{\omega_l} \right)^{\frac{2k-1+\alpha}{2N}} \quad (4.7)$$

Therefore, the function representing the Oustaloup approximation with syntax `G=oustaLoup(alpha,N,wl,wh)` can be written in Scilab as follows:

```
function G=oustaLoup(alpha,N,wl,wh)
k=1:N; wu=sqrt(wh/wl);
wkz=wl*wu.^((2*k-1-alpha)/N);
wkp=wl*wu.^((2*k-1+alpha)/N);
G=zpk(-wkz,-wkp,wh^alpha,"c");
G=zpk2tf(G);
endfunction
```

4.3.2 Refined Oustaloup Approximation

The refined Oustaloup approximation of s^α in the desired frequency range (ω_l, ω_h) is defined as:

$$s^\alpha \approx \left(\frac{d\omega_h}{b} \right)^\alpha \left(\frac{ds^2 + b\omega_h s}{d(1 - \alpha)s^2 + b\omega_h s + d\alpha} \right) \prod_{k=-N}^N \frac{s + \omega'_k}{s + \omega_k}, \quad 0 < \alpha < 1 \quad (4.8)$$

where ω'_k and ω_k are the respective zeros and poles defined in Eqs. (4.6) and (4.7) respectively.

In order to achieve good approximation, constants b and d in Eq. (4.8) are selected as 10 and 9 respectively. Thus, in Scilab, the function defining this approximation with syntax `G=refOustaloup(alpha,N,wl,wh)` is as follows:

```
function G=refOustaloup(alpha,N,wl,wh)
b=10; d=9; k=-N:N;
wu=sqrt(wh/wl);
K=(d*wh/b)^alpha;
wkz=wl*wu.^((2*k-1-alpha)/N);
wkp=wl*wu.^((2*k-1+alpha)/N);
G1=zpk(-wkz',-wkp',K,"c");
```

```

exec('tf.sci'); s=tf('s');
G2=syslin('c',((d*s^2)+(b*wh*s))/((d*(1-alpha)*s^2)...
+ (b*wh*s)+(d*alpha)));
G=G1*G2;
G=zpk2tf(G);
endfunction

```

In the above function, the `tf.sci` defining the transfer function variable ‘*s*’ can be written as follows:

```

function s=tf(s)
s=poly(0,'s');
s=syslin('c',s/((0*s)+1));
disp('Continuous-time transfer function.');
endfunction

```

4.3.3 Matsuda Approximation

The Matsuda approximation of s^α is achieved in two steps. First, a rational model of s^α is obtained using continued fraction expansions method. Then, fitting of the original function in a desired frequency points $\omega_0, \omega_1, \dots, \omega_N$ is performed. Thus, the approximated transfer function of s^α is defined as:

$$s^\alpha \approx d_0(\omega_0) + \frac{s - \omega_0}{d_1(\omega_1) + \frac{s - \omega_1}{d_2(\omega_2) + \frac{s - \omega_2}{\dots}}} \quad (4.9)$$

where

$$d_0(\omega_0) = |(j\omega_0)^\alpha| \quad (4.10)$$

$$d_{k+1}(\omega) = \frac{\omega - \omega_k}{d_k(\omega) - d_k(\omega_k)}, \quad k = 0, 1, 2, \dots, N \quad (4.11)$$

where N is the sum of the total number of zeros and poles which is known as the order of approximation. Here, N should be an even number, else the approximation results in an improper transfer function.

Thus, based on the above algorithm, the approximation can be written with also in Scilab with syntax `G=matsuda(alpha,N,wl,wh)` as following function.

```

function G=matsuda(alpha,N,wl,wh)
w=logspace(log10(wl),log10(wh),N+1);
k=20*log10(abs(w.^alpha)); K=abs(10.^(k/20));
temp1=[];
for j=0:length(w)-1
    temp(0+1,j+1)=K(j+1);
    for i=1:j

```

```

temp(i+1,j+1)=(w(j+1)-w(i))/(temp(i,j+1)-temp(i,i));
end
temp1=[temp1 temp(j+1,j+1)];
end
Gm=temp1(length(temp1));
for i=length(temp1)-2:-1:0
Gm=temp1(i+1)+(zpk([1 w(i+1)],1,1,"c")/Gm);
end
G=zpk2tf(Gm);
endfunction

```

4.3.4 Curve Fitting Approximation Algorithm

The procedure for obtaining the curve fitting approximation of s^α using the frequency response data is as follows:

1. Obtain the frequency response data for integer-order part of s^α within the frequency range (ω_l, ω_h) using the function `repfreq()`.
2. Obtain the exact frequency response data of s^α by powering the data obtained in the previous step with α .
3. Obtain the N th order transfer function model of response data using the function `frfit()`.

Based on the above procedure, the algorithm can be written as following function with syntax `G=curFit1(alpha,N,w1,wh)`

```

function G=curFit1(alpha,N,w1,wh)
w=logspace(log10(w1),log10(wh));
s=poly(0,'s'); s=syslin('c',s/((0*s)+1));
F=repfreq(s,w);
f=F.^alpha; G=frfit(w,f,N);
endfunction

```

4.3.5 Transfer Function Estimation

Similarly, the procedure for estimating the integer-order transfer function of s^α is as follows:

1. Obtain the frequency response data for integer-order part of s^α within the frequency range using the function `repfreq()`.
2. Obtain the exact frequency response data of s^α by powering the data obtained in the previous step with α .
3. Estimate the N th order transfer function model of response data using the function `frep2tf()`.

Thus, the above algorithm can be written as following function with syntax

`G=curFit2(alpha,N,w1,wh)`

```
function G=curFit2(alpha,N,w1,wh)
w=logspace(log10(w1),log10(wh),10000);
s = poly(0,'s');
s=syslin('c',s/((0*s)+1));
F=repfreq(s,w);
f=F.^alpha;
[G,Err]=frep2tf(w,f,N);
endfunction
```

4.3.6 Power Series Expansion Techniques

The approximation of discretized fractional-order differentiator in the form of finite impulse response (FIR) filter based on power series expansion is as follows:

$$\begin{aligned} {}_0\mathcal{D}_T^\alpha X(z) &= \left(\frac{1}{T}\right)^\alpha PSE\{(1-z^{-1})^\alpha\}_n \\ &\approx T^{-\alpha} R_n(z^{-1}) \end{aligned} \quad (4.12)$$

where

- n is order of approximation or truncation,
- T is sampling period,
- R is polynomial in the variable z^{-1} of order n and
- $PSE\{(1-z^{-1})^\alpha\}$ denotes the power series expansion of the function $(1-z^{-1})^\alpha$.

From Eq. (4.12), it can be noted that the approximated power series expansion of $(1-z^{-1})^\alpha$ in the form of FIR filter has only zeros. Therefore, the algorithm can be written with syntax `G=firPSE(alpha,N,T)` as follows:

```
function G=firPSE(alpha,N,T)
if alpha>0 then numCoeff=cumprod([1,1-((alpha+1)./[1:N]))];
numCoeff=numCoeff(:,\$:-1:1);
num=poly(numCoeff,'z','c');
z=poly(0,'z');
den=(T^alpha)*z^N;
G=syslin('d',num,den);
end
if alpha<0 then denCoeff=cumprod([1,1-((-alpha+1)./[1:N]))];
denCoeff=denCoeff(:,\$:-1:1);
den=poly(denCoeff,'z','c');
z=poly(0,'z');
num=(T^-alpha)*z^N;
```

```

G=syslin('d',num,den);
end
endfunction

```

Similarly, the approximation of discretized fractional-order differentiator in the form of infinite impulse response (IIR) filter based on power series expansion can be written with syntax as `G=iirPSE(alpha,N,T)` follows:

```

function G=iirPSE(alpha,N,T)
num(1)=1; den(1)=1;
for i=1:N
    num(i+1)=((-1)^i)*(gamma(abs(alpha)+1)./(gamma(i+1).* ...
    gamma(abs(alpha)-i+1)));
    den(i+1)=gamma(abs(alpha)+1)./(gamma(i+1).* ...
    gamma(abs(alpha)-i+1));
end
num=num'; den=den';
num=num(:, $:-1:1);
den=den(:, $:-1:1);
z=poly(0,'z');
num=poly(num,'z','c');
den=poly(den,'z','c');
if alpha>0 then
    G=((2/T)^alpha)*syslin('d',num,den);
end
if alpha<0 then
    G=((2/T)^alpha)*syslin('d',den,num);
end
endfunction

```

4.4 Fractional-order Parameters

In this section, using the Scilab based approximation techniques given in Sect. 4.3, the implementation of fractional-order differentiator and integrator will be briefly explained subsequently.

4.4.1 Fractional-order Differentiator

The fractional-order differentiator (FOD) from Eq. (4.4) is defined as follows:

$$G_d(s) = s^\alpha, \quad 0 < \alpha < 1 \quad (4.13)$$

From Eq. (4.13), the implementation of $G_d(s)$ will be done with the use of approximation algorithms with the syntax of `Gd=fod(alpha, func, N, wl, wh)`, where

- α is the order of differentiation,
- $func$ is the approximation technique,
- N is the order of approximation and
- (ω_l, ω_h) is the desired frequency range of interest.

The five available $func$ options are “Oustaloup”, “RefinedOustaloup”, “Matsuda”, “curveFitting” and “tfEstimate”. Therefore, the function defining the fractional-order differentiator can be written with syntax `G=fod(alpha, f, N, w)` as follows:

```
function G=fod(alpha,f,N,w)
wl=w(1); wh=w(2);
if f=='o' then
    exec('oustaloup.sci');
    G=oustaloup(alpha,N, wl, wh);
elseif f=='r' then
    exec('refOustaloup.sci');
    G=refOustaloup(alpha,N, wl, wh);
elseif f=='m' then
    exec('matsuda.sci');
    G=matsuda(alpha,N, wl, wh);
elseif f=='cf1' then
    exec('curFit1.sci');
    G=curFit1(alpha,N, wl, wh);
elseif f=='cf2' then
    exec('curFit2.sci');
    G=curFit2(alpha,N, wl, wh);
else
    disp('Check the name of the approximation function')
end
endfunction
```

4.4.2 Fractional-order Integrator

Similarly, the fractional-order integrator (FOI) from Eq. (4.4) is defined as follows:

$$G_i(s) = \frac{1}{s^\alpha}, \quad 0 < \alpha < 1 \quad (4.14)$$

Thus, the `foi` function will be implemented with the help of `fod` with the similar syntax as `G=foi(alpha, f, N, w)`, where α is the order of integration. Therefore, the function defining the `foi` is as follows:

```

function G=foi(alpha,f,N,w)
exec('fod.sci');
G=1/fod(alpha,f,N,w);
endfunction

```

4.5 Fractional-order Transfer Function

A fractional-order system will be represented in the form of fractional-order transfer function (FOTF) as follows:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_m s^{\gamma_m} + b_{m-1} s^{\gamma_{m-1}} + \cdots + b_1 s^{\gamma_1} + b_0}{a_n s^{\eta_n} + a_{n-1} s^{\eta_{n-1}} + \cdots + a_1 s^{\eta_1} + a_0} \quad (4.15)$$

where

- $b_i, a_i \in \mathbb{R}$ are the coefficients of numerator and denominator and
- $\gamma_i, \eta_i \in \mathbb{R}$ are the orders of numerator and denominator.

For the model in Eq. (4.15), the Scilab function can be established with the coefficients and orders of numerator and denominator. Therefore, let n and np be the coefficients and order of numerator respectively given as

$$\begin{aligned} n &= [b_m, b_{m-1}, \dots, b_1, b_0] \\ np &= [\gamma_m, \gamma_{m-1}, \dots, \gamma_1, \gamma_0] \end{aligned} \quad (4.16)$$

Similarly, let us consider d and dp as the coefficients and order of denominator also defined as

$$\begin{aligned} d &= [a_n, a_{n-1}, \dots, a_1, a_0] \\ dp &= [\eta_n, \eta_{n-1}, \dots, \eta_1, \eta_0] \end{aligned} \quad (4.17)$$

Thus, the function syntax is $G=fotf(n,np,d,dp,f,N,w)$. As mentioned earlier $func$, N , ω_l and ω_h are the parameters of the approximation algorithm. Consequently, the entire function in Scilab is as follows:

```

function G=fotf(n,np,d,dp,f,N,w)
s=poly(0,'s'); s=syslin('c',s/((0*s)+1));
[nr,nc]=size(n); [npr,npc]=size(np);
[dr,dc]=size(d); [dpr,dpc]=size(dp);
Gnum=0;Gden=0;
exec('fod.sci');
if nc==np & dc==dp then
for i=1:nc
npTemp(i)=fix(np(i));
np(i)=np(i)-npTemp(i);
Gnum=Gnum+n(i)*s^npTemp(i)*fod(np(i),f,N,w);

```

```

end
for i=1:dc
  dpTemp(i)=fix(dp(i));
  dp(i)=dp(i)-dpTemp(i);
  Gden=Gden+d(i)*s^dpTemp(i)*fod(dp(i),f,N,w);
end
G=syslin('c',Gnum/Gden);
else
  disp('Check the orders')
  disp('Order of n is not equal to np or')
  disp('the order of d is not equal to dp')
end
endfunction

```

The result from the above function, G will be the approximated transfer function of the FOTF given in Eq. (4.15). The transfer function data of G i.e., the coefficients of numerator and denominator can be deduced using the following function:

```

function [n,d]=tfdata(G)
n=coeff(G.num);
d=coeff(G.den);
endfunction

```

where n and d are the coefficients of numerator and denominator.

Similarly, the zeros (z) and poles (p) of the approximated transfer function G can also be deduced using the following function:

```

function [z,p]=zeros_poles(G)
G=syslin('c',G);
z=roots(G.num);
p=roots(G.den);
endfunction

```

4.6 Fractional-order Filters

In this section, the design of fractional-order filters of order α with the use of approximation techniques given in Sect. 4.3 are presented [41]. In all the cases, α is chosen between 0 and 2. This is because the system will be stable if and only if $0 < \alpha < 2$. On the other hand, for $\alpha = 2$, the system will produce oscillatory behavior. For values greater than 2, the system will be unstable.

4.6.1 Fractional-order Low-Pass Filter

The transfer function of fractional-order low-pass filter for $0 < \alpha < 2$ is given as follows:

$$F(s) = \frac{\omega_c^\alpha}{s^\alpha + \omega_c^\alpha}, \quad 0 < \alpha < 2 \quad (4.18)$$

where

- α is the fractional-order parameter and
- ω_c is the pole frequency.

The characteristics of the filter are obtained as

$$|F(s)|_{s=j\omega} = \frac{1}{\sqrt{\left(\frac{\omega}{\omega_c}\right)^{2\alpha} + 2\left(\frac{\omega}{\omega_c}\right)^\alpha \cos\left(\frac{\alpha\pi}{2}\right) + 1}} \quad (4.19)$$

$$\angle F(s)|_{s=j\omega} = -\tan^{-1} \frac{\left(\frac{\omega}{\omega_c}\right)^\alpha \sin\left(\frac{\alpha\pi}{2}\right)}{\left(\frac{\omega}{\omega_c}\right)^\alpha \cos\left(\frac{\alpha\pi}{2}\right) + 1} \quad (4.20)$$

The peak frequency of the filter ω_p is defined as the frequency at which the magnitude is maximum or minimum (i.e., $\frac{d}{d\omega}|F(j\omega)|_{\omega=\omega_p} = 0$) and is calculated as

$$\omega_p = \omega_c \left[-\cos\left(\frac{\alpha\pi}{2}\right) \right]^{1/\alpha} \quad (4.21)$$

Similarly, the right-phase frequency ω_{rp} is defined as the frequency at which phase is equal to $\pm\frac{\pi}{2}$ and is calculated as

$$\omega_{rp} = \frac{\omega_c}{\left[-\cos\left(\frac{\alpha\pi}{2}\right) \right]^{1/\alpha}} \quad (4.22)$$

The half-power frequency ω_{hp} is defined as the frequency at which $|F(j\omega)|_{\omega=\omega_h} = \frac{|F(j\omega)|_{\omega=\omega_p}}{\sqrt{2}}$ and is obtained as

$$\omega_{hp} = \omega_c \left(\sqrt{1 + \cos^2\left(\frac{\alpha\pi}{2}\right)} - \cos\left(\frac{\alpha\pi}{2}\right) \right)^{1/\alpha} \quad (4.23)$$

Thus, the phase angle of filter at half-power frequency ω_{hp} is defined as

$$\angle F(j\omega)_{\omega=\omega_{hp}} = -\tan^{-1} \left(\frac{\sin\left(\frac{\alpha\pi}{2}\right)}{2 \cos\left(\frac{\alpha\pi}{2}\right) + \sqrt{1 + \cos^2\left(\frac{\alpha\pi}{2}\right)}} \right) \quad (4.24)$$

Thus, the function with syntax `[H,wp,wrp,wh,Hwh]=foLPF(alpha,wc,f,N,w)` representing the filter using Scilab is given as

```
function [H,wp,wrp,wh,Hwh]=foLPF(alpha,wc,f,N,w)
alphaTemp=fix(alpha);
exec('fod.sci');
s=poly(0,'s'); s=syslin('c',s/((0*s)+1));
alp=alpha-alphaTemp;
num=wc^alpha;
den=((s^alphaTemp)*fod(alp,f,N,w))+(wc^alpha);
H=num/den;
wp=wc*(-cos(alpha*pi/2))^(1/alpha);
wrp=wc/(-cos(alpha*pi/2))^(1/alpha);
wh=wc*(sqrt(1+(cos(alpha*pi/2)^2))...
-cos(alpha*pi/2))^(1/alpha);
Hwh=-atan((sin((alpha*pi)/2))/(((2*cos(alpha*pi/2))...
+(sqrt(1+(cos(alpha*pi/2)^2))))));
endfunction
```

Here, H denotes the approximated N th order transfer function of the filter.

4.6.2 Fractional-order High-Pass Filter

In a similar way, the fractional-order high-pass filter is given as follows:

$$F(s) = \frac{\omega_c^\alpha s^\alpha}{s^\alpha + \omega_c^\alpha}, \quad 0 < \alpha < 2 \quad (4.25)$$

The characteristics responses of the filter $|F(j\omega)|$ and $\angle F(j\omega)$ respectively are defined as

$$|F(s)|_{s=j\omega} = \frac{\left(\frac{\omega}{\omega_c}\right)^\alpha}{\sqrt{\left(\frac{\omega}{\omega_c}\right)^{2\alpha} + 2\left(\frac{\omega}{\omega_c}\right)^\alpha \cos\left(\frac{\alpha\pi}{2}\right) + 1}} \quad (4.26)$$

$$\angle F(s)|_{s=j\omega} = \frac{\alpha\pi}{2} - \tan^{-1} \frac{\left(\frac{\omega}{\omega_c}\right)^\alpha \sin\left(\frac{\alpha\pi}{2}\right)}{\left(\frac{\omega}{\omega_c}\right)^\alpha \cos\left(\frac{\alpha\pi}{2}\right) + 1} \quad (4.27)$$

As defined in Eqs. (4.21), (4.22) and (4.23), the peak, right-phase and half-power frequencies are given in Eqs. (4.28), (4.29) and (4.30) respectively.

$$\omega_p = \left[\frac{\omega_c}{-\cos\left(\frac{\alpha\pi}{2}\right)} \right]^{1/\alpha} \quad (4.28)$$

$$\omega_{rp} = \left[-\omega_c \cos \left(\frac{\alpha\pi}{2} \right) \right]^{1/\alpha} \quad (4.29)$$

$$\omega_{hp} = \left(\omega_c \left(\sqrt{1 + \cos^2 \left(\frac{\alpha\pi}{2} \right)} + \cos \left(\frac{\alpha\pi}{2} \right) \right) \right)^{1/\alpha} \quad (4.30)$$

Thus, the phase angle of fractional-order high-pass filter at ω_{hp} is defined as

$$\angle F(j\omega)_{\omega=\omega_{hp}} = \left(\frac{\alpha\pi}{2} \right) - \tan^{-1} \left(\frac{\sin(\frac{\alpha\pi}{2})}{2 \cos(\frac{\alpha\pi}{2}) + \sqrt{1 + \cos^2(\frac{\alpha\pi}{2})}} \right) \quad (4.31)$$

Therefore, the function with syntax `[H, wp, wrp, wh, Hwh]=fOHPF(alpha, wc, f, N, w)` representing the filter using Scilab is given as

```
function [H,wp,wrp,wh,Hwh]=fOHPF(alpha,wc,f,N,w)
alphaTemp=fix(alpha);
alp=alpha-alphaTemp;
exec('fod.sci');
s=poly(0,'s'); s=syslin('c',s/((0*s)+1));
num=(wc^alp)*((s^alphaTemp)*fod(alp,f,N,w));
den=((s^alphaTemp)*fod(alp,f,N,w))+(wc^alpha);
H=num/den;
wp=wc/(-cos(alpha*pi/2))^(1/alpha);
wrp=wc*(-cos(alpha*pi/2))^(1/alpha);
wh=wc*(sqrt(1+(cos(alpha*pi/2)^2))...
+cos(alpha*pi/2))^(1/alpha);
Hwh=(alpha*pi/2)...
-atan((sin((alpha*pi)/2))/(((2*cos(alpha*pi/2))...
+(sqrt(1+(cos(alpha*pi/2)^2))))));
endfunction
```

4.6.3 Fractional-order Band-Pass Filter

The fractional-order band-pass filter for $0 < \alpha, \beta < 2$ is given in Eq. (4.32). From the equation, it should be noted that the response of fractional-order band-pass filter can only be obtained if and only if $\alpha > \beta$. Thus, the magnitude and phase responses of the filter are given in Eq. (4.33).

$$F(s) = \frac{\omega_c^\alpha s^\beta}{s^\alpha + \omega_c^\alpha}, \quad 0 < \alpha, \beta < 2 \quad (4.32)$$

$$|F(s)|_{s=j\omega} = \frac{\left(\frac{\omega}{\omega_c}\right)^\beta}{\sqrt{\left(\frac{\omega}{\omega_c}\right)^{2\alpha} + 2\left(\frac{\omega}{\omega_c}\right)^\alpha \cos\left(\frac{\alpha\pi}{2}\right) + 1}} \quad (4.33)$$

$$\angle F(s)|_{s=j\omega} = \frac{\beta\pi}{2} - \tan^{-1} \frac{\left(\frac{\omega}{\omega_c}\right)^\alpha \sin\left(\frac{\alpha\pi}{2}\right)}{\left(\frac{\omega}{\omega_c}\right)^\alpha \cos\left(\frac{\alpha\pi}{2}\right) + 1} \quad (4.34)$$

Similar to Eqs. (4.21) and (4.28), the peak frequency of the fractional-order band-pass filter ω_p is defined as

$$\omega_p = \omega_c \left(\frac{\cos\left(\frac{\alpha\pi}{2}\right)[(2\beta - \alpha) + \sqrt{\alpha^2 + 4\beta(\alpha - \beta)\tan^2\left(\frac{\alpha\pi}{2}\right)}]}{2(\alpha - \beta)} \right)^{\frac{1}{\alpha}} \quad (4.35)$$

Therefore, the function with syntax `[H, wp]=foBPF(alpha, bita, wc, f, N, w)` representing the filter using Scilab is given as

```
function [H,wp]=foBPF(alpha,bita,wc,f,N,w)
s=poly(0,'s'); s=syslin('c',s/((0*s)+1));
exec('fod.sci');
alphaTemp=fix(alpha);
alp=alpha-alphaTemp;
bitaTemp=fix(bita);
bi=bita-bitatemp;
num=(wc^alpha)*((s^bitatemp)*fod(bi,f,N,w));
den=((s^alphaTemp)*fod(alp,f,N,w))+(wc^alpha);
H=num/den;
wptemp=sqrt(alpha^2+4*bita*(alpha-bitatemp)...
*tan(alpha*%pi/2)^2);
wp=wc*((cos(alpha*%pi/2)*((2*bita-alpha)+(wptemp)))...
/(2*(alpha-bitatemp)))^(1/alpha);
endfunction
```

4.6.4 Fractional-order All-Pass Filter

The fractional-order all-pass filter is defined as

$$F(s) = \frac{s^\alpha - \omega_c^\alpha}{s^\alpha + \omega_c^\alpha}, \quad 0 < \alpha < 2 \quad (4.36)$$

Furthermore, the magnitude and phase responses of the filter are obtained as follows:

$$|F(s)|_{s=j\omega} = \sqrt{\frac{\left(\frac{\omega}{\omega_c}\right)^{2\alpha} - 2\left(\frac{\omega}{\omega_c}\right)^\alpha \cos\left(\frac{\alpha\pi}{2}\right) + 1}{\left(\frac{\omega}{\omega_c}\right)^{2\alpha} + 2\left(\frac{\omega}{\omega_c}\right)^\alpha \cos\left(\frac{\alpha\pi}{2}\right) + 1}} \quad (4.37)$$

$$\angle F(s)|_{s=j\omega} = -\tan^{-1} \frac{\left(\frac{\omega}{\omega_c}\right)^\alpha \sin\left(\frac{\alpha\pi}{2}\right)}{\left(\frac{\omega}{\omega_c}\right)^\alpha \cos\left(\frac{\alpha\pi}{2}\right) + 1} - \tan^{-1} \frac{\left(\frac{\omega}{\omega_c}\right)^\alpha \sin\left(\frac{\alpha\pi}{2}\right)}{\left(\frac{\omega}{\omega_c}\right)^\alpha \cos\left(\frac{\alpha\pi}{2}\right) - 1} \quad (4.38)$$

Therefore, the function with syntax `H=foAPF(alpha,wc,f,N,w)` representing the filter using Scilab is given as

```
function H=foAPF(alpha,wc,f,N,w)
s=poly(0,'s'); s=syslin('c',s/((0*s)+1));
exec('fod.sci');
alphaTemp=fix(alpha);
alpha=alpha-alphaTemp;
num=((s^alphaTemp)*fod(alpha,f,N,w))-(wc^alpha);
den=((s^alphaTemp)*fod(alpha,f,N,w))+(wc^alpha);
H=num/den;
endfunction
```

4.7 Fractional-order PID Controllers

In this section, the Scilab based design and implementation of various forms of fractional-order PI and PID based controllers with the use of approximation techniques given in Sect. 4.3 are presented.

4.7.1 PID Controller

The transfer function of conventional PID controller ($C(s)$) is given by:

$$C(s)_{PID} = K_p + \frac{K_i}{s} + K_d s \quad (4.39)$$

where, K_p , K_i and K_d are the gains of proportional, integral and derivative actions. The controller can be written in with syntax `C=pid(Kp,Ki,Kd)` as:

```
function C=pid(Kp,Ki,Kd)
s=poly(0,'s');
s=syslin('c',s/((0*s)+1));
C=Kp+(Ki/s)+(Kd*s);
endfunction
```

Over the years, many forms of fractional-order PID based controllers have been proposed by researchers as given in the following subsections.

4.7.2 I. Podlubny's Fractional-order PID Controller

The fractional-order PID (FOPID or $PI^\lambda D^\mu$) according to I. Podlubny's definition is obtained from Eq. (4.39) by fractional-ordering the integral action with λ and derivative actions with μ as follows [5]:

$$C(s)_{FOPID} = K_p + \frac{K_i}{s^\lambda} + K_d s^\mu, \quad 0 \leq \lambda, \mu \leq 2 \quad (4.40)$$

where λ is order of integration and μ is the order of differentiation. In the above equation, s^μ and $1/s^\lambda$ are approximated using Oustaloup approximation as given in Sect. 4.3.1. Thus, the controller can be written in Scilab with syntax `C=fpid(Kp, Ki, Kd, fop, f, N, w)` as follows:

```
function C=fpid(Kp,Ki,Kd,fop,f,N,w)
s=poly(0,'s'); s=syslin('c',s/((0*s)+1));
exec('fod.sci');
lam=fop(1); mu=fop(2);
lambdaTemp=fix(lam); lam=lam-lambdaTemp;
muTemp=fix(mu); mu=mu-muTemp;
C=Kp+ (Ki/(s^lambdaTemp*fod(lam,f,N,w)))...
+ (Kd*s^muTemp*fod(mu,f,N,w));
endfunction
```

From Eq. (4.40), the control transfer functions of FOPI (PI^λ) and FOPD (PD^μ) controllers are given as

$$C(s)_{FOPI} = K_p + \frac{K_i}{s^\lambda}; \quad C(s)_{FOPD} = K_p + K_d s^\mu; \quad (4.41)$$

Thus, the controllers can be written with the use of `fpid.sci` function as follows:

```
function C=fpi(Kp,Ki,fop,f,N,w)
s=poly(0,'s'); s=syslin('c',s/((0*s)+1));
exec('fpid.sci');
C=fpid(Kp,Ki,0,[fop 0],f,N,w);
endfunction
function C=fpd(Kp,Kd,fop,f,N,w)
s=poly(0,'s'); s=syslin('c',s/((0*s)+1));
exec('fpid.sci');
C=fpid(Kp,0,Kd,[0 fop],f,N,w);
endfunction
```

4.7.3 M. Tenoutit's Fractional-order PID Controller

The M. Tenoutit's definition of fractional-order PID i.e., $(PID)^n$ is defined as [42]:

$$C(s)_{(PID)^n} = \frac{1}{s^n} \left(K_p + \frac{K_i}{s} + K_d s \right), \quad 0 \leq n \leq 1 \quad (4.42)$$

where n is the fractional-order parameter.

Here, $1/s^n$ will be approximated using Oustaloup approximation. Thus, the controller can be written with syntax $C=fpidTen(K_p, Ki, Kd, f, n, N, w)$ as:

```
function C=fpidTen(Kp,Ki,Kd,f,n,N,w)
s=poly(0,'s'); s=syslin('c',s/((0*s)+1));
exec('fod.sci');
nTemp=fix(n);n=n-nTemp;
C=(1/(s^nTemp*fod(n,f,N,w)))*(Kp+(Ki/s)+(Kd*s));
endfunction
```

Similarly, M. Tenoutit's fractional-order PI controller (PI^n) and fractional-order PD controller (PD^n) for fractional-order parameter n are defined as:

$$C(s)_{(PI)^n} = \frac{1}{s^n} \left(K_p + \frac{K_i}{s} \right), \quad 0 \leq n \leq 1 \quad (4.43)$$

$$C(s)_{(PD)^n} = \frac{1}{s^n} (K_p + K_d s), \quad 0 \leq n \leq 1 \quad (4.44)$$

The controllers can be implemented with the use of above function by substituting $K_d = 0$ for $(PI)^n$ controller as $C=fpidTen(K_p, Ki, 0, f, n, N, w)$ and $K_i = 0$ for $(PD)^n$ controller as $C=fpidTen(K_p, 0, Kd, f, n, N, w)$.

4.7.4 Luo's Fractional-order PI Controller

The Luo's definitions of $FO[PI]$ (or $[PI]^\lambda$) and $FO[PD]$ (or $[PD]^\mu$) controllers are given in Eqs. (4.45) and (4.46) respectively [43].

$$C(s)_{FO[PI]} = \left[K_p + \frac{K_i}{s} \right]^\lambda, \quad 0 \leq \lambda \leq 2 \quad (4.45)$$

$$C(s)_{FO[PD]} = [K_p + K_d s]^\mu, \quad 0 \leq \mu \leq 2 \quad (4.46)$$

Equation (4.45) can be rewritten as follows:

$$C(s)_{FO[PI]} = K_p^\lambda \left[1 + \frac{K_i}{K_p s} \right]^\lambda \quad (4.47)$$

The 3rd order Binomial expansion of Eq. (4.47) is given by:

$$C(s)_{FO[PI]} = K_p^\lambda \left[1 + \frac{\lambda K_i}{K_p s} + \frac{\lambda(\lambda - 1)}{2!} \left(\frac{K_i}{K_p s} \right)^2 + \frac{\lambda(\lambda - 1)(\lambda - 2)}{3!} \left(\frac{K_i}{K_p s} \right)^3 \right] \quad (4.48)$$

Thus, the controller can be written with syntax `C=fpiLuo(Kp, Ki, lam)` as

```
function C=fpiLuo(Kp,Ki,lam)
s=poly(0,'s'); s=syslin('c',s/((0*s)+1));
C1=(lam*Ki)/(Kp*s);
C2=((lam*(lam-1))/2)*(Ki/(Kp*s))^2;
C3=((lam*(lam-1)*(lam-2))/6)*(Ki/(Kp*s))^3;
C=(Kp^lam)*(1+C1+C2+C3);
endfunction
```

Similarly, Eq. (4.46) can be rewritten as follows:

$$C(s)_{FO[PD]} = K_p^\mu \left[1 + \frac{K_d s}{K_p} \right]^\mu \quad (4.49)$$

Therefore, the 3rd order Binomial expansion of Eq. (4.49) can be derived as:

$$\begin{aligned} C(s)_{FO[PD]} = K_p^\mu & \left[1 + \frac{\mu K_d s}{K_p} + \frac{\mu(\mu-1)}{2!} \left(\frac{K_d s}{K_p} \right)^2 \right. \\ & \left. + \frac{\mu(\mu-1)(\mu-2)}{3!} \left(\frac{K_d s}{K_p} \right)^3 \right] \end{aligned} \quad (4.50)$$

Thus, the controller function can be written with syntax `C=fopdLuo(Kp, Kd, mu)` as follows:

```
function C=fopdLuo(Kp,Kd,mu)
s=poly(0,'s'); s=syslin('c',s/((0*s)+1));
C1=(mu*Kd*s)/(Kp);
C2=(mu*(mu-1)/2)*((Kd*s)/Kp)^2;
C3=(mu*(mu-1)*(mu-2)/6)*((Kd*s)/Kp)^3;
C=(Kp^mu)*(1+C1+C2+C3);
endfunction
```

4.7.5 IMC Based Fractional-order PID Controller

The fractional-order PID (FOPID) based on internal model controller (IMC-FOPID) proposed by Tavakoli et al. is represented as [44]

$$C(s)_{IMC-FOPID} = K_p + \frac{K_i}{s^\alpha} + \frac{K_d s^\alpha}{\gamma s^\alpha + 1}, \quad 0 \leq \alpha \leq 1 \quad (4.51)$$

where $\gamma s^\alpha + 1$ is added to make proper transfer function. Thus, the controller can be written with syntax `C=IMCfpid(Kp, Ki, Kd, gam, alpha, f, N, w)` as following function. Here, s^α will be approximated using Oustaloup approximation.

```

function C=IMCfpid(Kp,Ki,Kd,gam,alpha,f,N,w)
s=poly(0,'s'); s=syslin('c',s/((0*s)+1));
exec('fod.sci');
alphaTemp=fix(alpha);alpha=alpha-alphaTemp;
Ctemp=(s^alphaTemp)*fod(alpha,f,N,w);
C=Kp+(Ki/(Ctemp))+((Kd*Ctemp)/((gam*Ctemp)+1));
endfunction

```

4.7.6 Modified Fractional-order PI Controller

A modified form of FOPI controller (Mod FPI) proposed by V. Feliu-Batlle et al. is represented as follows [45, 46]:

$$C(s)_{ModFPI} = K_p \frac{1 + T_d s}{s^\alpha}, \quad 0 \leq \alpha \leq 1 \quad (4.52)$$

where T_d is the derivative time. Similar to the previous cases, the fractional-order parameter s^α will be approximated using Oustaloup technique. Thus, the controller function with syntax `C=modFPI (Kp, Td, alpha, f, N, w)` defining the controller is as follows:

```

function C=modFPI(Kp,Td,alpha,f,N,w)
s=poly(0,'s'); s=syslin('c',s/((0*s)+1));
exec('fod.sci');
alphaTemp=fix(alpha);alpha=alpha-alphaTemp;
Ctemp=(s^alphaTemp)*fod(alpha,f,N,w);
C=Kp*((1+(Td*s))/Ctemp);
endfunction

```

To improve the steady state performance, the further modified form of FOPI controller is as follows:

$$C(s)_{ModFPISS} = K_p \frac{1 + T_d s}{s^\alpha} \frac{s + \eta}{s}, \quad 0 \leq \alpha \leq 1 \quad (4.53)$$

where η is the minuscule which improves the steady-state behavior. The Scilab function defining the controller with syntax `C=modFPIss (Kp, Td, alpha, eta, f, N, w)` is as follows:

```

function C=modFPIss(Kp,Td,alpha,eta,f,N,w)
s=poly(0,'s'); s=syslin('c',s/((0*s)+1));
exec('fod.sci');
alphaTemp=fix(alpha);alpha=alpha-alphaTemp;
Ctemp=(s^alphaTemp)*fod(alpha,f,N,w);
C=Kp*((1+(Td*s))/Ctemp)*((s+eta)/s);
endfunction

```

4.7.7 Modified Fractional-order PID Controller

Based on the reported works by R. El-Khazali, the modified FOPID (Mod FPID) with assumptions $\lambda = \mu$ and $T_i = T_d$ can be written as follows [47]:

$$C(s)_{ModFPID} = K_c \frac{(1 + T_i s^\mu)^2}{s^\mu}, \quad 0 \leq \mu \leq 1 \quad (4.54)$$

Thus, the Scilab function with syntax `C=modFPID(Kc, Ti, mu, f, N, w)` defining the controller is as follows:

```
function C=modFPID(Kc,Ti,mu,f,N,w)
s=poly(0,'s'); s=syslin('c',s/((0*s)+1));
exec('fod.sci');
muTemp=fix(mu);mu=mu-muTemp;
Ctemp=(s^muTemp)*fod(mu,f,N,w);
C=Kc*(((1+(Ti*Ctemp))^2)/Ctemp);
endfunction
```

where K_c is the gain constant and T_i is the integral time. From both the modified forms given in Eqs. (4.52) and (4.54), it should be noted that there are only three controller parameters as in the case of classical PID. Thus, the summary of all the developed fractional-order PI and PID controllers including their Scilab functions is given in Table 4.3.

Table 4.3 Different variations of fractional-order PID controller

Controller	Transfer function	Syntax
PI	$K_p + \frac{K_i}{s}$	<code>C = pid(Kp,Ki,0)</code>
PD	$K_p + K_d s$	<code>C = pid(Kp,0,Kd)</code>
PID	$K_p + \frac{K_i}{s} + K_d s$	<code>C = pid(Kp,Ki,Kd)</code>
PI^λ	$K_p + \frac{K_i}{s^\lambda}$	<code>C = fpid(Kp,Ki,0,fop,f,N,w)</code>
PD^μ	$K_p + K_d s^\mu$	<code>C = fpid(Kp,0,Kd,fop,f,N,w)</code>
$PI^\lambda D^\mu$	$K_p + \frac{K_i}{s^\lambda} + K_d s^\mu$	<code>C = fpid(Kp,Ki,Kd,fop,f,N,w)</code>
$(PI)^n$	$\frac{1}{s^n} (K_p + \frac{K_i}{s})$	<code>C = fpidTen(Kp,Ki,0,f,n,N,w)</code>
$(PID)^n$	$\frac{1}{s^n} (K_p + \frac{K_i}{s} + K_d s)$	<code>C = fpidTen(Kp,Ki,Kd,f,n,N,w)</code>
$[PI]^\lambda$	$[K_p + \frac{K_i}{s}]^\lambda$	<code>C = fopiLuo(Kp,Ki,lam)</code>
$[PD]^\mu$	$[K_p + K_d s]^\mu$	<code>C = fopdLuo(Kp,Kd,mu)</code>
IMC-FOPID	$K_p + \frac{K_i}{s^\alpha} + \frac{K_d s^\alpha}{\gamma s^\alpha + 1}$	<code>C = IMCfpid(Kp,Ki,Kd,gam,alpha,f,N,w)</code>
Mod FPI	$K_p \frac{1+T_d s}{s^\alpha}$	<code>C = modFPI(Kp,Td,alpha,f,N,w)</code>
Mod FPI SS	$K_p \frac{1+T_d s}{s^\alpha} \frac{s+\eta}{s}$	<code>C = modFPIss(Kp,Td,alpha,eta,f,N,w)</code>
Mod FPID	$K_c \frac{(1+T_i s^\mu)^2}{s^\mu}$	<code>C = modFPID(Kc,Ti,mu,f,N,w)</code>

4.8 Numerical and Stability Analysis

In this section, the Scilab based implementation for numerical and stability analysis of fractional-order systems and controllers are presented. The numerical analysis includes both the time and frequency domain.

