12-2015

Gain tuning of proportional integral controller based on multiobjective optimization and controller hardware-in-loop microgrid setup

Kumaraguru Prabakar
University of Tennessee - Knoxville, kprabaka@vols.utk.edu

To the Graduate Council:

I am submitting herewith a dissertation written by Kumaraguru Prabakar entitled "Gain tuning of proportional integral controller based on multiobjective optimization and controller hardware-in-loop microgrid setup." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Energy Science and Engineering.

Fangxing Li, Major Professor

We have read this dissertation and recommend its acceptance:

Leon Tolbert, Kevin Tomsovic, Rapinder Sawhney

Accepted for the Council:

Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

Gain tuning of proportional integral controller based on multiobjective optimization and controller hardware-in-loop microgrid setup

A Dissertation Presented for the

Doctor of Philosophy

Degree

The University of Tennessee, Knoxville

Kumaraguru Prabakar

December 2015

This dissertation is dedicated to my mother, father, and my teachers.

Acknowledgements

I wish to thank Dr. Fangxing Li for guiding me throughout all stages of my dissertation study. My thanks and appreciation to Dr. Bailu Xiao for her technical guidance and assistance. I would also like to thank Tom Rizy, Dr. Sarina Adhikari, Kevin Dowling, Dr. Yan Xu, Dr. Michael Starke, Dr. Mark Buckner, Philip Irminger, Ben Ollis, and Dan King for sharing their knowledge, and support throughout my time at ORNL.

I am grateful for having wonderful friends who supported me during my hardship and made this dissertation possible.

Abstract

Proportional integral (PI) control is a commonly used industrial controller framework. This PI controller needs to be tuned to obtain desired response from the process under control. Tuning methods available in literature by and large need sophisticated mathematical modelling, and simplifications in the plant model to perform gain tuning. The process of obtaining approximate plant model conceivably become time consuming and produce less accurate results. This is due to the simplifications desired by the power system applications especially when power electronics based inverters are used in it. Optimal gain selection for PI controllers becomes crucial for microgrid application. Because of the presence of inverter based distributed energy resources.

In the proposed approach, a multi-objective genetic algorithm is used to tune the controller to obtain expected step response characteristics. The proposed approach do not need simplified mathematical models. This prevents the need for obtaining unfailing plant models to maintain the fidelity of modelling. Microgrid system and the PI controller are modelled in different software, hardware platform and tuned using the proposed approach. Gain values for PI controller in these different platform are tuned using the same objective function and multi-objective optimization. This proves the re-usability, scalability, and modularity of the proposed tuning algorithm.

Three different combination of software, hardware platform are proposed. First, the process and the PI controller are modelled in a computer based hardware. In order to increase the speed of the multi-objective optimization in the computer based hardware parallel computing is employed. This is a natural fit for paralleling the GA based optimization. Second, both the plant and control representation are modelled in the real time digital simulator (RTDS). Finally, a controller hardware in loop platform is used. In this platform, the plant will be modelled in RTDS and the PI controller will be modelled in an FPGA based hardware platform. Results indicate that the proposed approach has promising potentials since it does not need to simplify the switching model and can effectively solve the complicated tuning procedure using parallel computing. Similar advantage could be said for RTDS based tuning because RTDS simulates the models in real time.

Table of Contents

1 Introduction **1**

1.1 Types of sources in microgrids . . . 2

1.2 Use of microgrids in the power system . . . 3

1.2.1 Use of microgrids in military bases . . . 3

1.2.2 Use of microgrids during emergency by utility services . . . 3

1.3 Challenges in microgrid implementation . . . 4

1.3.1 Challenges in protection . . . 4

1.3.2 Challenges in communication protocols . . . 5

1.3.3 Challenges in control . . . 6

1.4 Dissertation outline . . . 12

2 Literature review **13**

2.1 Introduction . . . 13

2.1.1 Proportional-Integral controller . . . 13

2.1.2 Response characteristics of a system . . . 14

2.1.3 Performance criteria . . . 15

2.1.4 Stability margins (Gain margin - A_m and Phase margin - ϕ_m) . . . 16

2.2 Literature review . . . 17

2.2.1 Survey papers on gain tuning . . . 30

2.2.2 Optimization based methods for gain tuning . . . 30

2.3 Proposed tuning methodology . . . 34

2.4 Summary . . . 34

3 Software based gain tuning with parallel computing **37**

3.1 Genetic algorithm based single objective optimization . . . 37

3.1.1 Selection rules . . . 38

3.1.2 Elite individuals . . . 39
3.1.3 Crossover for recombination . . . 39
3.1.4 Mutation . . . 39
3.1.5 Stopping criteria . . . 39
3.1.6 Outline of genetic algorithm . . . 40
3.2 Simulated annealing based single objective optimization . . . 41
3.3 Introduction to multiple objective optimization . . . 42
3.4 Genetic algorithm based multiple objective optimization . . . 44
3.5 Parallel computing based optimization . . . 44
3.6 Models used for gain tuning problem in different software and hardware platforms . . . 46
3.6.1 Process and control model used in computer based simulation . . . 46
3.6.2 Model used in real time digital simulator based process and control model . . . 49
3.7 Proposed gain tuning methodology . . . 50
3.7.1 Objective function formulation for optimization . . . 52
3.8 Summary . . . 54

4 Results from software based gain tuning **55**
4.1 Comparison of single objective and multiple objective optimization results . . . 55
4.2 Function space comparison . . . 56
4.3 Speed up and efficiency for different process models . . . 56
4.4 Summary . . . 77

5 Models for controller hardware-in-loop based gain tuning **78**
5.1 Sample Microgrid inverter model with current and voltage control . . . 78
5.1.1 Real time digital simulator based microgrid model . . . 79
5.1.2 Inverter control description . . . 80
5.1.3 Experimental setup . . . 80
5.1.4 Objectives used in PQ and VF controls gain tuning . . . 82
5.1.5 Noise generated by real time digital simulator . . . 84
5.2 Summary . . . 85

6 Results from controller hardware-in-loop based gain tuning **86**
6.1 Results . . . 86
6.1.1 Multiple inverter tuning . . . 97
6.2 Alternate ways to visualize Pareto front . . . 101
6.3 Summary . . . 103

7 Conclusions and future work **104**

7.1 Proposed gain tuning methodology 104

7.2 Controller hardware in loop based gain tuning 105

7.3 Summary of the advantages of the proposed method 105

7.4 Future work 106

Bibliography **107**

Vita **118**

List of Tables

6.1 Gain value range for the different control methods tuned using CHIL setup 90
6.2 The π model paramenters used in the RTDS multi-inverter case 97

List of Figures

1.1 Electricity generation by energy source 2
1.2 Flowchart indicating the microgrid switch controller islanding and grid-connected mode transition 7
1.3 Transmission system and its respective phasor diagram 8
1.4 Inverter control based on current control technique 9
1.5 Inverter control based on active reactive power control technique 10
1.6 Inverter control based on active power and voltage feedback technique 11

2.1 Typical PI controller structure for process control 14
2.2 Step response of a system 14
2.3 Structure of proposed serial-parallel computing and single-multiple objective approach for solving gain tuning optimization problem 36

3.1 Pictorial representation of decision space and objective space 42
3.2 Parallel computing flow chart for two workers 45
3.3 Process models used for tuning in computer based modelling 47
3.4 Photovoltaic model used for gain tuning tuning with voltage regulator 48
3.5 Detailed structure of the voltage controller used in the Photovoltaic gain tuning model 48
3.6 Plant model used in the real time digital simulator 49
3.7 Generic optimization flow for gain tuning 50
3.8 Gain tuning flow chart using multi-objective optimization and real time simulator . 51

4.1 Results from genetic algorithm based single objective optimization of models 1-4 . . 58
4.2 Results from Simulated annealing based single objective optimization of models 1-4 . 59
4.3 Results from genetic algorithm based multi-objective optimization - model 1 60
4.4 Function spaces for model 1 61
4.5 Results from genetic algorithm based multi-objective optimization - model 2 62

4.6 Function spaces for model 2 . . . 63
4.7 Results from genetic algorithm based multi-objective optimization - model 3 . . . 64
4.8 Function spaces for model 3 . . . 65
4.9 Results from genetic algorithm based multi-objective optimization - model 4 . . . 66
4.10 Function spaces for model 4 . . . 67
4.11 Time for gain tuning optimization of models 1-4 using GA and MOGA . . . 68
4.12 Speedup for the four different models using GA and MOGA . . . 69
4.13 Efficiency for the four different models using GA and MOGA . . . 70
4.14 Response of the PV system to different PI gains obtained from single-objective optimization . . . 71
4.15 Response of the PV system to different PI gains obtained from multi-objective optimization . . . 72
4.16 Function spaces for power system based model . . . 73
4.17 Time, speed up and efficiency of power system model gain tuning . . . 74
4.18 Results comparison between single objective optimization and multi-objective optimization using real time digital simulator . . . 75
4.19 Function spaces for real time digital simulator based plant model . . . 76

5.1 Real time digital simulator based power system model for gain tuning . . . 79
5.2 Controller modelled in the FPGA based device . . . 81
5.3 Hardware-in-loop setup for FPGA based proportional integral controller gain tuning 82
5.4 Data flow and flow chart for HIL gain tuning . . . 83
5.5 Function spaces with the different multiple objective spaces . . . 84

6.1 Step response of the RTDS model with optimal gain values obtained from gain tuning for floating point based controller . . . 87
6.2 Robustness and sensitivity for optimal gain values with floating point controller . . . 88
6.3 Optimal gain values for floating point data type PI control . . . 88
6.4 Pareto front for gain tuning of floating point data type based current control . . . 89
6.5 Step response of the RTDS model with optimal gain values obtained from gain tuning for fixed point based controller . . . 90
6.6 Robustness and sensitivity for optimal gain values with fixed point controller . . . 91
6.7 Optimal gain values for fixed point data type PI control . . . 91
6.8 Pareto front for gain tuning of fixed point data type based current control . . . 92

6.9 Voltage response of the RTDS model with optimal gain values obtained from gain tuning for floating point based controller 93
6.10 Optimal gain values for floating point data type PI based VF control 93
6.11 Pareto front for gain tuning of floating point data type based voltage control 94
6.12 Voltage response of the RTDS model with optimal gain values obtained from gain tuning for fixed point based controller 95
6.13 Optimal gain values for fixed point data type PI control 95
6.14 Pareto front for gain tuning of fixed point data type based voltage control 96
6.15 RSCAD model for the two inverter case 98
6.16 Optimal gain values for fixed point data type PI control for multiple inverters 98
6.17 Step response of the real time digital simulator model for gains obtained from controller hardware in loop simulation under multiple inverter scenario 99
6.18 Pareto front for gain tuning of floating point data type based current control for multiple inverter case 100
6.19 Objective function results for fixed point data type based PQ controller 101
6.20 Objective function results for floating point data type based PQ controller 101
6.21 Objective function results for fixed point data type based VF controller 102
6.22 Objective function results for floating point data type based VF controller 102

Nomenclature

$G_c(s)$	Controller transfer function
$G_p(s)$	Plant transfer function
K_i or k_i	Integral gain constant
K_p or k_p	Proportional gain constant
DER	Distributed energy resources
DNP3	Distributed network protocol
DoF	Degree of Freedom
EA	Evolutionary algorithm
EMS	Energy management system
FOPDT	First order plus dead time plant
FOPTD	First order plus time delay system
GA	Genetic algorithm
IAE	Integral absolute error
IEC	International electrotechnical commission
IED	Intelligent electronic device
IEEE	Institute of electrical and electronics engineers
IMC	Internal model control
IPDT	Integral Plus Dead Time model
ISE	Integral square error

ITAE	Integral time absolute error
LOPTD	Lower order plus time delay system
LQR	Linear Quadratic Regulator
PD	Proportional derivative controller
PI	Proportional Integral controller
PI-PD	Proportional Integral-Proportional Derivative
PID	Proportional integral derivative
SCADA	Supervisory control and data acquisition
SISO	Single Input Single Output
SOPTD	Second order plus time delay system

Chapter 1

Introduction

Energy in any form is the pulse of an economy. The availability of electrical energy is becoming more important, touching everybody's life in various forms. People are now dependent on accessories ranging from intelligent hand washers to intelligent cars. Most of these smart devices are powered by electricity. Due to this increased need for electricity in the past few years, electricity needs to be delivered to the consumer in a more efficient, cost-effective, less polluting, and reliable way with high quality. It is the responsibility of the electrical utilities to ensure that the aforementioned goals are met.

The electrical energy generation in the US is shown in Figure 1.1. From the figure, it is clear that majority of the power generation comes from non-renewable resources. Also, it indicates that electricity generation and transmission happens in a mass scale. Electricity is generated in one location and moved to the load center using transmission lines. Due to this interconnected nature of the power system, a fault at one location in the power system can cascade or propagate throughout the entire power system [1, 2].

An alternate way of providing electricity is to generate electricity locally and utilize it where the load is located. The generating sources are called Distributed Energy Resources (DER). DERs can be a diesel generator, a wind turbine, a micro-turbine or a photovoltaic (PhV) based source. The grid which utilizes local generation and supplies its demand through DERs are known as microgrids or islanded system. *CIGRE C6.22 working group* and *U.S DOE* microgrid exchange group have defined the term microgrids as the following [3]:

1. CIGRE C6.22 Working group - microgrids are electric distribution system containing loads and DERs (such as distributed generators, storage devices, or controllable loads), that can be operated in a controlled, coordinated way while connected to the main power network or during islanded mode of operation.

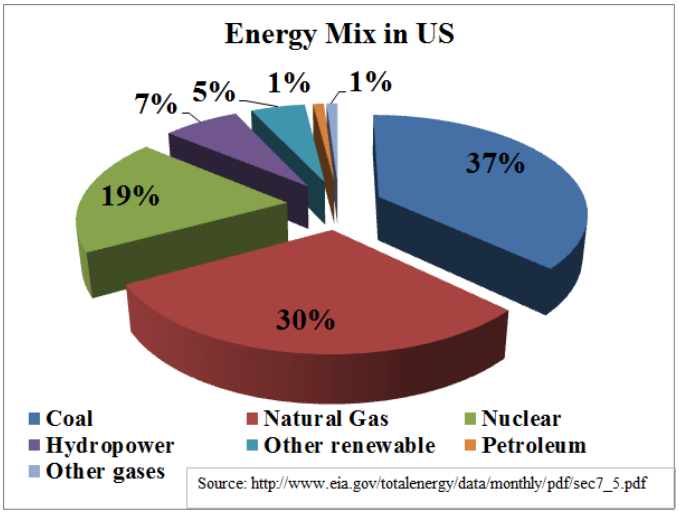


Figure 1.1: Electricity generation by energy source

2. U. S. DOE microgrid Exchange Group, 2010 - A microgrid is a group of interconnected loads and DERs within clearly defined electrical boundaries that acts as a single controllable entity with respect to the grid. A microgrid can connect and disconnect from the grid to enable it to operate in both grid-connected mode or in islanded mode of operation.

1.1 Types of sources in microgrids

The sources of generation used in the microgrids are unique and they possess unique problems unlike the traditional bulk transmission [4]. Some of the sources used in the microgrids that are currently in development are listed below:

- Reciprocating Internal Combustion (IC) engines,
- Gas turbine,
- Small Hydro-electrical systems,
- Photovoltaic systems,
- Wind Energy conversion systems,
- Energy storage systems,
- Microturbine.

1.2 Use of microgrids in the power system

The first generation of microgrids was used in Antarctica to power the INMARSAT earth station at Black Island, Antarctica [5, 6]. This system used wind/closed cycle vapour turbines along with batteries to supply power to the DC and AC loads. Use of such microgrids was also indicated in [7]. This literature presents microgrid as a way to economically electrify rural areas in Indonesia, India, and China. Microgrids were considered a viable and a faster solution at extreme environments since 1970s. But, improvements in the transmission systems and the advantages they possess over local generation moved investments towards large scale generation and transmission. The following section will present some of the present day uses of microgrids in the power system.

1.2.1 Use of microgrids in military bases

Department of Defense (DoD) and Department of Energy (DoE) has special interest towards using microgrids [8] to power U.S military bases. Moreover, there are concerns over dependency on fossil fuels [9, 10, 11]. These agencies are interested in moving towards energy efficient technologies that can use renewable energy to power military bases. Smart Power Infrastructure Demonstration for Energy Reliability and Security (SPIDERS) project is an initiative by DoD and DoE towards realizing smarter, and more resilient microgrid [12] with renewable and Electric Vehicle (EV) integration. The number of EV used in such military bases are on the rise. This is due to improvement in the EV technology and also due to military's interest in using alternative fuel sources [11, 13] based vehicles.

The resilient operation of such systems is a key factor in the success of such microgrids. SPIDERS initiative is aimed towards providing a comprehensive secure energy solution using energy surety microgrids and cyber security [14]. In a military power system, critical loads are highly dependent on electrical energy. The issue of fragile electricity grid poses a significant threat to the military operations. For a sustained mission assurance and emergency support, smart grid technologies along with intelligent demand-side management are needed. Microgrids pose a viable structure in order to meet this need.

1.2.2 Use of microgrids during emergency by utility services

During a natural disaster like a hurricane or a tornado, utility services lose their critical infrastructure like transmission lines and substation equipments [15]. These disasters not only affect everyday life but also affects critical infrastructures like hospitals [16] where losing power can lead to dire circumstances. One of the solutions to avoid losing the transmission infrastructure is to use underground cables but it is known to be very expensive [17]. Another solution is to use DERs

and operate the grid in islanded mode or in microgrid mode that powers only the most critical loads in the system until the transmission infrastructure is brought back [18, 19]. This is a significant cost effective solution compared to building an underground transmission infrastructure. Building DERs for natural calamities which can potentially occur without an imminent threat might seem costly and of rather limited necessity. However, the cost of harming patients due to interruption of critical medical care justifies the financial expenses of building a microgrid for critical load support.

1.3 Challenges in microgrid implementation

Even though microgrid possess a lot of advantages, its implementation faces challenges. For a proper microgrid operation three components are crucial: *a*) protection, *b*) control, and *c*) communication between the different intelligent electronics devices (IED) and the supervisory control and data acquisition (SCADA)/energy management system (EMS). There are challenges in these three components which upon improvements can improve the reliability of microgrids.

1.3.1 Challenges in protection

Microgrids are typical distribution level power system components with some crucial differences which makes protection of microgrids challenging. Protection systems designed to protect a microgrid should consider the following challenges: *a*) reduced fault currents during islanded operation, *b*) bidirectional power flow and looped feeder structure.

Reduced fault currents

A microgrid is designed to operate in both grid-connected mode and in an islanded mode of operation, the protection schemes should protect the microgrid in both modes of operation.

Traditional grid protection depends on high fault currents, which trigger the operation of over current devices. In inverter based DERs, low thermal overload capability presents a big drawback. Due to this low thermal overload capability, inverter fault current is limited to 2-3 times the rated current. This is a big issue mainly during the islanded mode of operation [20].

During on grid mode of operation, the grid act as an infinite source and provide fault current when a fault occurs. The protection devices will pick this high fault current and try to isolate the fault. But, the protection devices see low fault currents during islanded mode of operation. Thus, traditional over-current protection settings used during on grid mode will not trigger a fault isolation during islanded mode. Oversizing the inverter switches will allow larger output currents, but, this

strategy is expensive. Thus, the protective devices used in microgrids should be designed for small fault currents to avoid unnecessary tripping.

Bidirectional power flow protection and looped feeder protection

Usually distribution systems are connected in radial mode. Some have loop closing feeders. These loops are usually kept open by normally open switches. When a fault occurs, the loops might be opened and the normally open switches closes the other parts, thus trying to preserve the radial structure. But, islanded mode of operation enables a continuous power supply for loads during a major grid disturbance, thus mesh configuration is preferred. Protecting this mesh configuration again pose a challenge [21].

Protection schemes proposed for radial microgrids might not be effective for deployment in meshed microgrids. Fault current seen by relays in mesh configured microgrid will not have significant difference due to the short lines in microgrid. A communication channel will be really helpful for protection schemes under such meshed configurations.

Without DER, power flows in one direction even during fault conditions and normal conditions. Protection devices are coordinated based on unidirectional power flows from the feeder toward loads in radial system. Also, the fault currents reduce as the fault location move from the feeder. But during islanded mode, DERs along the microgrid will contribute to the fault current along the feeder. Thus the previous conception which predicts that the fault current will vary along the line will no longer hold true.

1.3.2 Challenges in communication protocols

Communication in microgrid supports the protection, control, and energy management system. Research in protection schemes and control schemes aims towards operation without communication. Nonetheless, communication availability will definitely provide support and increase the reliability of the decision made by protection and control devices. But, for energy management systems communication protocols are inevitable. International Electrotechnical Commission (IEC) 61850 [22] and Distributed Network Protocol (DNP3) [23] are a couple of protocols which are utilized by the industry for typical operation of EMS and protection systems. These protocols support interoperable nature of protection devices and other control devices.

Eventhough protection schemes do not majorly use communication for proper functioning, they do need communication for monitoring these devices and sending crucial information to SCADA or EMS. IEC 61850 [24] has a special section called IEC 61850 7 - 420 [25] for distributed energy resources. This section provides logical nodes for various DER components like *a*) generating

sources, *b*) reciprocating engine, *c*) fuel cell, *d*) photovoltaic system, *e*) Combined heat and power, *f*) fuel system, *g*) battery system, *h*) fuse, *i*) physical measurements, and *j*) metering.

IEDs contain a number of logical nodes, where each logical node implement a particular protection or control function. These logical nodes require data inputs, known as data sets, from other IEDs. Some protocols available in IEC 61850 for communication are Generic Object Oriented Substation Events (GOOSE), sampled values (SV), TimeSync (SNTP), Generic Substation State Events (GSSE), and Manufacturing Message Specification (MMS). Different types of files are used to describe the function of IED. They are: *a*) IED capability description (ICD) file, *b*) System Specification Description (SSD) file, *c*) Substation Configuration Description (SCD) file, *d*) Configured IED Description (CID) file, *e*) Instantiated IED Description (IID) file. They all use XML syntax format known as the system configuration description language (SCL). IEC 61850 have been used in the industry very widely to ensure interoperability among devices. Some of the logical nodes need to be modified in order to use this protocol for microgrids. Even though a dedicated section is available for DERs, still many IEDs are not covered in IEC 61850. This is just one of the many updates that are needed in the communication protocol, if the IEDs in microgrid were to be interoperable.

1.3.3 Challenges in control

As mentioned earlier, microgrid is the part of the power system which possess the capability to serve loads with the use of DERs, storage, and load control after separation from the main grid [26, 27]. In order to properly perform this action, some controls are needed to maintain stability in the microgrid. Typically microgrids have the potential to operate in: *a*) grid connected mode, *b*) islanded mode, and *c*) the capability to change from one mode to the other.

Islanding and resynchronization controls using microgrid switch controller

A microgrid is typically connected to the main grid through a circuit breaker. This circuit breaker should open when there is a fault in the main grid and this should close back when the grid comes back. Also, the control modes of the inverter depend on the operation of this circuit breaker. The controls of the inverter depend on the mode of operation of the microgrid.

For example, during islanded mode of operation the voltage and frequency of the microgrid should be controlled by the local DERs. And during on grid mode, the voltage and frequency will be controlled by the grid. These control changes should happen seamlessly to maintain stability in the system. A good communication between the microgrid switch controller and the DER is crucial for this.

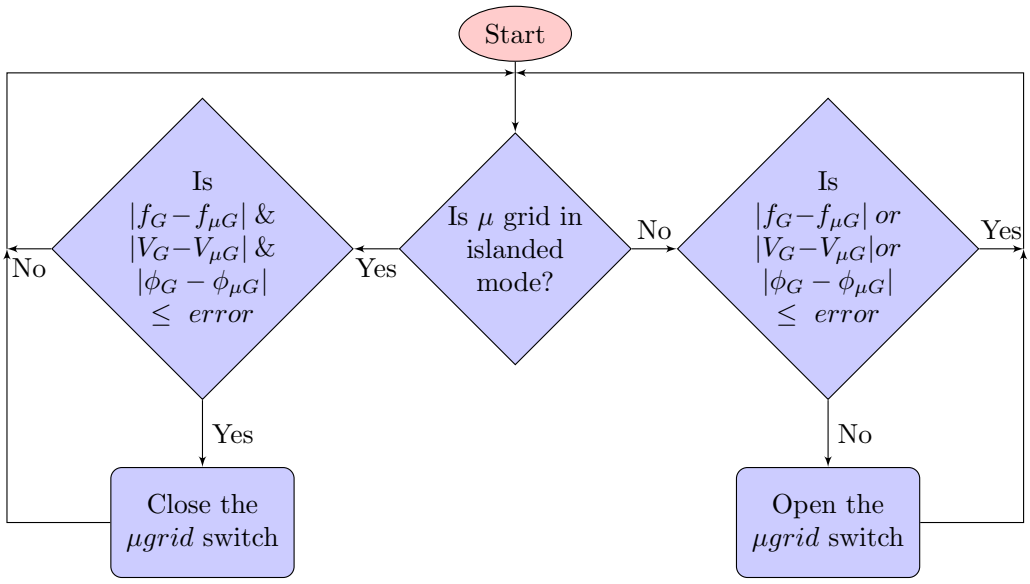


Figure 1.2: Flowchart indicating the microgrid switch controller islanding and grid-connected mode transition

Fig. 1.2 shows the decision making involved in controlling the microgrid switch circuit breaker. More details can be found in Institute of Electrical and Electronics Engineers (IEEE) draft standard P1547 for distributed resources interconnected with electric power systems [28, 26].

Conventional voltage and frequency (VF) control

The power flow from a point A to point B through a line is shown in Fig. 1.3a. The real and reactive power flow is shown in 1.4 and 1.5 [29].

$$P + jQ = \underline{S} = V_1 I^* = V_1 \left(\frac{V_1 - V_2}{\underline{Z}} \right)^* \tag{1.1}$$

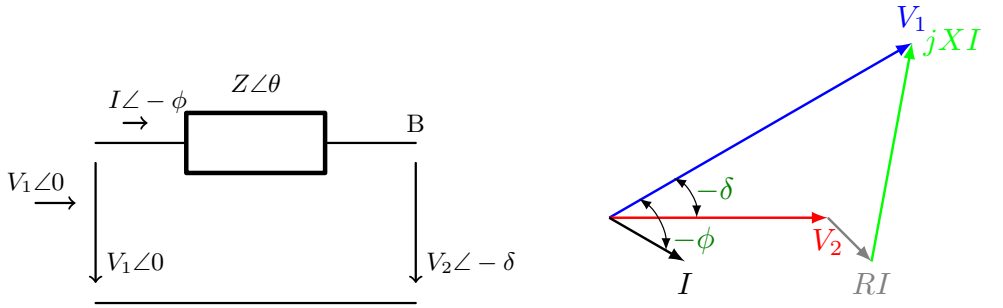
$$= V_1 \left(\frac{V_1 - V_2 e^{j\delta}}{Z e^{-j\theta}} \right) \tag{1.2}$$

$$= \frac{V_1^2}{Z} e^{j\theta} - \frac{V_1 V_2}{Z} e^{j(\theta+\delta)} \tag{1.3}$$

Thus, active and reactive power flowing into the line are described as:

$$P = \frac{V_1^2}{Z} cos\theta - \frac{V_1 V_2}{Z} cos(\theta + \delta) \tag{1.4}$$

$$Q = \frac{V_1^2}{Z} sin\theta - \frac{V_1 V_2}{Z} sin(\theta + \delta) \tag{1.5}$$



(a) Power flow through a line

(b) Phasor diagram of transmission system

Figure 1.3: Transmission system and its respective phasor diagram

with $Ze^{j\theta} = R + jX$, (1.4) and (1.5) can be rewritten as:

$$P = \frac{V_1}{R^2 + X^2}[R(V_1 - V_2 cos\delta) + XV_2 sin(\delta)] \tag{1.6}$$

$$Q = \frac{V_1}{R^2 + X^2}[-RV_2 sin\delta + X(V_1 - V_2 cos(\delta)] \tag{1.7}$$

For overhead lines $X >> R$, meaning that R can be neglected. Also, if the power angle is small, then $sin\delta = \delta$ and $cos\delta = 1$. Equations (1.6) and (1.7) become:

$$\delta = \frac{XP}{V_1 V_2} \tag{1.8}$$

$$V_1 - V_2 = \frac{XQ}{V_1} \tag{1.9}$$

Equations (1.8), and (1.9) show that power angle is directly proportional to the real power (P), and the voltage difference is directly proportional to the reactive power (Q). Thus, the angle can be controlled by regulating (P) and the voltage can be controlled by regulating (Q). Equations (1.8), and (1.9) can be used to form the basis of frequency and the voltage droop equations and they are the following:

$$f - f_0 = -k_p(P - P_0) \tag{1.10}$$

$$V_1 - V_2 = -k_q(Q - Q_0) \tag{1.11}$$

where f_0, and V_0 are nominal frequency and nominal voltage, and P_0, and Q_0 are the set points for active and reactive power of the inverter.

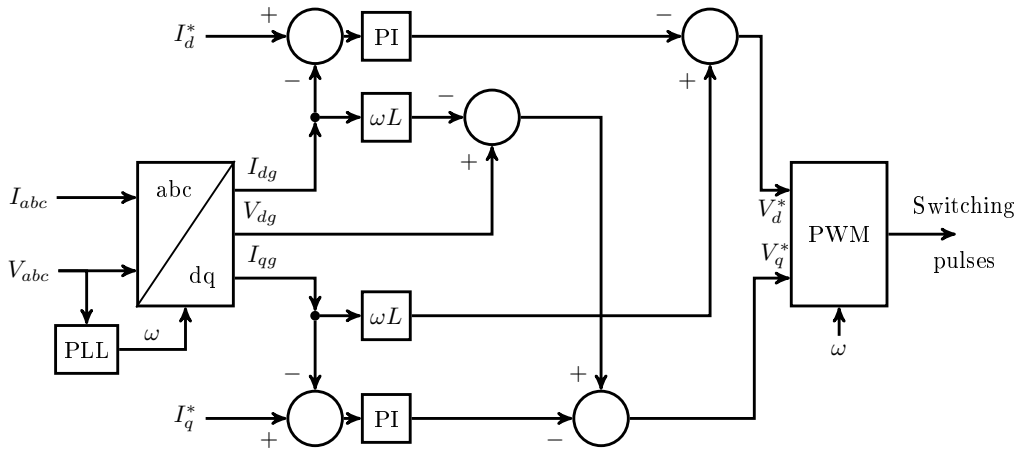


Figure 1.4: Inverter control based on current control technique

Voltage-frequency (VF) and real-reactive power (PQ) control in microgrid

In a microgrid setting, majority of the DERs import energy into the grid through power electronic interface. These power electronic interfaces such as inverters can operate in grid-connected and islanded mode of operation. In islanded mode of operation, inverter must maintain a constant output voltage and frequency. In grid connected mode of operation, voltage and frequency are determined by the grid. Power factor is maintained at unity during this mode of operation. These power electronic interfaces can provide harmonic mitigation, voltage control at point of common coupling (PCC), and other advanced controls. But, for such advanced features, communication protocols need to be implemented between the power electronics interfaces and the utility. Proportional integral controllers must be properly designed in order to optimize the control capability. Another additional challenge in decoupled control is the synchronization with grid frequency.

Mostly, inverters are designed to supply constant current output. A constant current control is implemented with reference frame transformation from three phase (abc) to stationary (dq) frame. Equation 1.12 shows the dynamics used for the feedback control. This loop limits the output voltage on the reference frame. A unit vector is synthesized using PLL that allows synchronization with the grid.

$$L\frac{d}{dt}\begin{bmatrix} i_d \\ i_q \end{bmatrix} = \begin{bmatrix} R & -\omega L \\ \omega L & R \end{bmatrix} \times \left(\begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} \Delta e_d \\ \Delta e_q \end{bmatrix} \right) \tag{1.12}$$

Where i_d, i_q are d and q axis current components, Δe_d and Δe_q are the instantaneous voltage difference between the PCC and the inverter output voltage of d and q axis components, R, L, ω are grid resistance, inductance, and fundamental angular frequency.

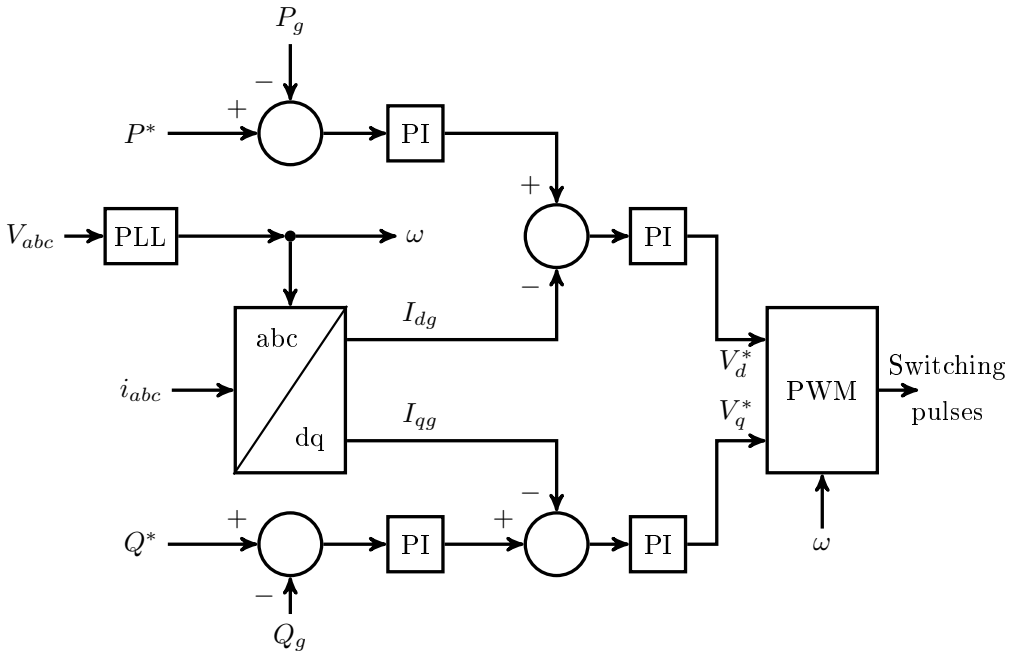


Figure 1.5: Inverter control based on active reactive power control technique

Control of grid connected inverters using instantaneous reactive power in three-phase circuits is shown in [30]. When using such control methodology, the calculated power does not vary by the transformation used. The q axis reference value for current is compared to the instantaneous grid q axis current and the d axis reference current value is compared to the instantaneous grid d axis current. Using this an outer PQ control over an inner current loop control can be achieved. The equations used is shown below.

$$p = \frac{3}{2}\left[v_{ds}i_{ds} + v_{qs}i_{qs}\right] \tag{1.13}$$

$$q = \frac{3}{2}\left[v_{qs}i_{ds} - v_{ds}i_{qs}\right] \tag{1.14}$$

The grid connected inverters can be used to control the voltage at the PCC. The voltage can be used to adjust the amount of reactive power required in the system. In this method, the instantaneous peak voltage at the utility grid must be followed at the PCC. These methods are explained in Fig. 1.4, Fig. 1.5, and Fig. 1.6. One feature common in all the three methods is the proportional controller. The inherent nature of this PI controller is that the performance depends on the gain values selected for the PI controller.

