

FAULT DETECTION AND FAULT-TOLERANT CONTROL OF SINGLE-ROD ELECTROHYDROSTATIC ACTUATED SYSTEM

by
AMIRREZA MIRBEYGI MOGHADDAM

A Thesis
Submitted to the Faculty of Graduate Studies
in Partial Fulfillment of the Requirements
for the Degree of

MASTER OF SCIENCE



Department of Mechanical Engineering
University of Manitoba
Winnipeg, MB, R2M 0X4.

© A. Mirbeygi Moghaddam, August 2021
134 pages

Abstract

Single-rod electrohydrostatic actuators present an efficient alternative to valve-controlled systems. However, such systems have significant nonlinearities and as is the case with hydraulic circuits, the possibility of faults exists in the system. Since these faults cannot be seen directly and have significant effects on the overall behaviour of the *Electrohydrostatic Actuator* (EHA), it is of crucial importance that they are detected and the performance of the EHA is restored. These behavioural complexities along with its asymmetrical dynamics, make the control task of the single-rod EHA challenging.

In this thesis, by acquiring accurate signals from a single-rod EHA using a novel fuzzy denoising method a fault detection analysis is performed in a multi-fault environment. To do so *Variance Fractal Dimension* (VFD), *Length Fractal Dimension* (LFD) and wavelet detail coefficients are utilized. Also, the severity of the faults (bulk modulus and internal leakage) is associated with the aforementioned measures. Building upon this comprehensive analysis, a fault decision algorithm is developed to allocate a quantitative value to the internal leakage severity in the system if such fault occurs.

With regards to control, first, a healthy condition and a faulty condition *Fractional-Order Proportional Integral Derivative* (FOPID) controller are developed and optimized using the *Modified Nelder-Mead* (MNM) algorithm. Using these two FOPID controllers, then, the fault-tolerant controller is designed which uses the generated degree of fault variable from the fault detection algorithm to assign a weight to each of the inputs from the two controllers through a fuzzy inference system.

The effectiveness of the fault detection and control strategies is demonstrated in several experimental results.

Acknowledgements

First and foremost, I owe a great dept of gratitude to my research supervisor, Dr. Nariman Sepehri, whose help and support have been there every step of the way. Working under his supervision was a privilege. My deepest gratitude is extended to Dr. Witold Kinsner whose teachings have made me grow as a researcher. Many thanks to Dr. Xihui Liang for serving on my M.Sc. examining committee and for their constructive comments on this thesis.

I would also like to acknowledge Dr. Gustavo Koury Costa, Dr. Ehsan Jalayeri, Dr. Khurram Mahmood Butt, Dr. Ali Maddahi, and my other peers in the “Fluid Power and Telerobotics Research Laboratory” at the University of Manitoba.

Special acknowledgment and recognition are given to my family for their endless love and support.

TABLE OF CONTENTS

Abstract.....	i
Acknowledgements	ii
TABLE OF CONTENTS.....	iii
LIST OF FIGURES.....	v
LIST OF TABLES.....	x
LIST OF NOMENCLATURES.....	xi
LIST OF ACRONYMS.....	xv
1. Introduction	1
1.1. Motivation.....	1
1.2. Problem Statement.....	1
1.3. Proposed Solutions	2
1.4. Thesis Formulation	3
1.5. Methodology.....	3
1.6. Thesis Organization	4
2. Mathematical Model.....	5
2.1. Excavator Arm.....	7
2.2. Hydraulic Circuit	8
2.3. Data Acquisition and Control	12
2.4. Model Verification.....	12
2.5. A Modification to The Control Scheme of The Directional Valve	13
2.6. Summary.....	19
3. CHAPTER 3: ACQUISITION AND GENERATION OF ACCURATE SIGNALS.....	20
3.1. Signal Denoising.....	20
3.1.1. Wavelet Transform	21
3.1.2. Denoising	22
3.1.3. Proposed Thresholding Method	23
3.2. Performance of The Proposed Fuzzy Denoising Method.....	26
3.3. Raspberry Pi-based Data Acquisition and Control.....	32
3.3.1. Architecture and Components.....	33
3.3.2. Programming.....	39

3.4. Experimental Results for Comparison of The Data Acquisition Systems.....	41
3.4.1. Accurate Acquisition of Pressure Signals in Operator-generated Input Mode.....	41
3.4.2. High-speed Performance in Operator-generated Input Mode.....	46
3.4.3. Performance in Operator-generated Reference Mode	48
3.5. Summary.....	50
4. Chapter 4: Fault Detection	52
4.1. Analysis Techniques.....	52
4.1.1. Stationarity Test.....	53
4.1.2. Variance and Length Fractal Dimensions.....	53
4.1.3. Wavelet Coefficients.....	54
4.2. Fault Detection Development in A Multi-fault Environment.....	56
4.2.1. Fault Detection in Non-operating Mode.....	57
4.2.2. Fault Detection in Operating Mode	62
4.3. Sensitivity to Faults with Variable Severity	70
4.3.1. Internal Leakage.....	70
4.3.2. Bulk Modulus Change	76
4.4. Degree of Fault Variable and Algorithm.....	82
4.5. Summary.....	89
5. Chapter 5: Fault-tolerant Control Design	90
5.1. Approximation of Fractional-order Transfer Functions	90
5.2. Modified Nelder-Mead Optimization Algorithm	92
5.3. Control Design for The Healthy system	94
5.4. Design of The Fault-tolerant Controller	101
5.4.1. Parameter Tuning for Fractional-order PID Control, Internal Leakage Fault	101
5.4.2. Fault-tolerant Velocity Control in The Presence of Internal Leakage Fault.....	103
5.4.3. Experiments	104
5.5. Summary.....	111
6. Chapter 6: Conclusions	112
6.1. Thesis Contributions.....	112
6.2. Potential Future Work.....	113
7. References.....	114
Appendix A	118

LIST OF FIGURES

Fig. 2.1 Experimental setup (single-rod EHA) subsystems: a) Hydraulic circuit, b) Excavator arm, and c) data acquisition and control.	5
Fig. 2.2 Experimental setup (single-rod EHA) subsystems.	6
Fig. 2.3 Experimental setup: (a) photograph of the experimental test bench; (b) schematic of the excavator arm.	8
Fig. 2.4 Schematic of the electrohydrostatic actuator circuit. (After [2]).....	8
Fig. 2.5 Four quadrants of the electrohydrostatic actuator system.	10
Fig. 2.6 Data acquisition and control subsystem inputs and outputs.	12
Fig. 2.7 Simulation vs experimental results in pre-generated input mode.....	14
Fig. 2.8 Additional results for the simulation shown in Fig. 2.7.	15
Fig. 2.9 Simulation vs experimental results in operator-generated input mode.	16
Fig. 2.10 Additional results for the simulation shown in Fig. 2.9.	17
Fig. 2.11 Hysteresis applied to switching.	18
Fig. 2.12 Effect of hysteresis on the velocity, experimental results in pre-generated input mode.	18
Fig. 3.1 Discrete wavelet transform algorithm.	21
Fig. 3.2 The expected behaviours in the proposed thresholding method compared to conventional thresholding functions.	24
Fig. 3.3 Membership functions for the proposed fuzzy thresholding.	24
Fig. 3.4 Comparison between the proposed thresholding method and the conventional thresholding functions.....	25
Fig. 3.5 Effect of a) α , and b) β , on the denoising behaviour of the proposed method.	26
Fig. 3.6 Results of denoising for the generated signal, $S(x)$, using simulations.	27
Fig. 3.7 Epoch denoising results for the word "measure", subject f06. a) original signal, b) signal with noise, c) proposed fuzzy threshold, d) soft threshold, and e) hard threshold.....	29
Fig. 3.8 Epoch denoising results for the word "test", subject f09. a) original signal, b) signal with noise, c) proposed fuzzy threshold, d) soft threshold, and e) hard threshold.	30
Fig. 3.9 Results of denoising the pressure signals in an operator-generated experiment.	31
Fig. 3.10 Raspberry Pi-based data acquisition and control.....	32
Fig. 3.11 Data acquisition and control subsystem components.	33
Fig. 3.12 Circuit design of the data acquisition and control system.	36

Fig. 3.13 P-Q controls joystick and its axis.	37
Fig. 3.14 Power spectral density analysis of the pressure signal, PA	39
Fig. 3.15 Input (from joystick) to the experimental setup.	42
Fig. 3.16 Pressure responses pertaining to input shown in Fig. 3.15.	42
Fig. 3.17 Pressure responses (between $t=15s$ to $t=40s$) pertaining to input shown in Fig. 3.15.	43
Fig. 3.18 Pressure responses with noise reduction pertaining to input shown in Fig. 3.15.	44
Fig. 3.19 Pressure responses with noise reduction (between $t=15s$ to $t=40s$) pertaining to input shown in Fig. 3.15.	44
Fig. 3.20 Statistical moments of the pressure signal, PB , pertaining to Fig. 3.18.	45
Fig. 3.21 Statistical moments of the pressure signal, PA , pertaining to Fig. 3.18.	46
Fig. 3.22 Experimental results in operator-generated input mode with Raspberry Pi data acquisition system.	47
Fig. 3.23 Experimental results of case 1 velocity control test in operator-generated reference mode (using joystick) with Raspberry Pi data acquisition system.	49
Fig. 3.24 Experimental results of case 2 velocity control test in operator-generated reference mode (using joystick) with Raspberry Pi data acquisition system.	50
Fig. 4.1 Algorithm for obtaining variance and length fractal dimensions.	55
Fig. 4.2 Algorithm for obtaining variance and length fractal dimension trajectories.	56
Fig. 4.3 Stationarity test for pressure signals, PA , for a) healthy system, b) system with internal leakage, and c) system with bulk modulus change in non-operating mode, simulation.	57
Fig. 4.4 Fractal dimension comparison for a healthy and faulty EHA in non-operating mode pertaining to Fig. 4.3.	58
Fig. 4.5 Wavelet detail coefficients of pressure signals, PA , for a EHA in non-operating mode for a healthy system pertaining to Fig. 4.3.	59
Fig. 4.6 Wavelet detail coefficients of pressure signals, PA , for a EHA in non-operating mode for a system with internal leakage pertaining to Fig. 4.3.	60
Fig. 4.7 Wavelet detail coefficients of pressure signals, PA , for a EHA in non-operating mode for a system with bulk modulus change pertaining to Fig. 4.3.	61
Fig. 4.8 Close-up view of wavelet detail coefficients of pressure signals, PA , for a healthy and faulty EHA in non-operating mode pertaining to Fig. 4.3.	62
Fig. 4.9 Stationarity test for jerk signals, $jerk$, for a) healthy system, b) system with internal leakage, and c) system with bulk modules change in operating mode, simulation.	63

Fig. 4.10 Fractal dimension comparison for a healthy and faulty EHA in operating mode pertaining to Fig. 4.9.....	64
Fig. 4.11 Wavelet detail coefficients of jerk signal, <i>jerk</i> , for a healthy EHA in operating mode pertaining to Fig. 4.9.....	65
Fig. 4.12 Wavelet detail coefficients of jerk signal, <i>jerk</i> , for an EHA with internal leakage fault in operating mode pertaining to Fig. 4.9.	66
Fig. 4.13 Wavelet detail coefficients of jerk signal, <i>jerk</i> , for an EHA with bulk modulus change fault in operating mode pertaining to Fig. 4.9.	67
Fig. 4.14 Level 1 wavelet detail coefficients' statistical characteristics for a healthy and faulty EHA in operating mode pertaining to Fig. 4.9.....	68
Fig. 4.15 Level 4 wavelet detail coefficients' statistical characteristics for a healthy and faulty EHA in operating mode pertaining to Fig. 4.9.....	69
Fig. 4.16 Experimental results for healthy and faulty EHAs with internal leakage in operating mode.....	71
Fig. 4.17 Fractal dimensions for healthy and faulty EHAs with internal leakage in operating mode pertaining to Fig. 4.16.....	71
Fig. 4.18 Experimental results for healthy and faulty EHAs with internal leakage in non-operating mode pertaining to Fig. 4.16.....	73
Fig. 4.19 Wavelet approximation coefficients of systems with different internal leakage levels pertaining to Fig. 4.16.....	74
Fig. 4.20 Wavelet detail coefficients of systems with different internal leakage levels pertaining to Fig. 4.16.....	75
Fig. 4.21 Simulation results for systems with different bulk moduli.....	76
Fig. 4.22 Length and Variance dimensions for systems with different bulk moduli pertaining to Fig. 4.21.....	77
Fig. 4.23 Wavelet detail coefficients of systems with bulk modulus $\beta_{0.5} = 344.5\text{MPa}$ pertaining to Fig. 4.21.....	78
Fig. 4.24 Wavelet detail coefficients of systems with bulk modulus $\beta_1 = 689\text{MPa}$ pertaining to Fig. 4.21.....	79
Fig. 4.25 Wavelet detail coefficients of systems with bulk modulus $\beta_{1.5} = 1033.5\text{MPa}$ pertaining to Fig. 4.21.....	80
Fig. 4.26 Wavelet detail coefficients of systems with bulk modulus $\beta_2 = 1378\text{MPa}$ pertaining to Fig. 4.21.....	81

Fig. 4.27 Statistical moments of wavelet detail coefficients of systems with different bulk moduli pertaining to Fig. 4.21.....	82
Fig. 4.28 Results of first experiments for validating the fault decision algorithm and DoF variable.....	85
Fig. 4.29 Results of second experiments for validating the fault decision algorithm and DoF variable.....	86
Fig. 4.30 Results of third experiments for validating the fault decision algorithm and DoF variable.....	87
Fig. 4.31 Additional results of third experiments for validating the fault decision algorithm and DoF variable.....	88
Fig. 5.1 Block diagram of the controller and system in healthy condition in pre-generated reference mode.....	95
Fig. 5.2 Optimization results for three cases: a) Case 1, b) Case 2 and c) Case 3.....	96
Fig. 5.3 Close-up view of optimization results for three cases: a) Case 1, b) Case 2 and c) Case 3.....	97
Fig. 5.4 Results of simulation for fractional-order PID control in pre-generated reference mode.....	99
Fig. 5.5 Experimental results for fractional-order PID control in pre-generated reference mode with reference velocity of 0.05 m/s, weights of 0 kg, 163 kg and 367 kg.....	100
Fig. 5.6 Experimental results for fractional-order PID control in pre-generated reference mode with reference velocity of 0.05 m/s, weights of 0 kg, 163 kg and 367 kg.....	101
Fig. 5.7 Block diagram of fuzzy-FOPID fault tolerant control scheme.....	103
Fig. 5.8 Inputs membership functions of the fuzzy inference system for fault-tolerant control strategy.....	104
Fig. 5.9 Results of the fault-tolerant control experiment in pre-generated reference mode, test 1.....	105
Fig. 5.10 Results of the fault-tolerant control experiment in pre-generated reference mode, test 2.....	106
Fig. 5.11 Results of the fault-tolerant control experiment in pre-generated reference mode, test 3.....	107
Fig. 5.12 Results of the fault-tolerant control experiment in pre-generated reference mode, test 4.....	108
Fig. 5.13 Results of the fault-tolerant control experiment in pre-generated reference mode, test 5.....	109

Fig. A.1 Schematic of the movement of the excavator arm..... 118

LIST OF TABLES

Table 2.1 Elements of the electrohydrostatic actuator circuit shown in Fig. 2.4.....	9
Table 2.2 Parameters of the experimental setup [3].	12
Table 2.3 Spikes with respect to hysteresis as shown in Fig. 2.12 at t=15s.	19
Table 3.1 Results of the denoising for the "measure" epoch as shown in Fig. 3.7.....	28
Table 3.2 Results of the denoising for the "test" epoch as shown in Fig. 3.8.....	28
Table 3.3 Signal properties of data acquisition and control subsystem shown in Fig. 3.11.	34
Table 3.4 Voltage supply needed for circuit components shown in Fig. 3.13.....	38
Table 3.5 Electric circuit's elements.....	38
Table 3.6 Config register of ADS1115/1015 [22, 23].	40
Table 3.7 Results of velocity control experiments using Raspberry Pi data acquisition system.	48
Table 4.1 Variance and length fractal dimension averages for faulty and healthy Conditions.	59
Table 4.2 Mean of fractal dimensions, internal leakage experimental data.....	73
Table 4.3 Mean of fractal dimensions for systems shown in Fig. 4.22 at t= [62s - 68s].	77
Table 4.4 Constants for DoF calculation in fault decision algorithm.	84
Table 5.1 Optimization parameters for the Tuning Simulations.....	95
Table 5.2 Controller Tuned Parameters in simulation and experiments for the healthy control system.	95
Table 5.3 Experimentally Tuned Parameters for the healthy control system.	97
Table 5.4 RMS errors of experimental tests.	100
Table 5.5 Results of online optimization of the system with internal leakage.	102
Table 5.6 Experimentally Tuned Parameters for the control system with internal leakage. .	102

LIST OF NOMENCLATURES

F_L	Load force (N)
l_1	Upper arm length (m)
θ	The angular displacement of linkage (rad)
m_L	Load mass (Kg)
g	Gravational accelaration (m/s ²)
l_2	Lower arm length (m)
I	Momen of inertia for the arm (Kg.m ²)
α_r	Angular acceleration (rad/s ²)
x_p	Piston displacement (m)
\dot{x}_p	Piston velocity (m/s)
\ddot{x}_p	Piston acceleration (m/s ²)
A_a	Cap-side area of piston (m ²)
A_b	Rod-side area of piston (m ²)
α_{ab}	Area ratio of the actuator
V_d	Displacement of the bidirectional pump (m ³ /rev)
τ_m	The time constant of the servo motor (1/s)
K_m	Servomotor gain (rev/(s·V))
β_e	Effective bulk modulus (Pa)
C_v	Hydraulic compliance (m ³ /Pa)
m_{rod}	Piston and rod mass (kg)
f	Viscous damping coefficient(N·s/m)
K_l	internal leakage coefficient (m ³ / $\sqrt{\text{Pa}}$ · s)
p_L	Load pressure (Pa)
p_a, P_A	Cap-side pressure (Pa)
p_b, P_B	Rod-side pressure (Pa)
Q_1	flow into the pump (m ³ /s)
Q_2	flow out of the pump (m ³ /s)
Q_l	Leakage flow (m ³ /s)
Q_{ac}	Compensation flows coming from the charge pump to the cap-side (m ³ /s)

Q_{bc}	Compensation flows coming from the charge pump to the rod-side (m ³ /s)
V_{oa}	Cap-side inner pipe and cylinder chamber volume (m ³)
V_{ob}	Rod-side inner pipe and cylinder chamber volume (m ³)
N	The number of objects (samples, points, etc.)
Ω	Hysteresis constant (Pa)
$\psi()$	Mother wavelet function
$h[n]$	Wavelet low-pass filter coefficient
$g[n]$	Wavelet high-pass filter coefficient
$a_j[n]$	Approximation coefficient at level j
$d_j[n]$	Detail coefficient at level j
λ_j^{fixed}	The fixed threshold at level j
$\text{sgn}()$	Sign function
\bar{x}	Thresholded value if x
$\mu_i(x)$	Membership function for a fuzzy inference system
α, β	Parameters of the fuzzy denoising method
V_{in}	Input voltage to a circuit element (V)
V_{out}	Output voltage of circuit element (V)
R_i	Resistance of resistor i (Ω)
$\exp()$	Expected value function
$LPF(s)$	Low-pass filter transfer function
ω_c	Cutt-off frequency of low-pass filter (rad/sec)
K_p	Proportional gain
K_I	Integral gain
K_D	Derivative gain
D_L	Length Dimension
D_σ, D_v	Variance Dimension
\bar{M}_σ	Threshold of mean value of variance fractal dimension for healthy system
\bar{M}_l	Threshold of mean value of length fractal dimension for healthy system
\bar{M}_4	Threshold of the mean value of level 4 detail coefficients dimension of healthy system
\bar{A}_σ	Threshold of absolute maximum value of variance fractal dimension for healthy system

\bar{A}_l	Threshold of absolute maximum value of length fractal dimension for healthy system
\bar{A}_4	Threshold of absolute maximum value of level 4 detail coefficients dimension of healthy system
\bar{V}_σ	Threshold of variance value of variance fractal dimension for healthy system
\bar{V}_l	Threshold of variance value of length fractal dimension for healthy system
\bar{V}_4	Threshold of variance value of level 4 detail coefficients dimension of healthy system
C_{DoF}	The normalization constant
ω_u	Unit frequency gain used in Oustaloup method (<i>rad/sec</i>)
ω_b	Low transitional frequency (<i>rad/sec</i>)
ω_h	High transitional frequency (<i>rad/sec</i>)
μ	Exponent of the integral term
λ	Exponent of the derivation term
ε	Reflection coefficient in MNM algorithm
γ	Expansion coefficient
ρ	Contraction coefficient
σ	Shrinkage coefficient
α_i	Dimensional parameters in the initial simplex
S	Region of interest
IS	Initial simplex of a run
V_i	Vertices of n-dimensional search space
R	Number of restarts
X_c	Centroid of the search
P	Projected points
J_{n*m}	A matrix in which all the elements are ones.
M	Memory matrix
P₀	Points that leads to the minimum objective function
OF	Objective function
<i>error</i>	Velocity error with respect to reference (m/s)
<i>jerk</i>	Motion jerk of the piston (m/s ³)
w_E	Error term weight in the objective function
w_J	Jerk term weight in the objective function

M	Number of poles and zeros multiplier in the Oustaloup method
Λ_{max}	FOPID parameters' upper bounds
Λ_{min}	FOPID parameters' lower bounds
N_{max}	Maximum number of iterations for each run of the MNM algorithm
U_H	Input from the healthy condition FOPID (V)
U_H	Input from the faulty condition FOPID (V)
U	Final input of the controller (V)
DoF_{set}	DoF which is reserved for the fault-tolerant controller after the 20s window

LIST OF ACRONYMS

PID	Proportional integral derivative
EHA	Electrohydrostatic actuator
VFD	Variance fractal dimension
LFD	Length fractal dimension
FOPID	Fractional-order proportional integral derivative
MNM	Modified Nelder-Mead
DAQ	Data acquisition
RMSE	Root mean square error
ADC	Analog to digital conversion
DAC	Digital to analog conversion
WT	Wavelet transform
CWT	Continuous wavelet transform
DWT	Discrete wavelet transform
MSD	Multiresolution signal decomposition
SNR	Signal to noise ratio
SURE	Stein's unbiased risk estimate
FIS	Fuzzy inference system
I/O	Input/output
HSC	Human signal controller
SSC	System signal controller
MPCL	Multiprocessing communication line
IC	Inter-integrated circuit
UART	Universal asynchronous receiver-transmitter
LPF	Low-pass filter
LSB	Least significant bit
MSB	Most significant bit
SDA	Serial data
SCL	Serial clock
TUE	Total unadjusted risk
WST	Wavelet spectrum test

VFDT	Variance fractal dimension trajectory
LFDT	Length fractal dimension trajectory
DoF	Degree of fault

CHAPTER 1:

INTRODUCTION

1.1. Motivation

Hydraulic actuators are commonly used in the industry due to the wide range of useful characteristics that they offer [1]. These actuators provide high force-to-weight ratios, high stiffness and high payload capacity. Often, since these actuators are controlled by valves; they are expensive to implement and highly inefficient due to the throttling losses of control valves. A proven solution to this problem is the use of bidirectional pumps for directly controlling the flow in the system. In these pump-controlled circuits, pumps replace the valves in driving the hydraulic actuators, which in turn mitigate the throttling loss problem [1].

A pump-controlled system uses a bi-directional pump as the means of driving the hydraulic actuator. Such systems where the two sides of the pump are connected to the two ports of the actuator are called electrohydrostatic actuators, Electrohydrostatic Actuator (EHA). In these systems, the movement of the actuator is controlled by the electric motor which controls the speed and direction of the pump. As opposed to valve-controlled systems, EHAs are light and have low oil pollution, they are easily implemented and have a simple structure [1]. The EHAs come with their disadvantages. A challenge related to the asymmetric shape of the single-rod EHA is the uneven flow for the two sides of the cylinder. This was addressed in [2] where the uneven flow was compensated by a novel circuit design for single-rod EHAs. The newly designed circuit [2] for controlling the single-rod EHA system is used in this thesis for controlling the actuator.

In the case of single-rod EHAs, these systems have high degrees of nonlinearity and asymmetric behaviour. Also, faults such as internal leakage and bulk modulus change could appear in the EHAs.

1.2. Problem Statement

Two main problems are to be addressed in this work.

With regards to control, The EHA systems have various uncertainties which appear in the form of parametric uncertainties (such as effective bulk modulus), and external disturbances

(such as the load force). Furthermore, the behaviour of the system is highly nonlinear mainly due to the stick-slip friction, and the orifice flows in the hydraulic circuit. Thus, the task of controlling the EHAs is challenging. This is further emphasized in the single-rod EHAs where their behaviour changes due to the asymmetry of the single-rod cylinder itself [3].

With regards to fault detection, the aforementioned faults and uncertainties could potentially change the performance of the EHA and drive the system to undesirable behaviour. Two noticeable faults are internal leakage and bulk modulus change. These faults are of importance since they could not be visibly seen and could cause accidents if not detected [4, 5].

1.3. Proposed Solutions

With regards to control, a fractional-order *Proportional Integral Derivative* (PID) controller is proposed as the method of controlling this system. These controllers have proven robustness characteristics similar to PID controllers which contribute towards dealing with uncertainties and nonlinearities in the EHA [6, 7]. This is because Fractional-order Proportional Integral Derivative (FOPID) controllers are generalized counterparts of the PID controllers where the exponent of the integral and derivator could be adjusted. This also means that while the characteristic behaviours of the PID controller are preserved, the FOPID control strategy provides the possibility of improved parameter tuning and performance compared to the classic method since it has five parameters that could be tuned [6]. These five parameters are then tuned using a modified Nelder-Mead algorithm to reach the optimal performance for the FOPID strategy.

With regards to fault detection, in order to pick up on the behavioural changes in system signals due to faults, fractal dimensions and wavelet transforms are used. From these measures, statistical moments are obtained as indicators of faults in a fault decision algorithm. This developed algorithm assigns a value to the degree of fault in the system if there is any.

By determining the occurrence of faults using the aforementioned algorithm, a fuzzy controller could operate as a higher-level control scheme that generates the control input based on a healthy condition FOPID controller and a faulty condition FOPID control. This design incorporates different aspects of this work which come together to develop a fault-tolerant fuzzy-FOPID controller.

1.4. Thesis Formulation

This thesis aims to develop a fault-tolerant robust controller for a single-rod EHA with internal leakage. Two low-level FOPID controllers are designed for faulty and healthy conditions which are then controlled by a high-level fuzzy inference system. Based on the degree of fault variable from the fault detection algorithm this fuzzy inference system produces a weighted expected value of these two inputs. This is achieved through developing fault detection algorithms using fractal dimension and wavelet detail coefficient indicators.

During the course of this thesis the following tasks will be performed:

1. investigate the proper way of acquiring accurate signals from raw data.
2. investigate the behaviour of the system using the acquired data from the EHA.
3. develop multi-scale and polyscale measures for detecting behavioural changes in the EHA signals and use them in a practical way to detect faults.
4. design an optimal FOPID controller using the Nelder-Mead algorithm for faulty and healthy conditions.
5. design an overall fault-tolerant scheme using the developed FOPID controller and fault detection method.

1.5. Methodology

Polyscale and multiscale measures have shown promise in detecting faults in the EHA system [5]. In this work variance fractal dimension, length fractal dimension and wavelet transform techniques are used. Also, the FOPID control is robust enough to be used on an EHA system while having enough design flexibility compared to the classical controllers.

In this thesis, to acquire accurate information about the system a new fuzzy denoising method has been developed. This denoising method provides a suitable alternative to the conventional low-pass filters. Also, a low-cost standalone Raspberry Pi-based data acquisition system has been developed for industry-friendly and commercial use.

With respect to fault detection, for the first time for a single rod EHA, internal leakage and bulk modulus change faults have been studied in a multi-fault environment and with different severities. Based on this analysis then indicators of the aforementioned three methods are incorporated in a single degree of fault variable by the means of a fault decision algorithm for internal leakage.

With regards to velocity control design, for the first time for a single rod EHA, two FOPID controllers have been designed for faulty and healthy conditions. These FOPID controllers are then tuned using the modified Nelder-Mead algorithm. Finally, using the fault

detection algorithm and the developed controllers, a fault-tolerant fuzzy-FOPID strategy has been implemented and tested in various internal leakages and conditions.

1.6. Thesis Organization

Chapter 2 describes the EHA system, its subsystems and their mathematical models which are then used in the rest of this thesis. Chapter 3 describes the design of the data acquisition system and the method of acquiring accurate real-time data. Chapter 4 includes the fault detection analysis, for actuator internal leakage and effective bulk modulus change, and the development of the fault decision algorithm. In Chapter 5, two FOPID controllers are designed for faulty and healthy conditions which are then used to develop a fault-tolerant control by employing a fuzzy inference system. Finally, Chapter 6 provides the contributions made in this thesis.

CHAPTER 2:

EHA SYSTEM AND ITS MATHEMATICAL MODEL

As shown in Fig. 2.1, the Electrohydrostatic Actuator (EHA) system has three subsystems. A *Data Acquisition* (DAQ) and control subsystem which based on user and sensor data, decides the control input to the hydraulic circuit actuators: a directional solenoid valve and a motor that drives the pump. The second subsystem is the hydraulic circuit which provides power for the actuator and consequently moves the third subsystem, the excavator arm.

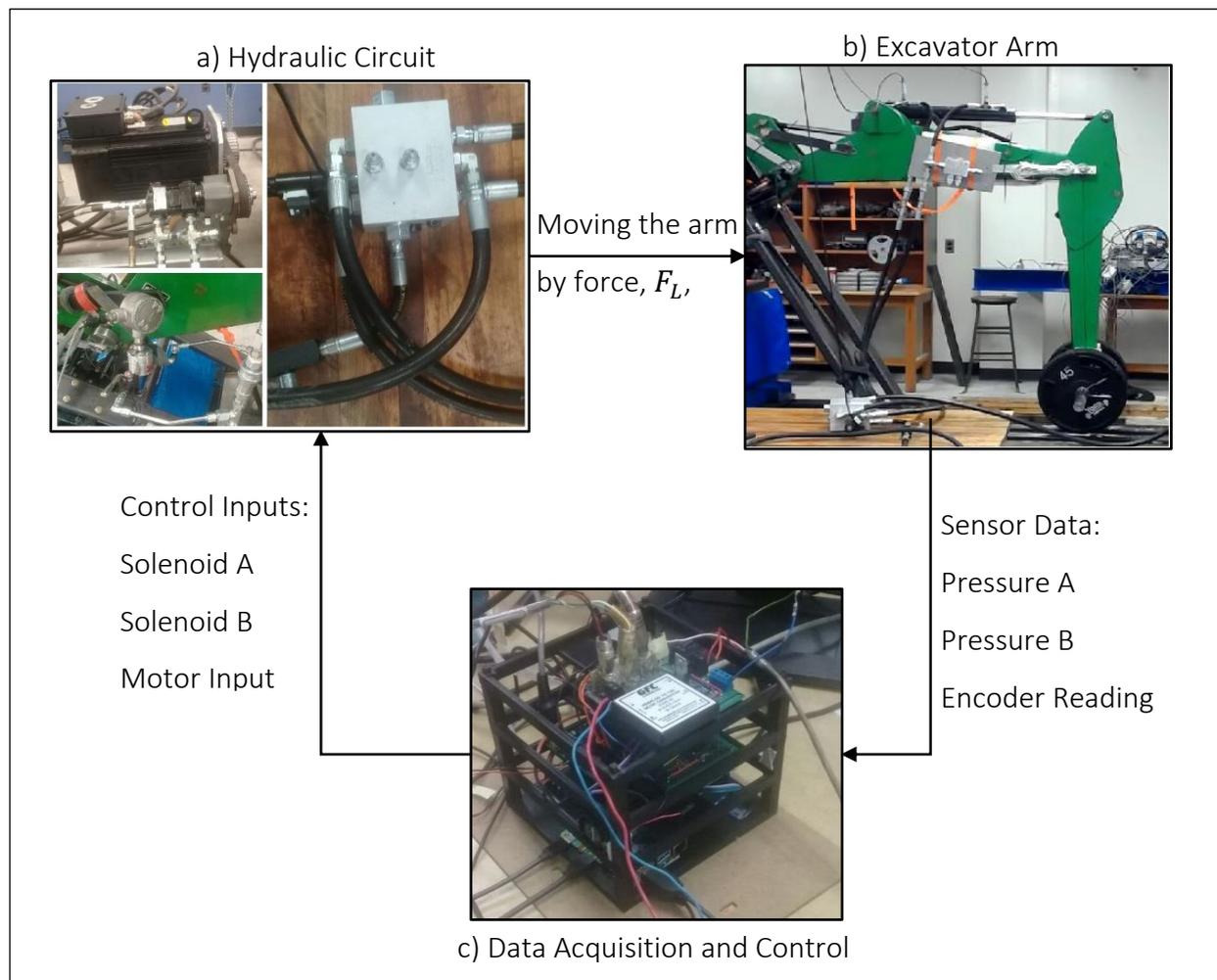


Fig. 2.1 Experimental setup (single-rod EHA) subsystems: a) Hydraulic circuit, b) Excavator arm, and c) data acquisition and control.

In this chapter, the designs of the excavator arm and the hydraulic circuit subsystems are thoroughly discussed. The DAQ subsystem is also briefly described, however, a comprehensive analysis of its design is left for Chapter 3. Furthermore, the mathematical model of the system is further investigated to reach a more depiction of the real system. Finally, using the basic principles of hysteresis a modification to the directional valve control is proposed to improve the performance of the system, both in open-loop and closed-loop scenarios.

The EHA system's operation is divided into five modes as shown in Fig. 2.2.

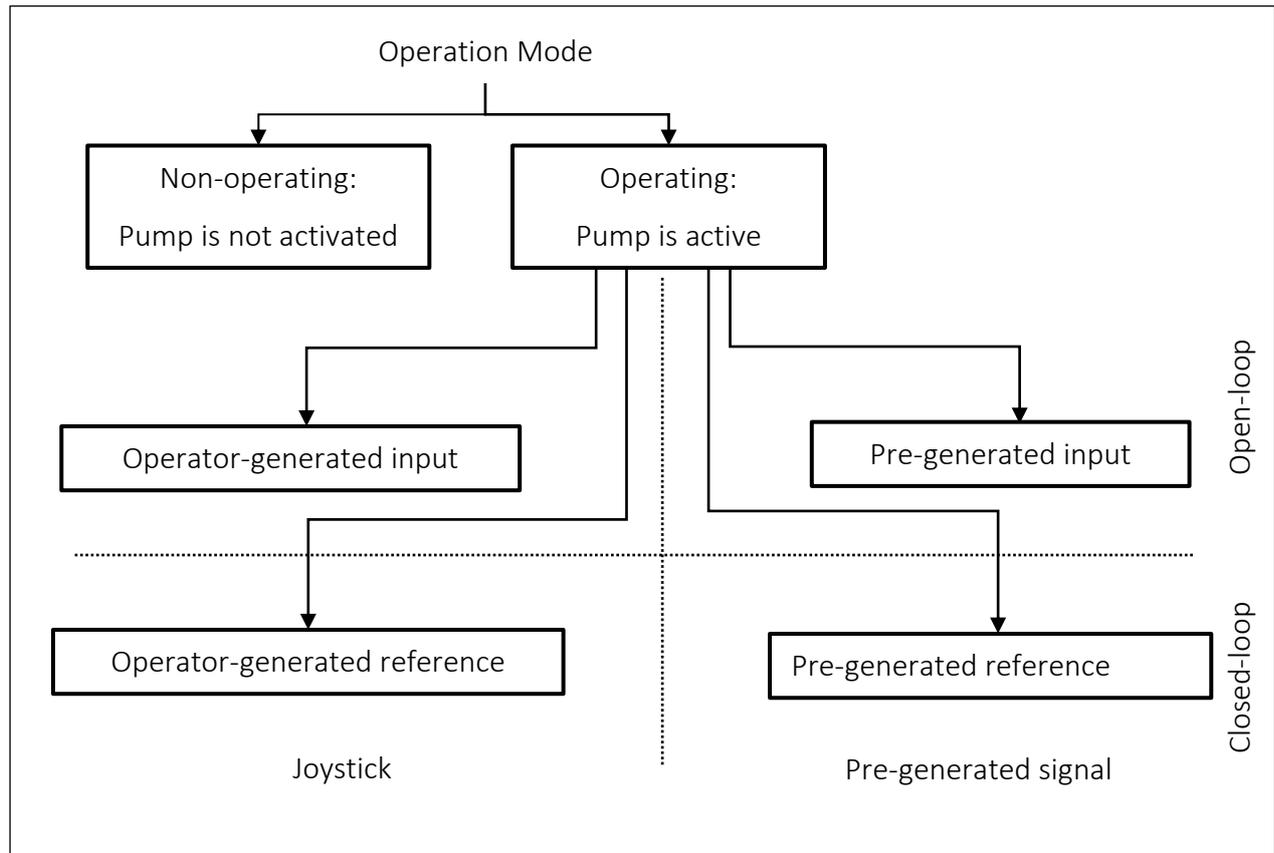


Fig. 2.2 Experimental setup (single-rod EHA) subsystems.

First, the non-operating mode is where the input to the system is zero.

Second, the operating mode incorporates all operations with a non-zero input to the system. There are further subtrees for operating mode with respect to open-loop or closed-loop configuration. Open- and closed-loop forms with different input formats have been referred to with specific terminology as follows:

1. operator-generated input mode refers to the system that is running in open-loop form with a joystick input directly to the motor.
2. operator-generated reference mode, refers to the system that is running in closed-loop form, with a joystick input as the velocity reference to the controller.

3. pre-generated input mode refers to the system that is running in open-loop form with a predesigned signal as input directly to the motor.
4. pre-generated reference mode refers to the system that is running in closed-loop form with a predesigned signal as the velocity reference to the controller.

2.1. Excavator Arm

The experimental setup of the single-rod EHA system is based on a John Deere JD-48 backhoe and is shown in Fig. 2.3. The rod position is measured by an incremental encoder. Load masses are adjusted by mounting disks on the arm and the pressure signals are acquired by two pressure sensors, each located on one side of the cylinder. As shown in Fig. 2.3, the excavator arm linkage carries the load, $m_L g$, by the actuator force, F_L , generated by the perpendicular movement of the cylinder piston. This force can be obtained from the net torque equation for the excavator arm,

$$F_L l_1 \cos\theta - m_L g l_2 \sin\theta = I \alpha_r \quad (1)$$

Where I is the moment of inertia, α_r is the angular acceleration, θ is the rotational angle, l_1 and l_2 are top and bottom lengths of the arm respectively, and g is the gravitational acceleration. Notice that by considering the cylinder as horizontal, the rotational angle θ can be translated as the piston displacement, x_p , since they are geometrically related¹.

Angular acceleration, α_r , is obtained by,

$$\alpha_r = \frac{\ddot{x}_p \cos(\theta)}{l_1} \quad (2)$$

where $\ddot{x}_p \cos(\theta)$ is the tangent acceleration of the excavator arm movement at point O .

Thus, using (2), (1) can be rewritten as follows:

$$F_L l_1 \cos\theta - m_L g l_2 \sin\theta = (\ddot{x}_p I \cos\theta) / l_1 \quad (3)$$

where \ddot{x}_p is actuator acceleration.

It should be noted that the load force, F_L , may vary as the cylinder moves, being either resistant or assistant.

¹ While in this thesis the cylinder is considered horizontal, the equations describing the relation between x_p and θ are investigated in Appendix A.

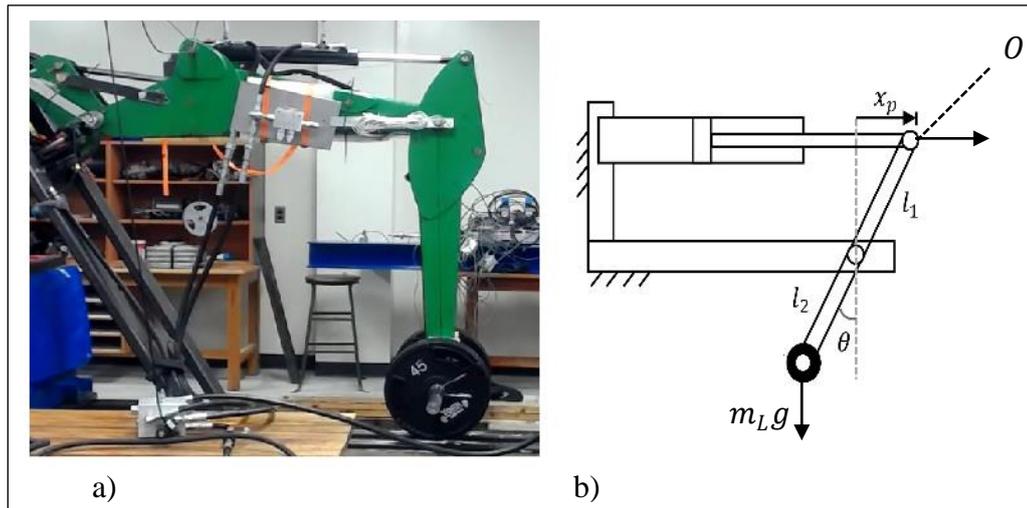


Fig. 2.3 Experimental setup: a) photograph of the experimental test bench, and b) schematic of the excavator arm.

2.2. Hydraulic Circuit

Fig. 2.4 shows the hydraulic circuit which is running the excavator arm. Elements of the EHA circuit in Fig. 2.4 are described in Table 2.1.

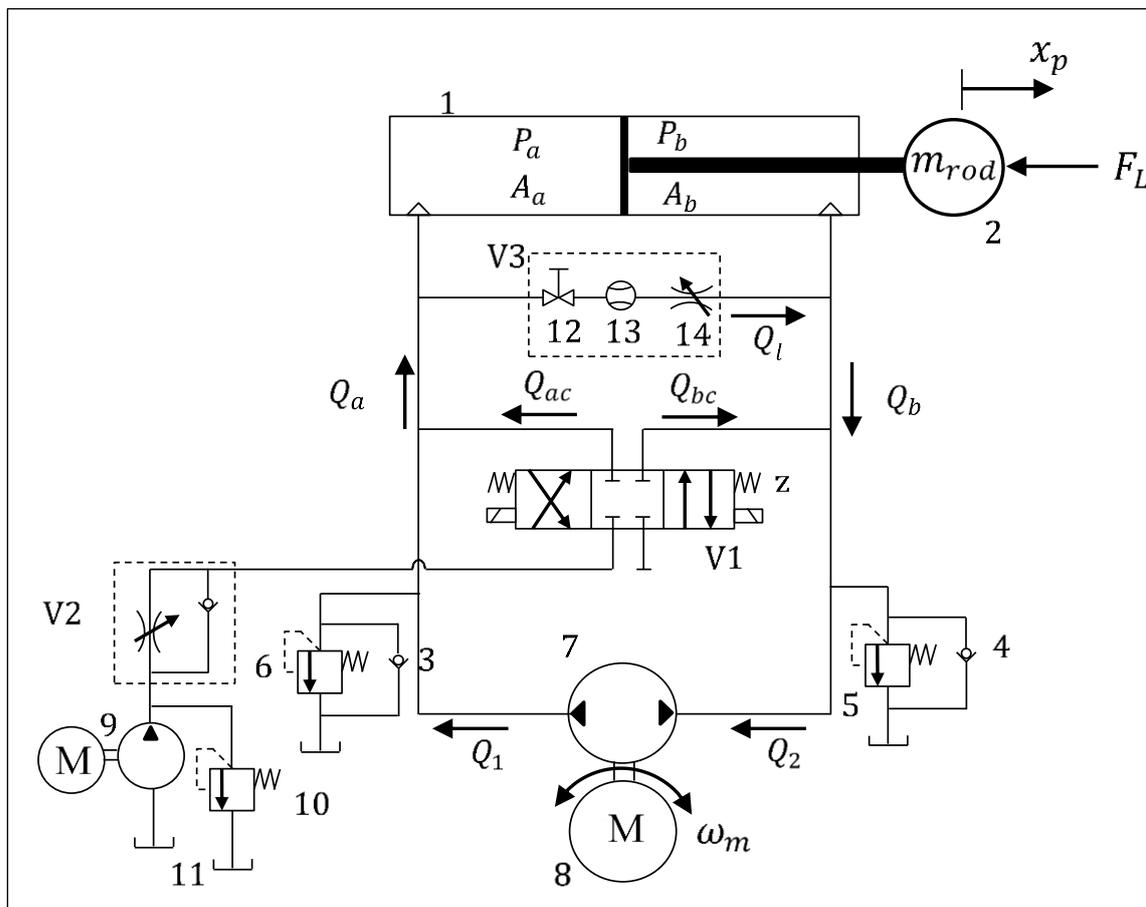


Fig. 2.4 Schematic of the electrohydraulic actuator circuit. (After [2])

Table 2.1 Elements of the electrohydrostatic actuator circuit shown in Fig. 2.4.

Item	Circuit	Description
1	Main	Actuator
2	Main	Mass of piston and rod
3	Main	Check valve
4	Main	Check valve
5	Main	Relief valve
6	Main	Relief valve
7	Main	Bidirectional pump
8	Main	Servomotor
9	Auxiliary	Auxiliary pump
10	Auxiliary	Relief valve
11	Auxiliary	Tank
V1	Main	Three-position four-way directional valve
V2	Auxiliary	One-directional flow control valve
V3	Main	Artificial leakage
12	Main	Manual ball valve
13	Main	Flowmeter
14	Main	Orifice

The bidirectional pump is driven by a servomotor. The auxiliary pump is connected to a relief valve and the one-directional flow control valve. Using a three-position four-way directional valve, the auxiliary circuit is used to compensate for the uneven cylinder flows to the single-rod cylinder. Two check valves have been used in the circuit to ensure the minimum flow and two relief valves have been used to set the maximum operating pressures in the actuator chambers. The outlet pressure of the auxiliary pump is 80 psi which is adjusted by a relief valve.

The area ratio of actuator cap and rod sides is,

$$\alpha_{ab} = A_b/A_a \quad (4)$$

where A_a is cap side area, A_b is the rod side area.

The load pressure is defined as,

$$p_L = p_a - \alpha_{ab}p_b \quad (5)$$

where p_a and p_b are the pressures in the cap side chamber, a , and rod side chamber, b , respectively.

The four quadrants of the operation are shown in Fig. 2.5.

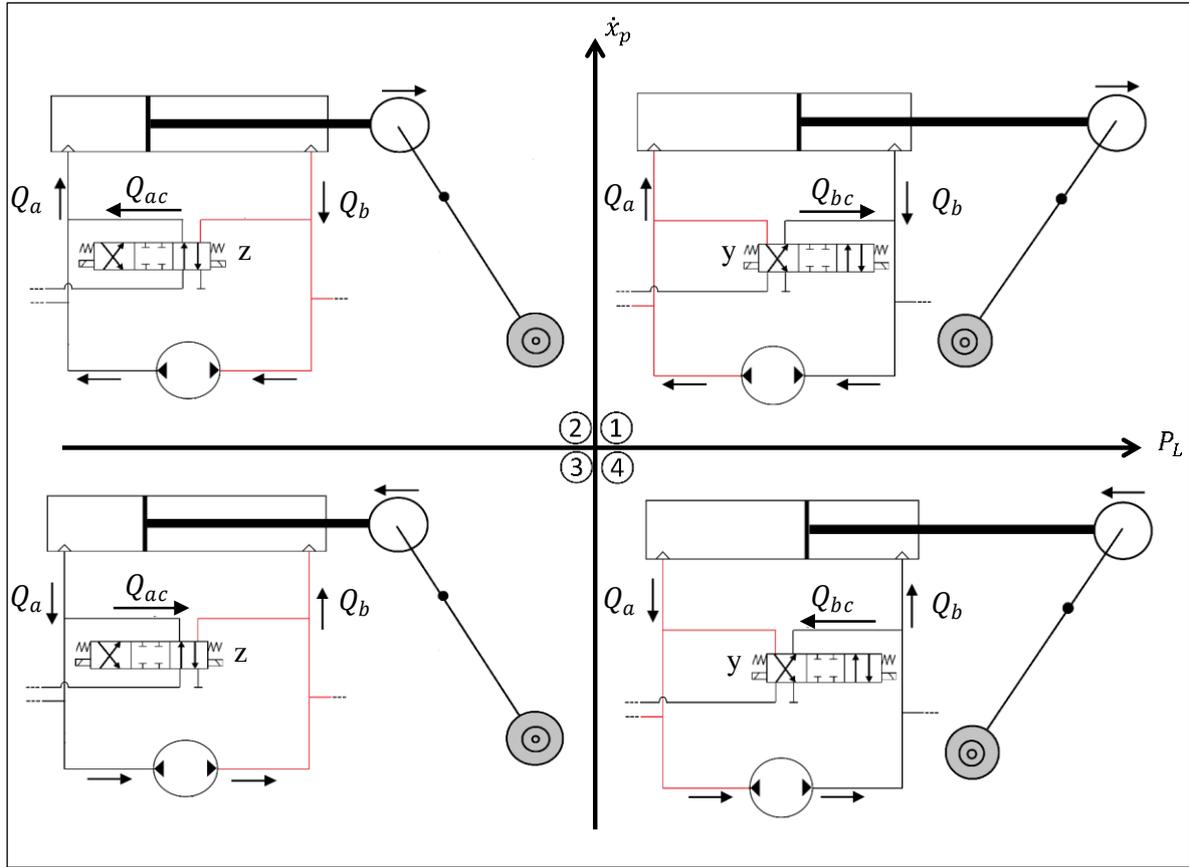


Fig. 2.5 Four quadrants of the electrohydrostatic actuator system.

As shown in Fig. 2.5, and based on the load pressure, p_L , the directional valve has two states: (a) when $p_L > 0$, the solenoid y of the directional is activated and the solenoid z is deactivated (quadrants 1 and 4). (b) when $p_L < 0$, the solenoid z is activated and the solenoid y is deactivated (quadrants 2 and 3).

The flow of the bidirectional pump can be described as,

$$Q_1 = Q_2 = \omega_m V_d \quad (6)$$

where Q_1 is the flow into the pump, Q_2 is the flow out of the pump, ω_m is the motor speed and V_d is the pump displacement.

The motor speed equation is,

$$\dot{\omega}_m = \tau_m (-\omega_m + K_m u) \quad (7)$$

where, K_m is servomotor gain, τ_m is servomotor time constant and u represents the input voltage to the servomotor and consequently the system.

By applying the conservation of mass to sides a and b of the actuator, (8) and (9) are obtained.

$$Q_l + Q_1 + Q_{ac} = Q_a = A_a \dot{x}_p + \frac{V_{oa} + A_a x_p}{\beta_e} \dot{p}_a \quad (8)$$

$$\begin{aligned} -Q_l + Q_{bc} - Q_2 &= -Q_b \\ &= -A_b \dot{x}_p + \frac{V_{ob} - A_b x_p}{\beta_e} \dot{p}_b \end{aligned} \quad (9)$$

where Q_l is the leakage flow from cap side to rod side, Q_{ac} and Q_{bc} are the compensation flows coming from the charge pump through the directional and flow-control valves to the actuator sides, a and b , respectively (see Fig. 2.4). Q_a is the flows into the actuator and Q_b is the flows into out of the actuator. x_p is the displacement of the piston rod, \dot{x}_p is the velocity of the piston, β_e is the effective bulk modulus and V_{oa} and V_{ob} each represent the inner pipe and cylinder chamber volumes at the associate sides of the actuator. Newton's second law for the actuator piston gives,

$$m_{rod} \ddot{x}_p = A_a p_L - f \dot{x}_p - F_L \quad (10)$$

where m_{rod} represents the piston rod mass and f is the viscous damping coefficient.