4.8.1 Time and Frequency Domain Analysis

The root locus plot of the transfer function model with the help of Scilab function evans can be written as follows:

```
function rlocus(G,Kmax)
    evans(G,Kmax);
    sgrid();
endfunction
```

where K_{max} is the maximum desired gain.

Similarly, the impulse and step responses of the transfer function model for a time period t can be calculated with the extension of Scilab function csim as follows:

```
function impulse(G,t)
    Gres=csim('impulse',t,G);
    plot(t,Gres);
    f=gcf();
    xlabel("Time (sec)");
    ylabel("Amplitude");
    title("Impulse Response");
endfunction

function step(G,t)
    Gres=csim('step',t,G);
    f=gcf();
    plot(t,Gres);
    xlabel("Time (sec)");
    ylabel("Amplitude");
    title("Step Response");
endfunction
```

The stepinfo function with syntax $sInfo=stepinfo(G, t, yf)$ is used to calculate the overshoot (%), undershoot (%), peak, peak time (s), rise time (s) and settling time (s) of the step response is given as follows:

```
function sinfo=stepinfo(G,t,yf)
    y=csim('step',t,G); exec('isfinite.sci');
    if y==[] then
```

```

    disp('Empty Data')
else
TF=isfinite(y);
if TF==1 then ns=length(y);
[sinfo.Peak, ipeak]=max(abs(y));
sinfo.PeakTime=t(ipeak);
if yf==0 then
sinfo.Overshoot=%inf;
if and(y>=0) then
sinfo.Undershoot=0;
else
sinfo.Undershoot=%inf;
end
else
yrel=y/yf; sinfo.Overshoot=100*max(0,max(yrel-1));
sinfo.Undershoot=-100*min(0,min(yrel));
end
tolBand=0.02; err=abs(y-yf);
tol=tolBand*max(err); nn=length(err);
nn1=nn;
for i=1:nn
err1(nn1)=err(i); nn1=nn1-1;
end
err1=err1';
isettle=find(err1>tol,1); nny=length(y);
nny1=nny;
for i=1:nny
y1(nny1)=y(i); nny1=nny1-1;
end
y1=y1';
t1=t(length(t)):- (t(2)-t(1)):t(1);
if isettle==[] then
sinfo.SettlingTime=0;
elseif isettle==ns
sinfo.SettlingTime=%nan;
elseif y1(isettle)~=y1(isettle+1)
y1Settle=yf+sign(y1(isettle)-yf)*tol;
sinfo.SettlingTime = t1(isettle) +...
(t1(isettle)-t1(isettle+1))/...
(y1(isettle)-y1(isettle+1))...
*(y1Settle-y1(isettle));
else
sinfo.SettlingTime=t1(isettle+1);
end
yLow=y(1)+0.1*(yf-y(1));

```

```

iLow=1+find((y(1:ns-1)-yLow).* (y(2:ns)-yLow)<=0,1);
if iLow==[] then
    tLow=%nan;
elseif iLow>1 && y(iLow)~=y(iLow-1)
    tLow=t(iLow)+(t(iLow)-t(iLow-1))/...
    (y(iLow)-y(iLow-1))*(yLow-y(iLow));
else
    tLow=t(iLow);
end
yHigh=y(1) + 0.9*(yf-y(1));
iHigh=1+find((y(1:ns-1)-yHigh).* (y(2:ns)-yHigh)<=0,1);
if iHigh==[] then
    tHigh=%nan;
    sinfo.SettlingMin=%nan; sinfo.SettlingMax=%nan;
else
    if iHigh>1 && y(iHigh)~=y(iHigh-1)
        tHigh=t(iHigh)+(t(iHigh)-t(iHigh-1))/...
        (y(iHigh)-y(iHigh-1))*(yHigh-y(iHigh));
    else
        tHigh=t(iHigh);
    end
    yRisen=y(iHigh:length(y));
    sinfo.SettlingMin=min(yRisen);
    sinfo.SettlingMax=max(yRisen);
end
sinfo.RiseTime=tHigh-tLow;
sinfo.SettlingMin=min(yRisen);
sinfo.SettlingMax=max(yRisen);
else
    sinfo.RiseTime=%nan;
    sinfo.SettlingTime=%nan;
    sinfo.SettlingMin=%nan;
    sinfo.SettlingMax=%nan;
    sinfo.Overshoot=%nan; sinfo.Undershoot=%nan;
    sinfo.Peak=%inf; sinfo.PeakTime=%inf;
end
end
endfunction

```

Here, G is the closed-loop transfer function, t is the time period and yf is the final steady-state value. In the function, by default the settling time threshold limit is chosen as 2%. Thus, values other than this can also be given by changing the variable tolBand .

On the other hand, the frequency domain plots such as Bode, Nyquist, and Nichols are available in Scilab with command `bode`, `nyquist` and `black` respectively. Furthermore, the gain margin, phase margin and crossover frequencies from the Bode plot of the response is calculated using the following function:

```
function [Gm,Pm,Wgm,Wpm] = margins(G)
    [Pm,Wpm] = p_margin(G);
    [Gm,Wgm] = g_margin(G);
    if Wpm ~= [] then
        Wpm = Wpm*2*pi;
    end
    if Wgm ~= [] then
        Wgm = Wgm*2*pi;
    end
    clf();
    show_margins(G, 'bode')
endfunction
```

where `p_margin` and `g_margin` are Scilab functions to calculate phase and gain margins respectively.

4.8.2 Stability Analysis

The stability of the approximated transfer function can also be deduced by using the following function with syntax `S=stability(n,np,d,dp,f,N,w)`. The function will also provide the list of stable and unstable poles of the system.

```
function S=stability(n,np,d,dp,f,N,w)
exec('fotf.sci');
G=fotf(n,np,d,dp,f,N,w);
G=syslin('c',G);
z=roots(G.num);
p=roots(G.den);
pr=real(p);
nu=0; ns=0; up=0; sp=0; temp=0;
for i=1:length(p)
    if pr(i)>0 then
        nu = nu+1; up(nu)=p(i);
    elseif pr(i)<0
        ns = ns+1; sp(ns)=p(i);
        temp=1;
    else
        disp('The system has no poles')
    end
```

Table 4.4 List of functions developed in Scilab based toolbox for fractional-order systems and controllers

Function	Syntax
Grünwald-Letnikov	<code>df = grunwald(f,t,alpha,h)</code>
Oustaloup	<code>G = oustaloup(alpha,N,wl,wh)</code>
Refined Oustaloup	<code>G = refOustaloup(alpha,N,wl,wh)</code>
Matsuda	<code>G = matsuda(alpha,N,wl,wh)</code>
Curve Fitting	<code>G = curFit1(alpha,N,wl,wh)</code>
Transfer Function Estimation	<code>G = curFit2(alpha,N,wl,wh)</code>
Finite Impulse Response	<code>G = firPSE(alpha,N,T)</code>
Infinite Impulse Response	<code>G = iirPSE(alpha,N,T)</code>
Differentiator	<code>G = fod(alpha,f,N,w)</code>
Integrator	<code>G = foi(alpha,f,N,w)</code>
Transfer Function	<code>s = tf('s')</code>
Fractional-order Transfer Function	<code>G = fotf(n,np,d,dp,f,N,w)</code>
Transfer Function Data	<code>[n,d] = tfdata(G)</code>
Zeros and Poles	<code>[z,p] = zeros_poles(G)</code>
PI	<code>C = pid(Kp,Ki,0)</code>
PD	<code>C = pid(Kp,0,Kd)</code>
PID	<code>C = pid(Kp,Ki,Kd)</code>
PI^λ	<code>C = fpid(Kp,Ki,fop,f,N,w)</code>
PD^μ	<code>C = fpid(Kp,Kd,fop,f,N,w)</code>
$\text{PI}^\lambda \text{D}^\mu$	<code>C = fpid(Kp,Ki,Kd,fop,f,N,w)</code>
$(\text{PI})^n$	<code>C = fpidTen(Kp,Ki,0,f,n,N,w)</code>
$(\text{PD})^n$	<code>C = fpidTen(Kp,0,Kd,f,n,N,w)</code>
$(\text{PID})^n$	<code>C = fpidTen(Kp,Ki,Kd,f,n,N,w)</code>
$[\text{PI}]^\lambda$	<code>C = fpiLuo(Kp,Ki,lambda)</code>
$[\text{PD}]^\mu$	<code>C = fpdLuo(Kp,Kd,mu)</code>
IMC-FOPID	<code>C = IMCfpid(Kp,Ki,Kd,gam,alpha,f,N,w)</code>
Mod FPI	<code>C = modFPI(Kp,Td,alpha,f,N,w)</code>
Mod FPI SS	<code>C = modFPIss(Kp,Td,alpha,eta,f,N,w)</code>
Mod FPID	<code>C = modFPID(Kc,Ti,mu,f,N,w)</code>
Low-pass Filter	<code>[H,wp,wrp,wh,Hwh] = foLPF(alpha,wc,f,N,w)</code>
High-pass Filter	<code>[H,wp,wrp,wh,Hwh] = foHPF(alpha,wc,f,N,w)</code>
Band-pass Filter	<code>[H,wp] = foBPF(alpha,bita,wc,f,N,w)</code>
All-pass Filter	<code>H = foAPF(alpha,wc,f,N,w)</code>
Pole Zero Plot	<code>plzr(G)</code>
Root locus	<code>rlocus(G,Kmax)</code>
Step	<code>step(G,t)</code>

(continued)

Table 4.4 (continued)

Function	Syntax
Step Info	sInfo=stepinfo(G,t,yf)
Impulse	impulse(G,t)
Nyquist	nyquist(G)
Bode	bode(G)
Magnitude Plot	gainplot(G)
Phase Plot	phaseplot(G)
Asymptotic Bode Plot	bode_asymp(G)
Peak Frequencies	fr = freson(G)
Margins	[Gm, Pm, Wgm, Wpm] = margins(G)
Nichols	black(G), nicholschart()
Singular-value Sigma-plot	svplot(G)
Stability	S = stability(n,np,d,dp,f,N,w)
H ₂ -norm	h2norm(G)
H _∞ -norm	h_norm(G)
L _∞ -norm	linf(G)

```

end
disp(p,'Pole(s) of the system:')
if length(sp)>0 & temp==1 then
  disp([sp],'Stable pole(s):')
end
if length(up)>0 & up~=0 then
  disp([up],'Unstable pole(s):')
end
if nu>0 then
  disp('Unstable System'); S=0;
else
  disp('Stable System'); S=1;
end
endfunction

```

The summary of all the developed functions in this toolbox including their Scilab syntax is given in Table 4.4. Furthermore, the organization chart of the complete toolbox is shown in Fig. 4.1.

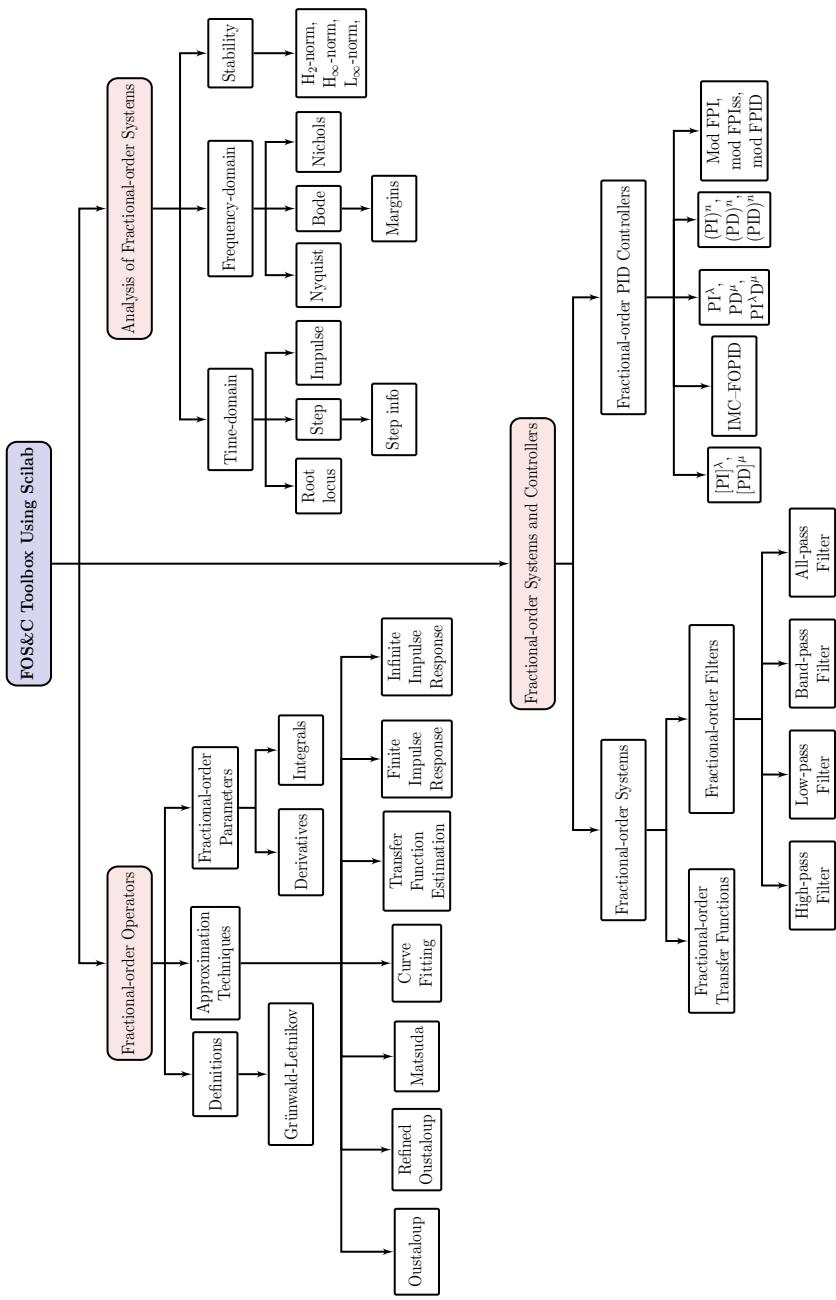


Fig. 4.1 Organization chart of the complete Scilab based toolbox for fractional-order systems and controllers

4.9 Simulation Study

In this section, a simulation study is conducted on several fractional-order based operators, PID controller, transfer function models, and filters to evaluate the performance of the developed Scilab based toolbox.

4.9.1 Fractional-order Operators and Transfer Functions

4.9.1.1 Grünwald-Letnikov

Consider the function $f(t) = e^{-4t} \cos(2t + 1)$. The fractional-order derivative of $f(t)$ for orders $\alpha = 0.6, 0.7$ and 0.8 for the time period $t = 0 - \pi$ using Grünwald-Letnikov's definition is obtained with the following commands:

```
--> t=0:0.01:%pi; alpha=[0.6 0.7 0.8]; h=0.001;
exec('grunwald.sci'); f=exp(-4*t).*cos(2*t+1);
for i=1:length(alpha)
df=grunwald(f,t,alpha(i),h);
plot(t,df')
end
xlabel('Time (sec)');
ylabel('Amplitude');
legend(['$\alpha=0.6$', '$\alpha=0.7$', '$\alpha=0.8$']);
```

Thus, the resultant plot from the above commands is shown in Fig. 4.2. Furthermore, for the same function, the fractional-order derivatives for different orders of

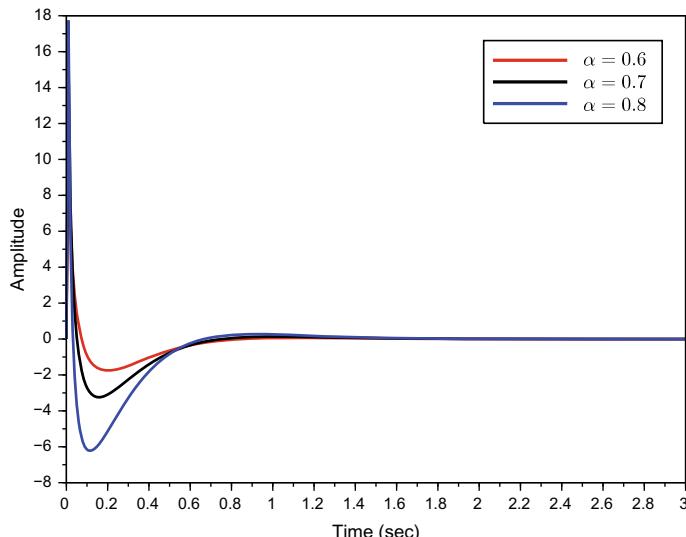


Fig. 4.2 Fractional-order derivatives of $f(t)$ for orders $\alpha = 0.6, 0.7$ and 0.8

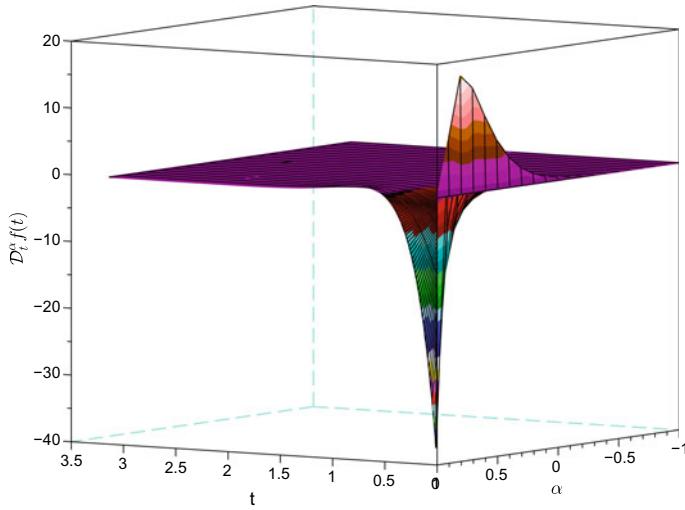


Fig. 4.3 Fractional-order derivatives of $f(t)$ for different orders of α

$\alpha \in (-1, 1)$ for the time period of $t = 0 - \pi$ shown in Fig. 4.3 can be obtained using the following commands:

```
--> t=0:0.01:%pi; alpha=[-1:0.1:1]; h=0.001; F=[];
exec('grunwald.sci'); f=exp(-4*t).*cos(2*t+1);
for i=alpha
F=[F grunwald(f,t,i,h)];
end
surf(t,alpha,F');
xlabel('t'); ylabel('$\alpha$');
zlabel('$\mathcal{D}_t^\alpha f(t)$');
```

Similarly, consider the sinusoidal function $f(t) = \sin(t)$. The fractional-order derivatives of $f(t)$ for different orders of $\alpha \in (0, 1)$ for the time period of $t = 0 - 5\pi$ shown in Fig. 4.4 can be obtained using the following commands:

```
--> t=0:0.01:5*pi; alpha=[0:0.1:1]; h=0.01; F=[];
exec('grunwald.sci'); f=sin(t);
for i=alpha
F=[F grunwald(f,t,i,h)];
end
surf(t,alpha,F');
xlabel('t'); ylabel('$\alpha$');
zlabel('$\mathcal{D}_t^\alpha f(t)$');
```

4.9.1.2 Approximation Techniques

Consider the fractional-order differentiator of order $\alpha = 0.8$, the approximated transfer function for order of approximation $N = 5$ and the desired frequency range

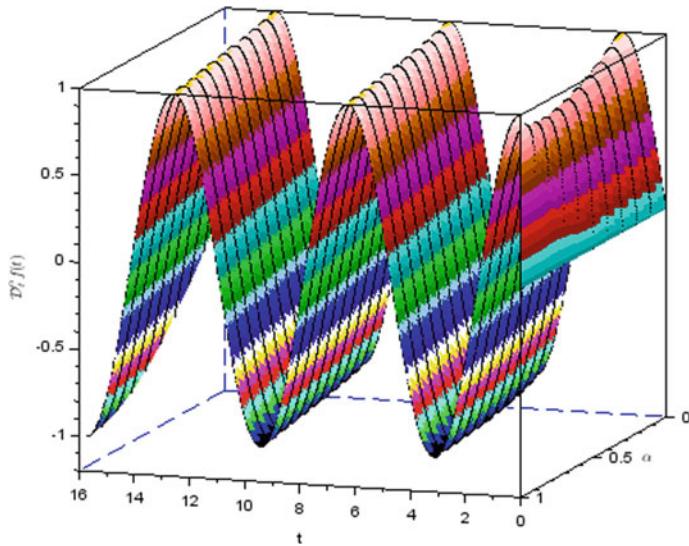


Fig. 4.4 Fractional-order derivatives of sinusoidal function for different orders of α

$\omega = (10^{-5}, 10^5)$ can be obtained using various approximation techniques presented in Sect. 4.3 as follows:

```
--> N=5; wl=10^-5; wh=10^5; alp=0.8;
exec('oustaloup.sci'); G1=oustaloup(alp,N,wl,wh);
exec('refOustaloup.sci'); G2=refOustaloup(alp,N,wl,wh);
exec('matsuda.sci'); G3=matsuda(alp,2*N,wl,wh);
exec('curFit1.sci'); G4=curFit1(alp,N,wl,wh);
exec('curFit2.sci'); G5=curFit2(alp,N,wl,wh);
bode([G1;G2;G3;G4;G5],wl,wh,"rad")
legend('Oustaloup','Refined Oustaloup','Matsuda',...
'Curve Fitting1','Curve Fitting2');
scf();
exec('firPSE.sci'); G6=firPSE(alp,20*N,0.01);
exec('iirPSE.sci'); G7=iirPSE(alp,20*N,0.01);
bode([G6;G7],"rad"); legend('FIR-PSE','IIR-PSE');
```

Thus, the approximated transfer function of $s^{0.6}$ using Oustaloup, refined Oustaloup, Matsuda and both the curve fitting techniques from the above commands are as follows:

$$G_1(s) \approx \frac{[1 + 63733.065s + 40216867s^2 + 2.538D + 08s^3 + 16009022s^4 + 10000s^5]}{[10000 + 16009022s + 2.538D + 08s^2 + 40216867s^3 + 63733.065s^4 + s^5]} \quad (4.55)$$

$$G_2(s) \approx \frac{[8.093D - 66s + 5.158D - 49s^2 + 3.255D - 34s^3 + 2.054D - 21s^4 + 1.296D - 10s^5 + 0.081758s^6 + 515858.01s^7 + 3.255D + 10s^8 + 2.054D + 13s^9 + 1.296D + 14s^{10} + 8.176D + 12s^{11} + 5.180D + 09s^{12} + 45958.306s^{13}]}{[2.524D - 57 + 4.040D - 42s + 6.404D - 29s^2 + 1.015D - 17s^3 + 1.609D - 08s^4 + 0.2571923s^5 + 75818.957s^6 + 5.676D + 09s^7 + 8.896D + 12s^8 + 1.410D + 14s^9 + 2.234D + 13s^{10} + 3.545D + 10s^{11} + 619288.62s^{12} + s^{13}]} \quad (4.56)$$

$$G_3(s) \approx \frac{[1 + 189966.01s + 1.292D + 08s^2 + 8.203D + 08s^3 + 51196362s^4 + 28613.773s^5]}{[28613.773 + 51196362s + 8.203D + 08s^2 + 1.292D + 08s^3 + 189966.01s^4 + s^5]} \quad (4.57)$$

$$G_4(s) \approx \frac{[5209.5414 + 5284093.2s + 74629531s^2 + 77813607s^3 + 15008004s^4 + 435054.73s^5]}{[752376.93 + 57692362s + 88806404s^2 + 23776455s^3 + 1074614.1s^4 + s^5]} \quad (4.58)$$

$$G_5(s) \approx \frac{[-5.062D - 09 - 0.0000041s - 1.288D - 09s^2 - 2.317D - 14s^3 + 8.300D - 20s^4 + 3.638D - 25s^5]}{[-0.0000136 - 8.206D - 09s - 2.598D - 13s^2 + 4.312D - 19s^3 + 5.758D - 24s^4 + 1.112D - 30s^5]} \quad (4.59)$$

Furthermore, the resultant bode plot of the approximated transfer function is shown in Fig. 4.5. On the other hand, the resultant Bode plot using the power series expansion techniques is shown in Fig. 4.6.

The above approximation techniques can also be implemented with the use of `fod.sci` function as follows:

```
--> N=5; wl=10^-5; wh=10^5; alp=0.8;
exec('fod.sci'); w=[wl wh];
G1=fod(alp,'o',N,w); G2=fod(alp,'r',N,w);
G3=fod(alp,'m',N,w); G4=fod(alp,'cf1',N,w);
G5=fod(alp,'cf2',N,w);
bode([G1;G2;G3;G4;G5],wl,wh,"rad")
legend('Oustaloup','Refined Oustaloup','Matsuda',...
'Curve Fitting1','Curve Fitting2');
```

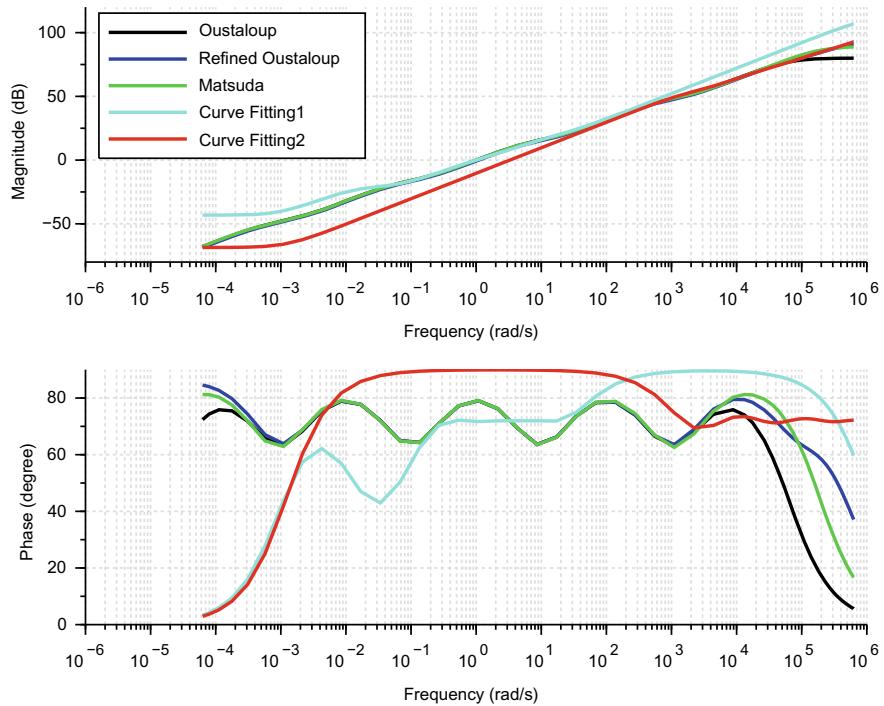


Fig. 4.5 Bode plot of fractional-order differentiator using various techniques

The results from the above commands are same as the results given in Eqs. (4.55), (4.56), (4.57), (4.58), (4.59), and Fig. 4.5. Similarly, the approximation of fractional-order integrator for order $\alpha = 0.75$ using Oustaloup, refined Oustaloup, Matsuda and both the curve fitting techniques can be done with the use of `foi.sci` function as follows:

```
--> N=5; wl=10^-5; wh=10^5; alp=0.75;
exec('foi.sci'); w=[wl wh];
G1=foi(alp,'o',N,w); G2=foi(alp,'r',N,w);
G3=foi(alp,'m',N,w); G4=foi(alp,'cf1',N,w);
G5=foi(alp,'cf2',N,w);
bode([G1;G2;G3;G4;G5],wl,wh,"rad")
legend('Oustaloup','Refined Oustaloup','Matsuda',...
'Curve Fitting1','Curve Fitting2');
```

From the above commands, it can be seen that the order of approximation and the desired frequency range are chosen as $N = 5$ and $\omega = (10^{-5}, 10^5)$. Thus, the approximated transfer functions obtained from the above commands using various techniques are as follows:

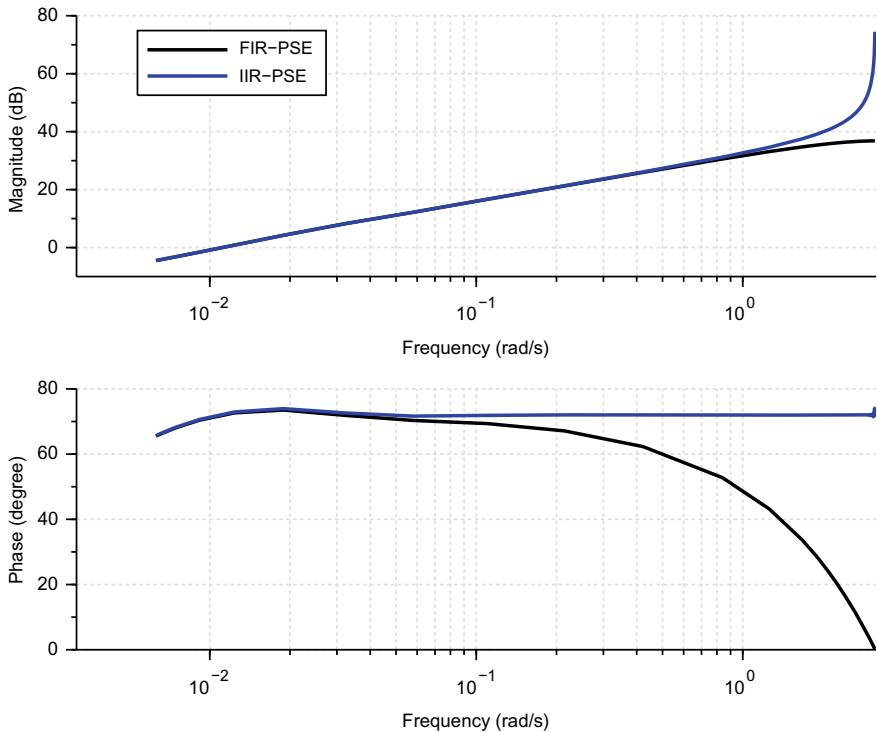


Fig. 4.6 Bode plot of fractional-order differentiator using FIR and IIR

$$G_1(s) \approx \frac{[5623.4133 + 10101010s + 1.796D + 08s^2 + 31945393s^3 + 56802.154s^4 + s^5]}{[1 + 56802.154s + 31945393s^2 + 1.796D + 08s^3 + 10101010s^4 + 5623.4133s^5]} \quad (4.60)$$

$$G_2(s) \approx \frac{[5.335D - 58 + 9.583D - 43s + 1.704D - 29s^2 + 3.031D - 18s^3 + 5.390D - 09s^4 + 0.0966347s^5 + 31240.335s^6 + 2.555D + 09s^7 + 4.490D + 12s^8 + 7.984D + 13s^9 + 1.420D + 13s^{10} + 2.528D + 10s^{11} + 501246.6s^{12} + s^{13}]}{[1.299D - 65s + 7.377D - 49s^2 + 4.149D - 34s^3 + 2.333D - 21s^4 + 1.312D - 10s^5 + 0.0737748s^6 + 414866.19s^7 + 2.333D + 10s^8 + 1.312D + 13s^9 + 7.377D + 13s^{10} + 4.149D + 12s^{11} + 2.347D + 09s^{12} + 20784.61s^{13}]} \quad (4.61)$$

$$G_3(s) \approx \frac{[13429.716 + 27013946s + 4.858D + 08s^2 + 85914451s^3 + 142763.38s^4 + s^5]}{[1 + 142763.38s + 85914451s^2 + 4.858D + 08s^3 + 27013946s^4 + 13429.716s^5]} \quad (4.62)$$

$$G_4(s) \approx \frac{[313626.66 + 32533731s + 52908176s^2 + 14724923s^3 + 695318.33s^4 + s^5]}{[3522.6379 + 3567540.8s + 44947701s^2 + 44843529s^3 + 8272345.1s^4 + 223169.53s^5]} \quad (4.63)$$

$$G_5(s) \approx \frac{[-0.0000011 - 1.422D - 09s - 7.391D - 14s^2 + 6.844D - 19s^3 + 6.229D - 24s^4 + 1.703D - 30s^5]}{[-6.124D - 09 - 0.0000003s - 1.619D - 10s^2 - 3.550D - 15s^3 + 5.439D - 20s^4 + 2.016D - 25s^5]} \quad (4.64)$$

Therefore, the resultant Bode plot of fractional-order integrator using various techniques obtained from the above commands is shown in Fig. 4.7.

Furthermore, the approximation of the transfer functions can be done using `fotf.sci` function. To illustrate this, consider the fractional-order transfer function given as follows:

$$G(s) = \frac{s+1}{10s^{3.2} + 185s^{2.5} + 288s^{0.7} + 1} \quad (4.65)$$

The approximated transfer function of $G(s)$ for the order of approximation $N = 5$ and the desired frequency range $\omega = (10^{-5}, 10^5)$ using Oustaloup, refined Oustaloup, and both the curve fitting techniques can be obtained using the following commands:

```
--> N=5; wl=10^-5; wh=10^5;
exec('fotf.sci'); n=[1 1]; np=[1 0];
d=[10 185 288 1]; dp=[3.2 2.5 0.7 0];
Go=fotf(n,np,d,dp,'o',N,[wl wh]);
Gr=fotf(n,np,d,dp,'r',N,[wl wh]);
Gc1=fotf(n,np,d,dp,'cf1',N,[wl wh]);
Gc2=fotf(n,np,d,dp,'cf2',N,[wl wh]);
bode([Go;Gr;Gc1;Gc2],wl,wh,"rad")
legend('Oustaloup','Refined Oustaloup',...
'Curve Fitting1','Curve Fitting2');
```

In the above commands, Go , Gr , $Gc1$ and $Gc2$ represents the approximated transfer functions of Oustaloup, refined Oustaloup, and both the curve fitting techniques. Thus, the resultant Bode plot from the above commands in comparison with these techniques is shown in Fig. 4.8.

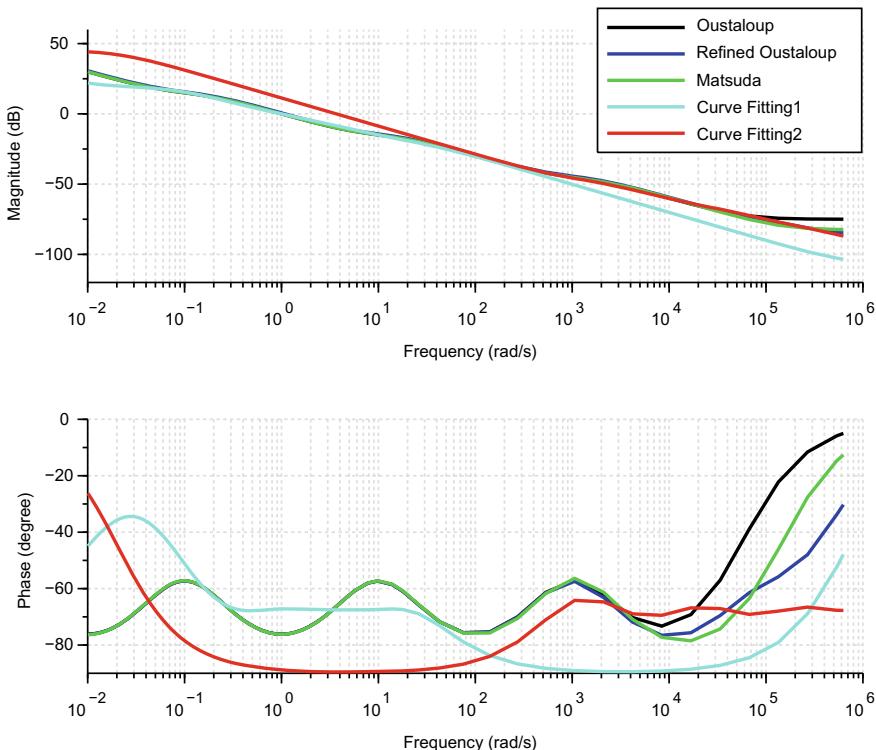


Fig. 4.7 Bode plot of fractional-order integrator using various techniques

The coefficients of numerator and denominator of the Oustaloup based approximated transfer function $G_o(s)$ can be obtained as `exec('tfdata.sci');` `[n, d] = tfdata(Go);`. On the other hand, the zeros and poles of the $G_o(s)$ can be obtained as `exec('zeros_poles.sci');` `[z, p] = zeros_poles(Go);`. Therefore, the resultants coefficients, zeros and poles of the $G_o(s)$ are given in Table 4.5.