Proportional integral (PI) controller is a commonly used feedback controller with two gain constants: proportional gain constant (k_p) and integral gain constant (k_i). Performance of the

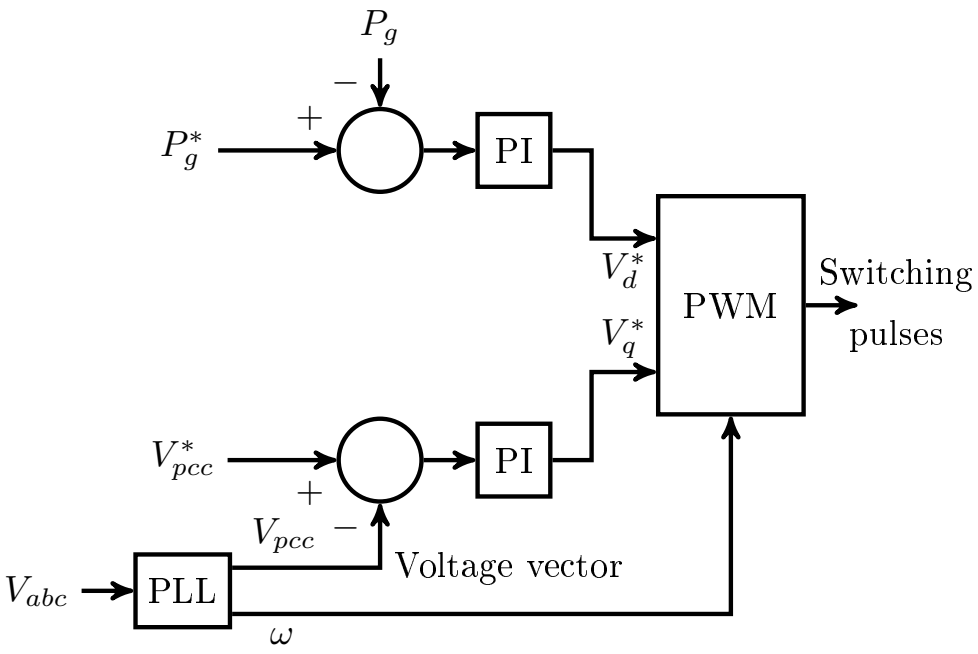


Figure 1.6: Inverter control based on active power and voltage feedback technique

controller depends on the values selected for these gains. In Ref. [31], the authors discuss how the gain parameters determine the performance of the controller. The issue of selecting proper gain constants is straightforward for simple systems [32]. However, it is much more complicated for complex plant models. One such system with complex plant model is renewable energy based power generating systems. Renewable energy based power sources inject real and reactive power to control the voltage and frequency of the grid. PI control is used to control the real and reactive power injected by such systems [33, 34, 35, 36]. Modelling of these systems can be complex, if the end goal is to utilize the model to tune the PI controller [37, 38]. Other factors which add complexity are the existence of multiple topologies to integrate photovoltaics (PVs) [39, 40, 41, 42] and multiple control methodologies that could be used to control such topologies [43]. Each topology has unique plant dynamics and the PI control needs to be tuned to address this unique dynamic nature individually, which could be a challenging task.

Obtaining a good plant model for all these topologies may require a large amount of time either for physical modelling and/or identification. An ideal strategy for PI gain tuning should not require a detailed and accurate mathematical model of the plant, since developing detailed and accurate mathematical model of the plant is usually time consuming and complex. This is of particular importance to practicing utility engineers in the field [44]. The tuning strategy should also be

robust irrespective of the topology of power electronics and the control strategy used. These two factors pose the need for a universal tuning strategy.

One such tuning method is proposed here. The proposed strategy uses the discrete time switching model of the inverters as the plant model. This avoids the need for complex mathematical models of the system and the possible error due to simplification. The strategy also considers the inverter plant model as a black box such that the tuning method can be used to tune realistic topologies and models.

However, this strategy itself presents a potential challenge. It uses switching model in-stead of simplified models. In general, switching models are more computationally expensive than simplified mathematical models. Further, the switching model is combined with an optimization model such as genetic algorithm for tuning, so it takes even more time to perform simulation as opposed to a simplified model.

Although one may argue that running time is not critical to a tuning method which is essentially an off-line application, the disadvantage of using the switching model is still undesired when tuning may take hours. Therefore, in order to overcome these challenges, parallel computing is used to perform individual time domain simulation to significantly reduce the overall running time for the proposed tuning approach. Note, the paralleling also well justifies the choice of the genetic algorithm (GA) for tuning optimization because GA naturally fits the proposed paralleling scheme.

Finally, this methodology is ported to tune a hardware based PI controller. The PI controller is modelled in an FPGA based hardware which has the capability to control actual inverter. In order to avoid damaging the inverters by bad gain values, the inverters are modelled in real time digital simulator and a controller hardware-in-loop simulation is performed to tune the gain values of the PI controller programmed in the FPGA device.

1.4 Dissertation outline

Chapter 2 presents literature review on the PID gain tuning topic and presents conclusions based on the observations made in the literature. Chapter 3 presents an introduction to the algorithms used for gain tuning and an introduction to parallel computing. Chapter 4 presents the models used for tuning in the two different software programs and the results of the gain tuning. Chapter 5 presents an introduction to controller hardware-in-loop simulation and the experimental setup with the plant model that will be used for the gain tuning. Chapter 6 presents the results from the controller hardware-in-loop gain tuning. Finally in chapter 7, the summary of this work is presented. The final chapter also talks about the potential future direction of this work.

Chapter 2

Literature review

2.1 Introduction

This chapter presents the work available in the literature that are related to proportional integral (PI) tuning methodologies. First, a short introduction on some basic concepts are presented. This will help in understanding some of the keywords and concepts presented in the literature and also in the rest of the thesis. Then a discussion on the tuning methodologies available in the literature are presented. Finally, a justification for the proposed methodology to tune PI gain parameter is presented.

2.1.1 Proportional-Integral controller

PI controls are common and well-known. PI control has two major types: series PI control and parallel PI control [45]. For our study, parallel PI control is used. The basic PI block diagram is shown in Fig. 2.1 and the equation is shown in 2.1.

$$u = k_p e + k_i \int_0^t e(\tau) d\tau = k_p \left(e + \frac{1}{T_i} \int_0^t e(\tau) d\tau \right) \tag{2.1}$$

where k_p is the proportional gain constant, k_i is the integral gain constant, and $e(t)$ is the error from the actual set point. The time constant T_i, called integral time constant can also be used instead of the integral gain. These gains have a significant influence on the performance of the controller.

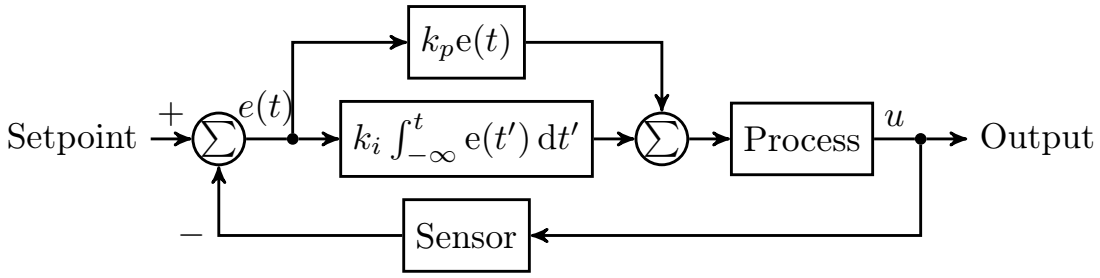


Figure 2.1: Typical PI controller structure for process control

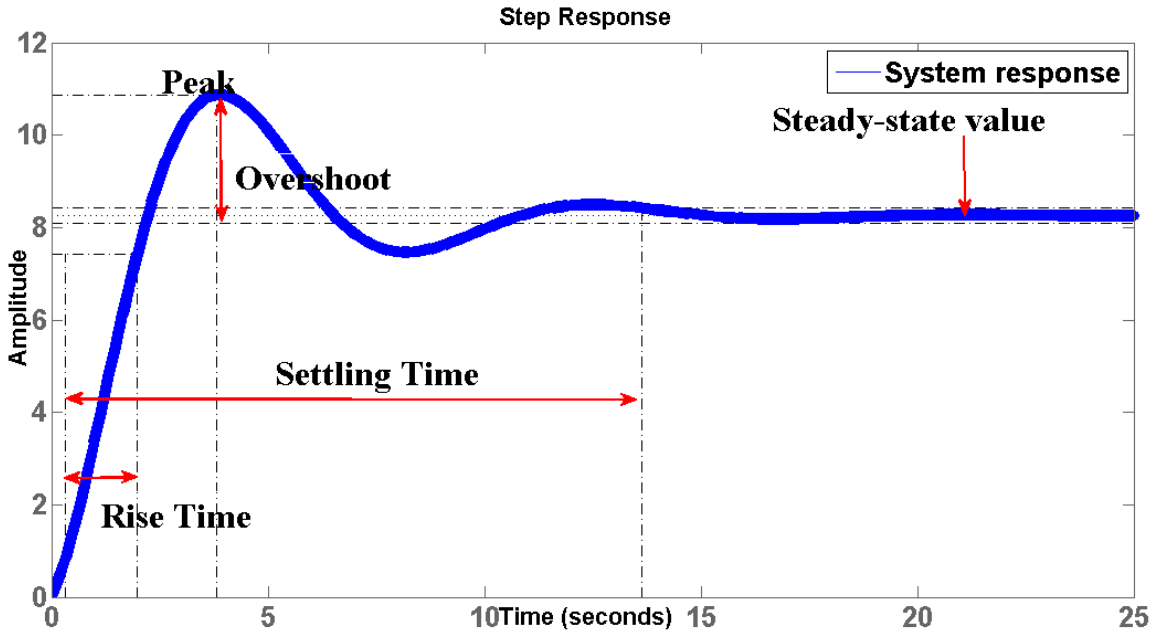


Figure 2.2: Step response of a system

2.1.2 Response characteristics of a system

The typical response characteristics of a system with PI control is shown in Figure 2.2. The waveform indicates certain critical parameters that can be used to quantify the performance of control. They are [46]:

- Rise time - the time required for the response to rise from 10% to 90% of its final value,
- Settling time - the time it takes to get within 10% of the final value, or to 90% of the final value,
- SettlingMin - minimum value of output amplitude once the response has risen,
- SettlingMax - maximum value of output amplitude once the response has risen,

- Overshoot (in %) - the maximum peak value of the response curve measured from the desired response of the system,
- Undershoot (in %) - percentage undershoot from the expected final value,
- Peak value - peak absolute value of system response,
- Peak time - time at which this peak is reached

2.1.3 Performance criteria

The following are some of the performance criteria used in the literature:

- Integral of time multiplied by absolute error (ITAE) - This integrates the absolute error multiplied by time over time. This criteria weighs error that exist later in the process harder than those at the start of the response.

$$ITAE \ = \ \int_0^\infty t|e(t)|dt \tag{2.2}$$

- Integral of squared error (ISE) - This integrates the square of the error over time. In this criteria, larger error will be penalized more than the smaller since the error squared will be larger for larger errors.

$$ISE \ = \ \int_0^\infty e^2(t)dt \tag{2.3}$$

- Integral of absolute error (IAE) - This criterion integrates absolute of error over time. There is no weight added to the error.

$$IAE \ = \ \int_0^\infty |e(t)|dt \tag{2.4}$$

- Integral of time multiplied by squared of error (ITSE) - This criterion integrates the error squared multiplied by time. The addition of time minimizes the effect of large initial error.

$$ITSE \ = \ \int_0^\infty t(e(t))^2dt \tag{2.5}$$

- Integral squared time squared error (IST^2E)

$$J_n(\theta) \ = \ \int_0^\infty t_n e(\theta,t)^2 \ dt \tag{2.6}$$

$$n \ = \ 2$$

where θ determines the variable parameters which can be chosen to minimize $J_n(\theta)$ and $e(t)$ is the error signal at time t.

2.1.4 Stability margins (Gain margin - A_m and Phase margin - ϕ_m)

In practice, having a stable system is not enough. Stability margins are needed in order to understand the stability of the system and its robustness to perturbations. In order to obtain the gain margin and phase margin, the plant transfer function $G_p(s)$ and controller transfer function $G_c(s)$ are used. These concepts could be understood using Nyquist plot and Bode plot [47]. A controller gain increase will expand the Nyquist plot radially. A controller phase increase will twist the Nyquist plot. Thus using Nyquist plot, gain or phase can be picked for a stable system.

Gain margin is defined as the smallest amount that the open loop gain can be increased without making the closed loop system unstable. The gain margin can be calculated based on the smallest frequency where the phase of the loop transfer function is -180°. In other words, gain margin is the amount of gain increase or decrease required to make the loop gain unity at the frequency where the phase angle is -180°. Gain margin is the gain perturbation that makes the system marginally stable. In Bode plot, the gain margin is the inverse of the gain at the crossover frequency. Crossover frequency is the frequency in Bode plot where the phase is -180°.

Phase margin is the amount of phase lag required to reach the stability limit. Phase margin is related to the damping of the system. In order to obtain the phase margin using bode plot, the gain crossover frequency ω_g should be obtained. The gain crossover frequency is the frequency where the gain of the loop transfer function is 1. Phase margin is the phase of the loop transfer function at that frequency plus 180°. From the above definitions, Gain margin and Phase margin can be written as:

$$A_m = \frac{1}{|G_c(j\omega_p)G_p(j\omega_p)|}, \tag{2.7}$$

$$\phi_m = arg[G_c(j\omega_g)G_p(j\omega_g)] + \pi, \tag{2.8}$$

Where ω_g is given by

$$|G_c(j\omega_g)G_p(j\omega_g)| = 1 \tag{2.9}$$

and ω_p,

$$arg[G_c(j\omega_p)G_p(j\omega_p)] = -\pi \tag{2.10}$$

There are some draw backs to the gain and phase margins using Nyquist and Bode plots. Both of these margins are needed in order to guarantee that the Nyquist curve is not close to the critical

point. Bode plot interpretation of the gain and phase margins can be incorrect if there are multiple frequencies at which the gain is equal to 1 or the phase is equal to -180°.

2.2 Literature review

Perhaps one of the earlier papers that presents insight on gain tuning is [48]. The method proposed in this paper is well known in the literature as "Ziegler-Nichols tuning method". This paper presents a couple of terminology: *a*) Ultimate sensitivity (S_u), and *b*) period of oscillation (P_u) . Ultimate sensitivity is the point beyond which any oscillation will increase to a maximum amplitude, and below which oscillation of any size will diminish to straight-line control. The period of the oscillation produced at the ultimate sensitivity (S_u) is P_u. This is a good index of required reset-rate adjustment. From these values, the optimum controller setting was approximated as

$$Sensitivity = 0.45 S_u \tag{2.11}$$

$$Reset\ rate = \frac{1.2}{P_u} \tag{2.12}$$

The following are the steps involved in obtaining the ultimate sensitivity and period of oscillation:

- The proportional gain should be increased from a low value until the system becomes marginally stable. The corresponding gain k_u is the ultimate gain or the ultimate sensitivity (S_u), and,
- The period of oscillation for this oscillation is the "ultimate period" P_u and is determined for the same ultimate gain value.

Equations 2.11 and 2.12 are modified in [49] to obtain the k_p and k_i

$$k_p = 0.45 \times k_u \tag{2.13}$$

$$k_i = \frac{k_p \times P_u}{1.2} \tag{2.14}$$

In [50], it was indicated that opening the control loop during tuning is not desirable and such methodology will not work for process which are not stable in the open loop configuration. This is a main disadvantage in using Ziegler-Nichols based tuning. Also, in some process it is not desirable to operate the process near instability. The gain values obtained as a result of Ziegler-Nichols based tuning might create unwanted oscillation as mentioned in [51].

Another most cited tuning methodology is proposed in [52]. The plant model used in this paper is a single-capacity process with dead-period lag. This model is also called as first order plus dead

time plant model (FOPDT). The methodology proposed in [52] uses the characteristic equations corresponding to the application of P, PI, PD, and PID responses. These responses are used to graph the controller parameter necessary to obtain the desired degree of stability. The degree of stability is taken to be associated with the amplitude ratio of the lowest frequency harmony mode. This methodology also needs process characterization. Obtaining good gain parameters beyond single-capacity process with dead-period lag using this methodology is a challenging endeavour and is not discussed well in [52].

The above two mentioned literature deal with off-line tuning methodologies. On-line tuning of PID controller was seen initially in [50]. This paper uses cross-correlation technique to obtain the gain values. This methodology introduces minor perturbation on the normal operation of the process along with prior information. There is a need to avoid such perturbations due to safety reasons while tuning on-line. Also, the plots indicate a certain level of overshoot in the response plot.

In [53], the authors suggest that Ziegler-Nichols and Cohen-Coon methods to be a bit aggressive for most chemical industry applications. The authors also suggest that closed loop response is sluggish when the process dead time is large relative to dominant open loop time constant. Internal model control-proportional integral derivative (IMC-PID) tuning method is attractive to industrial users because it has only one tuning parameter: the closed loop speed response. Also, the closed loop step load response exhibits no oscillation or overshoot. One disadvantage of this proposed methodology is that it assumes that the majority of processes are well approximated by dead time first order or dead time integrator model. As long as the process can be modelled as one of the two above mentioned process, the gain parameters can be tuned. The tuning rules proposed in this paper is specifically for models with dead time.

In [54], simple IMC based tuning rules are proposed. In [54], it was indicated that even the most detailed model is still only an approximation of reality, and an important property of the designed feedback controller is that it be insensitive to modelling error.

In [55], the controller parameter was tuned to minimize the difference between the actual transfer function and a target loop transfer function in the sense of H_∞ norm. The authors also assume that the formulation of loop shaping objective as a convex optimization problem that can be solved numerically in a reliable way. Reduced order approximations of the plant is used to enable analytic derivation of the tuned parameters. Moreover the approach assumes that the plant model is a linear, time invariant SISO plants with available known nominal model G(s) of the plant. This might not be the case for all processes and thus limits the approach. The paper concludes by indicating that the simplicity of the approach relies heavily on the convexity in the parameter structure of PID and may not be presented in tuning problems with different parametrizations or objectives.

The authors in [51] extended the frequency domain PID controller proposed in an earlier paper [56] for an integrating process. Taylor series expansion was applied to the stable part of the process transfer function. The integrating process is divided into two types based on the sign of the second coefficient of this expansion. The closed loop performance is specified in terms of desired control signal trajectory scaled with respect to the magnitude of this coefficient. The paper presents PID tuning rules for integrating plus delay processes along with their gain and phase margins. The authors assume that the integrating process can be described by the transfer function

$$G_1(s) = \frac{1}{s}G(s)$$

where $G(s)$ is stable with all poles strictly on the left half complex plane. Gain and phase margin of the closed loop system was used to show the stability of the tuning results.

A frequency response based tuning method was shown in [57]. Two graphs were plotted using the frequency response of the process and the intersection between these two graphs is used to find the gain values. The authors suggest that it is hard to achieve both gain and phase margins using the available tuning models and also hard to tune by assuming that the industrial process are a FOPDT model. The following are the equations used to obtain the intersecting points using a computer program:

$$G(j\omega_p)\left[k_p - j\frac{k_i}{\omega_p}\right] = -\frac{1}{A_m}, \tag{2.15}$$

$$G(j\omega_g)\left[k_p - j\frac{k_i}{\omega_g}\right] = -exp(j\phi_m) \tag{2.16}$$

where ω_p, ω_g are the phase and gain cross over frequencies of loop. Tune k_p and k_i such that the needed gain and phase margins are achieved. The number of unknowns will then be equal to the number of real equations. Regardless of the relation between the number of unknowns and the number of equations, these equations are hard to solve due to the non linearity of the equations. This is further complicated by the coupling among them. Modifying equations 2.15 and 2.16 will result in equations from 2.17 to 2.20.

$$k_P = Re\left[\frac{-1}{A_m G(j\omega_p)}\right], \tag{2.17}$$

$$k_I = -\omega_p Im\left[\frac{-1}{A_m G(j\omega_p)}\right]. \tag{2.18}$$

$$f_p(\omega) = Re\left[\frac{-1}{A_m G(j\omega)}\right] - j\omega Im\left[\frac{-1}{A_m G(j\omega)}\right],$$

$$-\frac{\pi}{2} < \angle g(j\omega) < -\pi, \tag{2.19}$$

$$f_g(\omega) = Re\left[\frac{-exp(j\phi_m)}{G(j\omega)}\right] - j\omega Im\left[\frac{-exp(j\phi_m)}{G(j\omega)}\right],$$

$$-\frac{\pi}{2} + \phi_m < \angle G(j\omega) < -\pi + \phi_m. \tag{2.20}$$

Graphs are plotted for $f_p(\omega)$ and $f_g(\omega)$ and the intersection between the two provide the controller gain values. The paper shows result of tuning for various plant models.

Different types of performance indices namely the integral square error (ISE), integral absolute error (IAE), and the integral time absolute error (ITAE) are presented in [58]. They are shown in equations 2.2 to 2.4. In this paper, the minimization of the ISE objective function is added to the four equations from the basic definition of phase and gain margins. These phase and gain margin equations are shown in equations 2.21 to 2.24. This paper also uses FOPTD models as the process model. These indices were explained in detail earlier in this chapter.

$$\phi_m = arg[G_c(j\omega_g)G_p(j\omega_g)] + \pi, \tag{2.21}$$

$$A_m = \frac{1}{|G_c(j\omega_p)G_p(j\omega_p)|}, \tag{2.22}$$

Where ω_g is given by

$$|G_c(j\omega_p)G_p(j\omega_p)| = 1 \tag{2.23}$$

and ω_p,

$$arg[G_c(j\omega_p)G_p(j\omega_p)] = -\pi \tag{2.24}$$

Tuning formulas were derived in [59] to tune the PID controller for unstable processes to meet certain gain and phase margin specification. The authors use FOPDT models to show the effectiveness of the tuning methodology. This is shown in 2.25.

$$G_p(s) = \frac{k_p}{s\tau - 1}e^{-sL} \tag{2.25}$$

The loop transfer function becomes

$$G_c(s)G_p(s) = \frac{k_c k_p(1 + sT_i)}{sT_i(s\tau - 1)}e^{-sL} \tag{2.26}$$

As mentioned earlier in the chapter, the phase crossover frequency is the frequency at which the Nyquist curve has a phase of $-\pi$. The gain crossover frequency ω_g is the frequency at which the Nyquist curve has an amplitude of 1. Tuning rules were presented by solving the above equation with the use of plant model information.

In [60], the feedback form of an Internal model controller (IMC) with a Maclaurin series in the Laplace variable is used to tune the controller. The paper presents reduced forms of single degree of freedom controllers and two degree of freedom (DoF) controllers. The PID controller is obtained by taking the first three terms of the Maclaurin series expansion of the single-loop form of the IMC controller.

Use of relay auto tuning and a comparison between performance of PID and PI-PD controller are shown in [61]. The relay auto tuning is different than the adaptive controller because the tuning only takes place periodically rather continuously. In relay auto tuning, the controller switches in the tuning mode to operate as an on-off relay and obtain data from the resulting limit cycle. The measured data can be used in two ways, either to estimate the critical point of the process or the parameters of an assumed plant transfer function. The controller used in this paper is a PI-PD controller. The transfer function of such controller is shown in 2.27.

$$U(s) = k_p \left(1 + \frac{1}{sT_i}\right) E(s) + \frac{k_p s T_d}{1 + \alpha s T_d} C(s) \tag{2.27}$$

This implementation avoids the derivative kick. The authors propose that from the k_c and ω_c information of the plant model, the controller can be tuned. A couple of examples was also proposed with this control structure. The results show a good performance over a normal PID controller.

In order to make accurate model of the process, information from experiments are needed. These results are needed to accurately model the process. The limitation on the amount of information that are needed to be extracted from these experiments is not possible to obtain. In short, more information from experiments will help in modelling the process model [61]. Quality information about the plant is preferred here over quantity.

A multi-loop PID controller tuning method with a control performance specification of 5% was proposed in [62]. Further detuning may also be necessary depending on the multi-loop interaction. Since, this is a multi-loop controller, a PID controller with no proportional kick was used.

The proportional part of the PID controller will act immediately on the error signal. When there is a set point change, the error becomes very large and the proportional controller act on this large error, thus creating a proportional kick. This plays a crucial part in multi-loop system because the large controller action on one loop will be a large load disturbance to the other loops. In order to correct this, the control action on the other loop will be large causing more disturbance on the other loop which initially had the set-point change. This will oscillate between the controllers.

The proposed solution in this paper is to not let the control action have the kick at the time of the set-point changes. Instead of having the proportional action on the error, the proportional action was used on the negative of the controlled variable alone. The Laplace transform of this is

shown below:

$$u(s) \;=\; k_c\{-y(s) + \left(\frac{1}{\tau_i s}\right) E(s) + \tau_d s[-y(s)]\} \tag{2.28}$$

Where $y(s)$ is the output of the plant model, τ_i is the integral time constant, τ_d is the differential time constant, $E(s)$ is the error signal. Complete step by step method used in this tuning method is presented in [62]. The process assumes that the plant model can be assumed as integral plus dead time (IPDT) or FOPDT. The plant examples used are *a*) Wood and Berry column, *b*) a reactor control problem,

Unfalsified control based PI tuning for industrial weight belt feeder is proposed and experimentally demonstrated in [49]. The control engineer will typically set multiple objectives that should be met by the controller. Unfalsified control utilizes the open and closed loop test data to identify a subset of controllers that meets these objectives. Nine hundred PI control laws are included in the initial set of candidate control laws. The following are the steps proposed by the author for the unfalsified control tuning method [49].

1. Initialize $k_u \;=\; k$ such that all controllers are initially unfalsified,
2. Take a set of open or closed loop data (u, y),
3. For each controller $k_i \in k_u$, compute e_{ki} andτ_{ki},
4. Falsify each controller for which
$$\frac{||u||_\infty}{||r_{ki}||_\infty} > \alpha, \tag{2.29}$$
5. Falsify 20 % of the unfalsified controller with the largest ratios. Let k_u denote the remaining 80% of the controller.
6. If the number of controllers in k_u is too large, go to step 2.

The results show that unfalsified controller provide good response for controlling the weigh belt feeder. Controller unfalsification concept here is not implemented in real-time.

Detailed procedures for using unfalsified control theory for real-time PID controller parameter tuning and adaptation is shown in [63]. With the use of unfalsified control theory, a controller can be modelled which is consistent with the performance objective and measured past data without plant models or assumption about the plant models. This method eliminates hypotheses that are not consistent with evolving experimental data. The hypotheses assumes that the members of a class of candidate controllers can meet prescribed closed-loop performance goals. Unfalsified control can adjust controller parameters based on measured data without any assumptions about the plant. For adaptively changing the gains, the algorithm uses only measured past data. This method rather

than obtaining the values of gain constants of PID controller, switches between already known, existing list of gain values.

A tuning method for Smith predictor in the presence of model uncertainty is presented in [64]. First order plus time delay system (FOPTD) and second order plus time delay systems (SOPTD) are used for tuning. The tuning method proposed here also copes up with uncertainties. The closed loop performance can be very poor, if there is inevitable mismatches between the model and the actual process. Closed loop transfer function for Smith predictor is shown below:

$$\frac{y(s)}{y_{sp}(s)} = \frac{G_c(s)G_p(s)}{1 + G_c(s)(G_m^*(s) + G_p(s) - G_m(s))} \tag{2.30}$$

A concept known as equivalent gain plus time delay (EGPTD) is utilized here for tuning. This approximates the complicate form $G_m^*(s) + G_p(s) - G_m(s)$ in the characteristic equation to the simple form consisting of delay free part and an equivalent gain plus time delay as follows:

$$G_m^*(s) + G_p(s) - G_m(s) \approx G_m^*(s)k_{eq}e^{-\theta_{eq}s} = G_w(s) \tag{2.31}$$

Where k_{eq} and θ_{eq} denote equivalent gain and the equivalent time delay. Three reasons are presented for the justification of EGPTD approximation: *a*) over low and middle frequencies, this provides good coincidence with corresponding transfer function, *b*) a less conservative approximation is obtained in high frequency range, *c*) the approximated model is same as that of process model which enables use of other existing tuning rules.

A frequency domain based optimization method for PID controller design scheme is proposed in [65]. The authors also propose here that Ziegler-Nichols rule often lead to oscillatory response to set point changes. In the proposed approach, a set of frequency domain performance functions are considered for gain tuning. They are gain-margin, phase-margin, crossover frequency and steady state error. This paper also propose here that a large number of industrial processes can be characterized by a FOPDT plant model. In the initial step an excitation signal shown in 2.32 is utilized to excite the plant.

$$x(t) = \sum_{k=1}^{n} A_k \sin(2\pi f_i t + \phi_k) \tag{2.32}$$

Where A_k, ϕ_k, and f_i are the amplitude, phase and frequency of the signal components. In addition to phase margin, and gain margin, the system is also expected to meet a speed-of-response specification like bandwidth. The crossover frequency, which is the frequency at which the gain is

unity, would be a good measurement in the frequency domain for systems speed of response.

$$\phi_1(k_c, T_i, T_d) = \left\{ \frac{|k(j\omega)G(j\omega)|}{G_M}, \angle k(j\omega)G(j\omega) = -180^\circ \right\}, \tag{2.33}$$

$$\phi_2(k_c, T_i, T_d) = \left\{ 2 - \frac{180 + \angle k(j\omega)G(j\omega)}{P_M}, \ |k(j\omega)G(j\omega)| = 1 \right\}, \tag{2.34}$$

$$\phi_3(k_c, T_i, T_d) = \left\{ \frac{\omega}{2\pi f_d}, \ |k(j\omega)G(j\omega)| = 1 \right\}, \tag{2.35}$$

$$\phi_4(k_c, T_i, T_d) = \left\{ \frac{1}{e_{ss}|k(j\omega)G(j\omega)|}, \ \omega = 1 \right\}, \tag{2.36}$$

Where $\omega_i(k_c, T_i, T_d)$, for $i = 1, 2, 3, 4$ are the normalised gain-margin, phase-margin, crossover frequency and the steady-state error functions with the desired values G_M, P_M, f_d, *and* e_{ss} respectively. The proposed method uses min-max optimization to solve this optimization problem.

The authors assume that rotary hydraulic test rig is representative of many industrial system that utilize fluid power. System identification is used to model the hydraulic test rig and frequency response of the hydraulic motor was measured only up to 120 Hz. The tuning method is first used in identified model and then the actual system was used.

In addition to critical point information and the gain margin specification, an extra equation is introduced by relating the closed-loop bandwidth to the process bandwidth in [66]. This increases the number of equations and thus equates the number of unknowns to number of equations. The critical point information and the gain margin specifications are used to calculate the proportional gain of the controller. Using this value, and the desired phase margin, another point on the Nyquist plot is searched and located. From the imaginary parts of the two points, the integral and derivative gains are calculated. One difference in this proposed method is to determine the ratio of closed loop to open loop bandwidth. This is the added equation which equates the number of unknowns with the number of equations. A second order plus dead time model is used in [67] to approximate plant models.

The plant model used in [68] for PID tuning is a temperature control of a three zone industrial diffusion furnace used in semiconductor manufacturing. The PID tuning problem is formulated in the frequency domain as convex optimization. This is used along with the system identification. The application presented is the temperature control of diffusion/chemical vapor deposition (CVD) furnaces used in semiconductor manufacturing. The authors propose that the advantage of linear parametrization is that any functional of the form $||W(GC - L)||^{L\infty}$, with W weighting function, is convex in the design parameters. Frequency loop shaping tuning of the PID parameters is the

following optimization problem.

$$\max_{k \in C} ||F(j\omega) - Z(j\omega)||_{\infty}^{2} \tag{2.37}$$

Where k is the vector of optimization parameters, $k = [k_1, k_2, k_3]^T$; C is the convex set of constraints for k; F, Z are frequency functions that depend on the frequency responses of the plant, target loop and weight. The authors provide insights about modelling the plant and controller. The target loop should be approximated by a fairly simple transfer function at least around the crossover frequency. In order to obtain a reasonable closed loop, the plant roll-off around the crossover should not exceed -40 $db/decade$. This implies that the plant should not contain more than two "slow" or unstable poles (the speed being determined by the desired closed-loop bandwidth). This is met in most practice applications where at the crossover frequency the plant transfer function roll-off with 0 $db/decade$ or -20 $db/decade$.

A linear quadratic regulator (LQR) based PI/PID controller tuning algorithms for low-order plus time delay processes was proposed in [69]. LQR was proposed here due its robustness property. If the process is of single-input and single-output, then the control system has at last the phase margin of 60° and the gain margin of ∞. A new criterion for selection of the Q and R matrices is proposed which will lead to the desired natural frequency and damping ratio of the closed-loop system. The authors approximate the plants in process industry by a FOPTD.

In [70], PI controller tuning method for processes with large dead time is proposed. A relay feedback experiment based identification method is used to identify the FOPDT model. Then, Nyquist plot is plotted using this model. The Nyquist plot for this model is considered extremely close to that of the real process with large dead time over the frequency range of $\omega_0 \sim \omega_c$. With this estimated model, PI controller is designed based on the sensitivity specification as shown below.

$$M_s = \max_{0 \leq \omega \leq \infty} \left| \frac{1}{1 + G_p(j\omega)G_c(j\omega)} \right|, \tag{2.38}$$

The authors in [71] propose three optimal tuning PID controller scheme for industrial systems. They are time-domain optimal-tuning PID control, frequency-domain optimal-tuning PID control and multi-objective optimal-tuning PID control. In this paper, three industrial systems are used, a hydraulic position control system, a rotary hydraulic speed control system and a gasifier. It was proposed that PID controller design of industrial system is complicated by a number of factors. They are

1. the system will have non-linearity such as direction dependent actuator and plant dynamics,
2. various uncertainty such as modelling error and external disturbances are involved in the system,

3. sub-optimal tuning may be necessary to cater for changes in the system with time such as ageing and general wear,

4. commissioning is easiest without load, but the loads are often variable and affects the dynamic performance.

Due to these difficulties, the PID controllers are rarely tuned optimally and the engineers will need to settle for a compromised performance within the time available for the tuning exercise. Thus, different engineers will achieve different, sub optimal performance from same equipment Poor tuning leads to wear associated with excessive control activity, poor control performance and even poor quality products. For the time domain optimal tuning PID control equations shown in 2.4 is used. The application of the tuning proposed in this paper is the same as the ones proposed in [65]. For the multi-objective optimal tuning, integrated gasification combined cycle using a linear model of the gasifier plant model was used. The objectives used are the following.

$$\phi_i\ (k) \ = \ \frac{|\tilde{y_i}|}{y_i^d},\ i \ = \ 1, 2, ..., r, \tag{2.39}$$

$$\phi_{r+j}\ (k) \ = \ \frac{|\tilde{u_j}|}{u_j^d},\ j \ = \ 1, 2, ..., m, \tag{2.40}$$

$$\phi_{r+m+k}\ (k) \ = \ \frac{|\dot{u_k}|}{\delta u_k^d},\ k \ = \ 1, 2, ..., m, \tag{2.41}$$

$$\phi_{r+2m+1}\ (k) \ = \ 1 + \max_{j=1,2,...,n} Re(\lambda_j) + \epsilon, \tag{2.42}$$

All the three controller design scheme consists of four basic parts - model estimation, a definition of desired system specification, an optimal tuning mechanism and a PID controller.