When the actuator is extending as in Quadrant 1 ($p_L > 0$ and $Q_{ac}=0$) and Quadrant 2 ($p_L < 0$ and $Q_{bc}=0$), the charge pump alternately supplies an oil flow to the cap and rod sides of the actuator through the flow-control valve, V1. Thus, the relation between the two flows is as follows:

$$Q_b = \alpha_{ab} Q_a \quad (11)$$

When the actuator is retracting (Quadrants 3 and 4), the oil flows from either side of the actuator to the tank through valves V1, V2 and the relief valve. In Quadrant 4, $p_L > 0$ and $Q_{ac}=0$. The flow through the orifice at the flow-control valve is,

$$Q_{bc} = -C_V \sqrt{\frac{(p_b - p_0)}{sg}} \quad (12)$$

where C_V is valve flow coefficient. sg is the specific gravity of the fluid and p_0 is the cracking pressure of the relief valve. In Quadrant 3, $p_L < 0$ and $Q_{bc}=0$, and Q_{ac} can also be obtained as,

$$Q_{ac} = -C_V \sqrt{\frac{(p_a - p_0)}{sg}} \quad (13)$$

Leakage flow, Q_l is obtained from (14) [4, 5] as follows:

$$Q_l = -K_l \sqrt{(p_a - p_b)} \text{sign}(p_a - p_b) \quad (14)$$

where K_l is the leakage turbulence coefficient [5]. It should be noted that the leakage flow introduced in the test rig is not part of the final design and is merely devised for fault detection analysis.

Equations (3) to (14) describe the behaviour of the system. The parameters of the test rig [3] are listed in Table 2.2. These parameters are used in the simulations using the Simulink model of the system.

Table 2.2 Parameters of the experimental setup [3].

Parameter	Symbol	Value	
		Nominal	Range
Displacement of the bidirectional pump (m ³ /rev)	V_d	8×10^{-6}	-
Time constant of the servo motor (1/s)	τ_m	3	2.3 - 4.0
Servomotor gain (rev/(s·V))	K_m	5.8	5.6 - 6.0
Cap-side area (m ²)	A_a	3167×10^{-6}	-
Area ratio of the actuator	α_{ab}	0.75	-
Effective bulk modulus (Pa)	β_e	689×10^6	$(151 - 1670) \times 10^6$
Hydraulic compliance (m ³ /Pa)	C	3.46×10^{-12}	$(2.20 - 7.03) \times 10^{-12}$
Piston and rod mass (kg)	m_{rod}	10	9 - 11
Viscous damping coefficient(N·s/m)	f	900	600 - 1200
Load mass (kg)	m_L	-	0 - 367
Length of effort arm (m)	l_1	0.3	-
Length of the load arm (m)	l_2	1.2	-
internal leakage coefficient (m ³ /√Pa·s)	K_l	-	9×10^{-7}

2.3. Data Acquisition and Control

The box diagram of the DAQ and control subsystem is shown in Fig. 2.6. This subsystem includes inputs from a joystick, two pressure sensors and an encoder from the excavator arm. It also provides the outputs in the system which include two digital signals to the two solenoids and an analog signal to the motor driver of the pump of the EHA circuit. The design of this subsystem is discussed in Chapter 3.



Fig. 2.6 Data acquisition and control subsystem inputs and outputs.

2.4. Model Verification

The developed mathematical model of the EHA is used now to describe the performance of the actual system. To assure the validity of the derivations, this model is

compared to the experimental setup. Fig. 2.7 shows the results of the comparative simulation for the mathematical model and the experimental setup of Fig. 2.1 in pre-generated input mode. Additional results corresponding to the simulation are shown in Fig. 2.8 shows.

Fig. 2.9 shows the results of the comparative simulation for the mathematical model and the experimental setup of Fig. 2.1 in operator-generated input mode. Additional results corresponding to this simulation are shown in Fig. 2.10.

As shown in the comparative results, given the same input to both systems the outputs (velocity) exhibit the same behaviour. The *Root Mean Square Error*¹ (RMSE) of the mathematic model output (velocity) with respect to the experimental setup is the small value of 0.0144. Also, the pressure signals of the model are accurately estimated by simulation and have the same trends as shown in Fig. 2.7 and Fig. 2.9. Thus, the mathematical model is used as a representation of the actual system in the simulations.

2.5.A Modification to The Control Scheme of The Directional Valve

As seen in Fig. 2.7 and Fig. 2.9, during the operation of the system at the point of quadrant change between quadrants 2,3 and 1,4 due to the switching behaviour of the directional valve, actuated by solenoids, a spike happens to the outputs of the system. While it is less apparent in the position trajectory, it is visually visible in the velocity trajectory.

While the existence of this spike is inevitable in this switching system [2], it can be reduced. By design, ideally, the switching happens at the point where the system exhibits equilibrium in the cylinder forces, in other words $p_L = 0$. At this point due to the equilibrium in the forces of the actuator, the grip of the control over the system is loose. In other words, a sudden change in the flow in either direction does not face any damping or resistance which leads to a fast transient response.

This sensitivity at the switching point could be managed by either introducing more stiffness where $p_L = 0$, or by postponing the switching point (and in turn the transient behaviour) to a stiffer state during the operation of the machine. This is achieved by introducing the hysteresis to the system. The working mechanics of the hysteresis is shown in Fig. 2.11.

¹ For N samples, root mean square error is $RMSE = \sqrt{\frac{\sum_{i=1}^N error_i^2}{N}}$.

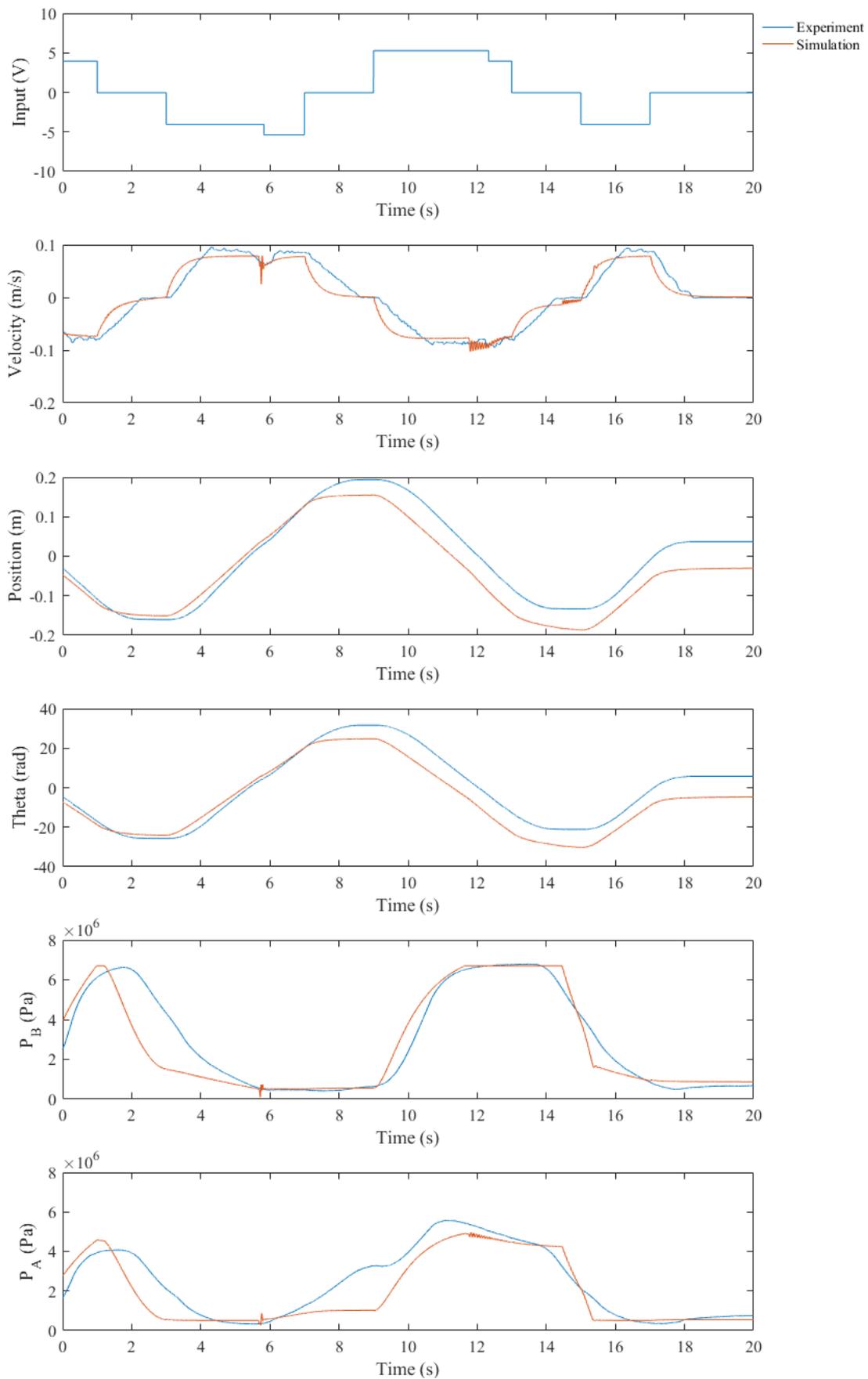


Fig. 2.7 Simulation vs experimental results in pre-generated input mode.

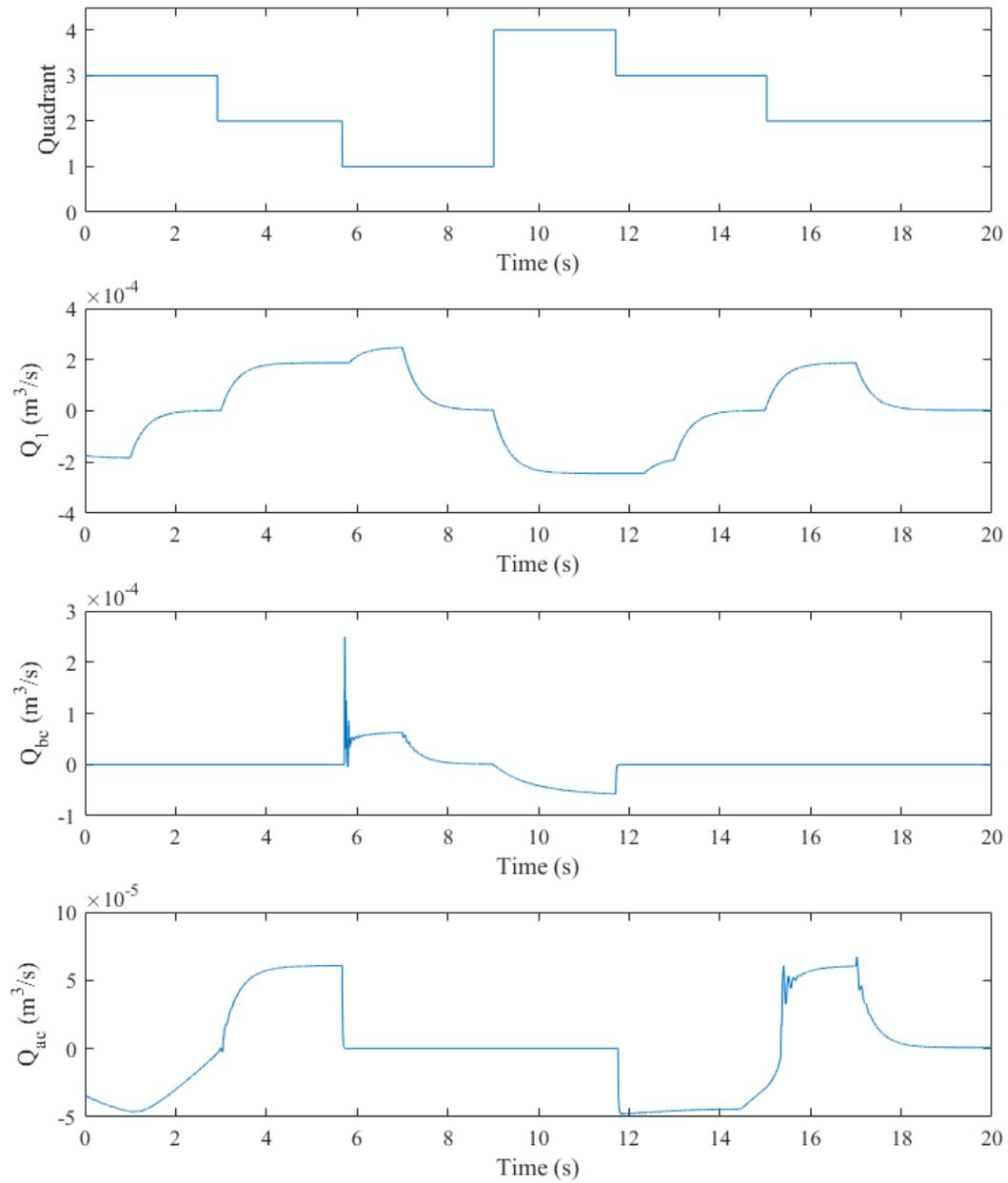


Fig. 2.8 Additional results for the simulation shown in Fig. 2.7.

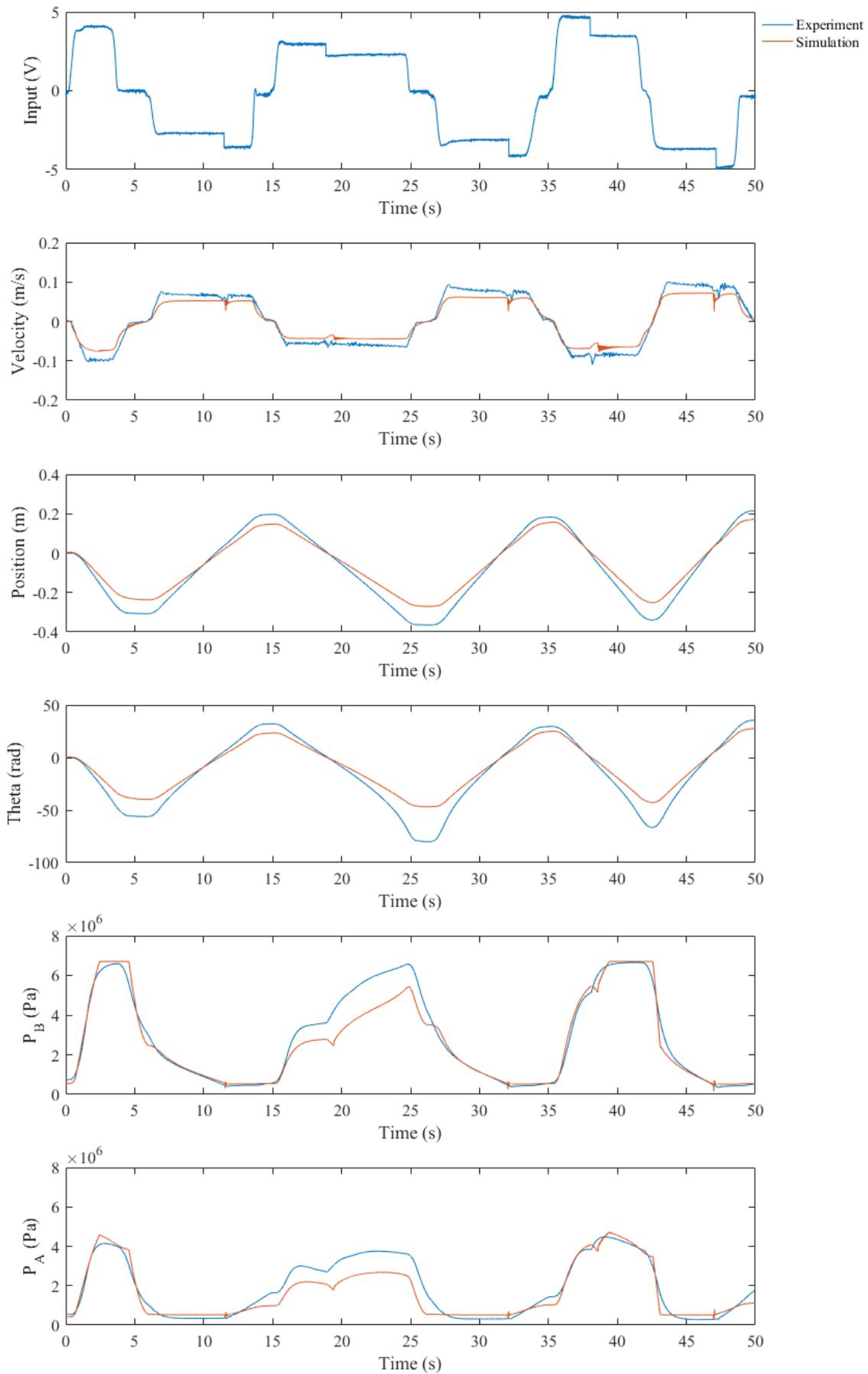


Fig. 2.9 Simulation vs experimental results in operator-generated input mode.

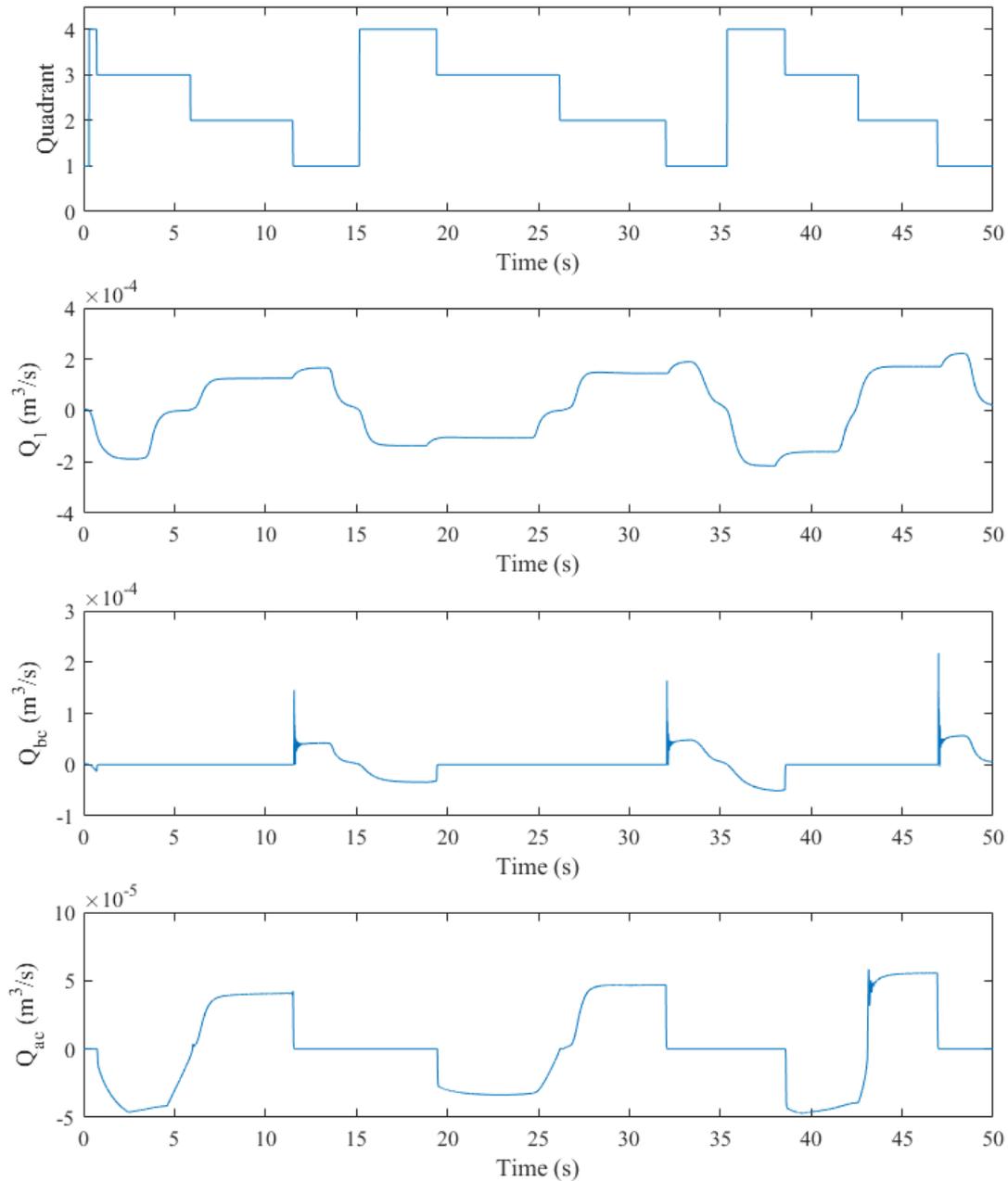


Fig. 2.10 Additional results for the simulation shown in Fig. 2.9.

As shown in Fig. 2.11, the hysteresis is a way of endowing the normal switching behaviour with directional memory to postpone the switching point [8]. While the normal solenoid would be activated at the point where p_L changes sign, the hysteresis postpones the switching until p_L has reached $\pm\Omega$. This allows for the resistive force in the cylinder to build up which in turn makes the system stiff enough to damp the transient behaviour around the switching point.

The effectiveness of the hysteresis is demonstrated in Fig. 2.12 by three simple open-loop experiments comparing the normal switching behaviour with no hysteresis ($\Omega_1 =$

0.0MPa), the switching with small hysteresis ($\Omega_1 = 0.2\text{MPa}$) and the switching big hysteresis with $\Omega_2 = 0.3\text{MPa}$. The + signs in this Fig. represent the solenoid switching times for the corresponding systems.

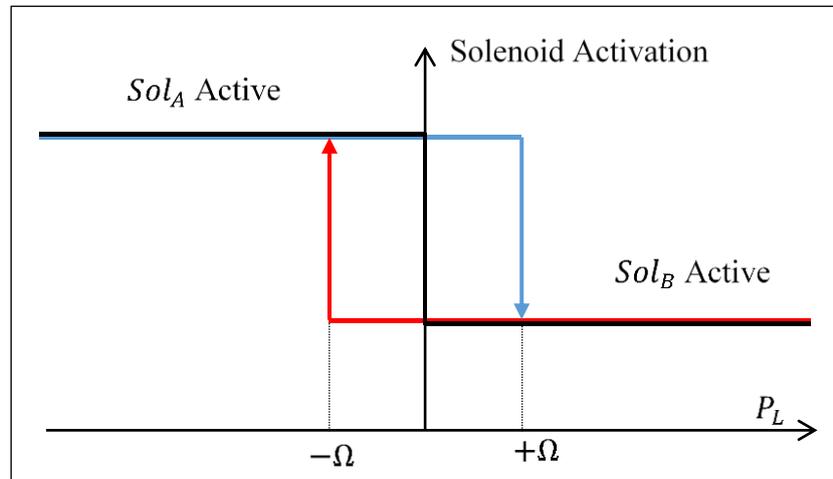


Fig. 2.11 Hysteresis applied to switching.

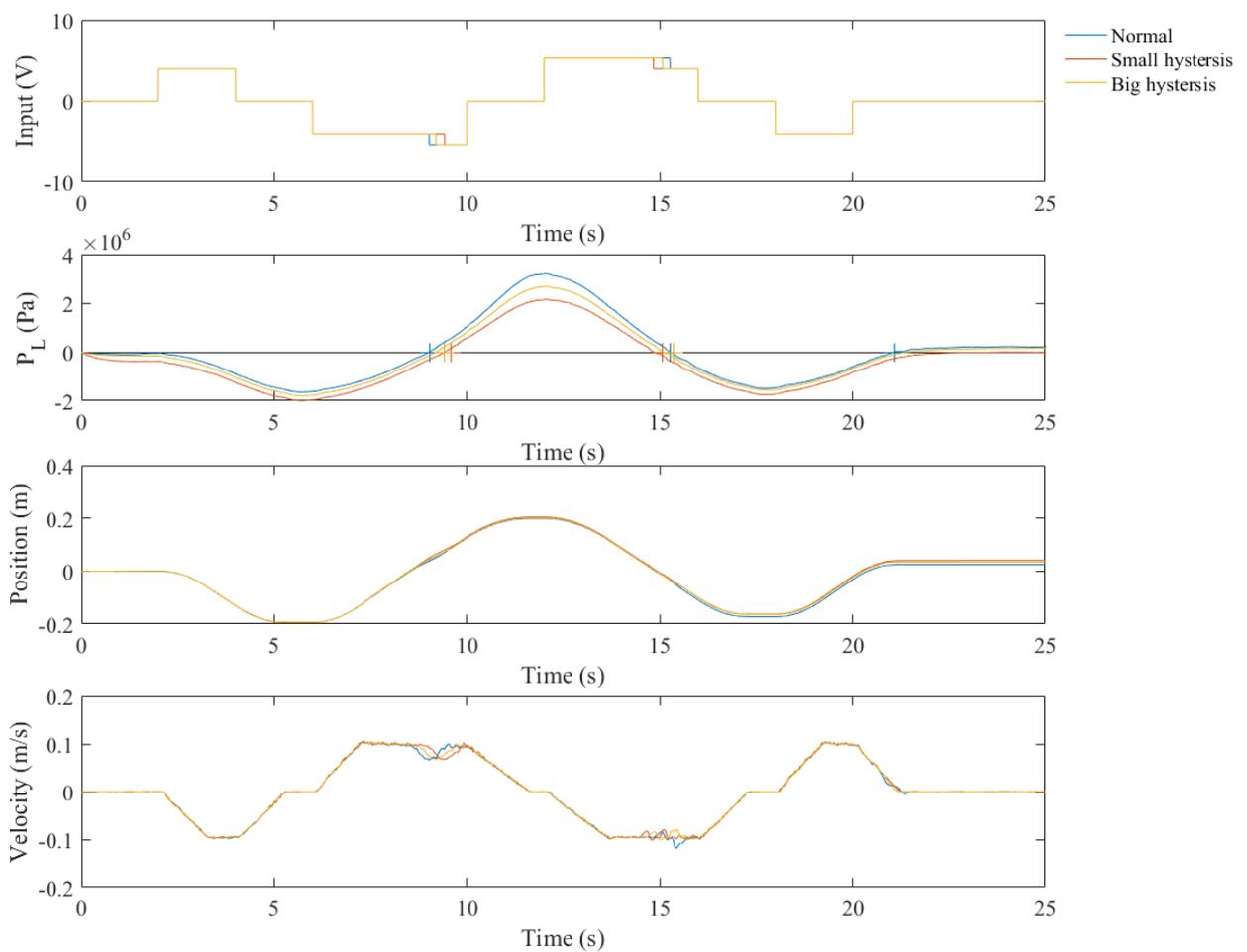


Fig. 2.12 Effect of hysteresis on the velocity, experimental results in pre-generated input mode.

As shown in Fig. 2.12, the introduction of hysteresis to the solenoid activation scheme lowers the amplitude of the spike while maintaining the same behaviour in the system as the normal switching. Furthermore, while the hysteresis is effective in attenuating the spike, it only partially reduces the spike and cannot eliminate it. This is because the switching behaviour exists inherently as a result of the operation of the directional valve. This is further seen in the big hysteresis results where although the threshold has increased the spike reducing performance is effectively the same as the small hysteresis.

The spike is defined as the maximum deviation of the velocity trajectory from its initial value at the time of switching. The spikes of the three systems at $t=15s$ are described in Table 2.3. The results show that the small hysteresis switching with $\Omega_1 = 0.2MPa$ has a lower spike in the velocity trajectory by $0.0056m/s$ (25.11%) and the big hysteresis switching with $\Omega_2 = 0.3MPa$ has a lower spike (but approximately the same as small hysteresis) in the velocity trajectory by $0.0054m/s$ (24.22%). This shows that, first, hysteresis is significantly effective in dealing with the switching behaviour in the system and, second, increasing the hysteresis does not guarantee further spike reduction. It should Also be noted that this reduction opens at quadrants 3 and 4, not 1 and 2.

Table 2.3 Spikes with respect to hysteresis as shown in Fig. 2.12 at $t=15s$.

No.	Ω	spike
1	$\Omega_0 = 0.0MPa$ (No hysteresis)	0.0223m/s
2	$\Omega_1 = 0.2MPa$	0.0167m/s
3	$\Omega_2 = 0.3MPa$	0.0169m/s

2.6. Summary

In this section, first, the three subsystems of the experimental test rig are described and the mathematical model for the EHA is thoroughly investigated to best represent the actual setup. Also, the design of the internal artificial leakage with respect to the hydraulic circuit is presented for fault detection purposes. Finally, the control algorithm for the solenoid activation is further modified to mitigate the spike in the system.

CHAPTER 3:

ACQUISITION AND GENERATION OF ACCURATE SIGNALS

Electrohydrostatic Actuator (EHA) systems require complicated control and data acquisition modules. They have to include different capabilities such as dealing with noise, and they are required to perform *Analog to Digital Conversion* (ADC) and *Digital to Analog Conversion* (DAC).

In this chapter, first, a novel denoising method based on fuzzy thresholding is proposed and tested in various scenarios to further enhance the process of obtaining accurate signals. Second, the methods and mechanisms behind the acquisition and generation of accurate signals in the experimental setup are described by investigating the full data acquisition and control process for the EHA. Also, a Raspberry Pi-based data acquisition and control system is developed and described in detail. The Raspberry Pi-based design is compared with a commercially available Data Acquisition (DAQ) Quanser Q8 [9].

3.1. Signal Denoising

In a mechatronics system, various underlying processes could add noise to the sensor signals [10]. Since the input signals from the plant are essential for monitoring and controlling the performance of the system, a noisy signal could cause catastrophic accidents. Adding a denoising element to the multiprocessing component of the data acquisition system will prevent such accidents caused by noisy signals. Furthermore, since denoising methods have a negligible effect on the actual signal, compared to the conventional low-pass filters which significantly affect the signal and attenuate it, utilizing them contributes to the conservation of confined information in each signal.

In this section, first, wavelet transform is described. Next, a novel fuzzy wavelet coefficient thresholding is proposed for denoising the signals. The proposed method is tested in various scenarios.

3.1.1. Wavelet Transform

A wavelet is a self-affine object with compact support and a mean value of zero [11]. Using *Wavelet Transform* (WT), the main goal is to obtain wavelet coefficients for different segments of data in different scales. In this sense, WT is a multiscale measure. To perform the wavelet transform, first, a mother wavelet is defined from a basic function. In a *Continuous Wavelet Transform* (CWT), the mother wavelet is then scaled and translated, corresponding to scale a and location b , as follows:

$$\psi(a, b) = \frac{1}{\sqrt{|a|}} \psi\left(\frac{t-b}{a}\right) \quad (15)$$

In (15), all wavelets are scaled and translated versions of the same mother wavelet. If the scaling and translation parameters are chosen as $a = 2^j$ and $b = k2^j$, then it constitutes an orthonormal basis called a dyadic orthonormal basis. Thus, (15) can be rewritten as follows:

$$\psi(j, k) = 2^{-\frac{j}{2}} \psi(2^{-j}t - k) \quad (16)$$

For a *Discrete Wavelet Transform* (DWT) of a discontinuous signal $x[n]$, however, DWT achieves a multiresolution decomposition of $x[n]$ [11], using the *Multiresolution Signal Decomposition* (MSD) algorithm shown in Fig. 3.1.

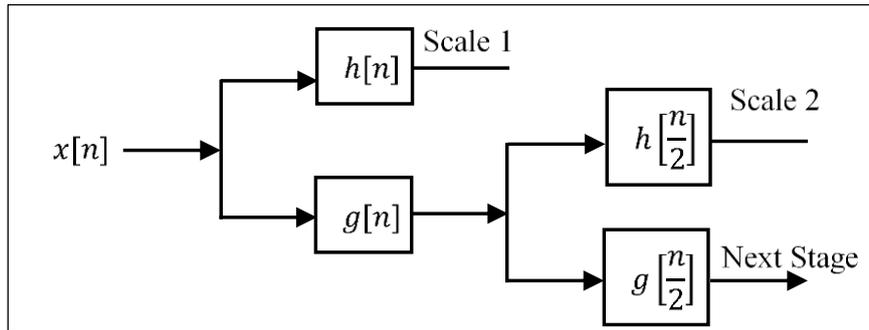


Fig. 3.1 Discrete wavelet transform algorithm.

Thus, using MSD the wavelet approximation and detail coefficients for the first scale, $j = 1$, can be obtained as follows [11].

$$a_1[n] = \sum_k h[k - 2n]x[k] \quad (17)$$

$$d_1[n] = \sum_k g[k - 2n]x[k] \quad (18)$$

where $h[n - 2^j k]$ and $g[n - 2^j k]$ are the associated low-pass filter and high-pass filter coefficients corresponding to the dyadic wavelets, at level, j . The wavelet used in this work is D oubachies 4 [11].

3.1.2. Denoising

For a signal, x_n , with additive white gaussian noise, ω_n , the measured signal, y_n , can be expressed as follows:

$$y_n = x_n + \omega_n, \quad (n = 0, 1, \dots, N - 1) \quad (19)$$

where N is the number of samples.

By applying the DWT decomposition with J levels to the measured signal of (19), it can be inferred that the resulting coefficients of DWT of y_n are a superposition of DWT of x_n , which would have a few coefficients with large values, and DWT of ω_n , which would have a large number of coefficients with small values. Thus, denoising comes down to thresholding the smaller coefficients and, reconstructing the signal using the thresholded coefficients. These three steps of: DWT (decomposition), thresholding, and inverse DWT (reconstruction) are the basis of wavelet denoising.

Significant research has been done on choosing the right threshold [12] and the mechanism of thresholding [13]. Regarding the right threshold value, [14] introduced a fixed threshold, λ_j^{fixed} , which has been widely used and is formulated as follows:

$$\lambda_j^{fixed} = \hat{\sigma} \sqrt{2 \ln \frac{N}{2^j}} \quad (20)$$

where $\hat{\sigma}^2$ is the estimated noise variance and $\hat{\sigma}$ is its standard deviation which can be obtained as follows using the level 1 detail coefficients, $d_1[n]$.

$$\hat{\sigma} = \frac{\text{median } |d_1[n]|}{0.6745} \quad (21)$$

While this is an effective thresholding for dealing with noise and achieves an acceptable *Signal to Noise Ratio* (SNR), it tends to be more conservative and zero out more coefficients at higher resolutions. As a solution, based on *Stein's Unbiased Risk Estimate* (SURE), another choice of the threshold was introduced [15]. In comparison to the fixed threshold, the SURE threshold leaves out fewer detail coefficients. However, this also means that the SURE method passed a higher number of noise coefficients through the thresholding process and those coefficients are thus included in the reconstruction step as well. Due to these behaviours, these two methods (fixed and SURE) are considered as the upper and lower limits for determining the threshold value [12].

Regarding the thresholding processes, both soft thresholding and hard thresholding methods are widely used [13]. The soft thresholding method is represented by (22), and (23) describes the hard thresholding method.

$$\bar{x} = \begin{cases} \text{sgn}(x)(|x| - \lambda) & |x| \geq \lambda \\ 0 & |x| < \lambda \end{cases} \quad (22)$$

$$\bar{x} = \begin{cases} x & |x| \geq \lambda \\ 0 & |x| < \lambda \end{cases} \quad (23)$$

where $\text{sgn}()$ is the sign function, λ is the chosen threshold, x is the input coefficient and \bar{x} is the processed coefficient.

The hard thresholding method achieves higher SNRs and does not change the local properties of the signal. On the other hand, due to its discontinuity, it results in fluctuations in the reconstruction. For this reason, the soft thresholding method is preferred for certain applications such as image denoising [16]. However, the soft thresholding process has lower SNRs [13] and it loses a part of the high-frequency coefficients above the threshold.

3.1.3. Proposed Thresholding Method

The hard thresholding and soft thresholding methods represent the two poles of the trade-off between achieving low Root Mean Square Error (RMSE) versus avoiding distortion. Also, to choose the right threshold value for these functions, one must deal with the trade-off between keeping all the information in the signal and properly removing as many noise coefficients as viable. These two trade-offs are the motivation behind the development of a novel thresholding algorithm, which while operating within these limits has acceptable performance.

Consider a typical thresholding plot, as shown in Fig. 3.2, where the green line represents a symbolic approximate of an ideal candidate for the thresholding function. To have the best position in the face of the two aforementioned trade-offs, two changes must take place; First, the ideal thresholding function in this thesis must allow for some of the coefficients which are smaller than the threshold but to a certain extend. Second, the threshold should reach and track the hard threshold line, while maintaining a smooth transition from the soft thresholding.

These criteria have led to the design of a novel thresholding method based on a simple three rule inference machine with three membership functions. Membership functions for the input (coefficients) to the *Fuzzy Inference System* (FIS) are demonstrated in Fig. 3.3. The membership functions which are two sigmoid and a gaussian function are formulated as follows:

$$\mu_0(x) = e^{-\frac{x^2}{2(a\lambda)^2}} \quad (24)$$

$$\mu^-(x) = \frac{1}{1 + e^{\beta/\lambda(x+\lambda)}} \quad (25)$$

$$\mu^+(x) = \frac{1}{1 + e^{-\beta/\lambda(x-\lambda)}} \quad (26)$$

where α and β are adjustment constants and λ is the fixed threshold.

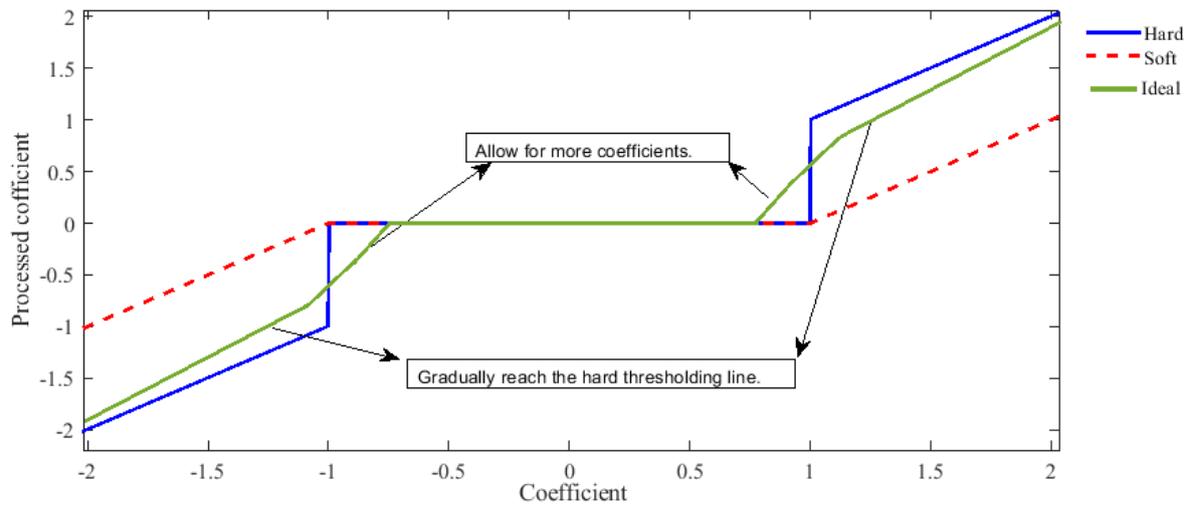


Fig. 3.2 The expected behaviours in the proposed thresholding method compared to conventional thresholding functions.

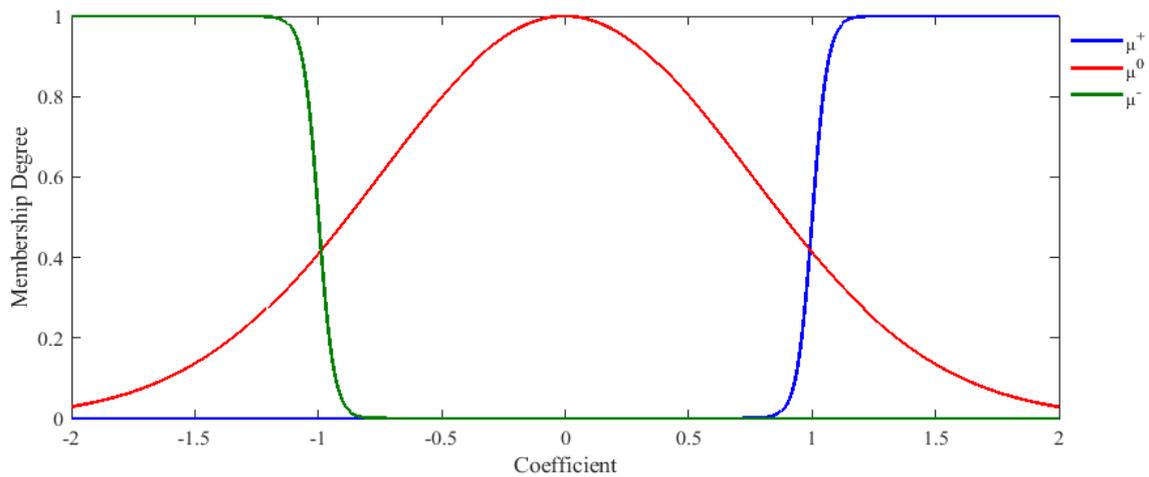


Fig. 3.3 Membership functions for the proposed fuzzy thresholding.

The three rules of the FIS in Takagi Sugeno [17] form are as follows:

1. If x is μ^- then $\bar{x}_1 = x$.
2. If x is μ^0 then $\bar{x}_2 = 0$.
3. If x is μ^+ then $\bar{x}_3 = x$.

Where \bar{x}_i is the fuzzy output of rule i .

Thus, the defuzzified output of the fuzzy inference system, which is the processed coefficient, \bar{x} , is obtained as follows:

$$\bar{x} = \frac{\mu^-(x)\bar{x}_1 + \mu^0(x)\bar{x}_2 + \mu^+(x)\bar{x}_3}{\mu^-(x) + \mu^0(x) + \mu^+(x)} \quad (27)$$

Fig. 3.4 demonstrates the proposed threshold compared to the classical methods. As shown, the expected behaviour is present in the proposed function; a number of smaller coefficients are considered to a smaller degree, and with a smooth transition, the proposed function reaches the proximity of the hard threshold function.

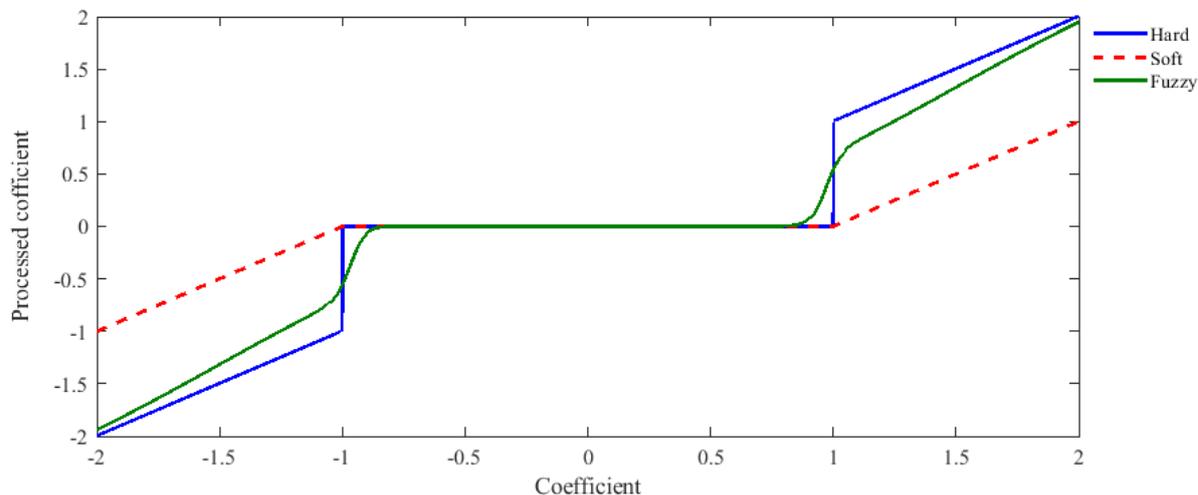


Fig. 3.4 Comparison between the proposed thresholding method and the conventional thresholding functions.

As shown in Fig. 3.5, α and β are adjustable parameters to allow more configurations for the designed algorithm. By increasing the value of α the settling time of the proposed threshold in case of it reaching the hard threshold line increases (i.e., there is an offset for a larger portion of the support). In other words, the performance becomes closer to the soft thresholding method.

β , on the other hand, affects the transient behaviour of the proposed fuzzy threshold. Decreasing the b value of β , increases the rise time of the function for reaching the threshold point (vertical line). This also comes hand in hand with an increase in the number of small coefficients that now have been passed through the thresholding algorithm with a smaller degree of effectivity.

In this work, the selected default values for α and β are 0.75 and 32, respectively, unless stated otherwise. These default values have produced the most desirable results in this work based on the experimental data available to us during this research.

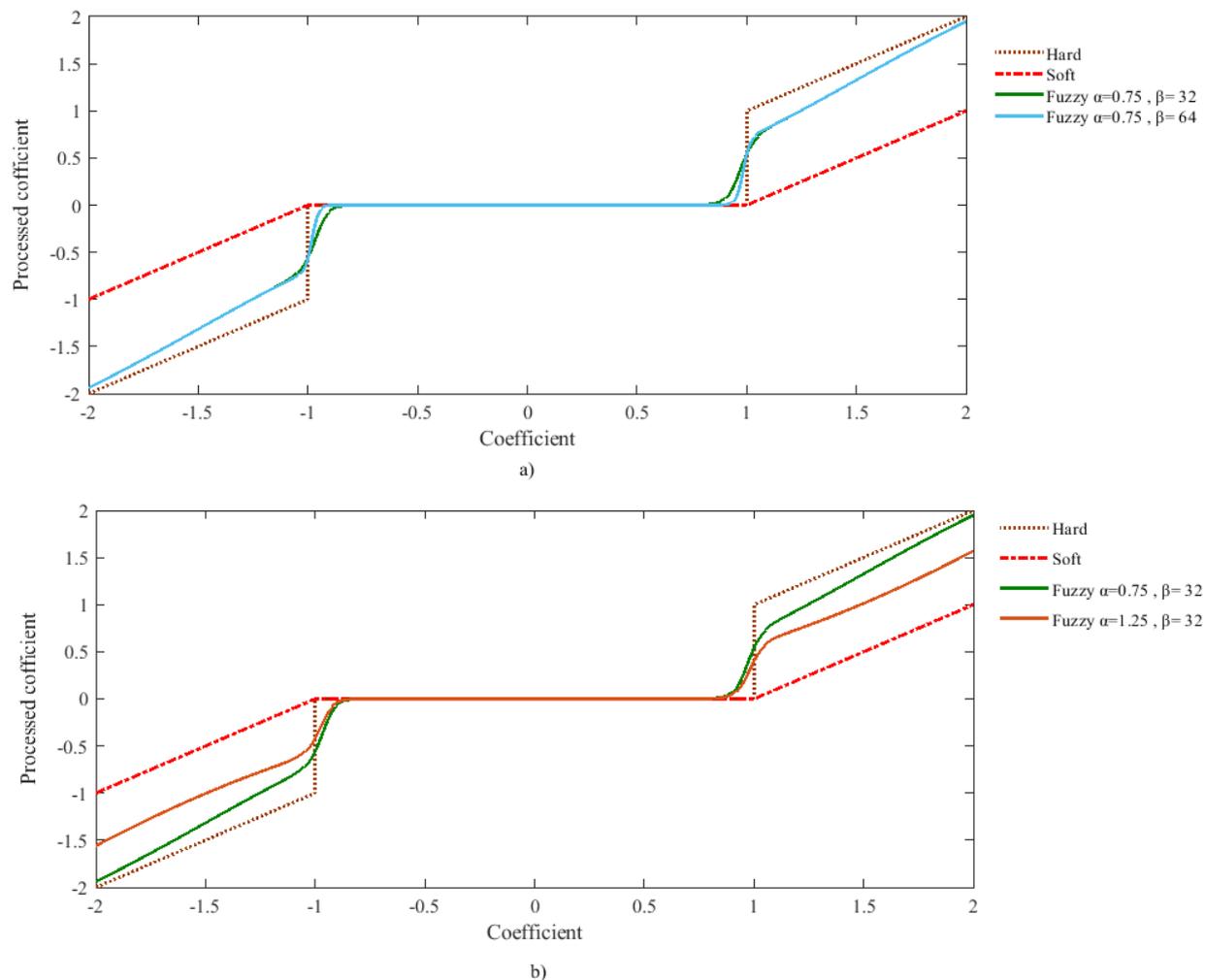


Fig. 3.5 Effect of a) α , and b) β , on the denoising behaviour of the proposed method.

3.2. Performance of The Proposed Fuzzy Denoising Method

In this section, the results of the proposed denoising algorithm are presented and compared with the classical methods.

Case 1. For the first case study, the errors of different methods are compared with respect to different original SNRs. The following signal with additive white gaussian noise with different SNR values has been considered for this simulation.

$$S(x) = e^{\left(\frac{7(x-250)}{100000}\right)} \cos\left(\frac{\pi(x-250)}{6}\right) \quad (28)$$

The results of these simulations for 65536 samples are shown in Fig. 3.6. This figure shows the RMSE value, with respect to different original SNRs. A zoomed version of the same plot is also shown in this Fig. for a clear comparison of the proposed fuzzy method (with different values of α) with the hard thresholding method.

As shown in Fig. 3.6, with respect to the reconstruction of the signal all three methods have close RMSEs in higher SNRs. However, in low SNRs, the proposed method's behaviour is close to the hard thresholding method for different values of α , while the soft thresholding method drifts to higher RMSEs. The proposed method outperforms the hard thresholding method in the case of $\alpha = 0.75$.