4.9.1.3 Time and Frequency-Domain Plots

To illustrate all time and frequency-domain plots, consists the fractional-order transfer function as follows:

$$G(s) = \frac{5s^{0.6} + 2}{s^{3.3} + 3.1s^{2.6} + 2.89s^{1.9} + 2.5s^{1.4} + 1.2} \quad (4.66)$$

The pole-zero plot and the root-locus plot of the Oustaloup based approximated transfer function $G(s)$ for the order of approximation $N = 4$ and the desired frequency range $\omega = (10^{-4}, 10^4)$ can be obtained as follows:

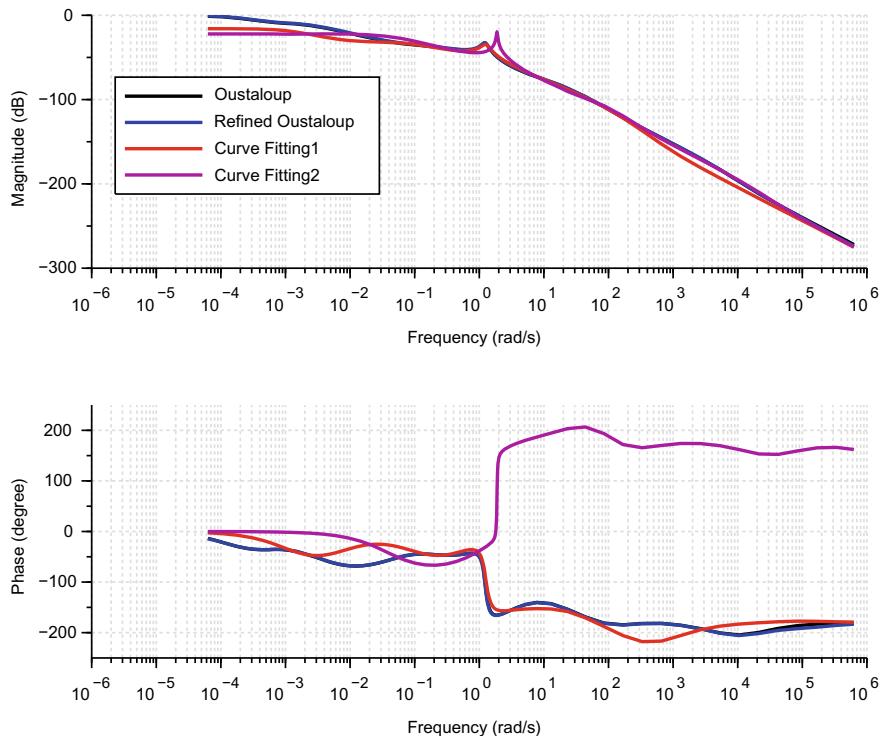


Fig. 4.8 Bode plot of fractional-order transfer function using various approximation techniques

```
--> clear; clc; N=4; w=[10^-4 10^4]; exec('fotf.sci');
n=[5 2]; np=[0.6 0];
d=[1 3.1 2.89 2.5 1.2]; dp=[3.3 2.6 1.9 1.4 0];
G=fotf(n,np,d,dp,'o',N,w);
scf(); subplot(121); plzr(G);
exec('rlocus.sci'); subplot(122); rlocus(G,100);
```

Therefore, the resultant root-locus plot from the above command is shown in Fig. 4.9. Similarly, the time domain plots both impulse and step response of approximated transfer function $G(s)$ shown in Fig. 4.10 can be obtained using the following commands

```
--> scf(); t=0:1:200;
Gri=csim('impulse',t,G); Grs=csim('step',t,G);
subplot(121); plot(t,Gri);
xlabel("Time(sec)"); ylabel("Amplitude");
title("Impulse Response");
subplot(122); plot(t,Grs);
xlabel("Time(sec)"); ylabel("Amplitude");
title("Step Response");
```

Table 4.5 Coefficients of numerator and denominator and the zeros and poles of $G(s)$

Coeff. of numerator	Coeff. of denominator	Zeros	Poles
10000000	10910736	-50118.723	-50118.723
1.158D+11	1.706D+11	-31622.777	-31955.834
4.020D+14	8.851D+14	-15848.932	-6424.1415
4.440D+17	1.628D+18	-501.18723	-501.19184
4.812D+19	6.198D+20	-316.22777	-406.64971
1.665D+21	5.053D+22	-158.48932	-91.832328
1.929D+22	1.206D+24	-5.0118723	-20.212329
3.666D+22	5.812D+24	-3.1622777	-4.8381822
2.543D+22	6.098D+24	-1.5848932	-1.550911
7.142D+21	5.236D+24	-1	-0.1349838 + 1.2346476i
7.113D+20	3.012D+24	-0.0501187	-0.1349838 - 1.2346476i
7.584D+18	5.683D+23	-0.0316228	-0.2047152
2.565D+16	2.755D+22	-0.0158489	-0.0316423
2.799D+13	3.292D+20	-0.0005012	-0.0158489
2.977D+09	1.124D+18	-0.0001585	-0.0026804
98577.194	1.250D+15	-0.0003162	-0.0001585
1	2.220D+11		-0.0001756
	8952550		-0.0003162
	100		

Furthermore, the step response characteristics and the stability analysis of the approximated transfer function $G(s)$ can be obtained as follows:

```
--> exec('stepinfo.sci'); yf=1.8;
Gsinfo=stepinfo(G,t,yf);
exec('stability.sci'); S=stability(n,np,d,dp,'o',N,w);
h2=h2norm(G); hi=h_norm(G); Li=linf(G);
```

Thus, the resultant numerical analysis obtained from the above commands are Peak: 5.0554, PeakTime: 6, Overshoot: 180.8588, Undershoot: 0, SettlingTime: 57.4028, SettlingMin: 0.5397, SettlingMax: 5.0554 and RiseTime: 1.6006. Furthermore, the stability analysis obtained are S=1 which denotes a Stable system. The H_2 -norm, H_∞ -norms and L_∞ -noms as 4.293D+12i, 10.3609 and 0.0029 respectively also justifies that the approximated transfer function is stable.

On the other hand, the frequency domain plots like Bode, Nyquist and Nichols plots shown in Figs. 4.11, 4.12 and 4.13 of the approximated transfer function $G(s)$ can be obtained using the following commands

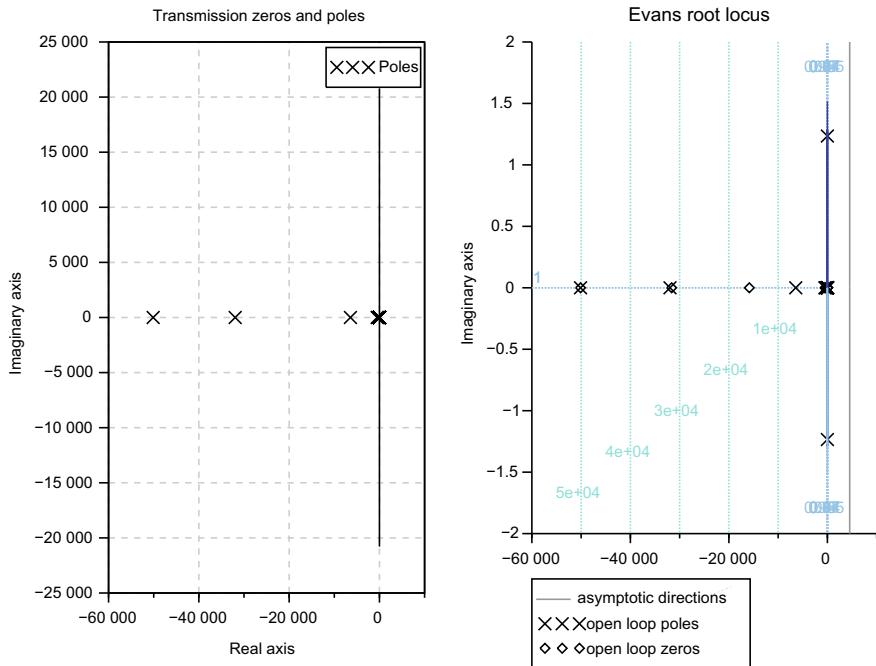


Fig. 4.9 Pole-zero and root-locus plots of fractional-order transfer function $G(s)$

```
--> scf(); bode(G, "rad");
scf(); subplot(121); nyquist(G);
subplot(122); nyquist(G); hallchart();
scf(); subplot(121); black(G);
subplot(122); black(G); nicholschart();
```

In the above commands, the functions `hallchart()` and `nicholschart()` are used as grids for `nyquist` and `nichols` plots respectively. Furthermore, the peak frequencies of the magnitude plot given in Fig. 4.11 can be obtained with the use of Scilab function `fr=freson(G)` as follows:

```
--> scf(); gainplot(G);
fr=freson(G); fig=gce();
p1=fig.children(1);
[phi,db]=phasemag(repfreq(G,fr));
plot(fr,db(:, 'ob');
t=datatipCreate(p1,[fr db(:)]);
datatipSetOrientation(t,"upper left");
```

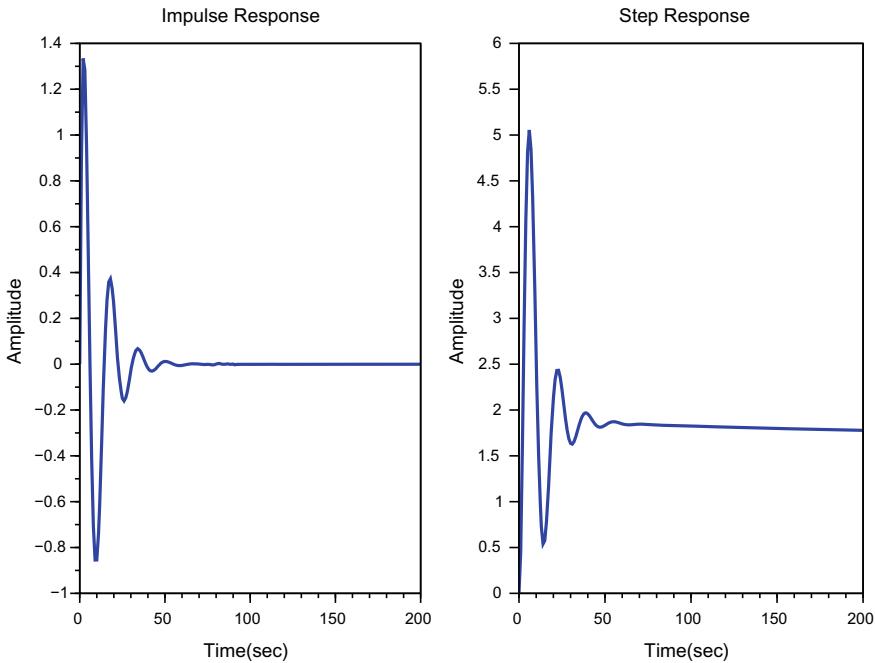


Fig. 4.10 Impulse and step response of fractional-order transfer function $G(s)$

In the above commands it can be noted that the peak frequency f_r has been plotted on magnitude plot of $G(s)$ which is obtained using the Scilab function `gainplot(G)` as shown in Fig. 4.14. Furthermore, the datatip shown in the figure has also been created using Scilab function `datatipCreate()`.

On the other hand, the Asymptotic Bode plot of $G(s)$ shown in Fig. 4.15 can be obtained as follows:

```
--> scf(); bode(G,wl,wh); bode_asymp(G,wl,wh);
```

4.9.2 Fractional-order PID Controllers

In this subsection, a simulation study on the performance of various fractional-order PI and PID controllers, non-linear stable spherical tank and an unstable bioreactor processes are considered to evaluate the performance of the proposed Scilab based fractional-order PI and PID controllers. In all the cases, the fractional-order operators are approximated using Oustaloup approximation with parameters $N = 5$, $\omega_l = 10^{-3}$ and $\omega_h = 10^3$.

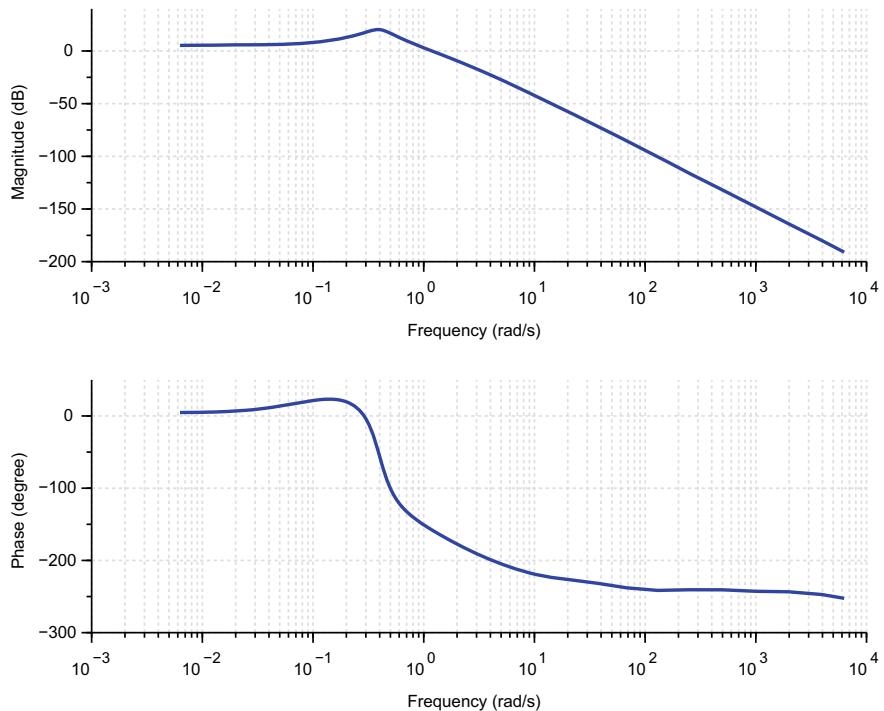
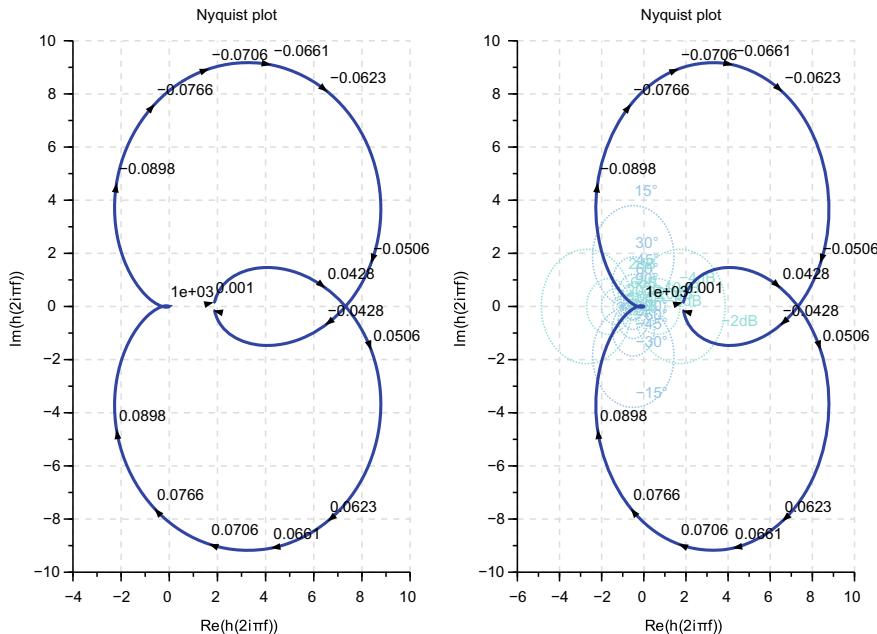
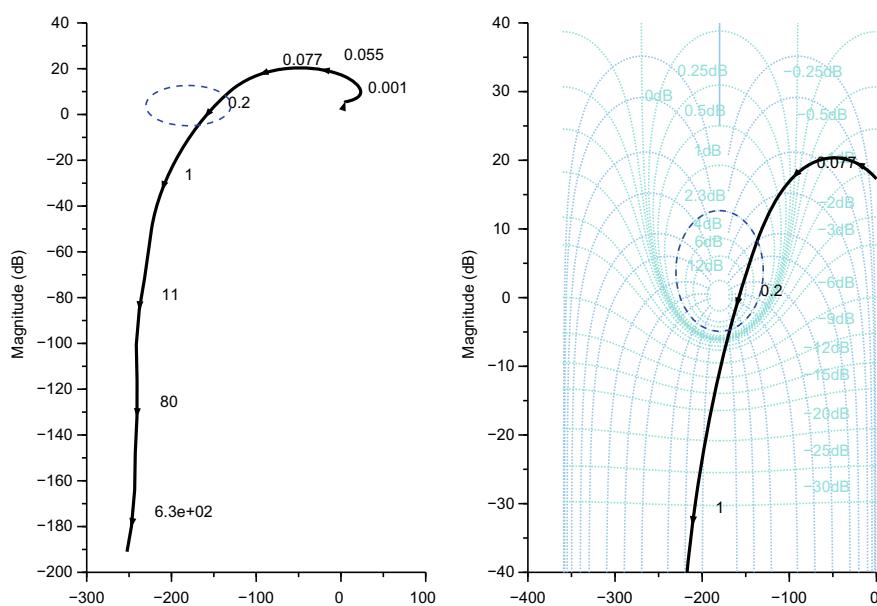


Fig. 4.11 Bode plot of fractional-order transfer function $G(s)$

4.9.2.1 Analysis of Fractional-order PID Controllers

The Bode plot of I. Podlubny's fractional-order PID controller is given in Sect. 4.7.2 approximated using Oustaloup, refined Oustaloup, Matsuda, and curve fitting techniques in comparison with conventional PID controller can be obtained using the following commands. For this analysis the gain parameters of the controller are chosen as $K_p = K_i = K_d = 1$ and the fractional-order parameters are chosen as $\lambda = 0.98$ and $\mu = 0.85$.

```
--> exec('pid.sci'); exec('fpid.sci'); C1=pid(1,1,1);
C2=fpid(1,1,1,[0.98 0.85],'o',5,[10^-5 10^5]);
C3=fpid(1,1,1,[0.98 0.85],'r',5,[10^-5 10^5]);
C4=fpid(1,1,1,[0.98 0.85],'m',2*5,[10^-5 10^5]);
C5=fpid(1,1,1,[0.98 0.85],'cf1',5,[10^-5 10^5]);
bode([C1;C2;C3;C4;C5],"rad");
legend('$PID$', '$PI^\lambda D^\mu - Oustaloup$', ...
'$PI^\lambda D^\mu - RefinedOustaloup$', ...
'$PI^\lambda D^\mu - Matsuda$', ...
'$PI^\lambda D^\mu - Curve Fitting$');
```

Fig. 4.12 Nyquist plot of fractional-order transfer function $G(s)$ Fig. 4.13 Nichols plot of fractional-order transfer function $G(s)$

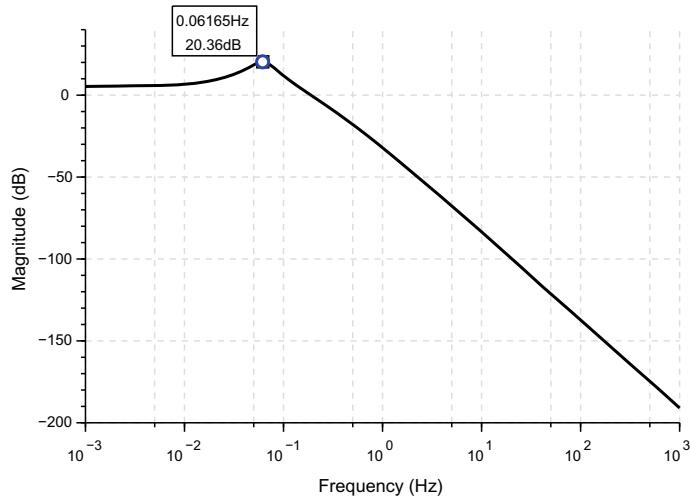


Fig. 4.14 Magnitude plot of $G(s)$ showing the peak frequencies

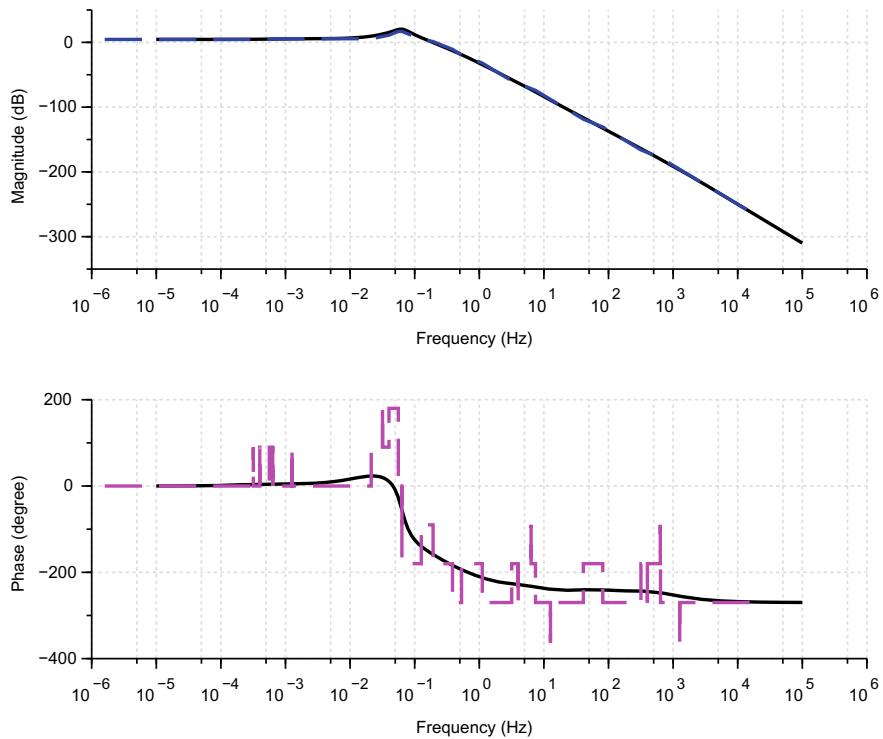


Fig. 4.15 Asymptotic Bode plot of fractional-order transfer function $G(s)$

From the above commands, the resultant transfer functions of the controller are as follows:

$$C_1(s) = \frac{1 + s + s^2}{s} \quad (4.67)$$

$$C_2(s) \approx \frac{[1.413D + 09 + 3.511D + 12s + 2.180D + 15s^2 + 5.484D + 16s^3 + 3.821D + 17s^4 + 4.773D + 17s^5 + 4.014D + 17s^6 + 5.680D + 16s^7 + 2.186D + 15s^8 + 3.515D + 12s^9]}{[17782.794 + 1.741D + 09s + 4.086D + 12s^2 + 2.388D + 15s^3]} \quad (4.68)$$

$$+ 5.535D + 16s^4 + 3.222D + 17s^5 + 7.467D + 16s^6 + 4.345D + 15s^7 + 1.003D + 13s^8 + 5.764D + 09s^9 + 79432.823s^{10}]$$

$$C_3(s) \approx \frac{[3.75D - 110 + 9.306D - 95s + 5.770D - 80s^2 + 1.391D - 66s^3 + 8.537D - 54s^4 + 2.057D - 42s^5 + 1.263D - 31s^6 + 3.043D - 22s^7 + 1.869D - 13s^8 + 0.0000045s^9 + 28.727814s^{10} + 13467045s^{11} + 2.443D + 12s^{12} + 1.994D + 17s^{13} + 6.297D + 21s^{14} + 1.475D + 25s^{15} + 8.977D + 27s^{16} + 2.248D + 29s^{17} + 1.547D + 30s^{18} + 1.778D + 30s^{19} + 1.389D + 30s^{20} + 1.942D + 29s^{21} + 7.424D + 27s^{22} + 1.206D + 25s^{23} + 5.009D + 21s^{24} + 8.725D + 16s^{25} + 3.883D + 11s^{26}]}{[8.38D - 122s + 8.20D - 105s^2 + 1.925D - 89s^3 + 1.125D - 74s^4]} \quad (4.69)$$

$$+ 2.608D - 61s^5 + 1.518D - 48s^6 + 3.519D - 37s^7 + 2.048D - 26s^8 + 4.747D - 17s^9 + 2.763D - 08s^{10} + 0.6438997s^{11} + 3810107.8s^{12} + 1.351D + 12s^{13} + 1.632D + 17s^{14} + 6.687D + 21s^{15} + 1.530D + 25s^{16} + 8.866D + 27s^{17} + 2.055D + 29s^{18} + 1.196D + 30s^{19} + 2.772D + 29s^{20} + 1.613D + 28s^{21} + 3.740D + 25s^{22} + 2.178D + 22s^{23} + 5.167D + 17s^{24} + 3.311D + 12s^{25} + 3582017.2s^{26}]$$

$$C_4(s) \approx \frac{[1.267D + 11 + 3.508D + 14s + 2.420D + 17s^2 + 6.161D + 18s^3 + 4.330D + 19s^4 + 5.405D + 19s^5 + 4.549D + 19s^6 + 6.380D + 18s^7 + 2.426D + 17s^8 + 3.512D + 14s^9 + 1.267D + 11s^{10}]}{[65011.218 + 1.534D + 11s + 4.074D + 14s^2 + 2.652D + 17s^3]} \quad (4.70)$$

$$+ 6.223D + 18s^4 + 3.657D + 19s^5 + 8.410D + 18s^6 + 4.843D + 17s^7 + 1.021D + 15s^8 + 5.273D + 11s^9 + 1949188.2s^{10}]$$

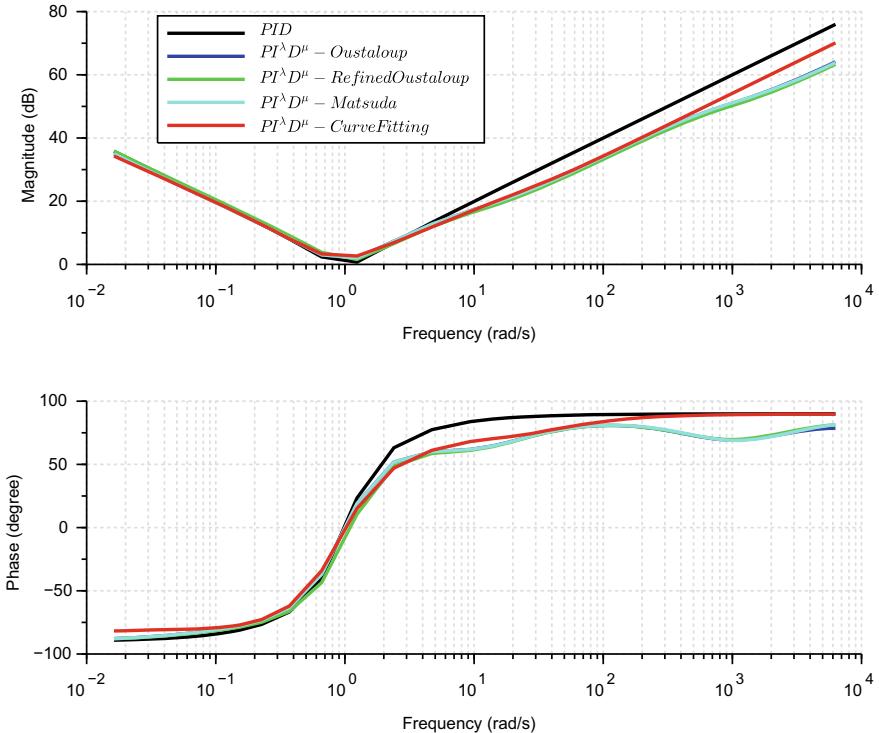


Fig. 4.16 Bode plot of $\text{PI}^\lambda \text{D}^\mu$ controller approximated using various techniques

$$\begin{aligned}
& [3.679D + 13 + 3.170D + 15s + 6.753D + 16s^2 + 2.307D + 17s^3 \\
& + 3.792D + 17s^4 + 3.514D + 17s^5 + 1.796D + 17s^6 \\
& + 4.594D + 16s^7 + 5.468D + 15s^8 + 2.581D + 14s^9 \\
& + 4.080D + 12s^{10}] \\
C_5(s) \approx & \frac{[4.459D + 10 + 4.818D + 13s + 3.652D + 15s^2 + 6.585D + 16s^3 \\
& + 1.583D + 17s^4 + 1.369D + 17s^5 + 4.809D + 16s^6 \\
& + 7.224D + 15s^7 + 4.185D + 14s^8 + 8.006D + 12s^9 \\
& + 4810788.6s^{10}]}{(4.459D + 10 + 4.818D + 13s + 3.652D + 15s^2 + 6.585D + 16s^3 + 1.583D + 17s^4 + 1.369D + 17s^5 + 4.809D + 16s^6 + 7.224D + 15s^7 + 4.185D + 14s^8 + 8.006D + 12s^9 + 4810788.6s^{10})} \quad (4.71)
\end{aligned}$$

Thus, the resultant Bode plot of $\text{PI}^\lambda \text{D}^\mu$ controller approximated using various techniques in comparison with conventional PID controller is shown in Fig. 4.16.

Similarly, the Bode plot of M. Tenoutit's $(\text{PI})^n$, $(\text{PD})^n$ and $(\text{PID})^n$ controllers given in Sect. 4.7.3 with gain parameters $K_p = K_i = K_d = 1$ and fractional-order parameter $n = 0.98$ in comparison with conventional PID controller can be obtained using the following commands. Here, the Oustaloup approximation technique has been used for the implementation of controllers.

```
--> exec('pid.sci'); exec('fpidTen.sci'); C1=pid(1,1,1);
C2=fpidTen(1,1,0,'o',0.98,5,[10^-5 10^5]);
C3=fpidTen(1,0,1,'o',0.98,5,[10^-5 10^5]);
C4=fpidTen(1,1,1,'o',0.98,5,[10^-5 10^5]);
bode([C1;C2;C3;C4],"rad");
legend('$PID$', '$(PI)^n$', '$(PD)^n$', '$(PID)^n$');
```

Thus, the resultant approximated transfer functions of $(PI)^n$, $(PD)^n$ and $(PID)^n$ controllers are given as follows:

$$C_2(s) \approx \frac{[79432.823 + 84095975 s + 9.639D + 08s^2 + 9.720D + 08s^3 + 92227983s^4 + 96464.898s^5 + s^6]}{[s + 96463.898s + 92131519s^3 + 8.798D + 08s^4 + 84016543s^5 + 79432.823s^6]} \quad (4.72)$$

$$C_3(s) \approx \frac{[79432.823 + 84095975 s + 9.639D + 08s^2 + 9.720D + 08s^3 + 92227983s^4 + 96464.898s^5 + s^6]}{[1 + 96463.898s + 92131519s^2 + 8.798D + 08s^3 + 84016543s^4 + 79432.823s^5]} \quad (4.73)$$

$$C_3(s) \approx \frac{[79432.823 + 84095975 s + 9.639D + 08s^2 + 1.056D + 09s^3 + 9.721D + 08s^4 + 92227984s^5 + 96464.898s^6 + s^7]}{[s + 96463.898s^2 + 92131519s^3 + 8.798D + 08s^4 + 84016543s^5 + 79432.823s^6]} \quad (4.74)$$

Therefore, the Bode plot of $(PI)^n$, $(PD)^n$ and $(PID)^n$ controllers in comparison with conventional PID controller is shown in Fig. 4.17.

In a similar way, the Bode plot of Luo's $[PI]^\lambda$ and $[PD]^\mu$ controllers given in Sect. 4.7.4 with gain parameters $K_p = K_i = K_d = 1$ and fractional-order parameters $\lambda = 0.98$ and $\mu = 0.85$ in comparison with conventional PI and PD controllers can be obtained as follows:

```
--> exec('pid.sci'); exec('fpiLuo.sci'); exec('fpdLuo.sci');
C1=pid(1,1,0); C2=fpiLuo(1,1,0.98);
C3=pid(1,0,1); C4=fpdLuo(1,1,0.85);
bode([C1;C2;C3;C4],"rad");
legend('$PI$', '$[PI]^{\lambda}$', '$PD$', '$[PD]^{\mu}$');
```

From the above commands, the transfer functions of $[PI]^\lambda$ and $[PD]^\mu$ controllers are obtained as follows:

$$C_2(s) = \frac{0.003332 - 0.0098s + 0.98s^2 + s^3}{s^3} \quad (4.75)$$

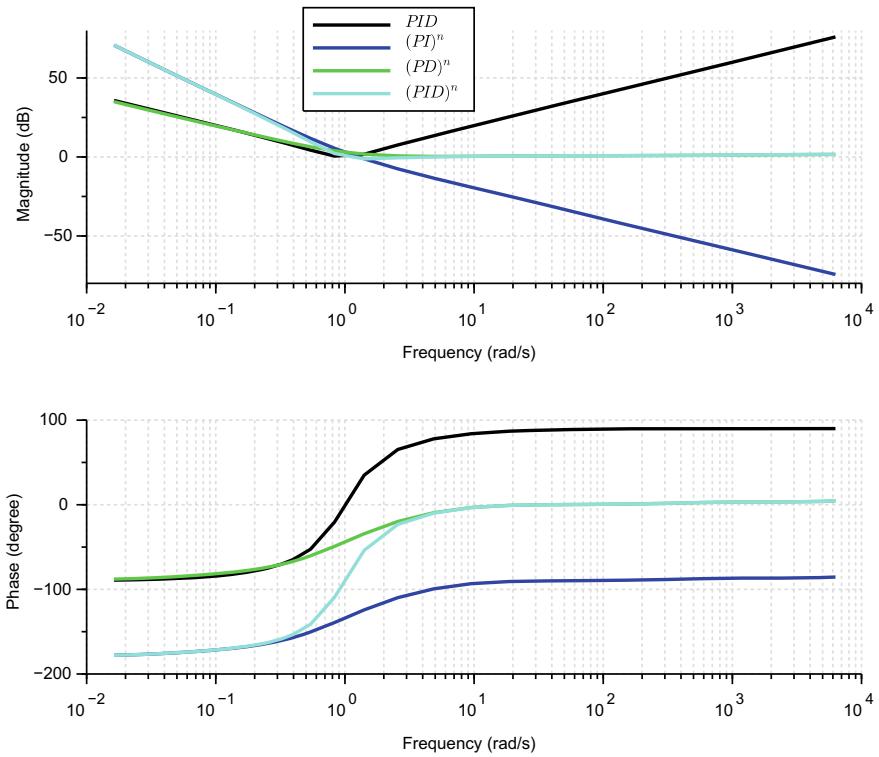


Fig. 4.17 Bode plot of $(PI)^n$, $(PD)^n$ and $(PID)^n$ controllers

$$C_4(s) = \frac{1 + 0.85s - 0.06375s^2 + 0.0244375s^3}{1} \quad (4.76)$$

Thus, the Bode plot of $[PI]^\lambda$ and $[PD]^\mu$ controllers in comparison with conventional PI and PD controllers is shown in Fig. 4.18.

On the other hand, the Bode plot of modified fractional-order PI and PID controllers given in Sects. 4.7.6 and 4.7.7 respectively with controller parameters $K_p = 1$, $K_i = 1$, $\alpha = 0.98$, $\eta = 0.1$ and $\mu = 0.85$ shown in Fig. 4.19 can be obtained using the following commands:

```
--> exec('pid.sci'); exec('modFPI.sci');
exec('modFPIss.sci'); exec('modFPID.sci');
C1=pid(1,1,0); C2=modFPI(1,2,0.98,'o',5,[10^-5 10^5]);
C3=modFPIss(1,1,0.98,0.1,'o',5,[10^-5 10^5]);
C4=pid(1,1,1); C5=modFPID(1,1,0.85,'o',5,[10^-5 10^5]);
bode([C1;C2;C3;C4;C5], "rad");
legend('PI','mod FPI','mod FPI ss','PID','mod FPID');
```

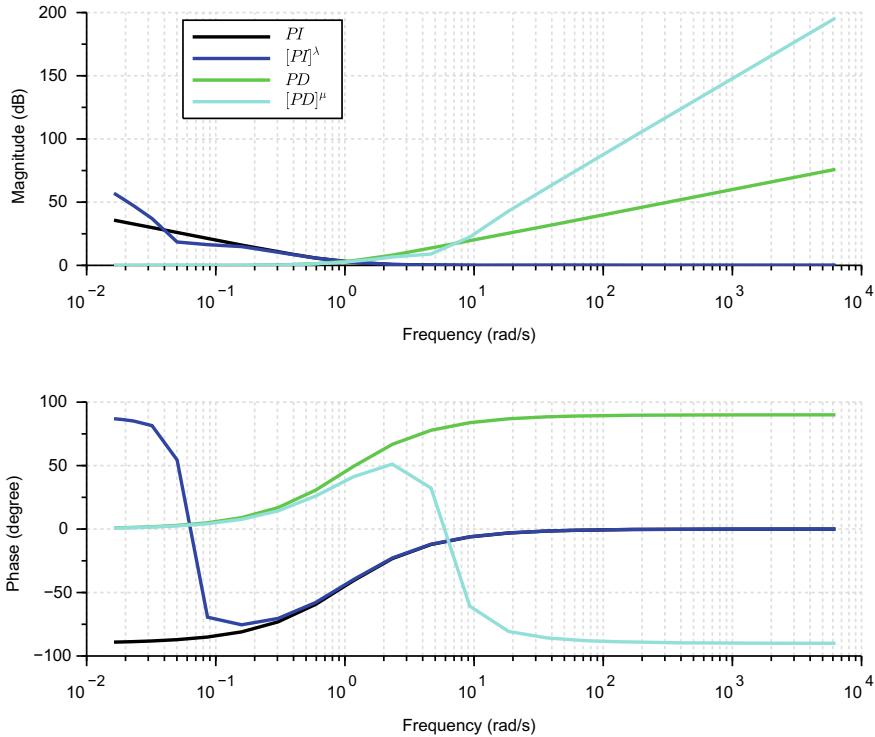


Fig. 4.18 Bode plot of $[PI]^\lambda$ and $[PD]^\mu$ controllers

From the above commands, the approximated transfer functions of modified fractional-order PI and PID controllers are obtained as follows:

$$C_2(s) \approx \frac{[79432.823 + 84175408s + 1.048D + 09s^2 + 1.852D + 09s^3 + 1.844D + 08s^4 + 192928.8s^5 + 2s^6]}{[1 + 96463.898s + 92131519s^2 + 8.798D + 08s^3 + 84016543s^4 + 79432.823s^5]} \quad (4.77)$$

$$C_3(s) \approx \frac{[7943.2823 + 8489030.4s + 1.805D + 08s^2 + 1.061D + 09s^3 + 9.812D + 08s^4 + 92237630s^5 + 96464.998s^6 + s^7]}{[s + 96463.898s^2 + 92131519s^3 + 8.798D + 08s^4 + 84016543s^5 + 79432.823s^6]} \quad (4.78)$$

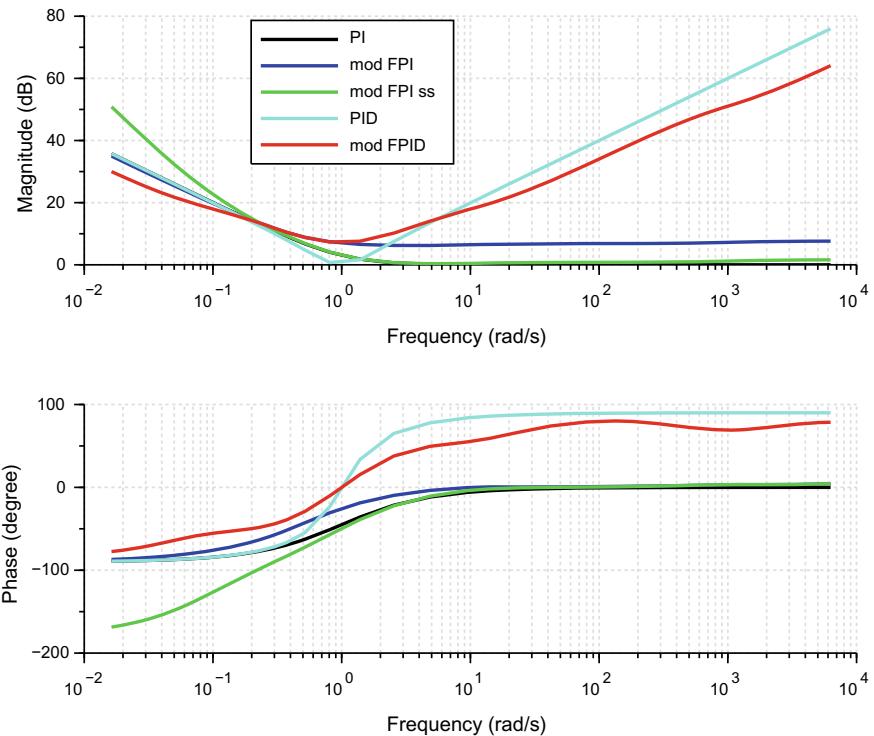


Fig. 4.19 Bode plot of modified fractional-order PI and PID controllers

$$\begin{aligned}
 C_5(s) \approx & \frac{[5.624D + 12 + 2.412D + 16s + 3.485D + 19s^2 + 1.749D + 22s^3 \\
 & + 7.692D + 23s^4 + 1.228D + 25s^5 + 7.702D + 25s^6 \\
 & + 1.347D + 26s^7 + 8.499D + 25s^8 + 1.703D + 25s^9 \\
 & + 1.305D + 24s^{10} + 3.533D + 22s^{11} + 9.329D + 19s^{12} \\
 & + 8.139D + 16s^{13} + 2.352D + 13s^{14} + 3.163D + 08s^{15}]}{[3.162D + 08 + 2.352D + 13s + 8.120D + 16s^2 + 9.277D + 19s^3} \\
 & + 3.486D + 22s^4 + 1.166D + 24s^5 + 1.317D + 25s^6 \\
 & + 4.927D + 25s^7 + 1.647D + 25s^8 + 1.861D + 24s^9 \\
 & + 6.958D + 22s^{10} + 2.321D + 20s^{11} + 2.611D + 17s^{12} \\
 & + 9.636D + 13s^{13} + 2.569D + 09s^{14} + 17782.794s^{15}]
 \end{aligned} \tag{4.79}$$