A comparison of alternative derivative algorithms was presented in [72]. Tuning method was also presented for processes with wide range of dead-time/lag ratio. A process with gain, dead time, and first-order lag transfer function is used for the study. Three types of PID controllers were shown in the paper and were used for the comparative study. They are shown below.

$$G_c(s) \ = \ k_c \left(1 + \frac{1}{\tau_1 s}\right) \left(\frac{\tau_D s + 1}{\alpha \tau_D s + 1}\right) \tag{2.43}$$

$$G_c(s) \ = \ k_c \left(1 + \frac{1}{\tau_1 s} + \frac{\tau_D s}{\alpha \tau_D s + 1}\right) \tag{2.44}$$

$$G_c(s) \ = \ k_c \left(1 + \frac{1}{\tau_1 s} + \tau_D s\right) \left(\frac{1}{\tau_F s + 1}\right) \tag{2.45}$$

The tuning methods used for comparison in this paper provides similar response for small dead time plant models. But, for large dead time, they provide sluggish responses. The authors provide results which could be used for the selection of derivative algorithm.

A framework for PID parameter optimization is proposed in [73]. The initial parameter choices are refined using this framework. The authors suggest that there exist a posterior refinement of the obtained tuned parameters. In order to prove the stability of the optimal values, the optimization problem is set-up such that stability is guaranteed for the choice of parameter inside a pre-specified set. The method proposed in [73] indicates that the PID controller will provide a stable closed-loop system if the roots of the closed-loop characteristic polynomial, are inside the unit circle. The main goals of the paper are *a*) to obtain a characterization of achievable PID designs, *b*) to preserve stability while tuning the controller from existing PID choice. The objective function used for the optimization problem is

$$\min_{c\,=\,c^d+\delta c} ||\delta c||_2 \leq \rho^*(P_{c1}^d) \tag{2.46}$$

Where P_{c1}^d, c^d are the initial design, δc is a small perturbation on c^d, ρ is the radius of the open ball within where the values exist. The paper provide results of tuning two plant models with two other tuning methods.

A tuning method for integrating and unstable processes with time delay to meet gain and phase margin specifications was proposed in [74]. The proposed method promises a phase margin of 60°, gain margin over 3, and real part close to -0.5 in low frequencies. The following are the tuning rules proposed in [74]:

$$\begin{array}{rclrcl} k_p' &=& \frac{2\pi+3}{15kL} & k_p' &=& \frac{1}{k}\sqrt{\frac{T}{L}}+\frac{\pi}{6k}\left(\frac{T}{L}-\sqrt{\frac{T}{L}}\right) \\ k_i &=& \frac{\pi}{30kL^2} & k_i &=& \frac{\pi}{6kL}\left(\sqrt{\frac{T}{L}}-1\right) \\ k_d &=& \frac{\pi(T+0.1L)}{6kL} & k_d &=& \frac{\pi}{12k}\sqrt{TL} \\ b &=& \frac{2\pi}{2\pi+3} & b &=& \frac{1-\sqrt{\left(\frac{L}{T}\right)}}{1+\left(\frac{6}{\pi}-1\right)}\sqrt{\frac{L}{T}} \end{array} \tag{2.47}$$

This is used to tune example plant functions. In this paper, the authors used an inner feedback loop design techniques and designing controller based on gain, and phase margin specifications.

Controller design for plant models such as stable, and minimum phase plants are proposed in [75]. The frequency at which the phase derivative with respect to the frequency becomes zero is the "tangent frequency". At this frequency, the Nyquist curve tangentially touches the sensitivity circle. Relay feedback tests are used to estimate the derivatives of amplitude and phase of the plant. One of the main contributions of this paper is the use of new tuning rule which gives a new relationship between T_i and T_d instead of equation $T_i = 4T_d$. An additional condition proposed here is: the phase Bode at a specified frequency ω_c at a point where the sensitivity circle touches Nyquist curve

is locally flat. This implies that the system will be more robust to gain variations.

$$\angle \left. \frac{dG(s)}{ds} \right|_{s=j\omega_c} = \angle G(s)|_{s=j\omega_c} \tag{2.48}$$

Where ω_c is the frequency at the point of tangency and $G(s) = k(s)P(s)$ is the transfer function of the open loop system including controller $k(s)$ and the plant $P(s)$. It was indicated that gains around the nominal gain will exhibit an iso-damping property. It is assumed in [75] that ω_c, the desired tangent frequency, Φ_m, desired tangent phase, measurement of $\angle P(j\omega_c)$ and $|P(j\omega_c)|$, and the estimation of $s_p(\omega_c)$ are given. Using these values, the formula for tuning presented are

$$k_p = \frac{\cos(\Phi_m)}{|P(j\omega_c)\sqrt{1+\tan^2(\Phi_m - \angle P(j\omega_c))}|} \tag{2.49}$$

$$T_i = \frac{-2}{\omega_c[s_p(\omega_c) + \hat{\Phi} + \tan^2(\hat{\Phi})s_p(\omega_c)]} \tag{2.50}$$

$$T_d = \frac{-T_i\omega_0 + 2s_p(\omega_0) + \sqrt{\Delta}}{2s_p(\omega_0)\omega_0^2 T_i} \tag{2.51}$$

Where $\hat{\Phi} = \Phi_m - \angle P(j\omega_c)$ and $\Delta = T_i^2\omega_0^2 - 8s_p(\omega_0)T_i\omega_0 - 4T_i^2\omega_o^2 s_p^2(\omega_0)$.

A two-by-two model for a boiler-turbine unit is presented in [76]. It was proposed that such a model can capture the essential dynamics of a unit. A tuning procedure for such a model is proposed. The controller used for the tuning was a coordinated PID controller for the boiler-turbine units. The PID controllers were tuned to maximize the integral action under the constraint of a certain degree of robust stability. The following is the optimization problem used.

$$\max \underline{\sigma}(k_i) \tag{2.52}$$

under the constraint

$$\varepsilon_m := \mu_\Delta \left(\begin{bmatrix} I \\ k \end{bmatrix} (I + Gk)^{-1} \begin{bmatrix} I & G \end{bmatrix} \right) < \gamma_m, \tag{2.53}$$

where k_i is the integral gain of a PID controller, ε_m is the robustness measure, and γ_m is a given robust stability requirement. Results are shown for different boiler-turbine units with their step responses.

Modelling the plant models using fractional order systems was utilized in [77]. The paper proposes that fractional order systems could model adequately that integer order ones. A common form of fractional order PID controller is $PI^\lambda D^\mu$ controller. The transfer function of such controller has the

form

$$G_c(s) = k + p + \frac{k_i}{s^\lambda} + k_d s^\mu \ , \ (\lambda, \mu \ > \ 0) \tag{2.54}$$

In time domain, this can be expressed as

$$u(t) \ = \ k_p e(t) + k_i D^{-\lambda} e(t) + k_d \ D^\mu \ e(t) \tag{2.55}$$

Results are presented in this paper for various plant models.

In [78], a more accurate model is presented for flow process. The authors stress the importance of the presence of non-linearities in the process model and how this will affect the performance of PI controller and the tuning. In this paper, ISE criteria is used as objective function for gain tuning. The linearised model of the plant is obtained using modified relay feedback test system. Due to this, gradient descent method was used. This paper focuses more on the accurate modelling of flow process than the tuning of the controller.

Tuning rules are presented in [79] to minimize the IAE with a constraint on maximum sensitivity M_s. Tuning rules are presented for a batch of plants and for several robustness levels in terms of M_s. Set point following performance is improved as the controller is tuned by optimizing the load disturbance rejection performance. Minimization of IAE is used to select optimal set point weight. Four different robustness levels for a test batch is presented. The process will be controlled with 2 Degree of Freedom (DoF) PI controller

$$u(s) \ = \ k \left(\beta + \frac{1}{T_i s} \right) r(s) - k \left(1 + \frac{1}{T_i s} \right) y(s) \tag{2.56}$$

Where k is the controller gain, T_i is the integral time constant, β set point weight factor. Here, the maximum sensitivity function of the closed loop system is used. When the sensitivity function $s(j\omega)$ is less than a given value M_s, it implies that the loop transfer function should be outside a circle with radius $\frac{1}{M_s}$ and center at $(-1, 0)$. M_s value is selected as 1.4, 1.6, 1.8, 2.0 and are considered as different levels of robustness. Normalized dead time Z of the plant is needed for the tuning. The results are compared with other unknown methods.

A similar method based tuning was proposed by the same authors in [80]. Here, the process dynamics are characterized using three parameters the static gain k_p, gain k_{180}, and frequency ω_{180} where the phase lag is 180°. The PI controller is tuned by optimizing the load disturbance rejection performance. This is very similar to the methodology proposed in [79]. An important difference here is the use of κ, the corresponding frequency parameter of relative time delay τ. The plant model shown in [79] was used as the plant model for tuning.

Optimal PI controller parameter of a quadruple tank process was obtained in [81] using swarm intelligence method. In this paper authors use ITAE as the objective function. A linearised state space equation was used to represent the quadruple tank process. Step response for minimum phase and non-minimum phase are used for the ITAE calculation.

A PI tuning method for modular drive control is shown in [82]. The results are compared with the Ziegler Nichols tuning method results.

A symbolic tuning for first order systems with time delay based on Pade approximation is proposed in [83]. This paper also needs the model of the plant and it needs to be limited to be a first order model with time delay.

2.2.1 Survey papers on gain tuning

Survey papers on gain tuning in the past have discussed about different tuning methods and rules that were available in the literature [84, 85]. They have also provided comparison between different methods/rules available [86, 87, 88] and also comparison with classical tuning methods [89]. Some presented a compilation of the rules that are available [90, 31] while some presented methods specific for certain plant models [91, 92].

2.2.2 Optimization based methods for gain tuning

Optimization based tuning of PI controllers were proposed in the literature. Genetic algorithm (GA) is one of the most commonly used to tune PID controller. In [93] the paper utilizes time weighted integral performance criteria for its optimization. First-order plus dead time model is used as the plant model for the tuning. Integral of the squared error (ISE) is shown in 2.57

$$J_n(\theta) = \int_0^\infty t_n e(\theta, t)^2 \, dt \tag{2.57}$$

where θ determines the variable parameters which can be chosen to minimize $J_n(\theta)$. The authors also suggest the use of other criteria like integral square time error (ISTE), integral square time squared error (IST^{2}E) and also the use of critical point data of the plant in the Nyquist locus. The critical point of the Nyquist locus is the point where the phase is 180°. The frequency is called the critical frequency and the modulus of the gain is $1/k_c$ where k_c is the critical gain. The authors use least squares fit and obtain an empirical formula for the gain parameters. In this paper relay tuning method is used to obtain the optimum gain parameters.

In [94] a proper objective function is used to define the performance of the PI Smith controller. The proposed technique has two steps. First, GA is used to identify a model for process using

on-line data. Second, identified model is used to tune the PI Smith predictor controller, to minimize a time-domain based cost function. The resultant offline tuning based gain values will then be used on the real process. The authors use time domain based criteria as an objective function for the tuning method.

A similar approach was taken in [95] to obtain the gain parameters of PID controller. A tuning formula was obtained for various process. Like most approaches, this also has a couple of steps to derive the formula. First, genetic algorithm is used applied to design the PID controllers for a variety of processes. Second, a correlation between the controller parameters and the parameters of the models is found. The formula is finally derived using this correlation. A unit step set-point change and unit step disturbance is given to the models and the corresponding error is used to calculate the integrated absolute error (IAE). This is shown in 2.58, 2.59, and 2.60. Equation 2.60 is used as the objective function of the optimization.

$$J_s = \int_0^\infty |e_s(t)|\, dt \tag{2.58}$$

$$J_d = \int_0^\infty |e_d(t)|\, dt \tag{2.59}$$

$$J = J_s + J_d \tag{2.60}$$

A methodology to tune non linear process models is proposed in [96]. A thermal process is used to show the performance of the proposed methodology. IAE is used as the objective function for the tuning methodology. The variables for the optimization are k_c gain constant, and T_i integral time constant. Also, as an improvement to the methodology, a small error was introduced into the process model and tuned with this modified model. Finally, a noise was added to the input and the gain parameters for the process model with the noise was tuned using the same model. A comparison between conventional and evolutionary tuning methods for PI and fuzzy PI controller is presented in [97]. The process model used for the tuning is a laboratory flow rig. The system response mainly depends on the characteristics of the actuators, transducers, transmission lines and controller. In this paper, the authors used k_p and k_i as variables. The authors suggest the use of such gains instead of T_i helps the computation program. Different objective functions are used for the tuning proposed here. The initial objective function uses the product of the IAE-value, a scaled, inverted and modified Gaussian function as valuation function for the phase margin and additionally a controller output observer. The second objective function replaces the output observer by two tangent-sigmoidal functions as valuation criteria for phase margin, crossover frequency and a smooth controller output without large overshoot. The paper also proposes using ITAE and IT^{3}AE as objective function. Four tests were used to test the generated gain values. Behavior of the

controller for large set-point changes, stepped reference input changes for different stepped change, and finally disturbance rejection. The paper presents comparison between these different tuning strategies based on GA.

A Ziegler-Nichols based tuning is proposed in [98]. In this paper, higher order process, a large time delay plant and a high non-minimum phase process are used as process models to tune k_p, T_i, and T_d. In this paper, the critical period method was used. The critical period method consists of determining the point where the Nyquist plot of the open-loop syystem intersects the negative real axis. This point obtained by connecting a purely proportional controller to the system, and by increasing the controller gain until the closed-loop system reaches the stability limit, at which oscillations occur. The oscillation period is denoted by T_c and the corresponding critical gain by k_c. High-order process, large time delay plant, and non-minimum phase process are used as process models for the gain tuning.

A cascaded GA based algorithm was proposed in [99]. GA was proposed for system identifictaion and PID tuning for optimum adaptive control. A step reference command signal was applied to the plant and the transient response is observed. From the response, GA estimates the plant transfer function. The plant transfer function is obtained through the integral squared error minimization between the measured transient response and the interior GA simulation transient response. For the PID tuning, the sum of the squared error between the reference input test signal and the simulated system response is used as objective function.

In [100], a discrete version of extremum seeking (ES) is utilized to optimize a step response of the controller. The ES algorithm iteratively modifies the arguments of a cost function so that the output of the cost function reaches a local minimum or local maximum. The cost function used in this paper is integrated square error as shown in 2.61

$$J(\theta) \;=\; \frac{1}{T \;-\; t_0} \int_{t_0}^{T} e^2(t,\theta)\; dt \tag{2.61}$$

Where $e(t,\theta)$ is the error signal which is the difference between the reference and the output signal of the closed-loop system. The proposed ES iteratively modifies the input of the cost function to reach a local minimizer. ES achieves this optimization by sinusoidally perturbing the input parameters of the system and estimating the gradient.Four transfer function models are used in the paper for tuning. The results are compared with other tuning methodologies. ISE cost function along with IAE, ITAE, and ITSE are used to compare the results. The authors suggest here that using a cost function comprised of the squared error (ISE and ITSE) versus teh absolute error (IAE and ITAE) decrease the time required for the output of a closed-loop system to initially reach the set point. The

authors suggest that even though ES is difficult to implement it can provide improved performance results.

In [101], a model based approach to tune a PI controller for first-order plus time delay (FOPTD) plants using second-order plus time delay (SOPTD) and FOPTD responses. The authors suggest that the response of the PI tuning on a FOPTD plant results in a response which has an upper SOPTD bound. The authors suggest that the time delay and damping factor of response must be greater or equal than those of such a bound. The response also has a lower FOPTD bound whose time delay and damping factor must be greater or equal than those of response. The authors also suggest that the tuning of PI controller can be considered ass a convergence procedure of upper and lower bounds and no unique solution for PI tuning. Depending on the aggressiveness of design, tuning of PI parameter may differ a lot.

PID parameters for a model bio reactor using GA is shown in [102]. A least square method is used to linearize the bioreaetcor model and the obtained transfer function was used for the gain tuning process. The performance of the gain tuning was compared with ZN method and Skogestad method for robustness and steady state performance.

A convex optimization based approach to PI controller tuning for first order lag plus delay plant is proposed in [103]. Again, Integrated absolute error is used as an objective function to be minimized by parameter tuning. The authors suggest that controller parameters governed from tuning rules are not guaranteed to be the global minimum of controller performance and it is often possible to reach a lower controller performance value by manual re-tuning.

Multi-objective optimization based

A multi-objective optimization based tuning was proposed in [104]. The paper also uses first order plus dead time processes. The authors present the conflicting requirement nature of the PI tuning problem. The authors present insight about the need for load disturbance rejection, set point regulation, robustness to model uncertainties. The authors present insight about conflicting objectives. The following are used as objective functions are minimised

- IAE criterion to a load disturbance step
- IAE criterion to a set point step
- Maximum sensitivity (M_s)
- Total variation (TV)

2.3 Proposed tuning methodology

There is a need for a tuning methodology which could work for different software and hardware platforms. Optimization based tuning is proposed here to tune PI controller. Different plant models are used as examples to show the effectiveness of the tuning algorithm for different plant models. A small photovoltaic system powering a load modelled with the actual switching model is used as the plant model and a PI based voltage controller was used as the controller model. The gains for this voltage control is tuned using the proposed algorithm. This optimization based algorithm is then used to tune plant and control models programmed in real time digital simulator.

Initially, the plants modelled in software will be tuned. Then, a hardware-in-loop simulation based gain tuning is proposed using the same optimization based tuning. This proves the capability of the proposed approach to reuse the methodology over a range of software and hardware based PI controllers. More details about the optimization algorithms used to solve the PI tuning problem is presented in chapter 5. The actual objective functions used for the gain tuning and the results showing the effectiveness of the proposed gain tuning is shown in chapters 4, 5, and 6.

2.4 Summary

From the above literature review, the following are observed:

- The gain tuning methods proposed in the literature reduce the industrial and chemical process to FOPTD, SOPTD, and other mathematical function.
- Seldom tuning methods use experiments for gain tuning.
- Insufficient number of papers transfer the tuned gain values from simulation based tuning to actual hardware based PI controller.
- In most methodologies, curve fitting is used to obtain formulas for gain values from the results obtained from exhausting simulations.
- There is no evidence presented in these papers to address the effect of model mismatch between the model used for tuning and the actual model.
- There is no discussion which involves the gain value changes between different software, hardware used to model PI controller and plant model.

In order to tune the PI controller over different software and hardware platforms, optimization based tuning is proposed here. The approach taken to identify the effectiveness of the proposed

method is shown in Fig. 2.3. This figure proposes strategy that tests the proposed tuning methodology in different software platform and uses parallel programming as a methodology to increase the speed with which the PI gains will be tuned. In addition to the work indicated in Fig. 2.3, controller hardware-in-loop experimental setup based gain tuning of FPGA based PI hardware is proposed. The results from this gain tuning approach is shown in chapter 6.

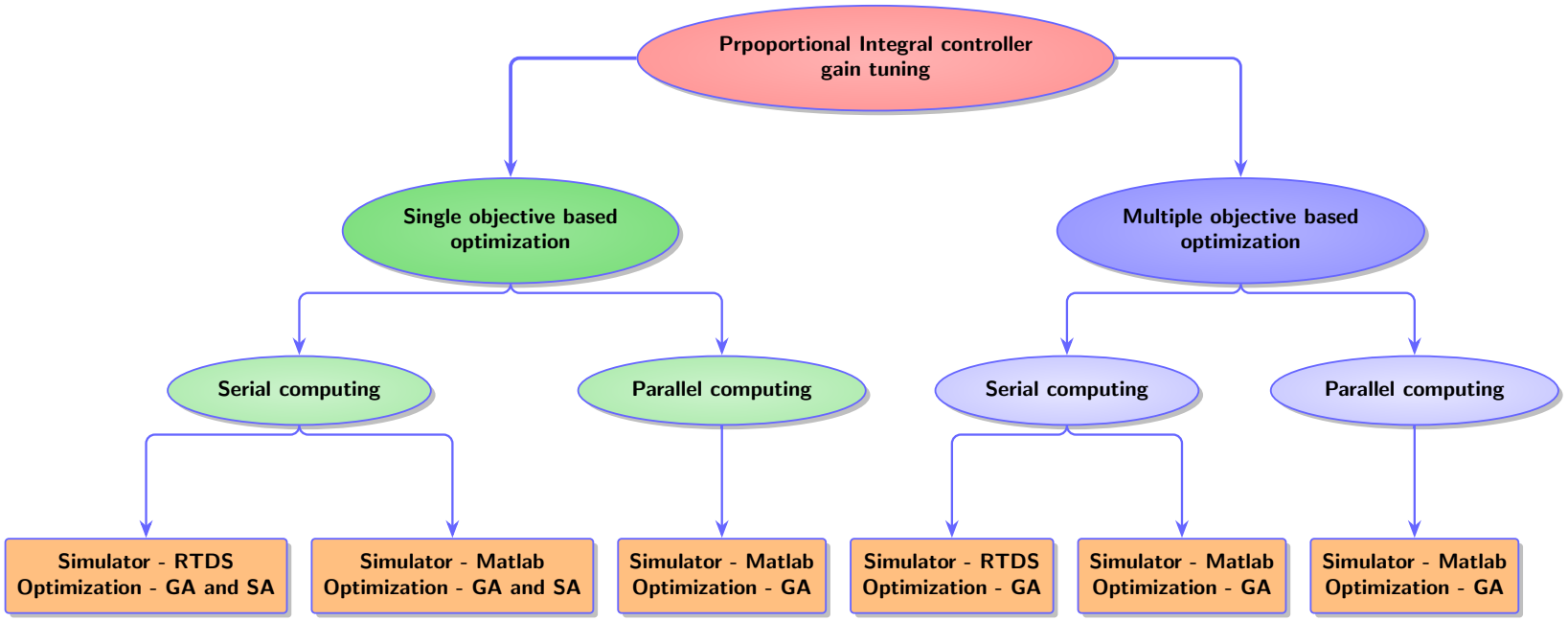


Figure 2.3: Structure of proposed serial-parallel computing and single-multiple objective approach for solving gain tuning optimization problem

Chapter 3

Software based gain tuning with parallel computing

This chapter presents introduction to the optimization algorithms that is used to solve the gain tuning problem. Genetic algorithm (GA) and simulated annealing (SA) are the algorithms which are utilized to solve the single objective and multi-objective optimization problems.

3.1 Genetic algorithm based single objective optimization

Genetic algorithms fall under the broad classification of evolutionary algorithms (EA) [105]. The following is a generic characterization of evolutionary algorithms:

- EA uses a collection of population of individuals. This collection of individuals represent the potential solution space to the given problem. EA algorithms utilize characteristics learned by these individuals.
- The off-springs generated by such individuals are a result of randomized process. The randomized processes utilize mutation and recombination to generate individuals. Mutation is a result of erroneous self-replication of individuals. Recombination is the exchange of characteristics between two or more existing individuals.
- Finally, these individuals are evaluated for performance. Every individual is given a measure of the fitness value. These fitness values have a binary decision: better or worse. According to this decision, the reproduction process prefers better individuals to reproduce more often.

Genetic algorithm is a stochastic method that imitates the evolution [106]. The following are some of the unique characteristics of genetic algorithm:

- Selection rules,
- Elite individuals,
- Recombination by crossover, and
- Mutation.

The following terminology will be used in the rest of this chapter to present the steps involved in Genetic algorithm:

- Fitness function - Fitness function is the main objective function that needs to be optimized.
- Individuals - Individuals are the data points that are present in the search space. This can also be called as the genome with the vector entries of the individual as genes. Typically, the fitness function for any individual becomes its performance score.
- Population and generations - Population is the list of individuals. Depending on the number of individuals, the size of population will vary. Generations are the list of population that is generated after every iteration.
- Diversity - Diversity refers to the average distance between the individuals in a population.
- Fitness values and best fitness values - Fitness value is the value of the fitness function for a particular individual. Best fitness value is the best fitness value in a population of individuals.
- Parents and children - Parents are set of individuals that are selected from the current population to create the next generation. Individuals that are created in the next generation are called children. Individuals with better fitness values have more probability to get selected as parents.

3.1.1 Selection rules

Selection rules are used to select the parents. Parents are the individuals inside the population which contribute to the population of the next generation. The selection function is used for scaling the individuals by scaling the fitness function performance.

3.1.2 Elite individuals

Elite individuals are individuals in the current generation with the best fitness value among the other individuals. Elite count is the number of elite individual that will move to the next generation. If the elite count is low, a low number of elites in the present population goes to the next generation. If the elite count is high, a high number of present population will move to the next generation.

3.1.3 Crossover for recombination

GA utilizes crossover to create crossover children. A pair of parents in the present population are selected and random entries are selected from the two parents and assigned to the children. This is crucial in GA because this helps the algorithm to obtain the best genes from different individuals and recombine them to form superior children.

3.1.4 Mutation

The process of randomly changing the genes of individual parents and creating children from them is known as Mutation. This is another crucial step in GA because this adds diversity into the population thus induces more chance of generating superior children.

In order to understand these concepts, consider the following scenario: *a*) Population size - 30, *b*) Elite count - 3, *c*) Crossover function - 0.6.

- With the elite count set as 3 and population size set as 30, the top 3 individuals will go to the next generation.
- With 3 out of 30 generated from elite individuals, the rest 27 will be generated using cross over and mutation.
- Since the crossover fraction is 0.6, the number of crossover children is $0.6 \times 27 = 16.2$. This is rounded to 16.
- The remaining 11 individuals are the mutation children.

3.1.5 Stopping criteria

The fitness scores are scaled and are returned to the selection function to make the fitness scores suitable. The following conditions can be used to stop the algorithm: *a*) Generation limit - the algorithm will stop once reaching the allotted generation number. *b*) Time limit - the algorithm will stop after the time limit. *c*) Fitness limit - algorithm will stop when the fitness limit is less than or equal to the fitness limit. *d*) Stall generations - algorithm will stop when the average change in

fitness function value is less than the function tolerance. *e*) Stall time limit - this is a time limit which checks if there is no improvement in the fitness function during an interval of time in seconds. *f*) Function tolerance - function tolerance is the average relative change in the fitness value over stall generations. The algorithm will stop if this value is less than function tolerance.

3.1.6 Outline of genetic algorithm

The general outline of genetic algorithm is shown in Algoritm 1. Genetic algorithm begins the algorithm by generating a random initial population. The algorithm creates new population by using the individuals in the current generation. To create new population, the algorithm utilizes a step by step approach. The algorithm scores each member of the current population by computing its objective function value. Depending on this, members called parents are selected. Some elite members with low objective function value is also selected as parents. The algorithm produces children from these two sets of parents. Children can be generated by making random changes to a single parent (mutation) or by combining vector entries of the parents (crossover). The generated children values are ranked again and the whole process is repeated until stopping criteria is met.

Algorithm 1: Generic genetic algorithm

```
1  Initialization,
2
3  Generate N population,
4
5  while Stopping criteria not met do
6  |
7  |  Compute each members' fitness value,
8  |
9  |  Score each member of the current population according to the calculated fitness value,
10 |
11 |  Members or individuals with lower fitness are chosen as elite. They are passed to the next
   |  population,
12 |
13 |  Off springs can be produced by making random changes to single parent (Mutation).
   |  Offspring can also produced by combining vector entries of a pair of parents (Crossover),
14 |
15 |  Next generation off springs are generated from parents using mutation and crossover,
16 |
17 |  Replace the existing population with new children and form new population,
```

3.2 Simulated annealing based single objective optimization

Annealing is a metallurgic process by which a metal or glass is heated and cooled slowly, in order to remove internal stresses and toughen the material. As the name indicates, "simulated annealing" simulates the metallurgy process in order to minimize the system energy. Annealing typically consists of two steps: *a*) the temperature of the heat bath will be increased to a maximum value to melt the solid under annealing process, *b*) the temperature is slowly reduced and the particles of the molten solid arrange themselves in the ground state of the solid. When the solid melts, that is, during the liquid phase, all particles of the solid arrange themselves randomly. When the liquid cools off, that is during the ground state the particles are arranged in a highly structured lattice and the energy of the system is minimal.

Algorithm 2: Simulated annealing algorithm

```
Initialization,

Generate initial point and temperature information,

Evaluate the function value at the initial point (FV_t),

while Current temperature > Minimum temperature do

    Make a random move and evaluate the randomly generated point (FV_{t+1})

    if Function value_{t+1} > Function value_t then

        Generate a random number between the high and low temperature value,

        if Random number generated > Present temperature then

            Remove the moved value,
            Decrease the temperature by ΔT, t = t − ΔT and move to the beginning of while
            loop,
        else
            Keep the moved value,
            Decrease the temperature by ΔT, t = t − ΔT and move to the beginning of while
            loop,
    else
        Keep the moved value,
        Decrease the temperature by ΔT, t = t − ΔT and move to the beginning of while loop,
```

When the algorithm is initiated, a temperature value is assigned to the function. Next, a random change is made and an evaluation for the function is performed at this new point. Depending on the value of this new function evaluation, it will either be stored or discarded. If the new point improves

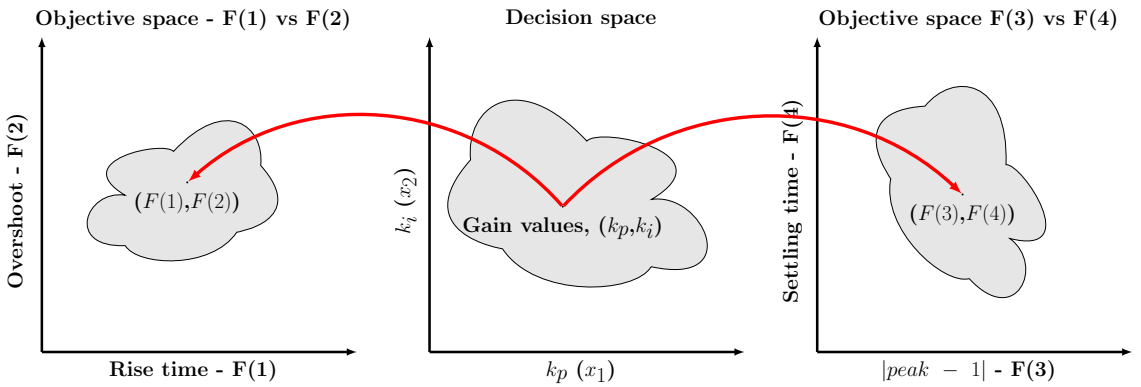


Figure 3.1: Pictorial representation of decision space and objective space

the function, it will be stored. Otherwise, the concept of temperature comparison is introduced and the new point is discarded.

After this, a new random number is generated within the range of maximum and minimum temperatures and is compared to the value of the current temperature. If the current temperature is higher than this randomly generated value, then a move from previously known good value to the present "inferior" generated value is made. If the current temperature is lower than the generated random value, this value is ignored and the process repeats.

This allows the algorithm to climb out of local minimum value and move towards global minimum value. This could only be achieved at high temperatures and these random moves are unlikely during low temperatures. The function will terminate once the temperature decreases as time progresses. During the start of the iteration, random moves are made at a higher step. That is, the distance between the steps are more during the beginning. And during the end, when the program nears reaching global minimum, the distance between the steps is reduced. This enables the algorithm to search closely near the global optimum value to find a value close to the true minimum. A very simple version of these steps are shown in Algorithm 2. More detailed analysis could be found in [107].

3.3 Introduction to multiple objective optimization

Multi-objective optimization problems possess the need to consider multiple, and conflicting objectives while obtaining the optimal solution. The following is a general multi-objective optimization problem.

$$\min_{x \in \mathbb{R}^n} F(x) = (f_1(x), f_2(x), \ldots, f_m(x))^T \tag{3.1}$$

where $f_t : \mathbb{R}^n \rightarrow \mathbb{R}^1$, and $f_1(x), f_2(x), \ldots, f_m(x)$ are a given set of criteria. The functions $f_1(x), f_2(x), \ldots, f_m(x)$ need to be optimized in order to minimize $F(x)$.

One of the striking differences between single objective and multi-objective optimization is the multi-dimensional nature of the objective space. Fig. 3.1 shows the mapping that takes place between the decision space (k_p, k_i) and the objective space($f_1(x), f_2(x), \ldots, f_m(x)$). For every k_p, and k_i there exists a function value ($f_1, ..., f_m$)in the objective space. In principle, the seach space can be divided into two non-overlapping regions (in the context of multiple objectives). One being optimal and the other being non-optimal. In case of conflicting objectives, usually the set of optimal solution contain more than one solution. In the presence of multiple Pareto-optimal solutions, it is difficult to prefer one solution over the other without any further information about the problem. There are two main goals in a multi-objective optimization. They are:

- to find a set of solutions as close as possible to the Pareto-optimal front, and
- to find a set of solutions as diverse as possible.

There exists a set of solutions for the multiple objective case called non-dominated solutions or Pareto optimal solutions [108]. The following two conditions must be true for a non-dominated set P,

- Any two solutions of P, must be non-dominated with respect to each other,
- Any solution not belonging to P, is dominated by at least one member of P

There is not a solution that is best with respect to all the objectives in multi-objective optimization [108]. One solution can be better from the perspective of one objective while worse in another. Finding a set of solutions as diverse as possible and as close as possible to Pareto front is very important. Gain tuning is not traditionally considered as multi-objective optimization. But, the results indicate that this behaves like a typical application for multi-objective optimization.

In multi-objective optimization problems, multiple objective functions are needed to be optimized simultaneously. There exist a set of solutions for the multiple objective problems and such solutions are called non-dominated solutions or Pareto optimal solutions [109]. There may not be an optimal solution that is best with respect to all the objectives in multi-objective optimization [109]. One solution can be better in one objective while worse in another. Finding a set of solutions as diverse as possible and as close as possible to Pareto front is very important. In the proposed research work multi-objective genetic algorithm is used to solve the multi-objective gain tuning problem.

3.4 Genetic algorithm based multiple objective optimization

The concept of GA was developed by Holland and his colleagues in 1960s and 1970s. In GA, a population of candidate solutions to an optimization problem is evolved toward better solutions. There are two reasons which motivates the use of multi-objective GA, when compared to other multi-objective solvers. The multi-objective GA does not require the function to be differentiable or continuous. This fits our needs since the non-linear, non-differentiable, and non-continuous switching model is employed in this work. Another important reason for using multi-objective GA here is that it could solve optimization problems in parallel as mentioned previously. That is, within each generation, the calculations are mutually independent. Thus, this is a natural fit for high parallelism.

This chapter has presented introduction to algorithms used to solve single objective optimizations. This section presents an introduction to solving multiple objective optimization with genetic algorithm based approach. Typically, some multiple objective problems are formulated as single objective with several constraints. This approach may not adequately represent the optimization problem. Consider the vector

$$F(x) \;=\; [F_1(x), F_2(x), ..., F_m(x)] \tag{3.2}$$

The vector equation 3.2 is a vector of the objective formulation. These objectives must be traded off in some way. Trade off between the objectives are not fully understood until the system's capabilities are understood. Unlike a weighted, sum of single objective, in a multiple objective optimization, the trade offs and quantification of objectives are unknown. Some of the objectives in the vector, may be competing with each other. In this case, there is no unique solution.