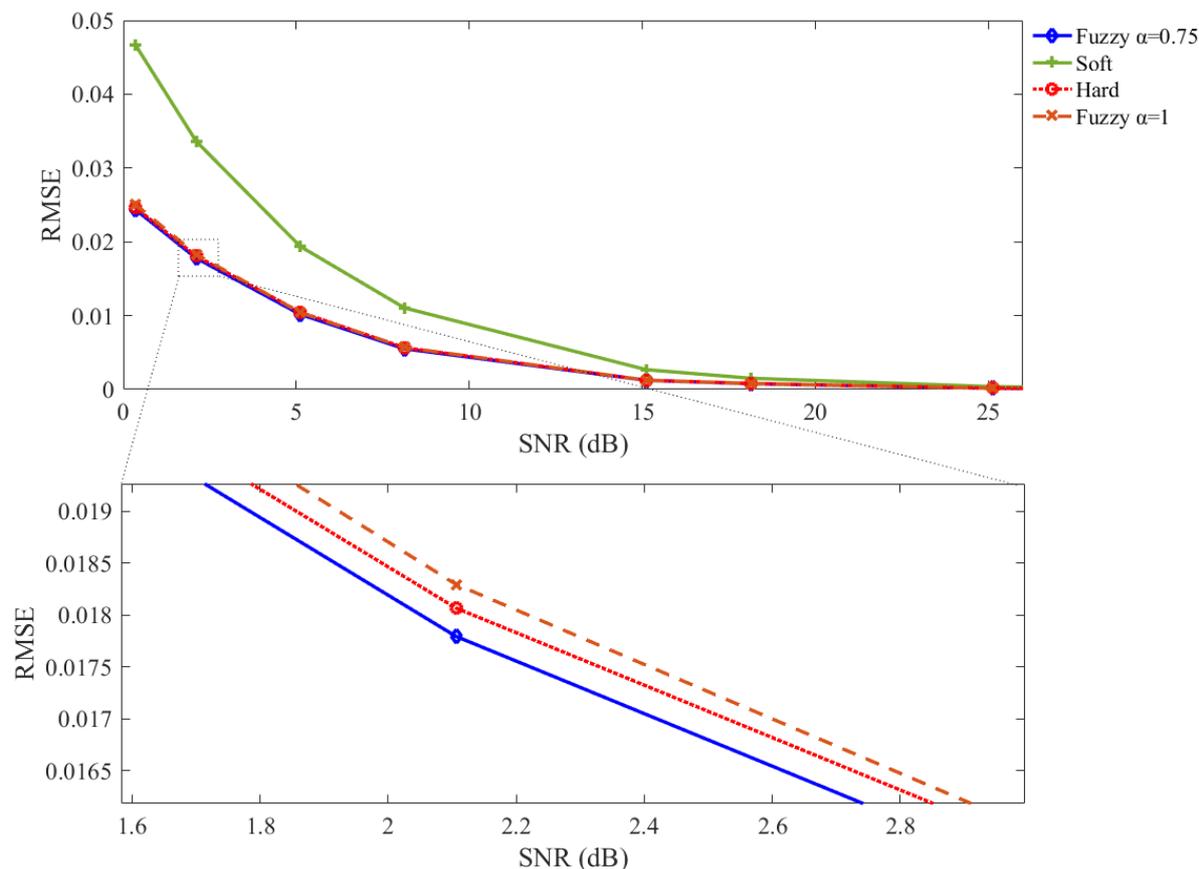


Fig. 3.6 Results of denoising for the generated signal, $S(x)$, using simulations.

Case 2. In the second case study, several speech signals of different subjects have been extracted from ‘A Manitoban Speech Dataset’ [18] and are used as experimental data with additive white gaussian noise. The speech data were recorded with a sampling frequency of 44.1 kHz with 16 bits of resolution.

For the purposes of the SNR calculation, the whole epoch with the silence part has been included as the input data to the algorithm. However, it should be clear that the utterance occurs at a portion of the whole epoch. Also, for the signal to be eligible for wavelet decomposition, 65536 samples have been considered and the decomposition has been performed for 15 stages.

Fig. 3.7 demonstrates the results of the denoising method for the utterance "measure" of subject f06. Table 3.1 describes the SNR and RMSE values of the denoising methods shown in Fig. 3.7. The noisy signal has an original SNR of 3.5dB.

Table 3.1 Results of the denoising for the "measure" epoch as shown in Fig. 3.7.

Measure	Proposed Thresholding	Soft Thresholding	Hard Thresholding
RMSE	0.00776	0.01004	0.00777
SNR (dB)	26.6	34.2	23.8

As shown in Fig. 3.7, while the soft threshold signal has the highest SNR, the produced signal is unacceptable due to the existence of distortion. This is further indicated by the high value of the RMSE as presented in Table 3.1. Regarding the hard threshold and the proposed method, while the results are fairly close, the hard threshold reconstructed signal is noticeably shaken in Fig. 3.7. The proposed fuzzy method outperforms the hard threshold function both with respect to SNR and RSME, as demonstrated in Table 3.1. In fact, the SNR value of the proposed method is approximately 2.8 dB greater than the hard threshold function.

Fig. 3.8 demonstrates the results of the denoising method for the utterance "test" of subject f09. Table 3.2 describes the SNR and RMSE values of the denoising methods shown in Fig. 3.8. The noisy signal has an original SNR value of 14.3dB.

Table 3.2 Results of the denoising for the "test" epoch as shown in Fig. 3.8.

Measure	Proposed Thresholding	Soft Thresholding	Hard Thresholding
RMSE	0.00335	0.00455	0.00337
SNR (dB)	22.1	24.3	21.5

As shown, the hard thresholding, soft thresholding and the proposed fuzzy thresholding have similar behaviours. However, still distortion exists in the reconstructed signal of the soft threshold function. This inadequacy in the performance of the soft threshold function is further emphasized by the higher value of RMSE corresponding to this method relative to its counterparts. In this case, also, the proposed fuzzy threshold outperforms the hard threshold function by a small margin of 10^{-5} with respect to RSME and 0.626 dB with respect to SNR. This is expected, as was indicated in Fig. 3.6, that in higher original SNRs soft threshold, hard thresholding and the proposed method have similar behaviours and closer RMSEs.

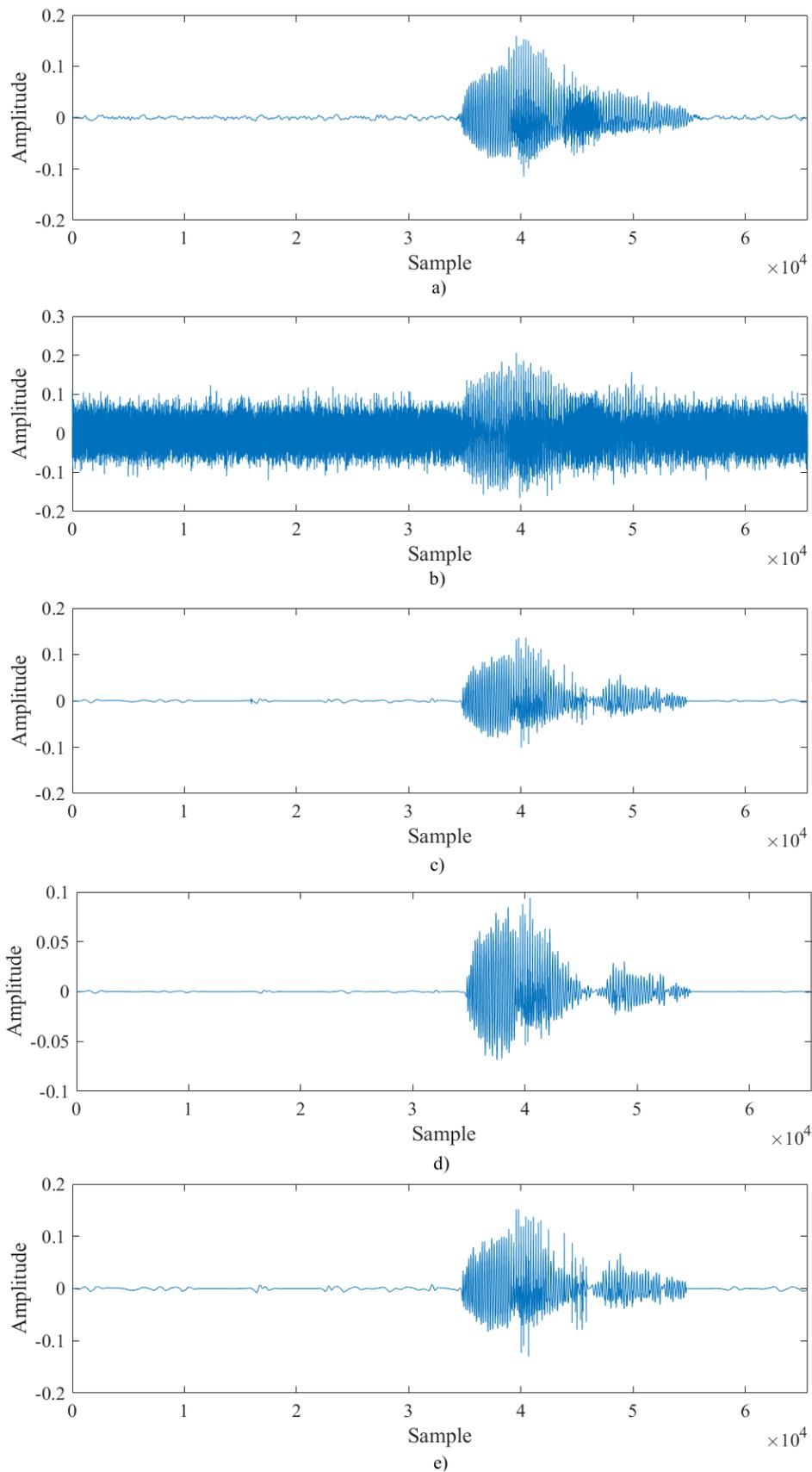


Fig. 3.7 Epoch denoising results for the word "measure", subject f06. a) original signal, b) signal with noise, c) proposed fuzzy threshold, d) soft threshold, and e) hard threshold.

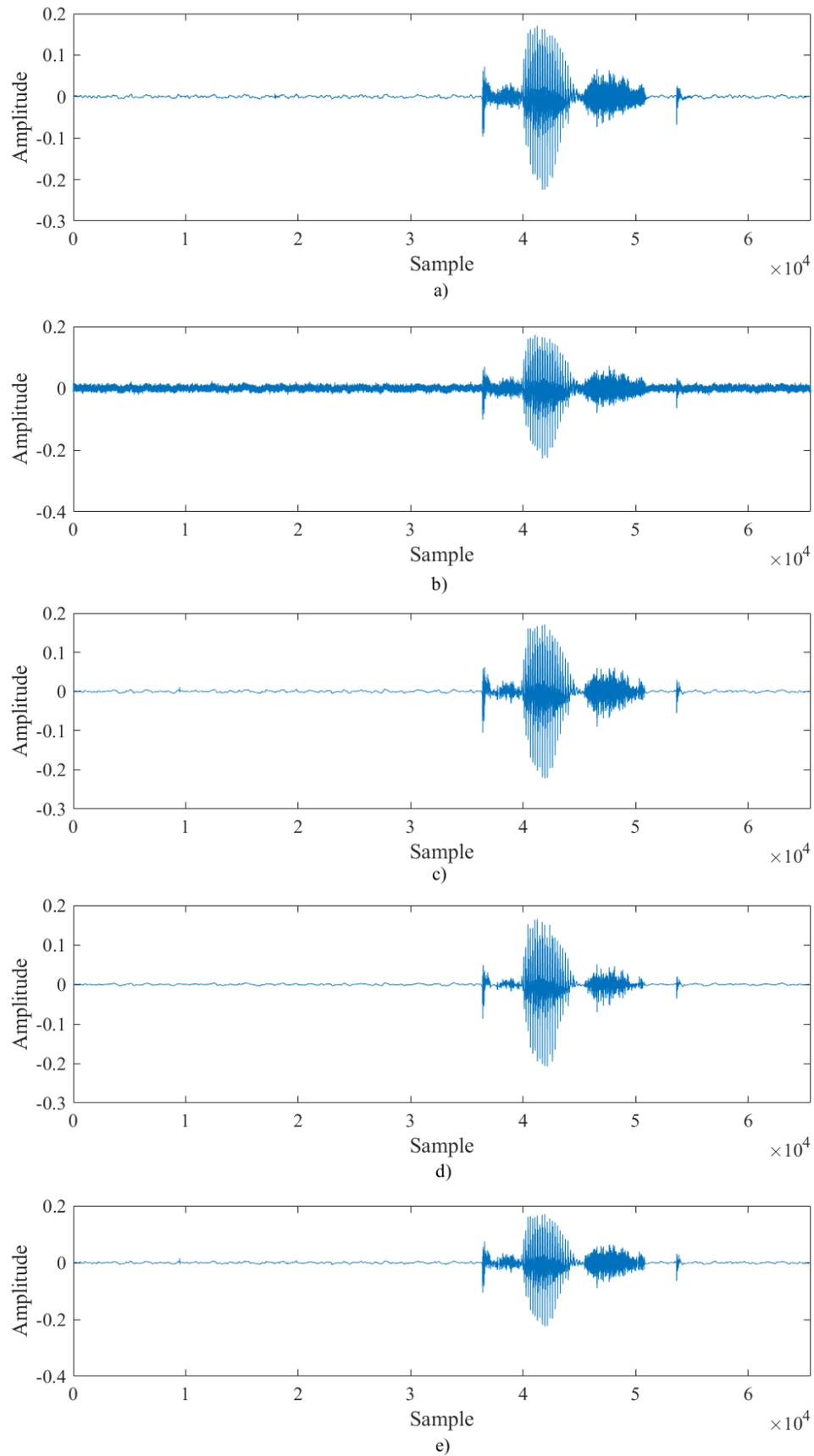


Fig. 3.8 Epoch denoising results for the word "test", subject f09. a) original signal, b) signal with noise, c) proposed fuzzy threshold, d) soft threshold, and e) hard threshold.

Case 3. For the third case study, the pressure signals P_A and P_B have been recorded in an operator-generated input mode experiment. The results of this experiment (including the noisy and denoised pressure signals) are shown in Fig. 3.9.

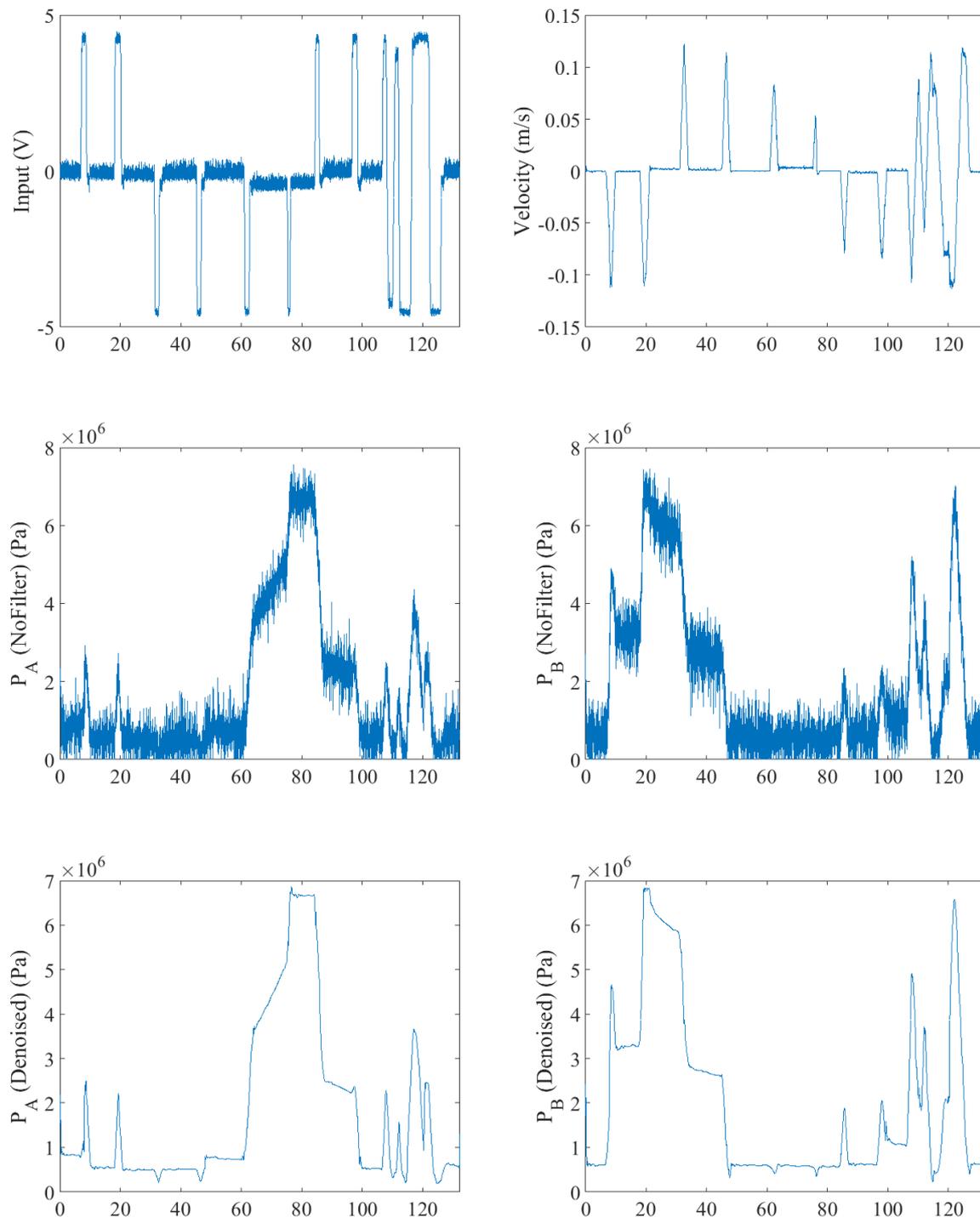


Fig. 3.9 Results of denoising the pressure signals in an operator-generated experiment.

As shown in Fig. 3.9, the pressure signals have a high level of noise with an original SNR of 6.6 dB 6.8 dB for P_A and P_B , respectively. The proposed fuzzy-denoising method has

successfully removed the noise from the signals without any visible distortion while preserving the small changes in the signals. The denoised signals have an SNR of 29.1 dB 30.6 dB for P_A and P_B , respectively.

Therefore, as the results indicate the satisfactory performance of the proposed fuzzy threshold function, this denoising method is used in the Raspberry Pi DAQ and control as the way of dealing with noise in the signals in Chapters 4 and 5.

3.3. Raspberry Pi-based Data Acquisition and Control

The DAQ and Control subsystem is the bottleneck of all electrical communication and *Input/Output (I/O)* signals. The real-time property, the closed-loop control and condition monitoring performances depend on the design of this subsystem. To guarantee this performance peripheral interfaces and communication between the controllers and I/O devices should be explicitly designed. Also, incorporating a multiprocessing scheme in the DAQ design provides the possibility of separating the human interfacing and machine interfacing, which could make the closed-loop and open-loop strategies robust. Furthermore, Analog to digital conversion happens with a pre-estimated error budget which is essential to ensuring real-time performance. The Raspberry Pi-based DAQ is shown in Fig. 3.10.

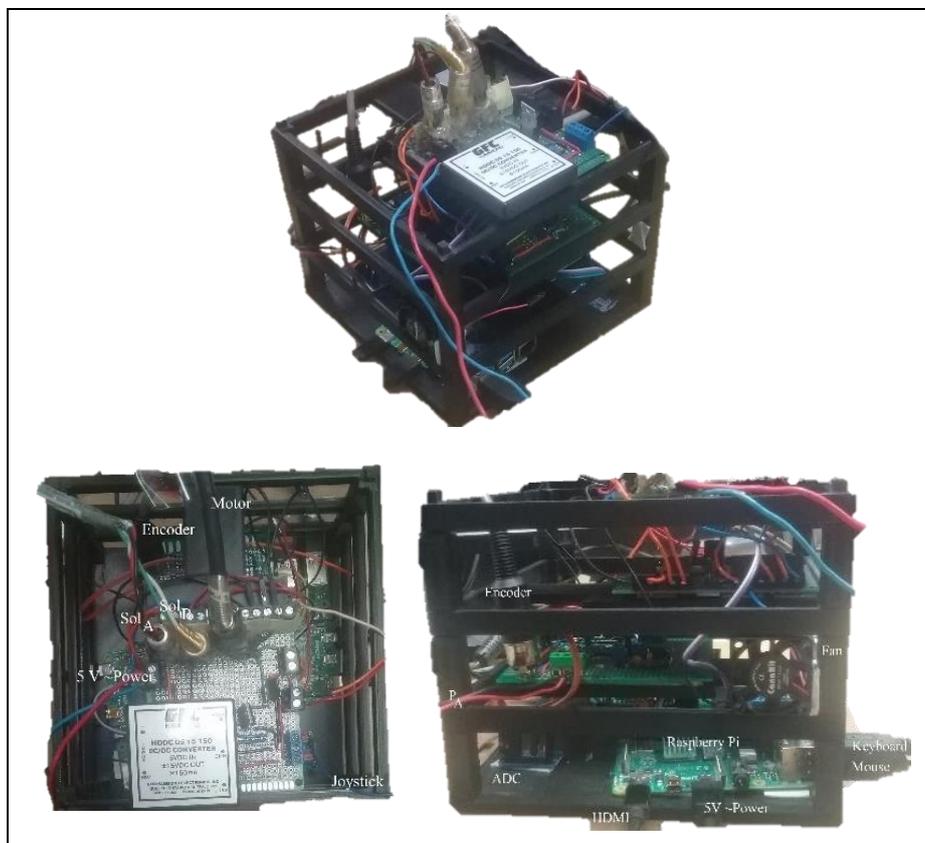


Fig. 3.10 Raspberry Pi-based data acquisition and control.

As shown in Fig. 3.10, The final product fits in a small box with dimensions of $16\text{cm} \times 16\text{cm} \times 13\text{cm}$. It has one main power supply, and sockets and terminals for direct connection to the system. Also, each Raspberry Pi has a keyboard, mouse and monitor allocated to it.

3.3.1. Architecture and Components

The overall architecture and the components of the DAQ and control subsystem with relation to the whole EHA system are as shown in Fig. 3.11. Two Raspberry Pis have been utilized; one is interfacing with a human input signal, called *Human Signal Controller* (HSC), and the other interfaces with the mechanical subsystems of the single-rod EHA, named *System Signal Controller* (SSC).

It should be noted that while DAQ and control is a subsystem of the EHA system, it is referred to as a system after this, with the understanding that it is described in this section as a standalone device.

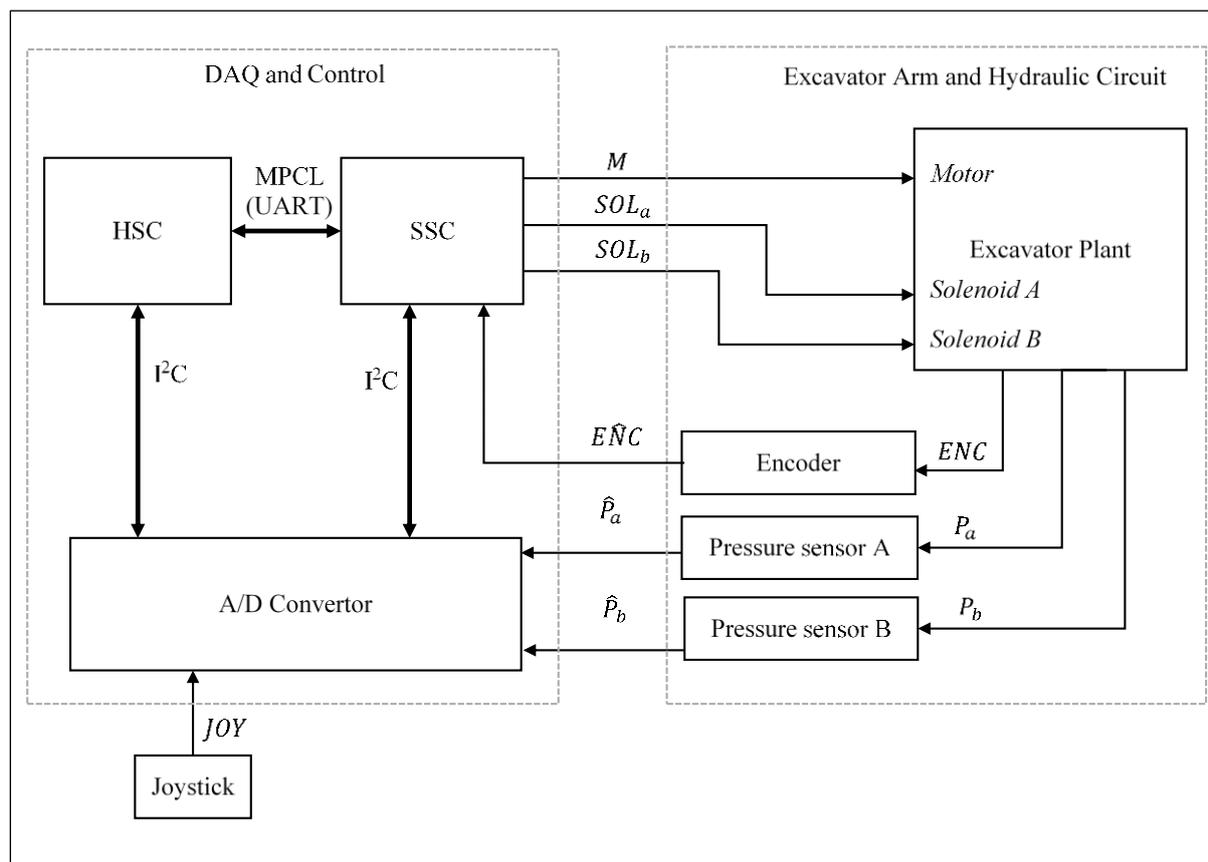


Fig. 3.11 Data acquisition and control subsystem components.

As shown in Fig. 3.11, the SSC Raspberry Pi receives the pressure signals and the encoder signal from the test rig. The HSC Raspberry Pi receives the joystick signal from the human interface and sends it to the SSC Raspberry Pi via the *Multiprocessing Communication*

Line (MPCL). Based on the received data, SSC generates the appropriate output signals which are then fed back to the test rig. Table 3.3 describes the signals which are mentioned and shown in Fig. 3.11.

Table 3.3 Signal properties of data acquisition and control subsystem shown in Fig. 3.11.

No	Sig.	Description	I/O	Range	Resolution	D/A	Speed
1	<i>MPCL</i>	Communication of the controllers (UART)	-	0/3.3 V	-	D	115200 baud/s
2	I2C	ADC and SSC/HSC peripheral interface (I2C)	-	0/3.3 V	16 bits (for ADC)	D	100 kHz
3	<i>JOY</i>	Joystick output signal	I	[0 +5V]	16 bits (for ADC)	A	3.3 kHz (sampling max)
4	<i>M</i>	Control signal to the motor driver	O	[-6.4V +6.4V]	10 bits	A	Same as system
5	<i>SOL_a</i>	Control signal to Solenoid A of valve	O	0/5 V	-	D	50 kHz
6	<i>SOL_b</i>	Control signal to Solenoid B of valve	O	0/5 V	-	D	50 kHz
7	<i>P_a</i>	Actual pressure value for rod-side	-	0-3000 psi	-	-	-
8	<i>P_b</i>	Actual pressure value for cap-side	-	0-3000 psi	-	-	-
9	<i>ENC</i>	Actual position reading	-	[-0.3m +0.3m]	-	-	-
10	\hat{P}_a	Sensor pressure value for rod-side	I	[1V 11V]	16 bits (for ADC)	A	3.3 kHz (sampling max)
11	\hat{P}_b	Sensor pressure value for cap-side	I	[1V 11V]	16 bits (for ADC)	A	3.3 kHz (sampling max)
12	\widehat{ENC}	Sensor encoder angular position reading	I	0/5 V	4 × 256 or 0.03 mm	D	5 kHz

There are two K1 ASHCROFT [19] pressure transmitters on the cylinder. These are used to obtain the pressure of each side of the cylinder. The minimum response time of the sensor is 5ms and requires a 15-36Vdc unregulated power source.

There is also one rotary optical encoder located on the hinge of the excavator arm. The encoder is Bourns® model EN – Rotary Optical Encoder [20]. This encoder has a resolution

of 256 cycles per revolution with square wave signal output. This translates to 0.03mm resolution for the displacement of the actuator connected to the link. This encoder can operate at a maximum angular velocity of 3000rpm which is well within the operating range of the system. Finally, the rise and fall times of the output signals are 200ns.

Also, a 3-axis P-Q Controls Joystick model 220 [21] is used which works as a voltage divider potentiometer with a power supply of +5V and three signals each representing the rotation of the handle in one axis.

Two pairs of ADCs have been set up in the DAQ which are Texas Instruments' ADS1115 [22] and ADS1015 [23] having 8 channels combined with 16- and 14-bits resolution and work with the sample rate of 8 to 860 SPS and 0.125 to 3.3 KSPS, respectively. The input range acceptable to this ADC is [-6.14 +6.14] volts and it supports *Inter-Integrated Circuit* (I²C) peripheral interfacing protocol. The ADCs have an integrated oscillator running at 1 MHz and do not require external clocks.

The Microcontrollers used in the setup are Raspberry Pi model 4 [24]. The microprocessor used in this model is a 64-bit Broadcom BCM2711, Quad-core Cortex-A72 (ARM v8) with a 4GB LPDDR4-3200 SDRAM. The two Raspberry Pis have been set up with the Raspbian operating system [25] and each have 40 GPIO pins which are the main source of obtaining input signals and generating output signals.

Fig. 3.12 describes the design of the DAQ circuit. As shown, HSC and SSC are interfacing with the MPCL which is a *Universal Asynchronous Receiver-Transmitter* (UART) peripheral interfacing protocol. The ADCs are connected to the Raspberry Pis using I²C communication protocol. The ADC connected to HSC receives the joystick signals. While the three axes of the joystick, as shown in Fig. 3.13, are read through ADC, only the x-axis is used in this DAQ. The ADC connected to SSC receives the pressure signals. The pressure signals are scaled down using a voltage divider as follows to be in the acceptable range of the ADCs.

$$V_{out} = V_{in} \frac{R_6}{R_7} = V_{in} \frac{R_8}{R_9} \quad (29)$$

The Encoder signal is received through GPIOs 11 and 13 of SSC. However, since the encoder uses a 5V logic level, a simple two resistor voltage divider has been implemented to reduce the high voltage to below 3.3V which is Raspberry Pi's voltage level. The resistor values for this purpose have been selected to produce a high voltage of above 1.7V to be detectable as high for Raspberry Pi as follows, based on the minimum high signal generated by the encoder which is 4V.

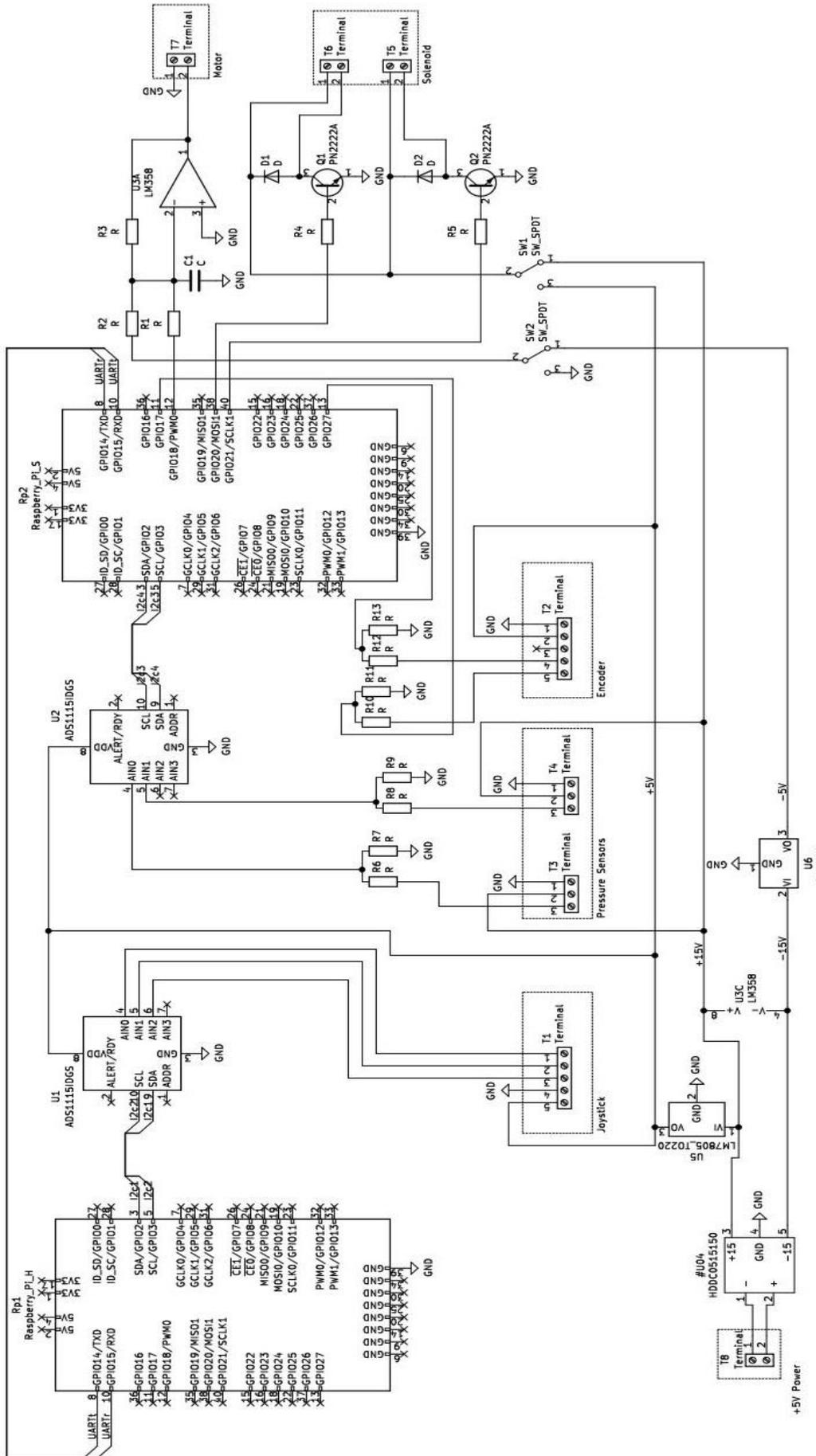


Fig. 3.12 Circuit design of the data acquisition and control system.

$$\begin{cases} \text{for } V_{in} = 4 & 1.7 < V_{out} \left(= V_{in} \frac{R_{10}}{R_{11}} = V_{in} \frac{R_{12}}{R_{13}} \right) \\ \text{for } V_{in} = 5 & V_{out} \left(= V_{in} \frac{R_{10}}{R_{11}} = V_{in} \frac{R_{12}}{R_{13}} \right) \leq 3.3 \end{cases} \quad (30)$$

The encoder signal is thus acquired using an edge-activated custom interrupt function, using GPIOs 11 and 13.

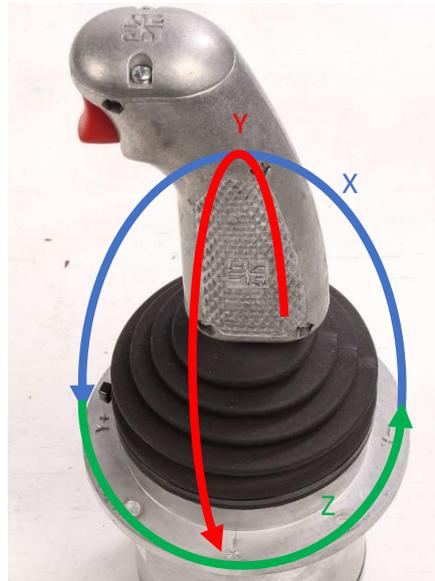


Fig. 3.13 P-Q controls joystick and its axis.

Regarding the outputs, the two solenoid signals are translated to a +5/+15V signal via a resistor-transistor circuit with a diode for circuit protection in case of any possible faults or surges. A switch has been implemented for the high voltage connected to the solenoids to choose between a +5V or +15V (or any other value simply by changing the switch pinouts). This gives the circuit a wide compatibility range for different industrial solenoid-controlled valves.

The signal to the motor that runs the main pump is generated using pulse modulation with 10 bits of resolution and a *Low-Pass Filter* (LPF) cutting the high-frequency component. This signal is then scaled to $[-6.4V \ +6.4V]$ using a negative feedback op-amp and a compensating $-5V$ voltage, V_{comp} , as follows:

$$V_{out} = V_{in} \frac{R_3}{R_1} + V_{comp} \frac{R_3}{R_2} \quad (31)$$

Since various sensors and actuators in the system require a wide range of powers, a power circuit has been incorporated into the design. The power adapter external to the circuit is an unregulated voltage of $\sim 5V$ from a common +5V commercial plug-in power supply. This

power is then amplified to provide -15V, GND and +15V. Furthermore, a positive voltage regulator and a negative voltage regulator are used on the +15V and -15V lines, respectively, to provide accurate +5V and -5V power supplies in the DAQ circuit. Table 3.4 describes the power supply used for different components in the circuit. Table 3.5 describes all components of the circuit.

Table 3.4 Voltage supply needed for circuit components shown in Fig. 3.12.

No.	function	Description	Supply
1	Sensor	Pressure transducer	15V
2	Sensor	Encoder	5V
3	User input	Joystick	5V
4	Active element	Op-Amp	±15V
5	Actuator	Solenoid Activation	12+/5+V
6	ADC	ADC	5V

Table 3.5 Electric circuit's elements.

No.	Tag	Description	No	Tag	Description
1	R1	Resistor, 6 kΩ	19	Rp1	HSC
2	R2	Resistor, 18 kΩ	20	Rp2	SSC
3	R3	Resistor, 24 kΩ	21	U1	ADS 1115/1015 [22, 23]
4	R4	Resistor, 6 kΩ	22	U2	ADS 1115/1015 [22, 23]
5	R5	Resistor, 6 kΩ	23	U3A	LM358 dual op-amp [28], #1
6	R6	Resistor, 6.81 kΩ	24	U3C	LM358 dual op-amp [28], Supply
7	R	Resistor, 6 kΩ	25	U4	DC/DC convertor 05 15 150
8	R8	Resistor, 6.81 kΩ	26	U5	LM 7905[29]
9	R9	Resistor, 6 kΩ	27	U6	MC 7805[30]
10	R10	Resistor, 5 kΩ	28	T1	Terminal for Joystick
11	R11	Resistor, 6 kΩ	29	T2	Terminal for Encoder
12	R12	Resistor, 5 kΩ	30	T3	Terminal for Pressure Sensor, A
13	R13	Resistor, 6 kΩ	31	T4	Terminal for Pressure Sensor, B
14	C1	Capacitor, 1 μF	32	T5	Terminal for Solenoid, A
15	D1	1N4007 diode [26]	33	T6	Terminal for Solenoid, B
16	D2	1N4007 diode [26]	34	T7	Terminal for Motor
17	Q1	2n2222A transistor [27]	35	T8	Terminal for power source
18	Q2	2n2222A transistor [27]	36	GND	Common ground

3.3.2. Programming

The program for this DAQ and control system has been written in C with constant interval consideration. While the designed system has a range of 8 SPS to 3.3 KSPS, the main program has been implemented with the sampling frequency of 100 Hz and ADS1115 which is adequate for experiments needed in this thesis.

Since the pressure signals are broadband, instead of cut-off frequency, the highest frequency components are used for the sampling test analysis [31]. The highest frequency component of the pressure signals is approximately 10 Hz [32]. This is further shown in Fig. 3.14 for the healthy operating mode. Consequently, using the Nyquist sampling theorem [10] it is inferred that the sampling frequencies which are greater than 20 Hz satisfy the reconstruction requirements of the signal. However, oversampling the pressure signals provides the ability to deal with noise in this signal which is of great importance and is discussed later. Furthermore, as shown in Fig. 3.14 the noise in the pressure signals is a combination of white noise (with constant power spectral density) and pink noise ($1/f$ noise). It should be mentioned that the classification of the broadband and narrow band, and frequency content of noise in the pressure signal entails extensive discussion which is outside the scope of this thesis.

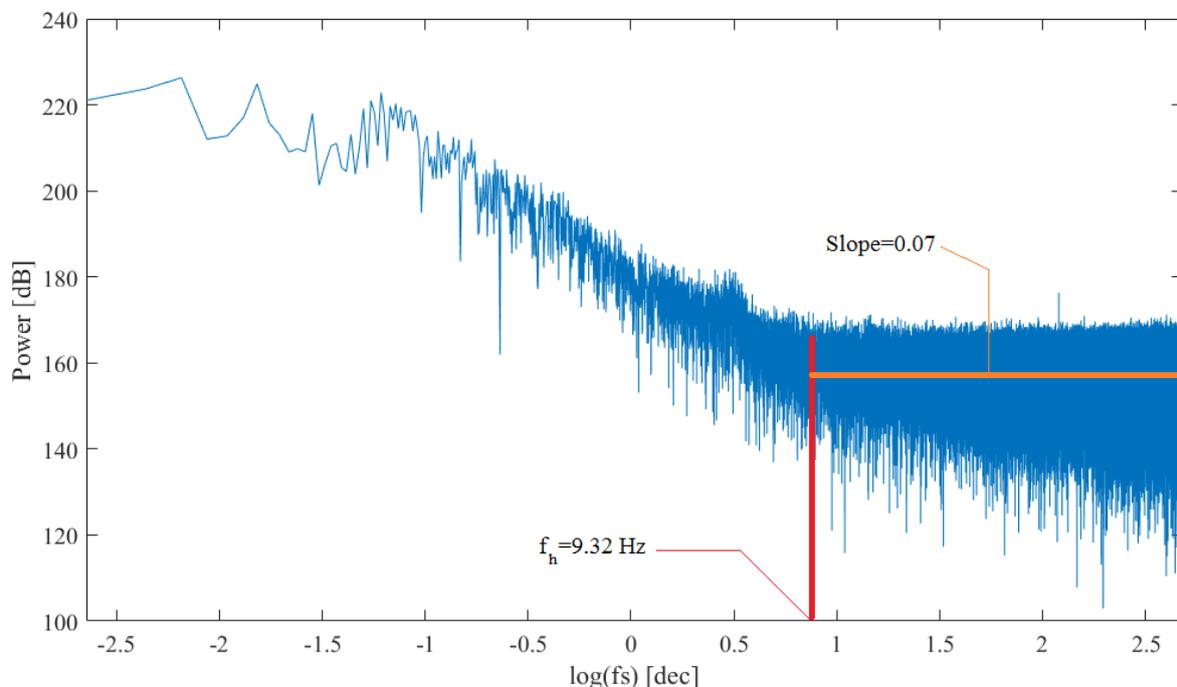


Fig. 3.14 Power spectral density analysis of the pressure signal, P_A .

There are two peripheral interfaces in the DAQ and control system. First, a universal asynchronous receiver-transmitter protocol, UART, is the means of MPCL communication

which incorporates connecting the transmitter GPIO of SSC to the receiver GPIO of the HSC and vice versa. In UART, bits are sent one at a time (serial). Each packet of data has a start and stop bit marking the beginning and end of the data bits which are placed from the *Least Significant Bit* (LSB) to the *Most Significant Bit* (MSB) [33]. This UART has been configured to run at a baud rate of 115200, with 12-bit bauds.

Second, inter-integrated circuit peripheral interfacing protocol, I²C, is used for communication with the ADCs at the speed of 100kHz [22, 23]. I²C is a synchronous communication protocol which means in addition to the *Serial Data* (SDA) line, it includes a *Serial Clock* (SCL) line [33]. In this system, HSC and SSC are configured to act as masters and ADCs are configured as slaves. The slave sends an interrupt request and the master if free establishes the communication using the two aforementioned lines, SDA and SCL.

The communication through I²C to the ADCs is done via the config register [22, 23]. The functionality of the 16 bits in this register is described in Table 3.6.

Table 3.6 Config register of ADS1115/1015 [22, 23].

No.	Bit/Bits	Description
1	{15}	Flag bit for single shot
2	{14:12}	Input selection
3	{11:9}	Programmable gain amplifiers
4	{8}	Operational mode (continuous/single-shot)
5	{7:5}	Data rate {000:111}
6	{4:0}	Comparator functions

Since offset and gain errors specifications are guaranteed over the full operating temperature range of -40°C to 85°C a *Total Unadjusted Error* (TUE) [33] calculation method is used to generate an error budget for the worst-case error that could possibly be observed in ADC operation. TUE is a comprehensive specification that incorporates the effects of nonlinearity, gain error and offset error on the performance of the device. In an ideal scenario, the TUE would be 0.5 LSB which is due to the quantization error.

The configured full scale of the ADC is ± 6.144 V and the LSB is 187.5 μ V. With these configurations, the nonlinearity error is 1 LSB, the offset error is 3 LSB, and the gain error is 0.15% (9.2 mV or 49 LSB). The temperature drift is already built into these specifications and is accounted for. The worst-case error would occur if all of these specifications had the same polarity, so the error budget would be 52 LSB or approximately 9.8 mV.

The majority of the error budget is due to the gain error, which is 0.15% for the extreme of the operating temperatures. However, gain error is substantially smaller (0.01%) for

temperatures around 25°C. Thus, if the operating temperature were to remain around 25°C, it would lower the error budget to 7 LSB or approximately 1.3 mV.

This high dependency of the error budget to temperature variation implies the need for a temperature sensor in the DAQ and control system. Thus, the existing temperature sensors in the Raspberry Pis are utilized for temperature monitoring.

The 100-point regression method is used for deriving velocity from the position of the excavator arm as follows:

$$v(t) = \frac{\sum_{i=1}^{100}[x_i - \exp(x)][t_i - \exp(t)]}{\sum_{i=1}^{100}[t_i - \exp(t)]} \quad (32)$$

where $\exp()$ is the expected value function and t_i is the time at the step ($t - i$).

Finally, it should be mentioned that for the designed DAQ and control system a container for shock and heat protection has been designed and 3D printed as shown in Fig. 3.10.

3.4. Experimental Results for Comparison of The Data Acquisition Systems

The designed Raspberry Pi-based DAQ and control system is tested and compared with a Quanser Q8 in this section.

Quanser Q8 [9] is a hardware-in-the-loop DAQ with 32 digital input/outputs at 5V logic level, 8 analog inputs with 14-bit ADC resolution and 8 Analog outputs with 12-bit DAC resolution both at $\pm 10V$. Quanser also provides 8 encoder reading sockets with built-in counters. The nominal maximum sampling speed of the Quanser is 56 kHz. Quanser uses a PCI interface at 33 MHz for communication with computers.

First, the performance of the Raspberry Pi and Quanser DAQs are investigated in the open-loop form with a comprehensive analysis of the effect of noise reduction processes. Then the high-speed performance of the Raspberry Pi is discussed with all signals described thoroughly. Then, closed-loop Proportional Integral Derivative (PID) control capabilities of the Quanser and Raspberry Pi are compared as position controllers. Also, the human in the loop experiments with Raspberry Pi as the velocity controller is presented. Finally, an overall comparison of the two DAQs is presented. The sampling frequency of the experiments is set at 100 Hz except if stated otherwise.

3.4.1. Accurate Acquisition of Pressure Signals in Operator-generated Input Mode

By connecting the pressure signals to the Quanser and Raspberry Pi and using the joystick signal as input to control the speed of the motor, an experiment has been performed. The data in Fig. 3.15 shows the original joystick signal and Fig. 3.16 describes the pressure

signal. After the experiment, a conservative low-pass filter with a cut-off frequency of 80Hz has been used for all three DAQs to make the signals visually comparable.

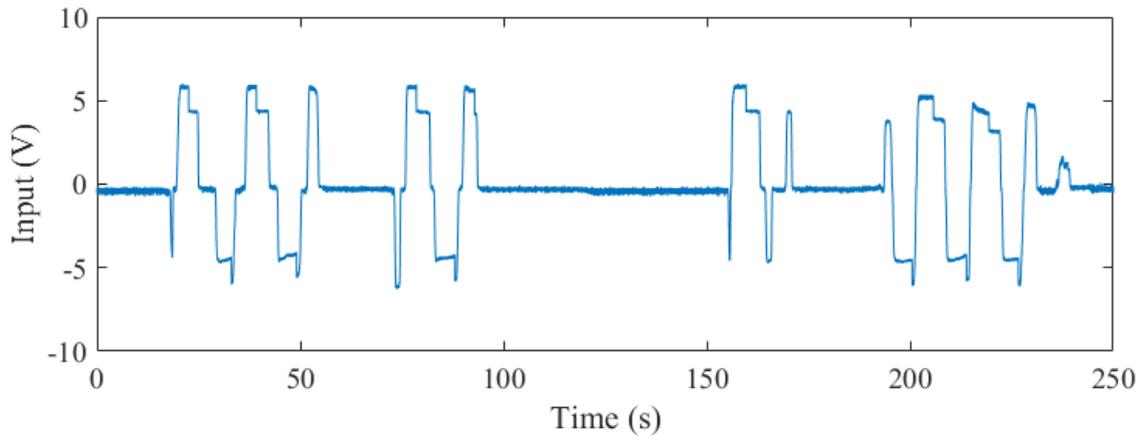


Fig. 3.15 Input (from joystick) to the experimental setup.

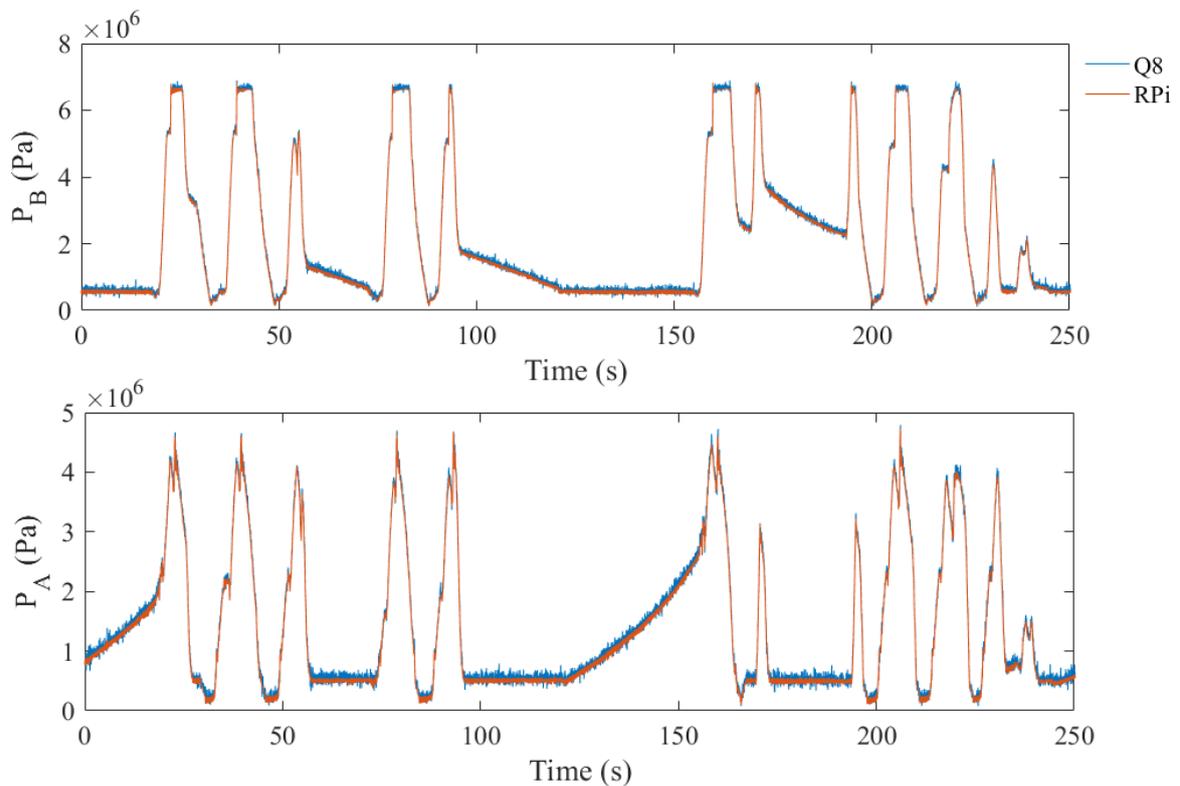


Fig. 3.16 Pressure responses pertaining to input shown in Fig. 3.15.