As a final analysis of this study, the Bode plot of IMC based fractional-order PID given in Sect. 4.7.5 with controller parameters $K_p = 1$, $K_i = 1$, $K_d = 1$, $\gamma = 0.98$ and $\alpha = 0.98$ in comparison with conventional PID can be obtained as follows:

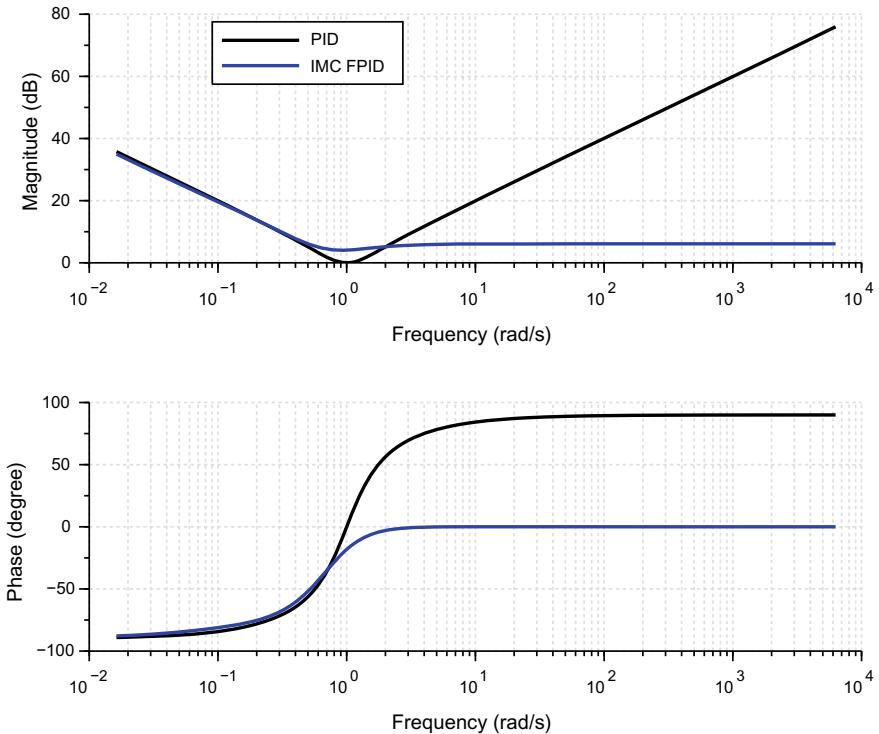


Fig. 4.20 Bode plot of IMC based fractional-order PID

```
--> exec('pid.sc'); exec('IMCfpid.sc'); C1=pid(1,1,1);
C2=IMCfpid(1,1,1,0.98,0.98,'o',5,[10^-5 10^5]);
bode([C1;C2], "rad"); legend('PID', 'IMC FPID');
```

Thus, the resultant Bode plot from the above commands is shown in Fig. 4.20. Furthermore, the approximated transfer function obtained as follows:

$$C_2(s) \approx \frac{[5.012D + 14 + 1.592D + 18s + 1.702D + 21s^2 + 6.321D + 23s^3 + 2.019D + 25s^4 + 2.383D + 26s^5 + 1.167D + 27s^6 + 2.061D + 27s^7 + 1.868D + 27s^8 + 4.459D + 26s^9 + 4.117D + 25s^{10} + 1.369D + 24s^{11} + 3.825D + 21s^{12} + 3.716D + 18s^{13} + 1.232D + 15s^{14} + 1.249D + 10s^{15}]}{[6.310D + 09 + 6.220D + 14s + 1.877D + 18s^2 + 1.932D + 21s^3 + 6.914D + 23s^4 + 2.077D + 25s^5 + 2.243D + 26s^6 + 9.277D + 26s^7 + 9.175D + 26s^8 + 2.207D + 26s^9 + 2.039D + 25s^{10} + 6.776D + 23s^{11} + 1.893D + 21s^{12} + 1.839D + 18s^{13} + 6.096D + 14s^{14} + 6.183D + 09s^{15}]} \quad (4.80)$$

4.9.2.2 Control of Non-linear Spherical Tank System

The transfer function model of the non-linear spherical tank system reported in [48] is given as

$$G_1(s) = \frac{3.62}{330.46s + 1} \quad (4.81)$$

The $G_1(s)$ can be defined in Scilab as follows:

```
--> s=poly(0,'s'); s=syslin('c',s/((0*s)+1));
G=(3.6215/(330.46*s+1));
```

The tuned PID and fractional-order PID based controllers are defined as follows:

```
--> Kp=6.81; Ki=1.01; Kd=1.91;
lam=0.98; mu=0.75; n=0.05;
alp=lam; gam=0.1*Kd;
Ti=Kp/Ki; Kc=Kp/Ti; Td=Kd/Kp;
```

The selected Oustaloup approximation parameters are specified as follows:

```
--> N=5; wl=10^-3; wh=10^3;
```

Thus, the respective controllers PID, PI, $P I^\lambda D^\mu$, PI^λ , $(PID)^n$, IMC PID, Mod FPID and $[PI]^\lambda$ as given in Sect. 4.7 are defined as

$$C_{PID}(s) = 6.81 + \frac{1.01}{s} + 1.91s \quad (4.82)$$

$$C_{PI}(s) = 6.81 + \frac{1.01}{s} \quad (4.83)$$

$$C_{PI^\lambda D^\mu}(s) = 6.81 + \frac{1.01}{s^{0.98}} + 1.91s^{0.75} \quad (4.84)$$

$$C_{PI^\lambda}(s) = 6.81 + \frac{1.01}{s^{0.98}} \quad (4.85)$$

$$C_{(PID)^n}(s) = \left[6.81 + \frac{1.01}{s} + 1.91s \right]^{0.05} \quad (4.86)$$

$$C_{IMC\ PID}(s) = 6.81 + \frac{1.01}{s^{0.98}} + \frac{1.91s^{0.98}}{0.19s^{0.98} + 1} \quad (4.87)$$

$$C_{Mod\ FPID}(s) = 1.01 \frac{(1 + 6.74s^{0.75})^2}{s^{0.75}} \quad (4.88)$$

$$C_{[PI]^\lambda}(s) = \left[6.81 + \frac{1.01}{s} \right]^{0.98} \quad (4.89)$$

The designed fractional-order controllers will be implemented using developed functions in Sect. 4.7 as follows:

```
--> exec('pid.sci'); C1=pid(Kp,Ki,Kd); C11=pid(Kp,Ki,0);
exec('fpid.sci');
C2=fpid(Kp,Ki,Kd,[lam mu],'o',N,[wl wh]);
exec('fpi.sci'); C21=fpi(Kp,Ki,1, lam,'o',N,[wl wh]);
exec('fpidTen.sci');
C3= fpidTen(Kp,Ki,0,'o',n,N,[wl wh]);
exec('IMCfpid.sci');
C4=IMCfpid(Kp,Ki,Kd,gam,alp,'o',N,[wl wh]);
exec('modFPID.sci'); C5=modFPID(Kc,Ti,mu,'o',N,[wl wh]);
exec('fpiLuo.sci'); C6=fpiLuo(Kp,Ki,1, lam);
```

The closed-loop transfer functions of $G_1(s)$ with controllers defined above are obtained as

```
--> Gcl1=(G*C1)/(1+(G*C1)); Gcl11=(G*C11)/(1+(G*C11));
Gcl2=(G*C2)/(1+(G*C2)); Gcl21=(G*C21)/(1+(G*C21));
Gcl3=(G*C3)/(1+(G*C3)); Gcl4=(G*C4)/(1+(G*C4));
Gcl5=(G*C5)/(1+(G*C5)); Gcl6=(G*C6)/(1+(G*C6));
```

The step response of the system with all the designed controllers is obtained using the following commands. Thus, the resultant plot is shown in Fig. 4.21. From the response, it can be seen that the fractional-order controllers can be effectively implemented in Scilab.

```
--> t=0:0.01:200; scf();
plot([t',t',t',t',t',t',t'],...
[csim('step',t,Gcl1)',csim('step',t,Gcl11)',...
csim('step',t,Gcl2)',csim('step',t,Gcl21)',...
csim('step',t,Gcl3)',csim('step',t,Gcl4)',...
csim('step',t,Gcl5)',csim('step',t,Gcl6)'']);
legend('PID','PI','$\mathit{PI}^{\lambda}$',...
'$\mathit{PI}^{\lambda}\mu$','$\mathit{PID}^n$',...
'IMC PID','Mod FPID','$\mathit{PI}^{\lambda}$',...
"in_lower_right");
xlabel("Time (sec)"); ylabel("Amplitude");
```

Furthermore, the numerical assessment of the figure given in Table 4.6 can be obtained with the use of `stepinfo.sci` function as follows:

```
--> exec('stepinfo.sci'); yf=1;
Gcl1s=stepinfo(Gcl1,t,yf); Gcl11s=stepinfo(Gcl11,t,yf);
Gcl2s=stepinfo(Gcl2,t,yf); Gcl21s=stepinfo(Gcl21,t,yf);
Gcl3s=stepinfo(Gcl3,t,yf); Gcl4s=stepinfo(Gcl4,t,yf);
Gcl5s=stepinfo(Gcl5,t,yf); Gcl6s=stepinfo(Gcl6,t,yf);
```

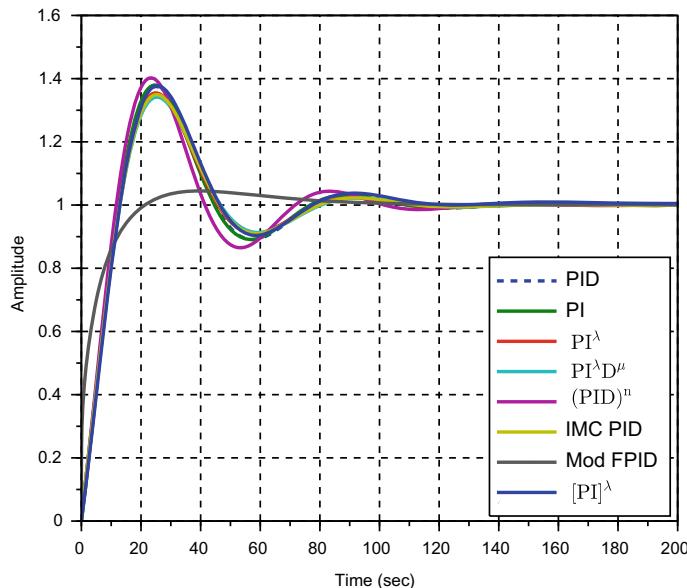


Fig. 4.21 Step response of $G_1(s)$ with various fractional-order controllers

Table 4.6 Step info of $G_1(s)$ with various controllers

Controller	t_r (s)	t_s (s)	%OS
PID	9.8296	99.6926	37.4939
PI	9.7107	98.3596	37.9709
PI^λ	9.8950	96.5135	35.4722
$PI^\lambda D^\mu$	10.2393	97.4630	34.1414
$(PID)^n$	9.1032	94.8763	40.2484
IMC PID	10.2300	97.5639	34.9400
Mod FPID	12.0992	71.0458	4.4632
$[PI]^\lambda$	10.0441	105.1809	37.7394

From the table, it can be seen that the Mod FPID controller has the least overshoot of 4.46% and the fastest settling time of 71.04 s. However, for rise time $(PID)^n$ controller has the fastest rise time of 9.10 s. Furthermore, the open-loop Bode plots of the system with various controllers can be obtained using the following commands. Thus, the resulting plot is shown in Fig. 4.22.

```
--> scf();
bode([G*C1;G*C11;G*C21;G*C2;G*C3;G*C4;G*C5;G*C6],wl,wh);
legend('PID','PI','$\mathit{PI}^\lambda$',...
'$\mathit{PI}^\lambda D^\mu$','$\mathit{(PID)}^n$',...
'IMC PID','Mod FPID','$\mathit{[PI]}^\lambda$',...
"in_lower_left");
```

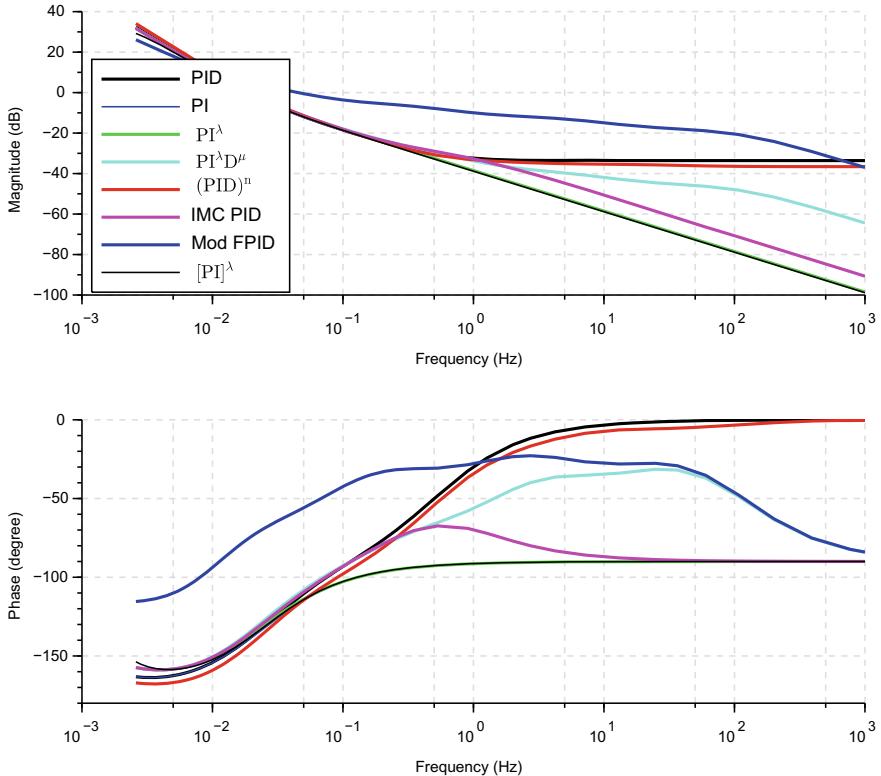


Fig. 4.22 Bode plot of $G_1(s)$ with various fractional-order controllers

Similarly, the other frequency-domain plots like Nyquist and Nichols are obtained using the following commands. Thus, the Nyquist and Nichols plot of $G_1(s)$ with various controllers are shown in Figs. 4.23 and 4.24 respectively.

```
--> scf();
nyquist([G*C1;G*C11;G*C21;G*C2;G*C3;G*C4;G*C5;G*C6]);
legend('PID','PI','$\mathbf{PI}^\lambda$',...
'$\mathbf{PI}^\lambda \mathbf{D}^\mu$','$\mathbf{(PID)}^n$',...
'IMC PID','Mod FPID','$\mathbf{[PI]}^\lambda$',...
"in_lower_left");

scf();
black([G*C1;G*C11;G*C21;G*C2;G*C3;G*C4;G*C5;G*C6]);
legend('PID','PI','$\mathbf{PI}^\lambda$',...
'$\mathbf{PI}^\lambda \mathbf{D}^\mu$','$\mathbf{(PID)}^n$',...
'IMC PID','Mod FPID','$\mathbf{[PI]}^\lambda$',...
"in_lower_left");
```

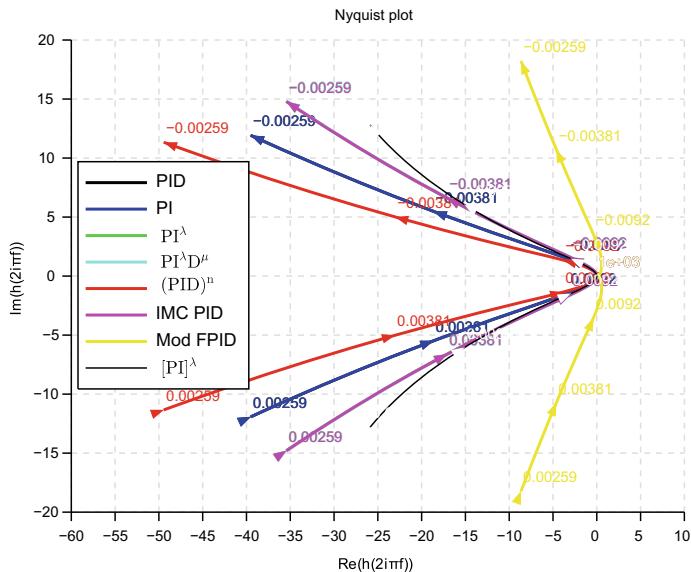


Fig. 4.23 Nyquist plot of $G_1(s)$ with various fractional-order controllers

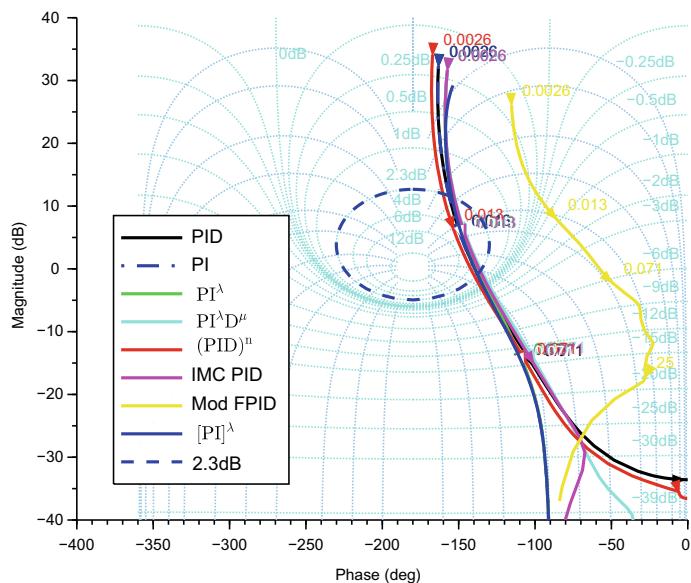


Fig. 4.24 Nichols plot of $G_1(s)$ with various fractional-order controllers

4.9.2.3 Control of Unstable Bioreactor Process

To further evaluate the performance of the proposed Scilab based fractional-order controllers, an unstable Bioreactor Process reported in [48] is selected. Thus, the transfer function model of the reactor process is given by

$$G_2(s) = \frac{-0.9951s - 0.2985}{s^2 + 0.1302s - 0.0509} \quad (4.90)$$

From Eq. (4.90), it can be seen that the process is unstable with a pole at 0.1697. For the control of this process, the tuned PID and fractional-order PID parameters are $K_p = -0.54$, $K_i = -0.07$, $K_d = -0.05$, $\lambda = \alpha = 0.90$, $\mu = 0.85$ and $n = 0.1$. Thus, the controller transfer functions of PID, $P I^\lambda D^\mu$, $(PID)^n$, IMC PID and Mod FPID are given as

$$C_{PID}(s) = -\left(0.54 + \frac{0.07}{s} + 0.05s\right) \quad (4.91)$$

$$C_{P I^\lambda D^\mu}(s) = -\left(0.54 + \frac{0.07}{s^{0.90}} + 0.05s^{0.85}\right) \quad (4.92)$$

$$C_{(PID)^n}(s) = -\left(0.54 + \frac{0.07}{s} + 0.05s\right)^{0.1} \quad (4.93)$$

$$C_{IMC\ PID}(s) = -\left(0.54 + \frac{0.07}{s^{0.90}} + \frac{0.05s^{0.90}}{0.0005s^{0.90} + 1}\right) \quad (4.94)$$

$$C_{Mod\ FPID}(s) = 7.71 \frac{(1 - 0.07s^{0.85})^2}{s^{0.85}} \quad (4.95)$$

The designed controllers will be implemented for the control of unstable bioreactor process in Scilab as follows:

```
--> s=poly(0,'s'); s=syslin('c',s/((0*s)+1));
G=(-0.9951*s-0.2985)/(s^2+0.1302*s-0.0509));
Kp=-0.5374; Ki=-0.0702; Kd=-0.0532;
lam=0.9; mu=0.85; n=0.1; alp=lam; gam=0.01*Kd;
Ti=Kp/Ki; Kc=Kp/Ti; Td=Kd/Kp; N=5; wl=10^-3; wh=10^3;
exec('pid.sci'); C1=pid(Kp,Ki,Kd); exec('fpid.sci');
C2=fpid(Kp,Ki,Kd,[lam mu],'o',N,[wl wh]);
exec('fpidTen.sci'); exec('IMCfpid.sci');
C3=fpidTen(Kp,Ki,Kd,'o',n,N,[wl wh]);
C4=IMCfpid(Kp,Ki,Kd,gam,alp,'o',N,[wl wh]);
exec('modFPID.sci');
C5=modFPID(Kc,Ti,mu,'o',N,[wl wh]);
Gc11=(G*C1)/(1+(G*C1)); Gc12=(G*C2)/(1+(G*C2));
Gc13=(G*C3)/(1+(G*C3)); Gc14=(G*C4)/(1+(G*C4));
Gc15=(G*C5)/(1+(G*C5));
```

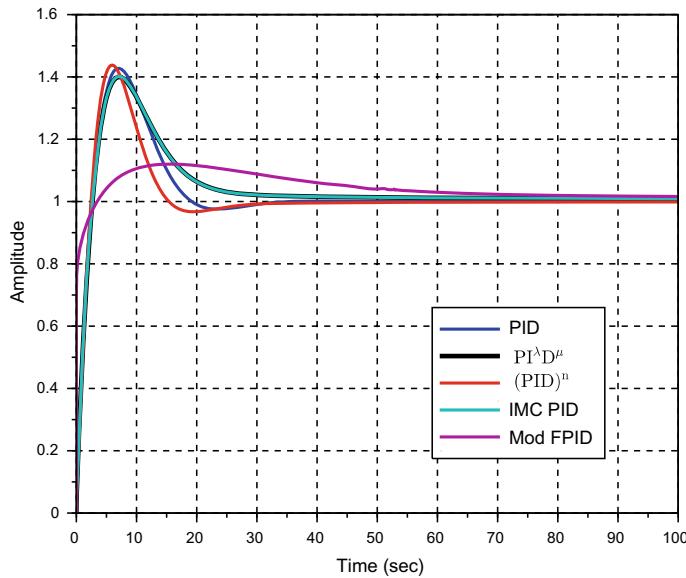


Fig. 4.25 Step response of $G_2(s)$ with various fractional-order PID controllers

Therefore, the step response of bioreactor with the above-designed fractional-order PID controllers can be obtained in Scilab as follows:

```
--> t=0:0.01:100; scf();
plot([t',t',t',t',t'],[csim('step',t,Gcl1)',...
csim('step',t,Gcl2)',csim('step',t,Gcl3)',...
csim('step',t,Gcl4)',csim('step',t,Gcl5')])
legend('PID','$\mathbf{PI}^\lambda \mathbf{D}^\mu$',...
'$\mathbf{(PID)}^n$','IMC PID','Mod FPID',...
"in_lower_right"); xgrid;
xlabel("Time (sec)"); ylabel("Amplitude");
```

Thus, the resultant step response plot is shown in Fig. 4.25. Furthermore, the assessment of the figure numerically in terms of t_r , t_s and %OS is given in Table 4.7 is obtained as follows:

```
--> yf=1; Gcl1s=stepinfo(Gcl1,t,yf);
Gcl2s=stepinfo(Gcl2,t,yf); Gcl3s=stepinfo(Gcl3,t,yf);
Gcl4s=stepinfo(Gcl4,t,yf); Gcl5s=stepinfo(Gcl5,t,yf);
```

From the assessment, it can be seen that the designed fractional-order PID controllers can be successfully implemented for this unstable process using Scilab. It can also be seen that the Mod FPID controller has outperformed all the other controllers with least overshoot of 12.01% and the fastest rise time of 1.22 s. However, for settling time the $(\text{PID})^n$ controller has the fastest settling time of 24.53 s.

Table 4.7 Step info of $G_2(s)$ with various controllers

Controller	t_r (s)	t_s (s)	%OS
PID	2.0840	26.6758	42.7678
$\text{PI}^\lambda \text{D}^\mu$	2.1322	31.0234	39.9924
$(\text{PID})^n$	1.9186	24.5391	43.7835
IMC PID	2.1436	30.8774	40.1460
Mod FPID	1.2216	78.6227	12.0132

Similarly, the frequency response plots like Bode, Nyquist, and Nichols for the process with the designed fractional-order PID controllers can be obtained as follows:

```
--> scf(); bode([G*C1;G*C2;G*C3;G*C4;G*C5],wl,wh);
legend('PID','$\mathbf{\text{PI}}^{\lambda} \mathbf{\text{D}}^{\mu}$',...
'$\mathbf{\text{PI}}^n$','IMC PID','Mod FPID',...
"in_lower_right");
scf(); nyquist([G*C1;G*C2;G*C3;G*C4;G*C5]);
legend('PID','$\mathbf{\text{PI}}^{\lambda} \mathbf{\text{D}}^{\mu}$',...
'$\mathbf{\text{PI}}^n$','IMC PID','Mod FPID');
```

Thus, the resultant Bode and Nyquist are shown in Figs. 4.26 and 4.27 respectively. Thus, in all the cases it can be concluded that the fractional-order controllers can be effectively designed, implemented and analyzed using Scilab.

4.9.3 Fractional-order Filters

This section analyses the designed fractional-order filters for various values of parameters. In all the cases, the pole frequency (ω_c) is chosen as 5 rad/s. Furthermore, the approximation is done using Oustaloup's technique with parameters $N = 5$, $\omega_l = 10^{-3}$ and $\omega_h = 10^3$.

4.9.3.1 Analysis of Fractional-order Low-Pass Filter

The Bode responses of the fractional-order low-pass for various values of α between (0, 1) shown in Fig. 4.28 can be obtained using the commands

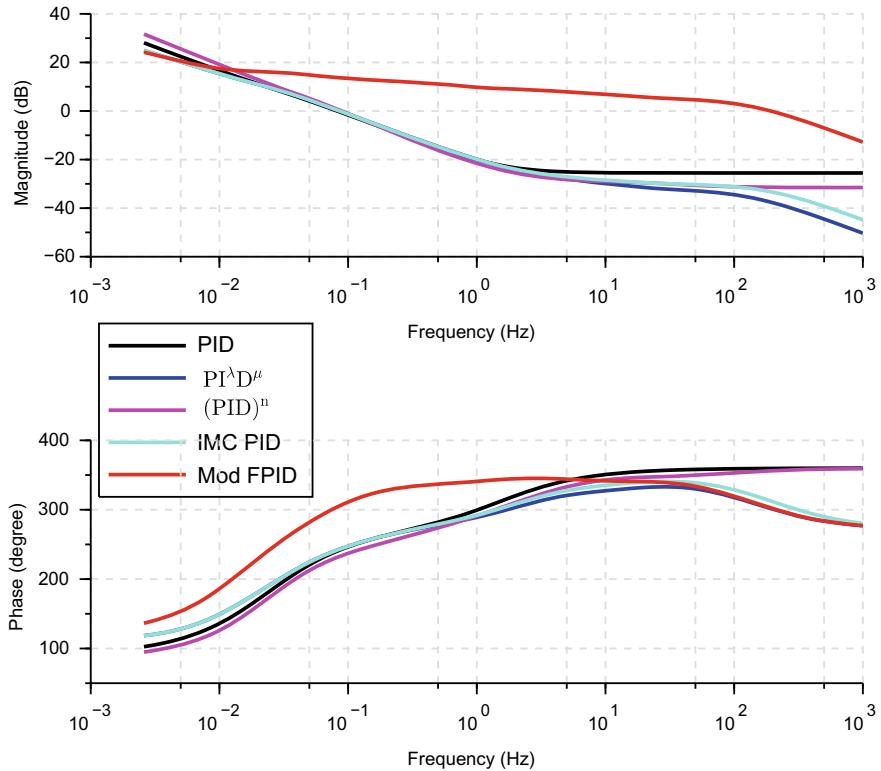


Fig. 4.26 Bode plot of $G_2(s)$ with various fractional-order PID controllers

```
--> omegaC=5; N=5; wl=10^-3; wh=10^3;
exec('foLPF.sci'); i=1;
for alpha=0.1:0.1:0.9
H(i)=foLPF(alpha,omegaC,'o',N,[wl wh]);
i=i+1;
end
scf();
bode([H(1);H(2);H(3);H(4);H(5);H(6);H(7);H(8);H(9)],...
wl,wh);
legend('$\alpha=0.1$','$\alpha=0.2$','$\alpha=0.3$',...
'$\alpha=0.4$','$\alpha=0.5$','$\alpha=0.6$',...
'$\alpha=0.7$','$\alpha=0.8$','$\alpha=0.9$',...
"in_lower_left");
```

The impulse and step responses of the fractional-order low-pass filter for $0 < \alpha < 1$ shown in Fig. 4.29 can be obtained as follows:

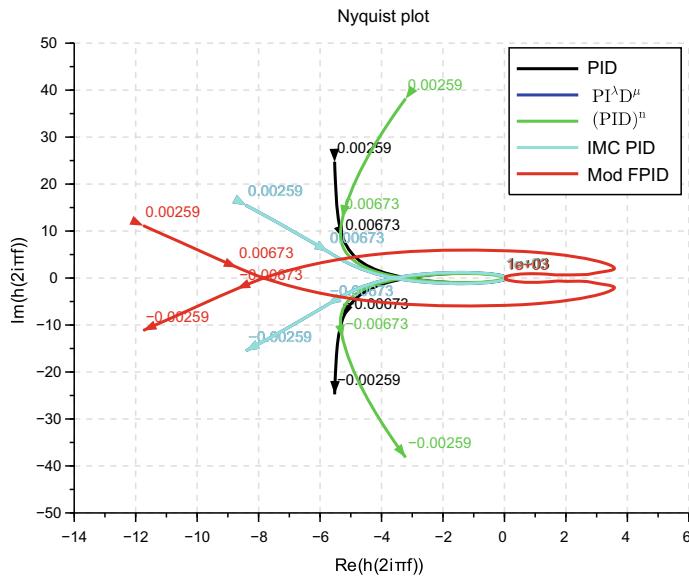


Fig. 4.27 Nyquist plot of $G_2(s)$ with various fractional-order PID controllers

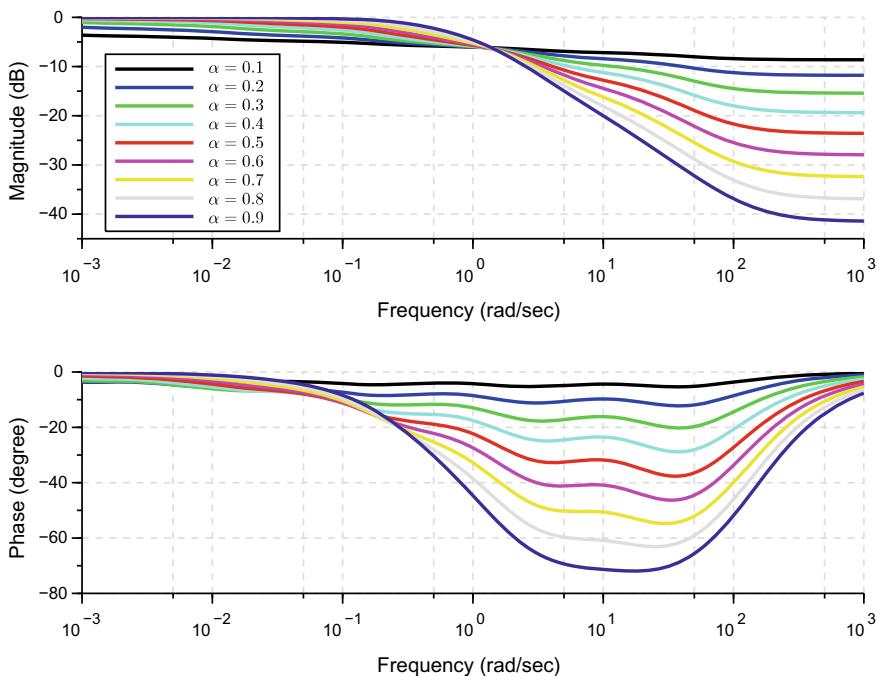


Fig. 4.28 Bode response of fractional-order low-pass filter for $0 < \alpha < 1$

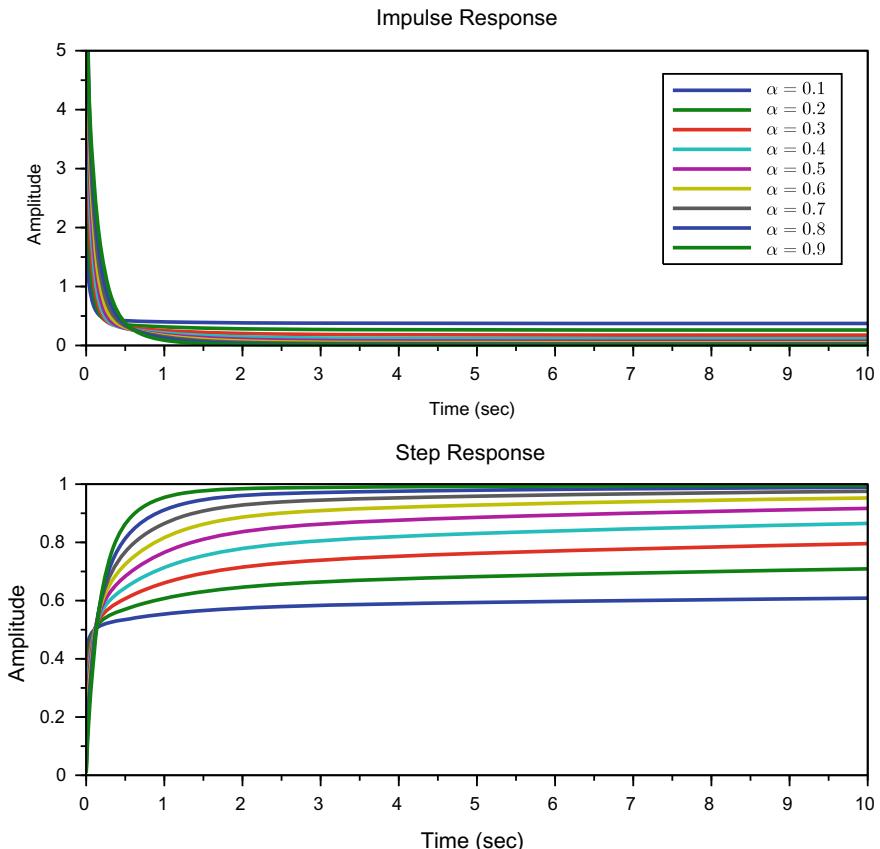


Fig. 4.29 Impulse and step responses of foLPF for $0 < \alpha < 1$

```
--> t=0:0.01:10; scf(); subplot(211);
plot([t',t',t',t',t',t',t',t',t'],...
[csim('impulse',t,H(1)'),csim('impulse',t,H(2)'),...
csim('impulse',t,H(3)'),csim('impulse',t,H(4)'),...
csim('impulse',t,H(5)'),csim('impulse',t,H(6)'),...
csim('impulse',t,H(7)'),csim('impulse',t,H(8)'),...
csim('impulse',t,H(9))']);
legend('$\alpha=0.1$', '$\alpha=0.2$', '$\alpha=0.3$', ...
'$\alpha=0.4$', '$\alpha=0.5$', '$\alpha=0.6$', ...
'$\alpha=0.7$', '$\alpha=0.8$', '$\alpha=0.9$', ...
"in_lower_left");
a=gca(); a.data_bounds=[0 0;10 5];
title("Impulse Response");
xlabel("Time (sec)"); ylabel("Amplitude");
```

```

subplot(212);
plot([t',t',t',t',t',t',t',t'],...
[csim('step',t,H(1))',csim('step',t,H(2))',...
csim('step',t,H(3))',csim('step',t,H(4))',...
csim('step',t,H(5))',csim('step',t,H(6))',...
csim('step',t,H(7))',csim('step',t,H(8))',...
csim('step',t,H(9))']); title("Step Response");
xlabel("Time (sec)"); ylabel("Amplitude");

```

The Bode responses of the fractional-order low-pass for various values of α between (1, 2) shown in Fig. 4.30 can be obtained using the commands

```

--> exec('foLPF.sci'); i=1;
for alpha=1.1:0.1:1.8
H(i)=foLPF(alpha,omegac,'o',N,[wl wh]);
i=i+1;
end
clf();
bode([H(1);H(2);H(3);H(4);H(5);H(6);H(7);H(8)]);
legend('$\alpha=1.1$', '$\alpha=1.2$', ...
'$\alpha=1.3$', '$\alpha=1.4$', '$\alpha=1.5$', ...
'$\alpha=1.6$', '$\alpha=1.7$', '$\alpha=1.8$', ...
"in_lower_left");

```

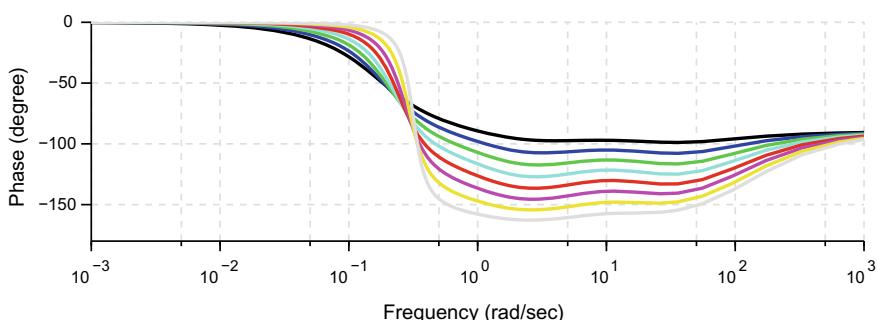
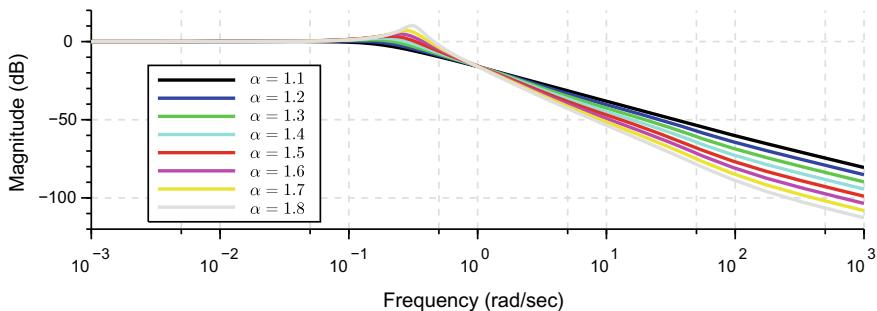