Under this circumstance, concept of Pareto optimal must be used to characterize the objectives. This is also known as non-inferior solution. A Pareto optimal solution or a non-inferior solution is one in which an improvement in one objective requires a degradation of another.

3.5 Parallel computing based optimization

Parallel computing is designed to circumvent physical and mechanical constraints on individual processor speed. It allows the user to solve data intensive and computational intensive problems using multi-core processors, graphic processing units (GPUs), or computer clusters. For the work proposed here, computations are performed using multi-core processors. The number of virtual parallel computers that is used for this multi-core application is up to 12. For future reference, these virtual computers will be referred to as "workers". The generic approach presented here could be used with GPUs or for multiple clusters. One primary requirement must be met in order to use such

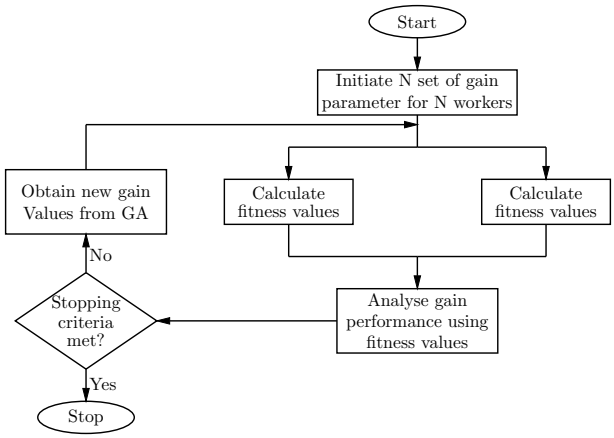


Figure 3.2: Parallel computing flow chart for two workers

parallel computing capability. That is, the computations to be paralleled should be independent from each other. Our tuning strategy fits this requirement since the performance of the gain values can be computed within each generation, which is also the reason that genetic algorithm is selected for optimization due to its easy implementation in a parallel computing architecture.

The workers are used to perform the power simulations and then return the step response values. There will be one supervisor for these workers. The supervisor will perform the genetic algorithm based optimization. Initial gain values will also be generated and sent to the workers to generate the response for that particular gain values. The workers after the computation, will inform the results to the supervisor. Since, the supervisor performs GA based optimization, more detailed explanation of the function of the supervisor will be explained in the next section.

For every generation in GA each worker will be given a set of gain values (a set of offsprings) to obtain the multiple objectives as indicated. Fig. 3.2 shows the flow chart explaining this procedure.

To evaluate the performance of parallel computing, the speedup and efficiency will be employed [109, 110]. More specifically, the speedup of the wall-clock running time of parallel computing, S_r, is to quantify the improvement of running time with multiple parallel processors. The formula used to calculate S_r is shown in (3.3).

$$S_r = \frac{T_{seq}}{T_{par}} \tag{3.3}$$

where T_{seq} is the time taken to perform the (sequential) computation using one worker and T_{par} is the time taken to perform the computation in parallel with more than one worker.

In addition, the efficiency of parallel computing, E_r, indicates the increase or decrease in the efficiency with different number of workers when compared to sequential computing (i.e., one worker).

The formula to calculate E_r is shown in (3.4).

$$E_r = \frac{S_r}{N_p} \tag{3.4}$$

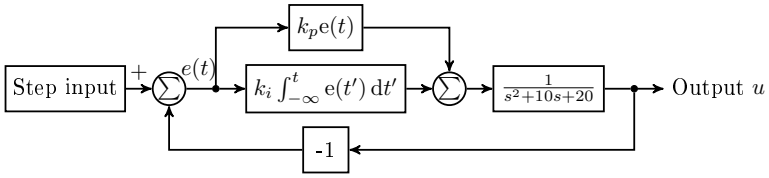
where N_p is the number of parallel workers used and S_r is the speedup described in 3.3.

3.6 Models used for gain tuning problem in different software and hardware platforms

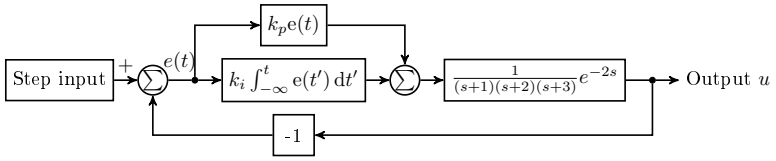
Matlab - simulink and RTDS are used to program the plant models. In both software platform, the plants are modelled using transfer function initially and then a switching model based power system model is used. More details about these plant models are explained later. Gains for the PI controller to optimally control the plant models are tuned using single objective and multiple objectives. Genetic algorithm is used to solve both single and multiple objectives, and simulated annealing is used to solve the single objective problems.

3.6.1 Process and control model used in computer based simulation

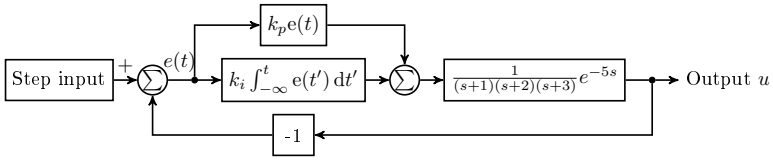
Transfer function based plant models used to verify the proposed gain tuning methodology are shown in Fig. 3.3. In these four examples, the plant are modelled as transfer function.



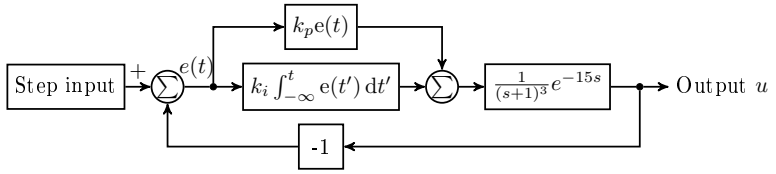
(a) Model 1 - used for plant simulation



(b) Model 2 - used for plant simulation



(c) Model 3 - used for plant simulation



(d) Model 4 - used for plant simulation

Figure 3.3: Process models used for tuning in computer based modelling

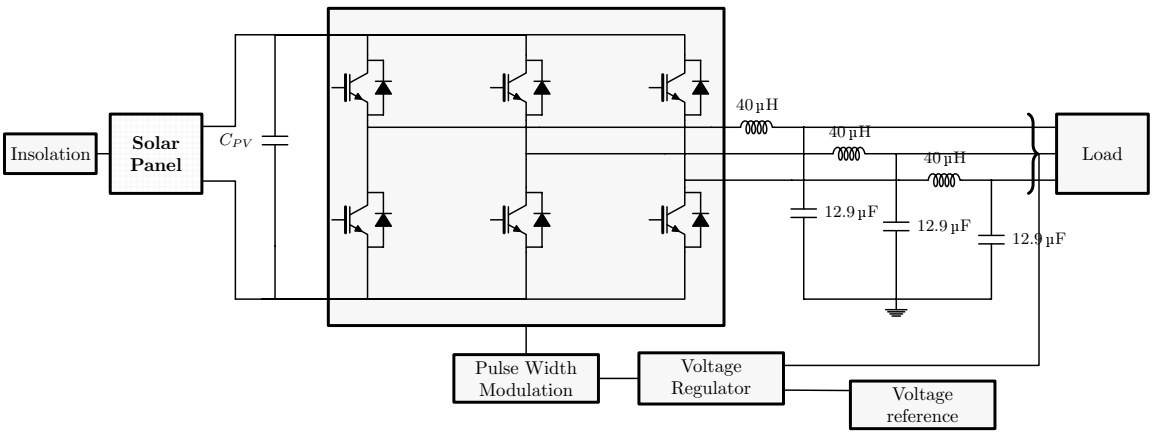


Figure 3.4: Photovoltaic model used for gain tuning tuning with voltage regulator

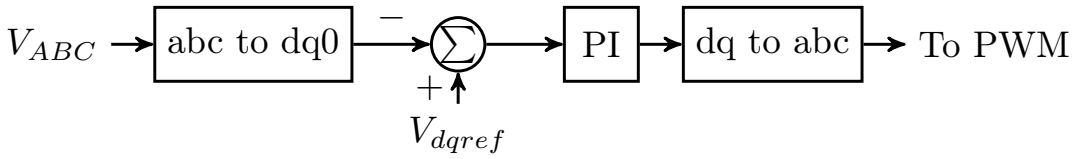


Figure 3.5: Detailed structure of the voltage controller used in the Photovoltaic gain tuning model

Another simple system is also used to test the performance of the tuning strategy since the objective is to present a strategy to tune PI controller rather than the controller topology. The model used to demonstrate the PI tuning method has a photovoltaic (PV) panel connected to an RLC load through a DC/AC converter and a filter. This plant model is chosen for this study due to the growing interests and popularity of PV integration to smart grids and microgrids. The pulse-width-modulation (PWM) used for the inverter is controlled using a PI controller. The PI controller obtains the voltage feedback at the point-of-common-coupling (PCC) and uses this as the feedback value. The primary goal of the PI controller is to regulate the voltage at the PCC which is set by the voltage regulator. The rated voltage at the PCC is 208 . The plant model used for testing this algorithm is shown in Fig. 3.4.

Fig. 3.5 shows the block diagram of the voltage control. The three phase voltage is converted into $dq0$ transformation and then subtracted from the reference. This error signal is fed into the PI control block. Then, it is used as a sinusoidal reference for the triangular wave-form to generate PWM signals. Other test systems shown in [40] can also be used for tuning. PI gain tuning has been revisited time to time due to the increasing complexity of the dynamics of the plant models. For simpler models, tuning rules available in the literature can be used to get a good performance

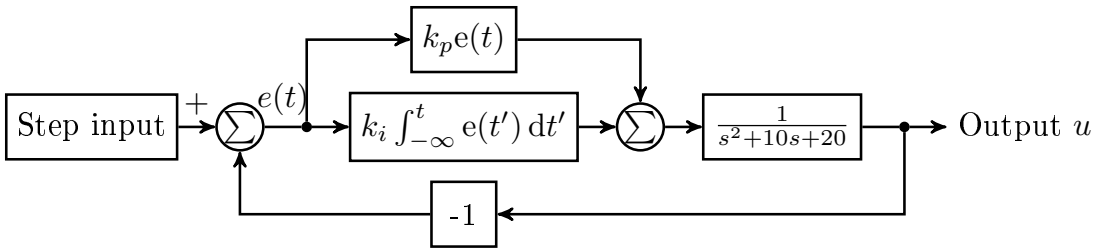


Figure 3.6: Plant model used in the real time digital simulator

from the controller [111]. For complex models, PI tuning is predominantly done by trial and error [44]. Some simple yet effective tuning methods that exist for simpler plant models are Ziegler-Nichols method [32], dominant pole design [112], simplified frequency domain methods, and other pole placement methods [113, 114]. As mentioned earlier, these tuning methods need reduced models of the plant in order to tune the gains. Depending on the tuning methodology, either step response or the frequency response may be applied to tune. For the method used here, the step response of the system is used. The key difference between the existing PI tuning methods and the method used here is that the tuning methodology is not specific to a particular plant model. For the optimization algorithm, the system is a complete black box. It observes the performance of the plant by utilizing the step response. The algorithm quantifies the step response by the rise time, overshoot, settling time and peak value. This type of black box based tuning is of crucial importance because a new simplified model need not be generated for different power electronics topology. Thus, it is more realistic when applied to practical applications. The algorithm will require just the switching model of the power electronics topology.

3.6.2 Model used in real time digital simulator based process and control model

Figure 3.6 shows the plant model used in the real time digital simulator to tune the PI gain values. This is the same model as "model 1" used in the computer based modelling. This is the only model used in real timed digital simulator to tune PI gain values. The advantage of using RTDS for the plant model simulation is that the models are simulated in real time. In order to observe the performance of a plant model from time at $t = 0$ seconds to time at $t = 10$ seconds, it will take only 10 seconds in real time digital simulator to perform the simulation. This could be used to test controller hardware and relay hardware in real time without the need for actual hardware plant models.

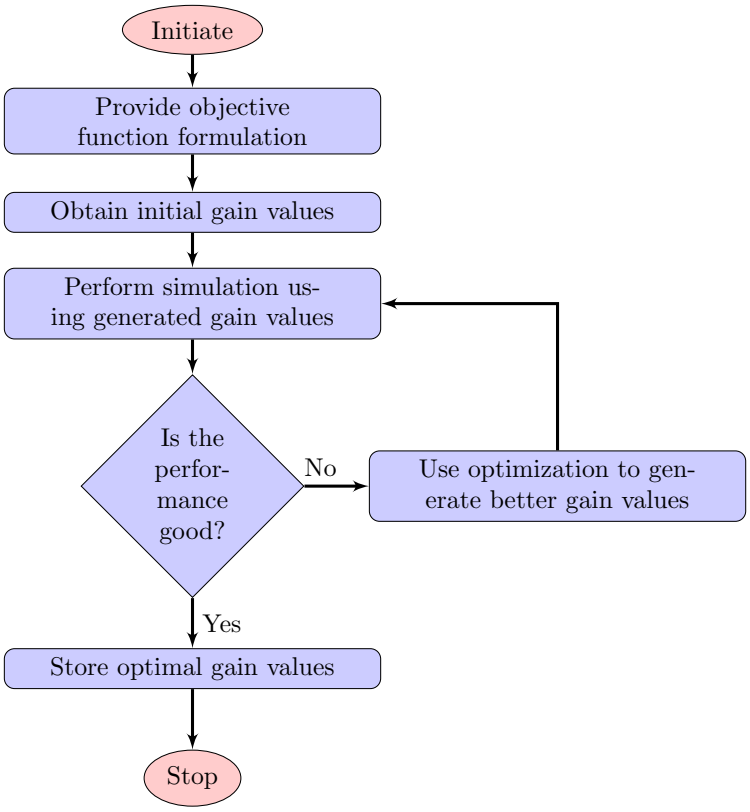


Figure 3.7: Generic optimization flow for gain tuning

This provides a huge advantage over modelling complex power system/power electronics models which will result in prolonged simulation time for even 10 seconds of model simulation. This is also a disadvantage when simple transfer function based models are simulated using the real time digital simulator. Due to this reason, only one transfer function based model is tuned using the RTDS. Also, tuning transfer function based models is not the final goal of this research. Tuning such transfer function model is just a necessary step, that need to be performed before moving to power system model tuning in real time digital simulator. A power system based plant model will be simulated in the real time digital simulator and the PI controller will be moved to FPGA based hardware. Model description and the controller model is described more in detail in chapter 5.

3.7 Proposed gain tuning methodology

In our proposed gain tuning methodology, optimization algorithm identifies the plant model as a black box, which indicates that in no way the plant model will be utilized in the objective function nor in the constraint. Regardless of the power electronics topology the program need not be changed

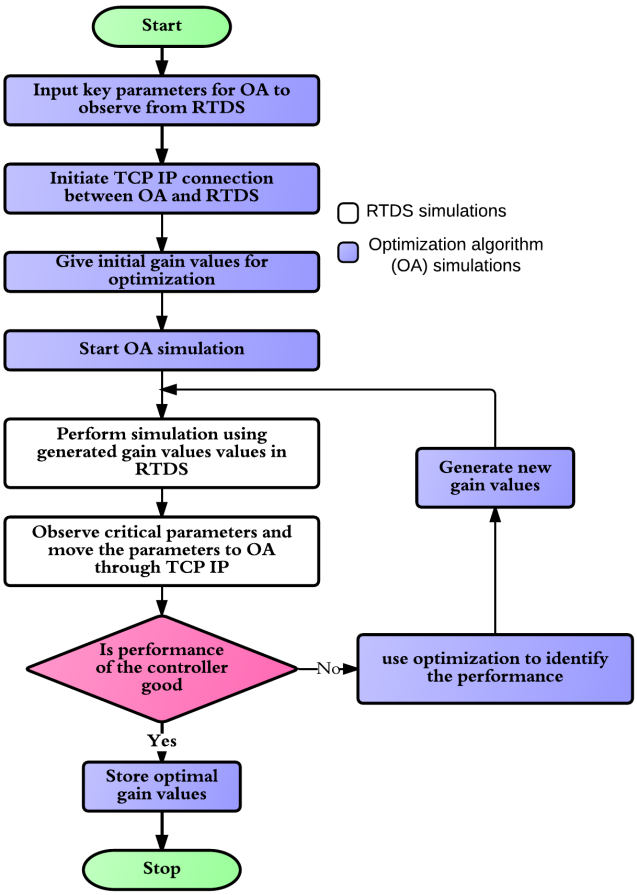


Figure 3.8: Gain tuning flow chart using multi-objective optimization and real time simulator

every time. Even though the simplified mathematical models are available for most existing topology, the power system model in which these PI controllers will be used, the control aim of these PI controllers and the outcome of these controllers vary a lot. It is very hard to justify deriving a mathematical model for each and every unique power system control objective.

Gain tuning process flow for computer based simulation of plant model and control model

A generic flow chart for gain tuning by utilizing optimization methodologies is shown in Fig. 3.7. The initial step is to identify proper feedback signals from the plant which can indicate the effectiveness of the PI controller. A generic signal would be the feedback signal. This is used to calculate the error by subtracting this signal from the reference.

Then, depending on the application, proper step response characteristics should be used as objectives for the optimization algorithm. After these steps the optimization solver will generate initial values (or the user can provide initial values) and then the solver will minimize the objectives thus providing optimal gain values.

Optimization tuning flow for real time digital simulator based simulation of plant model and control model

A high level optimization flow chart for tuning PI controller when the plant and the controller are modeled in real time digital simulator is shown in Figure 3.8. There are two parts to the gain tuning. First, the optimization algorithm (OA) which is either genetic algorithm or simulated annealing. Second, the simulation of plant models in the real time digital simulator. The real time digital simulator is controlled using TCP-IP connectivity. The data is stored by the RTDS in a common database in which the optimization algorithm also has access to.

3.7.1 Objective function formulation for optimization

The characteristics of step response indicated in chapter 2 is used in the objective functions. Some of these parameters need to be modified since they cannot be used directly as the objective functions. The following are the modeled step response parameters used as the multi-objective functions

1. Rise time (minimize)
2. Overshoot (minimize)
3. $|peak - expected\ peak\ value|$ (minimize)
4. Settling time (minimize)

Emphasis should be given to the second and third objective functions. Because, they might seem very redundant. But, a closer look will provide a different outlook. Consider the plant being controller and the step input given to the controller is 1. If the step response reaches a value of 1.1, the overshoot is 10 % and the peak response will be 1.1. The values might look redundant for this example. But, consider another example with a peak response of 0.8. Then, the overshoot becomes zero, while the peak is 0.8. The second objective function value will become 0 and third one will become 0.2. Even though, these values are minimum for the corresponding objectives, a response of 0.8 is undesired (1 being the desired response).

For the models 1 - 4, the set point reference is 1 and thus the expected value is 1. Generically, any response beyond the expected steady state value will provide similar second and third objective.

$$\min \left(\quad risetime(u), \quad overshoot(u), \quad |peak(u) - expected\ peak\ value|, \quad settlingtime(u) \right) \tag{3.1}$$

Where u is the output shown in Fig. 2.1

$$\min \left(\quad risetime(V_{total}), \quad overshoot(V_{total}), \quad |peak(V_{total}) - 3 \times 208|, \quad settlingtime(V_{total}) \right) \tag{3.2}$$

$$Where\ V_{total} \ = \ V_{A_{RMS}} + V_{B_{RMS}} + V_{C_{RMS}}$$

But, if the response is below the expected steady state value, the second and third objective will be different. Thus, in order to avoid this inherent trap while optimizing, both second and third objectives are used. Multi-objective optimization is useful in finding the gains because the PI response of the system consists of multiple parameters. Thus, it is better to optimize it in a multi-objective way instead of having one objective function or having weights or constraints to describe these parameters.

The variables used in the optimization are k_p and k_i. The time allowed for the plant simulation is 4 seconds. This indicates that for every k_p and k_i value generated the plant model is simulated for 4 seconds in order to observe the performance of that particular gain values. The detailed objective function is shown in equation (3.1). The optimization solver generates random initial gain values and identifies the performance of this gain values by performing simulation and ranks it accordingly. This allows re-usability, scalability, and robustness of the optimization algorithm.

It should be clarified that although a direct comparison between the proposed tuning results with previous works, especially with model simplification, is preferred, it is difficult to do so because many design details are not fully reported in the previous works and new model simplification must be carried out for the studied system. Thus, such comparison may be biased.

Objective function formulation for the photovoltaic model

The feedback signal used for the control methodology is the phase voltages of phases A, B, and C. The three voltages are summed together to include the effect of PI control on the three phases. Thus, the set point reference is 624. Normally, this will look similar to Figure 2.1. The multi-objective formulation for the gain tuning is shown in equation (3.2).

Objective function formulation for the single objective optimization

The objective function used in the single objective optimization is shown in eqn. 3.3. For the models 1-4 and the RTDS based transfer function plant model the step response is set to be at 1. For the PV based model, the expected step response is 624.

$$\min \left(\quad risetime(u) + settlingtime(u) + overshoot(u) + |peak(u) - expected\ peak\ value| \ \right)$$

$$Where\ u\ is\ the\ output\ shown\ in\ Fig.\ 2.1 \tag{3.3}$$

3.8 Summary

This chapter presented introduction to single objective optimization and multi-objective optimization. This chapter also presented an introduction to GA and simulated annealing which will be used to solve the optimization problems (both single and multiple). In order to speed up the process, parallel computing based tuning is proposed for the plants modelled in Matlab/Simulink. An RTDS based tuning is proposed to reduce the time needed for controller hardware-in-loop simulation based gain tuning. GA and simulated annealing based optimization methodologies are used to solve the plants modelled in real time digital simulator. This chapter presents models that will be used in the optimization tuning. The results of the gain tuning optimization is presented along with the comparison between different optimization methodologies and different plant models are shown in the upcoming chapter. Parallel computing and use of real time digital simulator are proposed for simulating the plant models.

Chapter 4

Results from software based gain tuning

This chapter presents results from the models that were simulated entirely in software platform. The results of the gain tuning optimization is presented along with the comparison between different optimization methodologies and different plant models. Parallel computing and use of real time digital simulator are proposed for simulating the plant models. The results from simulation of the plants modelled in simulink based modelling and plants modelled in real time digital simulator are shown in this chapter.

4.1 Comparison of single objective and multiple objective optimization results

Plant models 1-4

Fig. 4.1 and 4.2 indicate the results obtained for the single objective optimization with the use of genetic algorithm and simulated annealing. For the relatively simpler model 1, model 2, and model 3, single objective genetic algorithm and single objective simulated annealing present gain values which have reasonable rise time, overshoot, and settling time. But, for the time consuming model 4, only the optimization performed with eight workers provided good result. In single objective simulated annealing based approach, for the same objective function, models 1-3 performed well. But, model 4 had oscillatory response and the graph indicate that the system might not settle down to the expected set point value.

Figures 4.3, 4.5, 4.7, and 4.9 show the results for multi-objective optimization using multi-objective genetic algorithm. The plots show that there is variety of gain values that multi-objective genetic algorithm presents when compared to the single objective genetic algorithm and single objective simulated annealing. Focus should be presented to model 4 which is a harder model than the first three. The plots in 4.9 indicate that the multi-objective genetic algorithm presents superior gain values when compared to the single objective optimization.

PV based plant model

Similar results were obtained for the PV based plant model. While genetic algorithm was able to provide a good gain values, the gain values provided by simulated annealing was inferior. This could be observed in Figure 4.14.

RTDS based plant model and PI controller model

Even in the RTDS based simulation, the step response of the plant for gain values obtained from multi-objective genetic algorithm is superior when compared to the single objective genetic algorithm, and single objective simulated annealing.

A more deeper look into the results indicate that adding these different objectives from MO problem formulation does not provide a clear understanding of the needs of the engineer. Either the SO optimization algorithms gets stuck in a local optima (single objective simulated annealing) or it picks the gain value which provides the lowest sum of all the objectives. These traps are avoided in the MO based optimization, since the Pareto front plays a major role in identifying the optimal gain values.

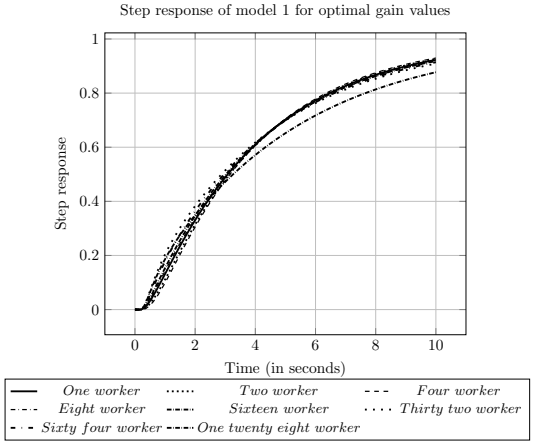
4.2 Function space comparison

Figures 4.4, 4.6, 4.17, 4.10, 4.16, and 4.19 provide the function spaces for the different objective functions used in the optimization. It could be observed there is no universal nature for these objective space in terms of conflicting or non-conflicting. This mainly depends on the nature of the plant model.

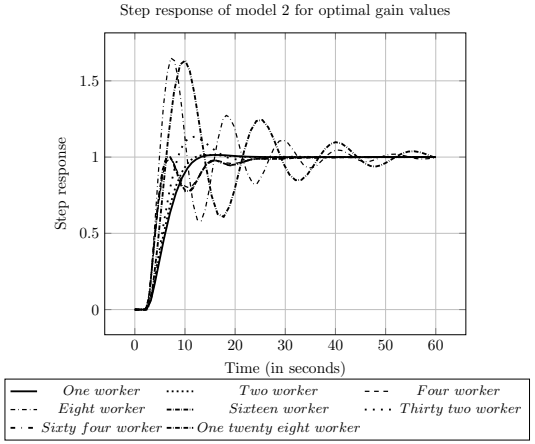
4.3 Speed up and efficiency for different process models

Speed up and efficiency proposed in equations 3.3, and 3.4 are used to understand the effectiveness of the paralleling the single objective genetic algorithm and multi-objective genetic algorithm calculations. The number of processors that could be used for parallel computing was 16.

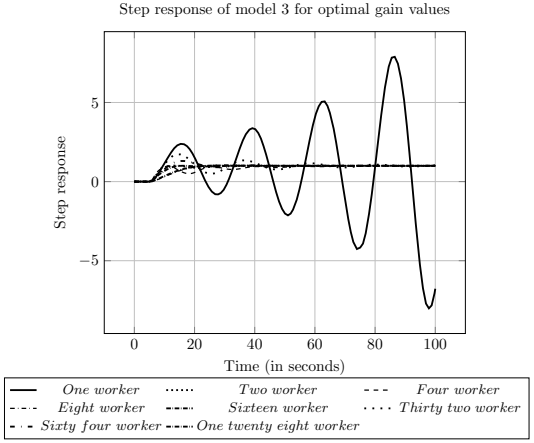
Computations more than 16 use virtual processors to model the increased number of processors. Due to this reason, the speedup reduces and attains a saturation after 16 workers. Due to the same reason, the efficiency decreases as the number of worker increases with saturated speed up. This could be observed for all the models used.



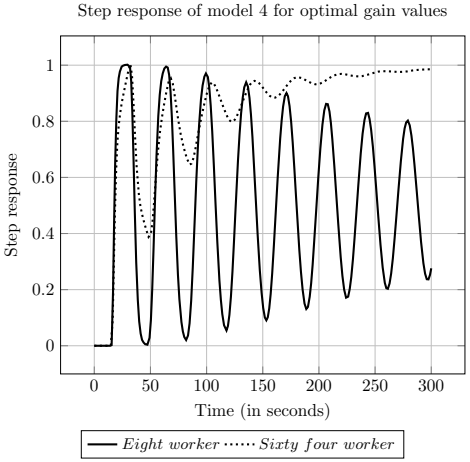
(a) Model 1 result



(b) Model 2 result

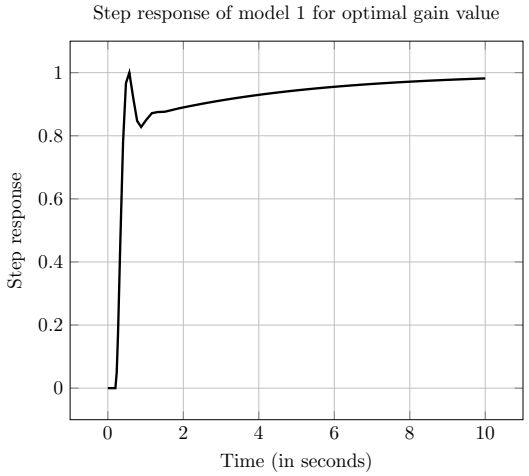


(c) Model 3 result

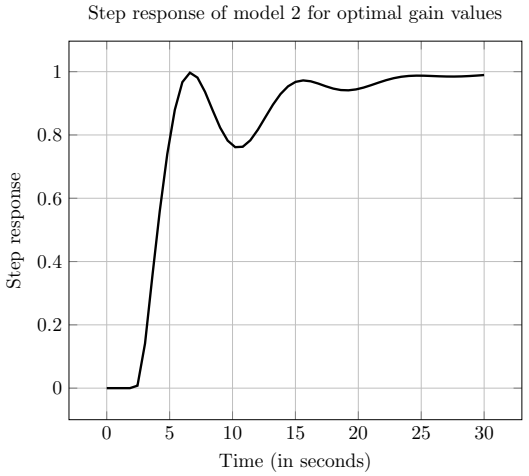


(d) Model 4 result

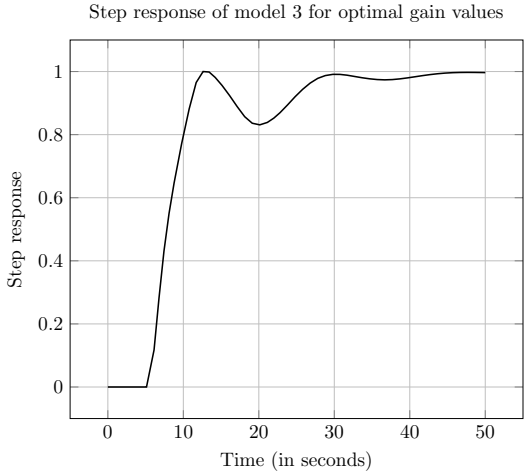
Figure 4.1: Results from genetic algorithm based single objective optimization of models 1-4



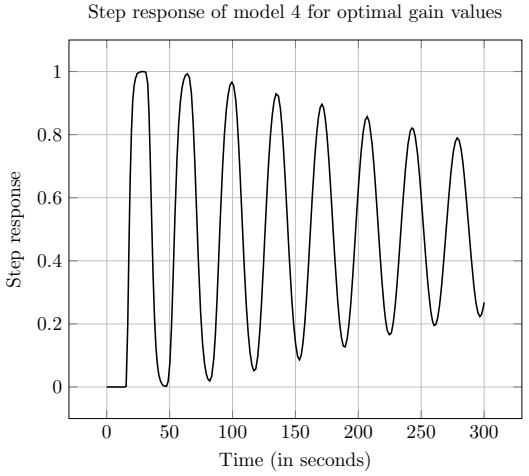
(a) Response to optimal gain value for model 1



(b) Response to optimal gain value for model 2

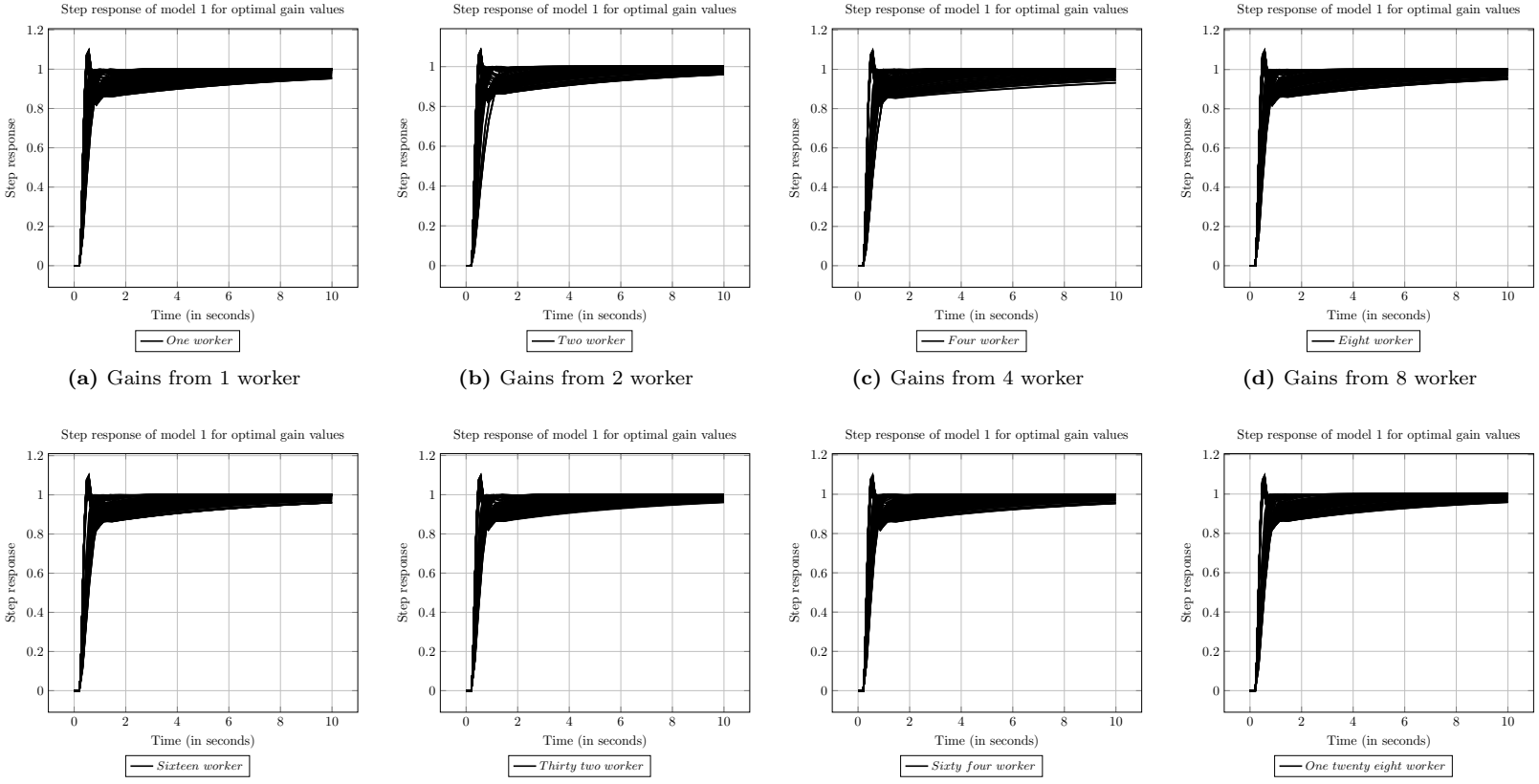


(c) Response to optimal gain value for model 3



(d) Response to optimal gain value for model 4

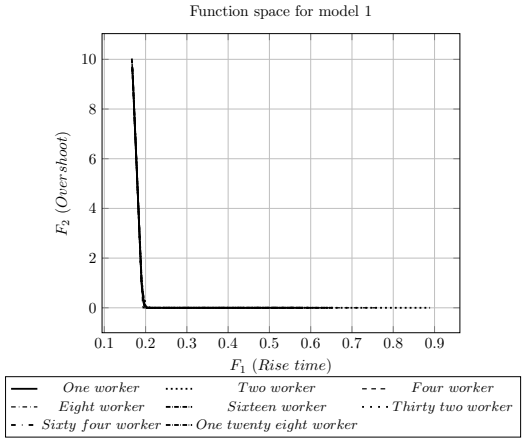
Figure 4.2: Results from Simulated annealing based single objective optimization of models 1-4