As shown, Raspberry Pi and Quanser follow the same trends with small offsets due to their biases. However, both DAQs have significant noise which can be seen in P_A at $t=100$ s to $t=120$ s. This high level of noise in the signal points towards the importance of denoising.

Fig. 3.17 provides a closer look at the same pressure signals from $t=15$ s to $t=40$ s. At first glance, it appears that the signals from the Quanser have a larger variance relative to the

Raspberry but are closely following the same trends. To quantify these observations, first, the methods for dealing with noise associated with each of the DAQs are implemented on their corresponding signals and then the statistical moments are compared.

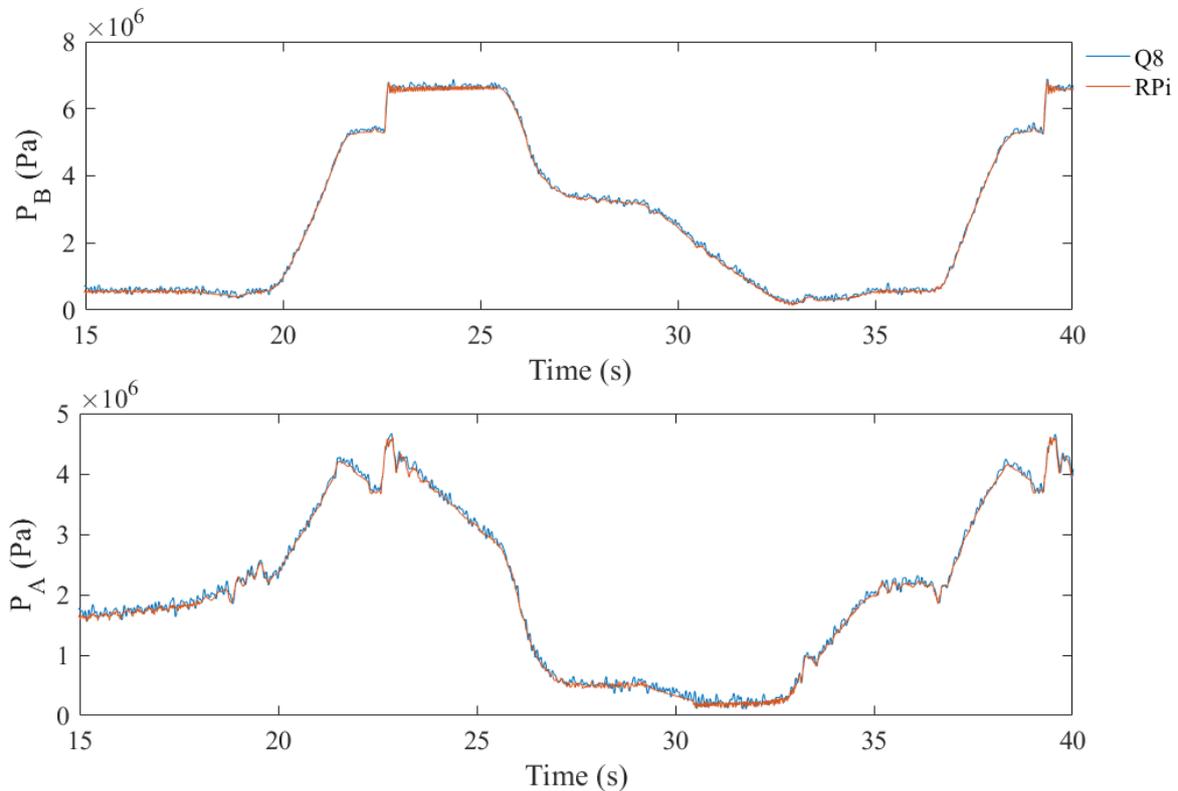


Fig. 3.17 Pressure responses (between $t=15s$ to $t=40s$) pertaining to input shown in Fig. 3.15.

It is of great importance to be noted, that the denoising method has been set on the program for the Raspberry Pi DAQ. On the other hand, the provided method of dealing with noise in Quanser is solely low-pass filters and other schemes for noise reduction are left for commercial users to be implemented on the device. In other words, the different processes for noise reduction are incorporated into and are a substantial part of each DAQ's overall design. The low-pass filters have the following transfer function and the cut-off frequency of $\omega_c = 10\text{Hz}$.

$$LPF(s) = \frac{1}{0.1s + 1} \quad (33)$$

Also, the proposed fuzzy denoising method has been used for Raspberry Pi with 5 levels of decomposition. The results of the pressure signals generated from the noise reduction processes are shown in Fig. 3.18. Also, Fig. 3.19 demonstrates the same signals within a smaller time frame for a clear comparison.

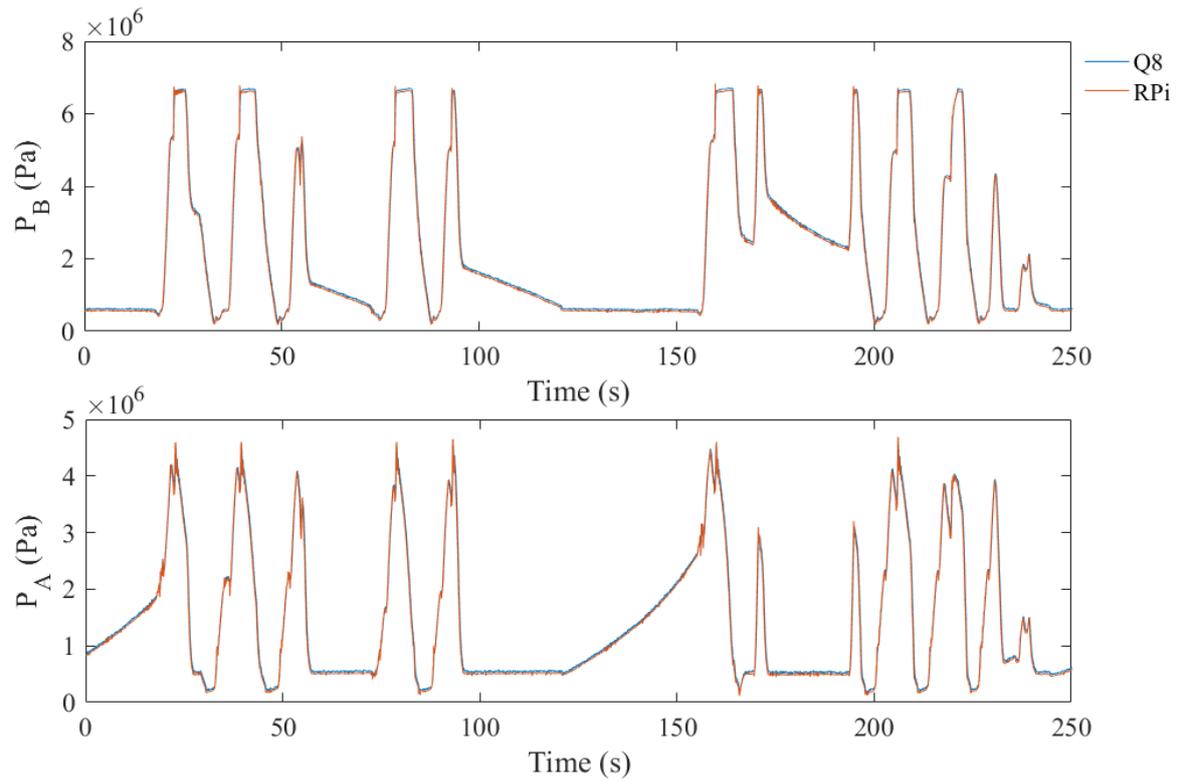


Fig. 3.18 Pressure responses with noise reduction pertaining to input shown in Fig. 3.15.

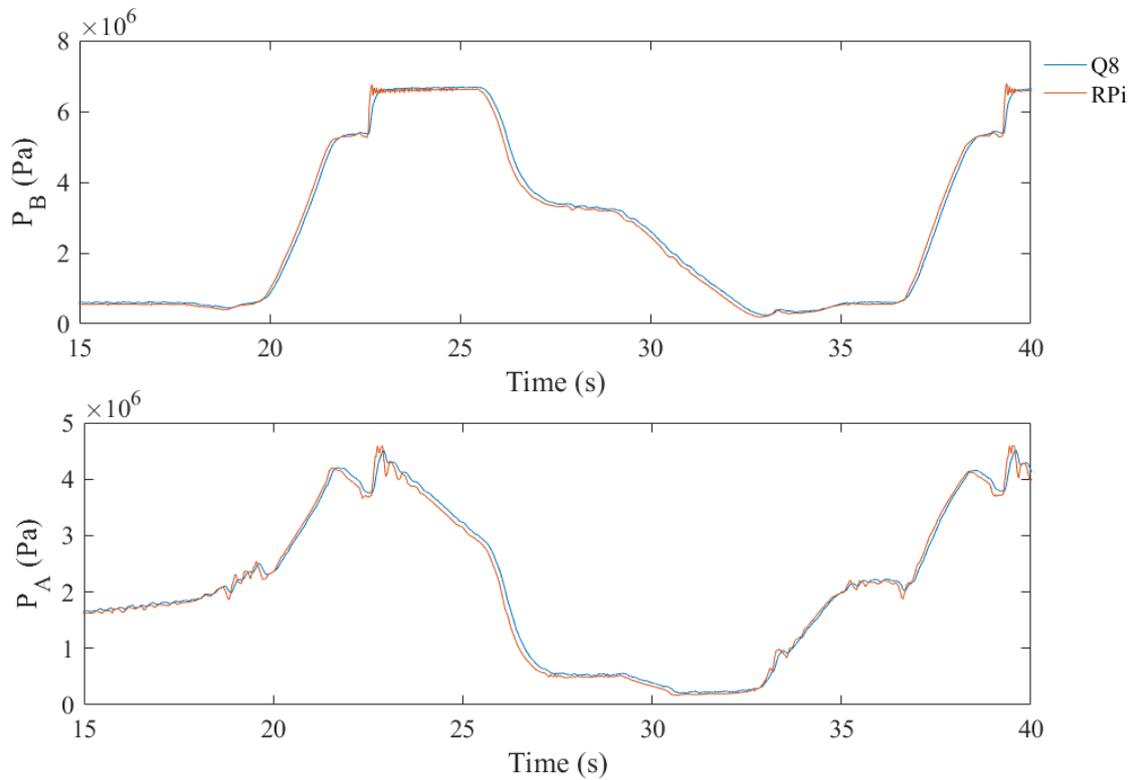


Fig. 3.19 Pressure responses with noise reduction (between $t=15$ s to $t=40$ s) pertaining to input shown in Fig. 3.15.

As shown, the pressure signals follow the same trends with insignificant offsets. However, the high-frequency components which exist in both P_A And P_B are distorted in the Quanser acquired signals where LPFs are used. This is more apparent at approximately $t=23s$ to $t=25s$. At these points, the pressure signal, P_B , has reached its maximum permissible value by hydraulic circuit design. Thus, the relief valve corresponding to the high-pressure side (B-side or rod-side) is opened. During this period, the pressure signal should be around ~ 6.7 MPa pressure point with damping oscillations. These expected behaviours exist in the pressure signal, P_A and P_B , obtained from the proposed denoising method. This is not the case for LPFs where these oscillations are present for P_A but significantly attenuated. Also, the oscillations have vanished in the P_B signal obtained by using LPFs.

At approximately $t=21s$ the pressure signal P_B is expected to follow the same corner trend in the noisy signals seen in Fig. 3.17. While this behaviour has remained intact in the signal obtained from the raspberry pi, it has been significantly smoothed in the Quanser generated signals due to the use of low-pass filters.

As mentioned, to quantify these behavioural characteristics of these signals their statistical moments have been obtained for a moving window with the size of 1024 samples. The first four statistical moments (mean, variance, skewness, and kurtosis) for pressure signals, P_B and P_A are shown in Fig. 3.20 and Fig. 3.21, respectively.

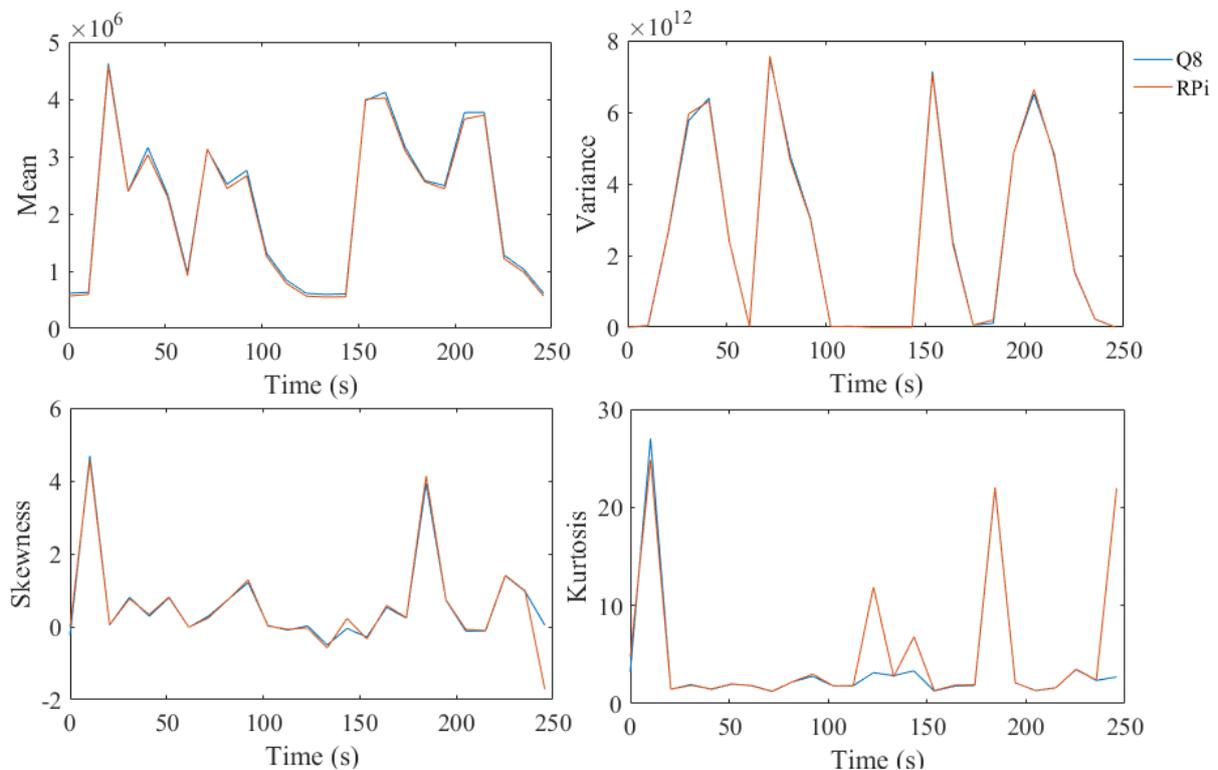


Fig. 3.20 Statistical moments of the pressure signal, P_B , pertaining to Fig. 3.18.

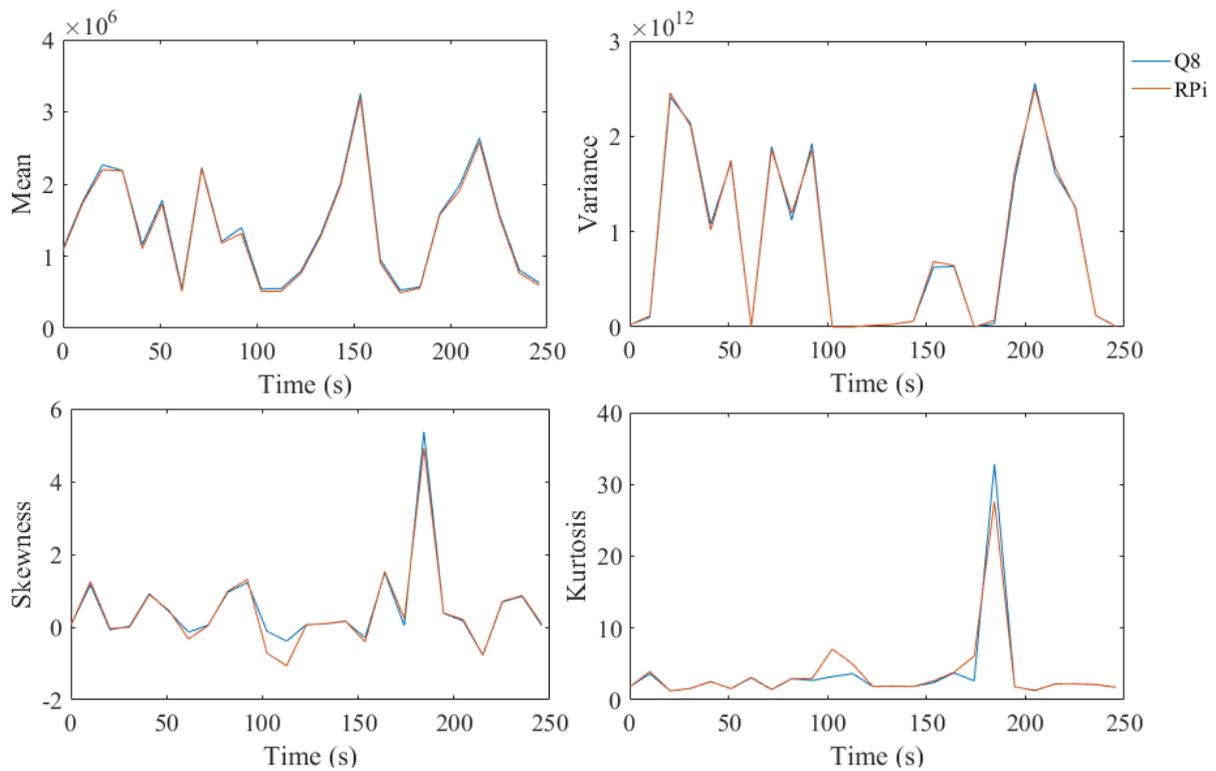


Fig. 3.21 Statistical moments of the pressure signal, P_A , pertaining to Fig. 3.18.

As shown, the statistical moments have the same values and trends for the majority of the simulation except for approximately $t=90$ s to $t=120$ s in the P_A signal and approximately $t=120$ s to $t=160$ s in the P_B signal. This is due to the stationary behaviour of the signals in those periods and the noise reduction algorithm¹. This means that both the Raspberry Pi and Quanser have acceptable performance concerning to the acquisition of pressure signals. However, concerning the denoising methods, the denoising method has a better performance than LPFs.

3.4.2. High-speed Performance in Operator-generated Input Mode

An experiment with the sampling frequency of 1 kHz has been conducted in the open-loop form with the joystick input voltage directly reflected in motor output. The results of this experiment are shown in Fig. 3.22.

¹ Since in these points the signal is roughly constant with small chattering due to noise, the noise in the signal is dominant. Thus, the residual noise in the signals is responsible for the values of the statistical moments in these periods. Since noise in these signals is different and different noise reduction techniques were used in the three DAQs, these differences are inevitable in the results.

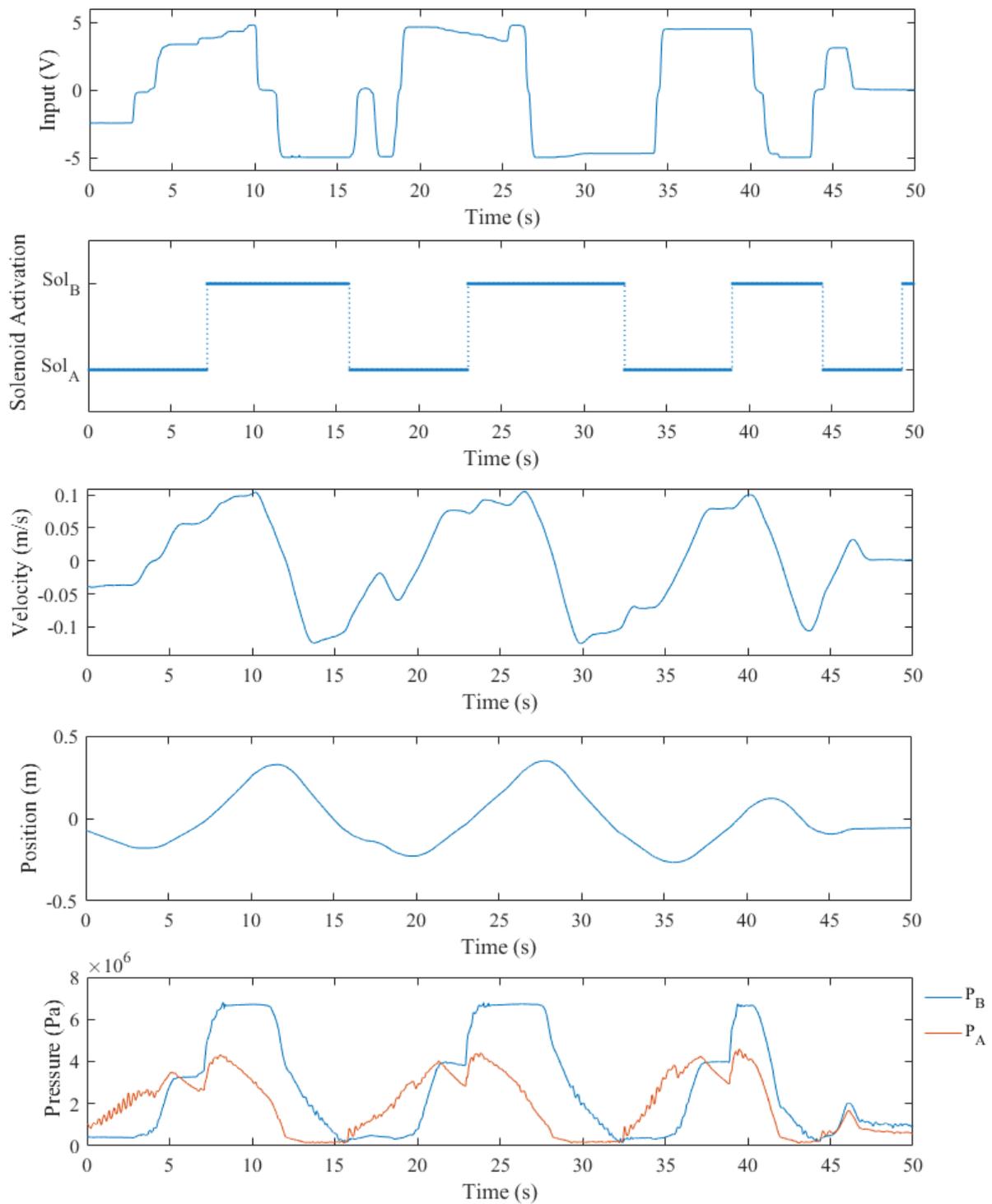


Fig. 3.22 Experimental results in operator-generated input mode with Raspberry Pi data acquisition system.

As shown, the designed DAQ is able to read all the input signals, encoder, and pressures, and provide accurate outputs, to the motor and solenoids. Also, the high-frequency components have been preserved due to the use of denoising rather than filters.

3.4.3. Performance in Operator-generated Reference Mode

Here the velocity control of the EHA system is performed using a Proportional Integral Derivative (PID) controller. The PID transfer function is as follows:

$$C_{PID}(s) = K_P + K_I \frac{1}{s} + K_D s \quad (34)$$

A velocity control scheme in operator-generated reference mode is designed for the Raspberry Pi system. In this case, the velocity of the EHA is controlled by considering the joystick input as the velocity reference for the system in the operator-generated reference mode with a PID controller. PID gains for velocity control simulation are $K_P = 4$, $K_I = 80$, and $K_D = 0.1$. The results of two experiments with different joystick references from different people operating the joystick are shown in Fig. 3.23 and Fig. 3.24.

As shown, the PID controller is able to control the system and track the provided reference. Furthermore, the RMSE errors and maximum overshoots are shown in Table 3.7.

Also, the spikes in the signals (mainly pressures and velocity) which occur due to the quadrant change (as indicated by the solenoid activations in $t=18s$ in Fig. 3.23 and $t=44s$ in Fig. 3.24), are well mitigated by the fast increase and decrease response in the input.

Table 3.7 Results of velocity control experiments using Raspberry Pi data acquisition system.

Measure	Case 1	Case 2
RMSE	0.0147	0.0107
MO	0.0090	0.0021

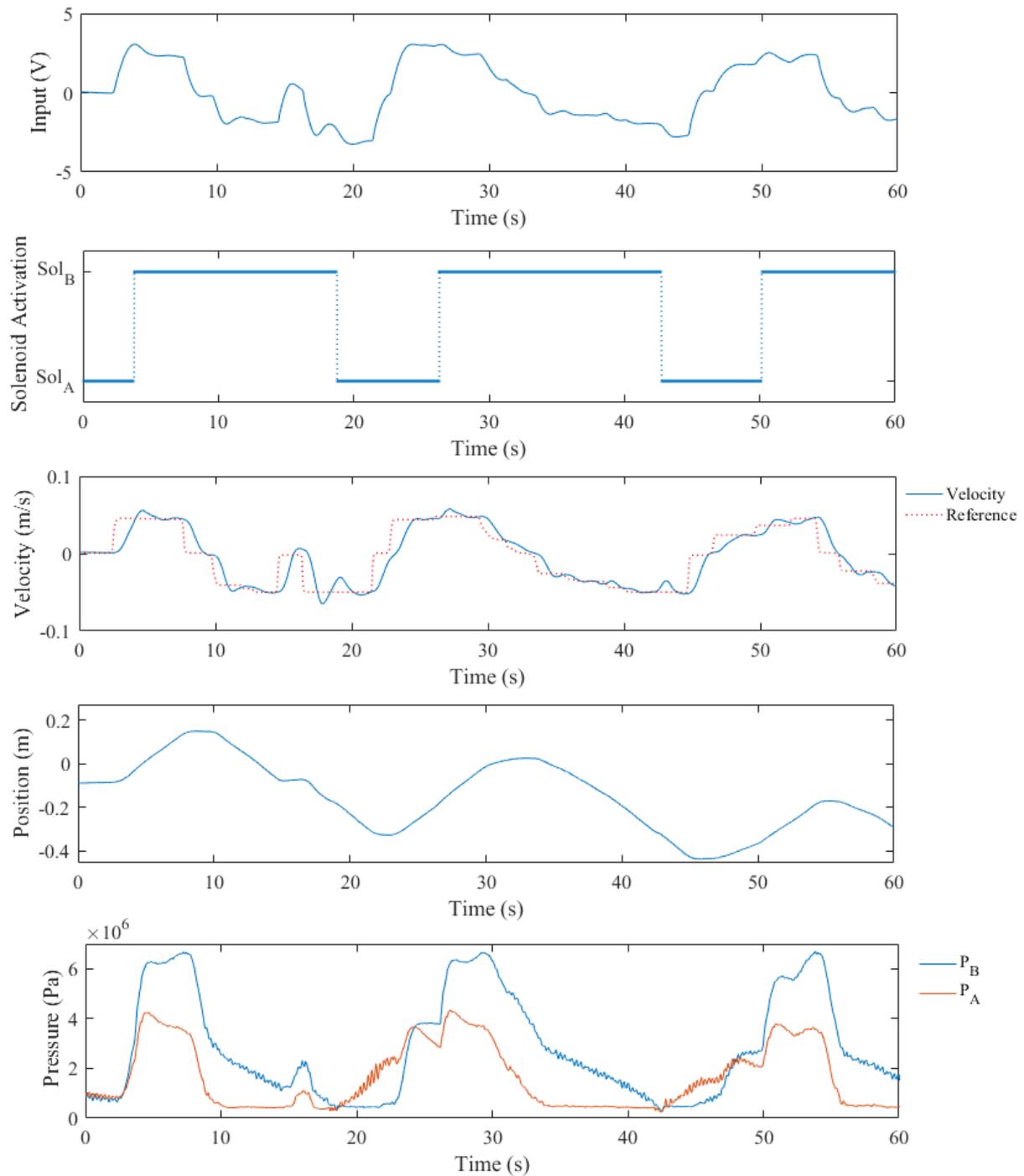


Fig. 3.23 Experimental results of case 1 velocity control test in operator-generated reference mode (using joystick) with Raspberry Pi data acquisition system.

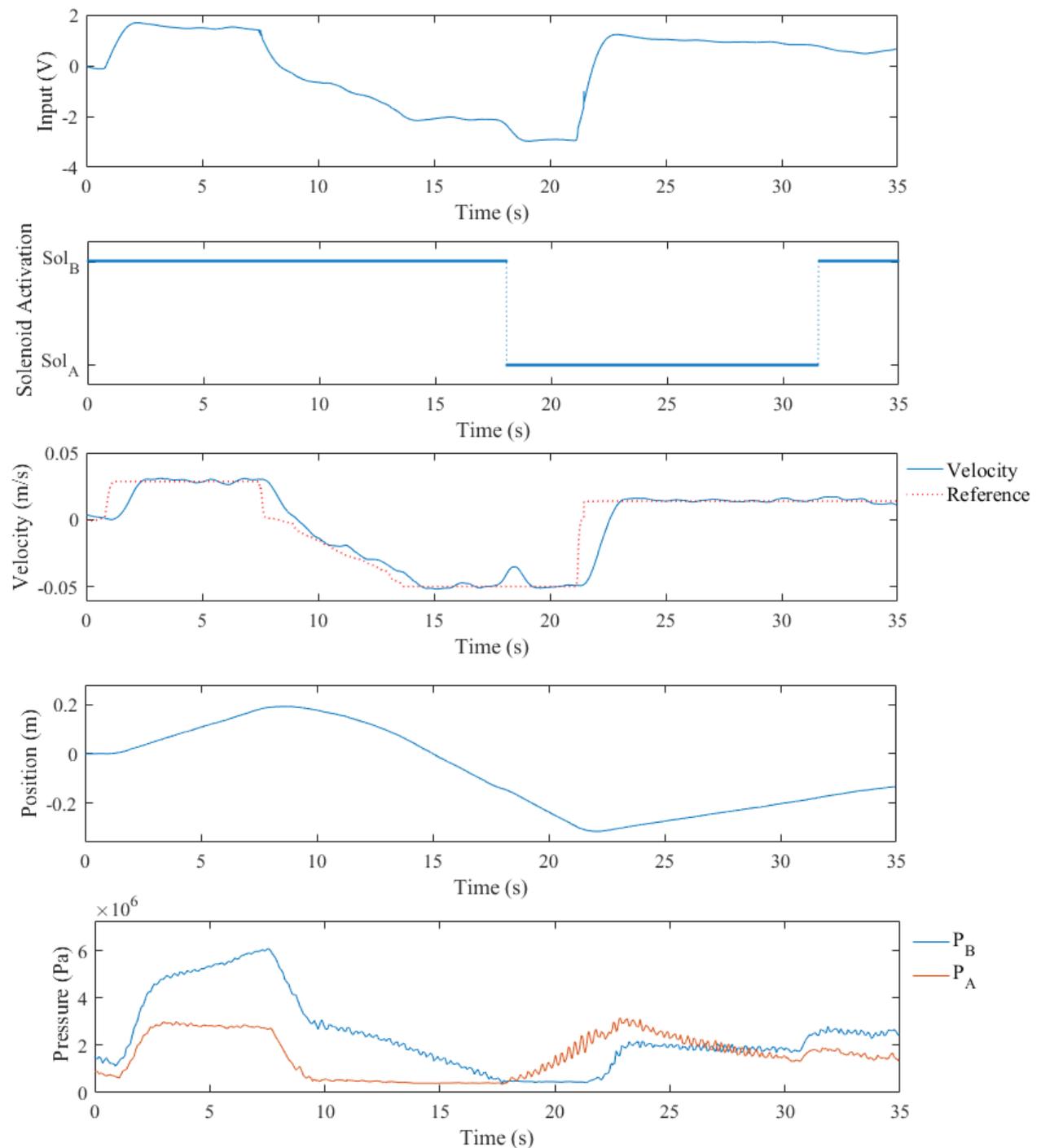


Fig. 3.24 Experimental results of case 2 velocity control test in operator-generated reference mode (using joystick) with Raspberry Pi data acquisition system.

3.5. Summary

In this chapter, a denoising method is proposed for dealing with noise in the signals. Also, a Raspberry Pi data acquisition system is designed and compared with a Quanser Q8.

The proposed denoising method is based on a newly developed fuzzy thresholding method. This method is compared with the conventional denoising methods and LPFs. The

results show that the fuzzy denoising method has a better performance in restoring the signal as indicated by the low RMSEs and maintains the signal's characteristics compared to the LPFs.

With regards to data acquisition systems, the Quanser Q8 DAQ has a wider range of high-speed capabilities up to a nominal 56 kHz sampling rate and the tested sampling rate of 1 kHz for the EHA system. Its ADC resolution, however, is coarser than that of Raspberry Pi's but it has a better DAC resolution. Furthermore, Quanser Q8 requires a constant connection to a computer and does not pose standalone capabilities. In fact, it is most useful when connected to supplementary software such as MATLAB. Thus, Quanser is mainly a research-friendly product rather than a robust industry-friendly device. On the other hand, in addition to having a denoising algorithm rather than LPFs, the Raspberry Pi-based DAQ has a 1 kHz tested sampling frequency (same as Quanser), 16-bit ADC, 10-bit DAC, and 52 digital I/O pins which makes it a prominent candidate for research use. Also, the proposed design is cheap, and it provides useful characteristics while preserving standalone capabilities which makes it a suitable industry-friendly DAQ. In this regard, the proposed Raspberry Pi DAQ and Control subsystem outperforms its competition.

Since the accuracy of the signals are of great importance, the proposed fuzzy denoising method is used in DAQs as the way of dealing with noise for fault detection and fault-tolerant control designs.

CHAPTER 4:

FAULT DETECTION

The main focus of this chapter is on detecting faults in the single-rod Electrohydrostatic Actuator (EHA) system. Here, the EHA signals (mainly pressure) are utilized to detect internal leakage between the two chambers of the actuator and effective bulk modulus change faults. These faults are of importance since they could not be visibly seen and could cause accidents if not detected [4].

Both polyscale analysis (variance and length fractal dimensions) and multiscale analysis (wavelet detail coefficients) are used in the analyses. These fault detection measures are tested in non-operating and operating modes with simulations and experiments. In order to make fault detection possible for both operating and non-operating modes of the system, a comprehensive analysis is first performed to study the systems' behaviours in the presence of the faults. Based on this analysis, a strategy is developed to detect and isolate each of the internal leakage and bulk modulus change faults with different levels. In these analyses, fractal dimension trajectories of the pressure signals and wavelet detail coefficients' statistical moments are combined for a robust fault detection performance. Comprehensive experiments and simulations are performed, to study the sensitivity of the proposed method to the severity of the faults. Finally, an online decision variable has been introduced for internal leakage to assign a value to the degree of fault in the system. Using the three measures provides more robustness for the voting system which in turn contributes to the performance of the fault detection scheme as a whole.

Note, the faults here occur one at a time in the system and the condition where both faults occur at the same time has not been investigated in this thesis.

4.1. Analysis Techniques

In this section, first, the stationarity test is described so that it could be utilized to ensure the validity of fractal dimension measures. Then fractal dimensions and wavelet transform are introduced as tools of signal analysis.

4.1.1. Stationarity Test

In order to use the fractal dimension measures to analyze any given signal, first, it has to be established that the epoch on which the analysis is being done is stationary. Without stationarity, any of the fractal measures would lead to meaningless results.

Many categories of data are inherently non-stationary. If these measures are to be used each signal has to be divided into frames that are stationary. These frames could be overlapping or non-overlapping. Overlapping frames make it possible to emphasize features that are significant for analysis.

In order to test for stationarity of the frames, there are several tests such as the Dickey-Fuller test [34] or *Wavelet Spectrum Test* (WST) [35] which is used in this work. Essentially, the WST algorithm computes an evolutionary wavelet spectral estimate and then computes the Haar wavelet coefficients of a predetermined range of scales of the spectral estimate. Any large Haar coefficients are indicative of nonstationarity [36]. These coefficients are thus used to estimate the p-values and test statistics and then test the null hypothesis of stationarity with the confidence level of 5%.

4.1.2. Variance and Length Fractal Dimensions

There is many fractal dimension definitions, each producing inherently different values for multifractals, and the same values for monofractals only [37]. The two fractal dimensions which are used in this thesis are variance fractal dimension and length fractal dimension. Fractal dimensions present a way of measuring the complexity of an object, in this case, a time series, and highlight explicit features that are useful for understanding and marking an occurrence of a fault. These measures have the advantage of being robust and precise. However, they can only be used on a stationary signal.

Variance and length fractal dimensions are measures for the complexity of a self-affine object and for a time series they are between 1 (for a line) and 2 (for a space-filling noise). These dimensions, while inherently different (one relates to the variance of distances for different vel^1 sizes and the other one relates to the distance between samples), both provide useful measures for the complexity and behaviour of an underlying process in a time series. These dimensions are obtained based on finding the power-law relations between variance/length increments and the time increments.

¹ Volume element

Since the signals in this work are segmented into several frames, *Variance Fractal Dimension Trajectory* (VFDT) and *Length Fractal Dimension Trajectory* (LFDT) are used [38]. D_σ and D_L are obtained using VFDT and LFDT as follows:

$$D_L = 1 - S_l \quad (35)$$

$$D_\sigma = 2 - \frac{S_v}{2} \quad (36)$$

The algorithm for calculating the variance and length fractal dimensions is shown in Fig. 4.1. The algorithm for calculating VFDT and LFDT is as shown in Fig. 4.2 [37, 38].

As shown in Fig. 4.1 and Fig. 4.2, for each segment the length and variance fractal dimensions are calculated by finding the power-law between different vel sizes and either length for D_L or variance for D_σ . This process incorporates a monoscale, multiscale and polyscale analysis. At the first step with the vel size of one, only the monoscale analysis is performed. By increasing the vel size based on monadic, triadic, or as is the case here dyadic and averaging between different passes in the same vel size, the analysis is transformed into a multiscale form. Finally, looking at all the scales at the same time and finding the power-law between all scales corresponds to a polyscale analysis which is used in this work for investigating the signals.

4.1.3. Wavelet Coefficients

Wavelet detail and approximation coefficients, as described in Chapter 3, are obtained for different segments of data in different scales. In this sense, WT is a multiscale measure. Thus, using these coefficients could prove effective in picking up behavioural change indicators in the signals. While approximation coefficients are representative of the signal in coarser resolutions, as the level of decomposition increases, the detail coefficients are high-frequency components that exist due to different processes and occurrences in the system. Therefore, detail coefficients have high potentials for presenting indicators of behavioural changes due to faults. Consequently, the detail coefficients are used in this work as the third measure for fault detection.

VFD/LFD MODULE
<p>1. Acquire inputs:</p> <p>1.3. The length of frame N_f;</p> <p>1.2. Array of samples $YY[0: N_f - 1]$;</p>
<p>2. Initialize variables:</p> <p>2.1. Set the vel size m to 1;</p> <p>2.2. Set the summation variables, sum_m, sum_v and sum_l, to 0;</p> <p>2.3. Set the time index, n, to 0;</p> <p>2.4. Set the variance variable, var, to 0;</p> <p>2.5. Set the length variable, len, to 0;</p> <p>2.6. Set the mean variable, μ_d, to 0;</p> <p>2.7. Compute the desired maximum number of coverings, K;</p> <p>2.8. Empty the arrays V, L and M to store $\log(var)$, $\log(len)$ and $\log(m)$;</p> <p>2.9. Set the number of vels for the first vel size as, D.</p>
<p>3. Compute length and variance for kth covering:</p> <p>3.1. Obtain mean: While the value of n is less than $(N_f - m)$, perform Steps 3.1 and 3.2. Otherwise, continue at 3.5.</p> <p>3.1.1. Calculate the increment d of the signal values from time n to time $n + m$;</p> <p>3.1.2. Add d to the sum_m variable;</p> <p>3.1.2. Increment n by m.</p>
<p>3.2. Reinitialize variables:</p> <p>3.2.1. Set the vel size m to 1;</p> <p>3.2.2. Set the time index, n, to 0;</p> <p>3.2.3. Set the μ_d variable to sum_m/D.</p>
<p>3.3. Obtain variance and length:</p> <p>3.3.1. recalculate the increment d of the signal values from time n to time $n + m$;</p> <p>3.3.2. Add the absolute value of d to the sum_l variable;</p> <p>3.3.2. Add the square value of $[d - \mu_d]$ to the sum_v variable;</p> <p>3.1.2. Increment n by m.</p>
<p>3.4. Compute len as sum_l/D;</p> <p>3.5. Compute var as sum_v/D;</p> <p>3.6. Obtain the logarithms, $\log(var)$, $\log(len)$ and $\log(m)$ and assign them to arrays V, L and M.</p>
<p>4. Select the next vel size: Increment the value of m dyadically. While m is less than or equal to K, repeat Step 3.</p>
<p>5. Obtain the power-law relations:</p> <p>5.1. Obtain the slope of the best fit line for arrays L and M. Set this slope as S_l.</p> <p>5.2. Obtain the slope of the best fit line for arrays V and M. Set this slope as S_v.</p>
<p>6. Obtain the dimensions: Use (35) and (36) to obtain the length and variance fractal dimensions, D_L and D_σ, respectively.</p>
<p>7. Return D_L and D_σ.</p>

Fig. 4.1 Algorithm for obtaining variance and length fractal dimensions.

VFDT/LFDT MODULE
1. Acquire inputs: 1.1. The length of time series L_t ; 1.2. Array of samples $Y[0:L_t - 1]$; 1.3. The length of frame N_f ; 1.3. The frame overlap I_f .
2. Initialize variables: 2.1. Empty the array of $Y[0:L_t - 1]$; 2.2. Set $fstart = 0$, establishing the starting sample on a frame; 2.3. Set $counter = 0$, establishing the count of number of dimensions obtained; 2.4. Empty the arrays $vfdt[0:L_t - N_f - 1]$ and $lfdt[0:L_t - N_f - 1]$.
3. Compute a point on each of the trajectories using VFD/LFD MODULE: While the value of $fstart$ is less than $(L_t - N_f)$, perform steps 3.1 to 3.6. Otherwise, continue at step 4. 3.1. Transfer a current frame of samples, $Y[fstart:fstart + N_f - 1]$, into $YY[0:N_f - 1]$; 3.2. Transfer the size of frame N_f and the frame array $YY[0:N_f - 1]$ into the VFD/LFD MODULE; 3.3. Execute the VFD/LFD MODULE; 3.4. Load the outputs from the VFD/LFD MODULE to the array elements $vfdt[counter]$ and $lfdt[counter]$; 3.5. Increment $fstart = fstart + I_f$; 3.5. Increment $counter = counter + 1$.
4. Return $fdt[counter]$ and $lfdt[counter]$.

Fig. 4.2 Algorithm for obtaining variance and length fractal dimension trajectories.

4.2. Fault Detection Development in A Multi-fault Environment

In this section, the multiscale (multiresolution analysis using wavelet transform) and polyscale (fractal dimensions) analysis techniques are implemented to detect faults and their type in the behaviour of the EHA system in operating mode and while in non-operating mode.

Both simulation and experiments have been performed and are shown further in the chapter for internal leakage. However, since producing measured bulk modulus change is not possible with the experimental setup, the developed mathematical model based on the experimental setup is used for simulating the bulk modulus change faults. A correlated fractional pink noise with the frequency of 1 kHz has been added to the simulations to study the fault detection's performance.

Note that the analysis during the non-operating mode is only effective just after the input has been set to zero and the signal has passed the transient response phase. In this period

the residual oscillations in the system are well above the noise level which permits the algorithm to detect the different behaviours corresponding to different conditions. After the oscillations have fallen below the noise level (signal to noise ratio of smaller than 1), the algorithm is not applicable.

4.2.1. Fault Detection in Non-operating Mode

Three simulations have been performed in non-operating mode, with zero input, to examine the behaviour of the EHA system in different conditions: a) a healthy system with no internal leakage and 687 MPa bulk modulus, b) a system with a high bulk modulus of 1500 MPa and no internal leakage, c) a system with severe internal leakage $K_l = 6 \times 10^{-7} (\text{m}^3 / \sqrt{\text{Pa}} \cdot \text{s})$ and 687 MPa bulk modulus (healthy value). Before proceeding to the fractal dimensions for the system, the pressure signal's stationarity is tested in each frame. The results of the WST for the stationarity of P_A with the three conditions are shown in Fig. 4.3. The results indicate that the frames are appropriately chosen based on the behaviour of the system.

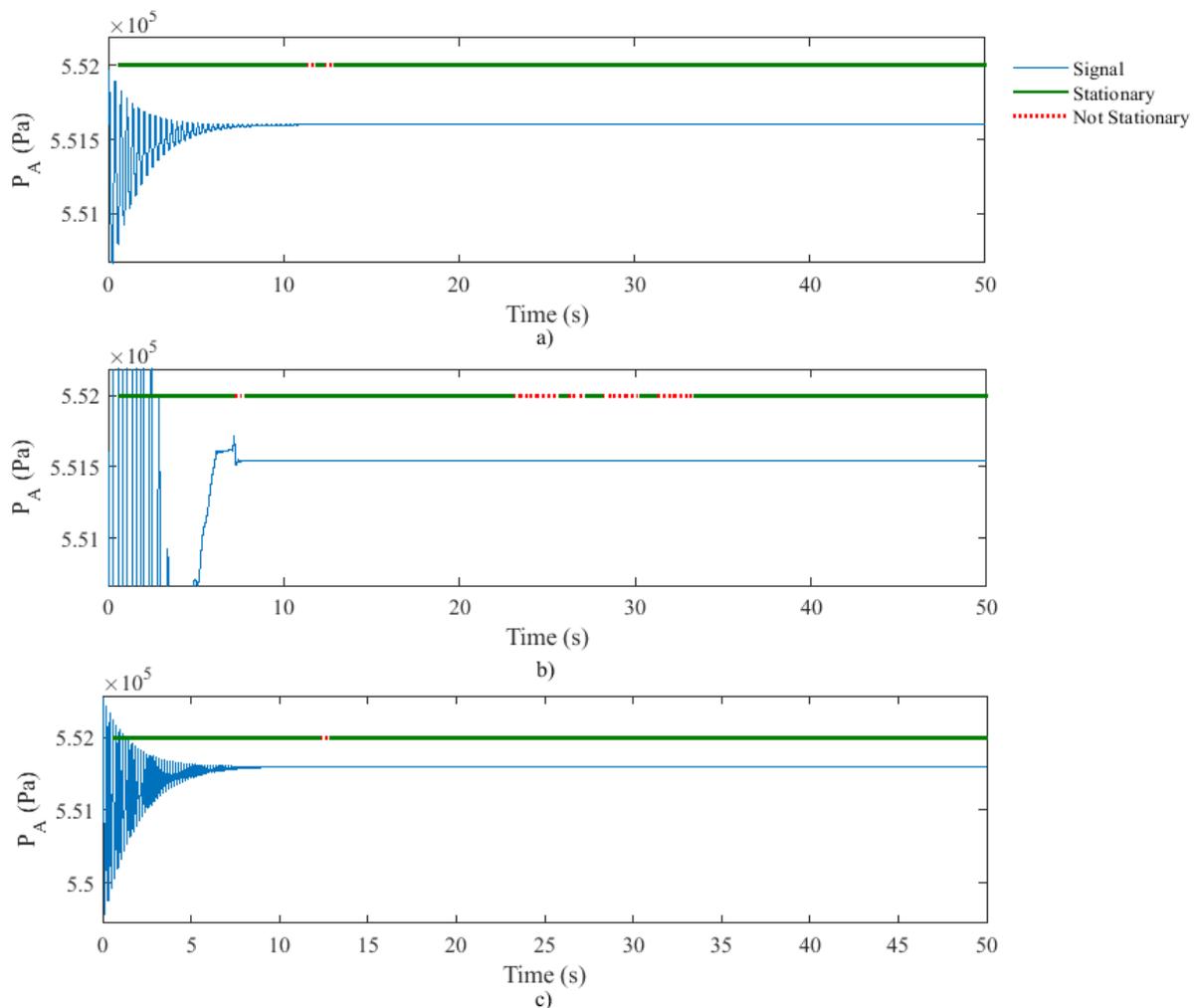


Fig. 4.3 Stationarity test for pressure signals, P_A , for a) healthy system, b) system with internal leakage, and c) system with bulk modulus change in non-operating mode, simulation.

For these simulations, the fractal dimensions are calculated for the pressure signal, P_A , and the frame size for fractal dimension trajectory calculations is considered 1024 with 50% of overlapping. This overlapping helps enhance and bring out the features that are important for the analysis. The results of the fractal analysis simulations are shown in Fig. 4.4.

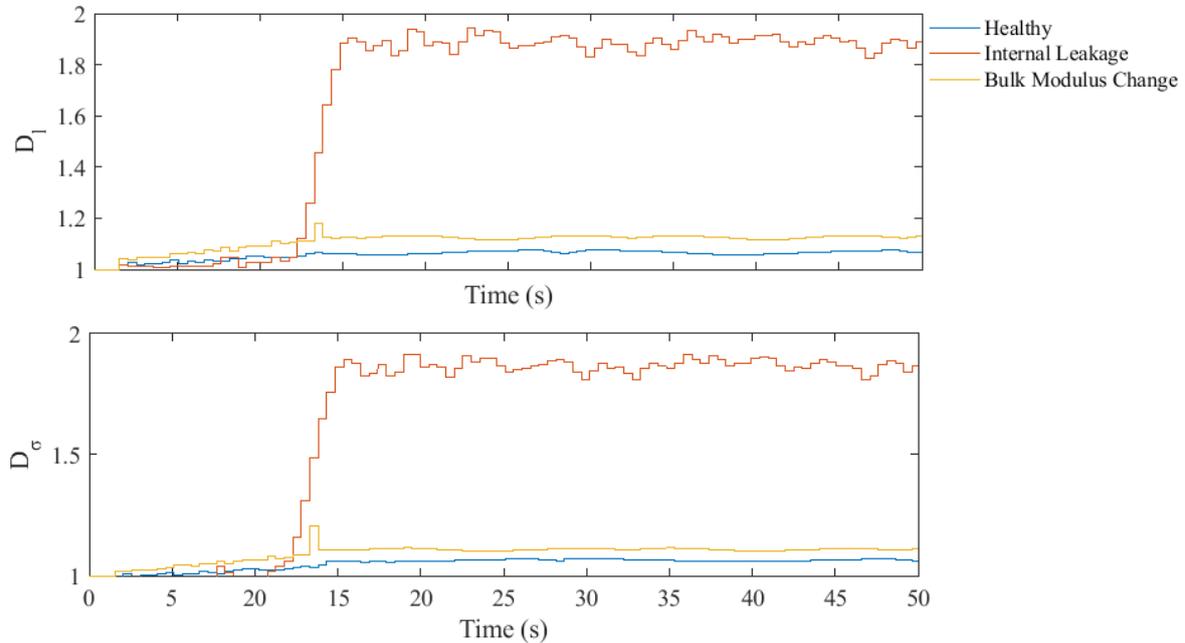


Fig. 4.4 Fractal dimension comparison for a healthy and faulty EHA in non-operating mode pertaining to Fig. 4.3.

As shown in Fig. 4.3, for the first 16 seconds system is in a transient response during which it has fluctuation with a limited range. This is due to the initial condition of the system, $P_A = 0.55$ MPa. As expected, even in the non-operating mode, internal leakage has the most effect on the behaviour of the system as seen by the initial range of oscillations (0.12 MPa for the system with internal leakage, 0.003 MPa for the system with bulk modulus change and 0.0013 MPa for the healthy system) in the pressure signal during the transient response in Fig. 4.3.