Fig. 4.30 Bode response of fractional-order low-pass filter for $1 < \alpha < 2$

Similarly, the impulse and step responses of the filter for $1 < \alpha < 2$ shown in Fig. 4.31 can be obtained as follows:

```
--> t=0:0.01:10; scf(); subplot(211);
plot([t',t',t',t',t',t',t',t'],...
[csim('impulse',t,H(1))',csim('impulse',t,H(2))',...
csim('impulse',t,H(3))',csim('impulse',t,H(4))',...
csim('impulse',t,H(5))',csim('impulse',t,H(6))',...
csim('impulse',t,H(7))',csim('impulse',t,H(8))']);
legend('$\alpha=1.1$', '$\alpha=1.2$', '$\alpha=1.3$',...
'$\alpha=1.4$', '$\alpha=1.5$', '$\alpha=1.6$',...
'$\alpha=1.7$', '$\alpha=1.8$', "in_lower_left");
title("Impulse Response");
xlabel("Time (sec)"); ylabel("Amplitude");

subplot(212);
plot([t',t',t',t',t',t',t',t'],...
[csim('step',t,H(1))',csim('step',t,H(2))',...
csim('step',t,H(3))',csim('step',t,H(4))',...
csim('step',t,H(5))',csim('step',t,H(6))',...
csim('step',t,H(7))',csim('step',t,H(8))']);
title("Step Response");
xlabel("Time (sec)"); ylabel("Amplitude");
```

4.9.3.2 Analysis of Fractional-order High-Pass Filter

Similarly, the Bode responses of the fractional-order high-pass for various values of α between (0, 1) shown in Fig. 4.32 can be obtained using the commands

```
--> omegaC=5; N=5; wl=10^-3; wh=10^3;
exec('foHPF.sci'); i=1;
for alpha=0.1:0.1:0.9
    H(i)=foHPF(alpha,omegaC,'o',N,[wl wh]);
    i=i+1;
end
scf();
bode([H(1);H(2);H(3);H(4);H(5);H(6);H(7);H(8);H(9)]',...
wl,wh)
legend('$\alpha=0.1$', '$\alpha=0.2$', '$\alpha=0.3$',...
'$\alpha=0.4$', '$\alpha=0.5$', '$\alpha=0.6$',...
'$\alpha=0.7$', '$\alpha=0.8$', '$\alpha=0.9$',...
"in_lower_left");
```

Furthermore, the impulse and step response the filter for order $0 < \alpha < 1$ shown in Fig. 4.33 can be obtained using the following commands

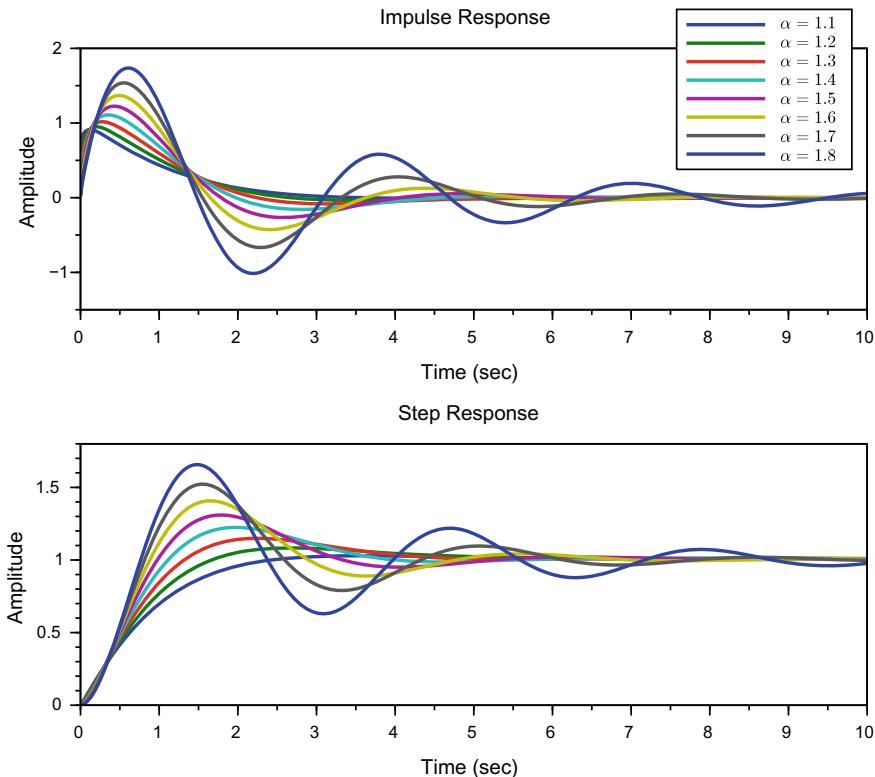


Fig. 4.31 Impulse and step responses of foLPF for $1 < \alpha < 2$

```
--> t=0:0.01:10; scf(); subplot(211);
plot([t',t',t',t',t',t',t',t'],...
[csim('impulse',t,H(1)'),csim('impulse',t,H(2)'),...
csim('impulse',t,H(3)'),csim('impulse',t,H(4)'),...
csim('impulse',t,H(5)'),csim('impulse',t,H(6)'),...
csim('impulse',t,H(7)'),csim('impulse',t,H(8)'),...
csim('impulse',t,H(9))']);
legend('$\alpha=0.1$', '$\alpha=0.2$', '$\alpha=0.3$', ...
'$\alpha=0.4$', '$\alpha=0.5$', '$\alpha=0.6$', ...
'$\alpha=0.7$', '$\alpha=0.8$', '$\alpha=0.9$', ...
"in_lower_left");
a=gca(); a.data_bounds=[0 0;10 5];
title("Impulse Response");
xlabel("Time (sec)"); ylabel("Amplitude");
```

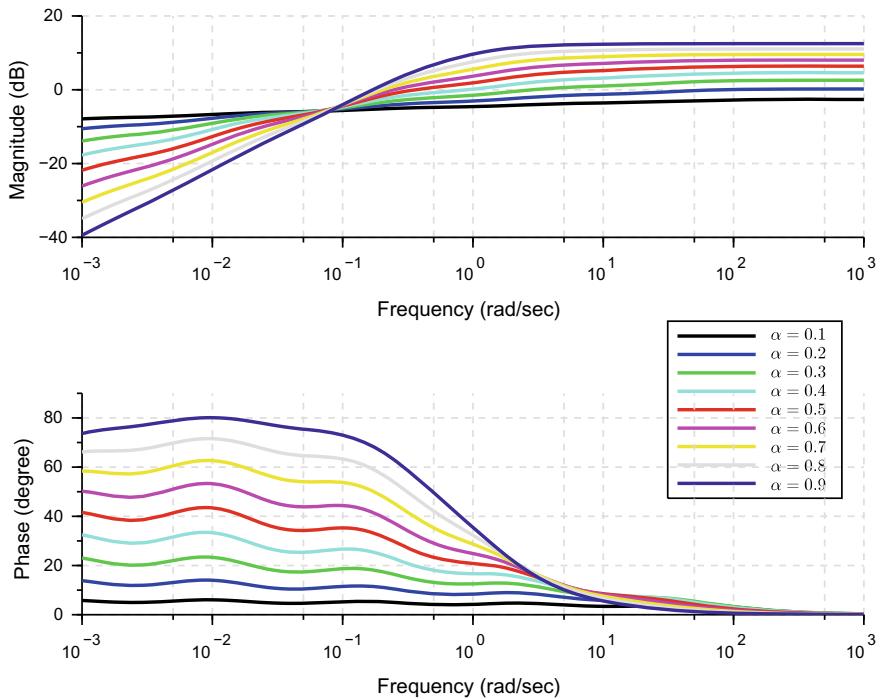


Fig. 4.32 Bode response of fractional-order high-pass filter for $0 < \alpha < 1$

```
subplot(212); plot([t',t',t',t',t',t',t',t'],...
[csim('step',t,H(1)'),csim('step',t,H(2)'),...
csim('step',t,H(3)'),csim('step',t,H(4)'),...
csim('step',t,H(5)'),csim('step',t,H(6)'),...
csim('step',t,H(7)'),csim('step',t,H(8)'),...
csim('step',t,H(9))']); title("Step Response");
xlabel("Time (sec)"); ylabel("Amplitude");
```

The Bode responses of the fractional-order high-pass filter for various values of α between (1, 2) shown in Fig. 4.34 can be obtained using the commands

```
--> exec('foHPF.sci'); i=1;
for alpha=1.1:0.1:1.8
    H(i)=foHPF(alpha,omegaC,'o',N,[wl wh]);
    i=i+1;
end
scf();
bode([H(1);H(2);H(3);H(4);H(5);H(6);H(7);H(8);])
legend('$\alpha=1.1$','$\alpha=1.2$',...
'$\alpha=1.3$','$\alpha=1.4$','$\alpha=1.5$',...
'$\alpha=1.6$','$\alpha=1.7$','$\alpha=1.8$',...
"in_lower_left");
```

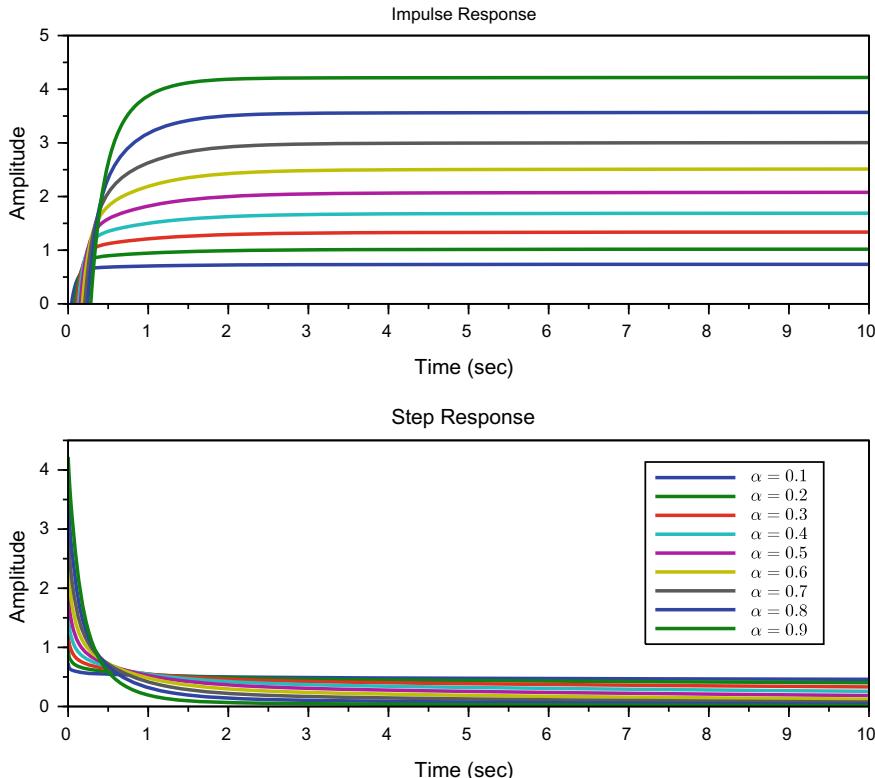


Fig. 4.33 Impulse and step responses of foHPF for $0 < \alpha < 1$

Furthermore, for this order of $\alpha \in (1, 2)$, the impulse and step responses of the filter shown in Fig. 4.35 can be obtained using the commands

```
--> t=0:0.01:10; scf(); subplot(211);
plot([t',t',t',t',t',t',t',t'],...
[csim('impulse',t,H(1)'),csim('impulse',t,H(2)'),...
csim('impulse',t,H(3)'),csim('impulse',t,H(4)'),...
csim('impulse',t,H(5)'),csim('impulse',t,H(6)'),...
csim('impulse',t,H(7)'),csim('impulse',t,H(8))'])
title("Impulse Response");
xlabel("Time (sec)"); ylabel("Amplitude");
```

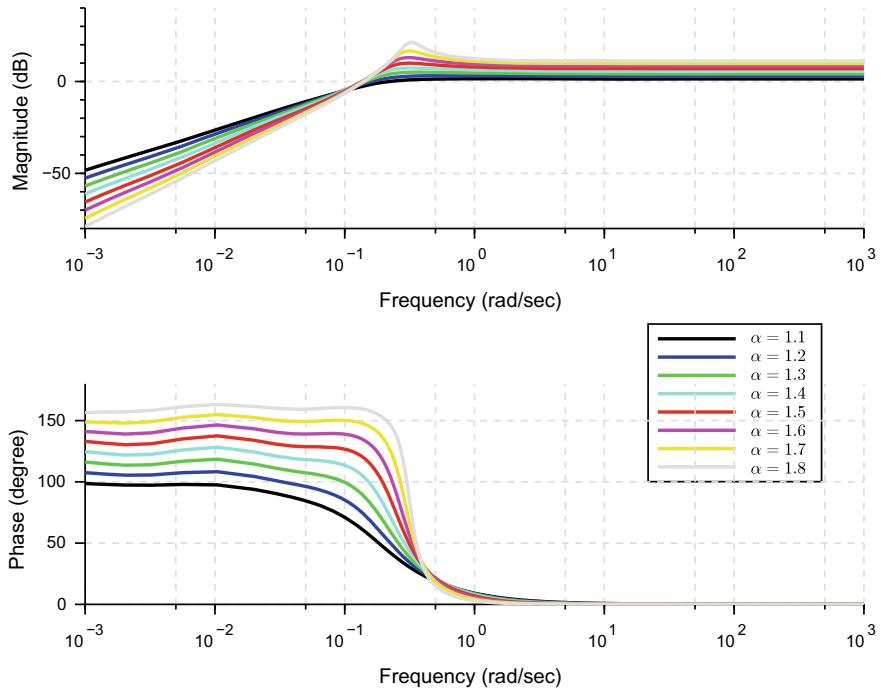


Fig. 4.34 Bode response of fractional-order high-pass filter for $1 < \alpha < 2$

```

subplot(212) plot([t',t',t',t',t',t',t',t'],...
[csim('step',t,H(1))',csim('step',t,H(2))',...
csim('step',t,H(3))',csim('step',t,H(4))',...
csim('step',t,H(5))',csim('step',t,H(6))',...
csim('step',t,H(7))',csim('step',t,H(8))'])
legend('$\alpha=1.1$', '$\alpha=1.2$', '$\alpha=1.3$', ...
'$\alpha=1.4$', '$\alpha=1.5$', '$\alpha=1.6$', ...
'$\alpha=1.7$', '$\alpha=1.8$', "in_lower_left");
title("Step Response");
xlabel("Time (sec)"); ylabel("Amplitude");

```

From the Bode responses of both Fractional-order Low-pass and High-pass filters given in Sects. 4.9.3.1 and 4.9.3.2 respectively, it can be observed that for $\alpha > 1$, the magnitude plot shows a resonant peak, which increases as the order approaches to 2. On the other hand, the variation in phase response is linear for $0 < \alpha < 1$. As the value α of increases from 1, the response exhibits saturation behavior towards the asymptotic values. Furthermore, from the impulse and step responses, it can be seen that for $1 < \alpha < 2$, the responses exhibits oscillations and the magnitude of oscillations increases as the value of α approaches to 2.

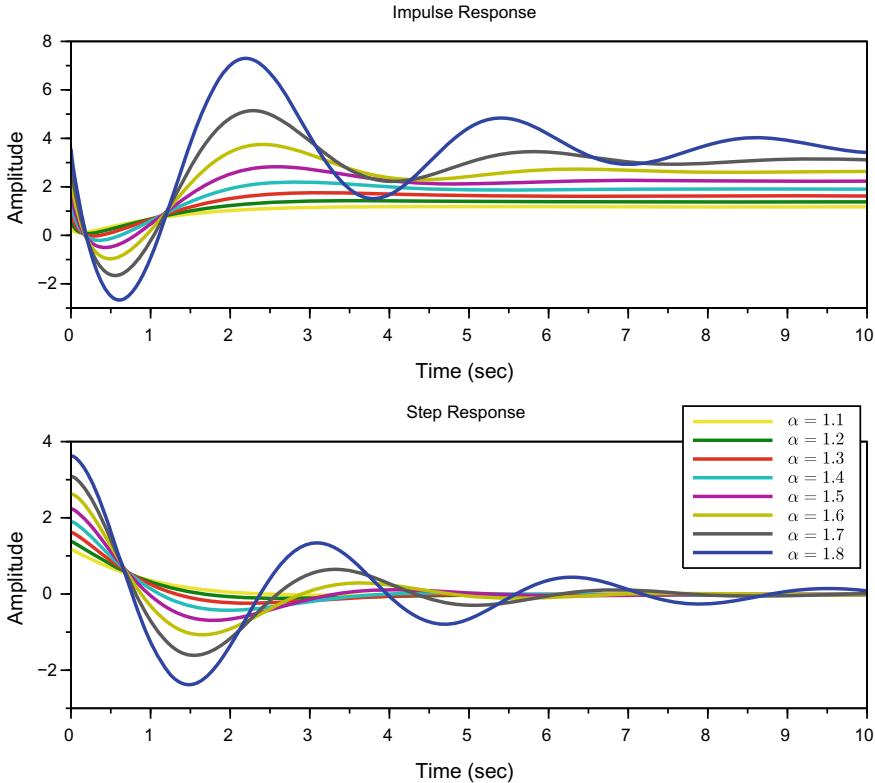


Fig. 4.35 Impulse and step responses of foHPF for $1 < \alpha < 2$

4.9.3.3 Analysis of Fractional-order Band-Pass and All-Pass Filter

On the other hand, the Bode response of fractional-order band-pass and all-pass filters for $0 < \alpha < 1$ shown in Figs. 4.36 and 4.37 respectively are obtained using the following commands. From the responses, it can be seen that the characteristics of the fractional-order band and all-pass filters can be achieved effectively using the developed Scilab commands.

```
--> // Fractional-order Band-pass Filter
exec('foBPF.sci'); N=5; wl=10^-6; wh=10^6; i=1;
for bita=0.1:0.1:0.8
    H(i)=foBPF(0.9,bita,omegaC,'o',N,[wl wh]); i=i+1;
end
bode([H(1);H(2);H(3);H(4);H(5);H(6);H(7);H(8)],wl,wh)
legend('$\beta=0.1$', '$\beta=0.2$', '$\beta=0.3$', ...
'$\beta=0.4$', '$\beta=0.5$', '$\beta=0.6$', ...
'$\beta=0.7$', '$\beta=0.8$', "in_lower_left");
```

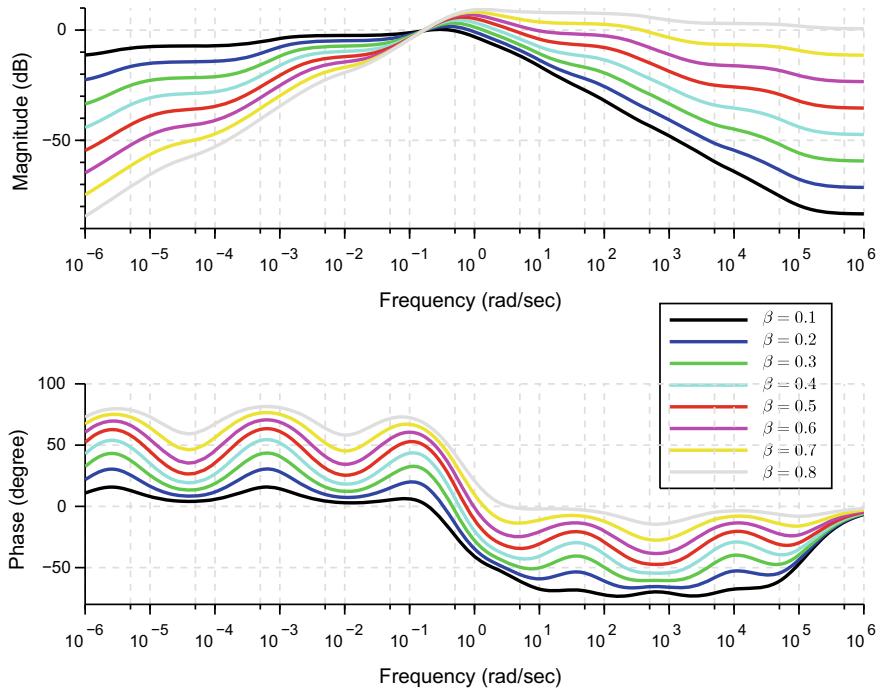


Fig. 4.36 Bode response of fractional-order band-pass filter

```
--> // Fractional-order All-pass Filter
exec('foAPF.sci'); N=5; wl=10^-6; wh=10^6; i=1;
for alpha=0.1:0.1:0.9
    H(i)=foAPF(alpha,omegac,'o',N,[wl wh]); i=i+1;
end
bode([H(1);H(2);H(3);H(4);H(5);H(6);H(7);H(8);H(9)],...
wl,wh);
legend('$\alpha=0.1$', '$\alpha=0.2$', '$\alpha=0.3$', ...
'$\alpha=0.4$', '$\alpha=0.5$', '$\alpha=0.6$', ...
'$\alpha=0.7$', '$\alpha=0.8$', '$\alpha=0.9$', ...
"in_lower_left");
```

4.9.3.4 Performance Analysis on Process Plant

To further analyze the performance of the designed filters, a second order stable process plant given in Eq. (4.96) is considered.

$$G(s) = \frac{1}{(s + 1)^2} \quad (4.96)$$

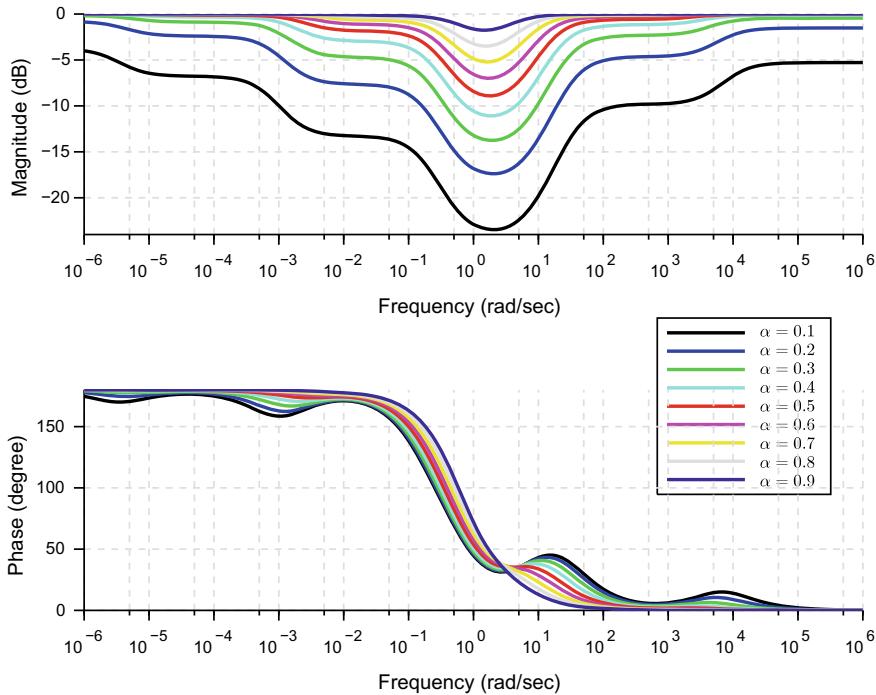


Fig. 4.37 Bode response of fractional-order all-pass filter

For this process, the designed integer-order and fractional-order low-pass filters with parameters $\omega_c = 0.5$ rad/s and $\alpha = 0.85$ are:

$$F(s) = \frac{20}{s + 20} \quad (4.97)$$

$$F(s) = \frac{(20)^{0.85}}{s^{0.85} + (20)^{0.85}} \quad (4.98)$$

The approximated transfer function of the filter using Oustaloup is given as follows:

$$F(s) \approx \frac{[0.5547s^5 + 481.31s^4 + 24783.02s^3 + 80196.30s^2 + 16309.05s + 196.84]}{[355.36s^5 + 298336.95s^4 + 169336.95s^3 + 124867s^2 + 17176.62s + 197.84]} \quad (4.99)$$

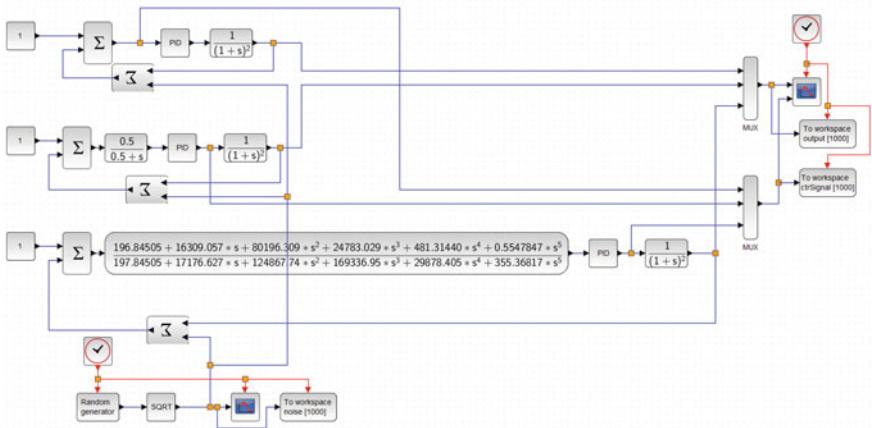


Fig. 4.38 Xcos diagram of the process with and without filters

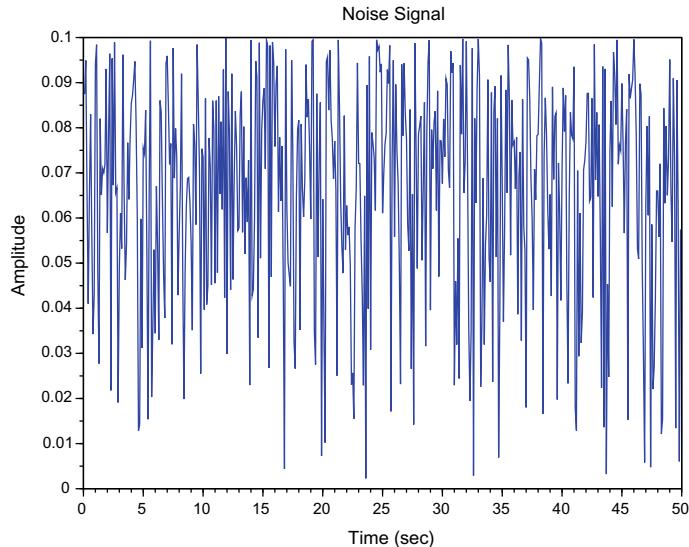


Fig. 4.39 Generated noise signal used for the simulation

Furthermore, the designed PID controller for the control of this process is given as follows:

$$C(s) = 2.03 + \frac{1.42}{s} + 0.65s \quad (4.100)$$

Therefore, the Xcos; a graphical editor in Scilab for the design of dynamical systems diagram with these filters is shown in Fig. 4.38. The noise signal used for the simulation is shown in Fig. 4.39. Therefore, the step response of the system

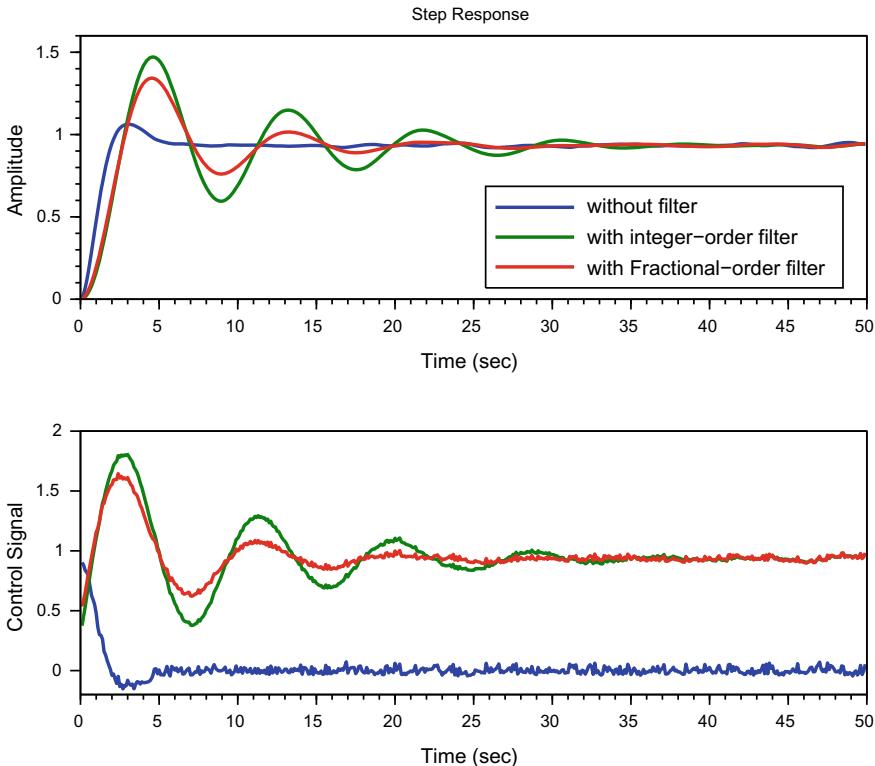


Fig. 4.40 Step responses of stable process with low-pass filters

with these filters is shown in Fig. 4.40. From the responses, it can be seen that the fractional-order filters can be effectively implemented and analyzed using Scilab-Xcos.

From the analysis, it can be seen that the developed fractional-order filters can be effectively used for real-time processes. Furthermore, the characteristics of the filter are also analyzed for various values of α in terms of Bode, impulse and step responses. The simulation study shows that the developed fractional-order filters can be effectively implemented and analyzed using Scilab.

4.10 Summary

In this work, a Scilab based toolbox for fractional-order systems and PID controllers is developed. The proposed toolbox is the first open-source toolbox in open-source software. The toolbox includes the definition of fractional-order parameters, approximation techniques, fractional-order differentiator/integrator, fractional-order based

systems, and PI/PID controllers. Furthermore, the toolbox allows for time and frequency domain as well as stability analysis of the fractional-order systems. The simulation study on various fractional-order based systems and controllers shows that the implementation of fractional-order systems can be done easily on the developed Scilab based toolbox.

References

1. Machado, J.T., Kiryakova, V., Mainardi, F.: Recent history of fractional calculus. *Commun. Nonlinear Sci. Numer. Simul.* **16**(3), 1140–1153 (2011)
2. Matušu, R.: Application of fractional order calculus to control theory. *Int. J. Math. Models Methods Appl. Sci.* **5**(7), 1162–1169 (2011)
3. Caponetto, R.: *Fractional Order Systems: Modeling and Control Applications*. World Scientific (2010)
4. Petráš, I.: *Fractional-Order Nonlinear Systems: Modeling, Analysis and Simulation*. Springer Science & Business Media (2011)
5. Shah, P., Agashe, S.: Review of fractional PID controller. *Mechatronics* **38**, 29–41 (2016)
6. Xue, D., Chen, Y., Atherton, D.P.: *Linear Feedback Control: Analysis and Design with MATLAB*. Siam (2007)
7. Monje, C.A., Chen, Y., Vinagre, B.M., Xue, D., Feliu-Batlle, V.: *Fractional-Order Systems and Controls: Fundamentals and Applications*. Springer Science & Business Media (2010)
8. Tavazoei, M.S.: Time response analysis of fractional-order control systems: a survey on recent results. *Fract. Calc. Appl. Anal.* **17**(2), 440–461 (2014)
9. Bingi, K., Ibrahim, R., Karsiti, M.N., Hassan, S.M., Harindran, V.R.: Real-time control of pressure plant using 2DOF fractional-order PID controller. *Arab. J. Sci. Eng.* **44**(3), 2091–2102 (2019)
10. Bingi, K., Ibrahim, R., Karsiti, M.N., Hassan, S.M.: Fractional order set-point weighted PID controller for pH neutralization process using accelerated PSO algorithm. *Arab. J. Sci. Eng.* **43**(6), 2687–2701 (2018)
11. Oustaloup, A., Melchior, P., Lanusse, P., Cois, O., Dancla, F.: The CRONE toolbox for Matlab. In: *CACSD. Conference Proceedings. IEEE International Symposium on Computer-Aided Control System Design*, Anchorage, USA, 25–27 Sept 2000
12. Malti, R., Melchior, P., Lanusse, P., Oustaloup, A.: Object-oriented CRONE toolbox for fractional differential signal processing. *Signal Image Video Process.* **6**(3), 393–400 (2012)
13. Malti, R., Victor, S.: Crone toolbox for system identification using fractional differentiation models. *IFAC-PapersOnLine* **48**(28), 769–774 (2015)
14. Malti, R., Melchior, P., Lanusse, P., Oustaloup, A.: Towards an object oriented CRONE toolbox for fractional differential systems. *IFAC Proc. Vol.* **44**(1), 10830–10835 (2011)
15. Valerio, D., Da Costa, J.S.: Ninteger: a non-integer control toolbox for MatLab. *Proceedings of Fractional Differentiation and Its Applications*, Bordeaux (2004)
16. de Oliveira Valério, D.P.M.: Ninteger v. 2.3 Fractional Control Toolbox for MATLAB. Lisboa, Universidade Technical (2005)
17. Tepljakov, A., Petlenkov, E., Belikov, J.: FOMCON: fractional-order modeling and control toolbox for MATLAB. In: *Proceedings of the 18th International Conference Mixed Design of Integrated Circuits and Systems-MIXDES 2011*, Gliwice, Poland, 16–18 June 2011
18. Tepljakov, A.: *Fractional-Order Modeling and Control of Dynamic Systems*. Springer (2017)
19. Tepljakov, A., Petlenkov, E., Belikov, J., Finajev, J.: Fractional-order controller design and digital implementation using FOMCON toolbox for MATLAB. In: *2013 IEEE Conference on Computer Aided Control System Design (CACSD)*, Hyderabad, India, 28–30 Aug 2013

20. Lachhab, N., Svaricek, F., Wobbe, F., Rabba, H.: Fractional order PID controller (FOPID)-toolbox. In: 2013 European Control Conference (ECC), Zurich, Switzerland, 17–19 July 2013
21. Chen, Y., Petras, I., Xue, D.: Fractional order control-a tutorial. In: 2009 American Control Conference, St. Louis, MO, USA, 10–12 June 2009
22. Dzieliński, A., Sierociuk, D.: Simulation and experimental tools for fractional order control education. IFAC Proc. Vol. **41**(2), 11654–11659 (2008)
23. Sierociuk, D.: Fractional order discrete state-space system simulink toolkit user guide. <http://www.ee.pw.edu.pl/~dsieroci/fsst/fsst.htm> (2005)
24. Pisoni, E., Visioli, A., Dormido, S.: An interactive tool for fractional order PID controllers. In: 2009 35th Annual Conference of IEEE Industrial Electronics, Porto, Portugal, 3–5 Nov 2009
25. Dormido, S., Pisoni, E., Visioli, A.: An interactive tool for loop-shaping design of fractional-order PID controllers. In: Proceedings of the 4th IFAC Workshop on Fractional Differentiation and Its Applications, Badajoz, Spain, 18–20 Oct 2010
26. Dormido, S., Pisoni, E., Visioli, A.: Interactive tools for designing fractional-order PID controllers. Int. J. Innov. Comp. Inf. Control **8**(7), 4579–4590 (2012)
27. Petráš, I.: Fractional derivatives, fractional integrals, and fractional differential equations in Matlab. In: Assi , A. (ed.) Engineering Education and Research Using MATLAB, pp. 239–264, IntechOpen (2011)
28. Marinov, T.M., Ramirez, N., Santamaria, F.: Fractional integration toolbox. Fract. Calc. Appl. Anal. **16**(3), 670–681 (2013)
29. Lanusse, P., Malti, R., Melchior, P.: Crone control system design toolbox for the control engineering community: tutorial and case study. Phil. Trans. R. Soc. A **371**(1990), 20120149 (2013)
30. Yousfi, N., Melchior, P., Rekik, C., Derbel, N., Oustaloup, A.: Design of centralized CRONE controller combined with MIMO-QFT approach applied to non-square multivariable systems. Int. J. Comput. Appl. **45**(16) (2012)
31. Tepljakov, A.: Fractional-Order Calculus Based Identification and Control of Linear Dynamic Systems. Tallinn University of Technology (2011)
32. Sohal, J.S.: Improvement of artificial neural network based character recognition system, using SciLab. Optik **127**(22), 10510–10518 (2016)
33. Campbell, S.L., Chancelier, J.P., Nikoukhah, R.: Modeling and Simulation in SCILAB. Springer, New York (2006)
34. Bunks, C., Chancelier, J.P., Delebecque, F., Goursat, M., Nikoukhah, R., Steer, S.: Engineering and Scientific Computing with Scilab. Springer Science & Business Media (2012)
35. Wouwer, A.V., Saucez, P., Vilas, C.: Simulation of ode/pde Models With Matlab, Octave and Scilab. Springer, Cham (2014)
36. Rohit, M.T., Ashish, M.K.: Digital Image Processing Using SCILAB. Springer, Cham (2018)
37. Ma, L., Xia, F., Peng, Z.: Integrated design and implementation of embedded control systems with scilab. Sensors **8**(9), 5501–5515 (2008)
38. Pendharkar, I.: Rltool for Scilab: a public domain tool for SISO system design. IEEE Control Syst. Mag. **25**(1), 23–25 (2005)
39. Magyar, Z., Žáková, K.: Scilab based remote control of experiments. IFAC Proc. Vol. **45**(11), 206–211 (2012)
40. Landau, I.D., Zito, G.: Digital Control Systems: Design, Identification and Implementation. Springer Science & Business Media (2007)
41. Bingi, K., Ibrahim, R., Karsiti, M.N., Hassan, S.M.: Fractional-order filter design for set-point weighted PID controlled unstable systems. Int. J. Mech. Mechatron. Eng. **17**(5), 173–179 (2017)
42. Bingi, K., Ibrahim, R., Karsiti, M.N., Hassan, S.M., Harindran, V.R.: Fractional order PI controllers for real-time control of pressure plant. In: 2018 5th International Conference on Control, Decision and Information Technologies, Thessaloniki, Greece, 10–13 Apr 2018
43. Luo, Y., Chen, Y.: Fractional order [proportional derivative] controller for a class of fractional order systems. Automatica **45**(10), 2446–2450 (2009)
44. Tavakoli-Kakhki, M., Haeri, M.: Fractional order model reduction approach based on retention of the dominant dynamics: application in IMC based tuning of FOPI and FOPID controllers. ISA Trans. **50**(3), 432–442 (2011)

45. Feliu-Batlle, V., Perez, R.R., Rodriguez, L.S.: Fractional robust control of main irrigation canals with variable dynamic parameters. *Control Eng. Pract.* **15**(6), 673–686 (2007)
46. Feliu-Batlle, V., Rivas-Perez, R., Castillo-Garcia, F.J.: Fractional order controller robust to time delay variations for water distribution in an irrigation main canal pool. *Comput. Electron. Agric.* **69**(2), 185–197 (2009)
47. El-Khazali, R.: Fractional-order $\text{PI}^\lambda \text{D}^\mu$ controller design. *Comput. Math. Appl.* **66**(5), 639–646 (2013)
48. Latha, K., Rajiniikanth, V., Surekha, P.M.: PSO-based PID controller design for a class of stable and unstable systems. *ISRN Artif. Intell.* **2013** (2013)

Chapter 5

Scilab Based Toolbox for Fractional-order Chaotic Systems



5.1 Introduction

Over the years the chaotic behavior of fractional-order nonlinear systems has been utilized in many real-world applications such as engineering, finance, microbiology, biology, physics, robotics, mathematics, economics, philosophy, meteorology, computer science and civil engineering [1, 2]. From the investigation of researchers, it was found that fractional-order chaotic systems possess less memory and display more sophisticated dynamics compared to its integer-order systems. Recently the attention of researcher's shifts towards creating a chaotic system and its fractional-order form with a more complicated topological structure becomes a desirable task and many of the researchers have made a great contribution. For example, Duffing introduced a duffing equation which can be extended to the complex domain to study strange attractors and chaotic behavior of forced vibrations of industrial machinery [3–5]. Similarly, Van der Pol introduced a model to study oscillations in vacuum tube circuits. The Van der Pol oscillator represents a nonlinear system with an interesting behavior that exhibits naturally in several applications, such as heartbeat, neurons, and acoustic models [6–8].