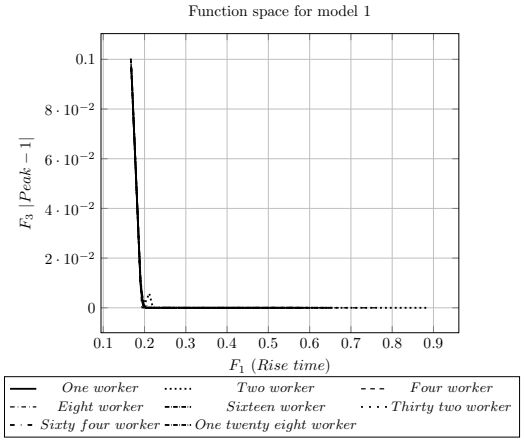
(a) Gains from 1 worker **(b)** Gains from 2 worker **(c)** Gains from 4 worker **(d)** Gains from 8 worker

(e) Gains from 16 worker **(f)** Gains from 32 worker **(g)** Gains from 64 worker **(h)** Gains from 128 worker

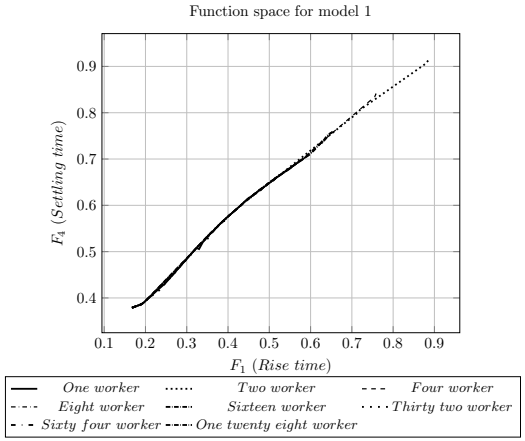
Figure 4.3: Results from genetic algorithm based multi-objective optimization - model 1



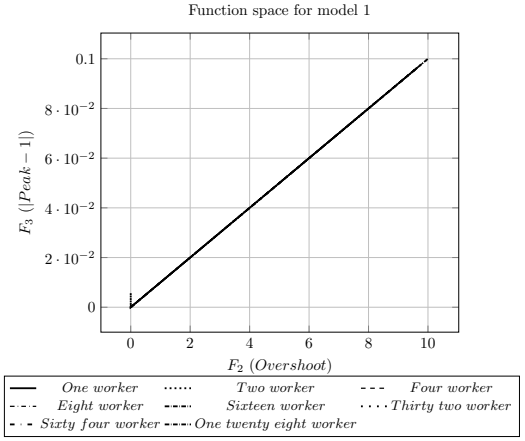
(a) Rise time vs Overshoot



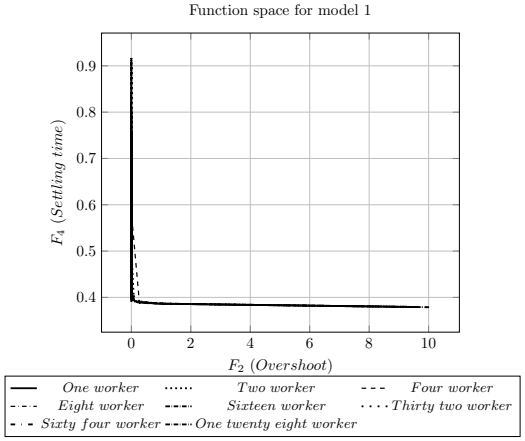
(b) Rise time vs $|Peak\ -\ 1|$



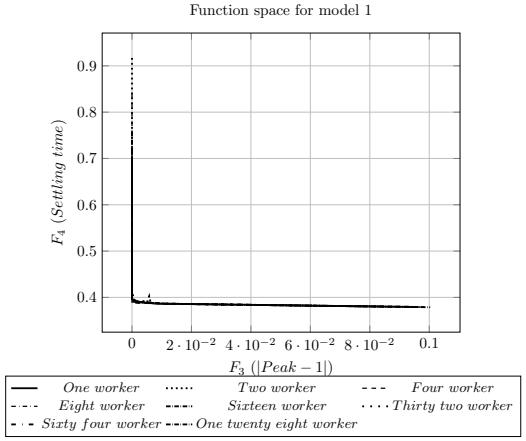
(c) Rise time vs settling time



(d) Overshoot vs $|Peak\ -\ 1|$

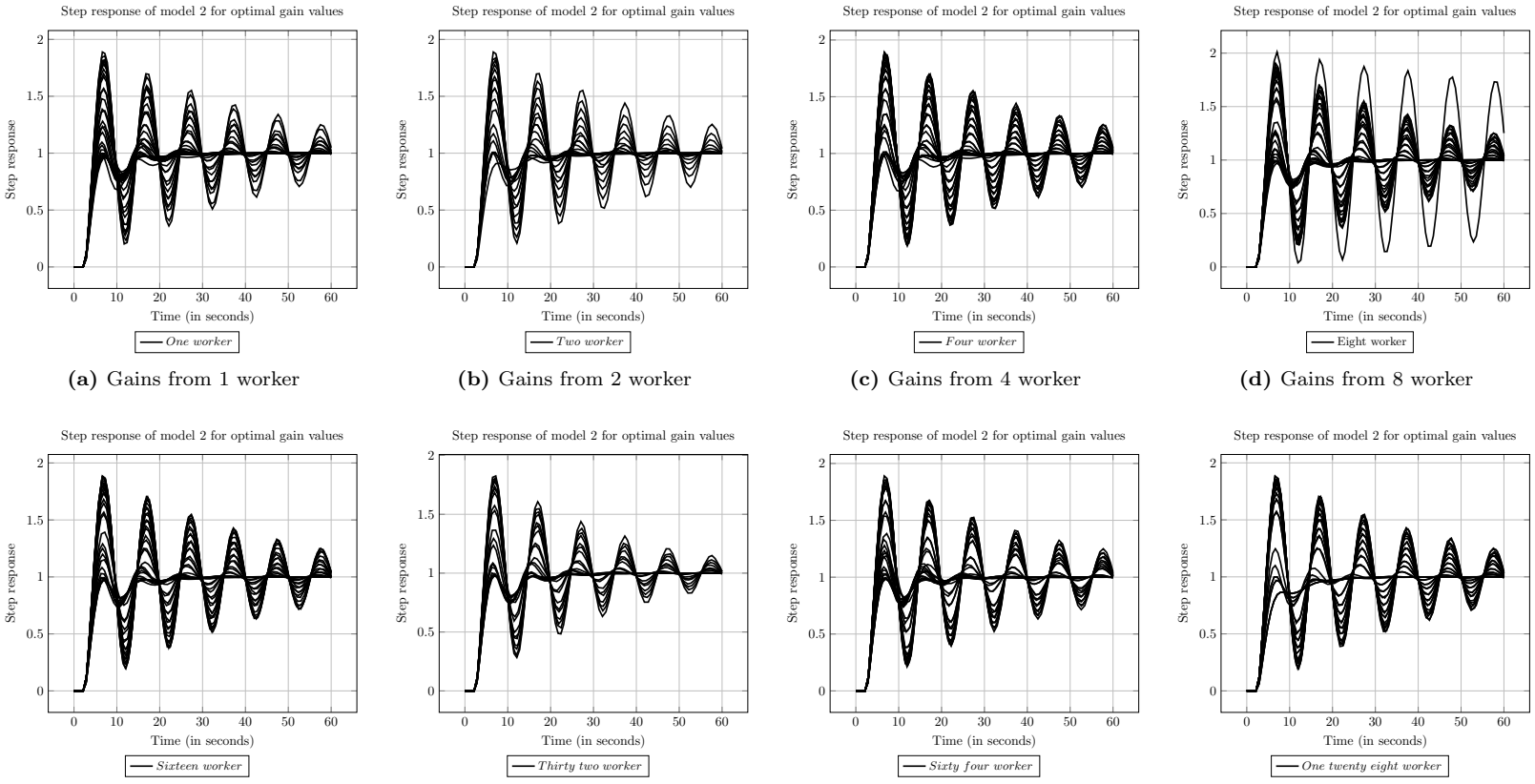


(e) Overshoot vs settling time



(f) $|Peak\ -\ 1|$ vs settling time

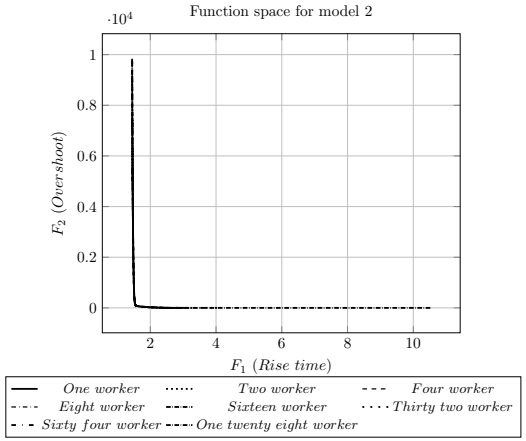
Figure 4.4: Function spaces for model 1



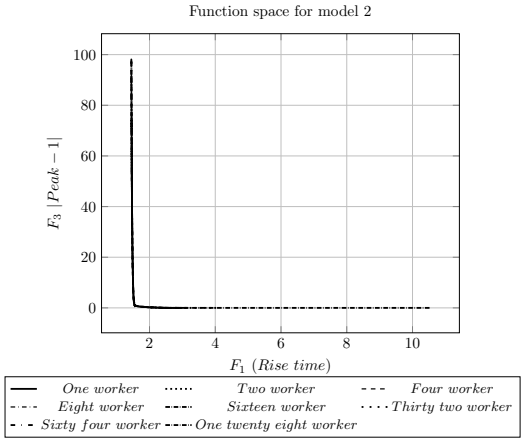
(a) Gains from 1 worker **(b)** Gains from 2 worker **(c)** Gains from 4 worker **(d)** Gains from 8 worker

(e) Gains from 16 worker **(f)** Gains from 32 worker **(g)** Gains from 64 worker **(h)** Gains from 128 worker

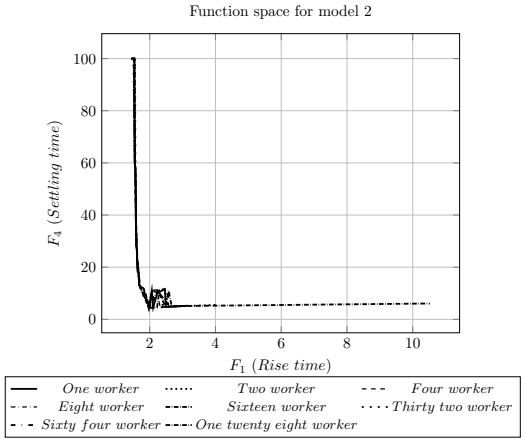
Figure 4.5: Results from genetic algorithm based multi-objective optimization - model 2



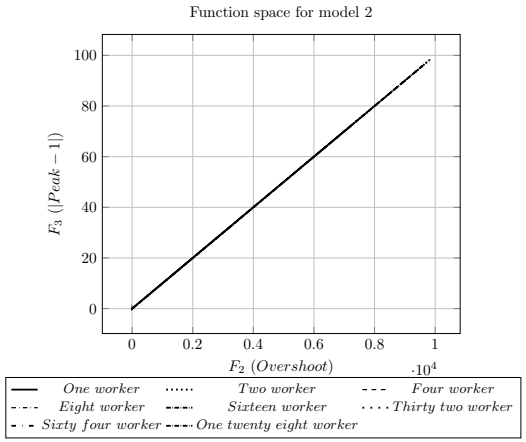
(a) Rise time vs Overshoot



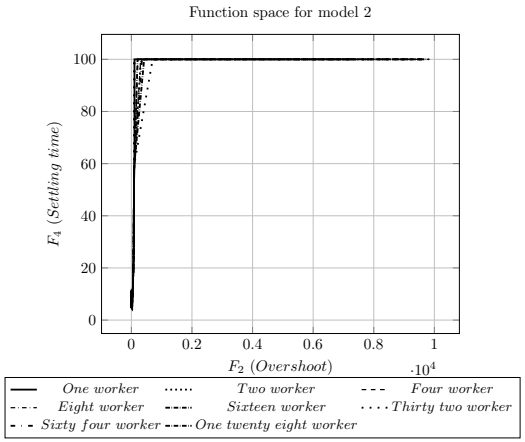
(b) Rise time vs $|Peak\ -\ 1|$



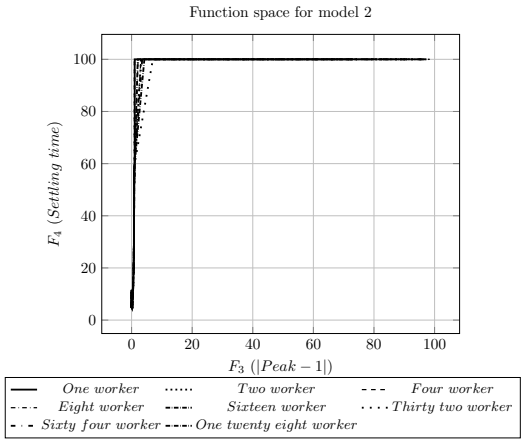
(c) Rise time vs settling time



(d) Overshoot vs $|Peak\ -\ 1|$

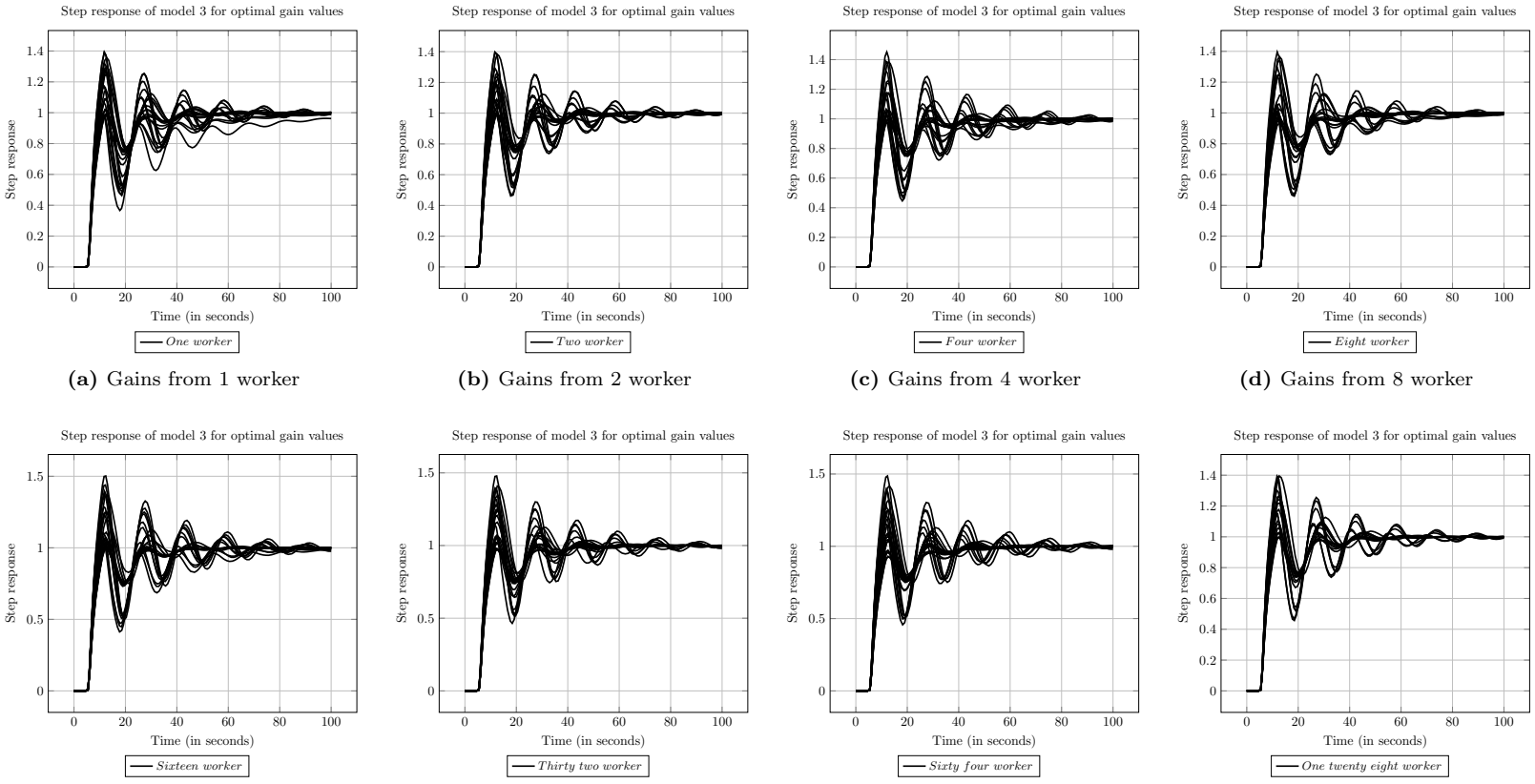


(e) Overshoot vs settling time



(f) $|Peak\ -\ 1|$ vs settling time

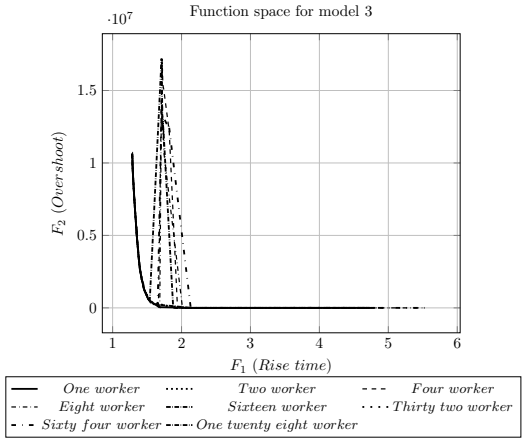
Figure 4.6: Function spaces for model 2



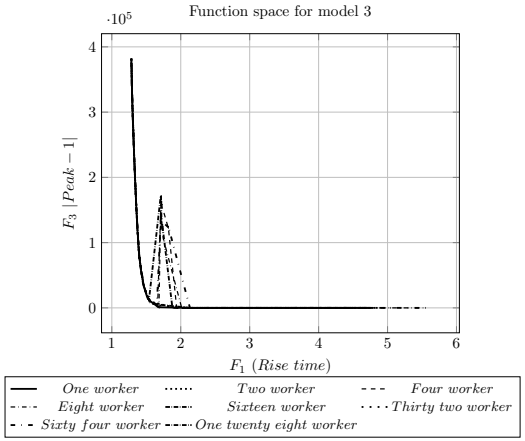
(a) Gains from 1 worker **(b)** Gains from 2 worker **(c)** Gains from 4 worker **(d)** Gains from 8 worker

(e) Gains from 16 worker **(f)** Gains from 32 worker **(g)** Gains from 64 worker **(h)** Gains from 128 worker

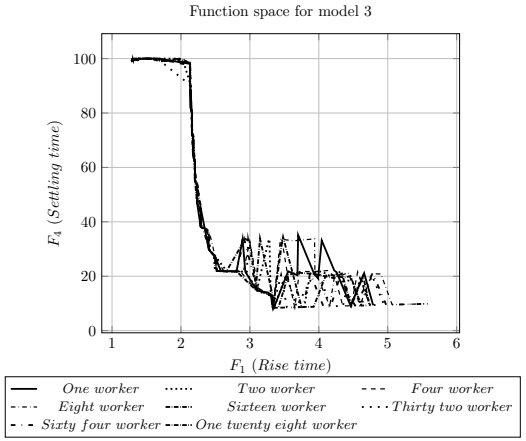
Figure 4.7: Results from genetic algorithm based multi-objective optimization - model 3



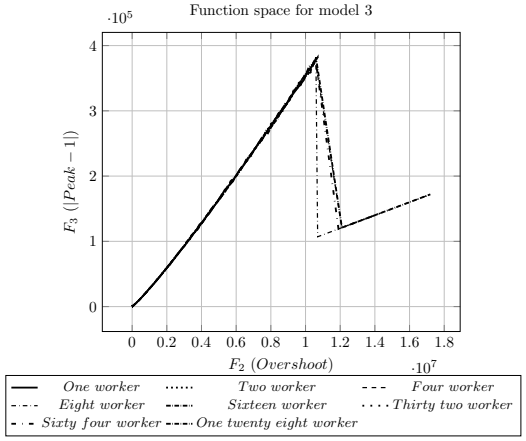
(a) Rise time vs Overshoot



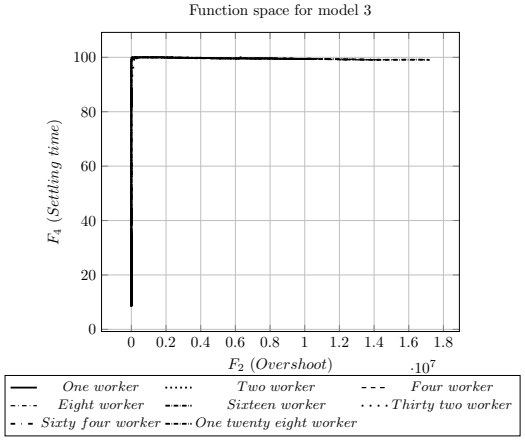
(b) Rise time vs $|Peak\ -\ 1|$



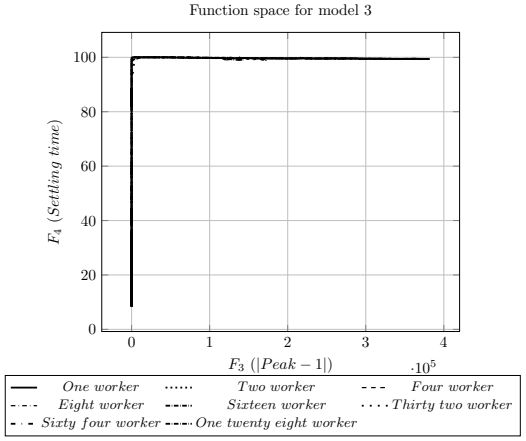
(c) Rise time vs settling time



(d) Overshoot vs $|Peak\ -\ 1|$

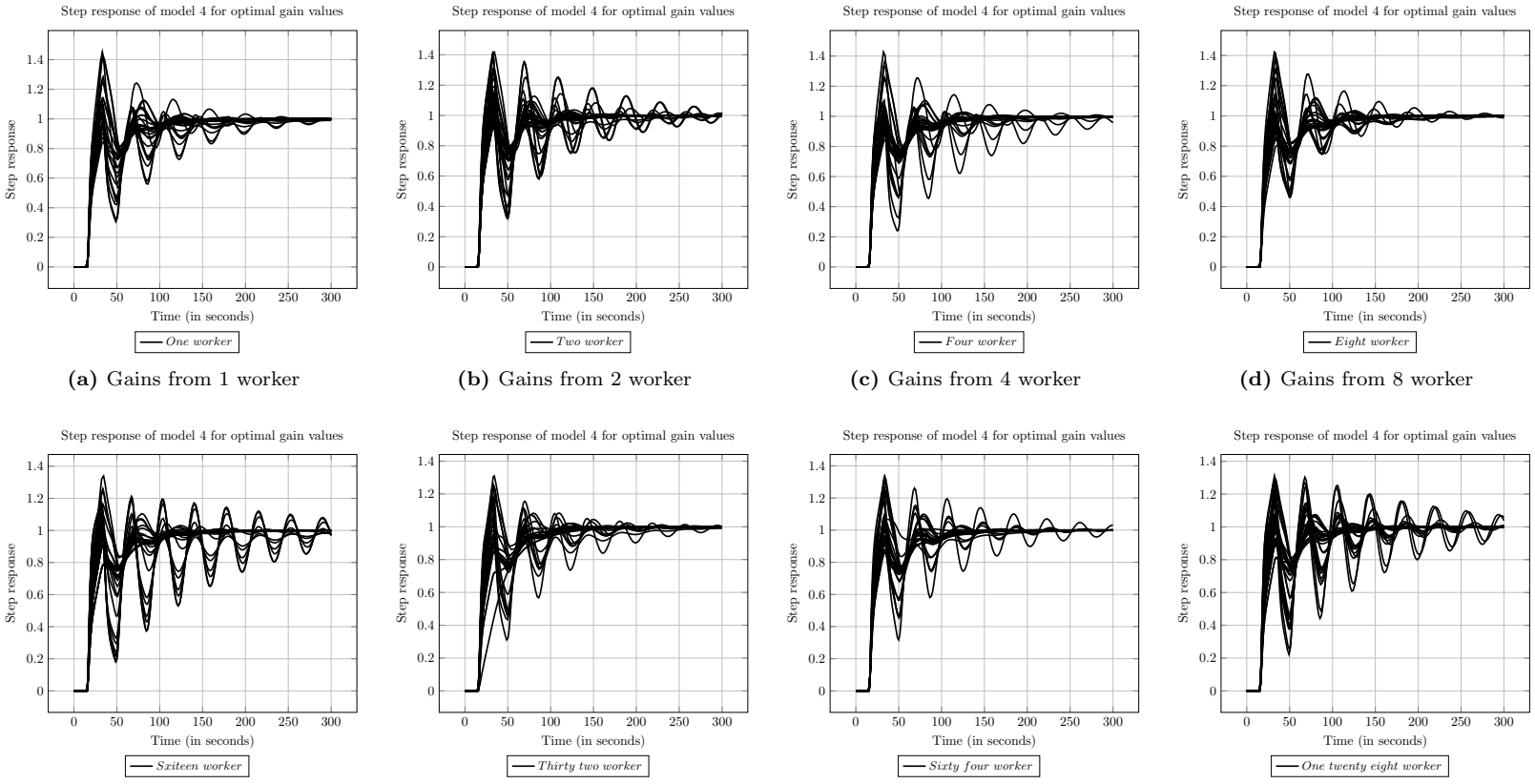


(e) Overshoot vs settling time



(f) $|Peak\ -\ 1|$ vs settling time

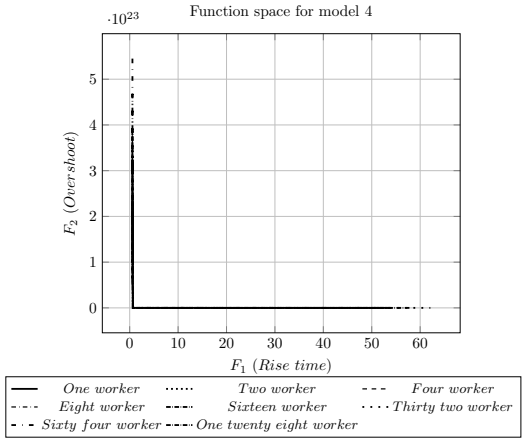
Figure 4.8: Function spaces for model 3



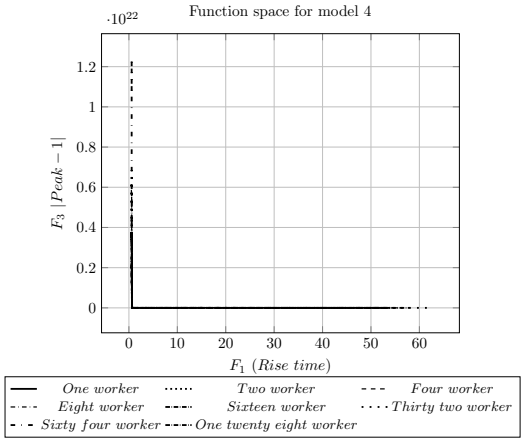
(a) Gains from 1 worker **(b)** Gains from 2 worker **(c)** Gains from 4 worker **(d)** Gains from 8 worker

(e) Gains from 16 worker **(f)** Gains from 32 worker **(g)** Gains from 64 worker **(h)** Gains from 128 worker

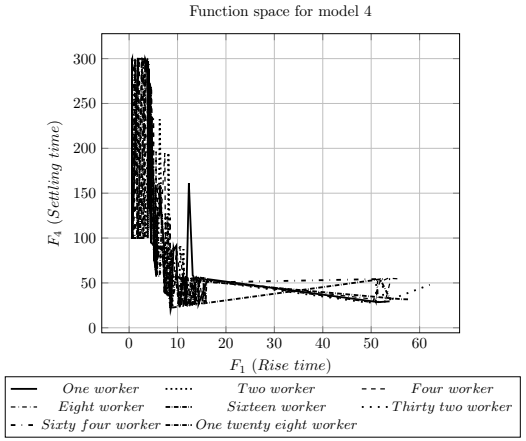
Figure 4.9: Results from genetic algorithm based multi-objective optimization - model 4



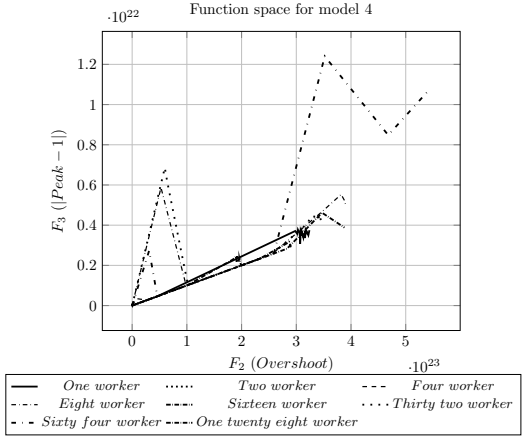
(a) Rise time vs Overshoot



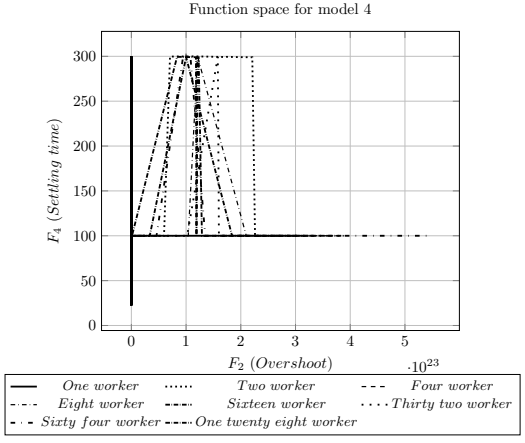
(b) Rise time vs $|Peak\ -\ 1|$



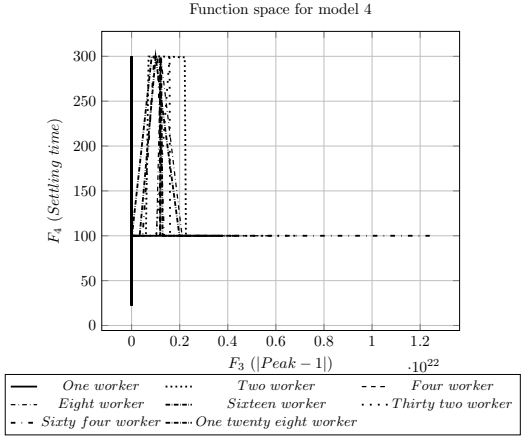
(c) Rise time vs settling time



(d) Overshoot vs $|Peak\ -\ 1|$

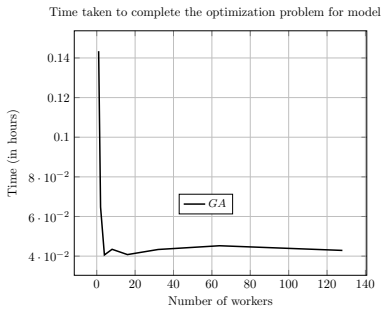


(e) Overshoot vs settling time

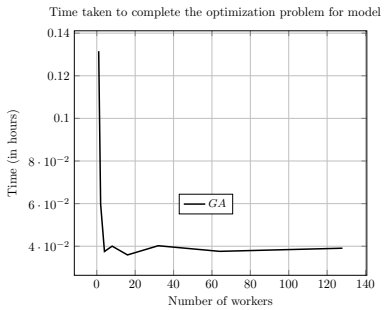


(f) $|Peak\ -\ 1|$ vs settling time

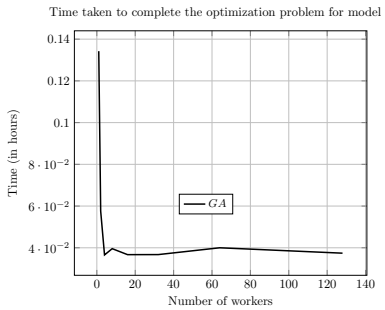
Figure 4.10: Function spaces for model 4



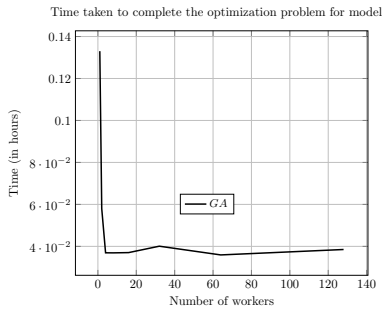
(a) Time for model 1 using GA



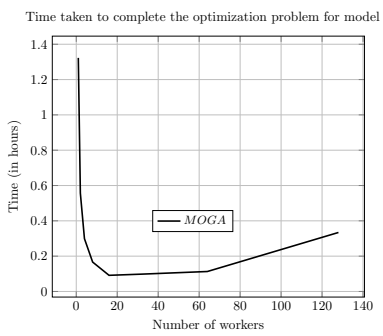
(b) Time for model 2 using GA



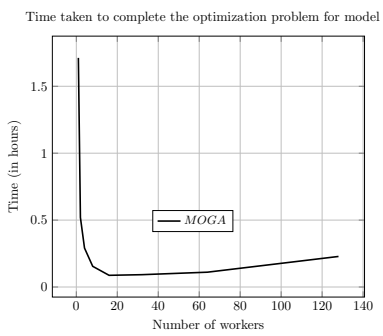
(c) Time for model 3 using GA



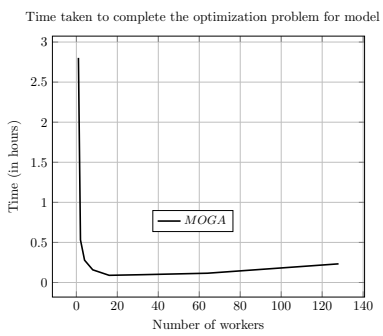
(d) Time for model 4 using GA



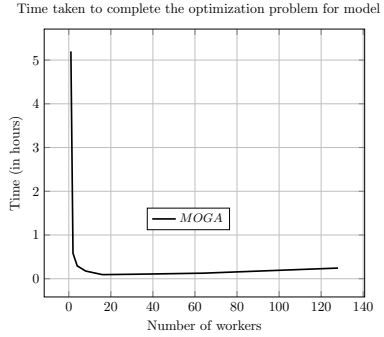
(e) Time for model 1 using MOGA



(f) Time for model 2 using MOGA

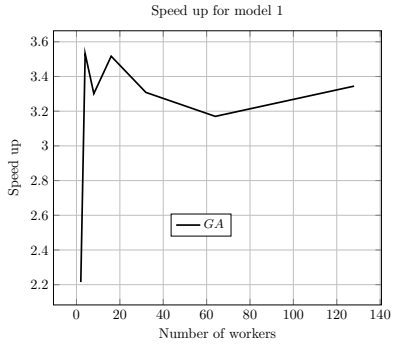


(g) Time for model 3 using MOGA

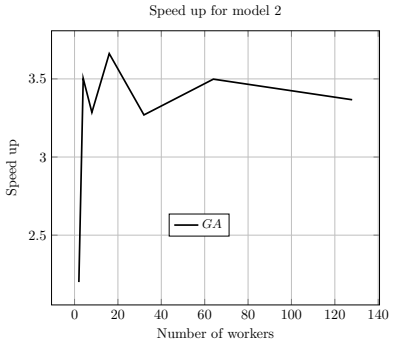


(h) Time for model 4 using MOGA

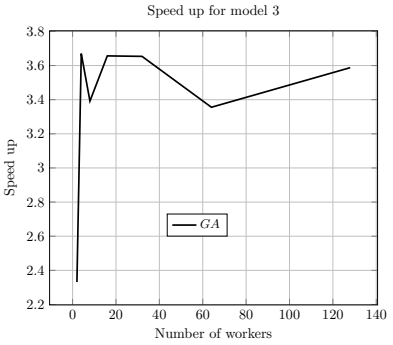
Figure 4.11: Time for gain tuning optimization of models 1-4 using GA and MOGA