During the steady-state phase, as shown in Fig. 4.4, the length and variance fractal dimensions of the pressure signal are different for the healthy system and system with internal leakage and bulk modulus change faults. Thus, the existence of faults could be inferred based on these differences of fractal dimensions in non-operating mode. Furthermore, while the increase in the average value of the fractal dimension for bulk modulus is approximately +0.05 for variance dimension and +0.06 for length dimension, the change in the dimensions with internal leakage fault is +0.80 for variance dimension and +0.82 for length dimension. These differences in the values of the fractal dimensions mean that in addition to the existence of

faults, the type of fault, whether internal leakage or bulk modulus change, is detectable using the fractal measures in the non-operating mode as well.

The differences between the average fractal dimensions values for a faulty system and a healthy system are shown in Table 4.1.

Table 4.1 Variance and length fractal dimension averages for faulty and healthy Conditions.

System Condition	\bar{D}_σ	$\bar{D}_\sigma - \bar{D}_{\sigma,healthy}$	\bar{D}_L	$\bar{D}_L - \bar{D}_{L,healthy}$
Healthy	1.0642	0	1.0680	0
Internal Leakage	1.8683	0.8041	1.8880	0.8400
Bulk Modulus Change	1.1084	0.0442	1.1252	0.0572

Fig. 4.5 describes the wavelet detail coefficients of the pressure signal for 8 levels for the system in healthy condition.

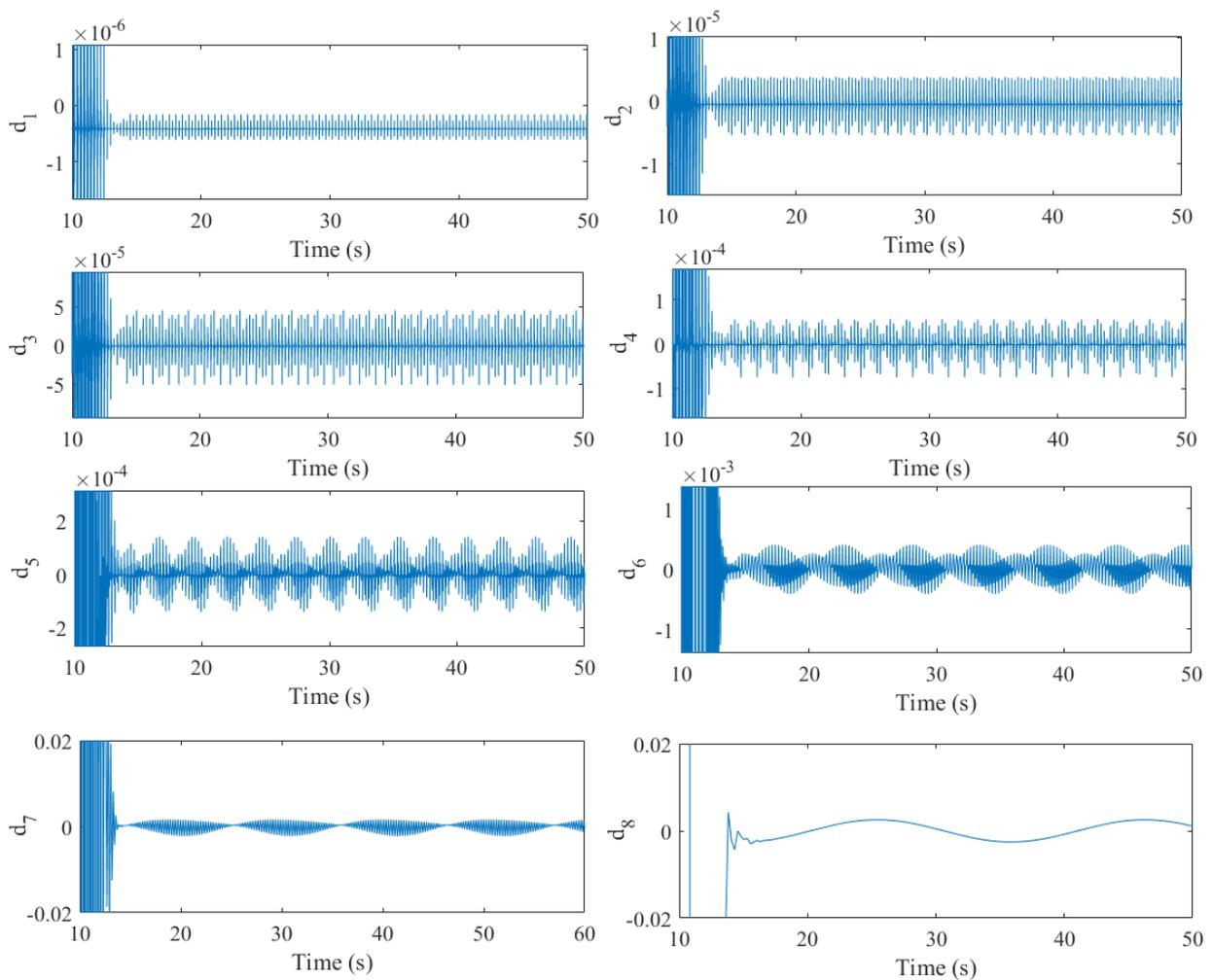


Fig. 4.5 Wavelet detail coefficients of pressure signals, P_A , for a EHA in non-operating mode for a healthy system pertaining to Fig. 4.3.

Fig. 4.6 describes the wavelet detail coefficients of the pressure signal for 8 levels for the system with internal leakage fault.

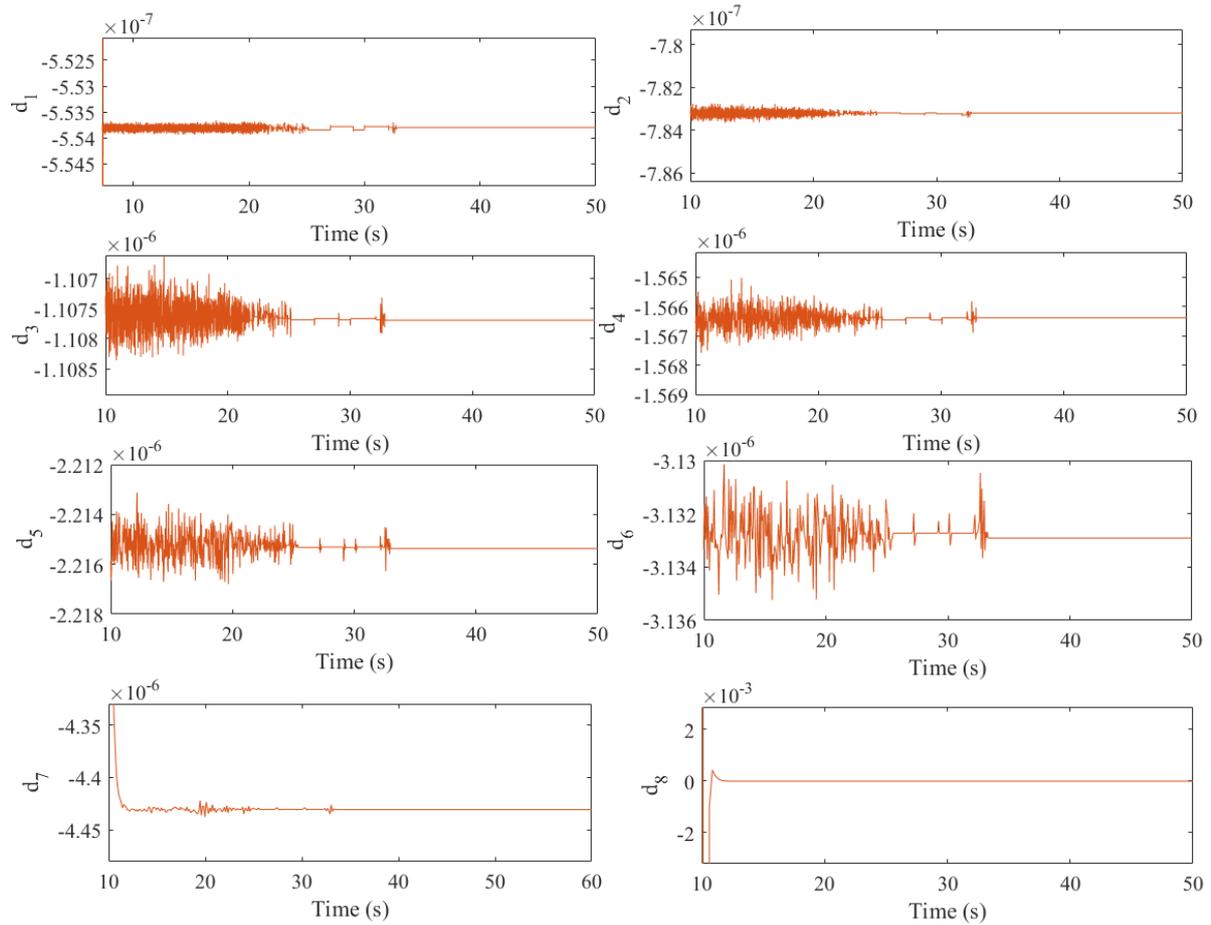


Fig. 4.6 Wavelet detail coefficients of pressure signals, P_A , for a EHA in non-operating mode for a system with internal leakage pertaining to Fig. 4.3.

Fig. 4.7 describes the wavelet detail coefficients of the pressure signal for 8 levels for the system with bulk modulus change.

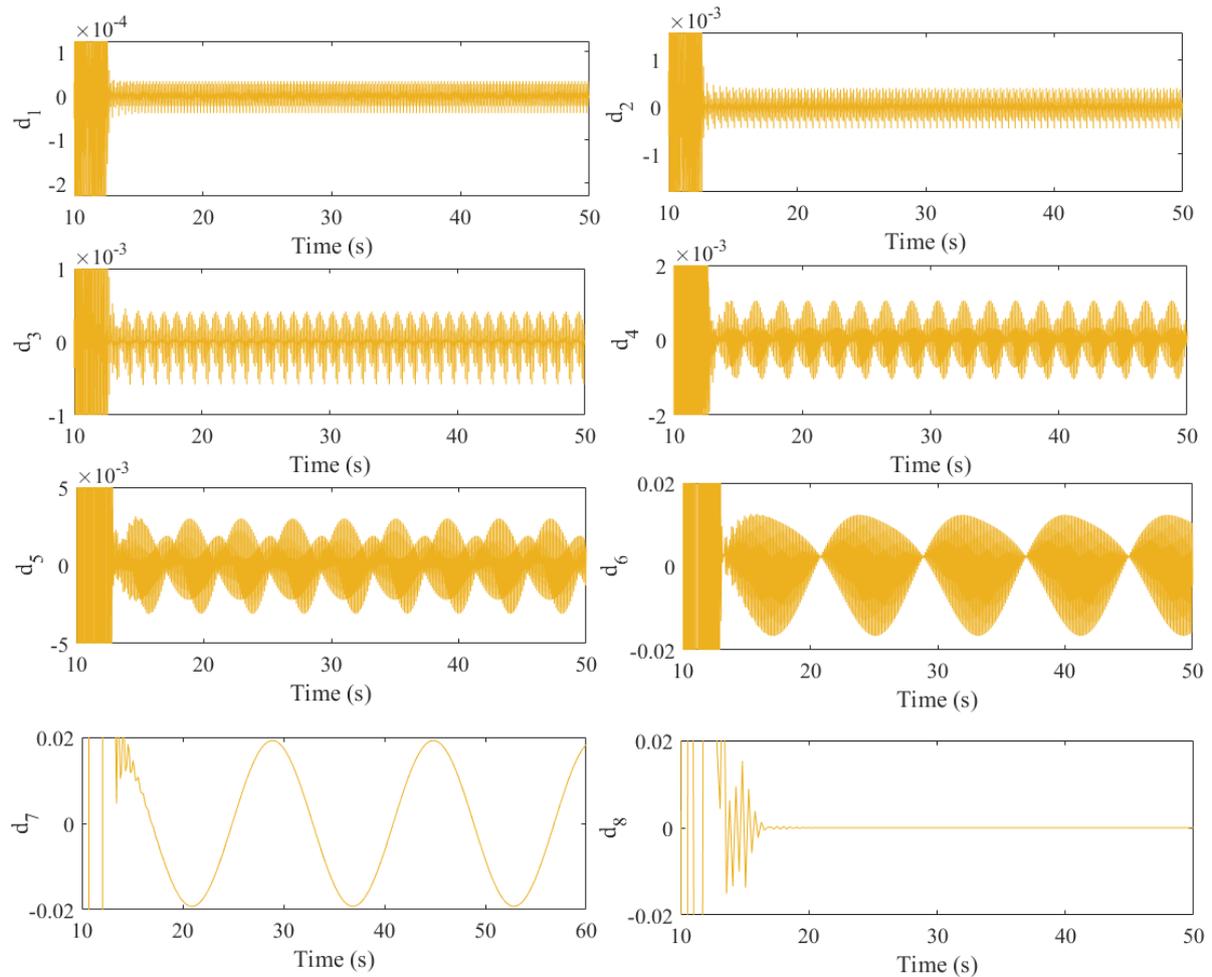


Fig. 4.7 Wavelet detail coefficients of pressure signals, P_A , for a EHA in non-operating mode for a system with bulk modulus change pertaining to Fig. 4.3.

Fig. 4.8, shows a closer look at the coefficients of the three shown experiments in healthy, internal leakage and bulk modulus change conditions. As shown in the detail coefficient plots, the first 7 level coefficients have detectably different magnitudes with the highest magnitudes for the coefficients corresponding to the system with bulk modulus change, as an instance approximately 0.04 for level 7, then the coefficients corresponding to the healthy system, as an instance approximately 0.0036 for level 7, and finally the lowest magnitudes with coefficients corresponding to the system with internal leakage, as an instance approximately 4×10^{-6} for level 7.

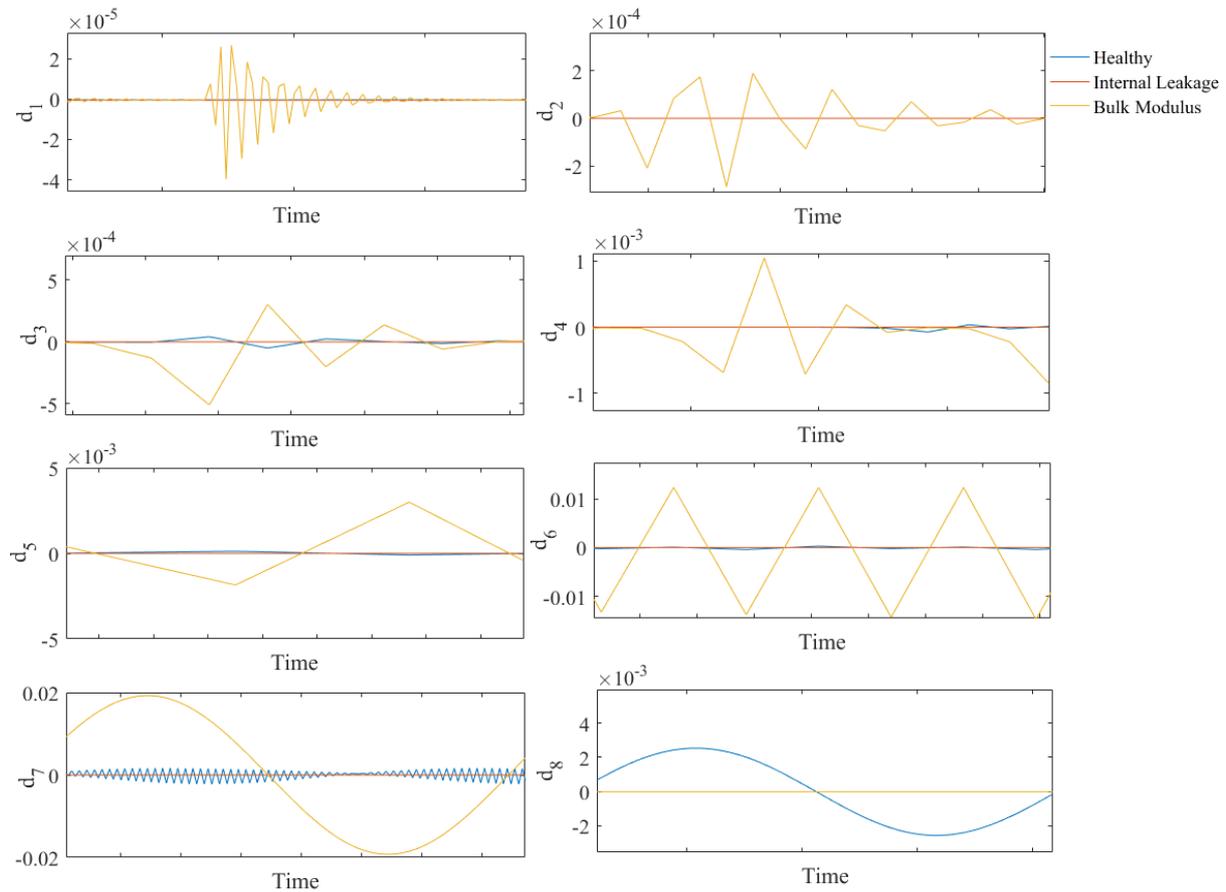


Fig. 4.8 Close-up view of wavelet detail coefficients of pressure signals, P_A , for a healthy and faulty EHA in non-operating mode pertaining to Fig. 4.3.

4.2.2. Fault Detection in Operating Mode

In this section, the results of the simulation for operation mode are presented. Since the pressure signals might be affected due to the closed-loop configuration of the system either by a controller or via operator decisions, the polyscale and multiscale measures are obtained for the jerk signal, *jerk*. Note that In this thesis jerk is always refers to motion jerk.

Three simulations have been performed in operating mode, all of them with the same input: a) a healthy system with no internal leakage and 687 MPa bulk modulus, b) a system with a high bulk modulus of 1500 MPa and no internal leakage, c) a system with severe internal leakage $K_l = 6 \times 10^{-7} (\text{m}^3/\sqrt{\text{Pa}} \cdot \text{s})$ and 687 MPa bulk modulus (healthy value). Before proceeding to the fractal dimensions for the system, the jerk signal's stationarity is tested in each frame. The results of the WST for the stationarity of the jerk signal with the three aforementioned conditions along with the velocity of the system are shown in Fig. 4.9. The results indicate that the frames are appropriately chosen based on the behaviour of the system.

For these simulations, the frame size for fractal dimension trajectory calculations is considered 1024 with 50% of overlapping. This overlapping helps enhance and bring out the features that are important for the analysis. The results of the fractal analysis simulations are shown in Fig. 4.10.

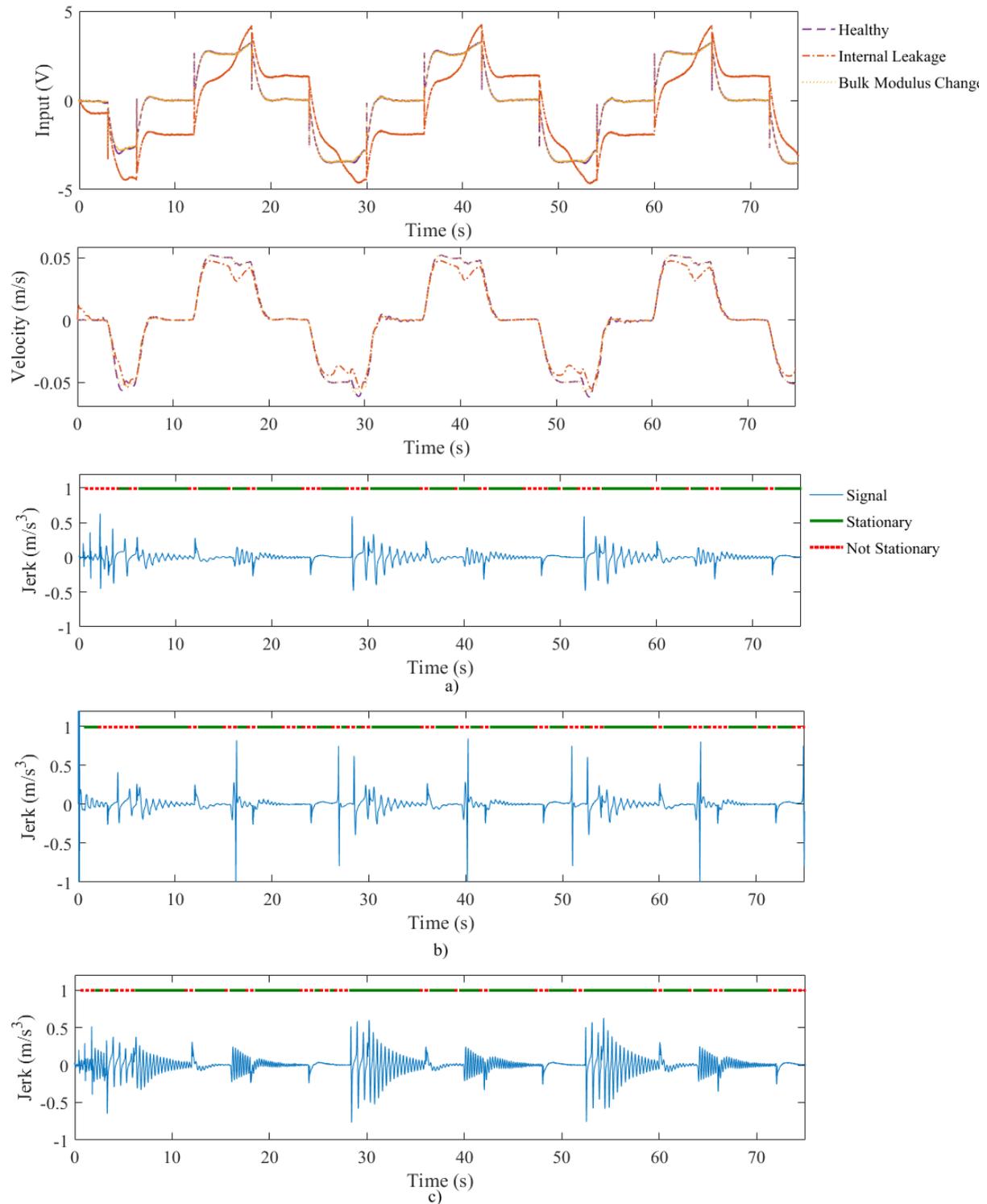


Fig. 4.9 Stationarity test for jerk signals, *jerk*, for a) healthy system, b) system with internal leakage, and c) system with bulk modules change in operating mode, simulation.

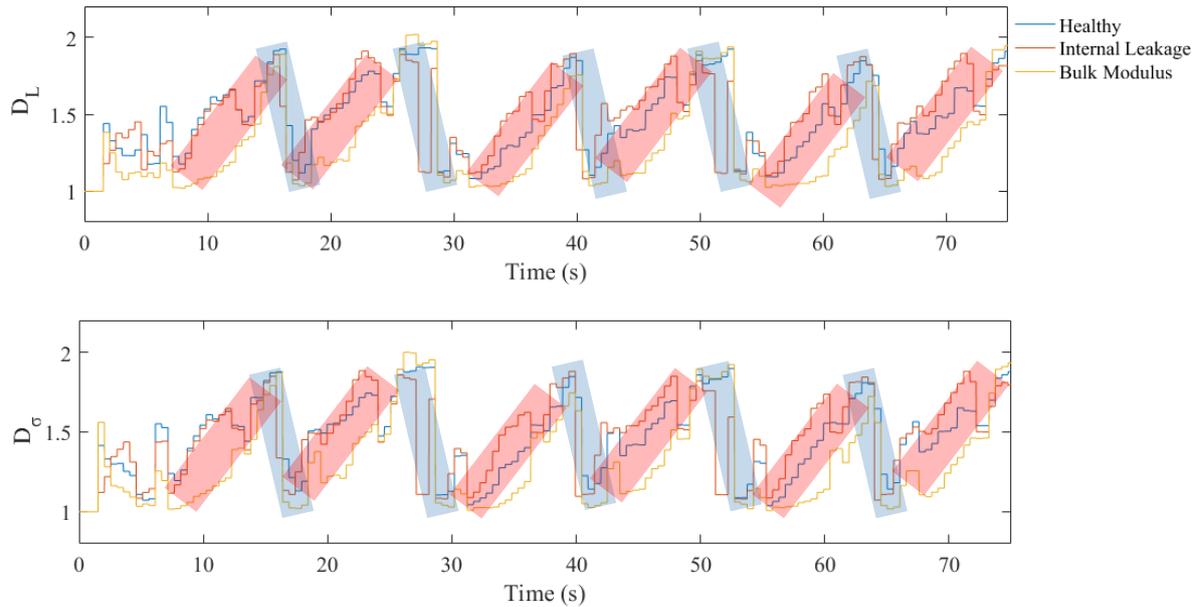


Fig. 4.10 Fractal dimension comparison for a healthy and faulty EHA in operating mode pertaining to Fig. 4.9.

As shown in Fig. 4.10, while the differences between fractal dimensions for a healthy and a faulty system seem insignificant at first, they are consistent and easily recognizable. These patterns of differences exist during the operations and are highlighted with a red film. In the shown parallel ramps of both variance and length fractal dimensions there exist an offset of approximately 0.09 to 0.36 between the faulty and healthy systems: a) the fractal dimensions for internal leakage have the highest value among the three conditions, as an instance $D_L = 1.52$ and $D_\sigma = 1.537$ for $t = 35(s)$, b) the healthy system has the medium values, as an instance $D_L = 1.29$ and $D_\sigma = 1.315$ for $t = 35(s)$, c) and the system with bulk modulus change has the lowest values among the three, as an instance $D_L = 1.095$ and $D_\sigma = 1.117$ for $t = 35(s)$. Furthermore, on the negative slope ramps (highlighted with blue films) which occurs just after the peak of the ramps the signal with internal leakage takes the lowest values closer to 1 (the low limit of fractal dimension) while the system with bulk modulus change and the healthy system have approximately the same values for their dimensions. The different behaviour of upwards and downwards ramps are due to the quadrant changes and different solenoid activations. The quadrant changes, first of all, cause a sudden flow change from and or to the auxiliary circuit and second a spike in the velocity and its derivatives due to sudden switching discontinuity in the system. For instance, at $t=28s$ a switch happens to the third and the first quadrant. Thus, a downward ramp starts immediately followed by an upward ramp. At

t=40s the next solenoid switching happens to reach quadrants 2 and 3 which in turn initiates another two successive downwards-upwards ramps.

Fig. 4.11 describes the wavelet detail coefficients of the signal in healthy condition for five levels.

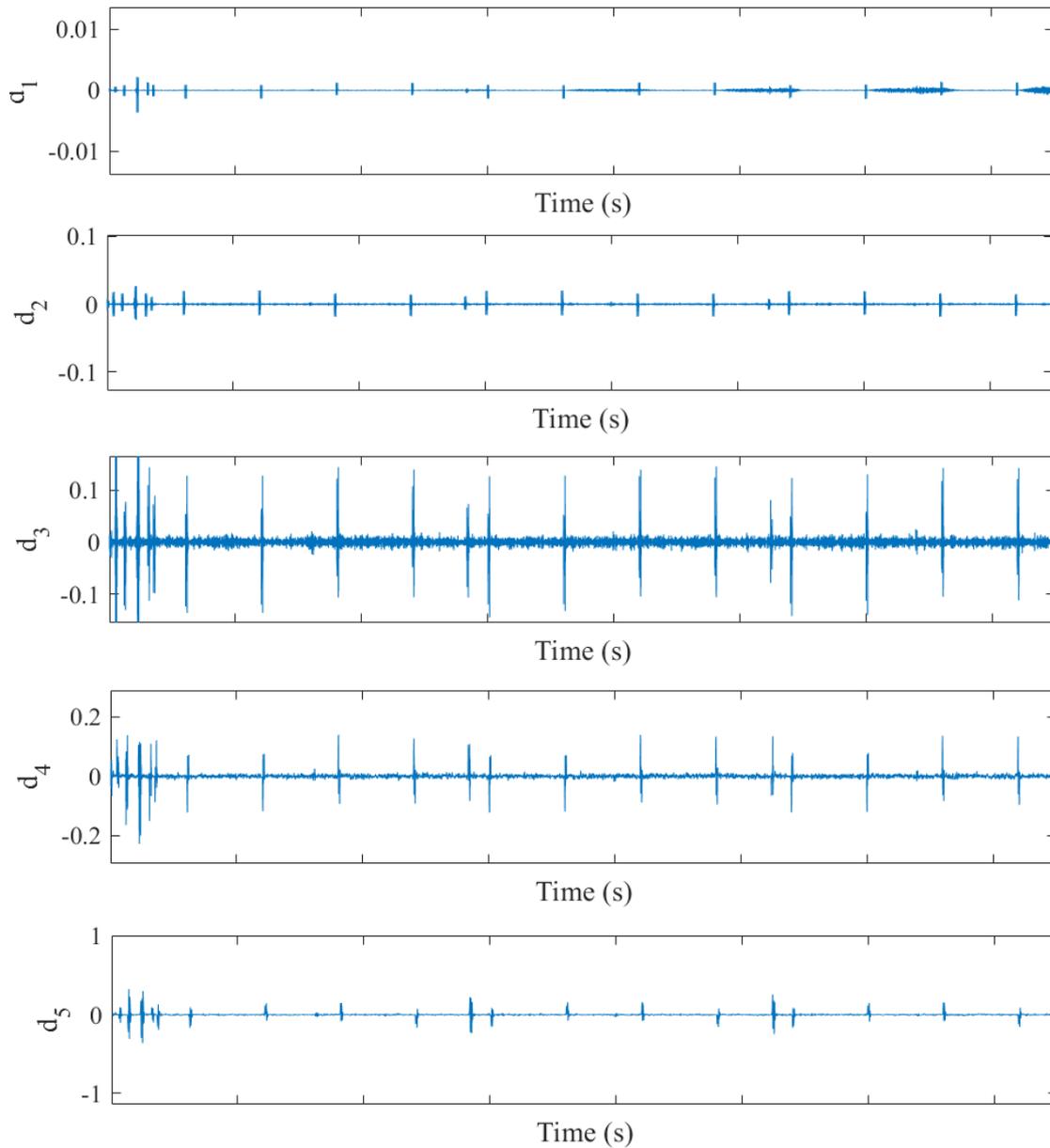


Fig. 4.11 Wavelet detail coefficients of jerk signal, *jerk*, for a healthy EHA in operating mode pertaining to Fig. 4.9.

Fig. 4.12 describes the wavelet detail coefficients of the jerk signal of the system with internal leakage for five levels.

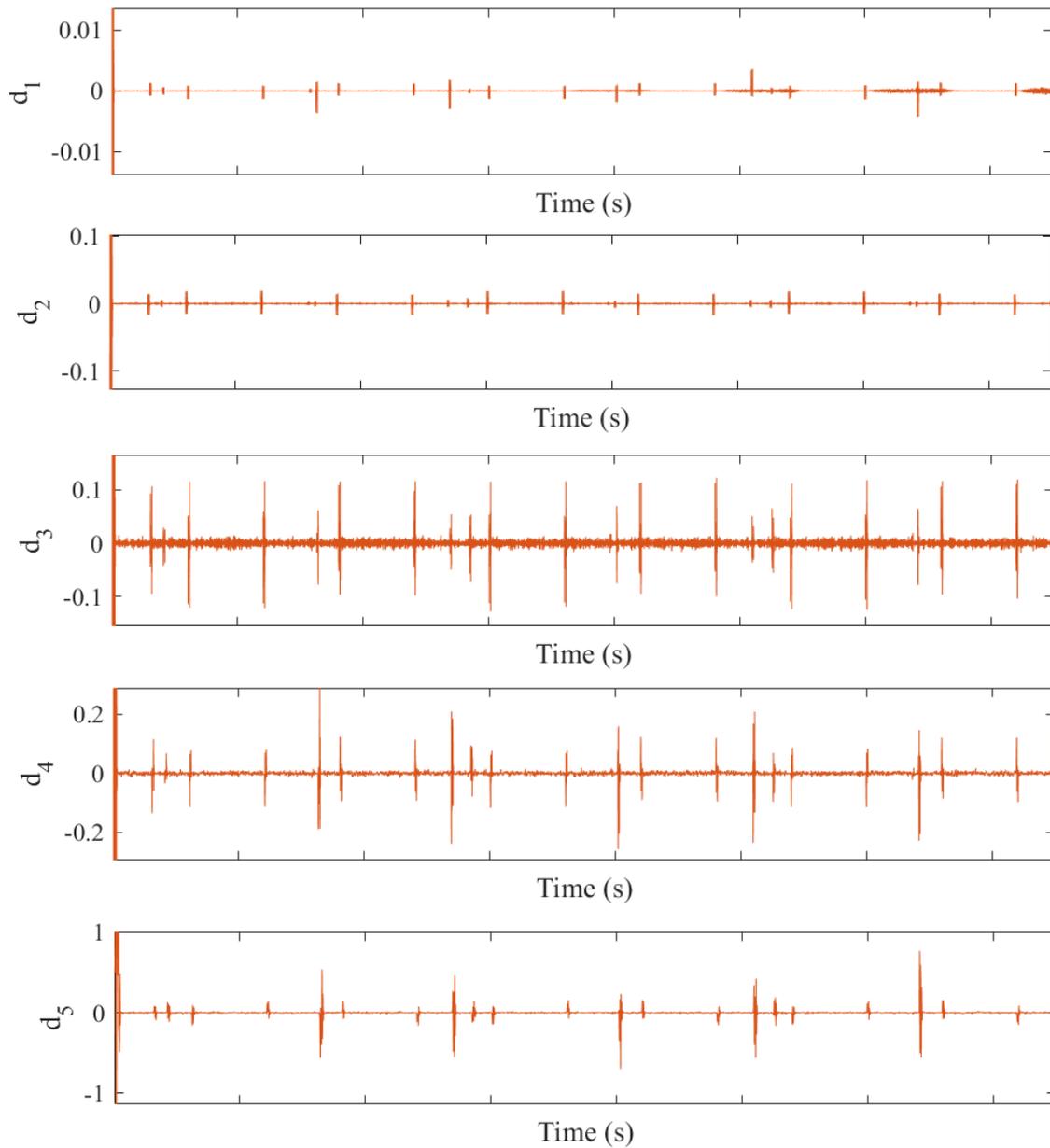


Fig. 4.12 Wavelet detail coefficients of jerk signal, *jerk*, for an EHA with internal leakage fault in operating mode pertaining to Fig. 4.9.

Fig. 4.13 describes the wavelet detail coefficients of the jerk signal in healthy condition for five levels.

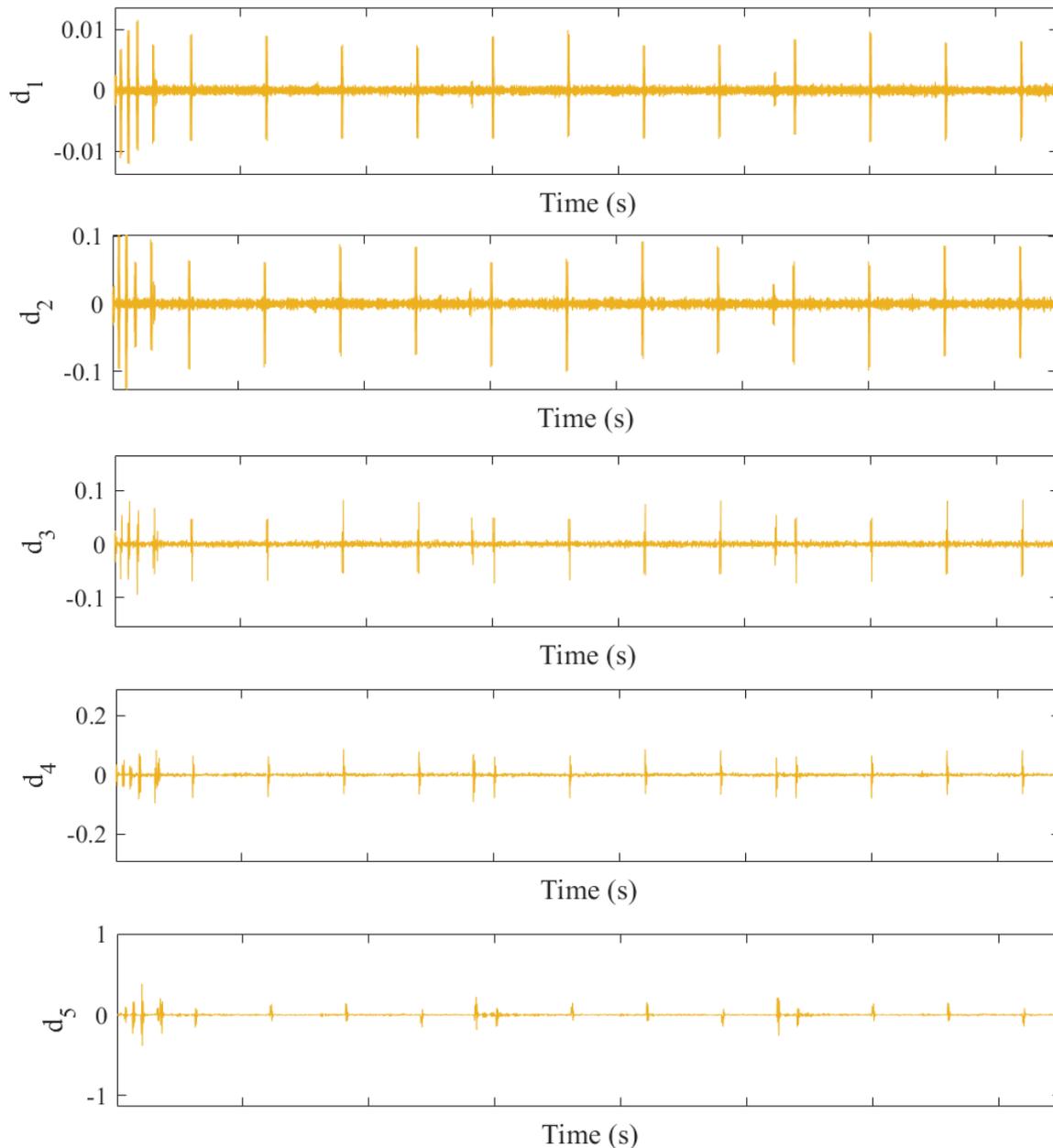


Fig. 4.13 Wavelet detail coefficients of jerk signal, *jerk*, for an EHA with bulk modulus change fault in operating mode pertaining to Fig. 4.9.

By comparing the detail coefficients, the following conclusions can be drawn:

Levels 1 and 2 detail coefficients of the healthy system and system with internal leakage have similar behaviour and are in a close range with respect to their amplitudes but have detectable differences in their variance and amplitude compared to the system with bulk modulus change. **Level 3 detail coefficients** for all three conditions are within a close range and not separable and thus not appropriate for fault detection purposes. **Level 4 detail coefficients** have bigger differences than level 3, but smaller differences than level 1. Thus, while they can be used in combination with another level, on their own they are harder to use

in a multi-fault environment. **Level 5 detail coefficients** of the healthy system and system with bulk modulus change have similar behaviours and are in a close range with respect to their amplitudes but have detectable differences in their variance and amplitude compared to the system with internal leakage.

Thus, it can be inferred that to successfully detect both internal leakage and bulk modulus change faults, a combination of level 1 (or 2) and level 4 (or 5) detail coefficients could be utilized. Consequently, levels 1 and 4 detail coefficients have been chosen for fault detection. This could be further investigated by obtaining three statistical moments (mean, variance, and autocorrelation with $lag = 1$) and the maximum values of the detail coefficients. These moments and maximums are computed for frames of 128 size on the wavelet detail coefficients of levels 1 and 4 which are acquired from the jerk signal and are shown in Fig. 4.14 and Fig. 4.15, respectively.

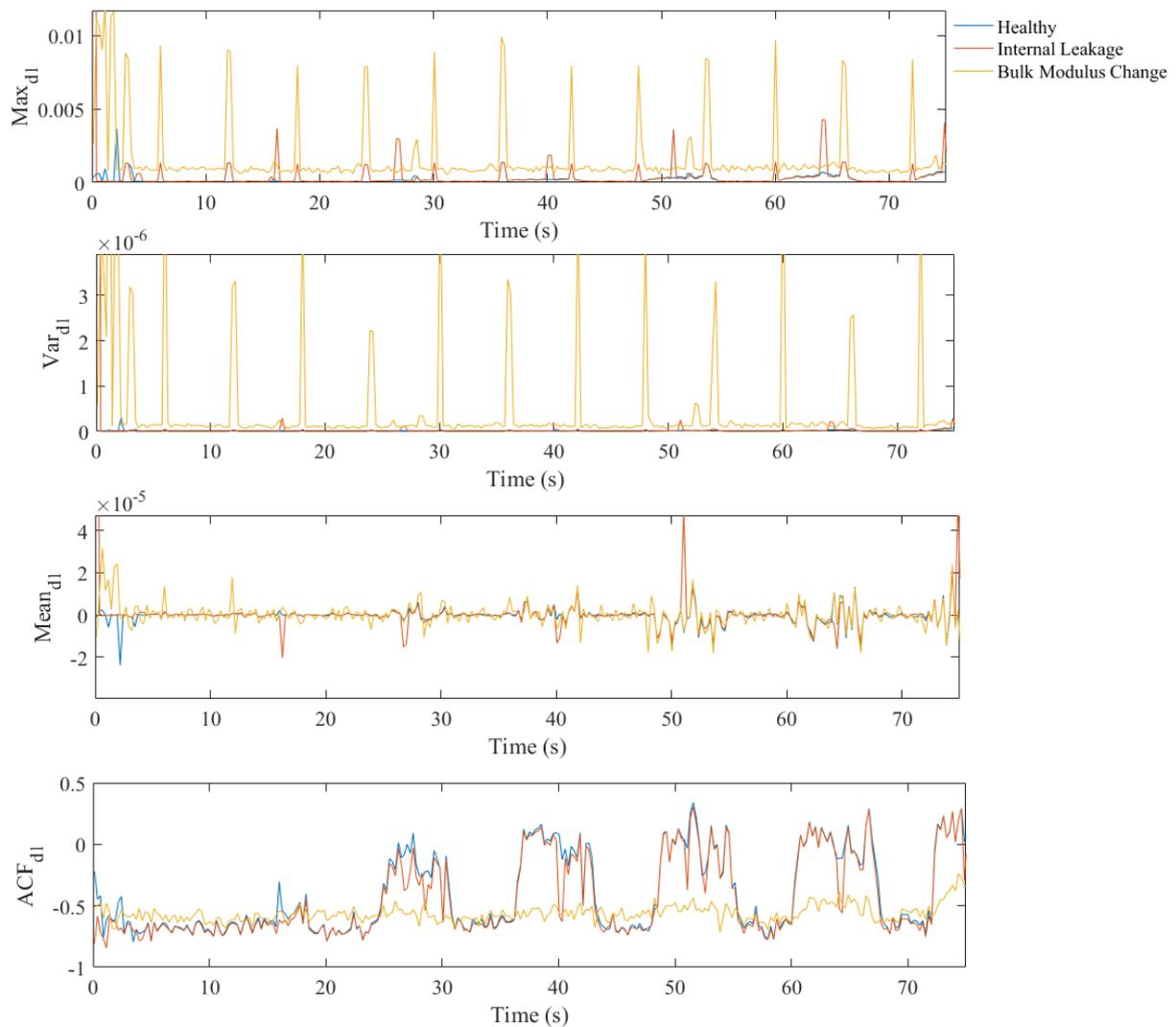


Fig. 4.14 Level 1 wavelet detail coefficients' statistical characteristics for a healthy and faulty EHA in operating mode pertaining to Fig. 4.9.

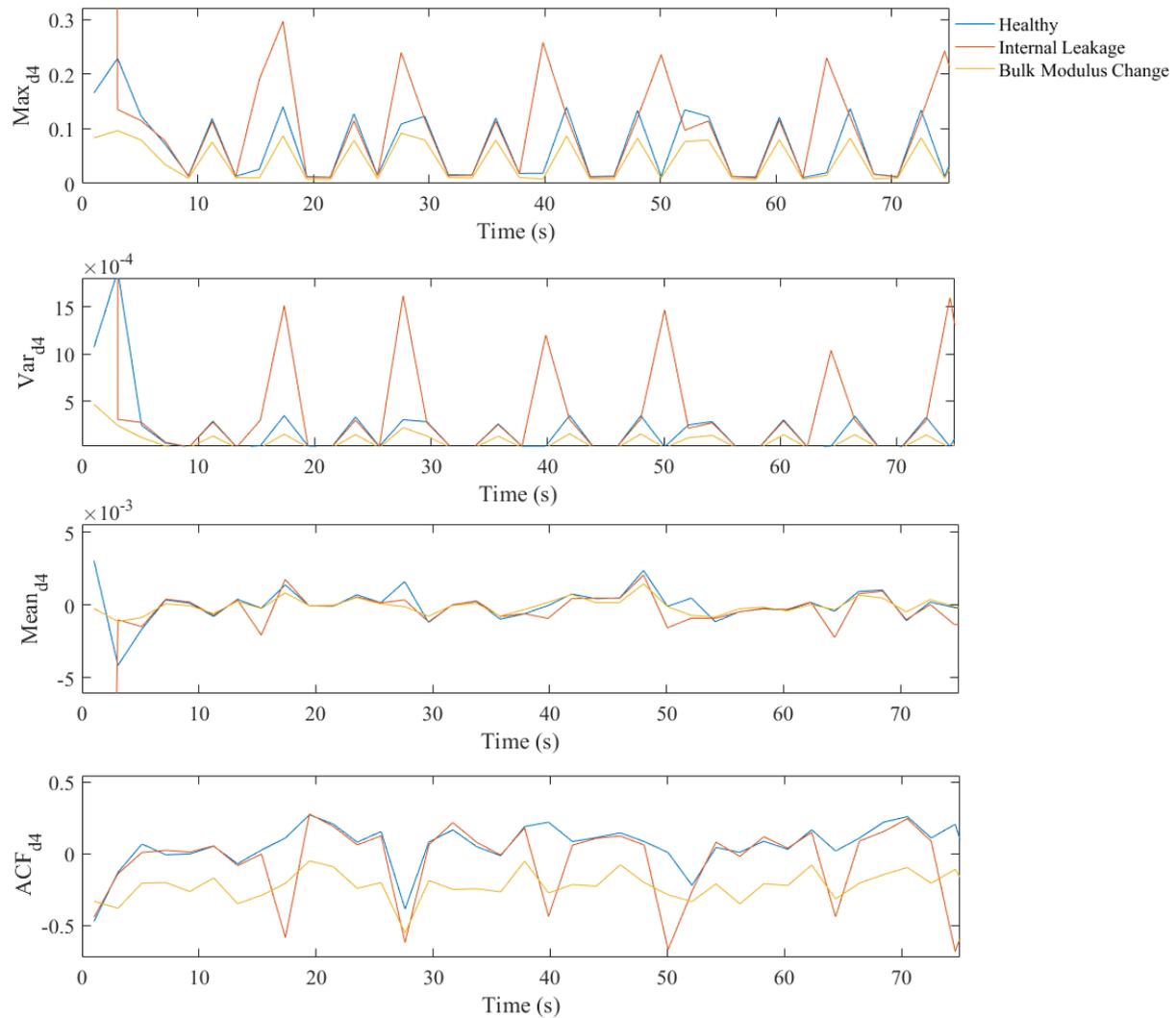


Fig. 4.15 Level 4 wavelet detail coefficients' statistical characteristics for a healthy and faulty EHA in operating mode pertaining to Fig. 4.9.

As shown in Fig. 4.14 and Fig. 4.15, by comparing the maximum or variance of level 1 detail coefficients, bulk modulus change could be detected in which case maximum values are larger than 0.00085 and variance values are larger than 1.3×10^{-7} . Internal leakage could be detected by comparing the maximum or variance of level 4 detail coefficients, where maximum values are larger than 0.13 and variance values are larger than 3.5×10^{-4} . Furthermore, while the mean value of coefficients is not useful, the autocorrelation function could be considered as another measure for fault detection with a periodic behaviour since for levels 1 and 4 the autocorrelation function of coefficients with bulk modulus change is an average of 0.24 and 0.35 smaller than a healthy system or a system with internal leakage within periods, respectively.

4.3. Sensitivity to Faults with Variable Severity

Based on the analysis performed in Section 4.2, variance and length fractal dimensions' values and wavelet detail coefficients present different values for different conditions of internal leakage and bulk modulus change. In this section, to elaborate on the correlation between the degree to which the faults occur and the fault detection measures, several experiments and simulations have been performed. The signal used for fault detection is the rod-side pressure signal, P_B .

Note that the same behaviours seen in the rod-side pressure signal, P_B , can be seen in cap-side pressure signal, P_A . This is because they are related as was discussed in Chapter 2.

4.3.1. Internal Leakage

Since internal leakage can be artificially introduced in the experimental setup, in this section the experimental results have been used for internal leakage fault detection.

In the experiments, we expect to see the same successive downwards and upwards ramps with internal leakage taking the lead in the upward ramps and the healthy system taking higher values in the downwards ramps, which is the same behaviour observed in the simulations.

To produce internal leakage in the experimental setup, the orifice in the internal leakage module of Fig. 2.4, is opened in four increments: fully closed (healthy system), 1/3 open (small leakage), 1/2 open (medium leakage), and fully open (severe leakage). The results of the experiments are shown in Fig. 4.16. In these experiments, the denoising method with the proposed fuzzy threshold was used. The denoising method contributes to the validity of the fault detection by removing noise from the pressure signals which are used for analysis.

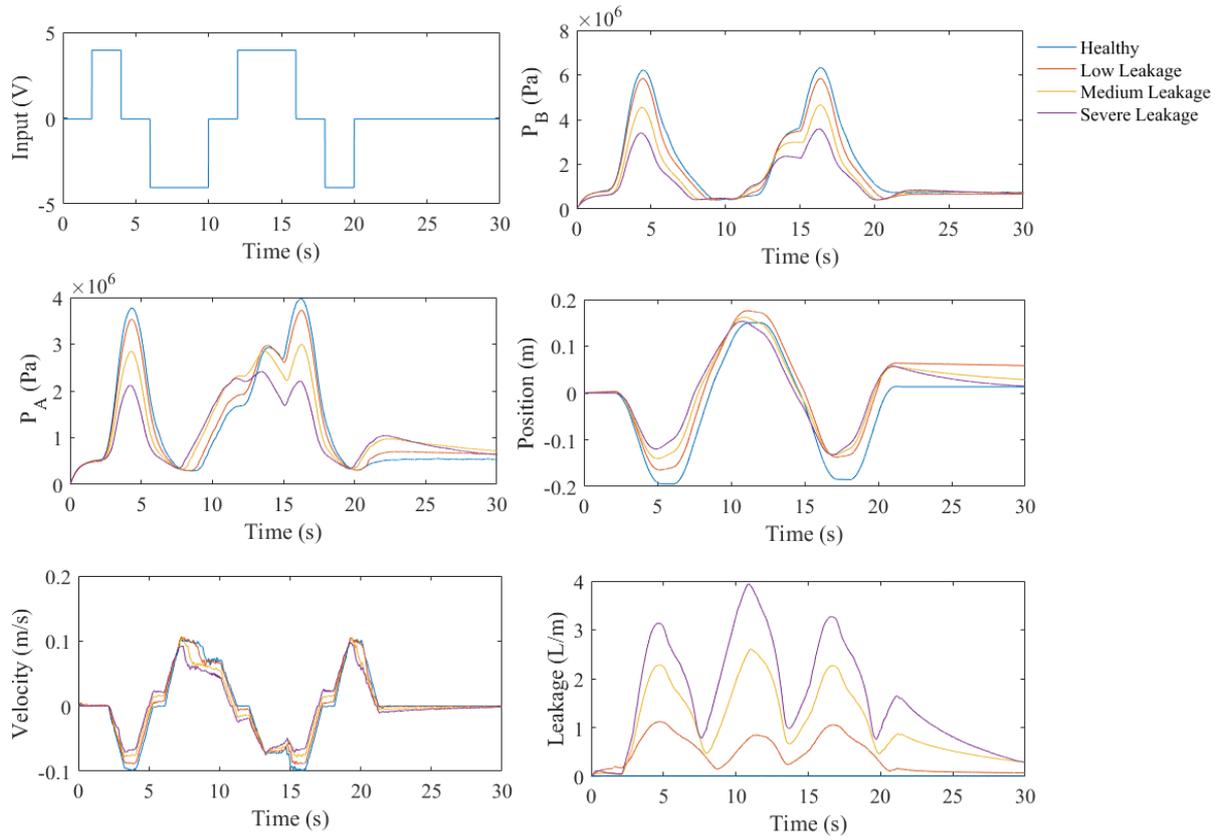


Fig. 4.16 Experimental results for healthy and faulty EHAs with internal leakage in operating mode.