On the other hand, Lorenz proposed the simplified equations of convection rolls arising in the equations of the atmosphere. From the numerical solution of the equations, it can be observed that the chaotic behavior is known as the butterfly effect which is very sensitive to initial conditions [9–11]. Similarly, Chen proposed a simple autonomous system which is similar but nonequivalent to the Lorenz's attractor. Furthermore, it should be noted that the attractors of Lorenz and Chen systems are double scroll attractor [12]. An example of a single manifold chaotic system was proposed by Rössler. The attractor has only one manifold which is useful in modeling the equilibrium of chemical reactions [13, 14]. The dynamical behavior of these chaotic systems together with many other chaotic systems has been analyzed in Matlab by Petrás in [15–18]. The authors developed a Matlab toolbox named Fractional Order Chaotic Systems which is very useful in analyzing the chaotic behavior of various

fractional-order nonlinear systems. Therefore, in this work, an attempt is made to develop and analyze the dynamic behavior of fractional-order chaotic systems using the open-source software called Scilab. This is because Scilab is extremely effective and available as a free open-source software package for scientific computations. It includes hundreds of general and specialized functions for numerical computations, 2D and 3D visualization and data analysis [19, 20].

5.2 Fractional-order Chaotic Oscillators

This section presents the design and analysis of fractional-order Van der Pol and duffing oscillator for various commensurate and non-commensurate orders using Scilab.

5.2.1 Fractional-order Van der Pol Oscillator

The model of Van der Pol oscillator describing the oscillations of triode in electrical circuit is defined as follows:

$$\ddot{x}(t) + \epsilon(x(t)^2 - 1)\dot{x}(t) + x(t) = 0 \quad (5.1)$$

where $\epsilon > 0$ is the control parameter and represents a relaxation oscillator. On the other hand, if $\epsilon = 0$, the oscillator represents a simple linear oscillator.

The state-space representation of oscillator in Eq. (5.1) is defined as follows:

$$\begin{aligned} \dot{x}_1(t) &= x_2(t) \\ \dot{x}_2(t) &= -x_1(t) - \epsilon(x_1(t)^2 - 1)x_2(t) \end{aligned} \quad (5.2)$$

The fractional-order form of the Van der Pol oscillator can be obtained by fractional-ordering the classical state-space equation in Eq. (5.2) with parameters α_1 and α_2 as follows:

$$\begin{aligned} \mathcal{D}_t^{\alpha_1} x_1(t) &= x_2(t) \\ \mathcal{D}_t^{\alpha_2} x_2(t) &= -x_1(t) - \epsilon(x_1(t)^2 - 1)x_2(t) \end{aligned} \quad (5.3)$$

where α_1 and α_2 are the fractional-order parameters in $(0, 2)$.

The solution for the fractional-order form of the Van der Pol oscillator can be obtained using the Grünwald-Letnikov's definition in Eqs. (4.2) and (4.3) as follows:

$$\begin{aligned}x_1(t_k) &= x_2(t_{k-1})h^{\alpha_1} - \sum_{j=1}^k c_j^{(\alpha_1)} x_1(t_{k-j}) \\x_2(t_k) &= (-x_1(t_k) - \epsilon(x_1^2(t_k) - 1)x_2(t_{k-1}))h^{\alpha_2} - \sum_{j=1}^k c_j^{(\alpha_2)} x_2(t_{k-j})\end{aligned}\quad (5.4)$$

Therefore, the function representing the fractional-order Van der Pol oscillator using Scilab with syntax `[x1, x2]=fracVdp(alpha, epsi, t, h, X)` can be written as follows:

```
function [x1,x2]=fracVdp(alpha,epsi,t,h,X)
n=round(t/h); x1(1)=X(1); x2(1)=X(2);
alpha1=alpha(1); alpha2=alpha(2);
c01=1;c02=1;
for j=1:n
    c1(j)=(1-((1+alpha1)/j))*c01;
    c2(j)=(1-((1+alpha2)/j))*c02;
    c01=c1(j); c02=c2(j);
end
for j=2:n
    x1(j)=x2(j-1)*h^alpha1 - sumGL(x1,c1,j);
    x2(j)=(-x1(j)-epsi*((x1(j)^2)-1)...
        *x2(j-1))*h^alpha2 - sumGL(x2,c2,j);
end
endfunction

function [f]=sumGL(r,c,k)
temp=0;
for j=1:k-1
    temp=temp+c(j)*r(k-j);
end
f=temp;
endfunction
```

In the above function, `alpha` is the fractional-order parameters, `epsi` is the control parameter, `t` is the simulation time, `h` is the step size and `X=(X1(0), X2(0))` is the initial condition. Furthermore, the coefficients `c1` and `c2` are calculated according to the relations given in Eq. (4.3).

Thus, the phase portraits of fractional-order Van der Pol oscillator will be obtained and analyzed for various values of fractional-order parameters and control parameters using the developed Scilab function `[x1, x2]=fracVdp(alpha, epsi, t, h, X)`. Furthermore, in all the cases, the parameters of simulation time and initial conditions used are $t = 50$, $h = 0.005$, $X1(0) = 1$ and $X2(0) = 0$. Therefore, the phase portrait of fractional-order Van der Pol oscillator for various commen-

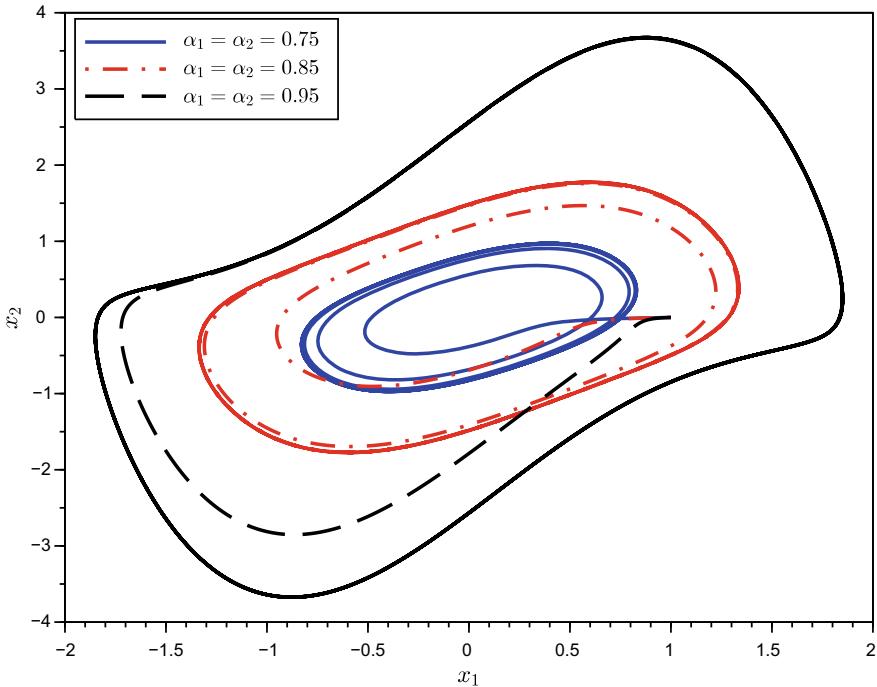


Fig. 5.1 Phase portrait of fractional-order Van der Pol oscillator for various values of fractional-order parameters

surate orders of fractional-order parameters $\alpha_1 = \alpha_2 = 0.75$, $\alpha_1 = \alpha_2 = 0.85$, and $\alpha_1 = \alpha_2 = 0.95$ is obtained as follows:

```
--> scf();
[x11,x21]=fracVanderpol([0.75,0.75],1,50,0.005,[1,0]);
[x12,x22]=fracVanderpol([0.85,0.85],1,50,0.005,[1,0]);
[x13,x23]=fracVanderpol([0.95,0.95],2,50,0.005,[1,0]);
plot(x11,x21,'b',x12,x22,'-r',x13,x23,'--k');
xlabel('$x_1$'); ylabel('$x_2$');
legend(['$\alpha_1=\alpha_2=0.75$',...
'$\alpha_1=\alpha_2=0.85$',...
'$\alpha_1=\alpha_2=0.95$'], "in_upper_left")
```

Thus, the resultant phase portrait plot is shown in Fig. 5.1. Furthermore, the time response of states x_1 and x_2 for the above condition is obtained using the following code and the resultant plot is shown in Fig. 5.2.

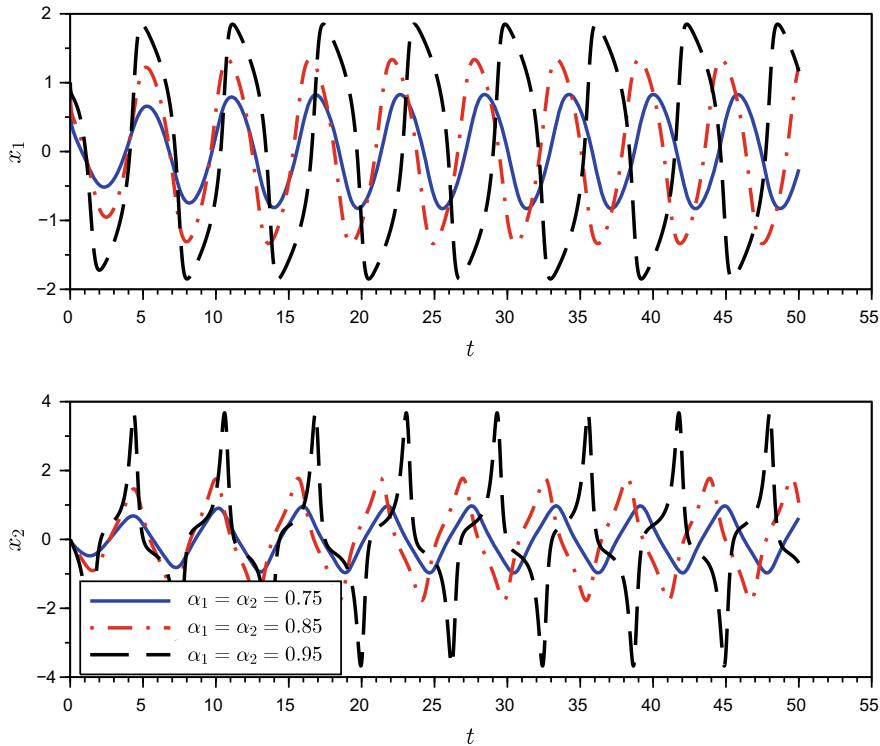


Fig. 5.2 Time response of fractional-order Van der Pol oscillator for various values of fractional-order parameters

```
--> scf(); t=0.005:0.005:50;
    subplot(211)
    plot(t,x11','b',t,x12','-.r',t,x13','--k');
    xlabel('$t$'); ylabel('$x_1$');
    subplot(212)
    plot(t,x21','b',t,x22','-.r',t,x23','--k');
    xlabel('$t$'); ylabel('$x_2$');
    legend(['$\alpha_1=\alpha_2=0.75$',...
        '$\alpha_1=\alpha_2=0.85$',...
        '$\alpha_1=\alpha_2=0.95$'], "in_lower_left")
```

From the figures, it can be observed that the oscillator generates a stable limit cycle when α_1, α_2 increases from 0 to 1. On the other hand, it can also be observed that the oscillator approaches towards the equilibrium point $(0, 0)$ when α_1, α_2 varies from 1 to 0.

In a similar way, the phase portrait of fractional-order Van der Pol oscillator for $\alpha_1 = \alpha_2 = 0.5$ and various values of control parameters $\epsilon = 1, 2, 3$ and 4 is obtained as follows:

```
--> exec('fracVdp.sci'); scf();
[x11,x21]=fracVdp([0.5,0.5],1,50,0.005,[1,0]);
[x12,x22]=fracVdp([0.5,0.5],2,50,0.005,[1,0]);
[x13,x23]=fracVdp([0.5,0.5],3,50,0.005,[1,0]);
[x14,x24]=fracVdp([0.5,0.5],4,50,0.005,[1,0]);
plot(x11,x21,'b',x12,x22,:r',x13,x23,'-m',...
x14,x24,'--k');
xlabel('$x_1$'); ylabel('$x_2$');
legend(['$\epsilon=1$','$\epsilon=2$',...
'$\epsilon=3$','$\epsilon=4$'],'in_upper_left')
```

On the other hand, the time response of the states x_1 and x_2 for this condition of $\alpha_1 = \alpha_2 = 0.5$ and various values of control parameters $\epsilon = 1, 2, 3$ and 4 is obtained using as follows:

```
--> scf(); t=0.005:0.005:50;
subplot(211)
plot(t,x11','b',t,x12,:r',t,x13,'-m',t,x14,'--k');
xlabel('$t$'); ylabel('$x_1$');
subplot(212)
plot(t,x21','b',t,x22,:r',t,x23,'-m',t,x24,'--k');
xlabel('$t$'); ylabel('$x_2$');
legend(['$\epsilon=1$','$\epsilon=2$',...
'$\epsilon=3$','$\epsilon=4$'],'in_lower_left')
```

Therefore, the resultant phase portrait and the time response plots of the states from the above commands are given in Figs. 5.3 and 5.4 respectively. From the figures, a similar observation of stable limit cycle has been observed when ϵ increases from 0 to 1. Also, in this case, the oscillator approaches towards the equilibrium point (0, 0) when ϵ varies from 1 to 0.

Similarly, to analyze the phase portraits of the oscillator for commensurate and non-commensurate orders, the oscillator has been simulated for commensurate and non-commensurate fractional-order parameters $\alpha_1 = \alpha_2 = 0.9$ and $\alpha_1 = 1.2, \alpha_2 = 0.8$ using the following commands:

```
--> [x11,x21]=fracVdp([0.9,0.9],1,50,0.005,[1,0]);
subplot(121); plot(x11,x21);
xlabel('$x_1$'); ylabel('$x_2$'); title('A');
legend(['$\alpha_1=\alpha_2=0.9$'])
[x12,x22]=fracVdp([1.2,0.8],1,50,0.005,[1,0]);
subplot(122); plot(x12,x22);
xlabel('$x_1$'); ylabel('$x_2$'); title('B');
legend(['$\alpha_1=1.2, \alpha_2=0.8$'])
```

Thus, the resultant phase portrait plots of commensurate and non-commensurate orders are shown in Figs. 5.5 and 5.6 respectively. From the figures, it can be observed that for orders less than 1, a stable limit cycle has been observed whereas, for orders

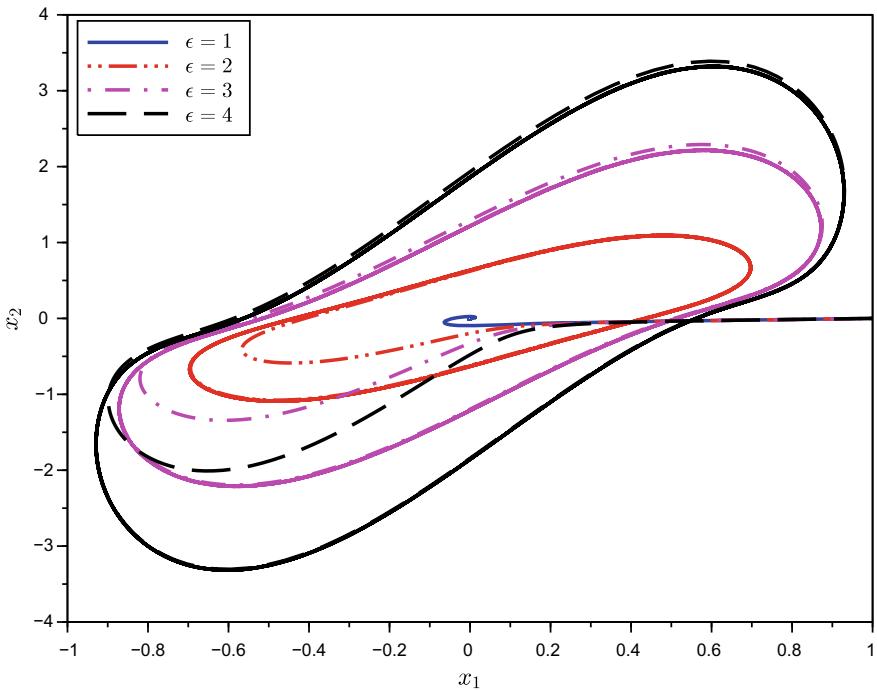


Fig. 5.3 Phase portrait of fractional-order Van der Pol oscillator for various values of control parameter

greater than 1, some strange nature has been found in the outer areas which may be described as bad bands.

5.2.2 Fractional-order Duffing Oscillator

The duffing oscillator is a model of a periodically forced steel beam which is deflected toward the two magnets as shown in Fig. 5.7. From the figure, it can be seen that the model is an example of a periodically forced oscillator with a nonlinear elasticity which can be written as

$$m \frac{d^2x}{dt^2} + \delta \frac{dx}{dt} + (\beta x(t) + \alpha x^3(t)) = \gamma \cos(\omega t) \quad (5.5)$$

where

- m is the mass of the ball,
- $\delta \geq 0$ is the damping constant,
- $\gamma \geq 0$ is the amplitude of external periodic force,

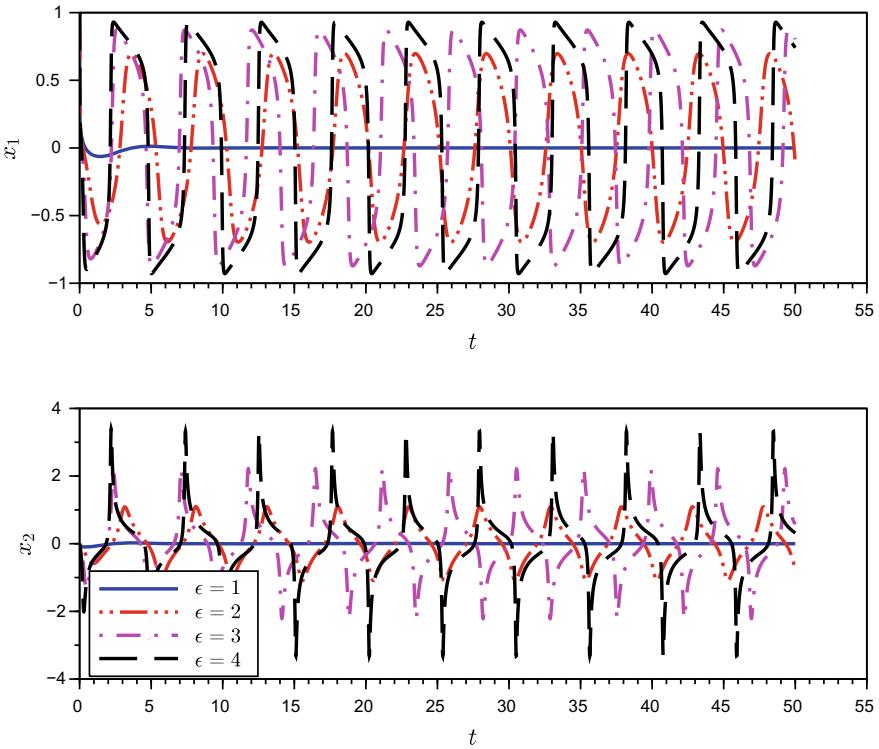


Fig. 5.4 Time response of fractional-order Van der Pol oscillator for various values of control parameter

- ω is the frequency of the periodic force,
- $F = -(\beta x(t) + \alpha x^3(t))$ is the restoring force of the rod.

From the restoring force (F), it should be noted that if $\alpha > 0$, then the spring is hardening spring. On the other hand, for $\alpha < 0$ then the spring is softening spring. Furthermore, for $\beta < 0$, then the Duffing oscillator can be regarded as a model of periodically forced steel rod which is deflected towards the two magnets.

From Eq. (5.5), the state-space formulation of duffing oscillator is given as follows:

$$\begin{aligned}\dot{x}_1(t) &= x_2(t) \\ \dot{x}_2(t) &= -\delta x_2(t) - \beta x_1(t) - \alpha x_1^3(t) + \gamma \cos(\omega t)\end{aligned}\tag{5.6}$$

In a similar way to fractional-order Van der Pol oscillator, the fractional-order form of the duffing oscillator is obtained by fractional-ordering the classical states with parameters α_1 and α_2 as follows:

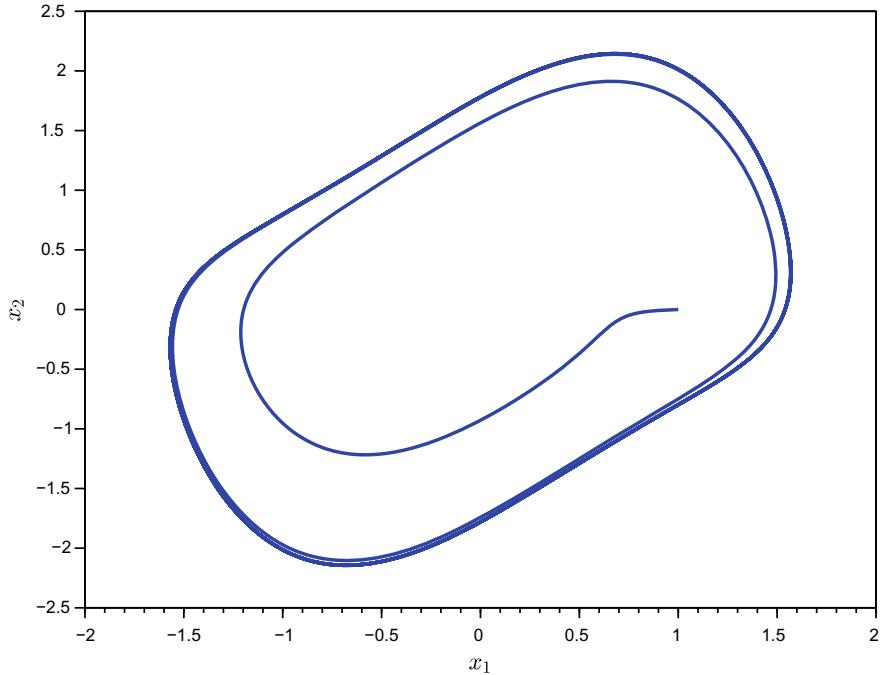


Fig. 5.5 Phase portrait of fractional-order Van der Pol oscillator for $\alpha_1 = \alpha_2 = 0.9$

$$\begin{aligned}\mathcal{D}_t^{\alpha_1} x_1(t) &= x_2(t) \\ \mathcal{D}_t^{\alpha_2} x_2(t) &= -\delta x_2(t) - \beta x_1(t) - \alpha x_1^3(t) + \gamma \cos(\omega t)\end{aligned}\quad (5.7)$$

The solution for the fractional-order duffing oscillator can be obtained using the Grünwald-Letnikov's definition in Eqs. (4.2) and (4.3) as follows:

$$\begin{aligned}x_1(t_k) &= x_2(t_{k-1})h^{\alpha_1} - \sum_{j=1}^k c_j^{(\alpha_1)} x_1(t_{k-j}) \\ x_2(t_k) &= \left(-\delta x_2(t_{k-1}) - \beta x_1(t_k) - \alpha x_1^3(t_k) + \gamma \cos(\omega t) \right) h^{\alpha_2} - \sum_{j=1}^k c_j^{(\alpha_2)} x_2(t_{k-j})\end{aligned}\quad (5.8)$$

Therefore, the function representing the fractional-order duffing oscillator using Scilab with syntax `[x1, x2]=fracDuff(alpha, par, t, h, X)` can be written as follows:

```
function [x1, x2]=fracDuff(alpha, par, t, h, X)
n=round(t/h); x1(1)=X(1); x2(1)=X(2);
d=par(1); b=par(2); a=par(3); g=par(4);
w=par(5); alpha1=alpha(1); alpha2=alpha(2);
```

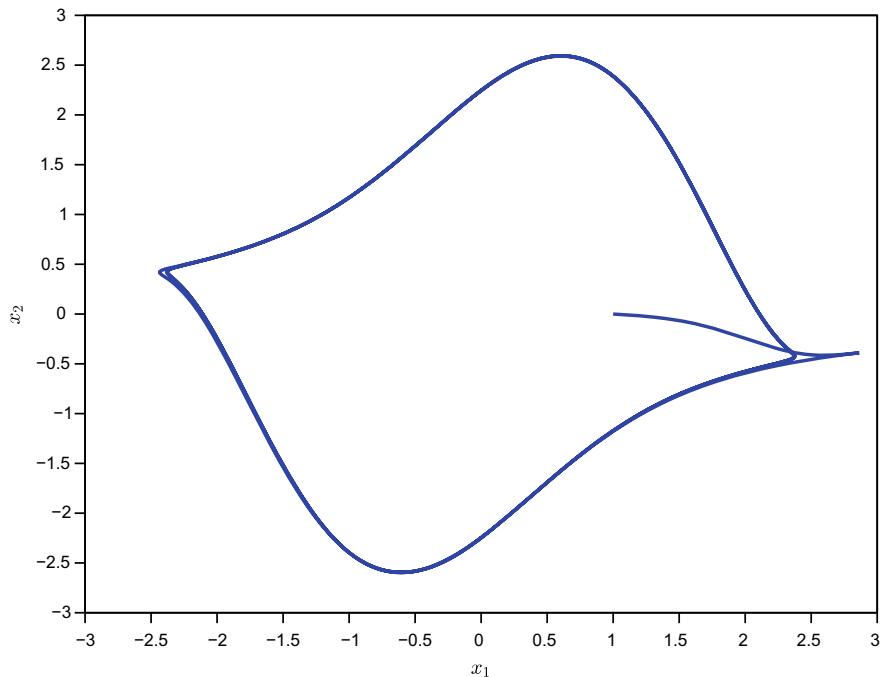


Fig. 5.6 Phase portrait of fractional-order Van der Pol oscillator for $\alpha_1 = 1.2$, $\alpha_2 = 0.8$

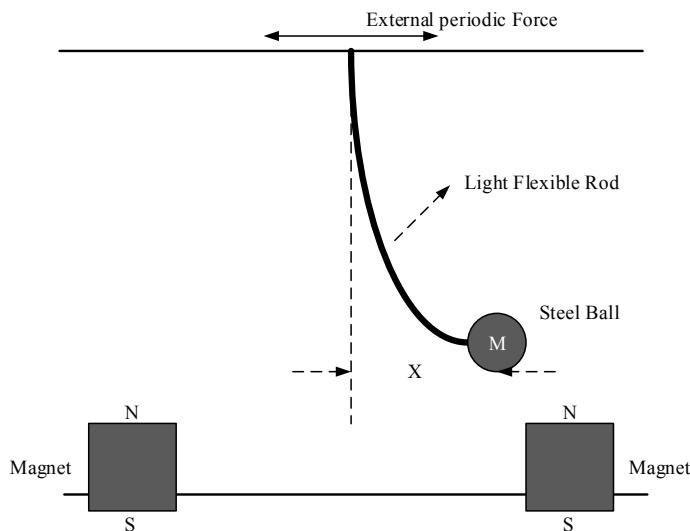


Fig. 5.7 Duffing oscillator model

```

c01=1;c02=1;
for j=1:n
  c1(j)=(1-((1+alpha1)/j))*c01;
  c2(j)=(1-((1+alpha2)/j))*c02;
  c01=c1(j); c02=c2(j);
end
for j=2:n
  x1(j)=x2(j-1)*h^alpha1 - sumGL(x1,c1,j);
  x2(j)=(-(d*x2(j-1))-(b*x1(j))...
    -(a*x1(j)^3)+(g*cos(w*h*j)))*h^alpha2...
    -sumGL(x2,c2,j);
end
endfunction

function [f]=sumGL(r,c,k)
temp=0;
for j=1:k-1
  temp=temp+c(j)*r(k-j);
end
f=temp;
endfunction

```

In the above function, α is the fractional-order parameters, par denotes the system parameters (i.e., δ , β , α , γ and ω), t is the simulation time, h is the step size and $X = (X_1(0), X_2(0))$ is the initial condition. Furthermore, the coefficients c_1 and c_2 are calculated according to the relations given in Eq. (4.3).

Thus, the phase portraits of fractional-order duffing oscillator will be obtained and analyzed for commensurate and non-commensurate orders of fractional-order parameters using the developed Scilab code presented in Sect. 5.2.2. Furthermore, in all the cases, the parameters of oscillator, simulation time and initial conditions used are $\delta = 0.15$, $\beta = -1$, $\alpha = 1$, $\gamma = 0.3$, $\omega = 1$, $t = 200$, $h = 0.05$, $X_1(0) = 0.21$ and $X_2(0) = 0.13$. Therefore, the phase portrait of fractional-order duffing oscillator for commensurate orders of fractional-order parameters $\alpha_1 = \alpha_2 = 0.95$ is obtained as follows:

```
--> exec('fracDuff.sci'); scf();
alpha=[0.95 0.95]; par=[0.15 -1 1 0.3 1];
[x1,x2]=fracDuff(alpha,par,200,0.05,[0.21 0.13]);
plot(x1,x2); xlabel('$x_1$'); ylabel('$x_2$');
```

Similarly, the time response of the states x_1 and x_2 for commensurate orders of fractional-order parameters $\alpha_1 = \alpha_2 = 0.95$ is obtained using the following commands:

```
--> t=0.05:0.05:200; subplot(211); plot(t,x1);
xlabel('$t$'); ylabel('$x_1$'); subplot(212);
plot(t,x2); xlabel('$t$'); ylabel('$x_2$');
```

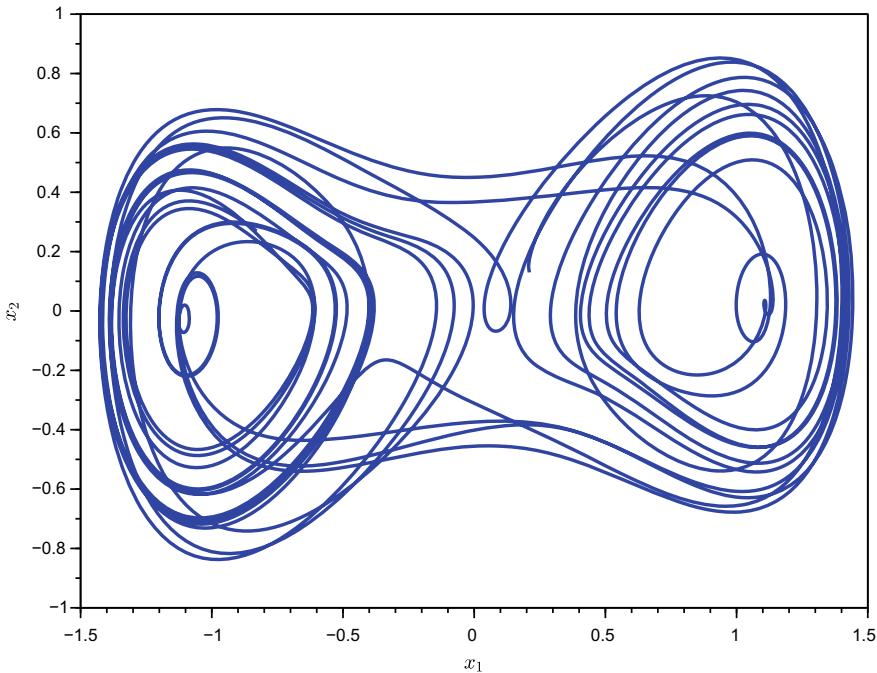


Fig. 5.8 Phase portraits of fractional-order duffing oscillator for commensurate order $\alpha_1 = \alpha_2 = 0.95$

Therefore, the resultant phase portrait and time response of fractional-order duffing oscillator for the commensurate orders is given in Figs. 5.8 and 5.9 respectively. From the figures, it can be seen very clearly that the oscillator produced a stable limit cycle around the three equilibrium points of the system. This can be seen more clearly when the system will be simulated for the integer-orders $\alpha_1 = \alpha_2 = 1$. Therefore, the phase portrait for $\alpha_1 = \alpha_2 = 1$ shown in Fig. 5.10 is obtained using the following commands:

```
--> exec('fracDuff.sci');
scf(); alpha=[1 1]; par=[0.15 -1 1 0.3 1];
[x1,x2]=fracDuff(alpha,par,200,0.05,[0.21 0.13]);
plot(x1,x2); xlabel('$x_1$'); ylabel('$x_2$');
```

Similarly, the phase portrait and time response of fractional-order duffing oscillator for non-commensurate orders of fractional-order parameters $\alpha_1 = 0.9$, $\alpha_2 = 1$ are obtained using the following commands.

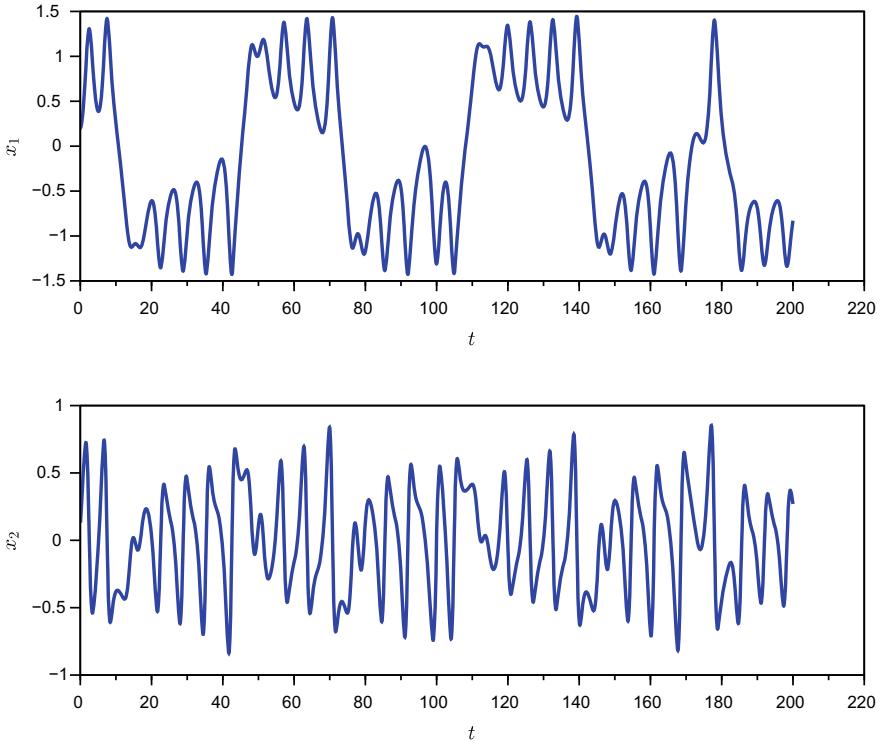


Fig. 5.9 Time response of fractional-order duffing oscillator for commensurate order $\alpha_1 = \alpha_2 = 0.95$

```
--> exec('fracDuff.sci'); scf();
alpha=[0.9 1.0]; pa=[0.15 -1 1 0.3 1];
[x1,x2]=fracDuff(alpha,pa,200,0.05,[0.21 0.13]);
plot(x1,x2); xlabel('$x_1$'); ylabel('$x_2$');
scf(); subplot(211); plot(t,x1);
xlabel('$t$'); ylabel('$x_1$');
subplot(212); plot(t,x2);
xlabel('$t$'); ylabel('$x_2$');
```

Therefore, the resultants phase portrait and time response plots of fractional-order duffing oscillator for non-commensurate orders $\alpha_1 = 0.9, \alpha_2 = 1$ are given in Figs. 5.11 and 5.12 respectively. From the figures, a similar observation of stable limit cycle has been observed. However, the oscillator approaches towards the two equilibrium point only. This is because the system having one unstable equilibrium which satisfies the chaotic behavior of the system. For a dynamical system to be classified as chaotic, it must have the following properties:

- must be period infinity orbit or a periodic orbit,
- must be bounded and
- must be sensitive to initial conditions.