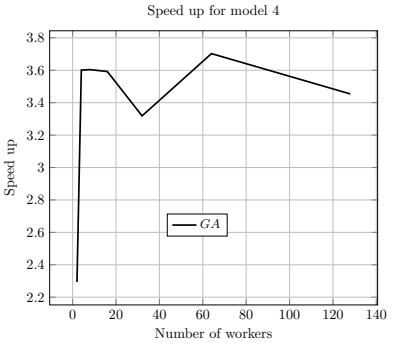
(a) Speed up for model 1 using GA



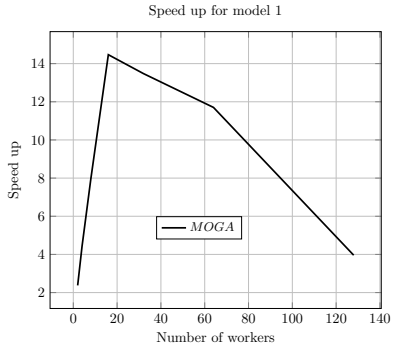
(b) Speed up for model 2 using GA



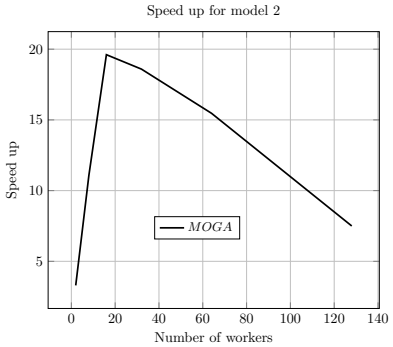
(c) Speed up for model 3 using GA



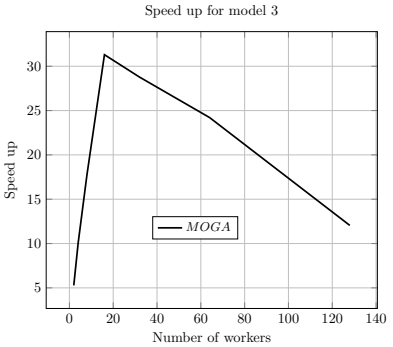
(d) Speed up for model 4 using GA



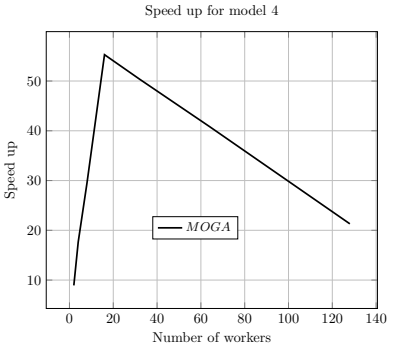
(e) Speed up for model 1 using MOGA



(f) Speed up for model 2 using MOGA

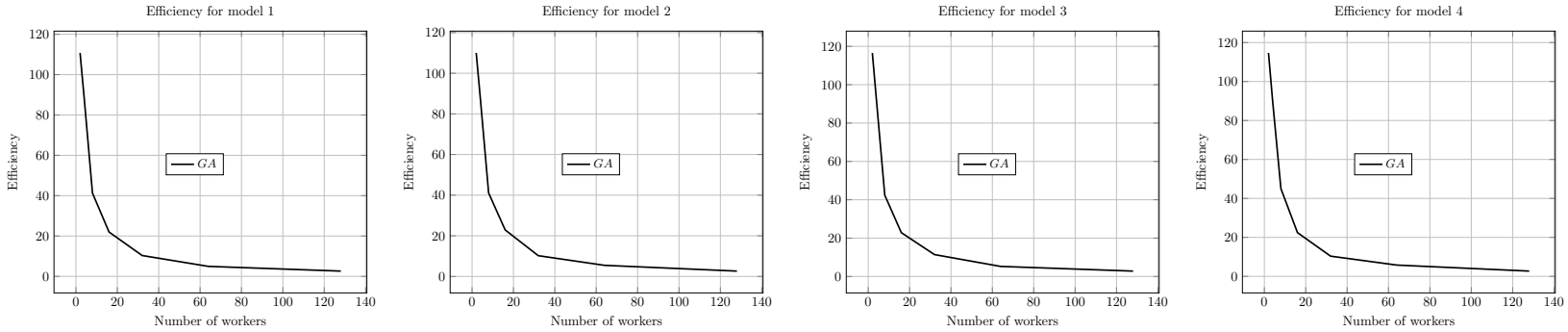


(g) Speed up for model 3 using MOGA

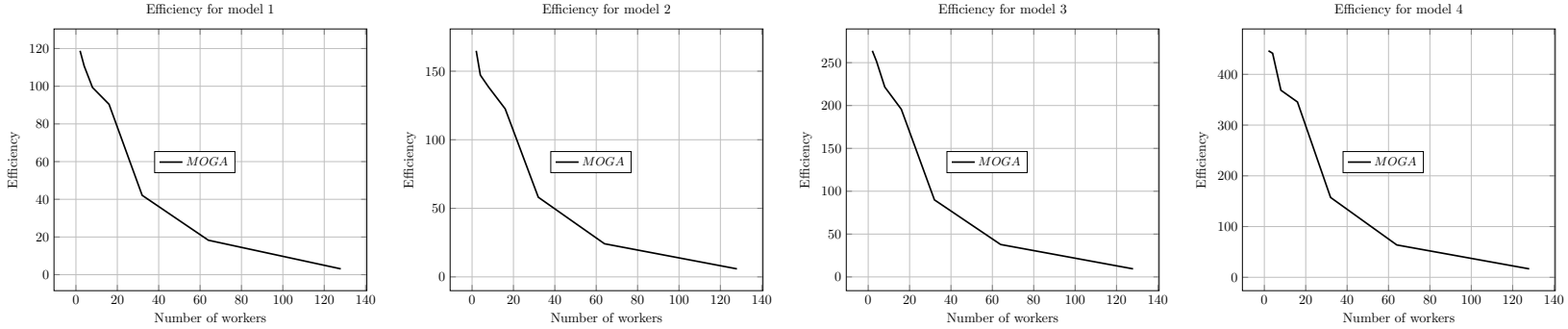


(h) Speed up for model 4 using MOGA

Figure 4.12: Speedup for the four different models using GA and MOGA



(a) Efficiency for model 1 when using GA **(b)** Efficiency for model 2 when using GA **(c)** Efficiency for model 3 when using GA **(d)** Efficiency for model 4 when using GA



(e) Efficiency for model 1 when using MOGA **(f)** Efficiency for model 2 when using MOGA **(g)** Efficiency for model 3 when using MOGA **(h)** Efficiency for model 4 when using MOGA

Figure 4.13: Efficiency for the four different models using GA and MOGA

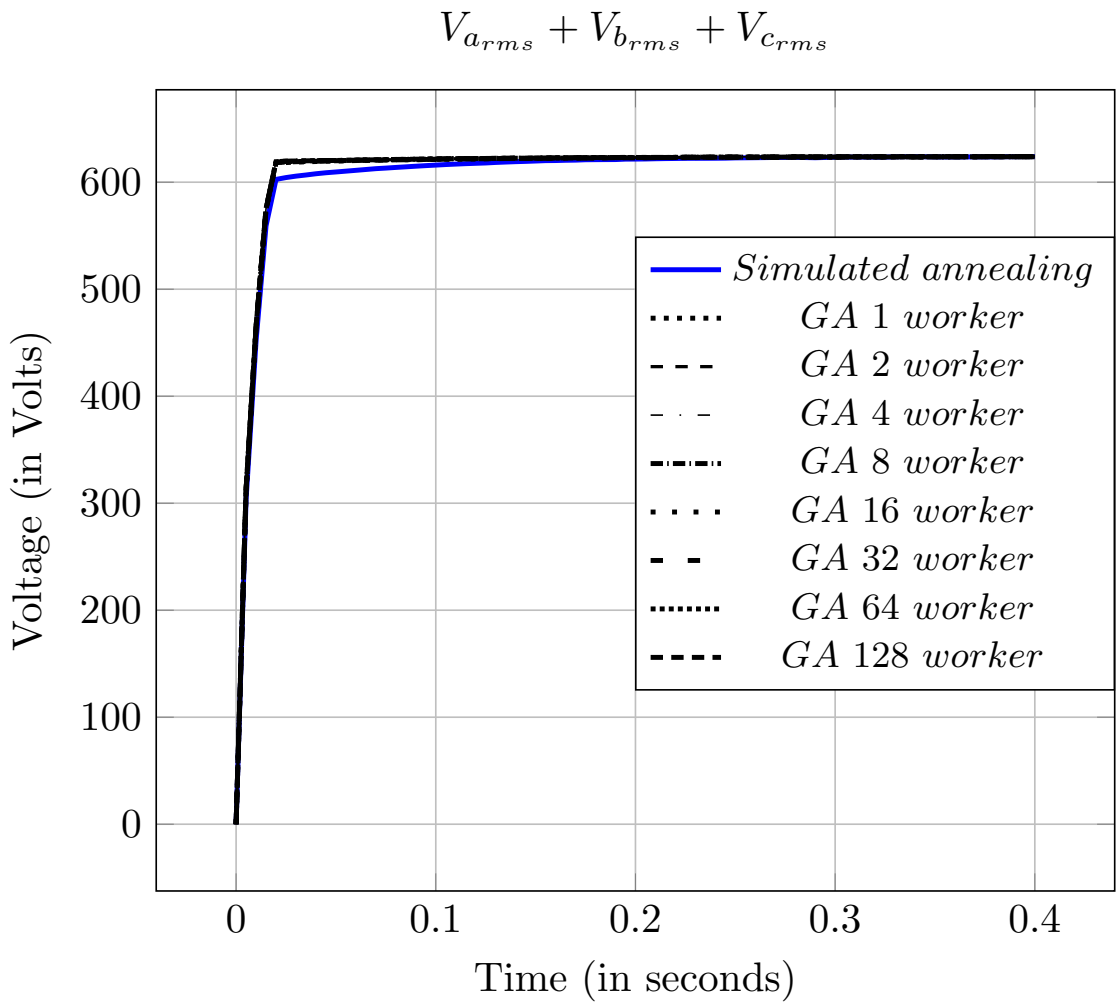
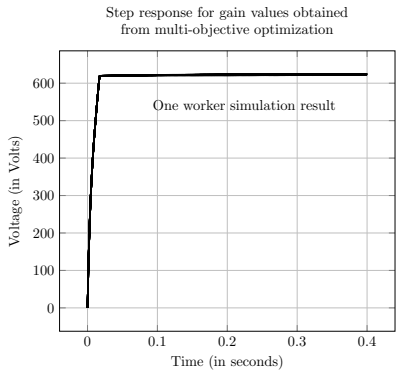
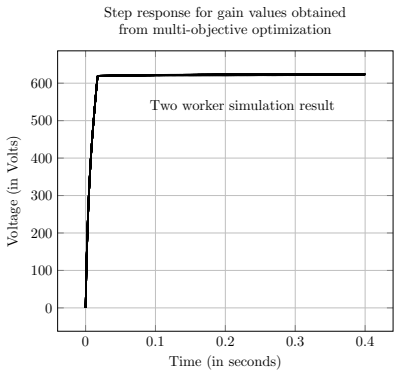


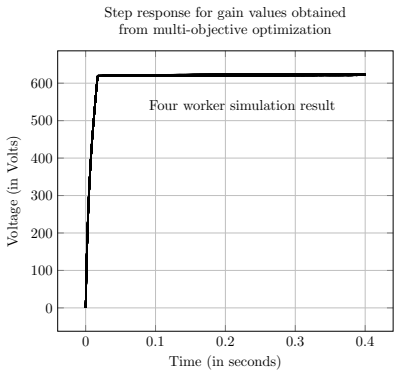
Figure 4.14: Response of the PV system to different PI gains obtained from single-objective optimization



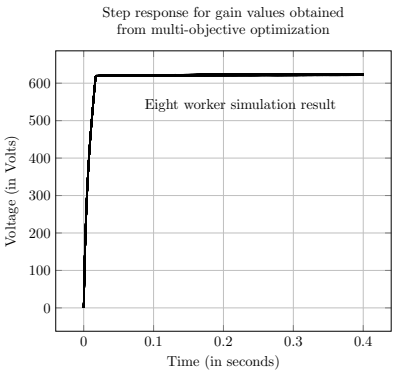
(a) 1 worker used for simulation



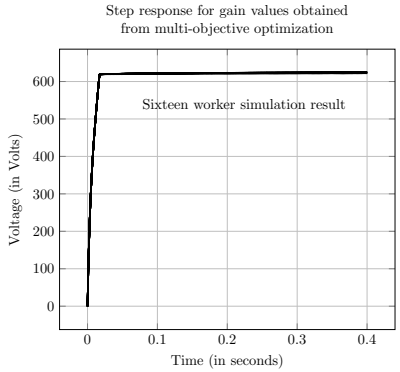
(b) 2 workers used for simulation



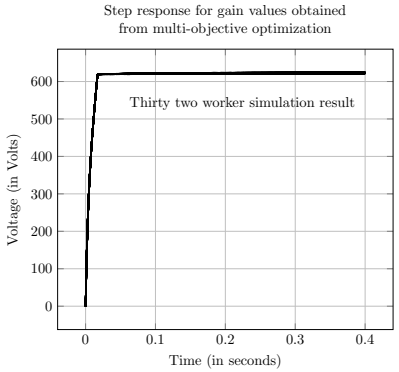
(c) 4 workers used for simulation



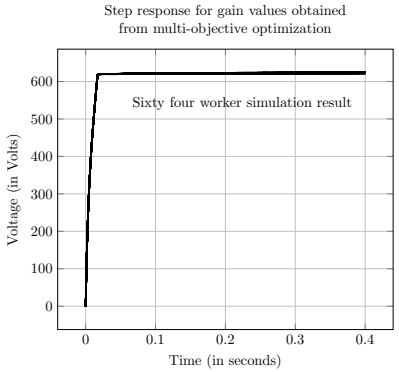
(d) 8 workers used for simulation



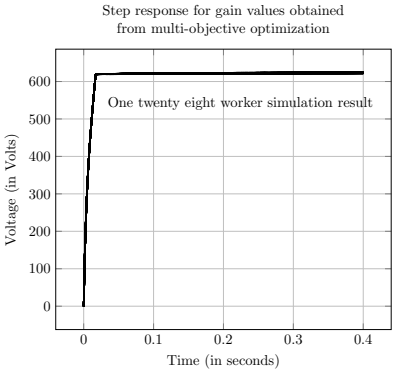
(e) 16 workers used for simulation



(f) 32 workers used for simulation

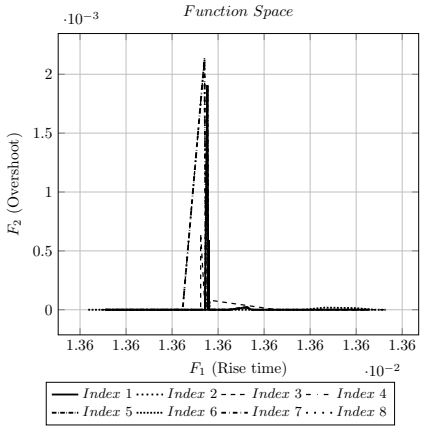


(g) 64 workers used for simulation

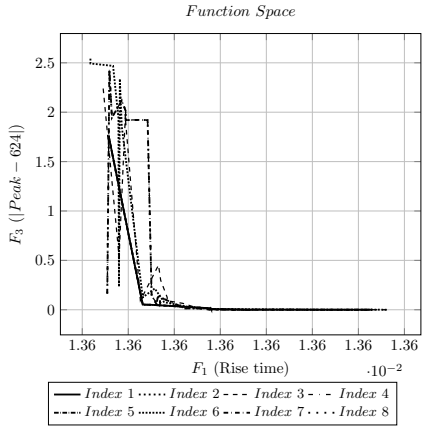


(h) 128 workers used for simulation

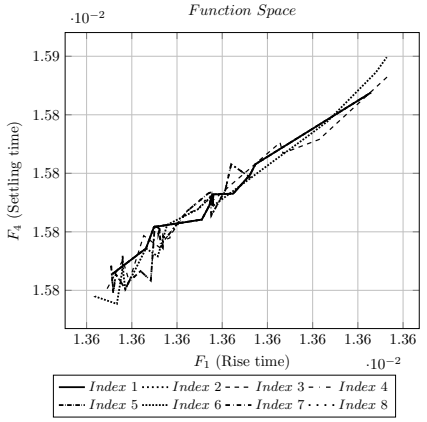
Figure 4.15: Response of the PV system to different PI gains obtained from multi-objective optimization



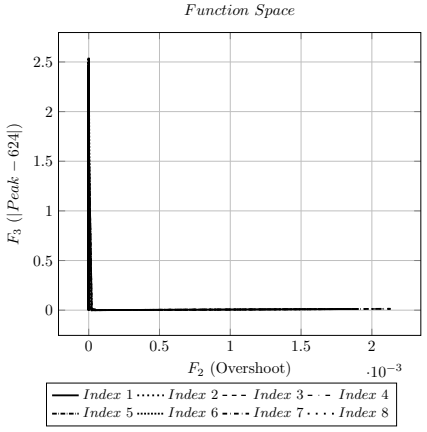
(a) Rise time vs Overshoot



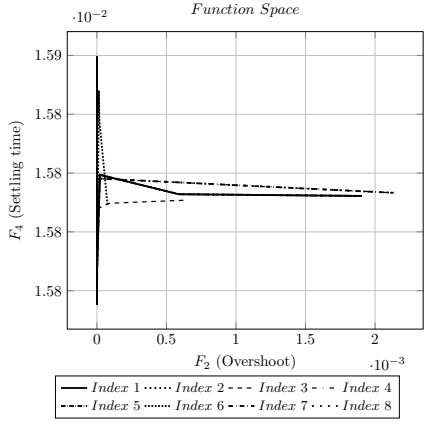
(b) Rise time vs $|Peak\ -\ 1|$



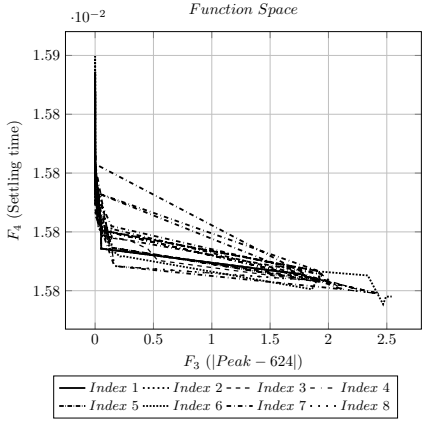
(c) Rise time vs settling time



(d) Overshoot vs $|Peak\ -\ 1|$

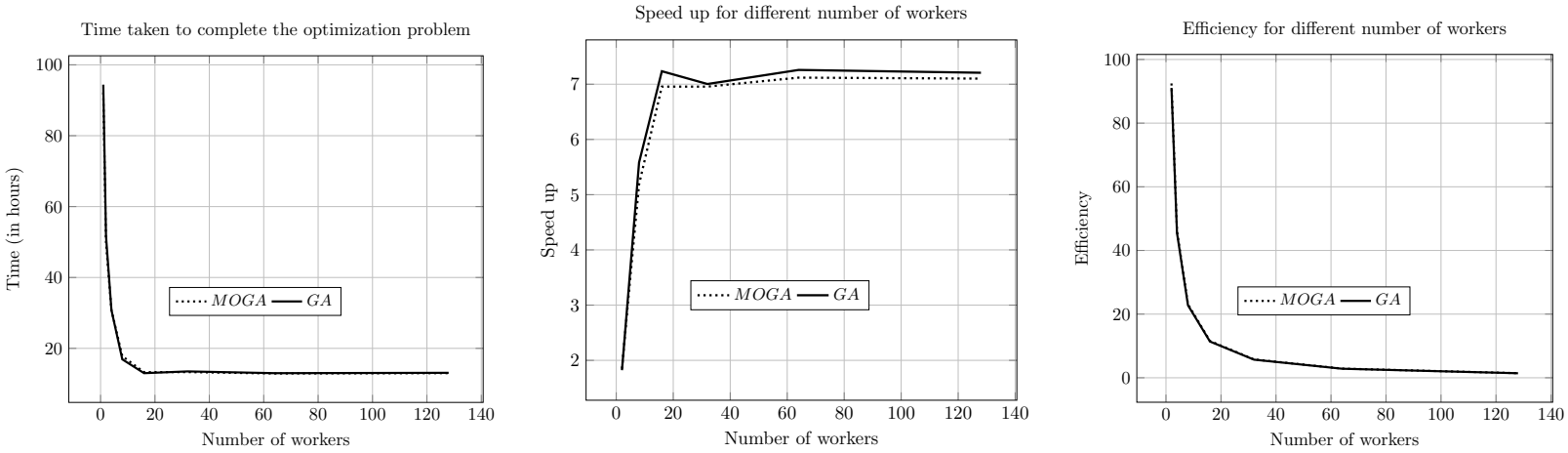


(e) Overshoot vs settling time



(f) $|Peak\ -\ 1|$ vs settling time

Figure 4.16: Function spaces for power system based model

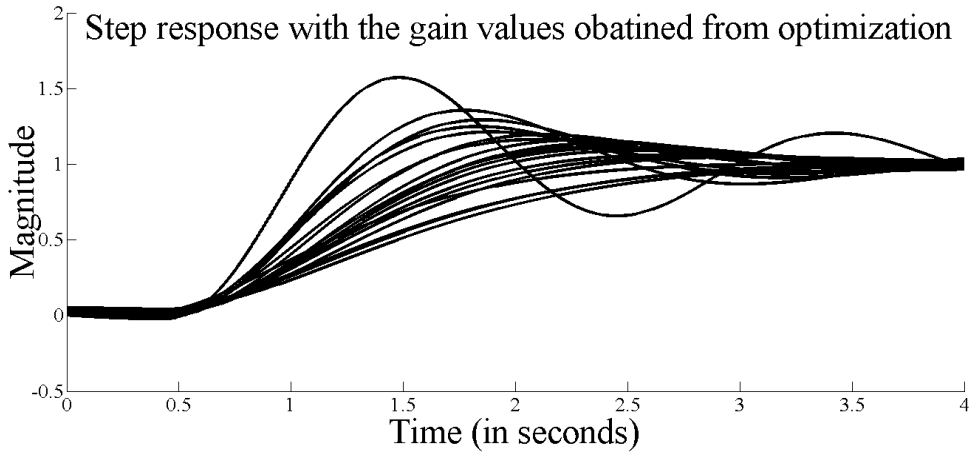


(a) Time used for optimization in power model

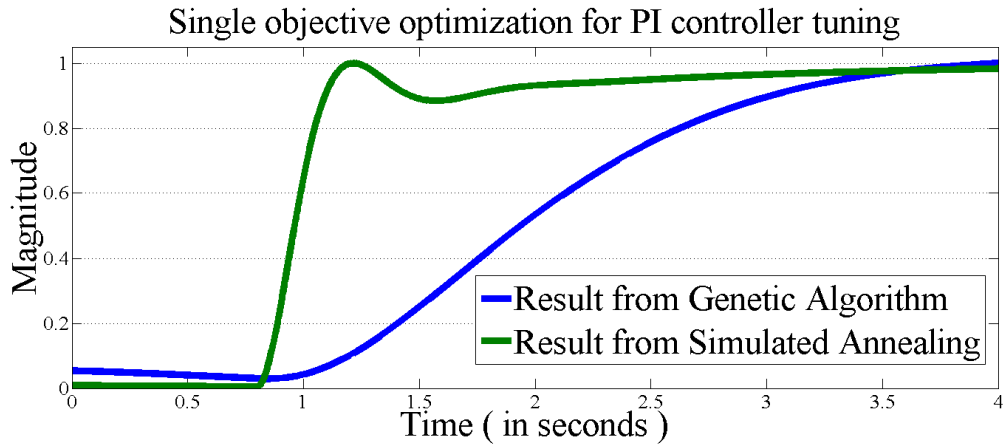
(b) Speed up of power model while using parallel computing

(c) Efficiency of power model while using parallel computing

Figure 4.17: Time, speed up and efficiency of power system model gain tuning

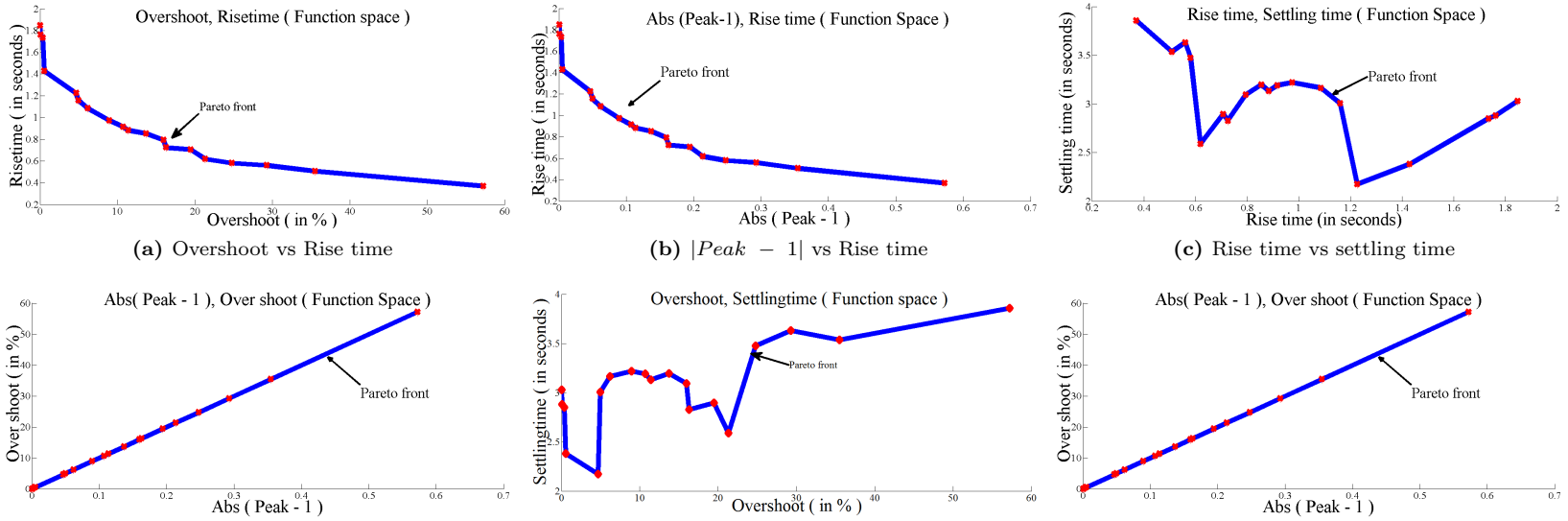


(a) Results from multi-objective optimization



(b) Results from single objective optimization

Figure 4.18: Results comparison between single objective optimization and multi-objective optimization using real time digital simulator



(a) Overshoot vs Rise time

(b) $|Peak\ -\ 1|$ vs Rise time

(c) Rise time vs settling time

(d) Overshoot vs $|Peak\ -\ 1|$

(e) Overshoot vs settling time

(f) settling time vs $|Peak\ -\ 1|$

Figure 4.19: Function spaces for real time digital simulator based plant model

4.4 Summary

The following can be summarized from the results.

- Multi-objective optimization is proposed here to tune the gain values of PI controller.
- This method has been used for tuning plants modelled in real time digital simulator and desktop based software tool.
- Parallel computing is proposed to make the optimization faster for desktop based software tool.
- A simple transfer function based plant is modelled in real time digital simulator along with the proportional integral controller. This model is used to tune the PI controller.
- Results indicate that the multi-objective optimization based tuning provided superior results when compared to single objective optimization.
- This has been indicated in both the desktop based software toolbox and the real time digital simulator based models.
- Function spaces provide a characteristic view of the plant models. These plots could be used to identify the conflicting and non-conflicting function spaces for the respective plant models.
- Speed up and the efficiency plots indicate that the use of parallel computing help in reducing the time used to perform the optimization.
- Use of real time digital simulator is proposed for controller hardware-in-loop based simulation. More details are shown in the next chapter.

Chapter 5

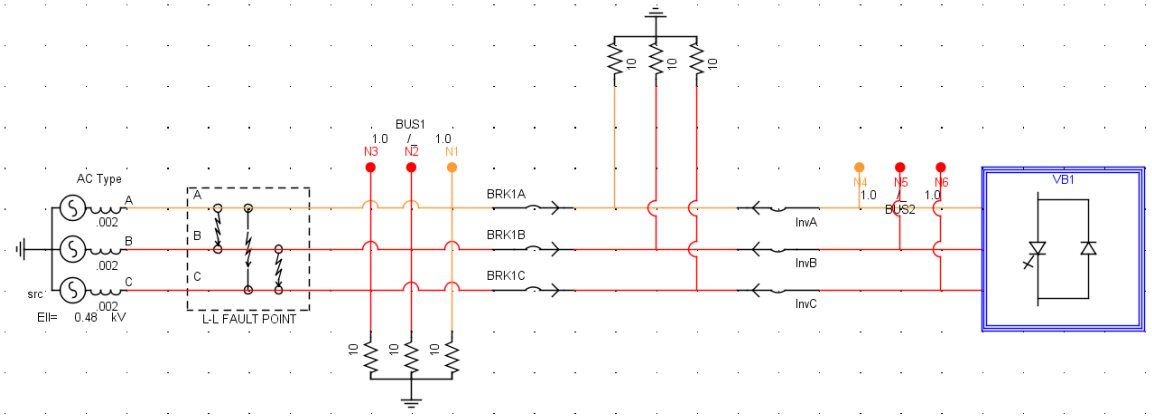
Models for controller hardware-in-loop based gain tuning

This chapter presents the controller hardware-in-loop setup for the gain tuning process. RTDS and FPGA based controller were utilized for the CHIL experiments. Initially, a simple power system model is used as the process/plant model and is modelled in the RTDS. Then, a detailed microgrid model with two inverters is used for the PI tuning. The PI controller is modelled in the FPGA based device. The details of the control algorithm used in the control architecture inside the FPGA is presented.

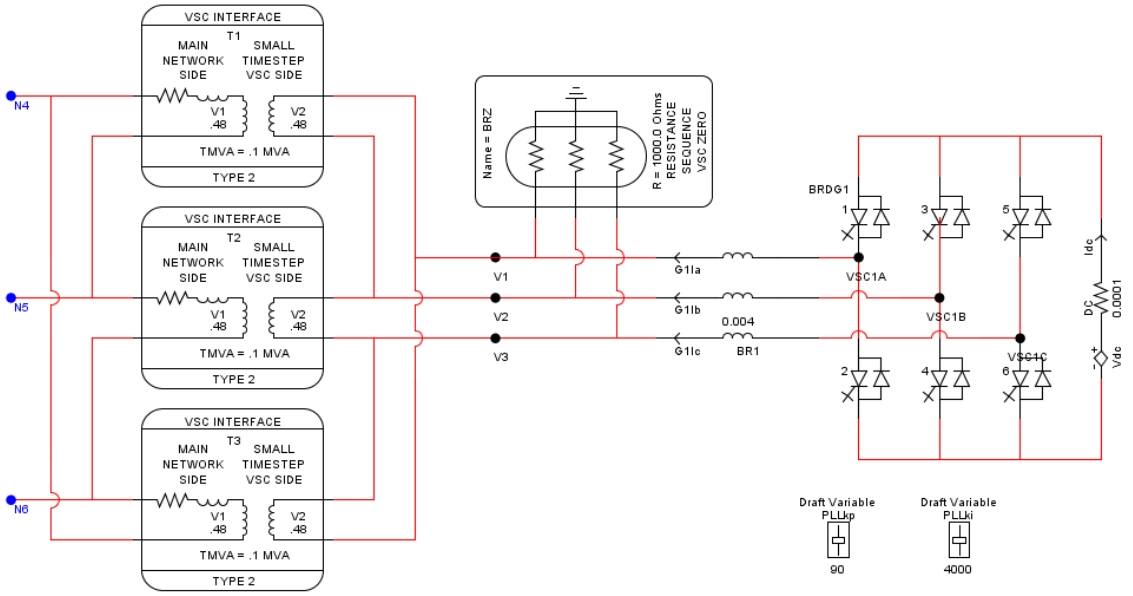
5.1 Sample Microgrid inverter model with current and voltage control

A step by step approach is presented here to provide an understanding of current control tuning and voltage control tuning. Inverters operating in microgrid conventionally has two control modes programmed in the inverter controller depending on the requirements of the utility.

- On-grid mode operation - During this mode, voltage and frequency are controlled by the main grid. The inverter delivers a constant PQ requested by the distributed/microgrid energy management system. During this mode, PI based current control is utilized. Functioning of this control used during on-grid mode is presented in the upcoming section.
- Island mode operation - During this mode, the microgrid is disconnected from the main grid. In the absence of grid, voltage and frequency of the microgrid should be maintained by the



(a) Simple power system model in RSCAD



(b) Power electronics model in small time step simulation in RSCAD

Figure 5.1: Real time digital simulator based power system model for gain tuning

inverters in the microgrid. In this mode of operation, generally one of the reliable inverters will be employed to maintain the voltage and frequency. In some cases, inverters share the load to provide real and reactive power support. The inverter owning utility will govern the inverter power sharing in islanded mode.