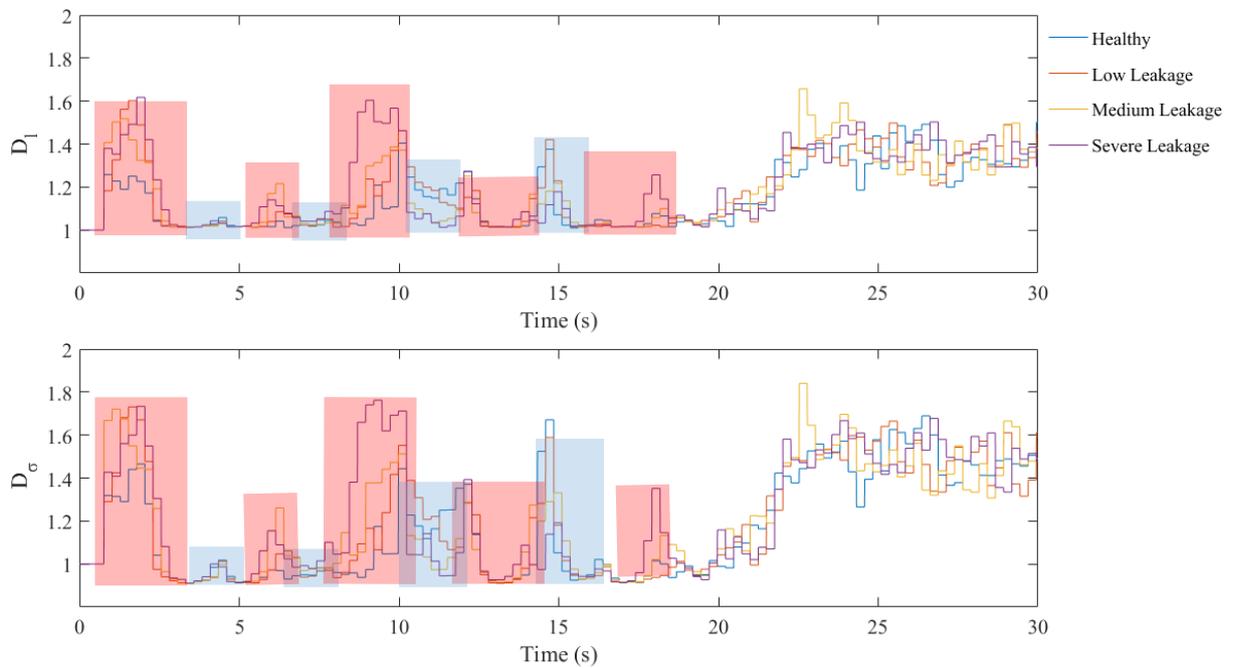


Fig. 4.17 Fractal dimensions for healthy and faulty EHAs with internal leakage in operating mode pertaining to Fig. 4.16.

As shown in Fig. 4.16, similar patterns as in the simulations are observed in the experimental results. These patterns are shown in blue and red films.

The blue films represent the quadrant change points where systems with internal leakage manifests themselves having lower values of fractal dimensions, close to 1. In the instances of the blue film at $t=12s$ and $t=14s$ where quadrants change and the internal leakage is high, the healthy system exhibits higher values of fractal dimension. Thus, there is a difference between the healthy system and the faulty system. However, at $t=4s$ and $t=7s$ where the internal leakage has dropped to around 1L/min, fractal dimensions of the healthy system have values close to 1 as well as the faulty systems.

The red films follow immediately after the blue films and represent the conditions where the system with the internal leakage has higher values of fractal dimensions compared to the healthy system. In fact, uniformly in all the red films, the system with the highest internal leakage has the highest fractal dimension, and the system with no internal leakage has the lowest. In other words, the difference between the fractal dimension values is proportional to the severity of the internal leakage in the system.

After approximately $t=20s$, the input to the system is kept at zero and the system enters the non-operating mode with the pressure signals being stationary. In this period the fractal dimensions fluctuate around $D_v = 1.5$ and $D_l = 1.4$ which indicates that the noise (which is close to pink) became dominant in the signals and the actual pressure is too small to be detectable by the algorithm. However, for the purposes of fault detection in non-operating mode, a more aggressive denoising method with a 50% greater threshold is used on the pressure signals and the associated fractal dimensions are shown in Fig. 4.18.

As shown in Fig. 4.18, the healthy system seems to have the lowest fractal dimension values among the four. Also, the system with severe internal leakage consistently shows higher fractal dimension values. In order to further elaborate on these trajectories, the expected values for different degrees of leakage in the system are described in Table 4.2.

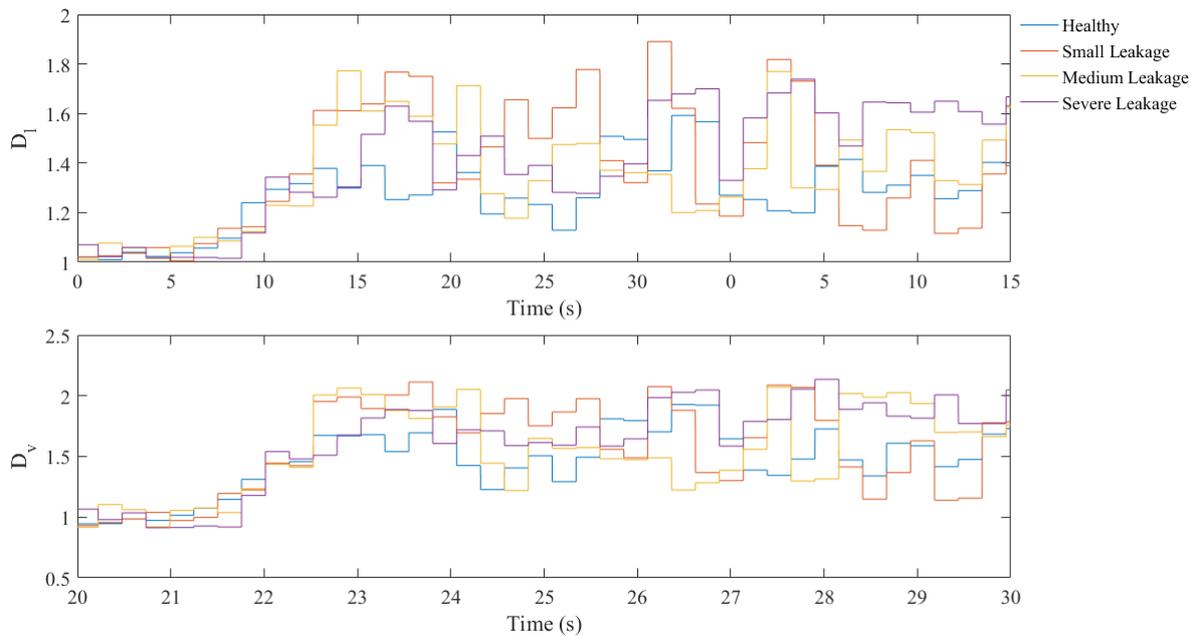


Fig. 4.18 Experimental results for healthy and faulty EHAs with internal leakage in non-operating mode pertaining to Fig. 4.16.

Table 4.2 Mean of fractal dimensions, internal leakage experimental data.

System Condition	\bar{D}_σ		\bar{D}_L	
	epoch	t>20s	epoch	t>20s
Healthy	1.1458	1.4674	1.1177	1.2794
Small internal leakage	1.2033	1.5667	1.1654	1.3819
Medium internal leakage	1.2094	1.5427	1.1648	1.3532
Severe internal leakage	1.2469	1.6189	1.1839	1.4042

As shown, despite their fluctuations, the expected values of the healthy system's fractal dimensions are the lowest among the four not only during the non-operating mode but for the whole epoch as well. This is then followed by the mean values for the small and medium leakage which have approximately the same means, greater than those of a healthy system. Finally, the greatest expected values among the four are seen in the system with severe internal leakage.

In the case of wavelet detail coefficients, the analysis is straightforward which points towards the high potential for this method. The decomposition of the pressure signal, P_B , for four levels to approximation and detail coefficients is performed using Daubechies 8 [11] mother wavelet. The approximation coefficients and the detail coefficients are shown in Fig. 4.19 and Fig. 4.20, respectively.

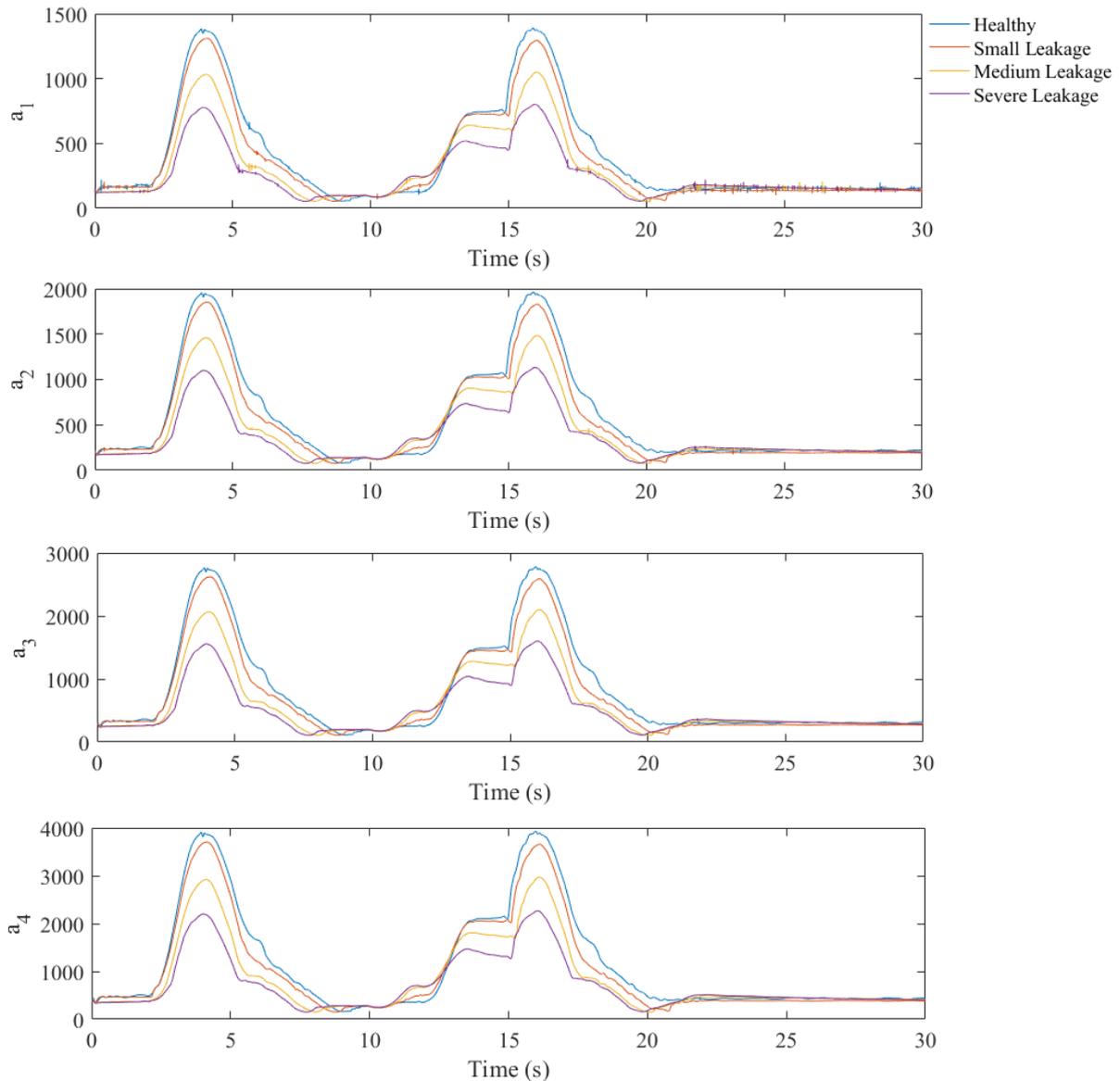


Fig. 4.19 Wavelet approximation coefficients of systems with different internal leakage levels pertaining to Fig. 4.16.

As shown, while the approximation coefficients only present the original signal with coarser resolution, the detail coefficients contain those features and high-frequency components that could operate as a flag for fault detection. Similar to observed behaviour in simulation results, level 4 detail coefficients are appropriate for the purposes of internal leakage fault detection.

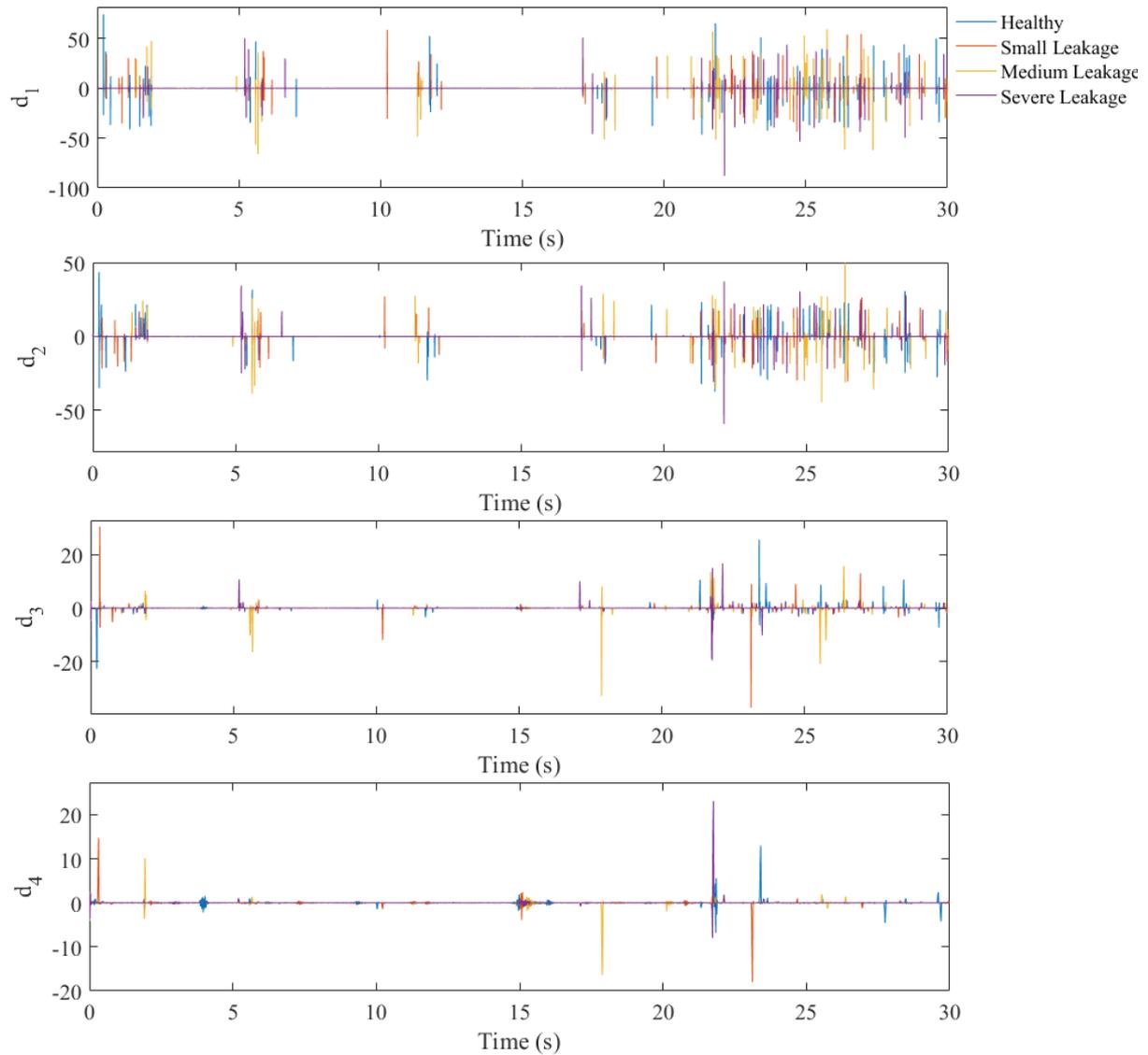


Fig. 4.20 Wavelet detail coefficients of systems with different internal leakage levels pertaining to Fig. 4.16.

The absolute maximum values of the level 4 detail coefficient for the four analyzed conditions are as follows:

$$D_{Severe} > D_{Small} \approx D_{medium} > D_{Healthy} \quad (37)$$

where $D_{Severe} = 23.1$, $D_{Small} = 18.0$, $D_{medium} = 17.6$ and $D_{Healthy} = 12.3$ are the maximum of the detail coefficients for the system with severe internal leakage, medium internal leakage, small internal leakage, and the healthy system, respectively.

Thus, it can be inferred that the wavelet detail coefficients are successful in detecting the fault by comparing the maximum of detail coefficients.

4.3.2. Bulk Modulus Change

Since replicating measurable bulk modulus change was not possible for the experimental setup, the verified simulation model has been used in this section for fault detection analysis.

Four systems with bulk moduli $\beta_{0.5} = 344.5\text{MPa}$, $\beta_1 = 689\text{MPa}$, $\beta_{1.5} = 1033.5\text{MPa}$ and $\beta_2 = 1378\text{MPa}$ are simulated, the results of which are shown in Fig. 4.21. The variance fractal dimension, D_v , and length fractal dimension, D_L , of the rod side pressure signal, P_B , for the four conditions are shown in Fig. 4.22.

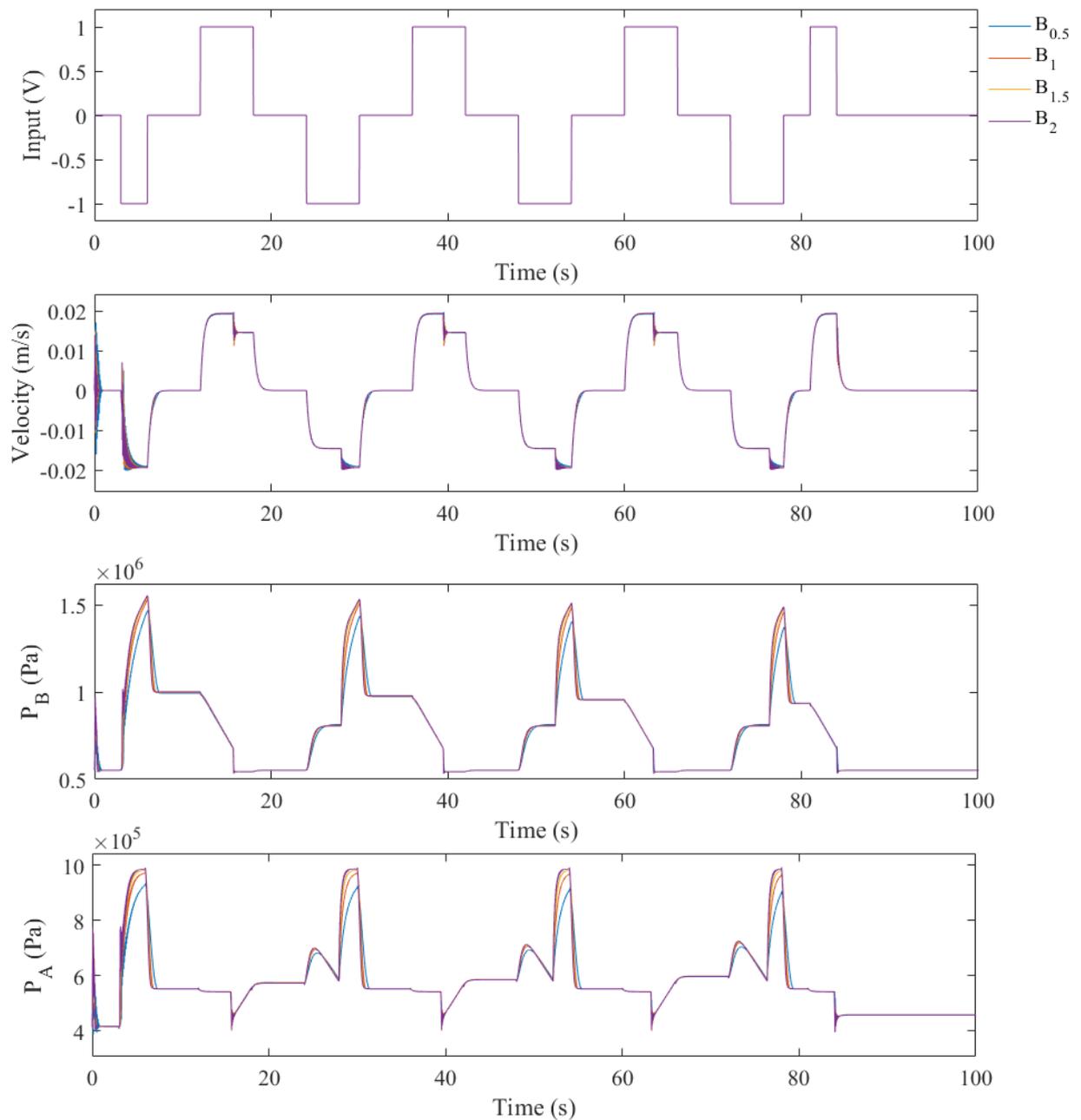


Fig. 4.21 Simulation results for systems with different bulk moduli

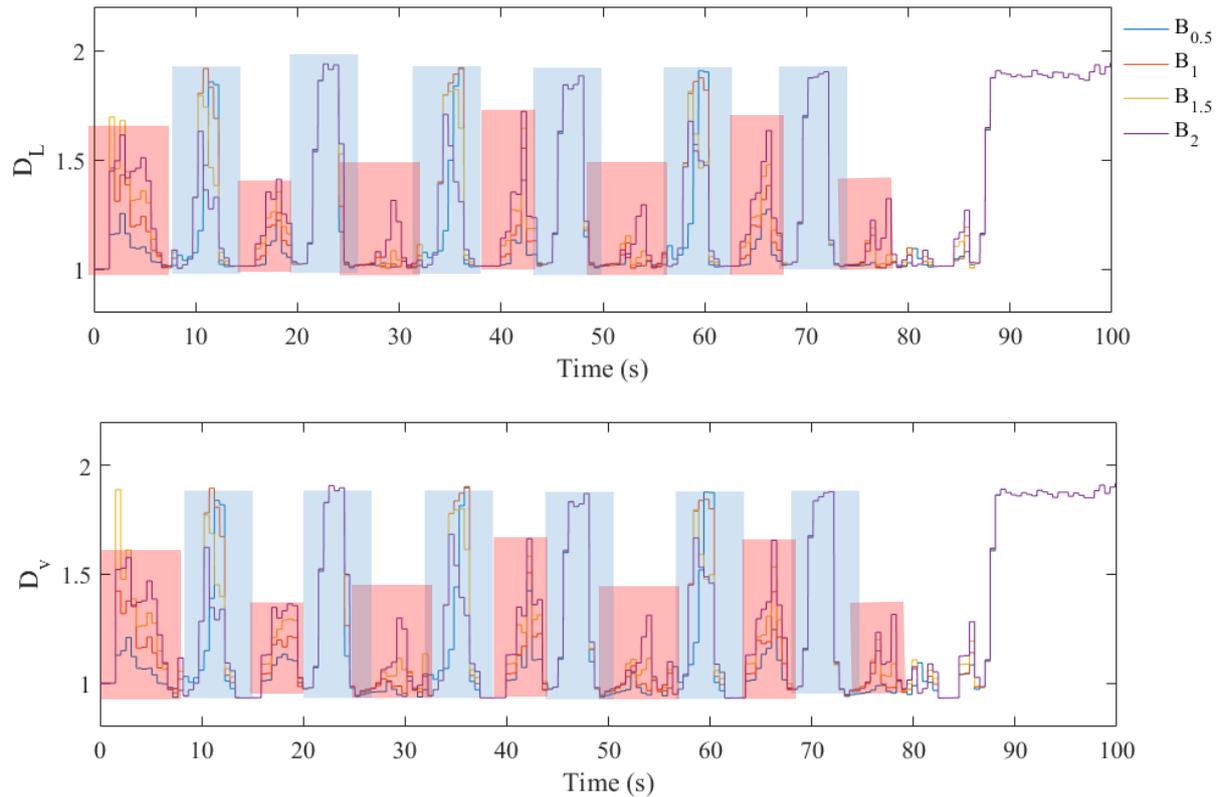


Fig. 4.22 Length and Variance dimensions for systems with different bulk moduli pertaining to Fig. 4.21.

As shown in Fig. 4.22, during the operating mode, the same behaviour which was seen in section 4.2 for bulk modulus change is observed here. These similar trends are highlighted with red and blue films and are highlighted with blue and red films. The red films represent the conditions where the lower bulk moduli correspond to the lower fractal dimensions (variance and length) and higher bulk moduli to the higher fractal dimension. For instance, during the red film, $t=62s-68s$, the average bulk moduli have the mean values which are shown in Table 4.3.

Table 4.3 Mean of fractal dimensions for systems shown in Fig. 4.22 at $t= [62s - 68s]$.

System Condition	\bar{D}_σ	\bar{D}_L
$\beta_{0.5} = 344.5\text{MPa}$	1.0686	1.0942
$\beta_1 = 689\text{MPa}$	1.1321	1.1473
$\beta_{1.5} = 1033.5\text{MPa}$	1.1934	1.2013
$\beta_2 = 1378\text{MPa}$	1.2466	1.2559

During the blue films where the quadrant changes happen, based on the direction of the quadrant change, either the fractal dimensions have approximately the same value or the higher

bulk moduli correspond to the lower fractal dimensions. The former happens when changing from quadrants 1 and 3 to quadrants 2 and 4, and the latter happens when changing from quadrants 2 and 4 to quadrants 1 and 3.

After $t=84s$ where input becomes zero and the oscillations are above the noise level, before $t=88s$, the four systems are in non-operating mode. While the same ascending order of correlation between bulk moduli and their corresponding fractal dimensions exists. The differences are smaller and consequently harder to detect. This is as expected from the analysis of Section 4.2.

The wavelet detail coefficients of the rod side pressure signal, P_B , for the four conditions are shown in Fig. 4.23.

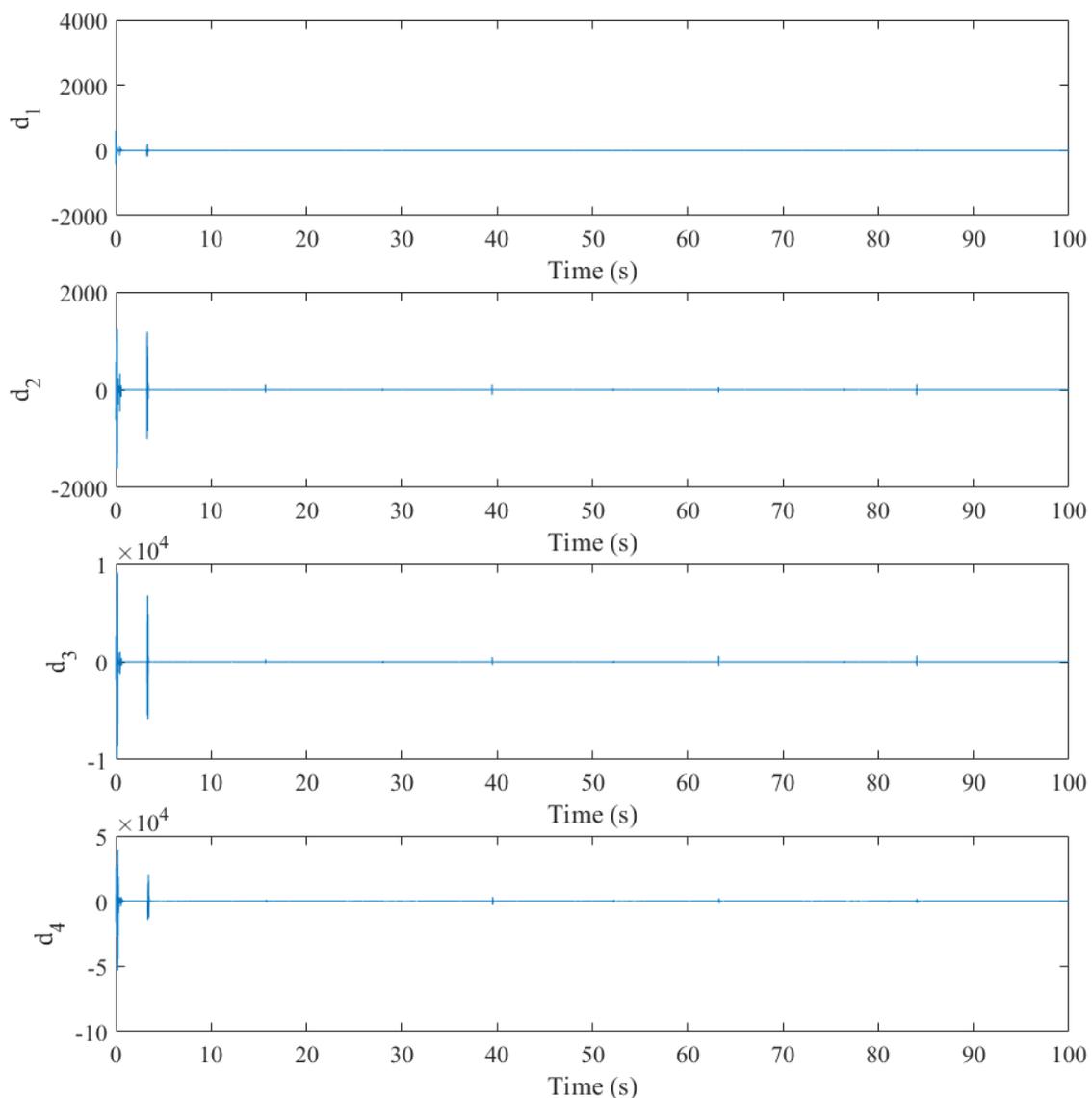


Fig. 4.23 Wavelet detail coefficients of systems with bulk modulus $\beta_{0.5} = 344.5\text{MPa}$ pertaining to Fig. 4.21.

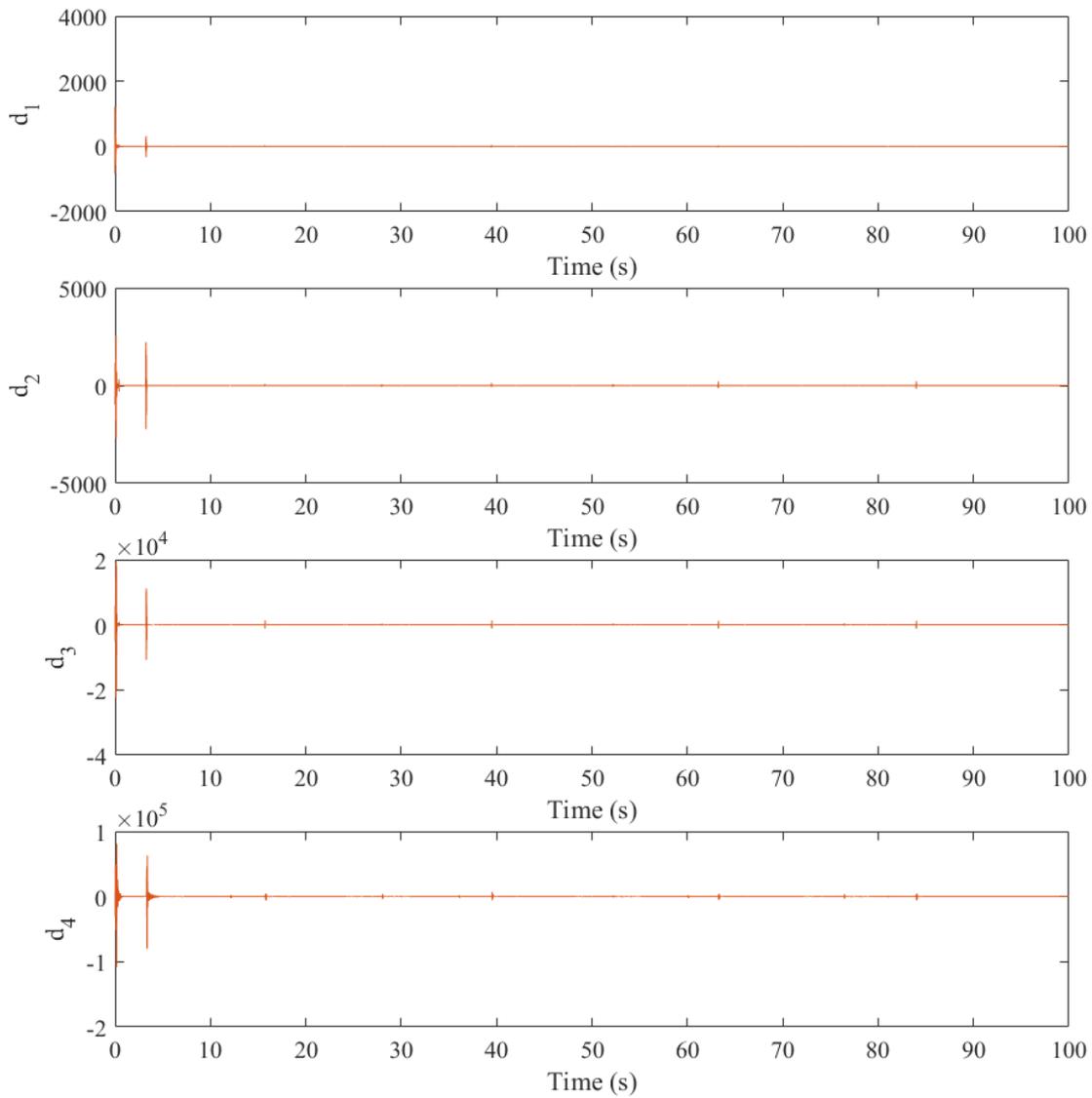


Fig. 4.24 Wavelet detail coefficients of systems with bulk modulus $\beta_1 = 689\text{MPa}$ pertaining to Fig. 4.21.

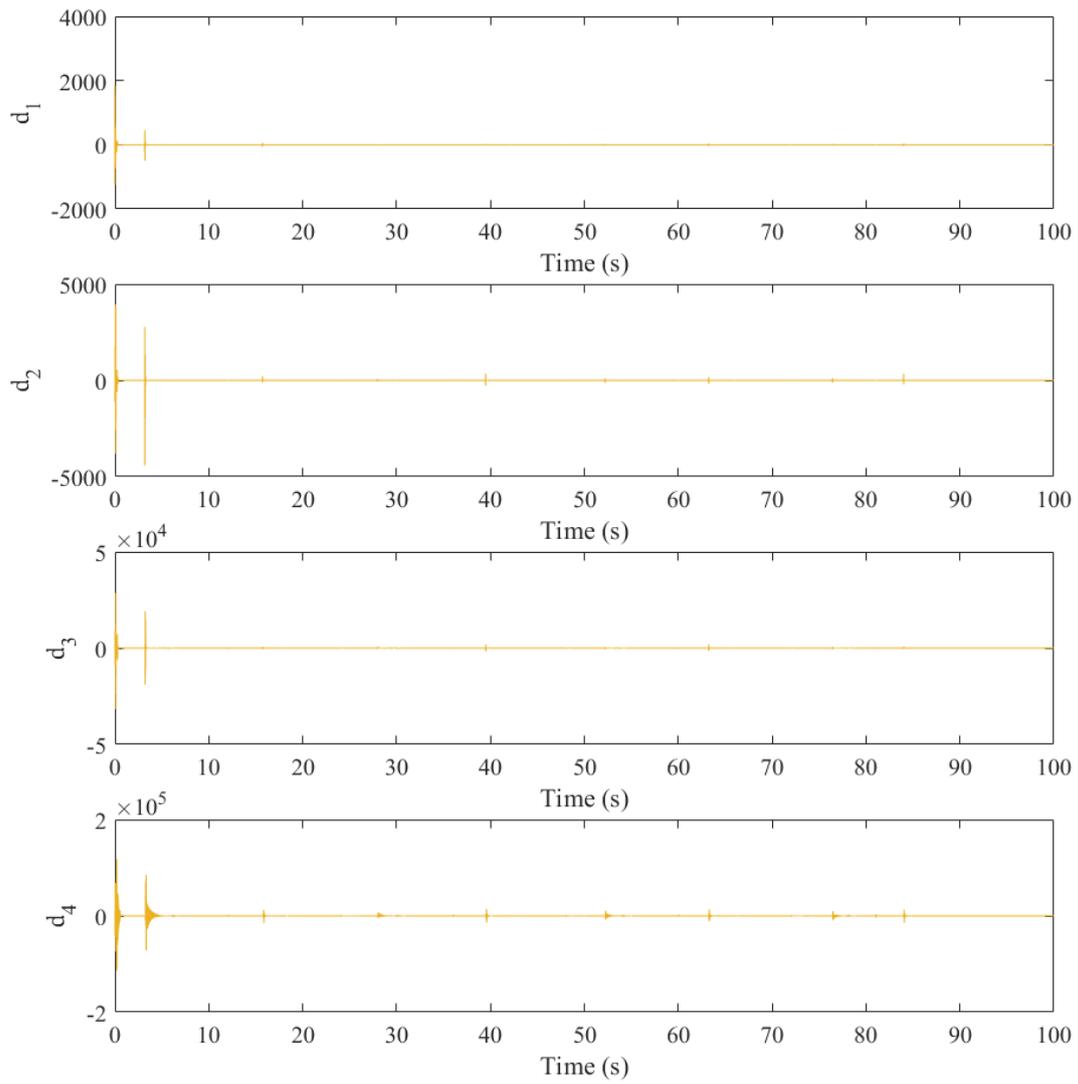


Fig. 4.25 Wavelet detail coefficients of systems with bulk modulus $\beta_{1.5} = 1033.5\text{MPa}$ pertaining to Fig. 4.21.

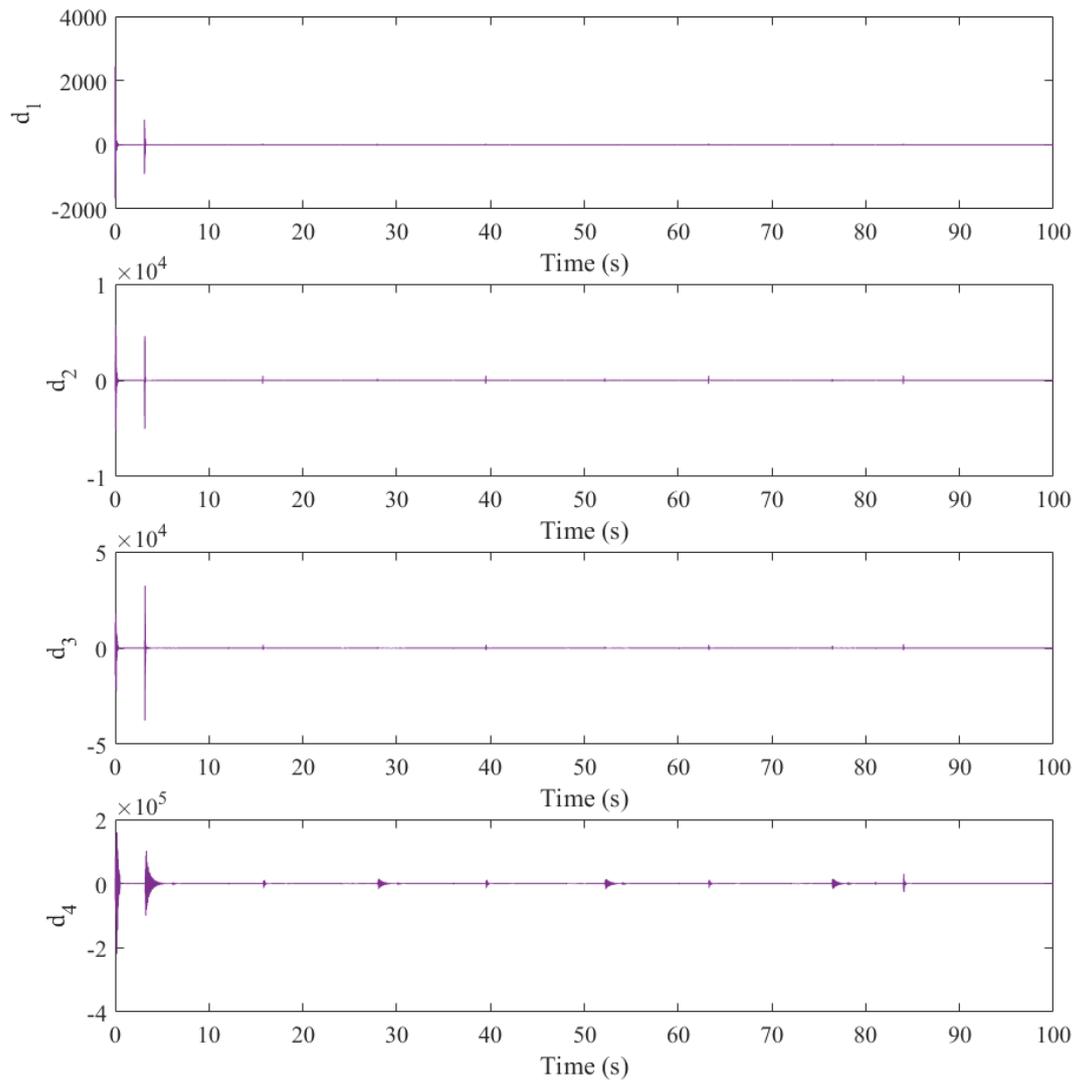


Fig. 4.26 Wavelet detail coefficients of systems with bulk modulus $\beta_2 = 1378\text{MPa}$ pertaining to Fig. 4.21.

As shown, the analysis in the case of wavelet detail coefficients is much easier. For level 4 detail coefficients, higher bulk moduli simply mean a higher value of mean, maximum and variance for the wavelet detail coefficients. In order to further emphasize this observation, the mean value, the maximum, variance, and autocorrelation for a moving window of 128 detail coefficients are contained and shown in Fig. 4.27.

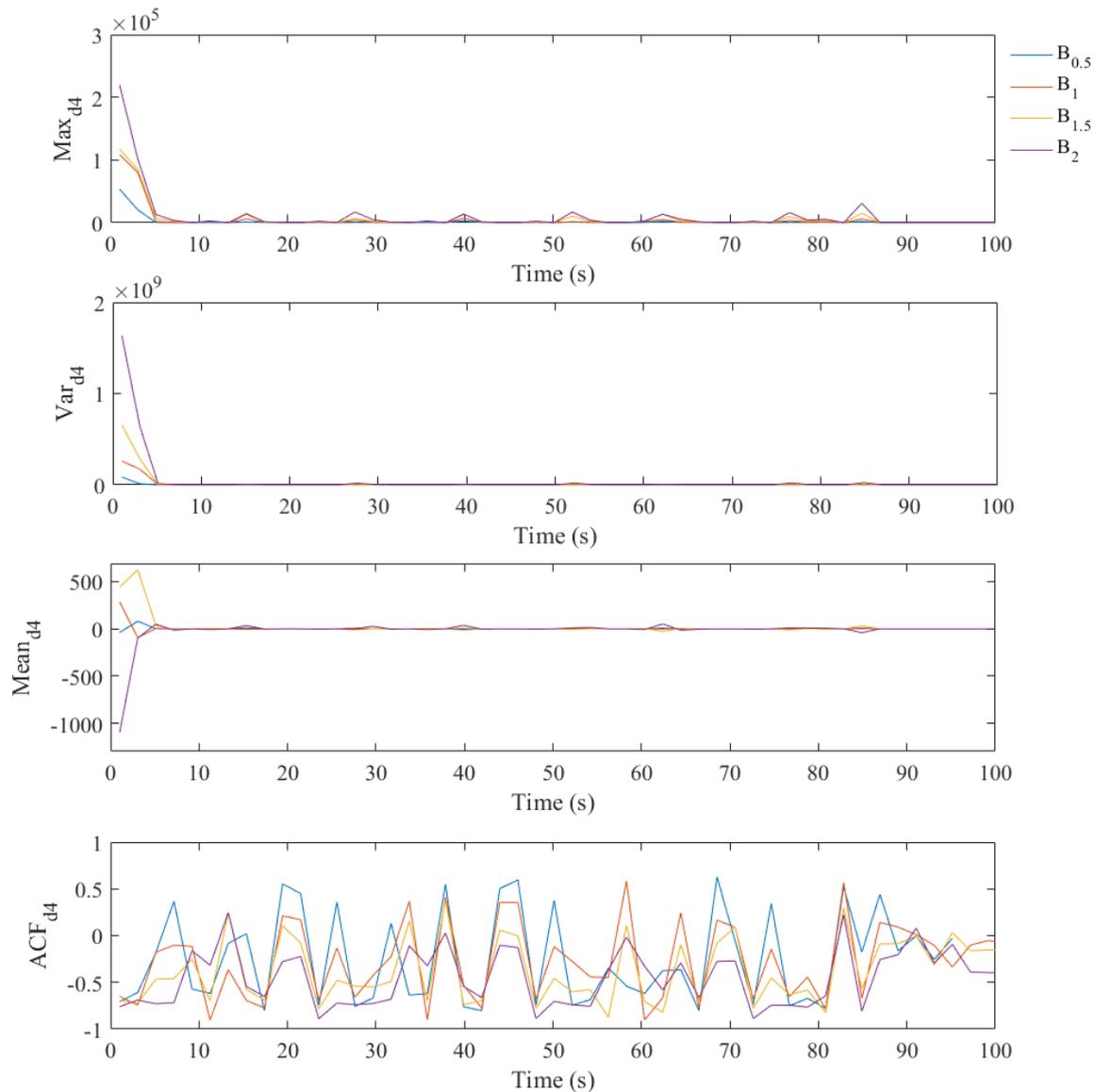


Fig. 4.27 Statistical moments of wavelet detail coefficients of systems with different bulk moduli pertaining to Fig. 4.21.

As shown, while the autocorrelation function does not represent any pattern, the mean and variance of the detail coefficients have distinctive values for different bulk moduli. Systems with lower bulk modulus have lower (in magnitude) variance and mean values for their level 4 detail coefficients.

4.4. Degree of Fault Variable and Algorithm

In order to unify the analysis in this chapter for internal leakage and give an online quantifiable measure to the fault existing in the experimental setup, a new variable fault has been introduced. This variable is named the *Degree of Fault* (DoF) and is obtained online based

on the three measures: wavelet detail coefficients, variance fractal dimension and length fractal dimension. The DoF can be obtained as follows using the *fault decision algorithm*.

1. Measures:

1.1. Obtain variance fractal dimension D_σ for a moving frame of 1024 samples with 512 samples overlapping.

1.2. Obtain length fractal dimensions D_l for a moving frame of 1024 samples with 512 samples overlapping.

1.3. Obtain level 4 detail coefficients d_4 for a moving frame of 16384 samples.

2. Indicators of Measures:

2.1. Obtain mean, $Mean(D_\sigma)$, variance, $Var(D_\sigma)$, and absolute maximum value, $AMax(D_\sigma)$, of D_σ for a moving frame of 60 measuring points (3072 samples in with no overlap or 30.72 seconds).

2.2. Obtain mean, $Mean(D_l)$, variance, $Var(D_l)$, and absolute maximum value, $AMax(D_l)$, of D_l for a moving frame of 60 measure points (30720 samples or 30.72 seconds).

2.3. Obtain mean, $Mean(d_4)$, variance, $Var(d_4)$, and absolute maximum value, $AMax(d_4)$, of detail coefficients d_4 at level 4.

3. Calculate three components of DoF corresponding to each measure as follows:

$$DoF_\sigma = \max \left\{ \begin{array}{l} \left[\frac{|Mean(D_\sigma)| - \bar{M}_\sigma}{\bar{M}_\sigma} \right], \\ \left[\frac{AMax(D_\sigma) - \bar{A}_\sigma}{\bar{A}_\sigma} \right], \\ \left[\frac{Var(D_\sigma) - \bar{V}_\sigma}{\bar{V}_\sigma} \right], 0 \end{array} \right\} \quad (38)$$

$$DoF_l = \max \left\{ \begin{array}{l} \left[\frac{|Mean(D_l)| - \bar{M}_l}{\bar{M}_l} \right], \\ \left[\frac{AMax(D_l) - \bar{A}_l}{\bar{A}_l} \right], \\ \left[\frac{Var(D_l) - \bar{V}_l}{\bar{V}_l} \right], 0 \end{array} \right\} \quad (39)$$

$$DoF_4 = \max \left\{ \begin{array}{l} \left[\frac{\bar{M}_4 - |Mean(d_4)|}{\bar{M}_4} \right], \\ \left[\frac{\bar{A}_4 - AMax(d_4)}{\bar{A}_4} \right], \\ \left[\frac{\bar{V}_4 - Var(d_4)}{\bar{V}_4} \right], 0 \end{array} \right\} \quad (40)$$

where \bar{M}_x , \bar{A}_x , \bar{V}_x are constants for the healthy system and are described in Table 4.4.

4. Obtain DoF as follows:

$$\text{DoF} = C_{DoF} \times DoF_{\sigma} \times DoF_l \times DoF_4 \quad (41)$$

where C_{DoF} is the normalization constant.

Table 4.4 Constants for DoF calculation in fault decision algorithm.

Variable	Description	Value
\bar{M}_{σ}	Threshold of mean value of variance fractal dimension for healthy system during 60 measuring points	1.22
\bar{M}_l	Threshold of mean value of length fractal dimension for healthy system during 60 measuring points	1.21
\bar{M}_4	Threshold of mean value of level 4 detail coefficients dimension of healthy system in a frame of 16384 samples	0.05
\bar{A}_{σ}	Threshold of absolute maximum value of variance fractal dimension for healthy system during 60 measuring points	1.56
\bar{A}_l	Threshold of absolute maximum value of length fractal dimension for healthy system during 60 measuring points	1.54
\bar{A}_4	Threshold of absolute maximum value of level 4 detail coefficients dimension of healthy system in a frame of 16384 samples	34.5
\bar{V}_{σ}	Threshold of variance value of variance fractal dimension for healthy system during 60 measuring points	0.035
\bar{V}_l	Threshold of variance value of length fractal dimension for healthy system during 60 measuring points	0.022
\bar{V}_4	Threshold of variance value of level 4 detail coefficients dimension of healthy system in a frame of 16384 samples	3.45
C_{DoF}	The normalization constant	410.6

The parameters in Table 4.4 are determined experimentally by conducting several experiments for the healthy system and averaging the results.