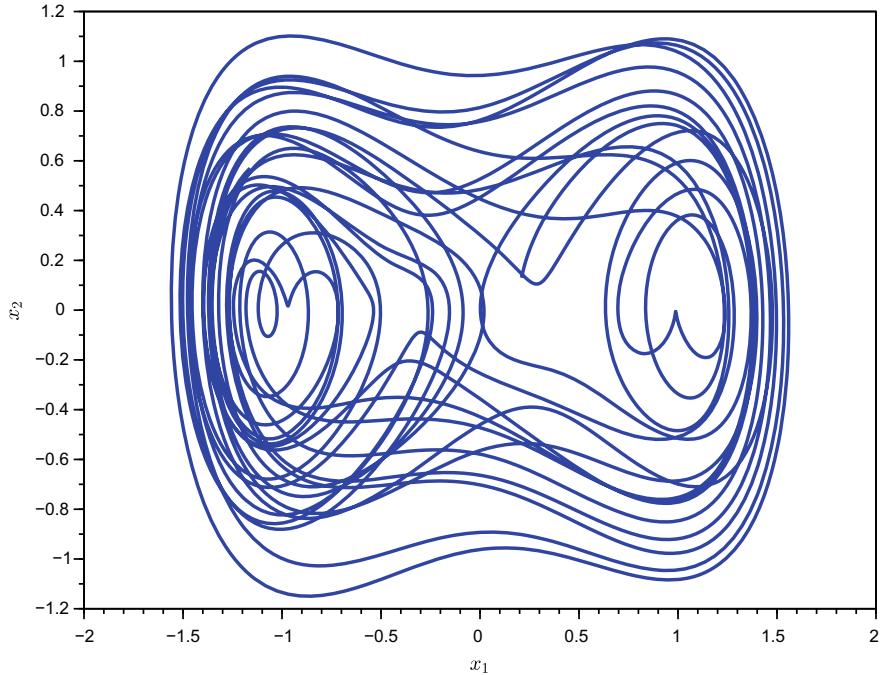


Fig. 5.10 Phase portraits of fractional-order duffing oscillator for order $\alpha_1 = \alpha_2 = 1$

From Figs. 5.11 and 5.12, it can be observed that the phase portrait is periodic and has been bounded. Furthermore, to test the oscillator for sensitive to initial conditions, the oscillator is simulated for two sets of initial conditions with very minimal difference i.e., (0.21, 0.13) and (0.211, 0.13). Therefore, the resultant phase portrait shown in Figs. 5.13 and 5.14 respectively can be obtained as follows:

```
--> exec('fracDuff.sci'); alp=[0.9 1.0]; t=200;
h=0.05; init=[0.211 0.13]; par=[0.15 -1 1 0.3 1];
[x1,x2]=fracDuff(alp,par,t,h,[0.21 0.13]);
[x11,x21]=fracDuff(alp,par,t,h,init);
plot(x1,x2,'b',x11,x21,'r');
xlabel('$x_1$'); ylabel('$x_2$');
legend(['[0.21 0.13]', '[0.211 0.13]'])

scf(); t=0.05:0.05:200;
subplot(211); plot(t,x1,'b',t,x11,'r');
xlabel('$t$'); ylabel('$x_1$');
subplot(212); plot(t,x2,'b',t,x21,'r');
xlabel('$t$'); ylabel('$x_2$');
legend(['[0.21 0.13]', '[0.211 0.13]'])
```

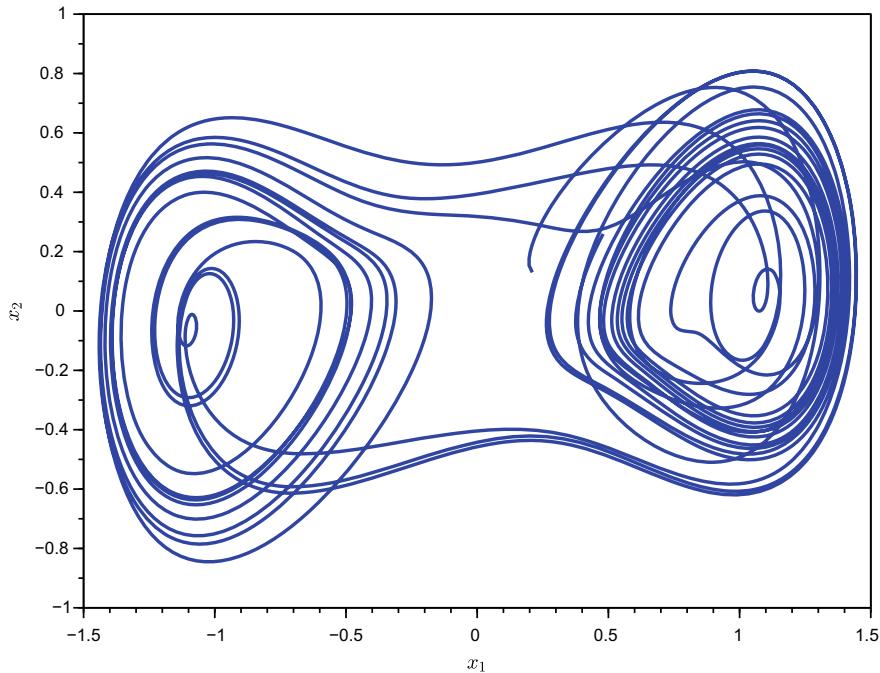


Fig. 5.11 Phase portraits of fractional-order duffing oscillator for non-commensurate order $\alpha_1 = 0.9$, $\alpha_2 = 1.0$

From the figure, it can be observed that the oscillator is very sensitive to initial conditions. Here, a small change in the initial value of the current trajectory has been lead to a significant change in future behavior. Therefore, the system can be classified under a chaotic system.

5.3 Fractional-order Lorenz's Chaotic System

The fractional-order form of the Lorenz's chaotic system is defined as follows:

$$\begin{aligned}\mathcal{D}_t^{\alpha_1} x(t) &= p(y(t) - x(t)) \\ \mathcal{D}_t^{\alpha_2} y(t) &= x(t)(q - z(t)) - y(t) \\ \mathcal{D}_t^{\alpha_3} z(t) &= x(t)y(t) - rz(t)\end{aligned}\tag{5.9}$$

where

- α_1 , α_2 and α_3 are the fractional derivative orders,
- p is the Prandtl number,

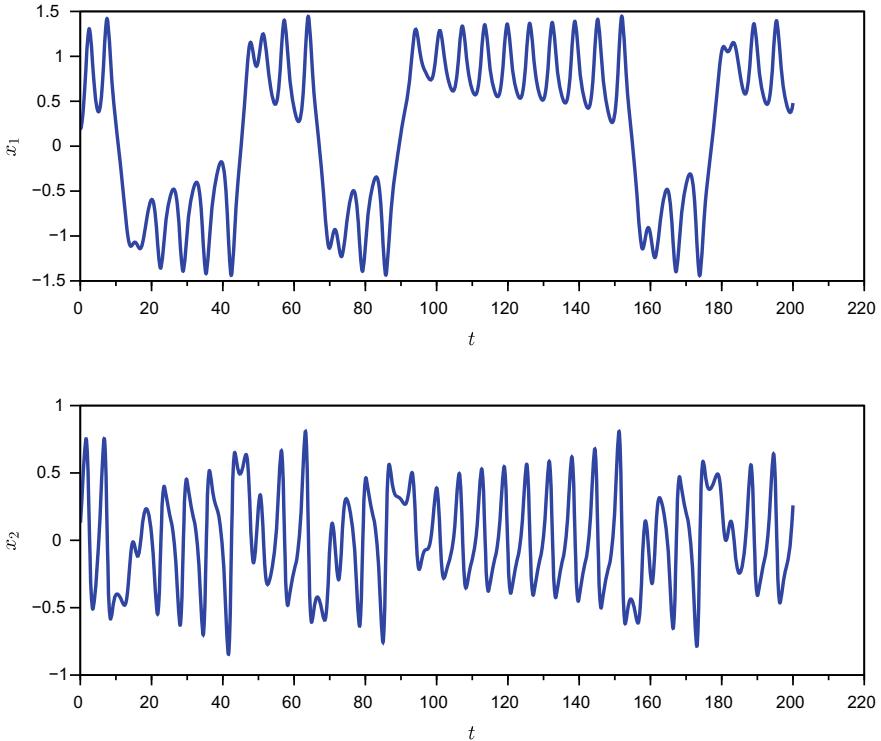


Fig. 5.12 Time response of fractional-order duffing oscillator for non-commensurate order $\alpha_1 = 0.9$, $\alpha_2 = 1.0$

- q is the Rayleigh number and
- r is the physical dimension of the layer.

Therefore, the numerical solution for the fractional-order Lorenz's chaotic system can be obtained using the Grünwald-Letnikov's definition in Eqs. (4.2) and (4.3) as follows:

$$\begin{aligned} x(t_k) &= (p(y(t_{k-1}) - x(t_{k-1})))h^{\alpha_1} - \sum_{j=1}^k c_j^{(\alpha_1)}x(t_{k-j}) \\ y(t_k) &= (x(t_k)(q - z(t_{k-1})) - y(t_{k-1}))h^{\alpha_2} - \sum_{j=1}^k c_j^{(\alpha_2)}y(t_{k-j}) \\ z(t_k) &= (x(t_k)y(t_k) - rz(t_{k-1}))h^{\alpha_3} - \sum_{j=1}^k c_j^{(\alpha_3)}z(t_{k-j}) \end{aligned} \quad (5.10)$$

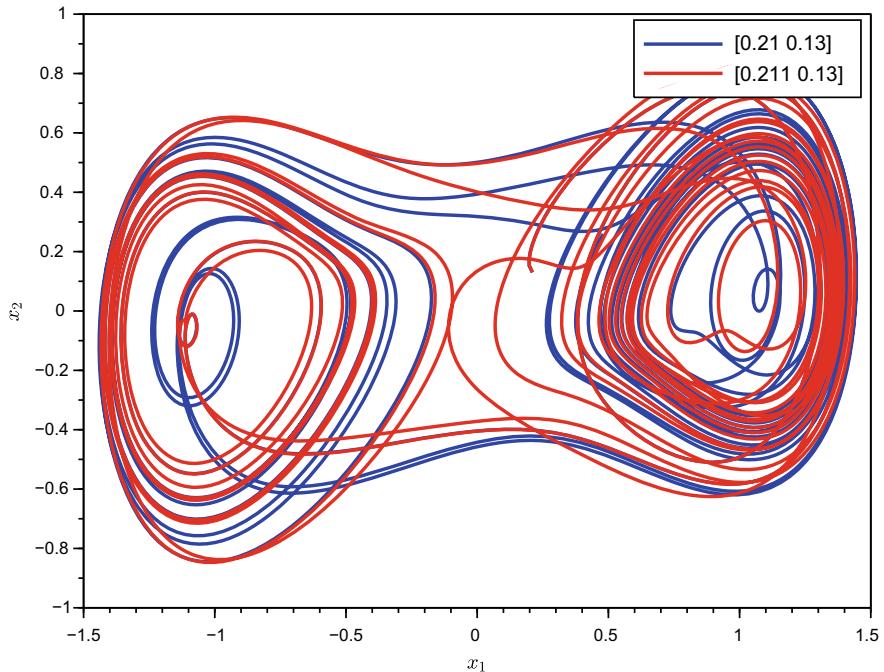


Fig. 5.13 Phase portrait of fractional-order duffing oscillator for sensitivity to initial conditions

Thus, the function representing the fractional-order Lorenz's chaotic system using Scilab with syntax `[x1, x2, x3]=fracLorenz(alpha, param, t, h, X)` can be written as follows:

```

function [x1,x2,x3]=fracLorenz(alpha,param,t,h,X)
n=round(t/h); x1(1)=X(1); x2(1)=X(2); x3(1)=X(3);
alpha1=alpha(1);alpha2=alpha(2); alpha3=alpha(3);
p=param(1);q=param(2);r=param(3);
c01=1; c02=1; c03=1;
for j=1:n
    c1(j)=(1-((1+alpha1)/j))*c01;
    c2(j)=(1-((1+alpha2)/j))*c02;
    c3(j)=(1-((1+alpha3)/j))*c03;
    c01=c1(j); c02=c2(j); c03=c3(j);
end
for j=2:n
    x1(j)=(p*(x2(j-1)-x1(j-1)))*h^alpha1-sumGL(x1,c1,j);
    x2(j)=(x1(j)*(q-x3(j-1))-x2(j-1))*h^alpha2-sumGL(x2,c2,j);
    x3(j)=((x1(j)*x2(j))-(r*x3(j-1)))*h^alpha3-sumGL(x3,c3,j);
end
endfunction

```

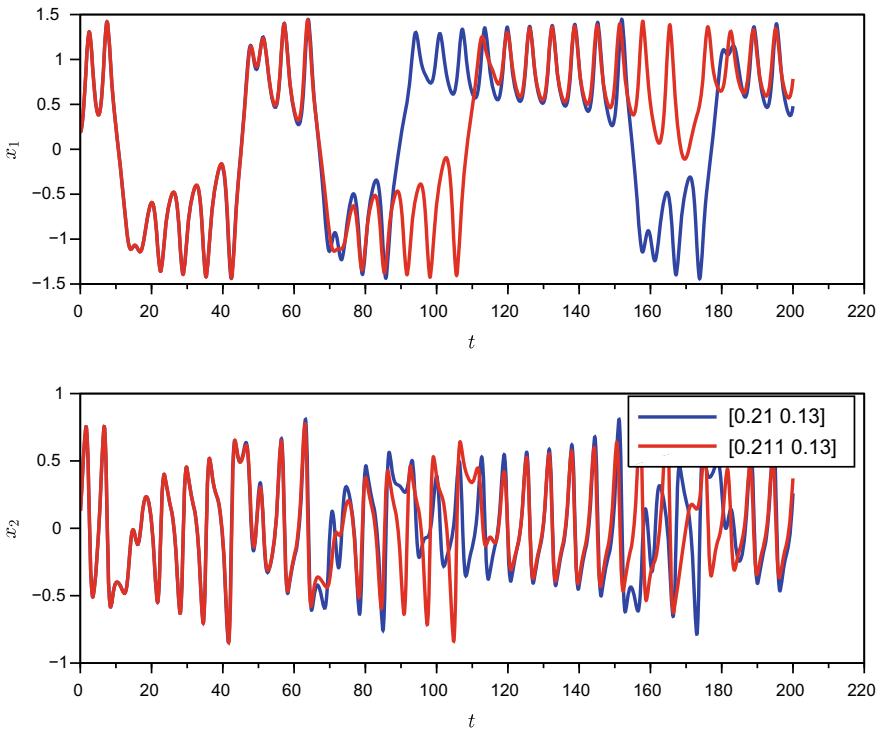


Fig. 5.14 Time response of fractional-order duffing oscillator for sensitivity to initial conditions

```
function [f]=sumGL(r,c,k)
temp=0;
for j=1:k-1
    temp=temp+c(j)*r(k-j);
end
f=temp;
endfunction
```

In the above function, α is the fractional-order parameters (i.e., α_1 , α_2 and α_3), par denotes the oscillators parameters (i.e., p , q and r), t is the simulation time, h is the step size and $X = (X_1(0), X_2(0), X_3(0))$ is the initial condition. Furthermore, the coefficients c_1 , c_2 and c_3 are calculated according to the relations given in Eq. (4.3).

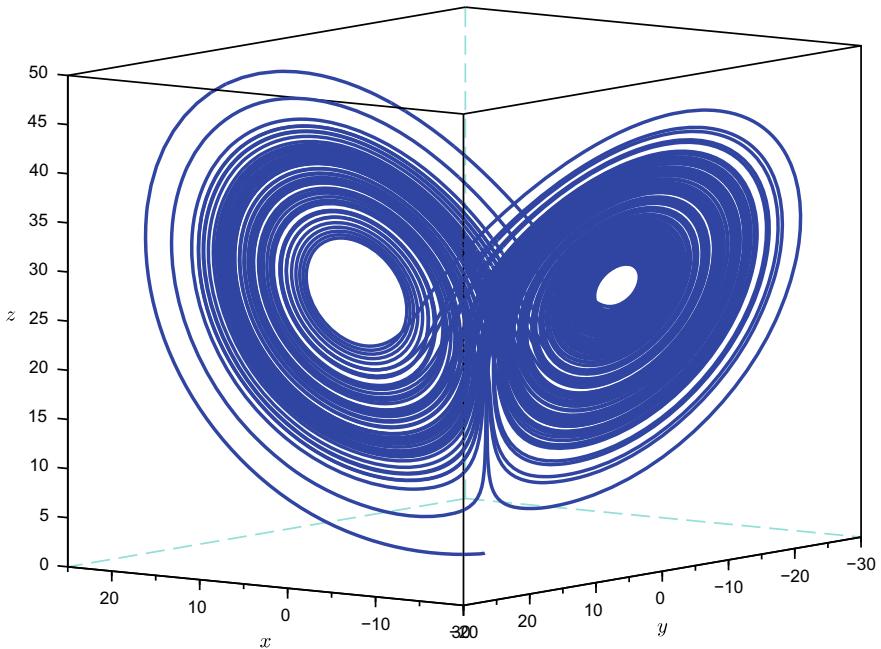


Fig. 5.15 Chaotic attractor of Lorenz's system for commensurate orders of $\alpha_1 = \alpha_2 = \alpha_3 = 0.995$

5.3.1 Commensurate Orders

In this simulation study, the chaotic attractor of Lorenz's system has been obtained for the commensurate orders $\alpha_1 = \alpha_2 = \alpha_3 = 0.995$. Furthermore, the attractor has been generated with system parameters $p = 10$, $q = 28$, $r = 8/3$ for time $t = 100$ and step size $h = 0.005$ with initial conditions as $x_1(0) = x_2(0) = x_3(0) = 0.1$. Thus, the Scilab commands for generating the chaotic attractor of Lorenz's system are as follows:

```
--> exec('fracLorenz.sci');
[x1,x2,x3]=fracLorenz([0.995,0.995,0.995],...
[10,28,8/3],100,0.005,[0.1,0.1,0.1]);
param3d(x1,x2,x3);
xlabel('$x_1$');
ylabel('$x_2$');
zlabel('$x_3$');
```

Thus, the resultant chaotic attractor of Lorenz's system is shown in Fig. 5.15. Furthermore, the projections of the attractor on $x - y$, $y - z$ and $x - z$ planes shown in Fig. 5.16a, b and c respectively can be obtained as follows:

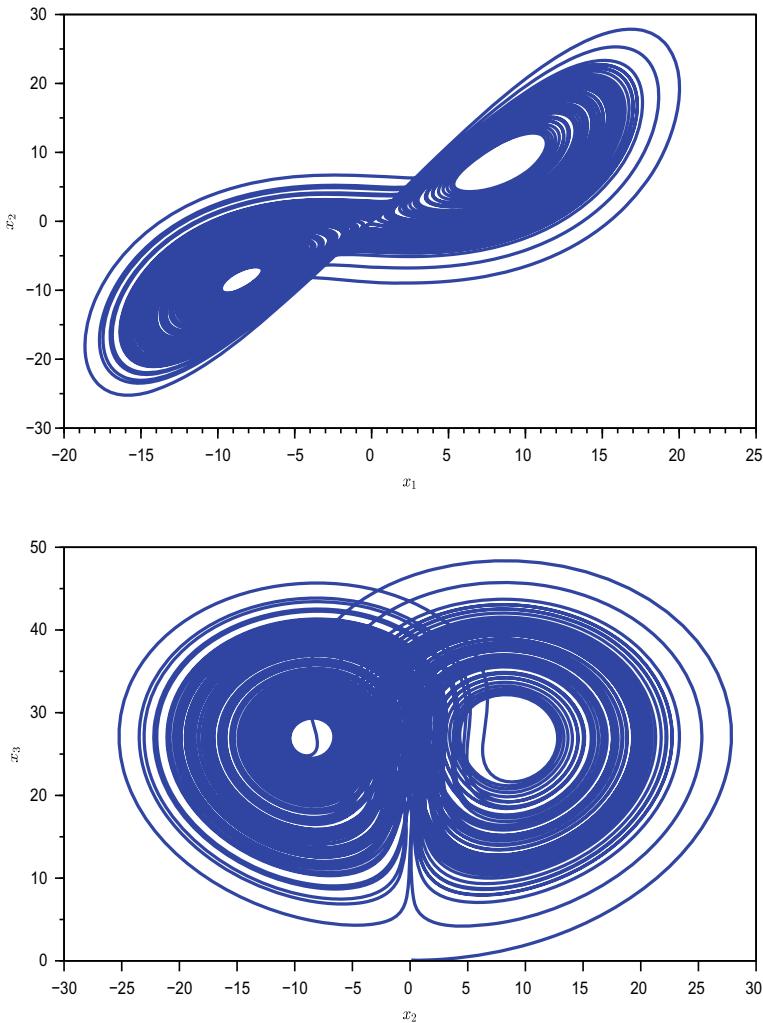
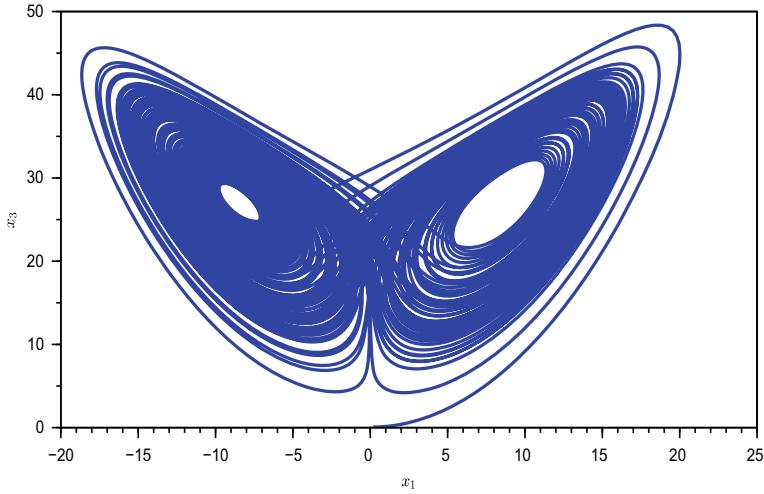


Fig. 5.16 Chaotic attractor of Lorenz's system projected on $x - y$, $y - z$ and $x - z$ planes

```
--> scf();
subplot(311);
plot(x1,x2); xlabel('$\mathbf{(a)}~x$'); ylabel('$\mathbf{y}$');
subplot(312);
plot(x2,x3); xlabel('$\mathbf{(b)}~y$'); ylabel('$\mathbf{z}$');
subplot(313);
plot(x1,x3); xlabel('$\mathbf{(c)}~x$'); ylabel('$\mathbf{z}$');
```

On the other hand, the time response of the system for the states x , y and z shown in Fig. 5.17 can be obtained as follows:

**Fig. 5.16** (continued)

```
--> scf(); t=0.005:0.005:100;
subplot(311);
plot(t,x1); xlabel('$(a)~t$'); ylabel('$x$');
subplot(312);
plot(t,x2); xlabel('$(b)~t$'); ylabel('$y$');
subplot(313);
plot(t,x3); xlabel('$(c)~t$'); ylabel('$z$');
```

5.3.2 Non-commensurate Orders

Similarly, the chaotic attractor of Lorenz's system for the non-commensurate orders of $\alpha_1 = \alpha_2 = 1$ and $\alpha_3 = 0.995$ with parameters $p = 10, q = 28, r = 8/3, t = 100, h = 0.005, x_1(0) = 0.1, x_2(0) = 0.1$ and $x_3(0) = 0.1$ can be obtained as follows:

```
--> scf(); exec('fracLorenz.sci');
[x1,x2,x3]=fracLorenz([1,1,0.99],...
[10,28,8/3],100,0.005,[0.1,0.1,0.1]);
subplot(221); param3d(x1,x2,x3);
xlabel('$x$'); ylabel('$y$'); zlabel('$z$');
subplot(223);
plot(x1,x2); xlabel('$x$'); ylabel('$y$');
subplot(225);
plot(x2,x3); xlabel('$y$'); ylabel('$z$');
subplot(224);
plot(x1,x3); xlabel('$x$'); ylabel('$z$');
```

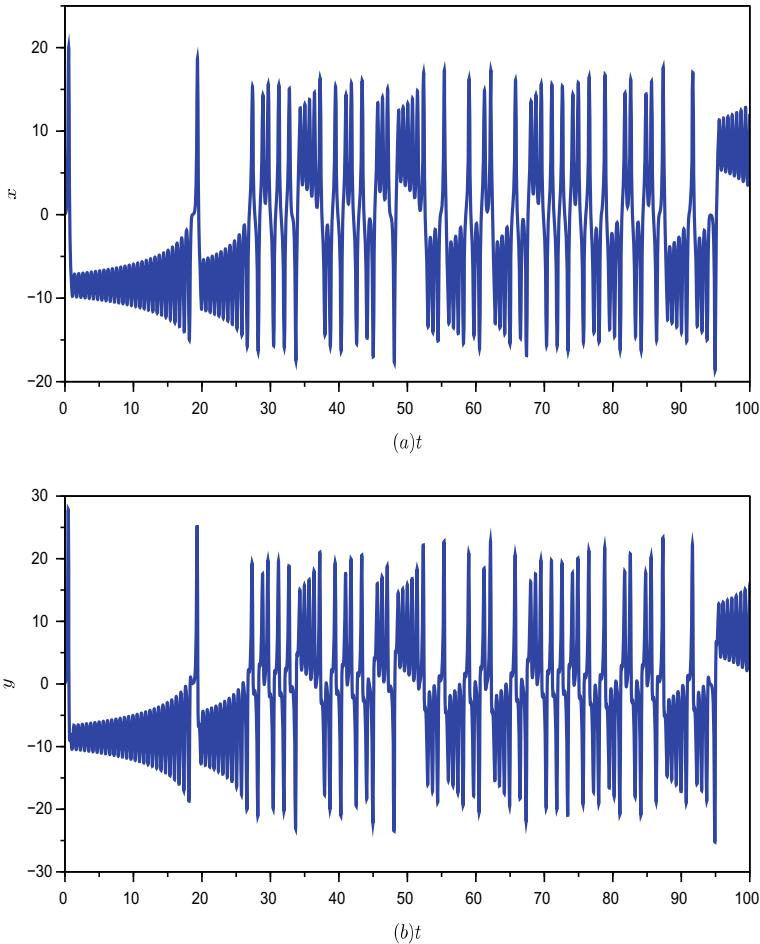


Fig. 5.17 Time responses of Lorenz's system for the states x , y and z

Therefore, the resultant chaotic attractor together with the projections on $x - y$, $y - z$ and $x - z$ planes is shown in Fig. 5.18.

Furthermore, the time response of the system shown in Fig. 5.19 for the states x , y and z can be obtained as follows:

```
--> scf(); t=0.005:0.005:100;
    subplot(311);
    plot(t,x1); xlabel('$ (a) \ t $'); ylabel('$ x $');
    subplot(312);
    plot(t,x2); xlabel('$ (b) \ t $'); ylabel('$ y $');
    subplot(313);
    plot(t,x3); xlabel('$ (c) \ t $'); ylabel('$ z $');
```

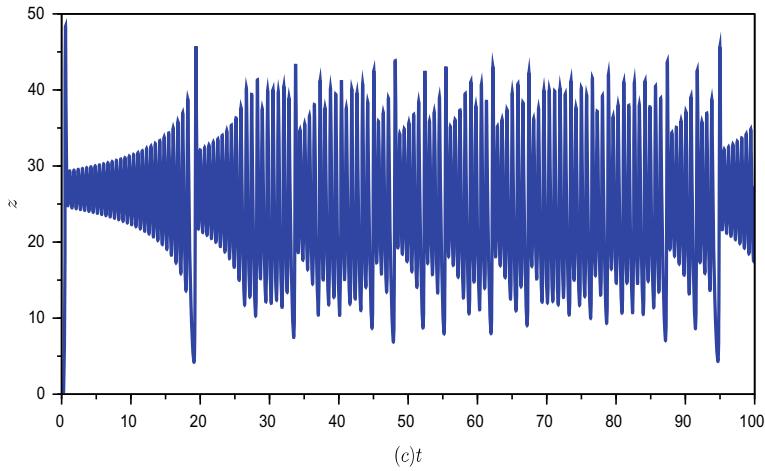


Fig. 5.17 (continued)

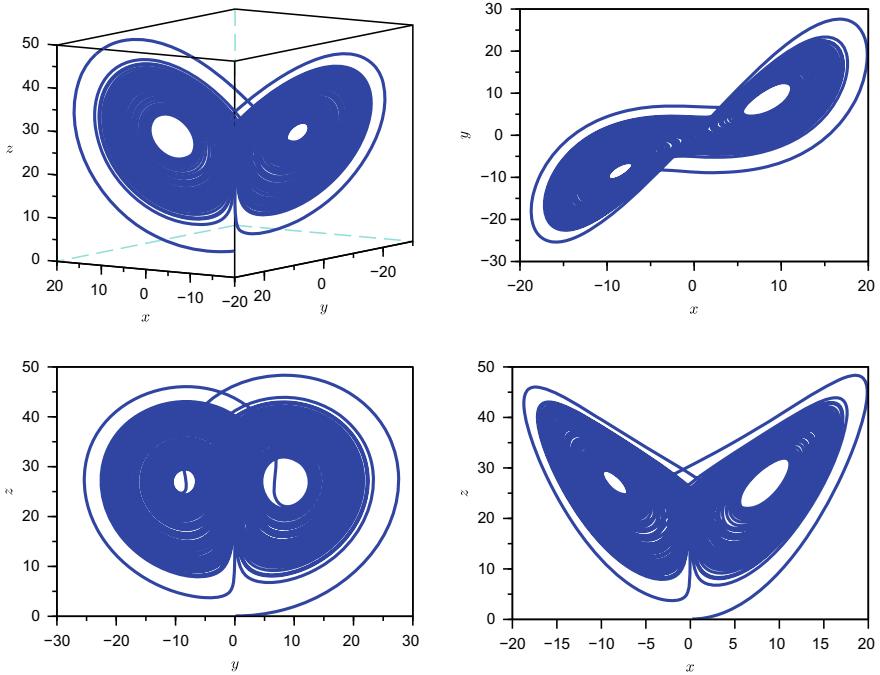


Fig. 5.18 Chaotic attractor of Lorenz's system for non-commensurate orders of $\alpha_1 = \alpha_2 = 1$ and $\alpha_3 = 0.995$

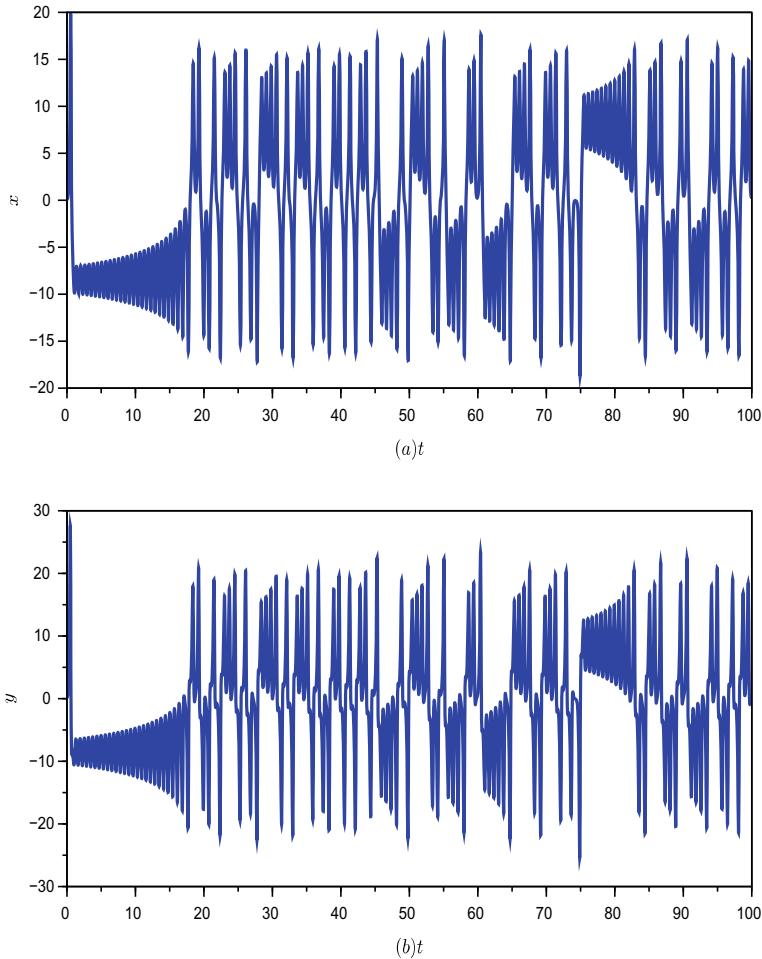


Fig. 5.19 Time responses of Lorenz's system for the states x , y and z

From the chaotic attractors shown in Figs. 5.15, 5.16 and 5.18, it can be observed that the chaotic behavior is known as butterfly effect which is very sensitive to initial conditions.

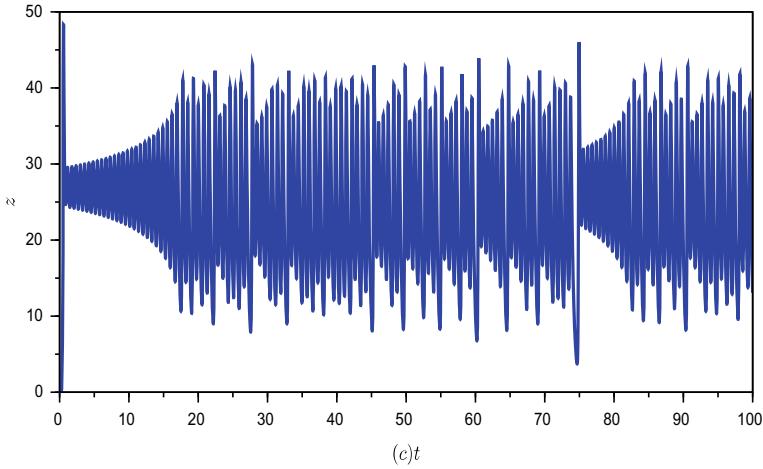


Fig. 5.19 (continued)

5.4 Fractional-order Chen's Chaotic System

The fractional-order form of the chaotic system proposed by the Chen is described as follows:

$$\begin{aligned}\mathcal{D}_t^{\alpha_1}x(t) &= p(y(t) - x(t)) \\ \mathcal{D}_t^{\alpha_2}y(t) &= -(r - p)x(t) - x(t)z(t) + ry(t) \\ \mathcal{D}_t^{\alpha_3}z(t) &= x(t)y(t) - qz(t)\end{aligned}\quad (5.11)$$

where

- α_1, α_2 and α_3 are the fractional derivative orders and
- p, q and r are the system parameters

Therefore, the numerical solution for the fractional-order Chen's chaotic system can be obtained using the Grünwald-Letnikov's definition in Eqs. (4.2) and (4.3) as follows:

$$\begin{aligned}x(t_k) &= p(y(t_k) - x(t_k))h^{\alpha_1} - \sum_{j=1}^k c_j^{(\alpha_1)}x(t_{k-j}) \\ y(t_k) &= (- (r - p)x(t_k) - x(t_k)z(t_{k-1}) + ry(t_{k-1}))h^{\alpha_2} - \sum_{j=1}^k c_j^{(\alpha_2)}y(t_{k-j}) \\ z(t_k) &= (x(t_k)y(t_k) - qz(t_{k-1}))h^{\alpha_3} - \sum_{j=1}^k c_j^{(\alpha_3)}z(t_{k-j})\end{aligned}\quad (5.12)$$

Thus, the function representing the fractional-order Chen's chaotic system using Scilab with syntax `[x1,x2,x3]=fracChen(alpha,param,t,h,X)` can be written as follows:

```

function [x1,x2,x3]=fracChen(alpha,param,t,h,X)
n=round(t/h); x1(1)=X(1); x2(1)=X(2); x3(1)=X(3);
alpha1=alpha(1);alpha2=alpha(2); alpha3=alpha(3);
p=param(1);q=param(2);r=param(3);
c01=1; c02=1; c03=1;
for j=1:n
    c1(j)=(1-((1+alpha1)/j))*c01;
    c2(j)=(1-((1+alpha2)/j))*c02;
    c3(j)=(1-((1+alpha3)/j))*c03;
    c01=c1(j); c02=c2(j); c03=c3(j);
end
for j=2:n
    x1(j)=(p*(x2(j-1)-x1(j-1)))*h^alpha1-sumGL(x1,c1,j);
    x2(j)=(-r*p*x1(j)-x1(j)*x3(j-1)+r*x2(j-1))*h^alpha2...
        -sumGL(x2,c2,j);
    x3(j)=(x1(j)*x2(j)-q*x3(j-1))*h^alpha3-sumGL(x3,c3,j);
end
endfunction

function [f]=sumGL(r,c,k)
temp=0;
for j=1:k-1
    temp=temp+c(j)*r(k-j);
end
f=temp;
endfunction

```

Similar to fractional-order Lorenz's function, here also, α denotes the fractional-order parameters (i.e., α_1 , α_2 and α_3), par denotes the oscillators parameters (i.e., p , q and r), t is the simulation time, h is the step size and $X=(X_1(0), X_2(0), X_3(0))$ is the initial condition. Furthermore, the coefficients c_1 , c_2 and c_3 are calculated according to the relations given in Eq. (4.3).

5.4.1 Commensurate Orders

This simulation study provides the chaotic attractor of fractional-order Chen's system for the commensurate orders of $\alpha_1 = \alpha_2 = \alpha_3 = 0.9$. Here, the parameters for the simulation are used as $p = 32$, $q = 3$ and $r = 28$. Thus, the chaotic attractor for the simulation time $t = 100$ and step size $h = 0.005$ can be obtained as follows:

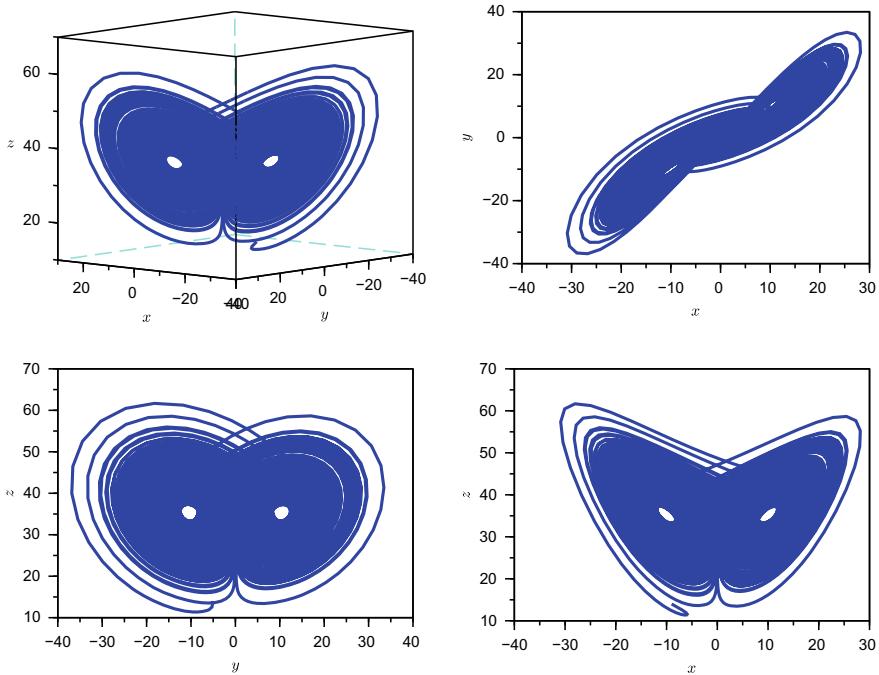


Fig. 5.20 Chaotic attractor of Chen's system for commensurate orders of $\alpha_1 = \alpha_2 = \alpha_3 = 0.9$

```
--> scf(); exec('fracChen.sci');
[x1,x2,x3]=fracChen([0.9,0.9,0.9],[35,3,28],...
100,0.005,[-9,-5,14]);
subplot(221);
param3d(x1,x2,x3); xlabel('$x$');
ylabel('$y$'); zlabel('$z$');
subplot(222);
plot(x1,x2); xlabel('$x$'); ylabel('$y$');
subplot(223);
plot(x2,x3); xlabel('$y$'); ylabel('$z$');
subplot(224);
plot(x1,x3); xlabel('$x$'); ylabel('$z$');
```

Thus, the resultant chaotic attractor of fractional-order Chen's system for the commensurate orders from the above commands is shown in Fig. 5.20. Furthermore, the figure also gives the projections of the attractor on $x - y$, $y - z$, and $x - z$ planes.