5.1.1 Real time digital simulator based microgrid model

A sample microgrid system is modelled in the RTDS. The model utilizes a simple three phase power supply, load, inverter model and a circuit breaker which acts as the microgrid switch. The microgrid

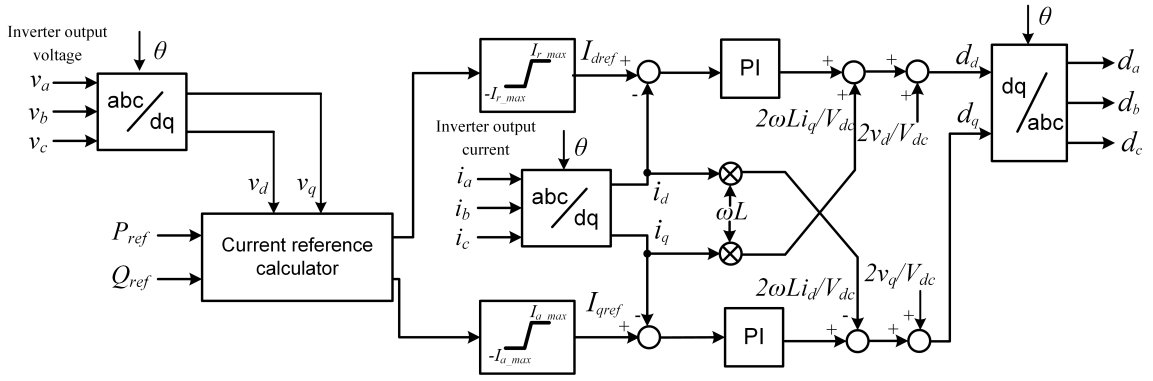
switch islands and synchronizes the microgrid back to the main grid. The inverter is connected to the grid through the microgrid circuit breaker. A small load is connected to the microgrid side of the power system. The RSCAD model of the system is shown in Figure. 5.1. The operating voltage level is 480 volts. A resistive load of 10 ohms is connected to the system to test the multi-objective optimization based gain tuning. The inverter model used in the HIL simulation has real power capability of 60 kW and reactive power capability of 60 kVAR. The DC link voltage for this inverter is 1000 volts and the AC voltage is 480 volts line to line RMS. This inverter model is used in the remainder of the HIL experiments. In RTDS, the inverter switching model is designed inside the small time step block. Because of this, the inverter model and the power system model run in two different time steps. This enables the inverter to be switched at higher frequencies than the normal time step. Signals will be moved between these models in the two time steps. This is handled by the small time step transformer. This data transfer enables the inverters to be switched at higher frequencies than the normal time step in RTDS. This is shown in Fig. 5.1 (a-b).

5.1.2 Inverter control description

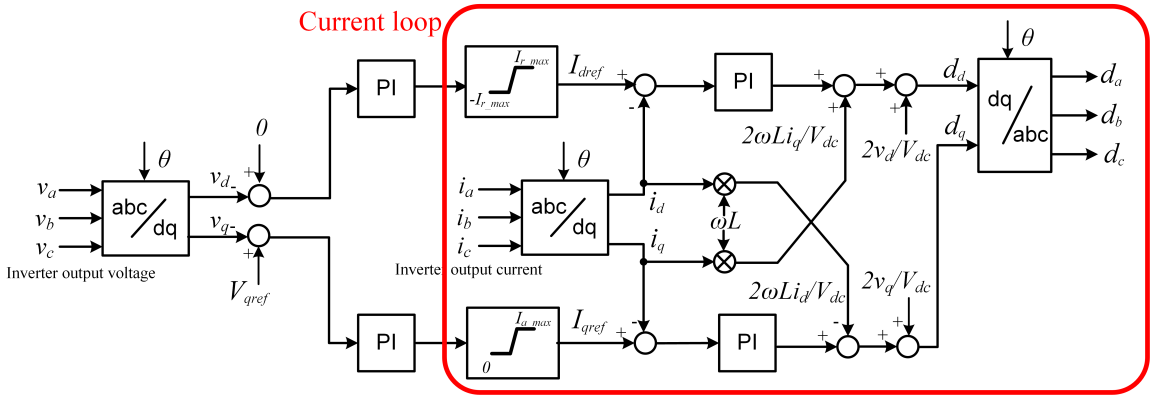
The essential microgrid inverter functions including grid-connected and islanding operation, transition between modes of operation, and stable voltage-frequency regulation in islanding mode, will all be performed inside the FPGA based controller. There are two control modes for inverter-based DERs presented here. They are active-reactive (PQ) power control, and voltage-frequency (VF) control. Fig. 5.2 (a) presents the control diagram of active and reactive power control. Open power loop is applied to calculate the active and reactive current reference. The inverter output currents in *abc* coordinates are converted to *dq* coordinates, and regulated through PI controllers to generate the necessary modulation signals. Voltage feedforward control is added to reduce the inrush current. The control diagram shown in Fig. 5.2 (a) will be utilized during the grid connected mode and the control diagram shown in Fig. 5.2 (b) will be utilized during the islanded mode. During islanded mode, the inverter should maintain voltage and frequency in the microgrid. The current reference for this controller loop will be calculated based on the voltage feedback. The gain constant for these two controls will be tuned using the proposed multi-objective optimization based algorithm. The objective functions used for gain tuning are based on the step response of the microgrid inverter plant model.

5.1.3 Experimental setup

The experimental setup is divided into two major parts. They are: *a*) the hardware component, and *b*) the software component . The hardware component of this experimental setup is shown in



(a) Active-reactive power control diagram



(b) Voltage-frequency control diagram

Figure 5.2: Controller modelled in the FPGA based device

Fig. 5.3. There are three major hardware component in the experiment. First, the real time digital simulator in which the power system models will be simulated. Second, the FPGA based control device. This device reads the signals from RTDS analog output, and generates modulation signals. These modulation signals will be sent into the RTDS through the analog input ports. And finally, the hardware component is the desktop computer that runs the multi-objective optimization. From a hierarchical perspective, the desktop computer that runs the optimization is the master. RTDS and the FPGA devices are slaves to the desktop computer and are peers with respect to each other.

The software component in the experiment complements these hardware components. The master desktop performing the optimization is programmed in Matlab. The two slaves, RTDS and the FPGA controller are programmed using RSCAD and Labview respectively. This is shown in Fig. 5.4 (a).

Desktop based optimization algorithm transmits and receives information from the RTDS through TCP/IP and communicates to the FPGA based hardware through UDP communication.

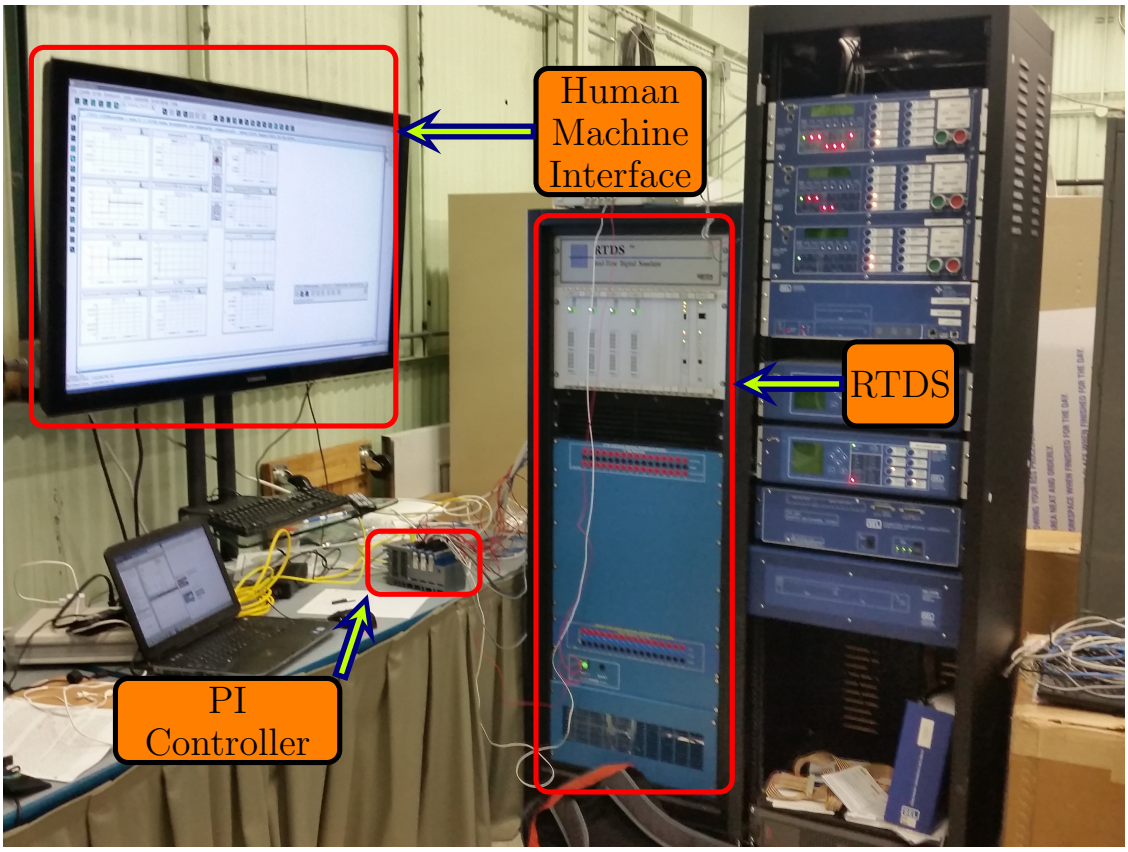
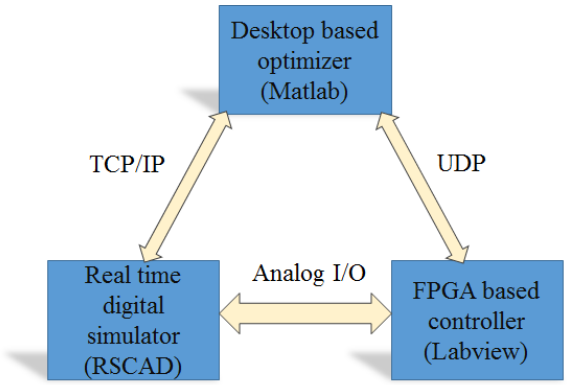


Figure 5.3: Hardware-in-loop setup for FPGA based proportional integral controller gain tuning

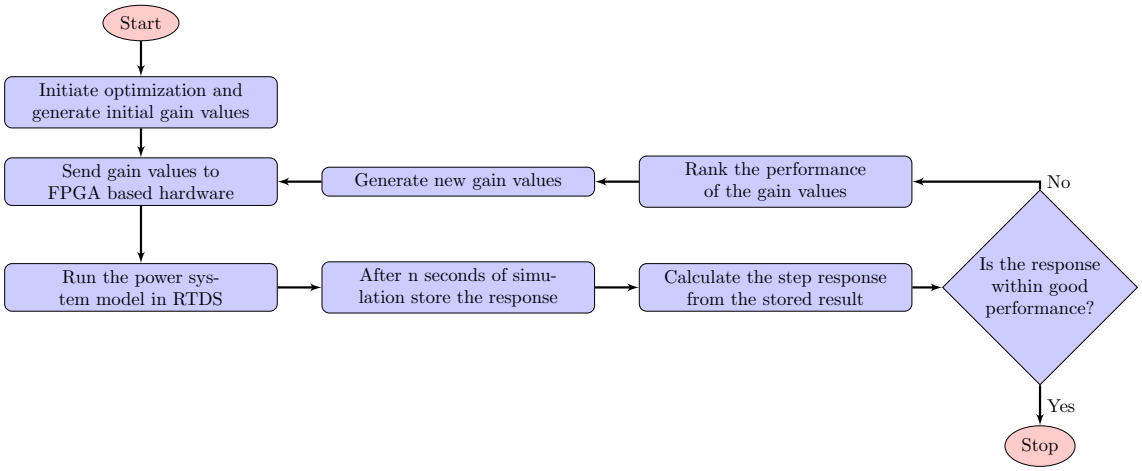
The communication protocols between these two hardware is programmed in a seamless way in order to facilitate proper tuning of PI controller. Optimization algorithm sends the k_p, and k_i value to the FPGA device. After this, the master optimizer starts the RTDS and initiates a step input in the RTDS. Once, the response to this step input is generated, RTDS stores the step response in a common database which is accessible by the optimizer. Master optimizer will utilize this step response to rank performance of these gain values. Multiobjective Pareto front criteria is used to rank the generated gain values. This flow is described in Fig. 5.4 (b).

5.1.4 Objectives used in PQ and VF controls gain tuning

The multiple objectives used for gain tuning is similar to the objectives proposed in chapter 4. The observation that could be made from the Pareto front of the optimization performed on models in chapter 4 is that the objective based on the peak value of the step response may become redundant in some cases to the overshoot objective. This is removed from the objective list for the CHIL epxerimental setup based gain tuning. The updated objective functions are rise time, overshoot,



(a) Hardware and software component in the experimental setup



(b) Gain tuning flow chart used in CHIL experimental setup

Figure 5.4: Data flow and flow chart for HIL gain tuning

and settling time. For this CHIL based gain tuning, these are the three major objectives for each PI block.

An added challenge in this gain tuning setup compared to the gain tuning in previous chapter is that there are two PI controllers. The first PI controller is used for the d-axis current control and the second PI controller is used for q-axis current control. Even though there are two PI controllers, both these controllers can have the same gain values. Due to this, the optimization algorithm need to generate only one pair of proportional and integral gain values. Nevertheless, this does not burden the optimization algorithm, an increase in the number of objective functions will add more complexity. The objective function used for the current controller gain tuning is shown in eqn. 5.1 and the objective function used for the voltage controller gain tuning is shown in eqn. 5.2. This is identical to the objective functions proposed in chapter 4 and is used for the i_d and i_q. Thus, in

$$\min (\quad risetime(i_d), \quad overshoot(i_d), \quad settlingtime(i_d)$$
$$risetime(i_q), \quad overshoot(i_q), \quad settlingtime(i_q) \ ) \tag{5.1}$$
$$Where\ i_d\ is\ the\ d-axis\ current\ and\ i_q\ is\ the\ q-axis\ current\ value.$$

$$\min (\quad peak(V_d), \quad peak(V_q), \quad peak\ time(V_d), \quad peak\ time(V_q) \ ) \tag{5.2}$$
$$Where\ V_d\ is\ the\ D-axis\ voltage\ and\ V_q\ is\ the\ Q-axis\ voltage\ value.$$

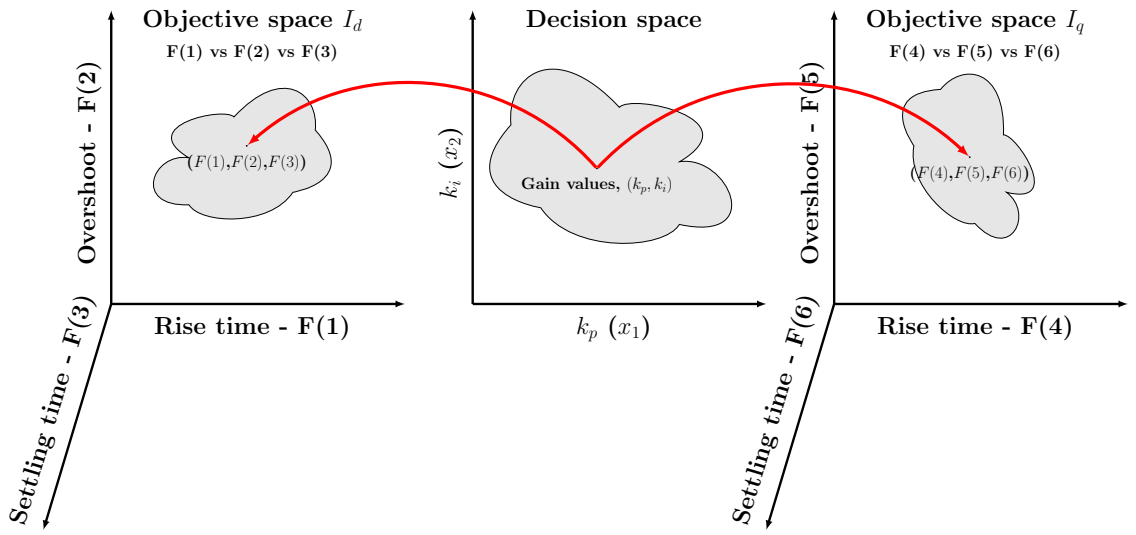


Figure 5.5: Function spaces with the different multiple objective spaces

total 6 objective functions are used for both the controllers. An example of the objective space and the decision space for these objective functions are shown in Fig. 5.5.

5.1.5 Noise generated by real time digital simulator

Transformation from *abc* to *dq* is performed in both real time digital simulator and FPGA based hardware. The transformation in RTDS generates noise due to the difference in the phase locked loop transformation performed in these two software. The optimization algorithm uses the step response calculated by RTDS for the objective function calculation. This will have noise added to the signal due to the minor calculation error.

Even in the presence of such noise generated by RTDS, optimization algorithm generated good gain values for the two PI controller. For the current controller gain tuning, a high value of 60 A of *d*-axis current and *q*-axis current was provided as the expected value. For the voltage controller gain

tuning, a slightly modified objective functions with the similar approach is used. These are shown in eqn. 5.2. The step change for the voltage tuning here is the transition of the microgrid from grid connected mode to islanded mode. Fig. 6.1 shows the step response for 60 A of dq axis step input with the gain values obtained from the optimization algorithm. Inherently, fixed point data type based computations are performed in FPGA. Due to the recent developments, floating point data type based computations can also be performed inside the FPGA. The difference between these two data types is that fixed point data type based controllers need more space in the FPGA while the floating point data type based controllers requiring less space.

5.2 Summary

This chapter presented the setup used for the controller hardware-in-loop based gain tuning. The power system models and control models used for the tuning were introduced in this chapter. The results of the gain tuning procedure will be presented in the next chapter.

Chapter 6

Results from controller hardware-in-loop based gain tuning

This chapter presents the results from gain tuning using the controller hardware-in-loop setup. The optimal k_p, k_i values along with its sensitivity for different step reference values is presented for all the controller hardware-in-loop models shown in earlier chapter. Pareto front results for the multi-objective optimization are shown to present the uniqueness and effectiveness of this approach.

6.1 Results

Fig. 6.1 and Fig. 6.5 shows the step response of floating point data type and fixed point data type based PI controller at optimal gain value. From the step responses, it could be observed that there is little difference between the performance of fixed point data type based PI and floating point data type based PI controller. Fig. 6.3 and 6.7 indicate the k_p and k_i values for this gain tuning process for both floating point and fixed point data type based PI control. In order to justify the performance of these gain values, the step input reference of the i_d, i_q values are increased from 0 A to 60 A in 30 A increments and then decreased from 60 A to 0 A in 30 A increments. The results from Fig. 6.2 and 6.6 indicate that the plant output followed the step input and settled down faster with low overshoot. The Pareto front result for the current control gain tuning using floating point and fixed point data type based PI controllers are shown in Fig. 6.4 and 6.8. These two plots indicate the relationship between the objective functions for both the data type. This is a major advantage to the engineer tuning the controllers. The tuning engineer can pick up a good gain value depending on his/her needs. The major contribution of the Pareto front is its capability to indicate whether two objective functions are conflicting or non-conflicting. The following are conflicting and

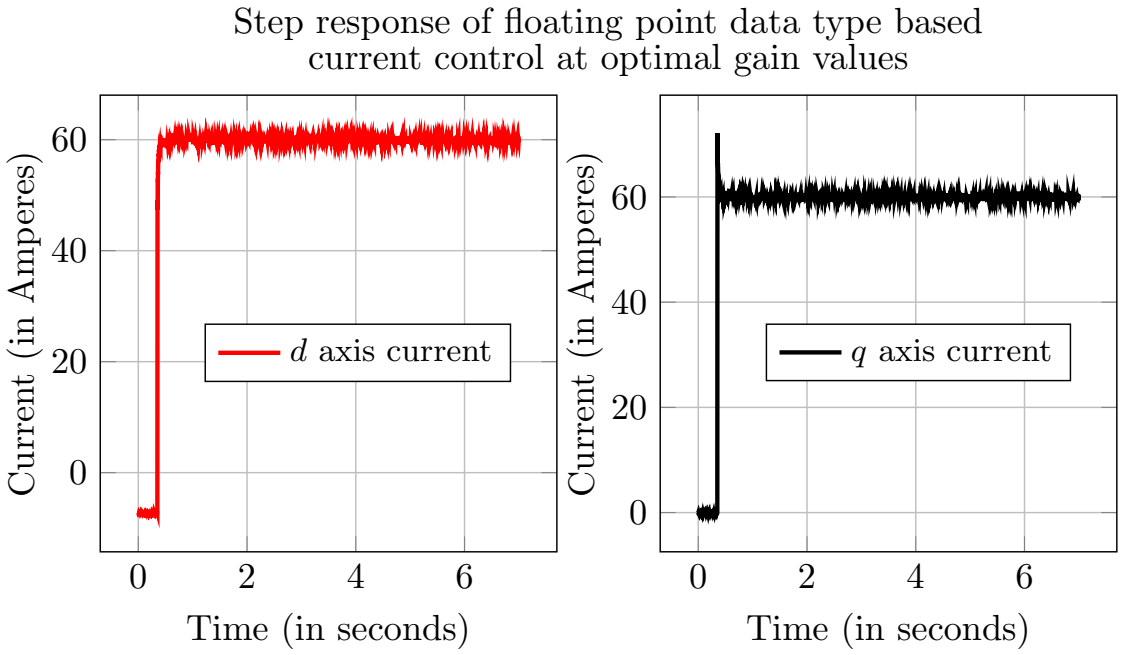


Figure 6.1: Step response of the RTDS model with optimal gain values obtained from gain tuning for floating point based controller

non-conflicting objectives for the floating point based active-reactive power control. The rest of the objective which were not included in the following list are inconclusive. These plots indicate the relationship between the d axis and q axis current step response for this particular plant model.

- Conflicting: Risetime (i_d) - Overshoot (i_d)
- Conflicting: Risetime (i_d) - Settling time (i_d)
- Conflicting: Risetime (i_d) - Overshoot (i_q)
- Conflicting: Risetime (i_d) - Settling time (i_q)
- Non-conflicting: Overshoot (i_d) - Settling time (i_d)
- Non-conflicting: Overshoot (i_d) - Overshoot (i_q)
- Non-conflicting: Overshoot (i_d) - Settling time (i_q)
- Non-conflicting: Settling time (i_d) - Settling time (i_q)
- Non-conflicting: Overshoot (i_q) - Settling time (i_q)

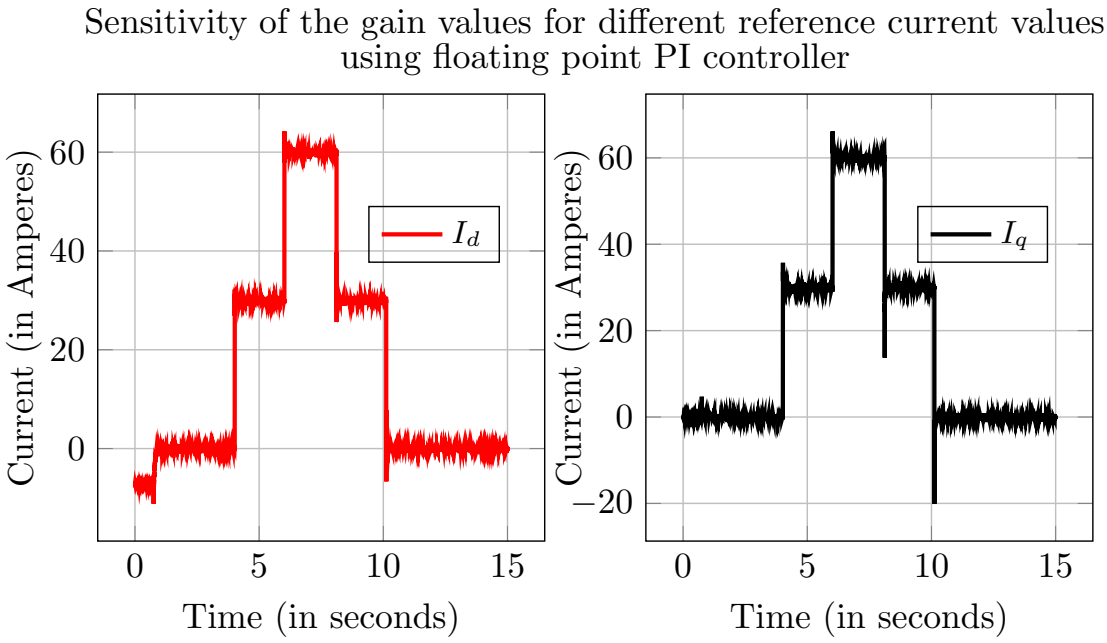


Figure 6.2: Robustness and sensitivity for optimal gain values with floating point controller

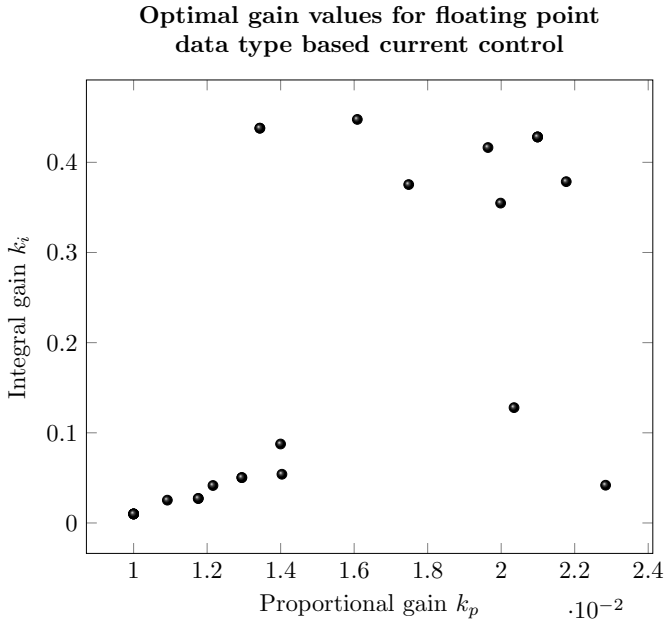


Figure 6.3: Optimal gain values for floating point data type PI control

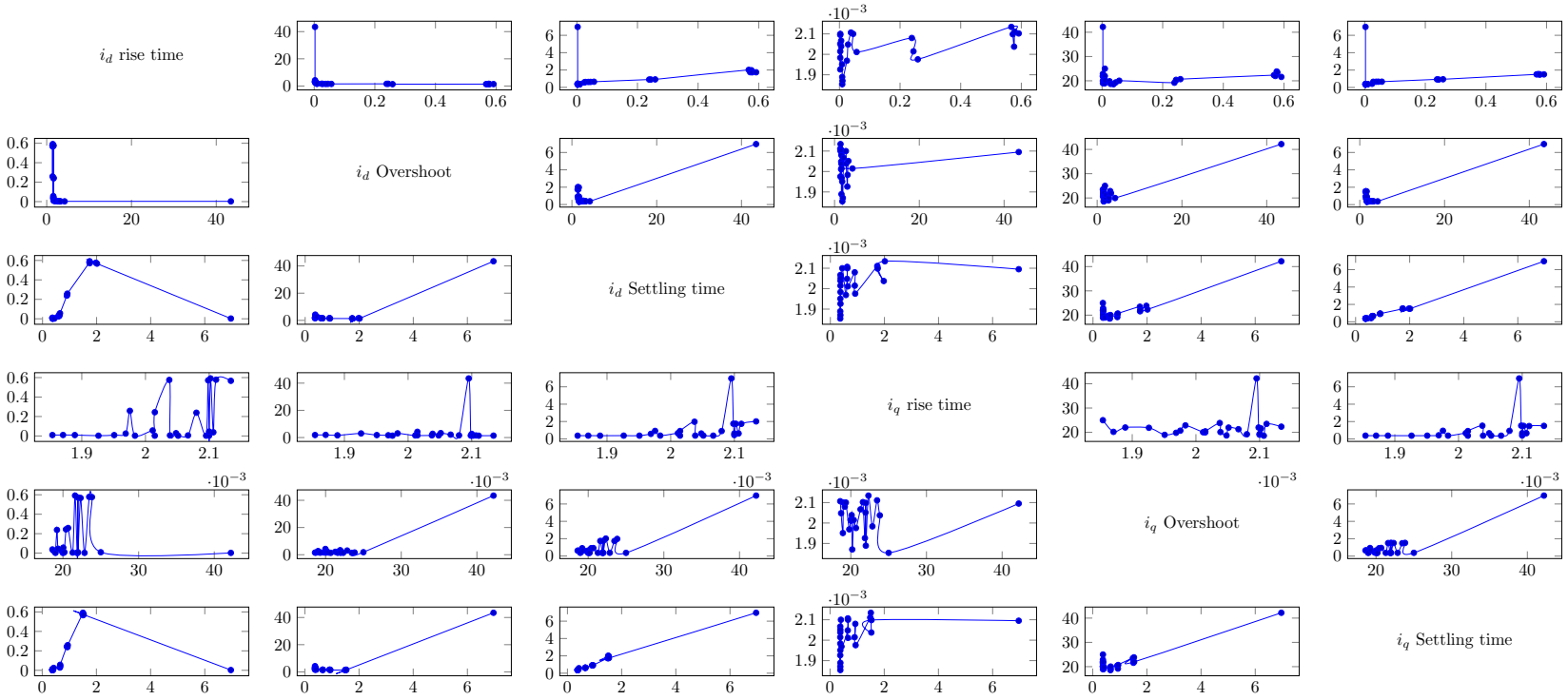


Figure 6.4: Pareto front for gain tuning of floating point data type based current control

Table 6.1: Gain value range for the different control methods tuned using CHIL setup

	k_p	k_i
Fixed point PQ control	0.013801 to 0.021088	6.28×10^{-5} to 0.00074815
Fixed point VF control	0.37 to 0.5	0.02 to 0.06
Floating point PQ control	0.011 to 0.022	0.027 to 0.43
Floating point VF control	0.59499 to 2.38	2 to 45

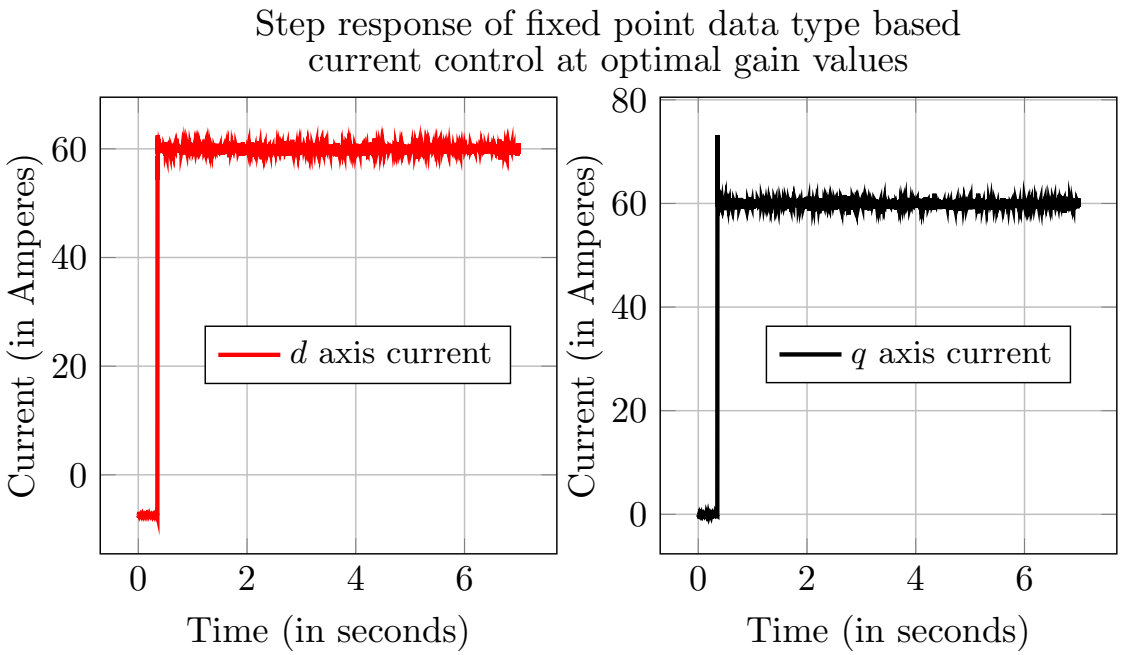


Figure 6.5: Step response of the RTDS model with optimal gain values obtained from gain tuning for fixed point based controller

The response of the system for islanding from grid connected mode is shown in Fig. 6.9 and Fig. 6.12. Again both floating point data type and fixed point data type based PI controllers will be used for the tuning. It could be observed that there is a small undershoot in d axis voltage during the transition but it immediately settles down. There is no overshoot in q axis voltage in floating point data type based PI controller. But, there is a small under voltage and over voltage in q axis voltage in fixed point data type based PI controller. In a way similar to floating point data type based controller, this also settles down very fast. The optimal k_p and k_i values for both the data type based voltage controller is shown in Fig. (insert reference here). The Pareto front for this gain tuning process is shown in Fig. 6.11 and 6.14.

Another advantage of using the results obtained from this method is the ability to find the band of gain values. Any gain value beyond this band of gain values will result in instability. The following is the band of gain values for the different models tuned in this chapter.