The reason behind using the three moments as indicators (variance, mean, and absolute maximum) for each of the three measures (variance fractal dimension, length fractal dimension, and wavelet detail coefficients) with an OR operator (maximum here) in (38), (39) and (40) is that while one indicator could fall short in detecting and indicating faults, as was shown in analyses of Section 4.2, the combination of these indicators is more robust towards fault detection. Furthermore, to improve the robustness against incorrect fault detection where the system is healthy, an AND operator (multiplication here) is used in (41) so that DoF has a non-zero value only if all three measures indicate a fault in the system, i.e. have non-zero values. Thus, employing the polyscale measures (variance and length fractal dimension) and the

multiscale measure (wavelet detail coefficient) contributes to the performance of the fault detection scheme against incorrect indications of faults.

Fig. 4.28 shows the results of an experiment with the load of 163kg in which an internal leakage fault occurs at approximately $t=210$ s and Fig. 4.29 shows the results of another experiment with the load of 163kg in which an internal leakage fault occurs at approximately $t=231$ s.

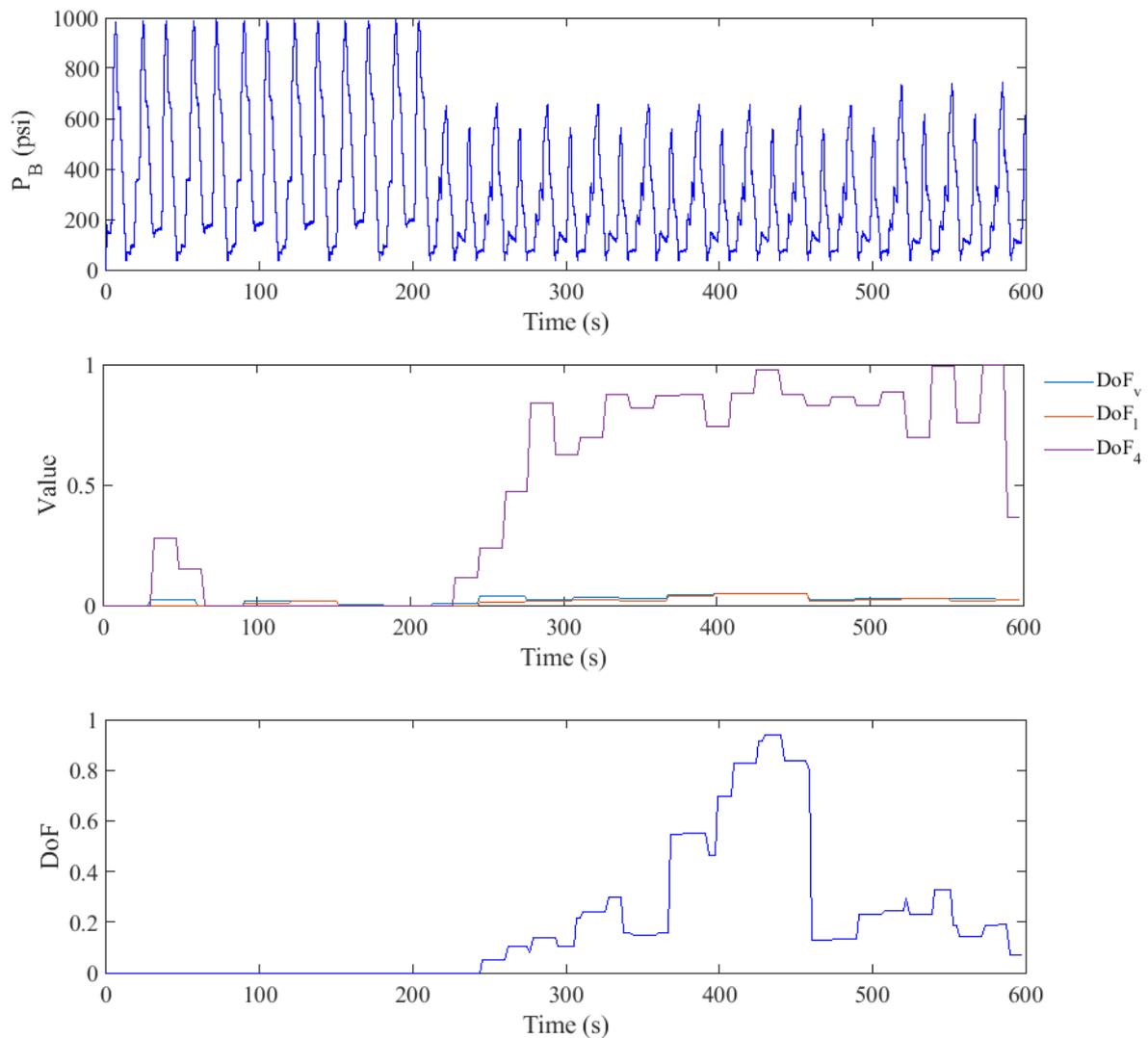


Fig. 4.28 Results of first experiments for validating the fault decision algorithm and DoF variable.

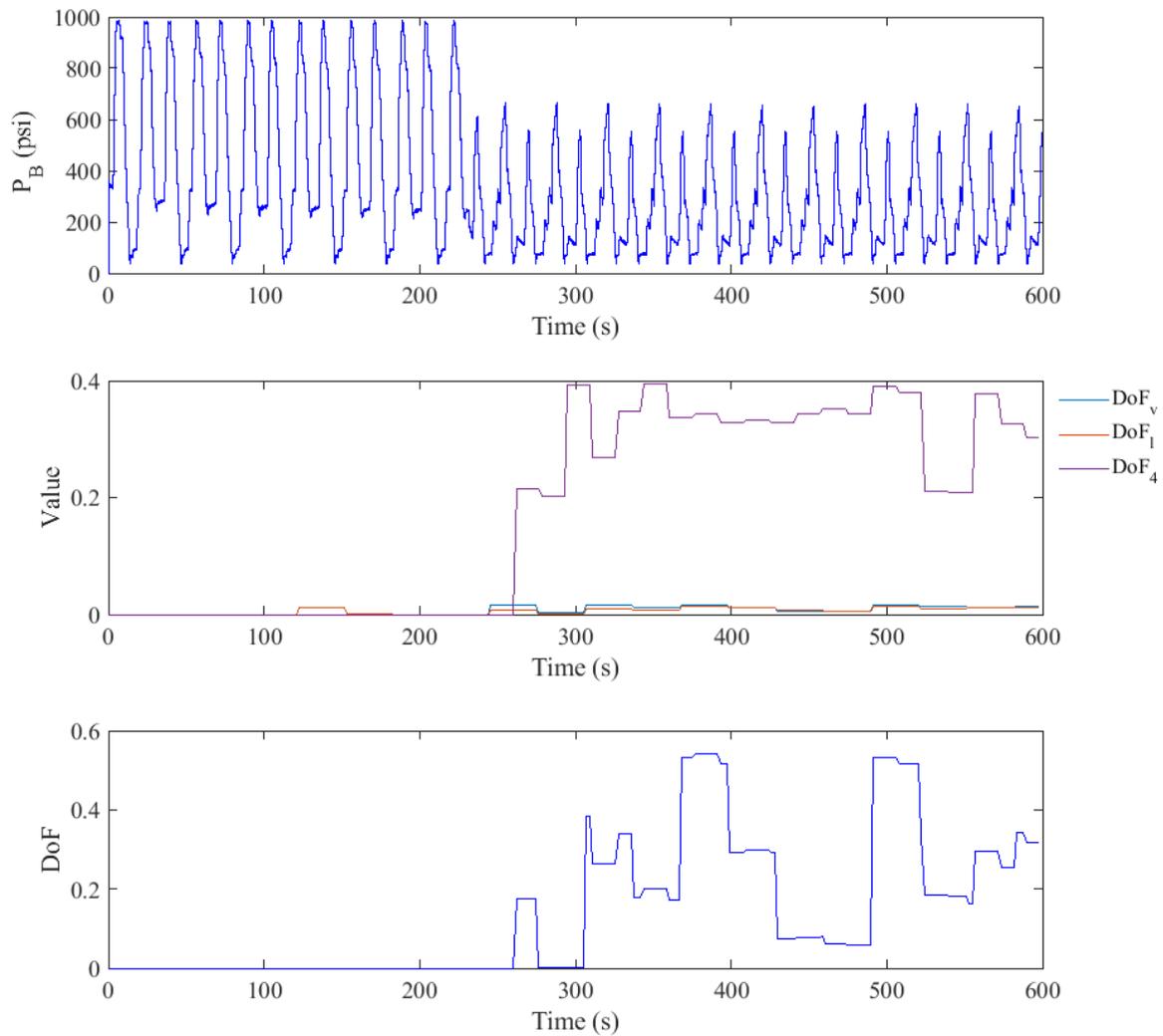


Fig. 4.29 Results of second experiments for validating the fault decision algorithm and DoF variable.

As shown in Fig. 4.28, in the first experiment the DoF variable, and in turn the fault decision algorithm, is successful in detecting the fault 86.6% of the time (relative to the period where fault was present). As shown in Fig. 4.29, in the second experiment the DoF variable, and in turn the fault decision algorithm, is successful in detecting the fault 83.2% of the time.

While the fault happens at the earlier time of $t=210$ s in the first experiment and at the time of $t=231$ s for the second experiment, in both experiments the first indication of the existence of the fault is at $t=262.1$ s. This is because the indicators of the fractal measures are updated approximately every 30.72 seconds. Thus, an approximate delay of up to 30.72 seconds is to be expected before the DoF variable indicates a fault. The majority of the error in the performance of the fault decision algorithm is due to this delay.

As expected, the use of three measures prevents the wrong detection of faults. For instance, at $t \cong 31$ s to $t \cong 60$ s the DoF_{σ} and DoF_4 are both non-zero. However, DoF_l is zero thus

the DoF variable remains zero. While this design is conservative, it contributes to reserving certainty in detecting healthy operations.

A third experiment with varying leakage has been performed as shown in Fig. 4.30. In this experiment, the load has increased to 367 kg (full load) to show the robustness of the method against varying conditions.

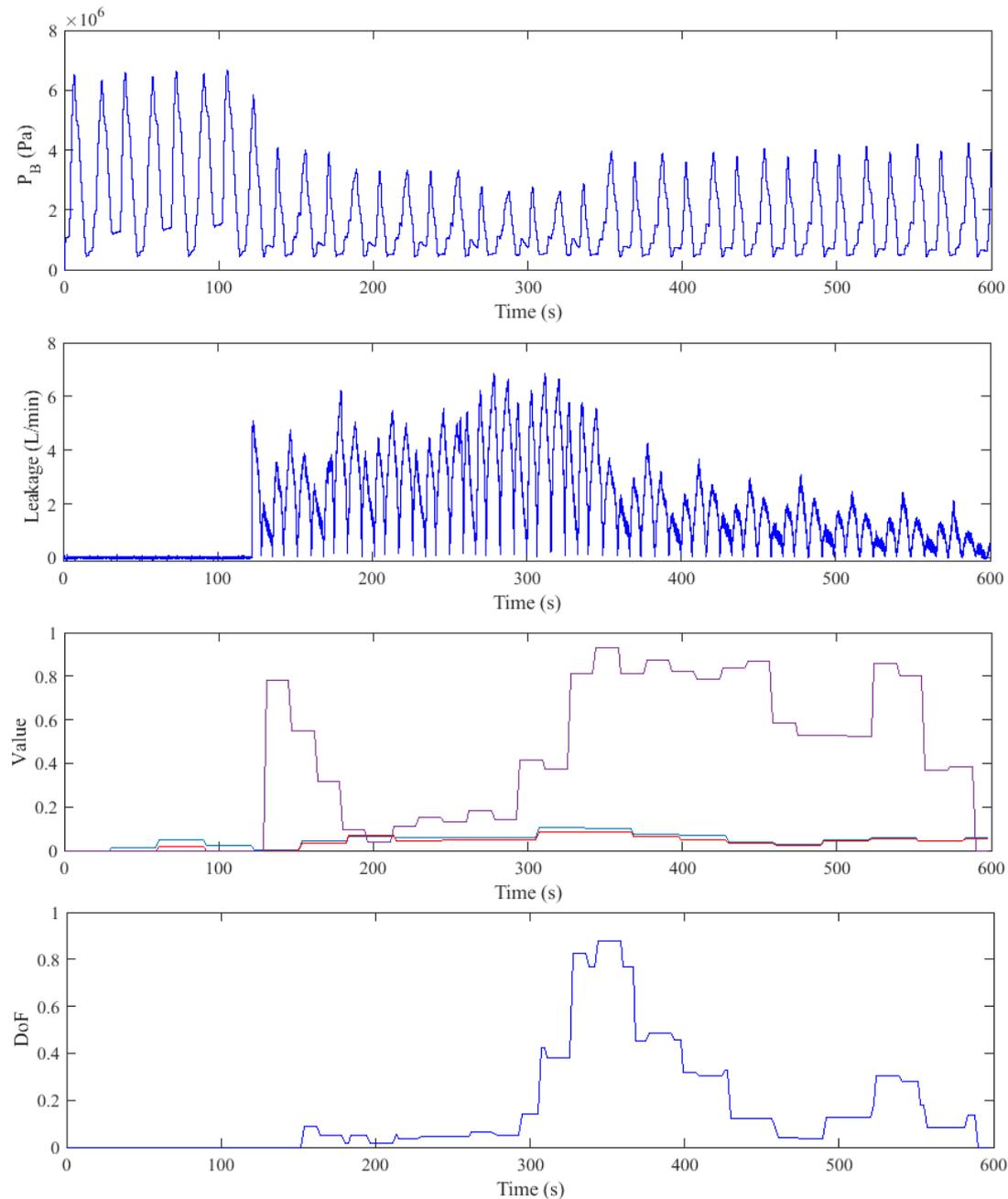


Fig. 4.30 Results of third experiments for validating the fault decision algorithm and DoF variable.

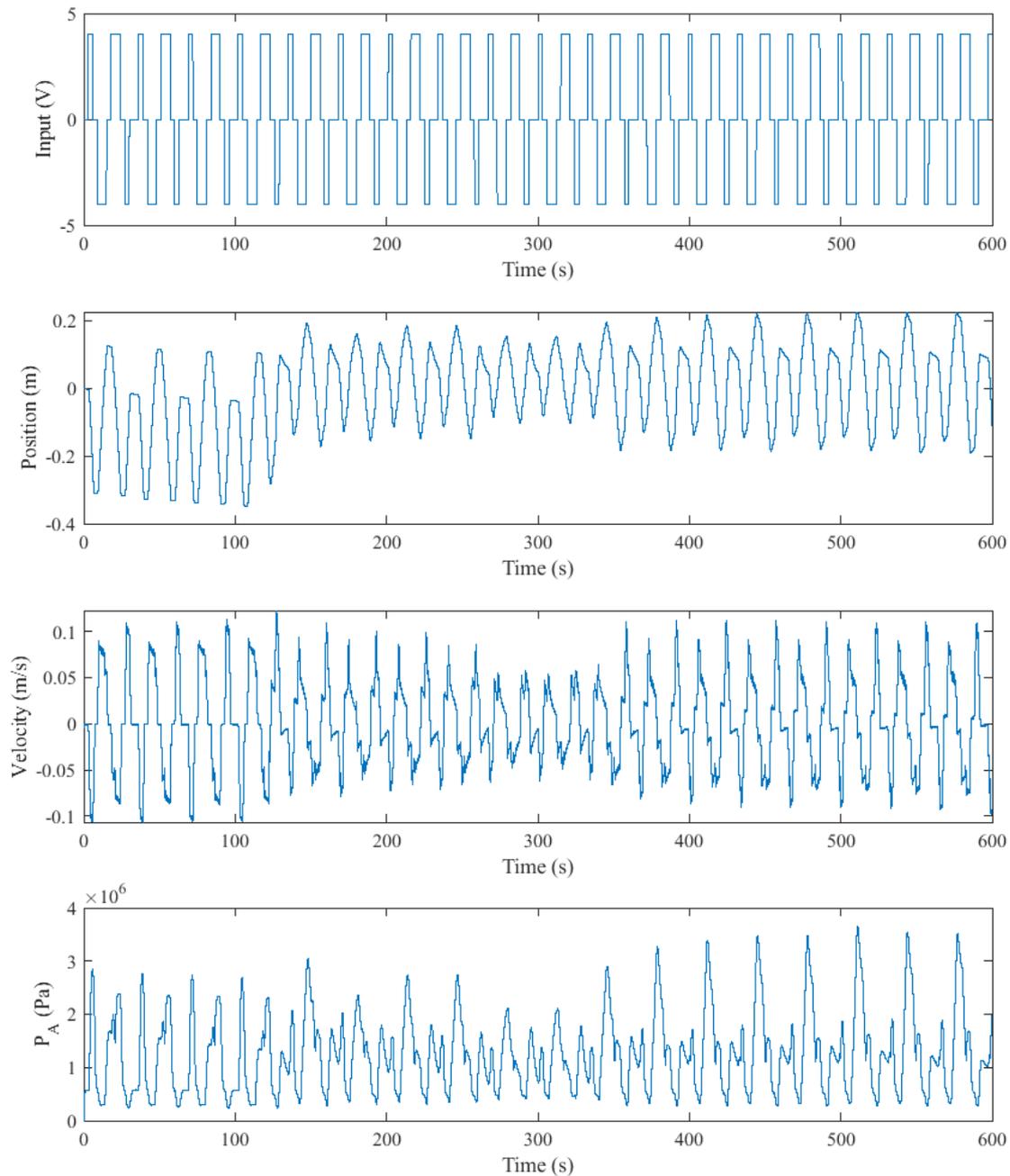


Fig. 4.31 Additional results of third experiments for validating the fault decision algorithm and DoF variable.

As shown, the DoF variable is able to indicate the existence of fault with 84.2% accuracy. Also, comparing the DoF variable with the leakage in the system indicates the proportionality of the DoF variable to the severity of internal leakage in the system. Furthermore, the proposed algorithm is successful in showing the leakages with the minimum value of approximately 1.2L/m.

4.5. Summary

Measures with polyscale and multiscale characteristics have been successful in a fault detection capacity [5] for hydraulic systems. In this work, length and variance fractal dimensions and wavelet detail coefficients are described to be used for fault detection analysis.

Using the described measures, the behaviour of the healthy system is compared against a system with bulk modulus change and a system with internal leakage. The results of the analysis indicate that in non-operating mode by comparing the variance and length fractal dimensions of the pressure signal for a faulty system with a healthy system, not only those faults are detectable but the type of faults in terms of internal leakage or bulk modulus change can be inferred. Also, it is shown that while the system is in operation mode, variance and length fractal dimensions or level 1 and 4 wavelet detail coefficients' maximum and variance could be utilized for detecting faults and their types in terms of internal leakage or bulk modulus change.

The analysis was then expanded by discussing the sensitivity of measures to different levels of internal leakage and bulk modulus change. With regards to fractal dimensions, similar results and behaviours and the repeating patterns emerged in the results. Based on these patterns, quantified differences in fractal dimensions of healthy system and system with internal leakage was recorded. These differences in the fractal dimensions proved proportional to the level of fault. The same was shown for wavelet detail coefficients. In fact, with respect to the degree of internal leakage and bulk modulus change, the more severe the fault was, the greater the difference was observed in the measures. In other words, severer faults present greater fractal dimensions and greater maximum and variance of level 4 detail coefficients. This was true for the system in operating mode as well as non-operating mode.

Finally, a fault decision algorithm was introduced to quantify the analysis utilizing a DoF variable in an online experiment. This variable, produced by the fault decision algorithm, was successful in fault detection 83.216% of the time in the worst-case scenario for a fault window of 390s. The DoF variable is essential for designing the fault-tolerant control scheme, as the means of assessing the degree of fault in the system.

CHAPTER 5:

FAULT-TOLERANT CONTROL DESIGN¹

In this chapter, a fractional-order proportional–integral–derivative controller is implemented to the newly developed single-rod Electrohydrostatic Actuator (EHA) [2]. The Oustaloup recursive method [39] is employed for obtaining a transfer function representing the Fractional-order Proportional Integral Derivative (FOPID) controller. The parameters of the controller are chosen based on the Modified Nelder-Mead optimization algorithm. Experiments have been conducted to test the performance of the designed control scheme based on the design criteria.

In the case of the faulty system with internal leakage, first, another FOPID controller is designed and tuned using the Modified Nelder-Mead (MNM) algorithm. Then, a decision algorithm and a fuzzy-FOPID controller are introduced for fault-tolerant control of the EHA which decides the final input to the system based on the degree of fault in the system and the control input provided by the two FOPID controllers.

5.1. Approximation of Fractional-order Transfer Functions

While the standard Proportional Integral Derivative (PID) controllers are used extensively in the industry due to their straightforward implementation, they are not robust enough to produce satisfactory results under a wide range of system uncertainties [6, 40]. The shortcomings of the standard PID controller have been mitigated considerably by generalizing the main concept to the fractional-order PID controller [39, 41].

The FOPID controller generalizes the integer-order derivative and integral to fractional-order derivative and integrator. This class of controllers has the advantage of two more variables (i.e., the exponent of the derivative and the integrator) to adjust in order to achieve improved performance for the controller. Furthermore, a fractional-order differentiator significantly reduces the magnitude amplification of the high-frequency noise as compared to the integer-order one [40, 41].

¹ A part of this chapter has been published in [39].

Fractional-order derivator and integrators in the Laplace domain are shown as S^α and $S^{-\alpha}$, respectively. Thus, a fractional-order integration of input $U(s)$ is obtained as [39],

$$Y(s) = \left(\frac{S}{\omega_u}\right)^\alpha U(s) \quad (42)$$

where w_u is called the unit frequency gain. Based on (42), the transmittance (s) is defined as [39],

$$D(s) = \left(\frac{S}{\omega_u}\right)^\alpha \quad (43)$$

The $\frac{S}{\omega_u}$ transfer function can then be limited to a predetermined frequency range, $[\omega_A, \omega_B]$, by cutting off high and low frequencies. Thus $\frac{S}{w_u}$ is replaced by,

$$\frac{S}{\omega_u} \cong C_0 \frac{1 + \frac{S}{\omega_b}}{1 + \frac{S}{\omega_h}} \quad (44)$$

where,

$$\begin{cases} \omega_u = (\omega_b \omega_h)^{0.5} \\ C_0 = \frac{\omega_b}{\omega_u} = \frac{\omega_u}{\omega_h} \end{cases} \quad (45)$$

$$\begin{cases} \omega_b \leq \omega_A \\ \omega_b \geq \omega_B \end{cases} \quad (46)$$

where ω_b and ω_h denote the low-transitional and high-transitional frequencies, respectively. Substituting (44) into (43) gives,

$$D(s) = \left(\frac{\omega_u}{\omega_h}\right)^\alpha \left(\frac{1 + \frac{S}{\omega_b}}{1 + \frac{S}{\omega_h}}\right)^\alpha \quad (47)$$

In the Oustaloup recursive method, to approximate the fractional-order transfer function of (47), a recursive distribution of real poles and zeros is used that leads to,

$$D(s) = \lim_{M \rightarrow \infty} D_m(s) \quad (48)$$

where,

$$D_m(s) = \left(\frac{\omega_u}{\omega_h}\right)^\alpha \prod_{k=-M}^M \frac{1 + \frac{S}{\acute{\omega}_k}}{1 + \frac{S}{\omega_k}} \quad (49)$$

where $\acute{\omega}_k$ and ω_k are defined as,

$$\acute{\omega}_k = \omega_b \left(\frac{\omega_u}{\omega_h}\right)^{\left[\frac{k+M+0.5-\alpha}{2M+1}\right]} \quad (50)$$

$$\omega_k = \omega_b \left(\frac{\omega_u}{\omega_h} \right)^{\left[\frac{k+M+0.5+\frac{\alpha}{2}}{2M+1} \right]} \quad (51)$$

If we select M between 2 and 4 for the EHA system [40], $D_m(s)$ becomes an integer-order transfer function approximating the fractional-order differentiator $D(s)$. Also, it should be noted that there are $2M + 1$ numbers of zeros and poles in $D_m(s)$.

Note that, to find an approximation of the non-integer integrators, the fractional-order α takes negative values [39].

Finally, the FOPID controller can be formulated as,

$$G(s) = K_p + K_i s^{-\lambda} + K_d s^\mu \quad (52)$$

The five parameters, $[K_p, K_i, K_d, \mu, \lambda]$, of the FOPID controller in (52) are to be optimized using the MNM algorithm.

5.2. Modified Nelder-Mead Optimization Algorithm

The Nelder-Mead downhill simplex algorithm [42] is a nonlinear optimization algorithm that is gradient-free and unconstrained. The Guin variation of this method [43] incorporates implicit constraints in the algorithm. This algorithm uses reflection, expansion, contraction, and shrinkage to find local minimizer following five steps. It uses $(n + 1)$ vertices x_1, x_2, \dots, x_{n+1} as initial simplex where n is the number of parameters to be optimized for minimization of the objective function. The objective function or cost of each of these points are $f(x_1), f(x_2), \dots, f(x_{n+1})$. The algorithmic steps are as follows:

1. The algorithm takes the centroid of all points, x_0 , except x_{n+1} where $f(x_1) \leq f(x_2) \leq \dots \leq f(x_{n+1})$.
2. It calculates the reflected point $x_r = x_0 + \varepsilon(x_0 - x_{n+1})$ where $\varepsilon = 1$ is the reflection coefficient.
3. If the reflected point satisfies $f(x_r) \leq f(x_{n+1})$, it calculates the expanded point $x_e = x_0 + \gamma(x_0 - x_{n+1})$ where $\gamma = 2$ is the expansion coefficient.
4. If the expanded point is such that $f(x_e) \leq f(x_r)$, the algorithm replaces x_{n+1} by x_e and makes a new simplex. Otherwise, it replaces x_{n+1} by x_r and makes a new simplex.
5. If $f(x_e) > f(x_{n+1})$, the algorithm computes the contracted points $x_c = x_0 \pm \rho(x_{n+1} - x_0)$ where $\rho = 0.5$ is the contraction coefficient.
6. If the better of contracted points is such that $f(x_c) \leq f(x_{n+1})$, the algorithm replaces x_{n+1} by x_c and makes a new simplex.
7. If $f(x_c) > f(x_{n+1})$, the algorithm replaces all points of simplex except x_1 by $x_i = x_1 + \sigma(x_i - x_1)$ for $i = \{2, \dots, n + 1\}$ where $\sigma = 0.5$ is shrinkage coefficient.

The Nelder-Mead algorithm terminates when at least one of small simplex convergence test, function value convergence test, no convergence test becomes true.

The method used in this thesis has been developed by Butt et al. [44] based on the Guin variation of the Nelder-Mead algorithm and is shown to be effective for the EHA system. In this method, first, based on the bound on the design variables an n -dimensional search space is formed. Then, the algorithm identifies the region of interest, $\mathbf{S} = [V_1, V_2, \dots, V_{2^n}]^T$, to search for optimal parameters. V_1, V_2, \dots, V_{2^n} are vertices of n -dimensional search space with the order of $1 \times n$. This algorithm runs for a preset maximum of R restarts.

For the first run ($r = 1$) the initial simplex is obtained as follows:

$$\mathbf{IS} = (\mathbf{X}_c^T \cdot \mathbf{J}_{1,n+1})^T + \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ \alpha_1 & 0 & 0 & \dots & 0 \\ 0 & \alpha_2 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \alpha_n \end{bmatrix} \quad (53)$$

where \mathbf{X}_c is the centroid of the search space and \mathbf{J} is the matrix of ones with the order shown in subscript. α_i is obtained as,

$$\alpha_i = \begin{cases} -\frac{x_{i \max} - x_{i \min}}{\beta} & i = 1 \\ \frac{x_{i \max} - x_{i \min}}{\beta} & i = 2, \dots, n \end{cases} \quad (54)$$

By running the Guin augmented variant of the Nelder-Mead algorithm in each run, the local minimizer, \mathbf{X}_0 , is found and used for constructing initial simplex for the succeeding run.

If the centroid is at the origin, \mathbf{X}_0 is projected to 2^n points, \mathbf{P} , as follows,

$$\mathbf{P} = 0.5\mathbf{S} - 0.5(\mathbf{X}_0^T \cdot \mathbf{J}_{1,n+1})^T \quad (55)$$

If the centroid of the search space does not lie at the origin, the search space is moved to make the centroid lie at origin,

$$\hat{\mathbf{X}}_c = \mathbf{X}_c + [a_1 \quad a_2 \quad \dots \quad a_n] \quad (56)$$

where $\hat{\mathbf{X}}_c$ is the translated centroid, and a_i ($i=1, \dots, n$) denotes the distances from the origin.

Thus, variables $\hat{\mathbf{S}}$ and $\hat{\mathbf{X}}_0$ are also the translated versions by the matrix $[a_1 \quad a_2 \quad \dots \quad a_n]$. Consequently, the translated projected points, $\hat{\mathbf{P}}$, is computed using the translated variables from,

$$\hat{\mathbf{P}} = 0.5\hat{\mathbf{S}} - 0.5(\hat{\mathbf{X}}_0^T \cdot \mathbf{J}_{1,n+1})^T \quad (57)$$

The projected points are then translated back to the original search space using,

$$\mathbf{P} = \hat{\mathbf{P}} - ([a_1 \quad a_2 \quad \dots \quad a_n]^T \cdot \mathbf{J}_{1,n+1})^T \quad (58)$$

For all points in \mathbf{P} , the objective function is evaluated and one of the points is chosen as \mathbf{P}_0 which is then added to the linearly growing memory matrix \mathbf{M} .

For $r = 2$, the point that leads to the minimum objective function is \mathbf{P}_0 . For $r = 3, \dots, R$, the point that leads to the minimum objective function and is not an element of \mathbf{M} is \mathbf{P}_0 . The initial simplex for all the next runs is calculated as follows [44]:

$$\mathbf{IS} = (\mathbf{P}_0^T \cdot \mathbf{J}_{1,n+1})^T + \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ \alpha_1 & 0 & 0 & \dots & 0 \\ 0 & \alpha_2 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \alpha_n \end{bmatrix} \quad (59)$$

where for each of i dimensions, α_i ($i = 1, \dots, n$) is obtained as follows:

$$\alpha_i = \begin{cases} -\frac{x_{i \max} - x_{i \min}}{\beta} & \mathbf{P}_0 > \mathbf{X}_c \\ \frac{x_{i \max} - x_{i \min}}{\beta} & \mathbf{X}_c \geq \mathbf{P}_0 \end{cases} \quad (60)$$

This multimodal optimization algorithm stops when either one of the found minima is less than or equal to the desired minimum, or all projected points already exist in the memory vector in which case it selects the optimal local minimum found as the global minimum. Otherwise, the algorithm runs to finish the predetermined maximum number of restarts and then point with the minimum objective function.

For the FOPID controllers in this thesis, the five parameters in $\Lambda = [K_p, K_i, \lambda, K_d, \mu]$ from (52) are tuned based on the objective function of an error in velocity with respect to reference and jerk for each simulation of the program as follows:

$$OF = w_E \sqrt{\frac{\sum error_i^2}{n}} + w_J \sqrt{\frac{\sum jerk_i^2}{n}} \quad (61)$$

where w_E and w_J are the error and jerk term weights, and n is the number of samples.

5.3. Control Design for The Healthy system

Fig. 5.1 shows the block diagram of the controller and system. In order to tune the FOPID parameters, the MNM algorithm is used. The objective function is as described by (61). Using this block diagram, the objective function is evaluated at each point of the MNM algorithm (in both simulation and experimental optimizations). The EHA system model in the simulations is the model developed in Chapter 2 and in the experiments it is the experimental setup shown in Fig. 2.3. Table 5.1 shows the optimization parameters which are used in the simulations and experiments.

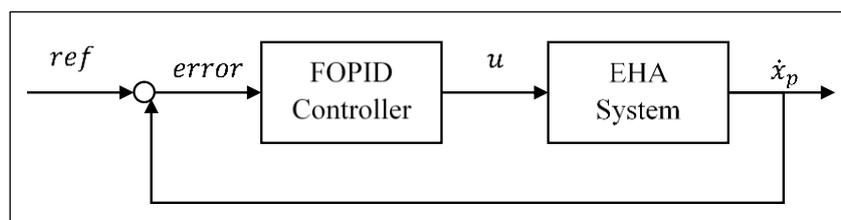


Fig. 5.1 Block diagram of the controller and system in healthy condition in pre-generated reference mode.

Table 5.1 Optimization parameters for the Tuning Simulations.

Parameter	Symbol	Value
Approximation method parameter	M	2
FOPID parameter upper bounds	Λ_{max}	[200,200,200,1,1]
FOPID parameter lower bounds	Λ_{min}	[0.1,0.1,0.1,0.01,0.01]
Maximum number of iterations for each run	N_{max}	sim = 500, exp = 100
Maximum number of restarts (runs)	R	sim = 20, exp = 6
Weight of error term	w_E	1,1.5
Weight of jerk term	w_J	0.2111,0.1111

Table 5.2 shows the results of the optimization simulations and experiments for three cases: Case 1: $w_E = 1$, $w_J = 0.1111$; Case 2: $w_E = 1.5$, $w_J = 0.1111$, and Case 3: $w_E = 1$, $w_J = 0.2111$. The objective function values with respect to the fractional-order exponents of the integrator and derivate are shown in Fig. 5.2 and Fig. 5.3.

Table 5.2 Controller Tuned Parameters in simulation and experiments for the healthy.

Case		Value				
		K_p	K_i	K_d	μ	λ
1	Simulation	7.0544	106.8000	0.1000	0.9844	0.9801
	Experiment	6.5739	76.4043	0.2401	0.9914	0.7046
2	Simulation	8.9941	113.4588	0.1631	0.9184	0.9580
	Experiment	4.9881	80.9567	0.3419	0.9233	0.6554
3	Simulation	4.8279	102.4354	0.1038	0.9187	0.9835
	Experiment	4.3002	76.1277	0.2001	0.9443	0.6913

Out of the three controllers studied in Fig. 5.2 and Fig. 5.3, the third case is chosen as the final controller because it shows less sensitivity to noise and disturbances which exist in the actual system compared to the other two. Based on the case 3 parameters of Table 5.2 (obtained based on the MNM algorithm), a FOPID controller is to be designed and further tuned to compensate for the many different nonlinearities and uncertainties in the system. The simulation-based optimization has the advantage of a higher number of iterations and runs. Consequently, it is more precise. On the other hand, the experimental optimization incorporates

the uncertainties and nonlinearities which are not accounted for in the mathematical model used for simulations. Thus, the final FOPID parameters from Table 5.2 are further adjusted by conducting trial experiments and chosen as follows in Table 5.3 based on the average of the online (experimental) and offline (simulation) optimizations. For the healthy condition experiments, the block diagram of the system is as in Fig. 5.1, with the only difference that the parameters of the FOPID controller are chosen based on Case 3 tuned parameters obtained from the MNM algorithm.

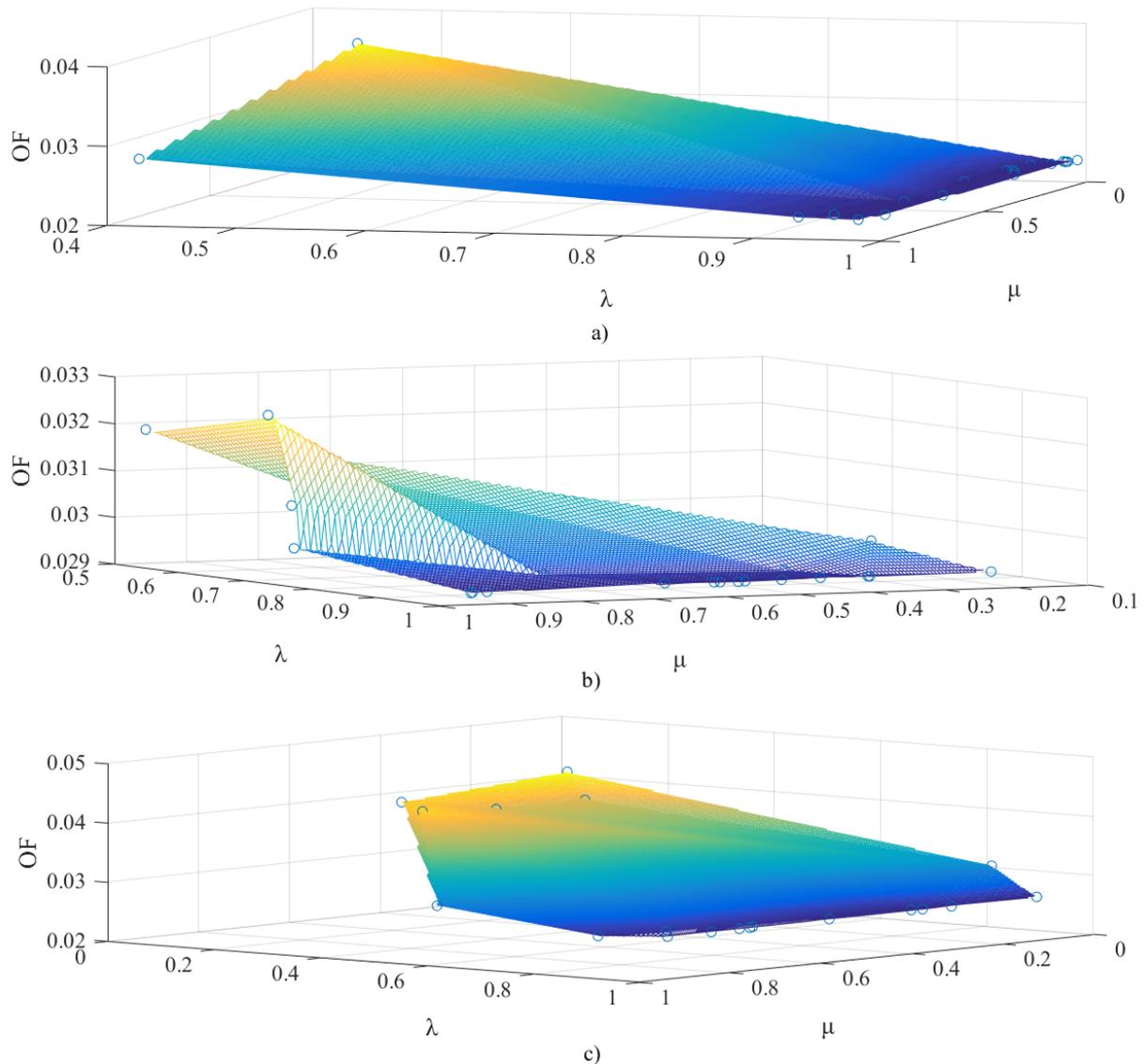


Fig. 5.2 Optimization results for three cases: a) Case 1, b) Case 2 and c) Case 3.

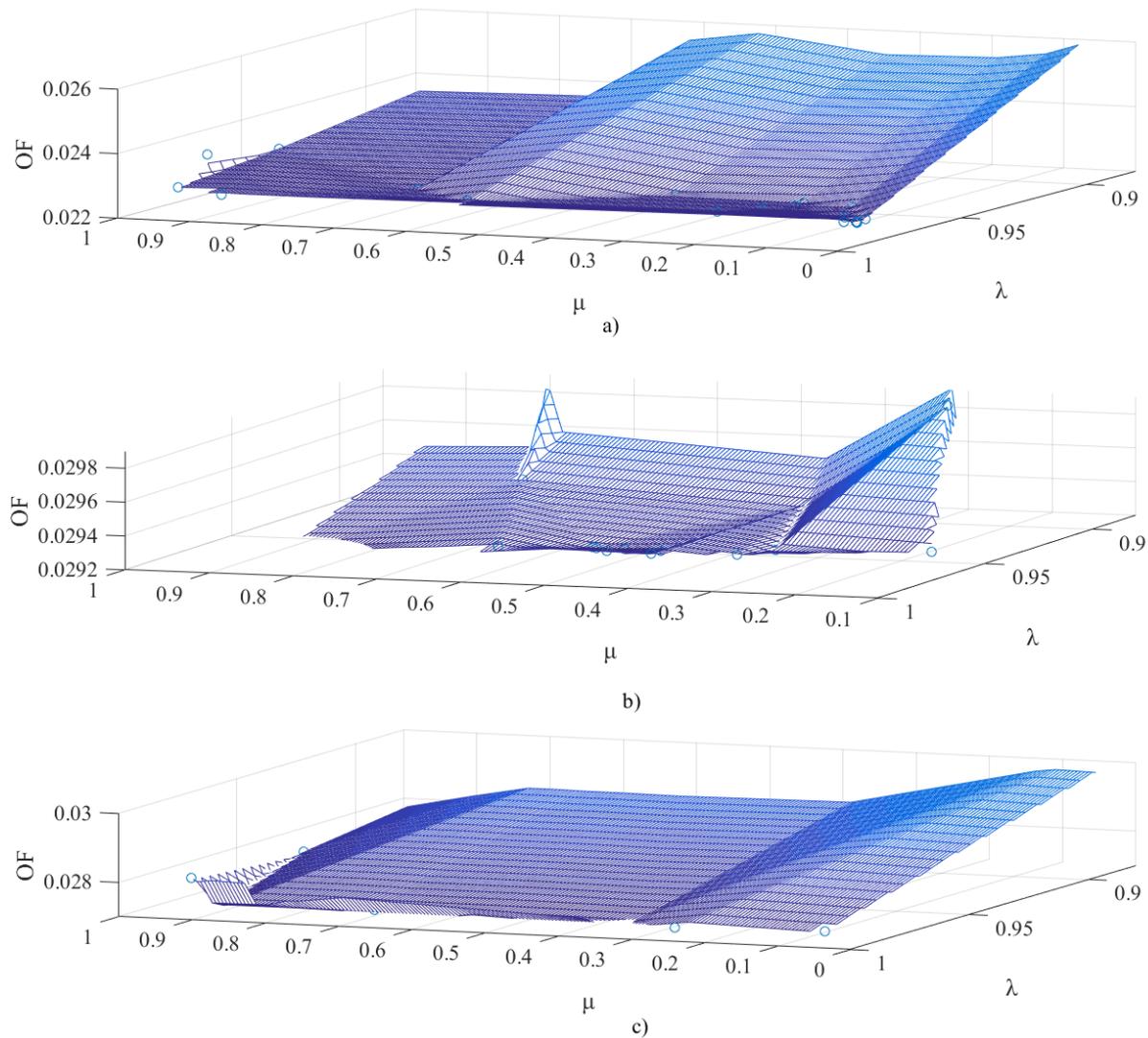


Fig. 5.3 Close-up view of optimization results for three cases: a) Case 1, b) Case 2, and c) Case 3.

The block diagram of the system is as in Fig. 5.1, with the only difference that the parameters of the FOPID controller are chosen based on Case 3 tuned parameters obtained from the MNM algorithm.

Table 5.3 Experimentally Tuned Parameters for the healthy control system.

Case	Value				
	K_p	K_i	K_d	μ	λ
$3_{average}$	4.53	89.28	0.15	0.9315	0.8374
Tuned	4.50	81.5	0.10	0.9187	0.9835

A simulation has been done with the sampling frequency of 1kHz in pre-generated reference mode. The results of the simulation are shown in Fig. 5.4. Furthermore, a correlated

fractional pink noise with the power of 0.001 and frequency of 1kHz has been added to the simulation output to study the controller's performance.

The reference signal for the velocity in the simulation incorporates four pulses with the magnitude of 0.05 m/s. Starting from the home position where the arm is vertical, this reference brings up the arm to emulate a dig and dump task. This maneuver ends with the arm goes back to the home position.

As shown in Fig. 5.4, the FOPID controller is able to control the velocity well. This is further indicated by the Root Mean Square Error (RMSE) of 0.0132 m/s and the maximum overshoot of 0.00418 m/s. Also, Fig. 5.4 shows that during the operations, while there are fluctuations in the pressure signal, the controller maintains the fluctuation in the velocity signal. Thus, the position signal is smooth, and the maximum control input is 4V. In fact, the RMS of jerk is 0.0836 m/s³, and the jerk does not exceed 0.6 m/s³ during the whole simulation.

Using the experimental test rig of Fig. 2.3, the FOPID controller with the MNM tuned parameters was tested for various loads and velocity reference profiles. The results of the experiments are shown in Fig. 5.5 and Fig. 5.6 for a 0.05 m/s and 0.07 m/s maximum reference velocity, respectively, each for weights of 0 kg, 163 kg and 367 kg.

As shown, the controller is able to track the reference with limited oscillation and overshoot in the velocity trajectory which inherently exists in the system because of the switching nature of the system. Also, oscillations in the position trajectory are minute. The RMSEs for all experimental tests are compared in Table 5.4. These results show that the controller is able to track the desired velocity with an insignificant RMSE of 0.0176 m/s for the worst-case scenario (full load and high velocity) without exceeding the input voltage of approximately 4 V. Also, the observed performance and consistent low RMSEs for different scenarios show that the controller is robust towards the disturbance, parameter uncertainties and model changes.

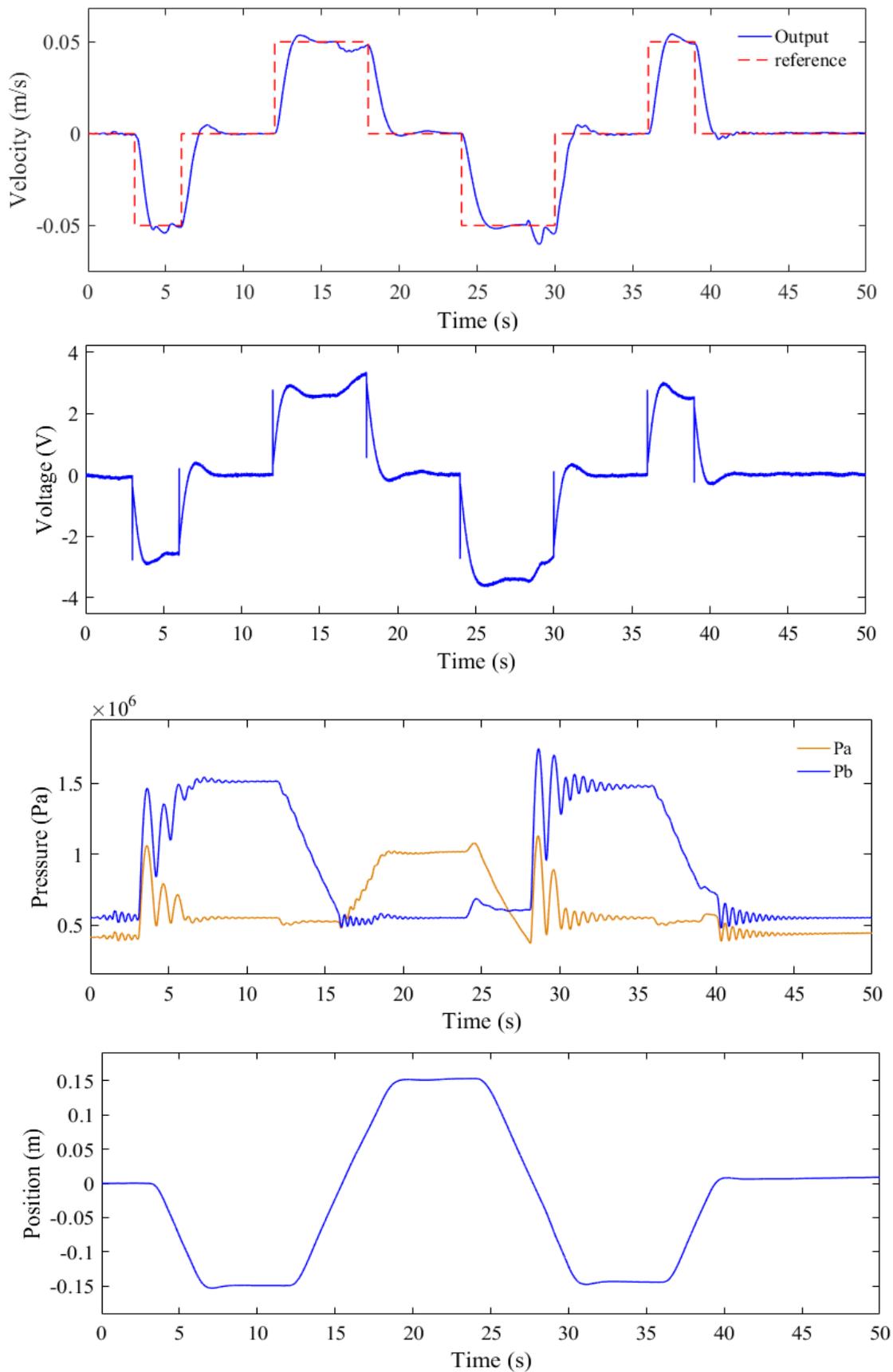


Fig. 5.4 Results of simulation for fractional-order PID control in pre-generated reference mode.

Table 5.4 RMS errors of experimental tests.