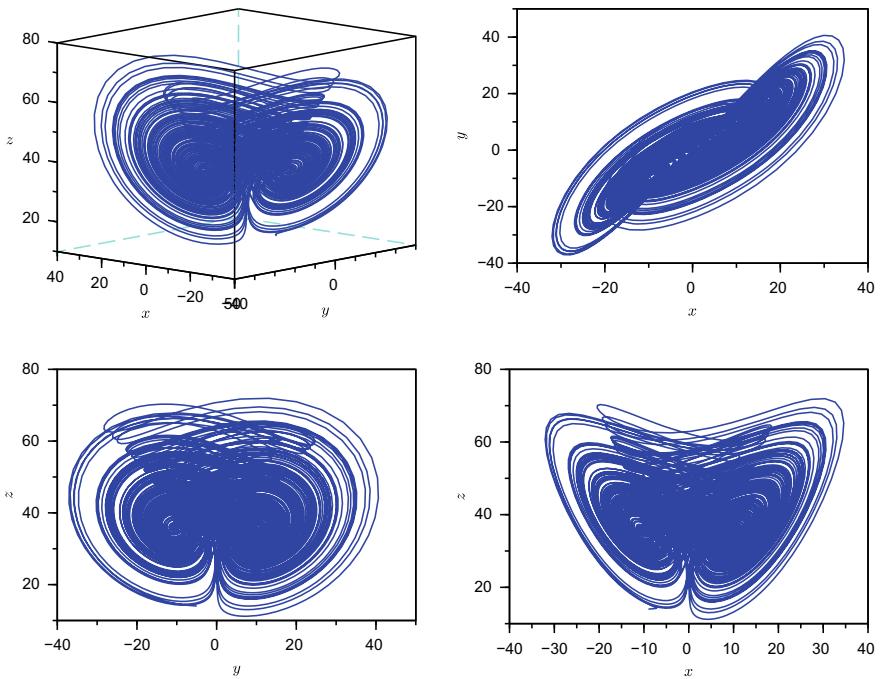


Fig. 5.21 Chaotic attractor of Chen's system for non-commensurate orders of $\alpha_1 = 0.995$, $\alpha_2 = 1.0$ and $\alpha_3 = 1.005$

5.4.2 Non-commensurate Orders

Similarly, the chaotic attractor of fractional-order Chen's system for the non-commensurate orders of $\alpha_1 = 0.995$, $\alpha_2 = 1.0$ and $\alpha_3 = 1.005$ with system parameters $p = 32$, $q = 3$ and $r = 28$ with simulation parameters $t = 100$ and $h = 0.005$ shown in Fig. 5.21 can be obtained as follows:

```
--> scf(); exec('fracChen.sci');
[x1,x2,x3]=fracChen([0.995,1.0,1.005],[35,3,28],...
100,0.005,[-9,-5,14]);
subplot(221);
param3d(x1,x2,x3); xlabel('$x$');
ylabel('$y$'); zlabel('$z$');
subplot(222);
plot(x1,x2); xlabel('$x$'); ylabel('$y$');
subplot(223);
plot(x2,x3); xlabel('$y$'); ylabel('$z$');
subplot(224);
plot(x1,x3); xlabel('$x$'); ylabel('$z$');
```

From the chaotic attractors given in Figs. 5.20 and 5.21, it can be observed that the behavior of Chen's system is similar but nonequivalent to the Lorenz's attractor. Furthermore, it should be noted that the attractors of Lorenz and Chen systems are double scroll attractor. An example of a single manifold chaotic system will be discussed in the next section.

5.5 Fractional-order Rössler's Chaotic System

The fractional-order form of the Rössler's chaotic system which is useful in modeling the chemical equilibrium reactions that has only one manifold is described as follows:

$$\begin{aligned}\mathcal{D}_t^{\alpha_1}x(t) &= -(y(t) - z(t)) \\ \mathcal{D}_t^{\alpha_2}y(t) &= x(t) + py(t) \\ \mathcal{D}_t^{\alpha_3}z(t) &= q + z(t)(x(t) - r)\end{aligned}\quad (5.13)$$

where

- α_1, α_2 and α_3 are the fractional derivative orders and
- a, b and b are the system parameters

Therefore, the numerical solution for the fractional-order Lorenz's chaotic oscillator can be obtained using the Grünwald-Letnikov's definition in Eqs. (4.2) and (4.3) as follows:

$$\begin{aligned}x(t_k) &= -(y(t_{k-1}) - z(t_{k-1}))h^{\alpha_1} - \sum_{j=1}^k c_j^{(\alpha_1)}x(t_{k-j}) \\ y(t_k) &= (x(t_k) + py(t_{k-1}))h^{\alpha_2} - \sum_{j=1}^k c_j^{(\alpha_2)}y(t_{k-j}) \\ z(t_k) &= (q + z(t_{k-1})(x(t_k) - r))h^{\alpha_3} - \sum_{j=1}^k c_j^{(\alpha_3)}z(t_{k-j})\end{aligned}\quad (5.14)$$

Thus, the function representing the fractional-order Rössler's chaotic system using Scilab with syntax `[x1, x2, x3]=fracRossler(alpha, param, t, h, X)` can be written as follows:

```
function [x1,x2,x3]=fracRossler(alpha,param,t,h,X)
n=round(t/h); x1(1)=X(1); x2(1)=X(2); x3(1)=X(3);
alpha1=alpha(1); alpha2=alpha(2); alpha3=alpha(3);
p=param(1); q=param(2); r=param(3);
c01=1; c02=1; c03=1;
for j=1:n
```

```

c1(j)=(1-((1+alpha1)/j))*c01;
c2(j)=(1-((1+alpha2)/j))*c02;
c3(j)=(1-((1+alpha3)/j))*c03;
c01=c1(j); c02=c2(j); c03=c3(j);
end
for j=2:n
x1(j)=(- (x2(j-1)+x3(j-1)))*h^alpha1-sumGL(x1,c1,j);
x2(j)=(x1(j)+(p*x2(j-1)))*h^alpha2-sumGL(x2,c2,j);
x3(j)=(q+x3(j-1)*(x1(j)-r))*h^alpha3-sumGL(x3,c3,j);
end
endfunction

function [f]=sumGL(r,c,k)
temp=0;
for j=1:k-1
temp=temp+c(j)*r(k-j);
end
f=temp;
endfunction

```

5.5.1 Commensurate Orders

This simulation study provided the chaotic attractor of fractional-order Rössler's system for commensurate orders $\alpha_1 = \alpha_2 = \alpha_3 = 0.9$ with system parameters $p = 0.5$, $q = 0.2$ and $r = 10$. Thus, the chaotic attractor of the system shown in Fig. 5.22 for the time $t = 120$ and $h = 0.005$ with initial conditions $x(0) = 0.5$, $y(0) = 1.5$ and $z(0) = 0.1$ can be obtained as follows

```

--> scf(); exec('fracRossler.sci');
[x1,x2,x3]=fracRossler([0.9,0.9,0.9],[0.5,0.2,10],...
120,0.005,[0.5,1.5,0.1]);
param3d(x1,x2,x3);
xlabel('$x(t)$'); ylabel('$y(t)$'); zlabel('$z(t)$');
title('Fractional-order Rossler's system');

```

5.5.2 Non-commensurate Orders

In a similar way, the chaotic attractor of fractional-order Rössler's system for non-commensurate orders $\alpha_1 = 0.9$, $\alpha_2 = 0.85$ and $\alpha_3 = 0.95$ with parameters $p = 0.5$,

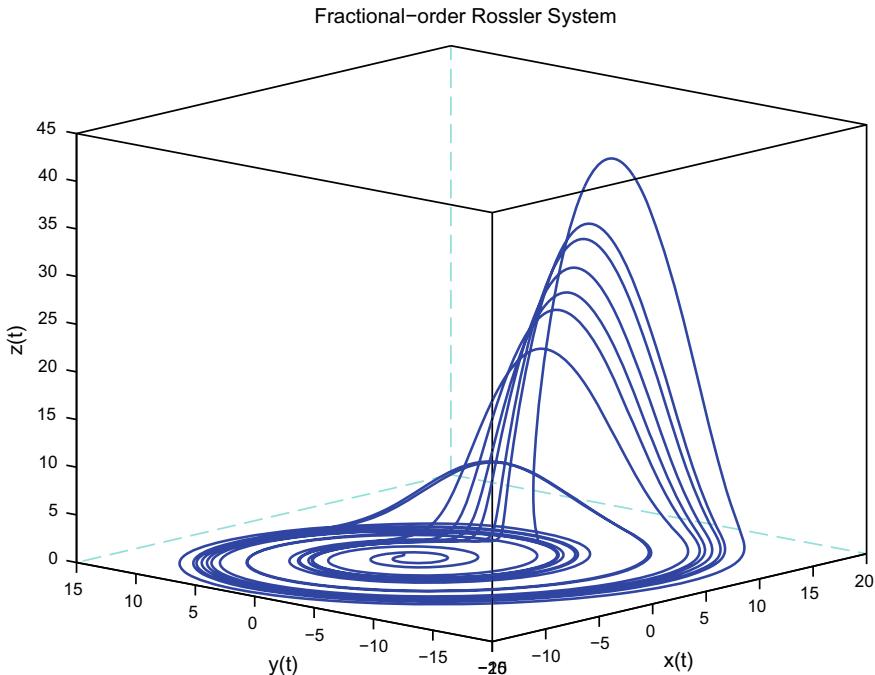


Fig. 5.22 Chaotic attractor of Rössler's system for commensurate orders of $\alpha_1 = \alpha_2 = \alpha_3 = 0.9$

$q = 0.2$, $r = 10$, $t = 120$, $h = 0.005$, $x(0) = 0.5$, $y(0) = 1.5$ and $z(0) = 0.1$ can be obtained as follows:

```
--> scf(); exec('fracRossler.sci');
[x1,x2,x3]=fracRossler([0.9,0.85,0.95],[0.5,0.2,10],...
120,[0.5,1.5,0.1]);
param3d(x1,x2,x3);
xlabel('$x(t)$'); ylabel('$y(t)$'); zlabel('$z(t)$');
title('Fractional-order Rossler's system');
```

Thus, the resultant chaotic attractor of the fractional-order Rössler's system is shown in Fig. 5.23. From the figure, it can be noted that the attractor is similar to the Lorenz's attractor, but is simpler and has only one manifold.

The summary of all the developed functions for the fractional-order chaotic systems in this toolbox including their Scilab syntaxes is given in Table 5.1.

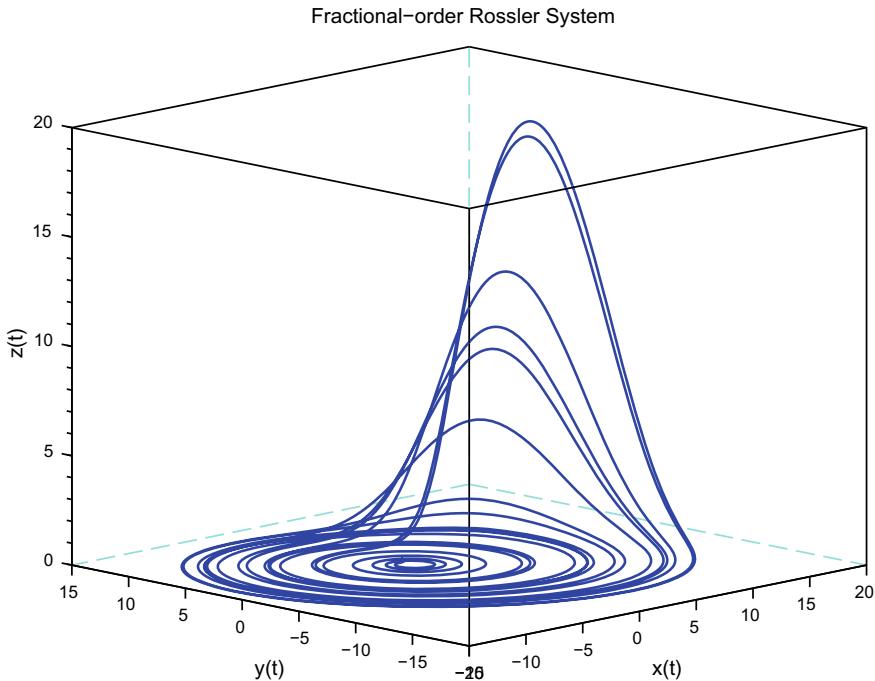


Fig. 5.23 Chaotic attractor of Rössler's system for non-commensurate orders of $\alpha_1 = 0.9$, $\alpha_2 = 0.85$ and $\alpha_3 = 0.95$

Table 5.1 List of functions developed in Scilab based toolbox for fractional-order chaotic systems

Function	Syntax
Fractional-order Van der Pol Oscillator	<code>[x1, x2]=fracVanderpol(alpha, epsi, t, h, X)</code>
Fractional-order Duffing Oscillator	<code>[x1, x2]=fracDuff(alpha, param, t, h, X)</code>
Fractional-order Lorenz's System	<code>[x1, x2, x3]=fracLorenz(alpha, param, t, h, X)</code>
Fractional-order Chen's System	<code>[x1, x2, x3]=fracChen(alpha, param, t, h, X)</code>
Fractional-order Rössler's system	<code>[x1, x2, x3]=fracRossler(alpha, param, t, h, X)</code>

5.6 Summary

In this work, a Scilab based toolbox for fractional-order chaotic systems is developed. The proposed toolbox is the first open-source toolbox in open-source software. The toolbox includes fractional-order Van der Pol and duffing oscillators and fractional-order Lorenz, Chen and Rössler's systems. Using the developed functions, the dynamic behavior of these chaotic systems has been analyzed effectively

for various commensurate and non-commensurate orders using Scilab. However, it should be noted that the execution time of Scilab is slower than Matlab which can be observed while running these complex fractional-order chaotic systems [21, 22].

References

1. Zhang, W., Liao, S.K., Shimizu, N.: Dynamic behaviors of nonlinear fractional-order differential oscillator. *J. Mech. Sci. Technol.* **23**(4), 1058–1064 (2009)
2. Elwakil, A.S.: Fractional-order circuits and systems: an emerging interdisciplinary research area. *IEEE Circuits Syst. Mag.* **10**(4), 40–50 (2010)
3. Ge, Z.M., Ou, C.Y.: Chaos in a fractional order modified Duffing system. *Chaos Solitons Fractals* **34**(2), 262–291 (2007)
4. Shen, Y., Yang, S., Xing, H., Gao, G.: Primary resonance of Duffing oscillator with fractional-order derivative. *Commun. Nonlinear Sci. Numer. Simul.* **17**(7), 3092–3100 (2012)
5. Baleanu, D., Machado, J.A.T., Luo, A.C.: *Fractional Dynamics and Control*. Springer Science & Business Media (2011)
6. Shen, Y., Yang, S., Sui, C.: Analysis on limit cycle of fractional-order van der Pol oscillator. *Chaos Solitons Fractals* **67**, 94–102 (2014)
7. Matouk, A.E.: Chaos, feedback control and synchronization of a fractional-order modified autonomous Van der Pol-Duffing circuit. *Commun. Nonlinear Sci. Numer. Simul.* **16**(2), 975–986 (2011)
8. Tavazoei, M.S., Haeri, M., Attari, M., Bolouki, S., Siami, M.: More details on analysis of fractional-order van der Pol oscillator. *J. Vib. Control* **15**(6), 803–819 (2009)
9. Grigorenko, I., Grigorenko, E.: Chaotic dynamics of the fractional Lorenz system. *Phys. Rev. Lett.* **91**(3), 034101 (2003)
10. Wu, X.J., Shen, S.L.: Chaos in the fractional-order Lorenz system. *Int. J. Comput. Math.* **86**(7), 1274–1282 (2009)
11. Munmuangsaen, B., Srisuchinwong, B.: A hidden chaotic attractor in the classical Lorenz system. *Chaos Solitons Fractals* **107**, 61–66 (2018)
12. Li, C., Peng, G.: Chaos in Chen's system with a fractional order. *Chaos Solitons Fractals* **22**(2), 443–450 (2004)
13. Zhang, W., Zhou, S., Li, H., Zhu, H.: Chaos in a fractional-order Rössler system. *Chaos Solitons Fractals* **42**(3), 1684–1691 (2009)
14. Li, C., Chen, G.: Chaos and hyperchaos in the fractional-order Rössler equations. *Physica A* **341**, 55–61 (2004)
15. Petráš, I.: *Fractional-Order Nonlinear Systems: Modeling, Analysis and Simulation*. Springer Science & Business Media (2011)
16. Petráš, I.: A note on the fractional-order Volta's system. *Commun. Nonlinear Sci. Numer. Simul.* **15**(2), 384–393 (2010)
17. Petráš, I.: A note on the fractional-order Chua's system. *Chaos Solitons Fractals* **38**(1), 140–147 (2008)
18. Petráš, I.: Stability of fractional order systems with rational orders: a survey. *Fract. Calc. Appl. Anal.* **12**(3), 269–298 (2009)
19. Campbell, S.L., Chancelier, J.P., Nikoukhah, R.: *Modeling and Simulation in SCILAB*. Springer, New York (2006)
20. Bunks, C., Chancelier, J.P., Delebecque, F., Goursat, M., Nikoukhah, R., Steer, S.: *Engineering and Scientific Computing with Scilab*. Springer Science & Business Media (2012)
21. Sharma, N., Gobbert, M.K.: A Comparative Evaluation of Matlab, FreeMat, and Scilab for Research and Teaching. UMBC Faculty Collection, Octave (2010)
22. Bordeianu, C.C., Besliu, C., Jipa, A., Felea, D., Grossu, I.V.: Scilab software package for the study of dynamical systems. *Comput. Phys. Commun.* **178**(10), 788–793 (2008)

Appendix

Tuning of Controller Parameters Using Accelerated Particle Swarm Optimization

A.1 Introduction

This chapter focus on tuning of PID parameters (K_p , T_i & T_d), set-point weighting parameter (b) and fractional-order parameters (λ & μ) of the designed SWPI $^{\lambda}$ D $^{\mu}$ and PI $^{\lambda}$ -PD $^{\mu}$ controllers presented in Sects. 2.2.3 and 2.3. First, the standard SWPI $^{\lambda}$ D $^{\mu}$ controller parameters $\theta_{cp} = \{K_p, T_i, T_d, b, \lambda \text{ & } \mu\}$ presented in Sect. 2.2.1 will be tuned using accelerated particle swarm optimization (APSO) algorithm. Then for effective comparison, the controller parameters of industrial and parallel configurations of SWPI $^{\lambda}$ D $^{\mu}$ will be derived using the conversion relations developed in Sect. 2.2.7.

Furthermore, the PI $^{\lambda}$ -PD $^{\mu}$ controller parameters $\theta_{cp} = \{k_p, k_i, k_d, \lambda \text{ & } \mu\}$ are derived from parallel SWPI $^{\lambda}$ D $^{\mu}$ presented in Sect. 2.3.4. The summary of all these conversion relations developed in this research for the proposed fractional-order controllers is given in Fig. A.1. Therefore, the tuning of standard SWPI $^{\lambda}$ D $^{\mu}$ controller using APSO algorithm and the development of APSO from particle swarm optimization (PSO) algorithm is explained in the succeeding part of the section.

A.2 Accelerated Particle Swarm Optimization

The standard particle swarm optimization (PSO) is a metaheuristic optimization algorithm inspired by birds' flocking behavior [1, 2]. At the beginning of the algorithm, velocities and positions of a group of N birds (or particles) are initialized randomly within the dimensional search space D . In the next stage, each particle is attracted towards the current global best position G^* and its own best position S_t^* based on cost function J .

Therefore, the update of velocity and position equations for the PSO algorithm are given as follows:

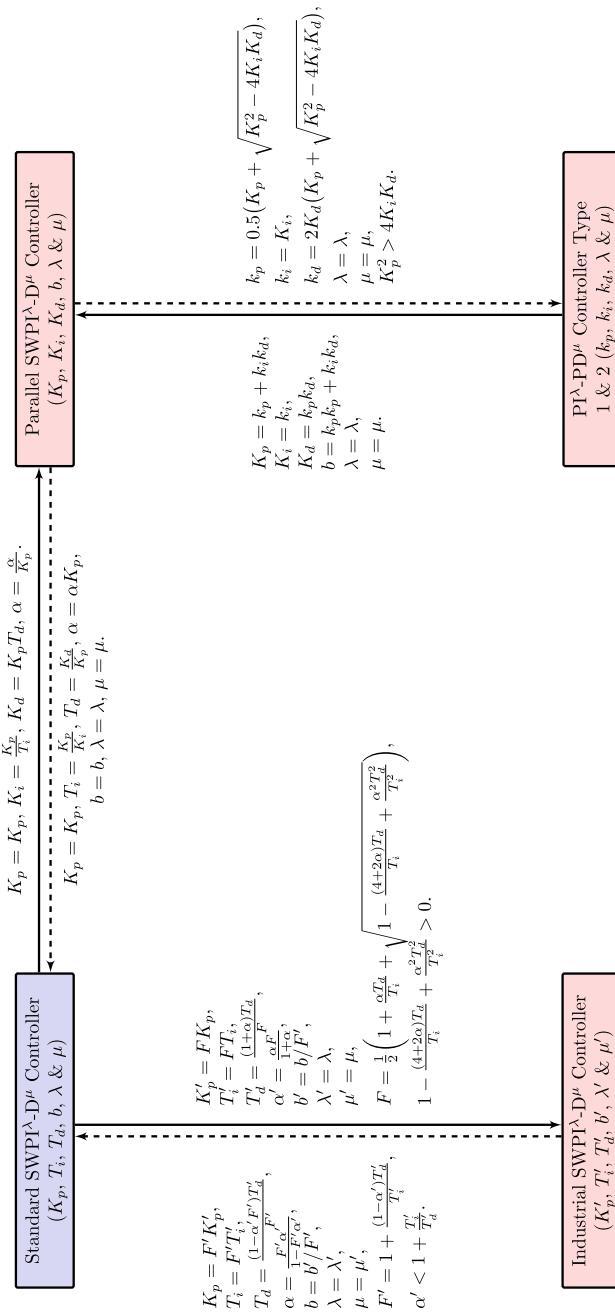


Fig. A.1 Conversion of SWPI $^\lambda$ -D $^\mu$ and PI $^\lambda$ -PD $^\mu$ controller parameters

$$V_i^{t+1} = V_i^t + \alpha \varepsilon_1 (G^* - S_i^t) + \beta \varepsilon_2 (S_i^* - S_i^t) \quad (\text{A.1})$$

$$S_i^{t+1} = S_i^t + V_i^{t+1} \quad (\text{A.2})$$

where

- S_i is the position of i th particle,
- V_i is the velocity of i th particle,
- ε_1 and ε_2 are random numbers $N(0, 1)$ and
- α and β are acceleration constants.

The use of local best position S^* in (A.1) is to increase the diversity of the solutions. However, using some randomness, this diversity can be simulated in the simplified APSO algorithm. This simplified version accelerates the convergence of PSO using global best only. Therefore, the velocity and position updates are achieved using the following equations:

$$V_i^{t+1} = V_i^t + \alpha \varepsilon_n + \beta (G^* - S_i^t) \quad (\text{A.3})$$

$$S_i^{t+1} = (1 - \beta) S_i^t + \beta G_i^t + \alpha \varepsilon_n \quad (\text{A.4})$$

A further advancement to the APSO is to reduce the randomness. This can be done by using a monotonically decreasing function as:

$$\alpha = \alpha_o \gamma^t \quad (\text{A.5})$$

where

- ε_n is the random number drawn from $N(0, 1)$,
- $\alpha = 0.1L \sim 0.5L$ is the acceleration constant of S_i ,
- L is the scale of each variable,
- $\beta = 0.1 \sim 0.7$ is the acceleration constant of search domain,
- $\alpha_o = 0.5 \sim 1$ is the starting value of the level of randomness and
- $0 < \gamma < 1$ is the control variable.

The pseudo-code for the accelerated particle swarm optimization (APSO) algorithm is given as follows:

Require: $N, birdStep, D, \alpha_o, \gamma, \beta$; **Output:** θ_{cp} ;

```

1: Initialization:
2: for  $i = 1$  to  $N$  do
3:   for  $d = 1$  to  $D$  do
4:      $S_i^d = rand(d, i)$ 
5:      $V_i^d = rand(d, i)$ 
6:   end for
7:    $\varepsilon_i = rand(i)$ 
8:    $P_i = S_i$ 
9:   if  $J(P_i) < J(G)$  then
10:     $G^* = P_i$ 
```

```

11: end if
12: end for
13: Iteration Steps:
14: repeat
15:   for  $i = 1$  to  $N$  do
16:     if  $J(S_i) < J(P_i)$  then
17:        $P_i = S_i$ 
18:     end if
19:     if  $J(P_i) < J(G)$  then
20:        $G^* = P_i$ 
21:     end if
22:   end for
23:   for  $i = 1$  to  $N$  do
24:     for  $d = 1$  to  $D$  do
25:        $V_i^d = V_i^d + \alpha \epsilon_i + \beta (G^* - S_i^d)$ 
26:        $S_i^d = (1 - \beta) S_i^d + \beta G^* + \alpha \varepsilon_i$ 
27:     end for
28:   end for
29:    $iter = iter + 1$ 
30:    $\alpha = \alpha_o \gamma^{iter}$ 
31: until  $iter > birdStep$ 

```

The block diagram of the SWPI $^\lambda$ D $^\mu$ controller based on the APSO algorithm for tuning the controller parameters of standard SWPI $^\lambda$ D $^\mu$ controller with an objective of minimizing error function (J) is given in Fig. A.2. In the figure, θ_{cp} denotes the controller parameters.

On the other hand, it is reported from the literature based on the previous works in [3–5], the commonly used objective functions for optimization are error-based time domain performance criteria. These are in the form of integral squared error (ISE), integral time squared error (ITSE), mean squared error (MSE), integral absolute error (IAE), integral time absolute error (ITAE). Furthermore, it should be noted that the integral criteria such as ISE, IAE, ITAE and ITSE often result in a better closed-loop response of a control system.

The ISE criteria should be used where there is a large deviation of error from the set-point. In the case of small deviations, IAE criterion is used. On the other hand, when the error persists for a long time, ITAE criterion helps. This is because multiplying the error with time to augment the error function (J) term at high values. Therefore, in this work for the purpose of optimization, the performance indices used are the integral absolute error (IAE), integral squared error (ISE) and integral time absolute error (ITAE) as given in the following equations:

$$J_{IAE} = \int_0^T |e(\tau)| d\tau \quad (\text{A.6})$$

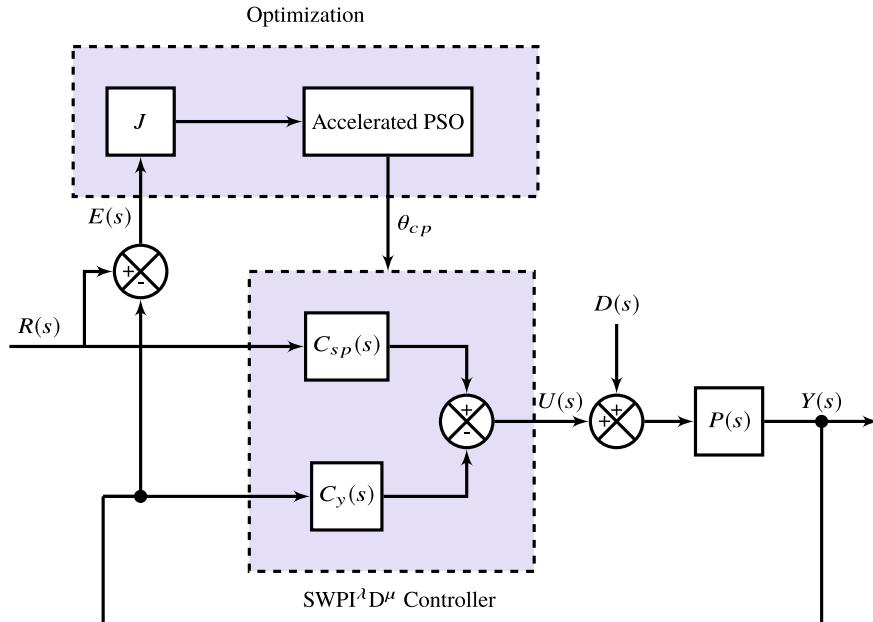


Fig. A.2 Block diagram for tuning of $\text{SWPI}^\lambda \text{D}^\mu$ controller

$$J_{ISE} = \int_0^T e(\tau)^2 d\tau \quad (\text{A.7})$$

$$J_{ITAE} = \int_0^T \tau |e(\tau)| d\tau \quad (\text{A.8})$$

References

1. Yang, X.S.: *Engineering Optimization: An Introduction with Metaheuristic Applications*. Wiley (2010)
2. Yang, X.S., Deb, S., Fong, S.: Accelerated particle swarm optimization and support vector machine for business optimization and applications. In: International Conference on Networked Digital Technologies, Macau, China, 11–13 July 2011
3. Vilanova, R., Visioli, A.: *PID Control in the Third Millennium*. Springer (2012)
4. Xue, D., Chen, Y., Atherton, D.P.: *Linear Feedback Control: Analysis and Design with MATLAB*. Siam (2007)
5. Sahib, M.A., Ahmed, B.S.: A new multiobjective performance criterion used in PID tuning optimization algorithms. *J. Adv. Res.* **7**(1), 125–134 (2016)

Index

A

Accelerated particle swarm optimization, 247, 249
Acetic acid, 39, 42, 43
Agitator, 40
AI, *see* analog input
Analog input, 44
Analog output, 44
AO, *see* analog output
Approximation table, 114
APSO, *see* accelerated particle swarm optimization
Asymptotic Bode plot, 164
Automatic voltage regulator, 19

B

Binomial expansion, 155
Bode, 11, 107, 111, 120, 137, 138, 162, 164, 182–184, 190, 195, 200

C

Caputo, 17, 139
Carlson, 3, 101, 102
CFE, *see* continued fractional expansion
Chaotic attractor, 231, 233–236, 238–243
Chaotic behavior, 225
Charef, 102
Chemical mixing tank, 39
Commande Robuste d'Ordre Non Entier (CRONE), 17, 137, 138
Commensurate orders, 231, 238, 242
Component Separated type, 29
Conductivity transmitters, 40
Continued fractional expansion, 3, 101, 102

Continuous stirred-tank reactor, 13, 39
CSTR, *see* continuous stirred-tank reactor
Curve fitting approximation, 102, 143

D

Derivative kick effect, 1, 11, 13, 14, 19, 22, 26, 47, 48, 51, 55, 58, 61, 62, 73, 77, 80, 82, 85, 94, 95
Deterioration rate, 85, 94, 95
DFOC, 138
Differential calculus, 1
DISO, *see* dual-input-single-output
Disturbance rejection, 9, 11, 14, 20, 22, 23, 30, 34–37, 44, 45, 47, 49–55, 57–61, 63, 65, 67, 69, 72, 75, 76, 78, 80, 81, 83, 84, 86, 87, 89, 90, 92, 94, 95
Dual-input-single-output, 13, 95

E

Electroneutrality principle, 43
EnPID, 9
Equivalent circuit implementation, 2, 17, 101

F

Feedback, 14, 15, 17, 18, 27, 28, 95
Feedforward, 14, 27
Field-programmable gate array, 19
Finite impulse response, 101
FIR, *see* finite impulse response
FIT, *see* fractional integration tool
Fixed-pole approximation, 101
Flow control valve, 44

Flow transmitters, 39
 FOTF, *see* Fractional-order Transfer Function
 FPGA, *see* field-programmable gate array
 Fractional calculus, 1, 2
 Fractional integration tool, 137
 Fractional State Space Tool (FSST), 17, 137, 138
 Fractional-order all-pass filter, 152
 Fractional-order band-pass filter, 151
 Fractional-order capacitor, 19
 Fractional-order chaotic oscillators, 214
 Fractional-order chaotic systems, 213
 Fractional-order Chen, 237
 Fractional-order controllers, 1, 17
 Fractional-Order Differentiator (FOD), 145
 Fractional-order duffing, 219
 Fractional-order filters, 148
 Fractional-order high-pass filter, 150
 Fractional-order impedance, 19
 Fractional-Order Integrator (FOI), 146
 Fractional-order Lorenz, 227
 Fractional-order low-pass filter, 149
 Fractional-Order Modeling and Control (FOMCON), 17, 137, 138, 210
 Fractional-order Rössler, 241
 Fractional-Order Transfer Function (FOTF), 147
 Fractional-Order Van der Pol, 214
 Frequency distribution mode, 102
 Frequency response data, 102, 104, 105, 143
 Fundamental modeling, 39
 Fuzzy, 14

G

Grünwald-Letnikov, 17, 139, 163, 166, 214, 221, 228, 237, 241
 GUI, 138

H

Hand valve, 63
 Heuristic optimization, 13
 H_∞ -norm, 106, 164
 H_2 -norm, 106, 164

I

IAE, *see* integral absolute error
 Ideal structure, 23
 IIR, *see* infinite impulse response
 IMC-FPID, 156, 158, 163
 Improper, 101

Industrial structure, 25
 Infinite impulse response, 101
 Integral absolute error, 250
 Integral squared error, 250
 Integral time absolute error, 45, 49, 250
 Integral time squared error, 250
 Ionic product of water, 43
 ISE, *see* integral squared error
 Iso-damping, 2, 11, 16, 17, 137
 ITAE, *see* integral time absolute error
 ITSE, *see* integral time squared error

K

Kalman filter, 138

L

Level control, 19
 Level sensors, 40
 Levy's cost function, 103, 133
 L_∞ -norm, 164
 Load frequency, 19
 Load regulation, 2, 10, 20, 32, 47, 60, 94, 95

M

Maclaurin Series, 101
 MAE, *see* mean absolute error
 Magnitude plot, 164
 Mass balance, 43
 Matrix Laboratory (MATLAB), 17, 137, 138
 Matsuda, 101, 142
 Mean absolute error, 105, 107, 111, 121, 124, 127, 131
 Metaheuristic optimization algorithm, 247
 Model predictive control, 17
 Model reference adaptive control, 17
 Mod FPI, 157, 158
 Mod FPID, 158
 Modified PID controllers, 11, 13

N

Nichols, 138, 162, 164, 175, 195
 Noise filter, 17
 Non-commensurate orders, 233, 240, 242
 Non-Integer (NINTEGER), 17, 137, 138
 Non-linear PID (NPID), 9, 11
 Non-linear stable spherical tank, 177, 188
 Nyquist, 138, 162, 164, 175, 195

O

Optimal control, 17

Oscillatory, 1, 9, 11, 14, 15, 20, 49, 54, 58, 61, 72, 76, 80, 85, 94, 148
Oustaloup, 3, 102, 140, 163
Overshoot, 11, 14, 22, 44, 45, 49, 52, 54, 58, 60, 67, 72, 73, 76, 78, 82, 83, 85, 94, 95, 190, 194

P
Parallel structure, 24
Particle swarm optimization, 247
PCI, *see* Peripheral Component Interconnect
PCV, *see* process control valve
 $(PD)^n$, 155
 PD^μ , 158
 $[PD]^\mu$, 155, 158, 183, 184
Peak frequencies, 164
Peripheral Component Interconnect (PCI), 44, 64
PH, 39, 40, 43, 45, 64
PH Neutralization, 39
PH transmitter, 40
Phase plot, 164
Phase portraits, 215, 218, 223, 226
 PI^λ , 158
 $PI^\lambda-PD^\mu$, 35–39, 51, 53–57, 59, 61–65, 67, 69, 78, 79, 81–83, 85–95, 247, 248
 $PI^\lambda-PD^\mu 1$, 35, 55–57, 59–61, 64, 66, 68, 81–83, 87, 90, 94
 $PI^\lambda-PD^\mu 2$, 37, 55–61, 64, 66, 68, 81–83, 85, 87, 90, 94
 $PI^\lambda D^\mu$, 2, 9–11, 15–18, 22, 37, 44, 45, 47–50, 52, 55–58, 60–62, 64–66, 68, 69, 72–77, 79, 81, 82, 87, 90, 94, 101, 126, 129, 130, 133, 158, 163, 182, 188, 193, 195
PI-PD, 2, 4, 9–11, 13–15, 20, 32–35, 37, 38, 52–58, 62–65, 78–80, 82–85, 89–91, 94, 95
PI-PD1, 33, 53–57, 61, 64, 66, 68, 80–83, 85, 90, 94
PI-PD2, 34, 53–57, 61, 64, 66, 68, 80, 82, 83, 85, 90, 94
PIC, *see* pressure indicating controller
 $(PI)^n$, 17, 155, 158
 $(PID)^n$, 154, 158
P&ID, *see* Process and Instrumentation Diagram
 $PID\tau_d$, 10, 11
 $[PI]^\lambda$, 155, 183, 184, 188
 $[PI]^\mu$, 158
Pole zero plot, 163
Power series expansion, 3, 17, 101, 102, 144, 145, 169

PPI, 9, 11
Practical realization, 2, 17, 101
Pre-filter, 14, 27, 29
Pressure indicating controller, 63
Pressure process plant, 62
Pressure transmitter, 63
Process and Instrumentation Diagram (P&ID), 39, 63
Process control valve, 63
PSO, *see* particle swarm optimization

R
Refined Oustaloup, 141
Relay-based auto-tuning approach, 14
Riemann-Liouville (RL), 17, 138, 139
Rise time, 44, 46, 50, 52, 54, 56, 58, 60, 65, 72, 78, 82, 83, 85, 159, 190, 194
Robotic manipulator, 19
Robustness, 11, 14, 18

S
Sanathanan-Koerner (SK), 103, 105
Scilab, 139, 141, 145, 147, 150, 158, 164, 176, 193–195, 205, 208, 209, 214, 215, 221, 229, 238, 243, 244
Second order stable process plant, 206
Set-point tracking, 2, 10, 11, 13, 18, 20, 22, 30, 32, 34–37, 44, 45, 47, 49, 51–55, 57–59, 61, 63, 65, 67, 69, 73, 75, 76, 78, 80–82, 84–95
Set-point weighting parameters, 13, 31
Settling time, 44, 47, 50, 52, 54, 56, 58, 60, 65, 72, 75, 76, 78, 82, 83, 85, 94, 159, 161, 190, 194
Simulink, 17, 44, 64, 138
Single-input-single-output, 13, 27, 95
Singular-value sigma-plot, 164
SISO, *see* single-input-single-output
Sliding mode control, 17
Sodium hydroxide, 39, 42
Stability boundary locus, 3, 102
Standard structure, 20
State-space approach, 102
Steady-state, 31, 157
Step info, 159, 164
 $SWPI^\lambda-D^\mu$, 32, 33, 39, 47, 48, 50, 52, 59–62, 66, 68, 75, 77, 79, 81, 90, 94
 $SWPI^\lambda D^\mu$, 20, 22–24, 26, 27, 37–39, 44, 45, 47, 49, 51, 53, 59, 62, 65, 67, 69, 72–80, 82, 87, 92–94, 247, 248, 250
SWPID, 9–11, 13
Sysquake, 17, 137, 138

T

Taylor series, 101
TID, *see* tilt-integral-derivative
Tilt-integral-derivative, 19
Time moments approach, 18, 102
Titration, 39, 43
Transfer function estimation, 105, 143
Transient response, 31
Twin-rotor system, 19

U

Unstable bioreactor processes, 177, 193

V

Vector fitting method, 102

W

Wolfram Mathworld, 17

X

Xcos, 139, 208, 209

Z

Ziegler-Nichols (ZN), 67
ZN, *see* Ziegler-Nichols