Velocity ($\frac{m}{s}$)	Weight (Kg)		
	0	163	367
0.05	0.0114	0.0114	0.0117
0.07	0.0173	0.0173	0.0176

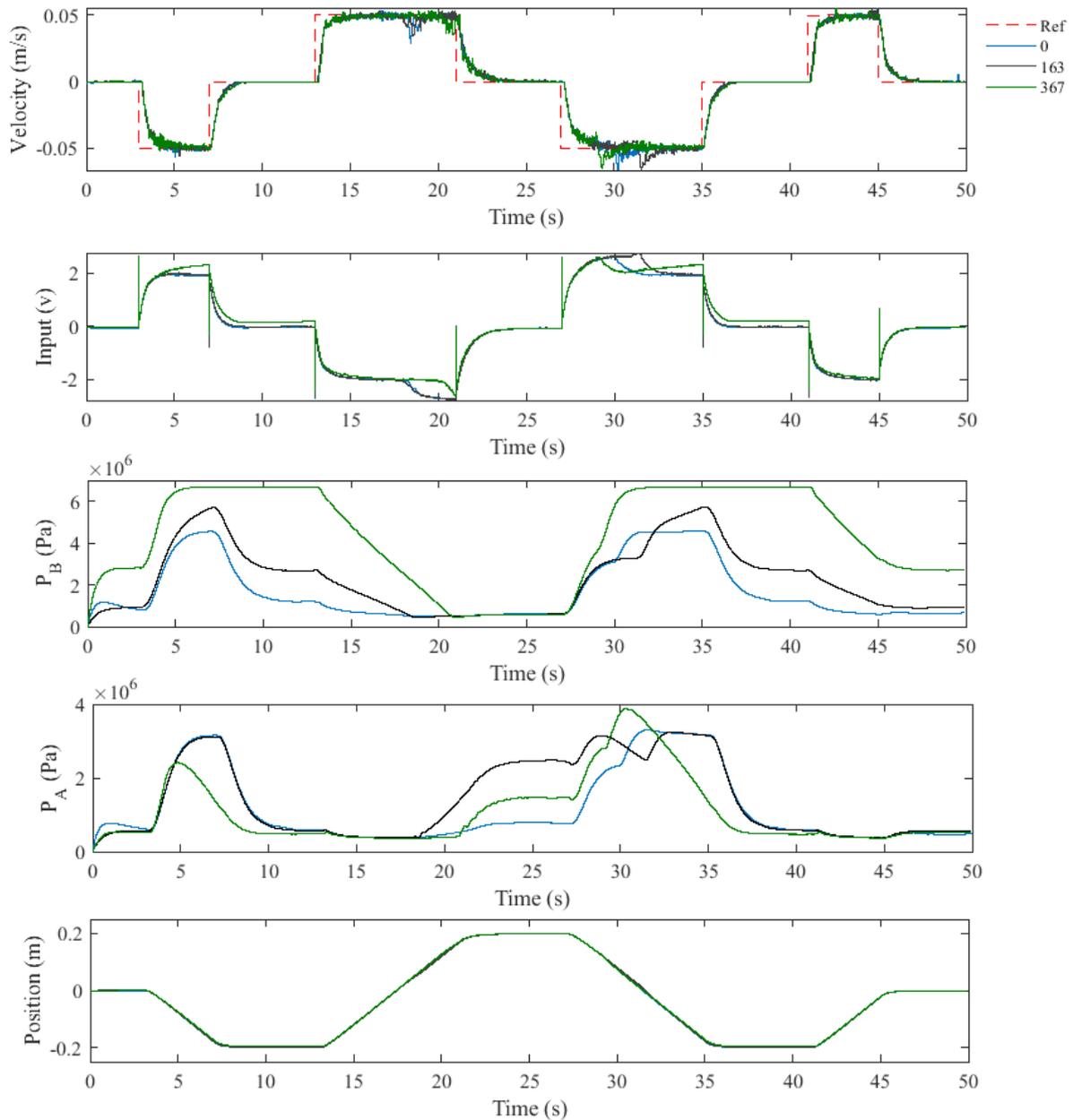


Fig. 5.5 Experimental results for fractional-order PID control in pre-generated reference mode with reference velocity of 0.05 m/s, weights of 0 kg, 163 kg and 367 kg.

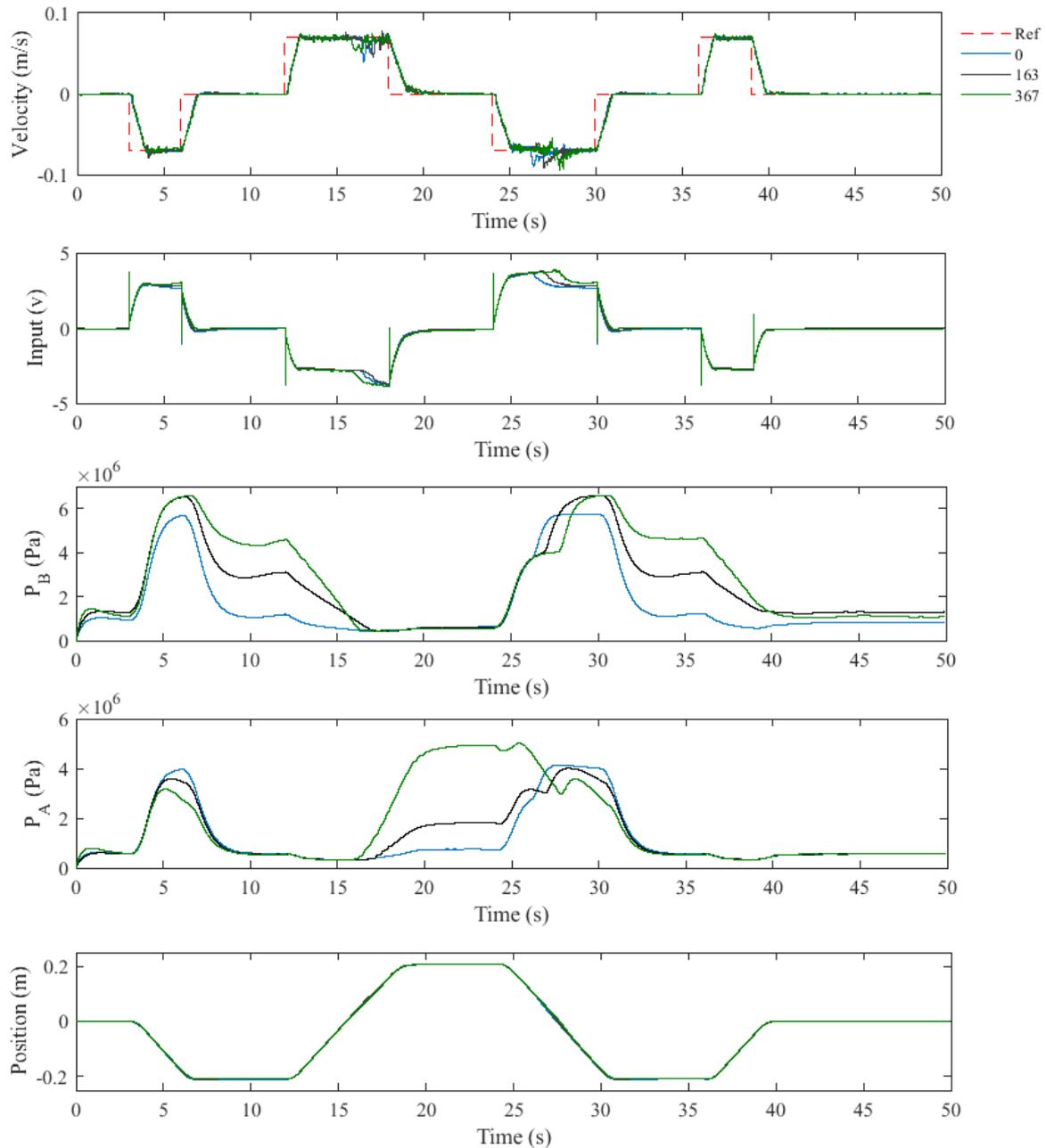


Fig. 5.6 Experimental results for fractional-order PID control in pre-generated reference mode with reference velocity of 0.05 m/s, weights of 0 kg, 163 kg and 367 kg.

5.4. Design of The Fault-tolerant Controller using Fuzzy Fractional-order PID Control

5.4.1. Parameter Tuning for Fractional-order PID Control in The Presence of Internal Leakage Fault

In this section, first, Assuming the system has severe internal leakage, the modified Nelder-Mead algorithm is used to optimize the parameters of a new FOPID controller, for velocity control of this system.

The optimization has been performed for $R = 6$ runs and $N_{max} = 100$ maximum iterations. Case 3 weights ($w_E = 1$, $w_J = 0.2111$) are used for the objective function. The remaining parameters of the experimental optimization are as described in Table 5.1. Table 5.5 shows the value of objective functions for each of the runs of the MNM algorithm.

Table 5.5 Results of online optimization of the system with internal leakage.

Run	K_p	K_i	λ	K_d	μ	Objective function
1	21.3	273.9	0.912	4.15	0.729	0.231
2	18.4	310.3	0.903	4.20	0.727	0.217
3	11.3	373.9	0.908	3.68	0.721	0.206
4	18.4	343.1	0.899	5.07	0.716	0.222
5	13.3	368.5	0.914	3.97	0.713	0.201
6	10.3	344.6	0.919	5.04	0.711	0.213

Based on the parameters obtained from the MNM algorithm, a FOPID controller was designed and further tuned by trial experiments to compensate for the many different nonlinearities and uncertainties in the system. Thus, the FOPID parameters are further adjusted and chosen as follows in Table 5.6.

Table 5.6 Experimentally Tuned Parameters for the control system with internal leakage.

Case	Value				
	K_p	K_i	K_d	μ	λ
Best run of MNM	13.312	368.480	3.856	0.8813,	0.7194
Further tuned	10	370	4	0.9	0.7

Comparing the final parameters chosen for the healthy and faulty condition FOPID controllers, two observations can be made. First, the gain parameters have all increased with the integral gain having the largest change. This shows that in a system with internal leakage a higher accumulated input to the motor is required to reach the same velocity. Second, while the derivator exponent has remained almost unchanged, the integrator exponent has decreased by approximately 24.6%. This is a result of an increase in integral gain; with a higher gain of integration, the jerkiness and overshoots in the velocity trajectory increases. By lowering the magnitude of the integration exponent, these oscillations are mitigated. It is worth noting that the inclusion of the jerk RMS in the objective function of the optimization has contributed to obtaining these tuned parameters.

5.4.2. Fault-tolerant Velocity Control in The Presence of Internal Leakage Fault

The design of the fault-tolerant control includes the simultaneous operation of two main strategies as shown in Fig. 5.7. the performance of each of these two strategies is dependent on the other. The first strategy is the *Fault Detection Strategy* which includes the calculations of fault detection measures (variance fractal dimension, length fractal dimension and wavelet detail coefficients) and derivation of the Degree of Fault (DoF) variable from these measures using the *fault decision algorithm*. The working principles of this block are described in Chapter 4.

The second strategy is the *Control Strategy* which includes the fault-tolerant control of the system using the two optimized FOPID controllers and the DoF variable obtained from the *Fault Detection Strategy*. The bottleneck in the control strategy is the *FIS block*. The Fuzzy Inference System (FIS) here is a simple three rule fuzzy system which translates different DoFs to different weights in the final input to the EHA, U , for the inputs coming from the Healthy Condition FOPID Controller, U_H , and the Faulty Condition FOPID Controller, U_F . In other words, if a high level of internal leakage is indicated by DoF, the weight of the input from faulty condition FOPID controller is set higher by the FIS.

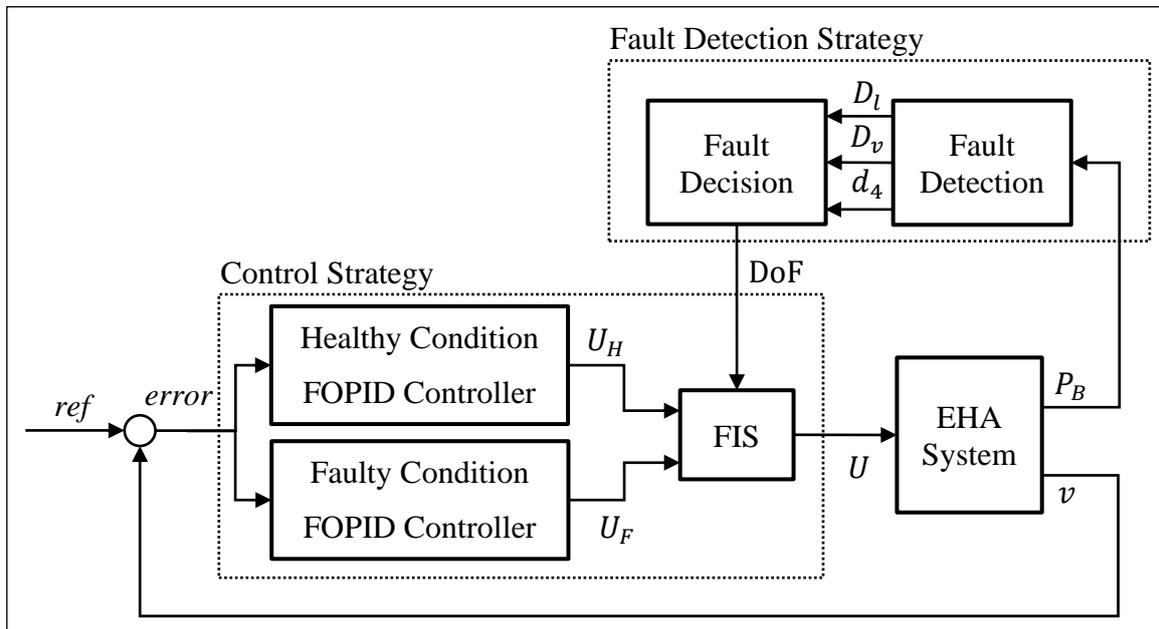


Fig. 5.7 Block diagram of fuzzy-FOPID fault tolerant control scheme.

The rule base of this FIS in Takagi Sugeno [17] is as follows:

1. If DOF is μ^- then $U_1 = U_H$.
2. If DOF is μ^+ then $U_2 = (U_H + U_F)/2$.
3. If DOF is μ^{++} then $U_3 = U_F$.

The triangular membership functions of this rule base are described in Fig. 5.8.

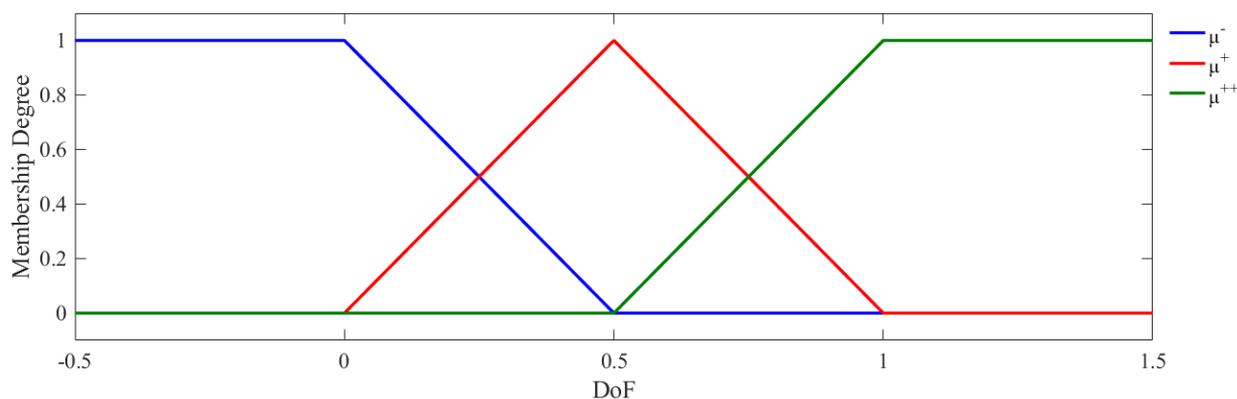


Fig. 5.8 Inputs membership functions of the fuzzy inference system for fault-tolerant control strategy.

The final defuzzified output of the FIS, which is the control input to the EHA, U , is decided based on the degree of fault in the system and is obtained as follows:

$$U = \frac{\mu^-(DoF)U_1 + \mu^+(DoF)U_2 + \mu^{++}(DoF)U_3}{\mu^-(DoF) + \mu^+(DoF) + \mu^{++}(DoF)} \quad (62)$$

The fault detection which is incorporated in the DoF variable which is the input to in the consequent of the rule base of FIS, endows the proposed fuzzy-FOPID control scheme with fault-tolerant characteristics.

The FIS receives the DoF after it has continuously detected a fault for 20 seconds. This value of DoF at the 20 seconds mark is noted as DoF_{set} . Once a transition has been made to the faulty condition (after the 20s window), the fault detection algorithm will remain dormant and the system will keep performing within the acceptable design criteria by the use of this fault-tolerant control scheme. When the fault is fixed and the required maintenance has been performed on the EHA, the fault detection algorithm is brought back online until the next occurrence of the fault and activation of the FIS.

5.4.3. Experiments

Several experiments with varying degrees of internal leakage have been performed on the system as shown in Fig. 5.9 and Fig. 5.10 for severe leakage, Fig. 5.11 and Fig. 5.12 for medium leakage and finally Fig. 5.13 for small leakage.

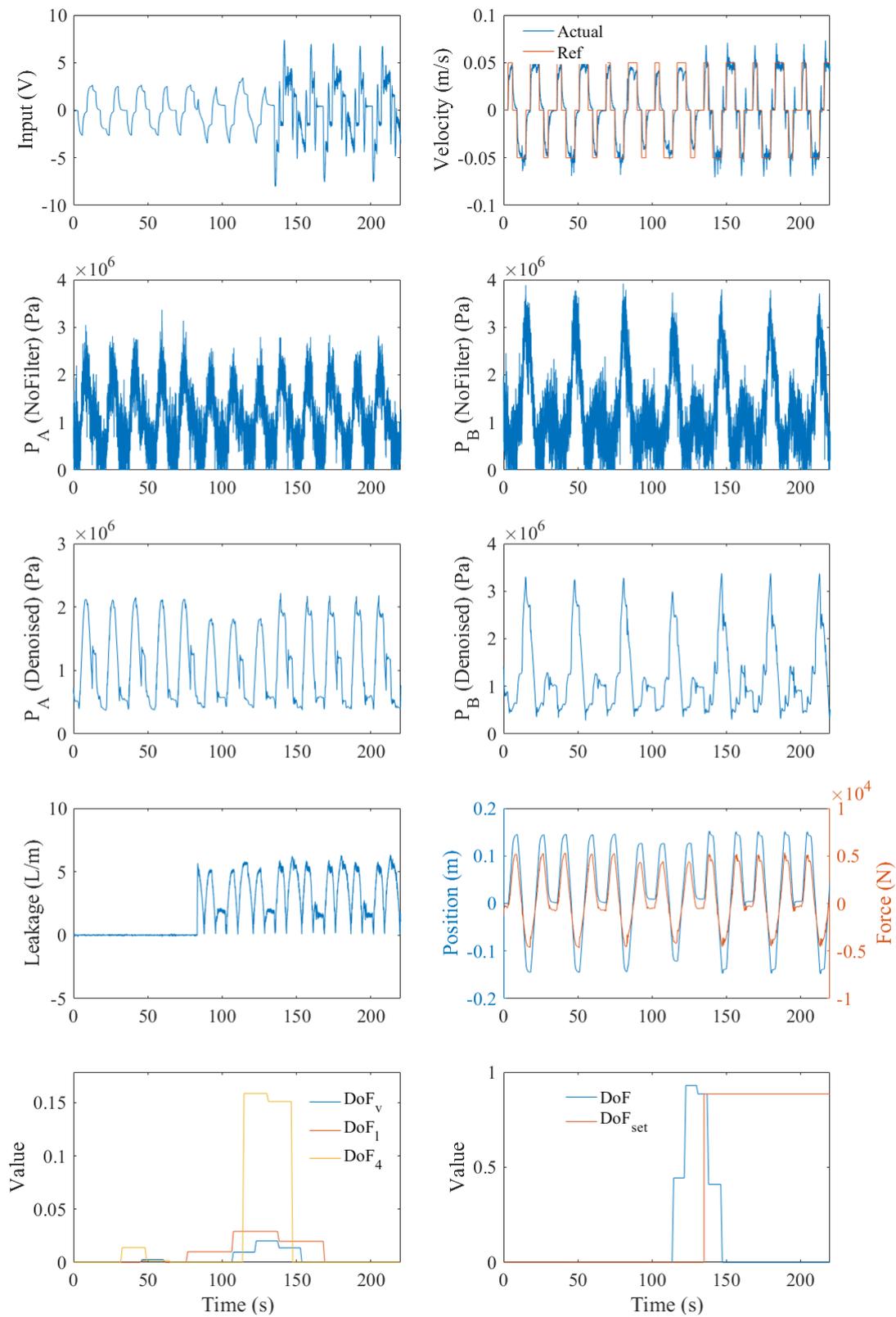


Fig. 5.9 Results of the fault-tolerant control experiment in pre-generated reference mode, test 1.

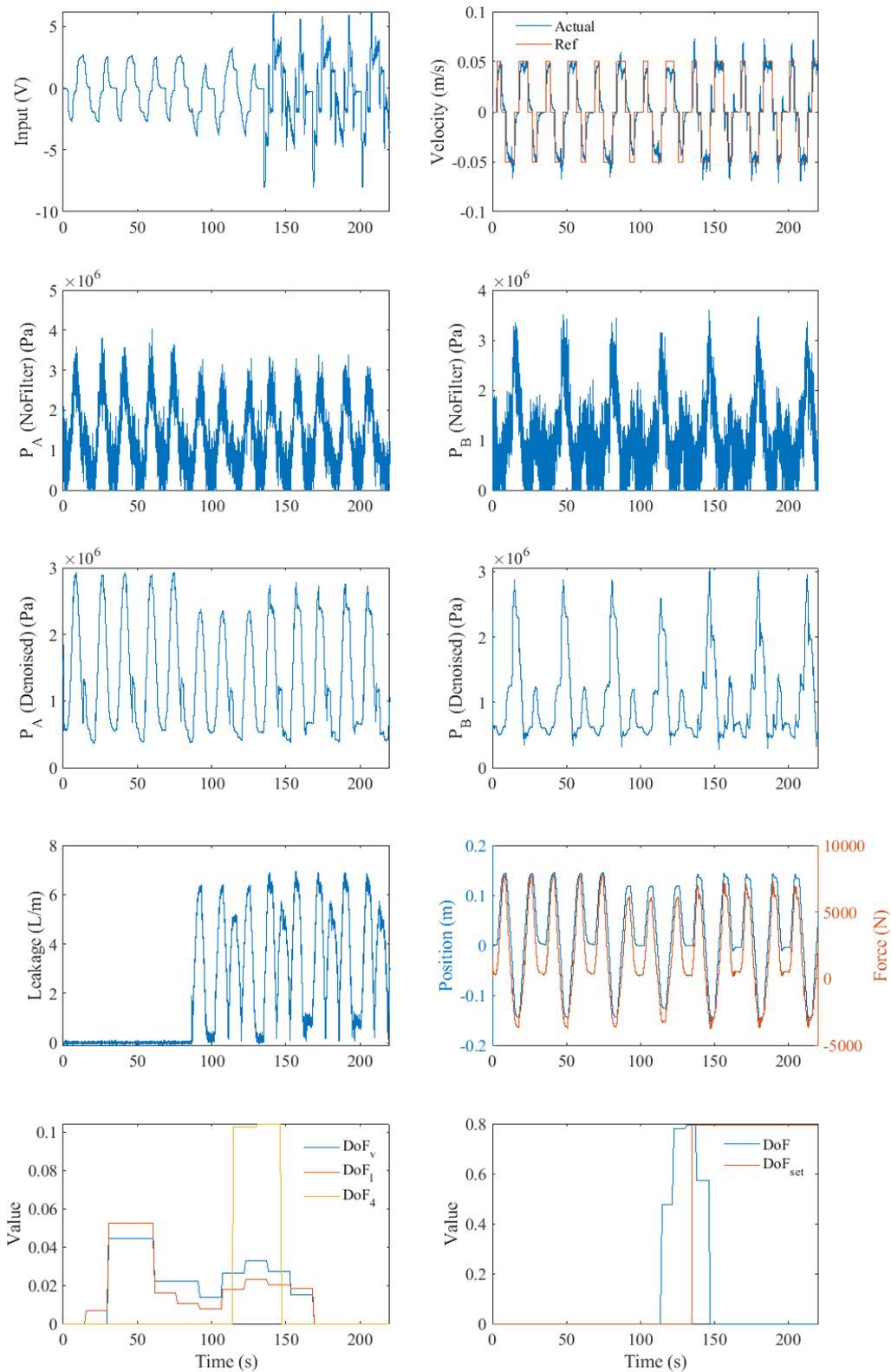


Fig. 5.10 Results of the fault-tolerant control experiment in pre-generated reference mode, test 2.

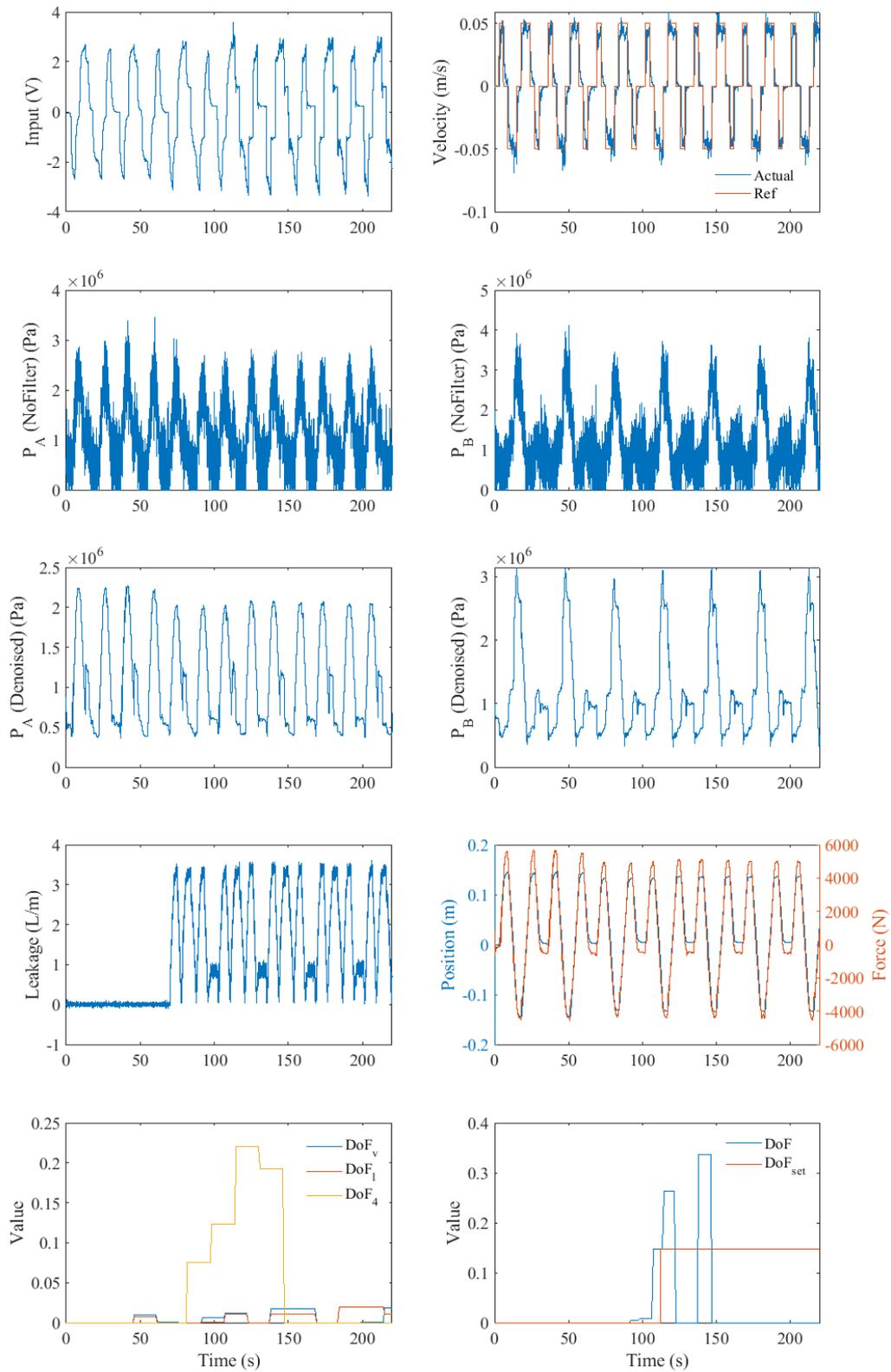


Fig. 5.11 Results of the fault-tolerant control experiment in pre-generated reference mode, test 3.

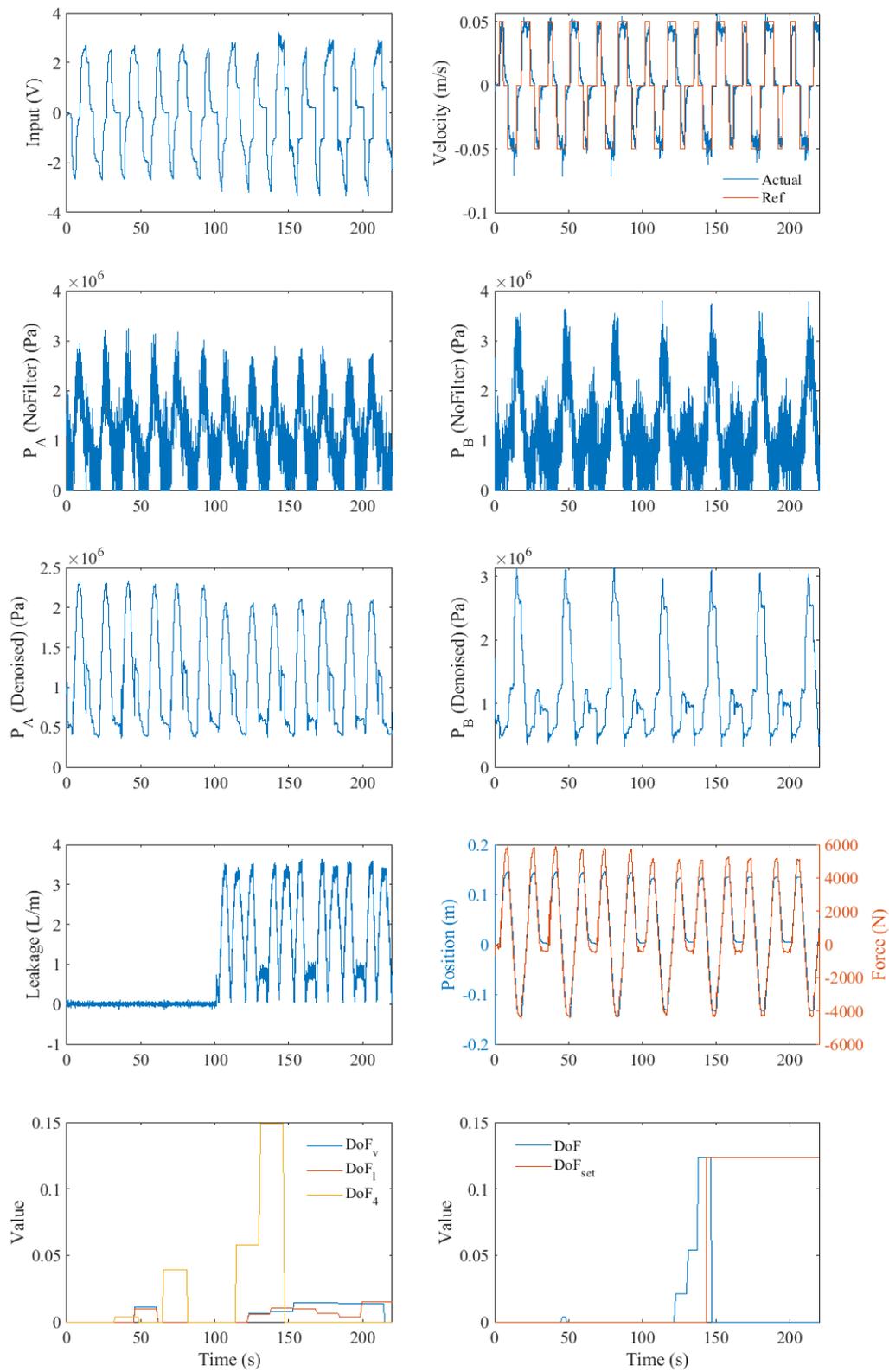


Fig. 5.12 Results of the fault-tolerant control experiment in pre-generated reference mode, test 4.

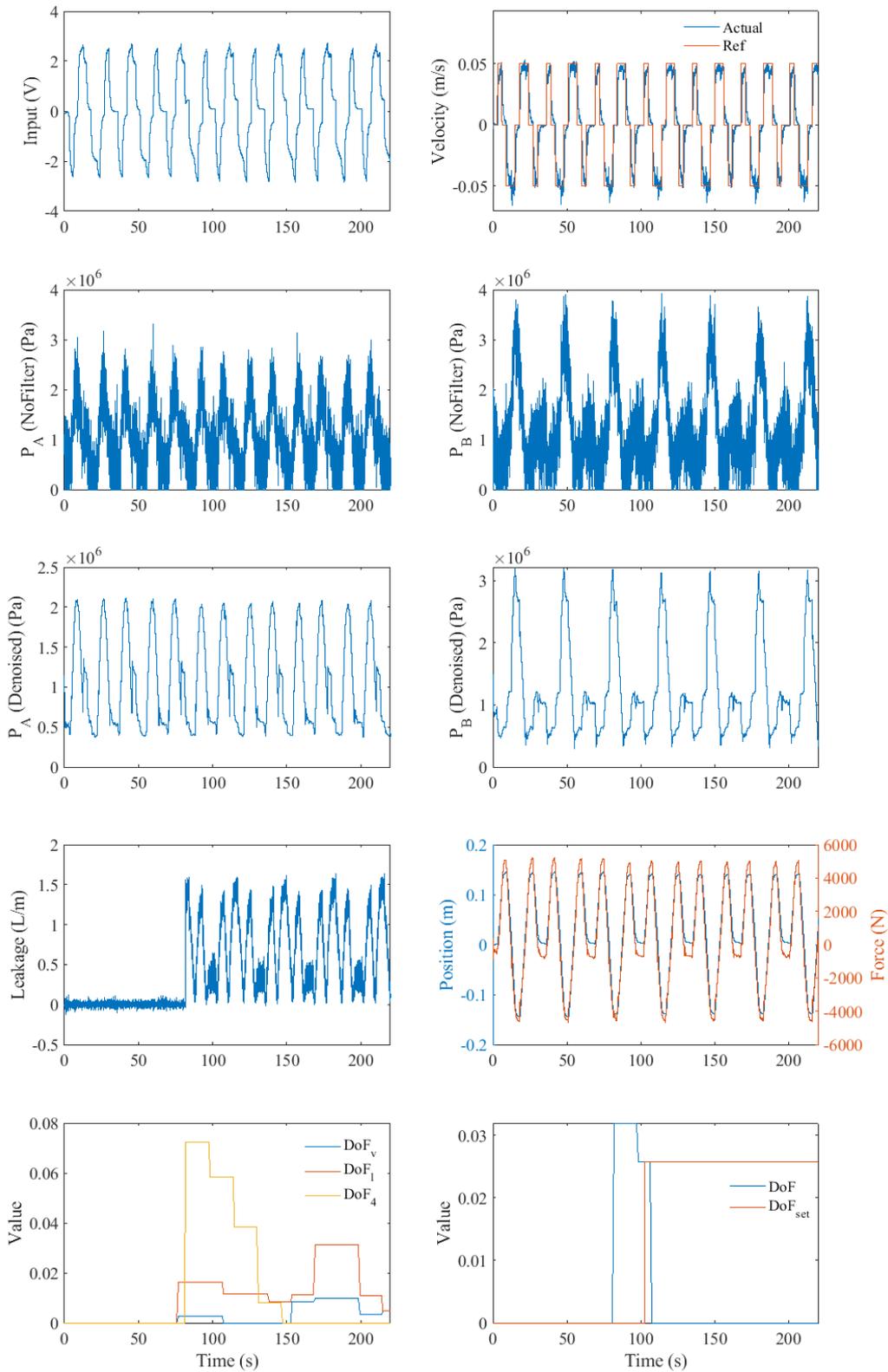


Fig. 5.13 Results of the fault-tolerant control experiment in pre-generated reference mode, test 5.

As shown, while the system is running in healthy condition the FOPID controller designed for healthy condition is able to track the velocity reference. When the internal leakage

fault occurs, the performance of the system deteriorates and the steady-state error appears in the performance of the controller. At this point, the fault decision algorithm indicates the existence of fault and after 20 seconds of consistent indication, the FIS algorithm changes to the weighted sum of the faulty condition and healthy condition FOPIDs.

The DoF variable set by the FIS is proportional to the degree of fault as expected. For a leakage of 7 L/min, the DoF variable is approximately 0.8; for a leakage of 3.2 L/min the DoF variable is set at 0.15; for a small leakage of 1.5 L/min, which is close to the sensitivity margin of the fault decision algorithm¹, the fault is detected with a small DoF set at approximately 0.07.

It should be noted that, as shown, while a longer wait time (more than 20s) for setting the DoF variable, is adjustable based on the working environment of the EHA, but the DoF variable has small variation after this point as 20s covers more than one period of the indicators' behaviours. This wait time has been chosen based on experimental trials.

As seen in the results, by comparing the original pressure signals to the denoised signals it is shown that the denoising method is able to remove the noise from the signal without distortion while keeping their important characteristics and components.

With regards to comparing the FOPID performances in different conditions, while the healthy controller is still active and the system is experiencing a fault, the tracking capability has deteriorated and the steady-state error appears in the velocity trajectory. As an instance, this can be seen in Fig. 5.9 after $t=94s$. Also, the pressure levels relatively drop which again points to the loss of control performance. When the FIS algorithm activates the faulty condition controller, the magnitude of the input and its sensitivity increases. Consequently, the tracking performance is recovered, the steady-state error is attenuated, and the pressure levels rise. As an instance, this can be seen in Fig. 5.9 after $t=132s$. This is also shown in the force-position plots where the relative force to position of the system has recovered by the activation of the FIS. In turn, the increase of sensitivity and magnitude (which is set by design) in the input behaviour means that compared to the healthy system, while not noticeable in position trajectory, the velocity trajectory has small overshoots. However, this is within the acceptable performance and it is well expected that the faulty condition system will still perform well, but not as well as the healthy system.

¹ See Section 4.4.

5.5. Summary

In this chapter, a healthy FOPID controller is designed and optimized using the MNM algorithm for velocity control of the single-rod EHA. This control scheme is successful in tracking the desired velocity in healthy condition. Also, another FOPID controller is designed and optimized for severe internal leakage conditions.

Based on these two FOPID controllers and the DoF variable, which was developed in Chapter 4, a fault tolerant control scheme is proposed for internal leakage. This is done by utilizing a FIS which based on different DoFs, assigns different weights to the input from each of the FOPID controllers. In other words, if a higher level of internal leakage is indicated by DoF, the weight of the input from faulty condition FOPID controller is set higher by the FIS.

The results of experiments show that if an internal leakage fault occurred the fault-tolerant controller is able to restore the performance of the system and prevent steady-state errors.

CHAPTER 6:

CONCLUSIONS

In this thesis, by acquiring accurate real-time signals from a single-rod Electrohydrostatic Actuator (EHA) a fault detection scheme has been designed using variance fractal dimension, length fractal dimension, and wavelet detail coefficients. Internal leakage and bulk modulus change faults were studied extensively, and the measures were associated with different levels of faults.

The developed fault detection algorithm along with two Fractional-order Proportional Integral Derivative (FOPID) controllers, one for healthy condition and the other for faulty conditions, parallel to a fuzzy inference system led to the development of a fault-tolerant control for the single-rod EHA system. The fault-tolerant control strategy utilizes a fuzzy inference system, which assigns different weights to the input generated from each of the FOPID controllers based on the DoF in the system. This fault-tolerant control scheme proved efficient and capable of controlling the system in healthy and faulty conditions by detecting internal leakage fault and its relative severity.

6.1. Thesis Contributions

1. A fuzzy denoising method was developed in this work to acquire accurate signals and remove the noise. The developed denoising method, which is based on fuzzy thresholding, outperformed the conventional methods (soft thresholding and hard thresholding) in various experiments using speech signals, and pressure signals of the electrohydrostatic actuator. For a pressure signal with an original SNR of approximately 7 dB, the proposed fuzzy denoising method proved capable of reaching an SNR of approximately 30 dB.

2. A comprehensive analysis of the behaviour of pressure and jerk signals was performed in a multi-fault environment to detect bulk modulus change and internal leakage faults for the first time for the single-rod electrohydrostatic actuator. The analysis was performed using two polyscale measures (variance and length fractal dimensions) and one multiscale measure (wavelet detail coefficients). Also, detecting the severity of faults due to

internal leakage and changes in bulk modulus was investigated through experiments and simulations, respectively.

3. Based on the observed behaviours of the measures and pressure signals in the aforementioned comprehensive analysis, a fault decision algorithm was developed for the first time for the single-rod electrohydrostatic actuator. By obtaining indicators of the polyscale (length and variance fractal dimensions) and multiscale (wavelet detail coefficients) measures the algorithm generated a degree of fault variable which indicate the existence and severity of the internal leakage fault in the single-rod electrohydrostatic actuator. In the worst-case scenario, the algorithm indicated the existence of the internal leakage fault 83.2% of a 390s fault window with a minimum sensitivity of approximately 1.2 L/min.

4. A fault-tolerant control strategy was developed based on two FOPID controllers and a fuzzy inference system for the first time for the single-rod electrohydrostatic actuator. Using the modified Nelder-Mead optimization algorithm, one of the FOPID controllers was tuned for healthy condition and the other one was tuned for the system with severe internal leakage of up to 7 L/min. The fuzzy inference system was utilized to assign different weights to the inputs from the two FOPID controllers, based on the degree of fault in the system. The healthy condition FOPID controller was able to track the desired velocity with an RMSE of 0.0176 for the maximum load of 367 Kg and velocity of 0.07 m/s. Also, the fault tolerant control scheme proved capable of recovering tracking performance of the system and eliminated the steady-state error which appeared due to internal leakage.

6.2. Potential Future Work

1. Since the artificial generation of bulk modulus change faults was not possible in the experimental test rig of this work, this thesis concentrated on simulation results for bulk modulus change faults. In future works, an experimental investigation of the behaviour of the system with bulk modulus change could validate the results of this work.

2. While this work is concentrated on fault-tolerant control as the final contribution, a self-healing control scheme that could adjust the behaviour of the system to fix the fault would be a significant step in future works.

References

- [1] Gustavo Koury Costa and Nariman Sepehri, *Hydrostatic Transmissions and Actuators-Operation, Modelling and Applications*. Hoboken, NJ, USA: Wiley, 2015.
- [2] Gustavo Koury Costa and Nariman Sepehri, “Four-quadrant analysis and system design for single-rod hydrostatic actuators,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 141, no. 2, 021011, 2019.
- [3] Guangan Ren, Gustavo Koury Costa, and Nariman Sepehri, “Position control of an electro-hydrostatic asymmetric actuator operating in all quadrants,” *Mechatronics*, vol. 67, 102344, 2020.
- [4] Mark Karpenko, *Quantitative fault tolerant control design for a hydraulic actuator with a leaking piston seal*, Ph.D. Dissertation, Winnipeg, Manitoba: University of Manitoba, Department of Mechanical Engineering, 2008, 220 pages.
- [5] Ali Maddahi, Witold Kinsner, and Nariman Sepehri, “Internal leakage detection in electrohydrostatic actuators using multiscale analysis of experimental data,” *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 12, pp. 2734-2747, 2016.
- [6] Ali Maddahi, Nariman Sepehri, and Witold Kinsner, “Fractional-order control of hydraulically-powered actuators: controller design and experimental validation,” *IEEE Transactions on Mechatronics*, vol. 24, no. 2, pp. 796-807, 2019.
- [7] Chengbin Ma and Yoichi Hori, “Fractional-order control: Theory and applications in motion control [past and present],” *IEEE Industrial Electronics Magazine*, vol. 1, no. 4, pp.6-16, 2007.
- [8] Katsuhiko Ogata, *Modern Control Engineering*, Hoboken, NJ, USA: Prentice Hall, 2010.
- [9] Quanser Q8 DAQ datasheet, 2003 model, available at:
<http://sivirt.utsa.edu/Documents/Manuals/Q8%20Manual.pdf>
- [10] Steven Smith, *The Scientist and Engineer’s Guide to Digital Signal Processing*. San Diego, CA, USA: California Technical Publishing, 1997.
- [11] Stéphane Mallat, *A Wavelet Tour of Signal Processing: The Sparse Way*, 3rd ed. New York, NY, USA: Elsevier/Academic, 2009.

- [12] Lik-Kwan Shark and Chunyang Yu, “Denoising by optimal fuzzy thresholding in wavelet domain,” *Electronics Letters* 36, no. 6, pp. 581-582, 2000.
- [13] Jing-Yi Lu, Hong Lin, Dong Ye, and Yan-Sheng Zhang, “A new wavelet threshold function and denoising application,” *Mathematical Problems in Engineering*, vol. 2016, 2016.
- [14] David Donoho and Iain Johnstone, “Ideal spatial adaptation by wavelet shrinkage.” *Biometrika*, vol. 81, pp. 425–455, 1994.
- [15] David Donoho, “De-noising by soft-thresholding.” *IEEE Transactions on Information Theory*, vol. 42, no. 3, pp. 613–627, 1995.
- [16] Pankaj Hedao and Swati Godbole, “Wavelet thresholding approach for image denoising,” *International Journal of Network Security & Its Applications (IJNSA)*, vol. 3, no. 4, pp. 16-21, 2011.
- [17] Tomohiro Takagi and Michio Sugeno, “Fuzzy identification of systems and its applications to modeling and control,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 15, pp. 116–132, 1985.
- [18] Sina Sedigh and Witold Kinsner, “A Manitoban Speech Dataset,” *IEEE DataPort*, January 2018. {DOI: 10.21227/H2KM16}
- [19] K1 ASHCROFT Pressure datasheet, available at:
http://ashcroft.com/datasheet_pdf/upload/datasheet-k1-transducers.pdf
- [20] Bourns® rotary encoder datasheet, available at:
https://www.bourns.com/docs/default-document-library/enc1j.pdf?sfvrsn=2c0f9df1_0
- [21] P-Q Controls model 220 joystick datasheet, available at:
http://www.pqcontrols.com/assets/data/m220_datasheet.pdf
- [22] ADS1115 Texas Instruments datasheet, available at:
<https://www.ti.com/lit/gpn/ADS1115>
- [23] ADS1015 Texas Instruments datasheet, available at:
<https://www.ti.com/lit/gpn/ads1015>
- [24] Raspberry Pi4 raspberry Pi 4 datasheet, available at:
<https://www.raspberrypi.org/documentation/hardware/raspberrypi/>
- [25] Raspbian operating system documentation, available at:
<https://www.raspberrypi.org/documentation/raspbian/>

- [26] 1n4007 diode datasheet, available at:
<https://www.vishay.com/docs/88503/1n4001.pdf>
- [27] 2n2222A transistor datasheet, available at:
<https://www.st.com/resource/en/datasheet/cd00003223.pdf>
- [28] LM358 dual op-amp datasheet, available at:
<https://www.ti.com/lit/gpn/lm358-n>
- [29] LM 7905 positive voltage regulator datasheet, available at:
<https://www.ti.com/lit/ds/symlink/lm79.pdf>
- [30] MC 7805 negative voltage regulator datasheet, available at:
<https://www.onsemi.com/pub/Collateral/MC7800-D.PDF>
- [31] Ali Maddahi, *Fault-tolerant control of hydraulically-powered actuators using fractional-order PID schemes*, Ph.D. Dissertation, Winnipeg, Manitoba: University of Manitoba, Department of Mechanical Engineering, 2019, 168 pages.
- [32] Alireza Akhoondi Asadi, Shahriar Bagheri, Ahmed Imam, Ehsan Jalayeri, Witold Kinsner, and Nariman Sepehri, "A data acquisition system based on Raspberry Pi: Design, construction and evaluation," 2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference, Vancouver, BC, Canada, pp. 1-5, 2016.
- [33] Jonathan Valvano, *Embedded Microcomputer Systems: Real Time Interfacing*. Pacific Grove, CA, USA: Brooks/Cole, 2000.
- [34] David Dickey and Wayne Fuller, "Distribution of the estimators for autoregressive time series with a unit root," *Journal of the American Statistical Association*, vol. 74, no. 366a, pp. 427–431, 1979.
- [35] Rainer von Sachs and Michael Neumann, "A wavelet-based test for stationarity," *Journal of Time Series Analysis*, vol. 21, no. 5, pp. 597-613, 2000.
- [36] Guy Nason, "A test for second-order stationarity and approximate confidence intervals for localized autocovariances for locally stationary time series," *Journal of the Royal Statistical Society Series B*, vol. 75, no. 5, pp. 879-904, 2013.
- [37] Witold Kinsner, *Fractal and Chaos Engineering: Monoscale, Multiscale and Polyscale Analyses*. Winnipeg, MB, Canada: OCO Research, February 2020, 1106 pages. {ISBN: 978-0-9939347-2-8, eBook}

- [38] Witold Kinsner and Warren Grieder, "Speech segmentation using multifractal measures and amplification of signal features," 2008 7th IEEE International Conference on Cognitive Informatics, Stanford, CA, USA, pp. 351-357, 2008.
- [39] Alain Oustaloup, Francois Levron, Benoit Mathieu, and Florence Nanot, "Frequency-band complex noninteger differentiator: characterization and synthesis," IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, vol. 47, no. 1, pp. 25–39, 2000.
- [40] Amirreza Mirbeygi Moghaddam, Witold Kinsner, Gustavo Koury Costa, Lokesh Kumar, Khurram Butt, and Nariman Sepehri, "FOPID control with a modified Nelder-Mead parameter optimization of an EHA," IEEE Instrumentation & Measurement Magazine, vol. 24, no. 2, pp. 109-117, 2021.
- [41] Yangquan Chen, Blas Vinagre, and Igor Podlubny, "Continued fraction expansion approaches to discretizing fractional order derivatives – an expository review," Nonlinear Dynamics, vol. 38, pp. 155-170, 2004.
- [42] John Nelder and Roger Mead, "A simplex method for function minimization," The Computer Journal. vol. 7, no. 4, pp. 308-313, 1965.
- [43] J. A. Guin, "Modification of complex method of constrained optimization," The Computer Journal, vol. 10, no. 4, pp. 416-417, 1968.
- [44] Khurram Butt, Ramhuzaini Abd Rahman, Nariman Sepehri, and Shaahin Filizadeh, "Globalized and bounded Nelder-Mead algorithm with deterministic restarts for tuning controller parameters: Method and application," Optimal Control Applications and Methods, vol. 38, pp. 1042–1055, 2017.

Appendix A

Fig. A.1 shows the movement of the excavator arm from position 1 to position 2. Considering the triangle in position 1, C can be obtained as follows:

$$C = \sqrt{L^2 - l_1^2} \quad (63)$$

Also, the angle, ϕ , can be obtained from position 1 as follows:

$$\phi = \arccos\left(\frac{l_1}{C}\right) \quad (64)$$

Angle, $(\phi + \theta)$, in the triangle in position 2 can be obtained as follows:

$$(L + x_p)^2 = l_1^2 + C^2 - 2l_1C\cos(\phi + \theta) \quad (65)$$

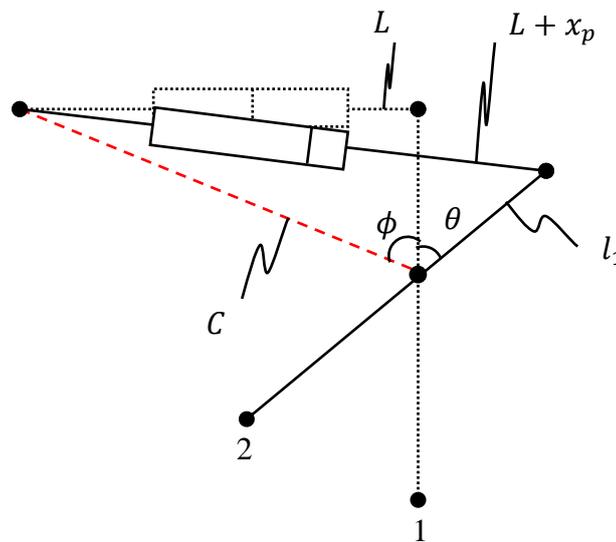


Fig. A.1 Schematic of the movement of the excavator arm.

Using (64) and simplifying $C\cos(\phi + \theta) = l_1 \cos(\theta) - L \sin(\theta)$, (65) can be rewritten as follows:

$$(L + x_p)^2 = l_1^2 + C^2 - 2l_1^2 \cos(\theta) + 2l_1L \sin(\theta) \quad (66)$$

Equation (66) directly relates θ to x_p .