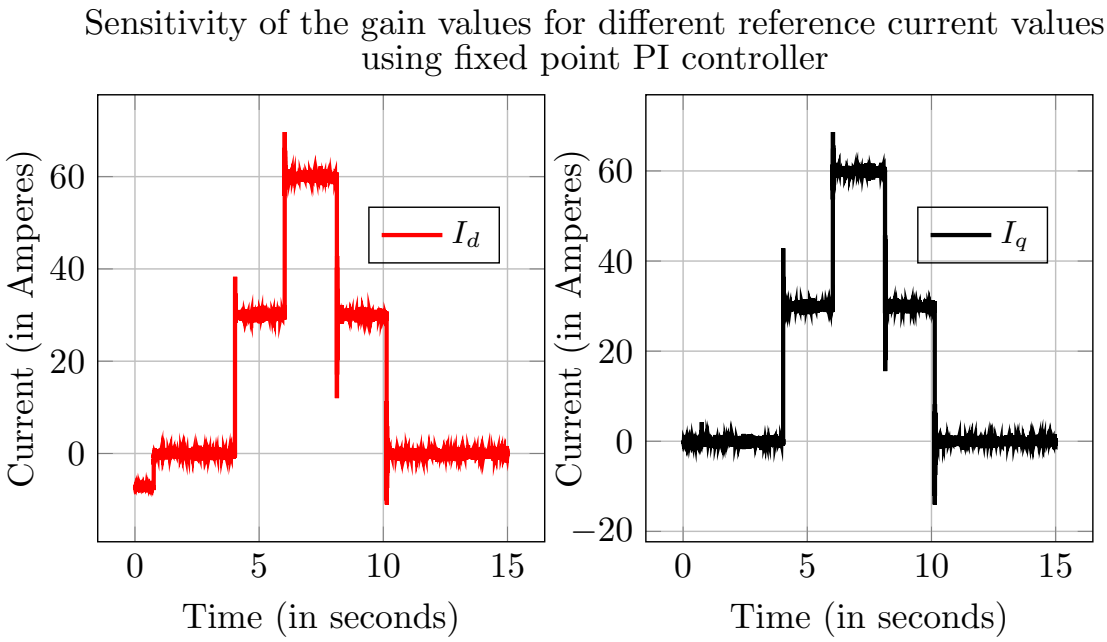


Figure 6.6: Robustness and sensitivity for optimal gain values with fixed point controller

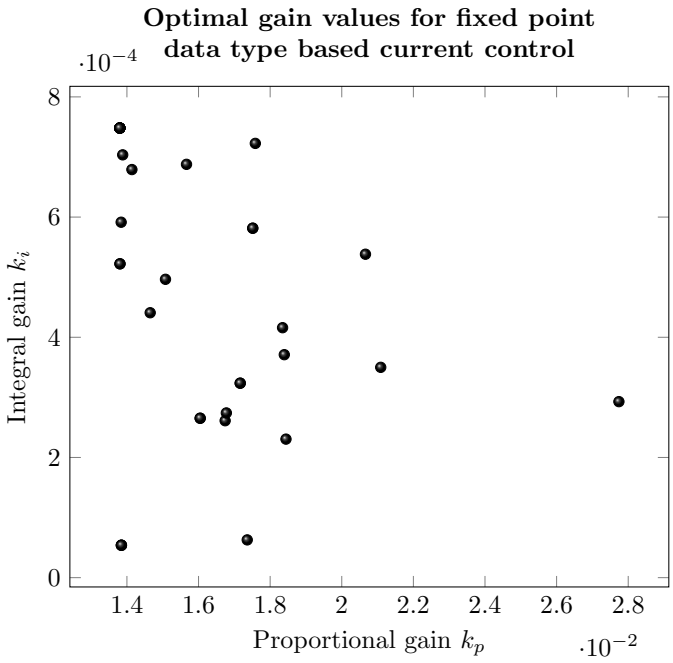


Figure 6.7: Optimal gain values for fixed point data type PI control

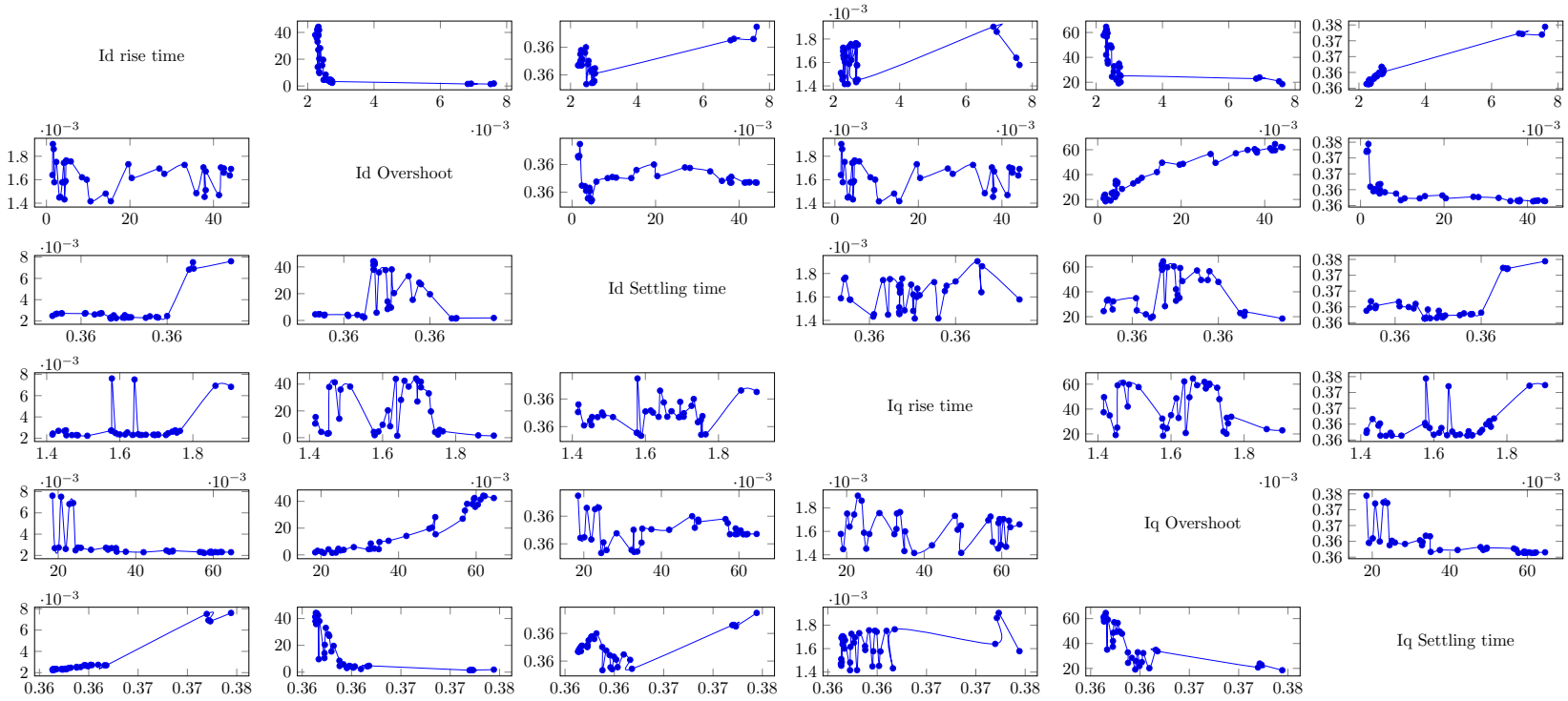


Figure 6.8: Pareto front for gain tuning of fixed point data type based current control

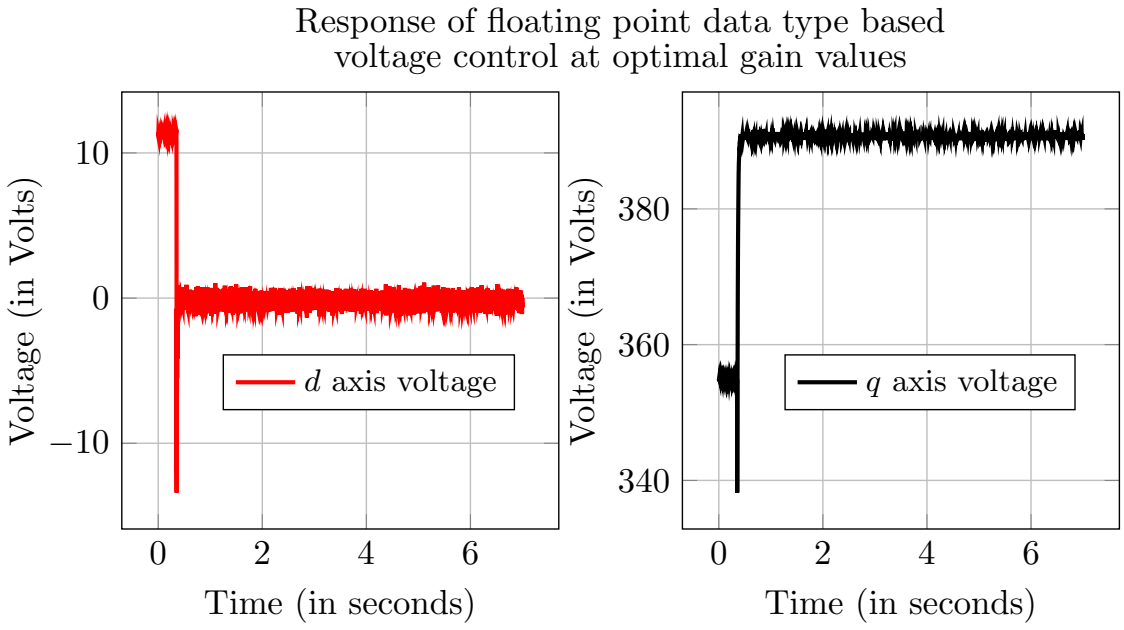


Figure 6.9: Voltage response of the RTDS model with optimal gain values obtained from gain tuning for floating point based controller

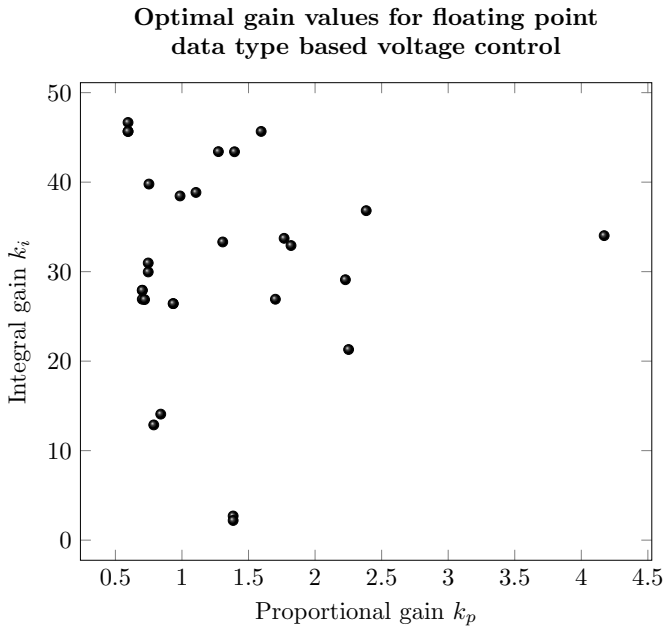


Figure 6.10: Optimal gain values for floating point data type PI based VF control

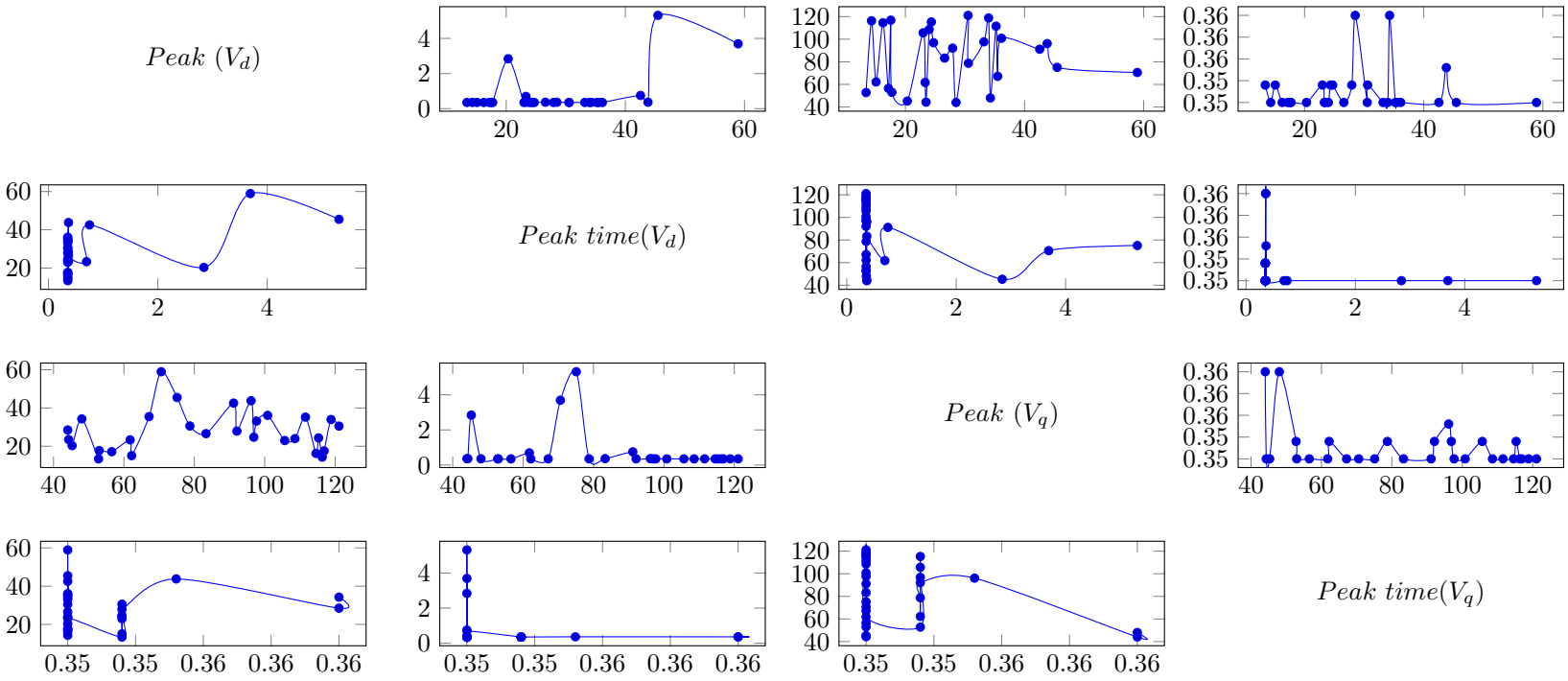


Figure 6.11: Pareto front for gain tuning of floating point data type based voltage control

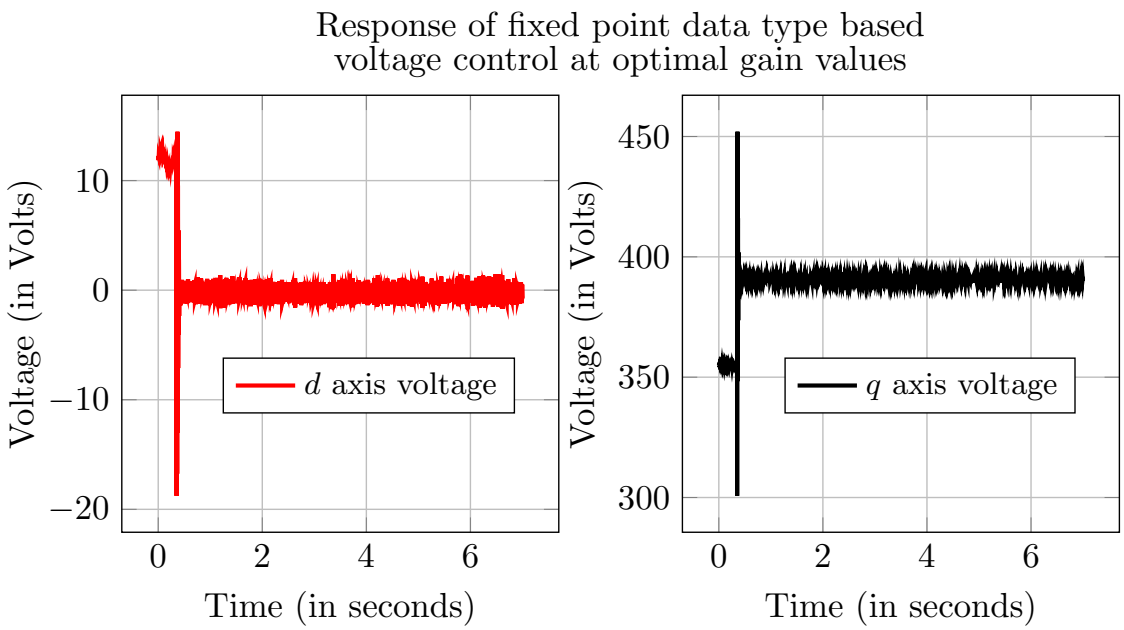


Figure 6.12: Voltage response of the RTDS model with optimal gain values obtained from gain tuning for fixed point based controller

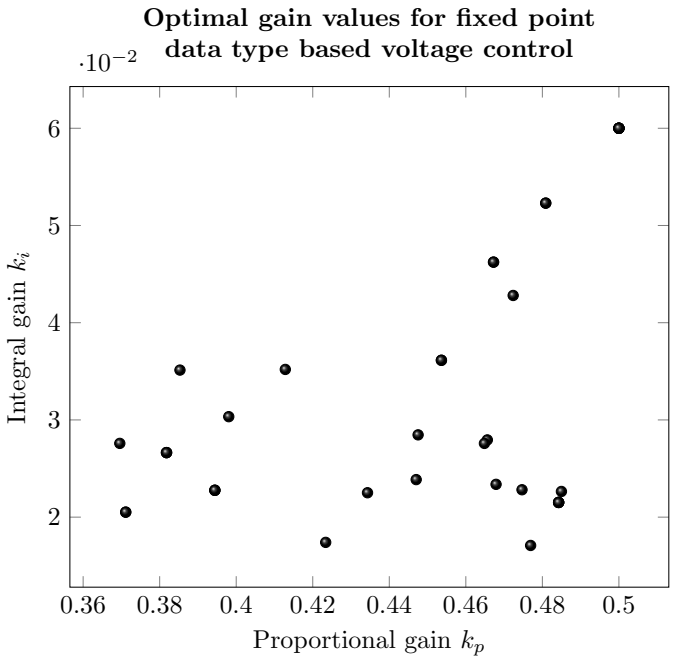


Figure 6.13: Optimal gain values for fixed point data type PI control

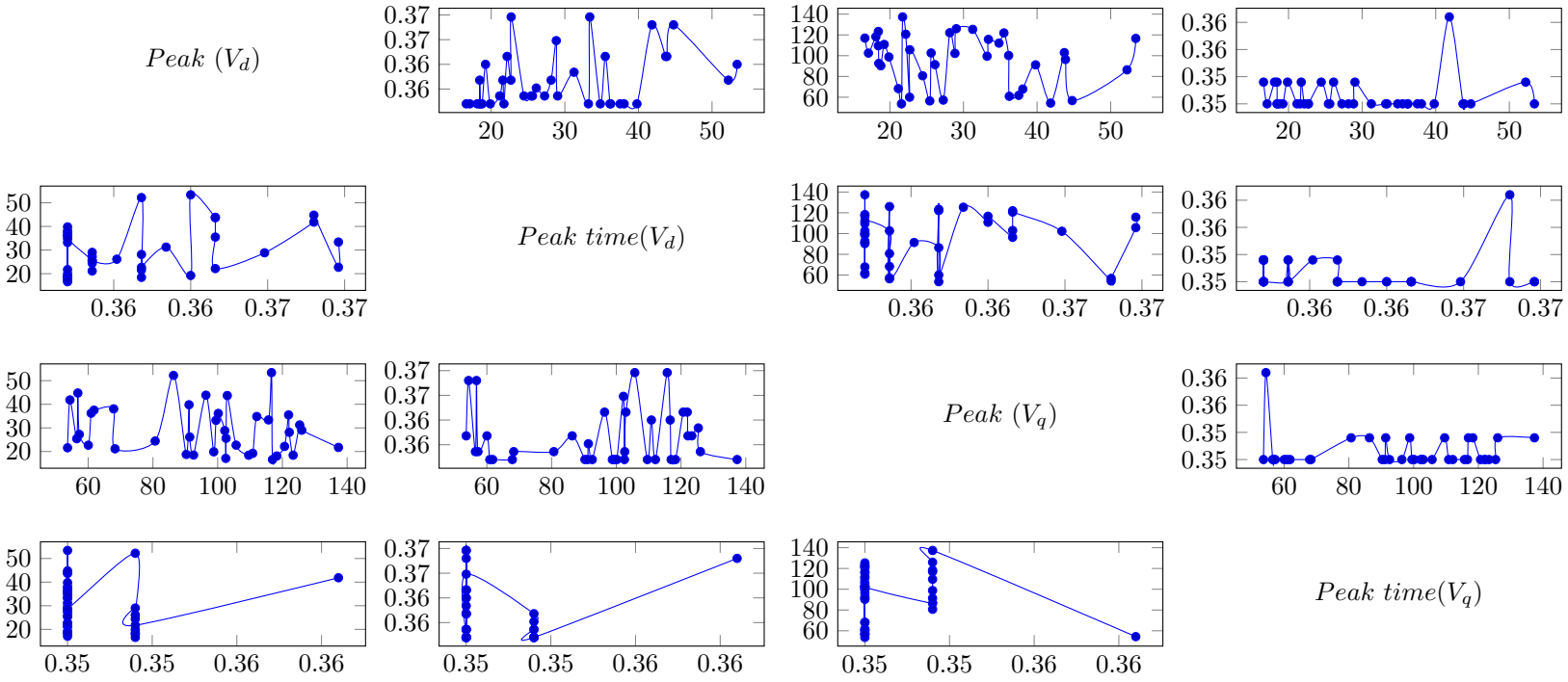


Figure 6.14: Pareto front for gain tuning of fixed point data type based voltage control

Table 6.2: The π model paramenters used in the RTDS multi-inverter case

	π_1	π_2	π_3	π_4	π_5	π_6	π_7
R_P Ω	0.0355908	0.004844	0.025964	0.0127501	0.0127501	0.000796	0.025964
X_P Ω	0.0676289	0.002576	0.014645	0.0072	0.0072	0.000417	0.014645
X_{CP} MΩ	0.0158157	0.089428	1.22826	0.3458598	0.3458598	1.48414	1.22826
R_Z Ω	0.1067724	0.017603	0.0614004	0.0252056	0.0252056	0.004738	0.0614
X_Z Ω	0.2028868	0.018219	0.15792	0.077242	0.077242	0.005507	0.15792
X_{LZ} MΩ	0.0474472	0.089428	1.22826	0.3458598	0.3458598	1.48414	1.22826

6.1.1 Multiple inverter tuning

The process setup involved tuning current and voltage control in a single inverter setup. This upcoming sections will involve tuning multiple inverters at same time. Also, these inverters will be located electrically far away from each other. This makes tuning harder due to the complex relationship between the two inverters due to the limitations of the number of RTDS analog output. Only PQ control can be tuned. In order to tune the voltage control, the same frame work of objectives can be used. and there is no change in the number of objectives as only one inverter will perform voltage tuning. The power system diagram used in this multiple inverter tuning case is shown in Fig. (insert figure number here). Inverter model described in FIg. is used here for the inverters at bus 1 and bus 3.

Since, the objective functions increased the number of kp and ki values also increased. The objective functions used for this scenario is shown in eqn. The step response of these plants at optimal kp and ki value is shown in Fig. The Pareto front for these different multiple objectives is shown in . This final test scenario justifies and completes the capability of this tuning method towards gain tuning of inverters at microgrid setting. With a better RTDS more inverters and voltage control can be tuned.

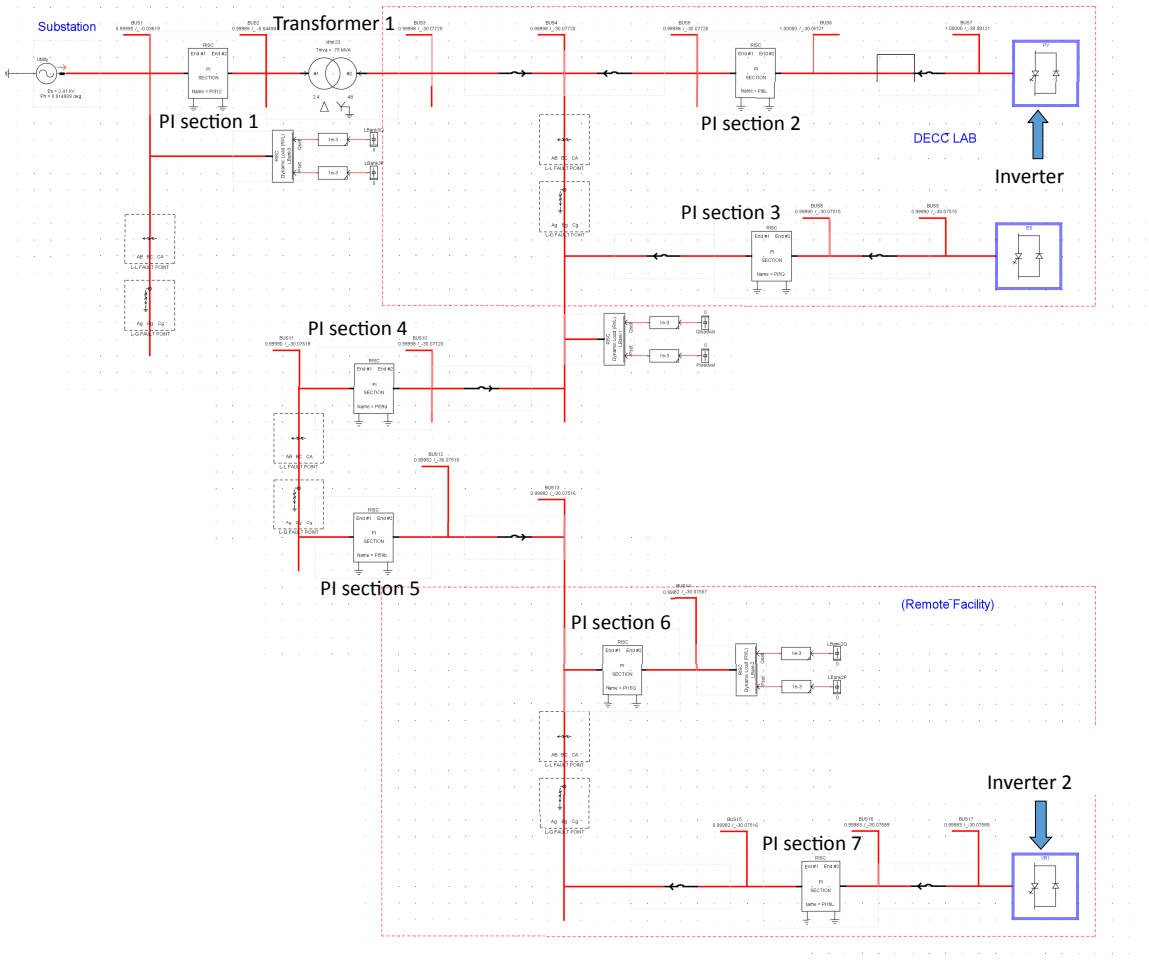


Figure 6.15: RSCAD model for the two inverter case

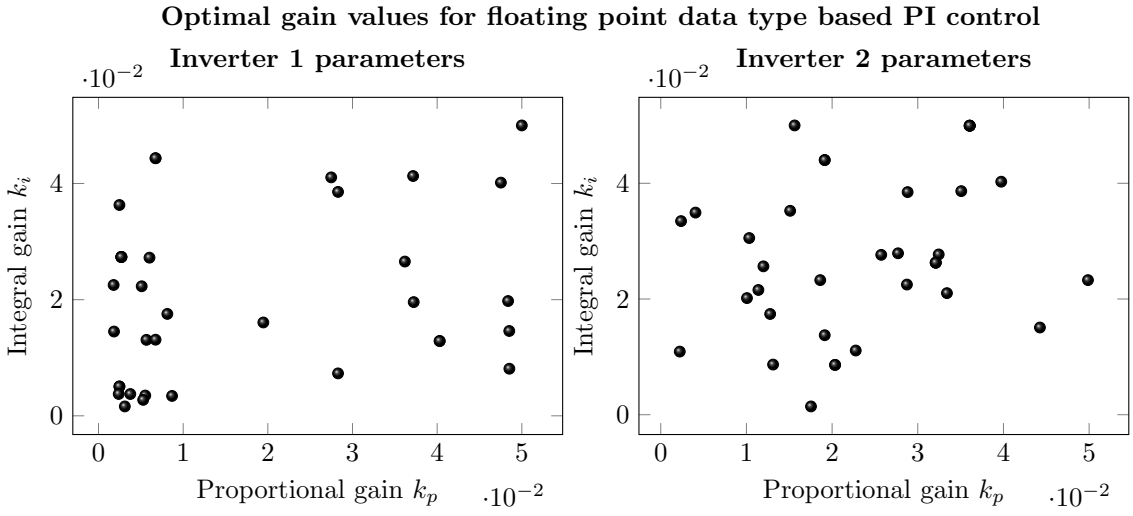


Figure 6.16: Optimal gain values for fixed point data type PI control for multiple inverters

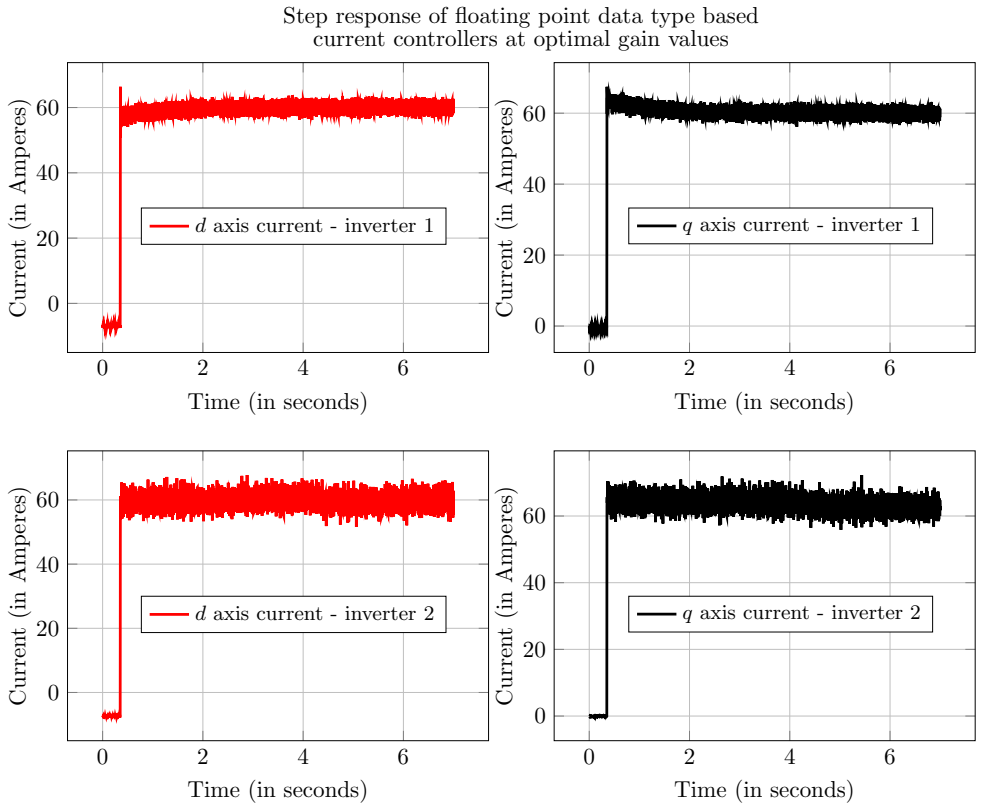


Figure 6.17: Step response of the real time digital simulator model for gains obtained from controller hardware in loop simulation under multiple inverter scenario

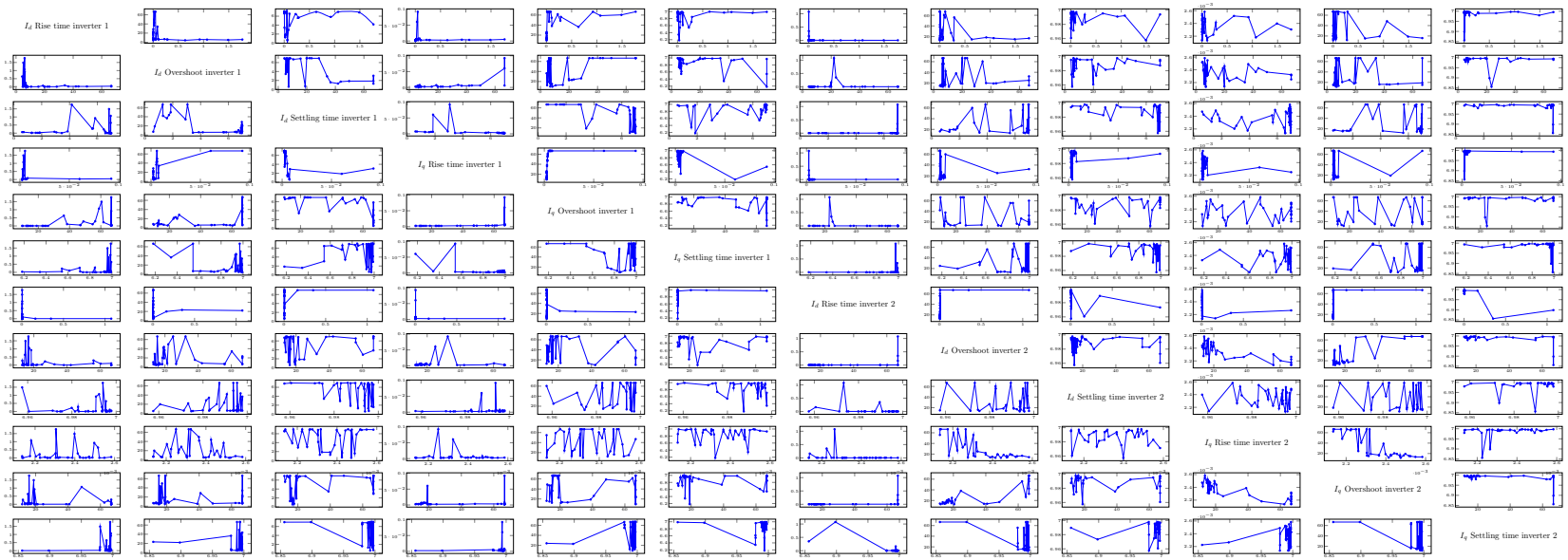
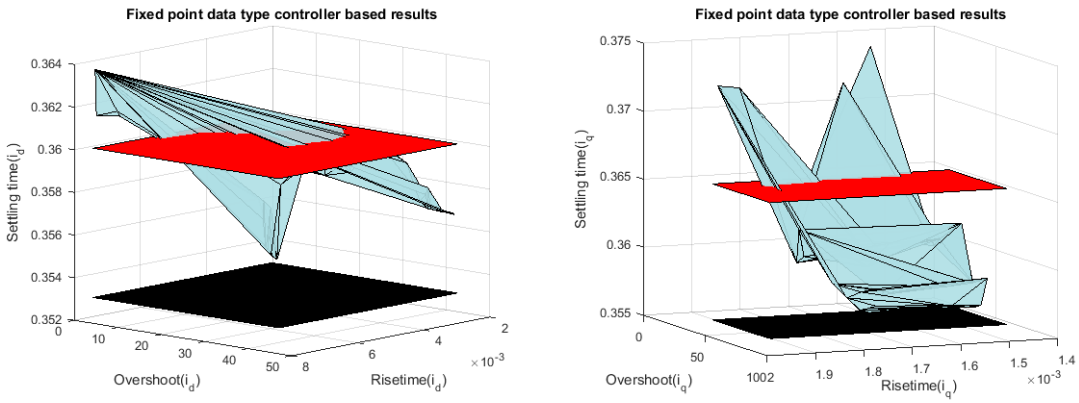


Figure 6.18: Pareto front for gain tuning of floating point data type based current control for multiple inverter case

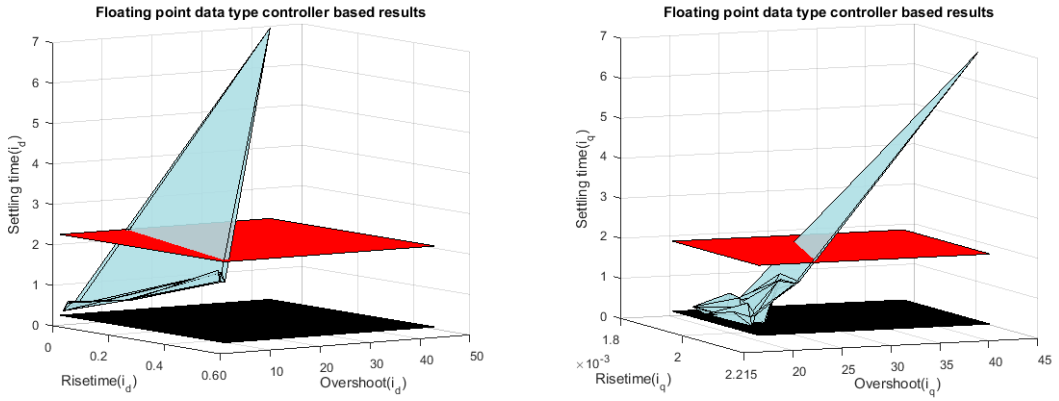


(a) f(1) vs f(2) vs f(3) **(b)** f(4) vs f(5) vs f(6)

Figure 6.19: Objective function results for fixed point data type based PQ controller

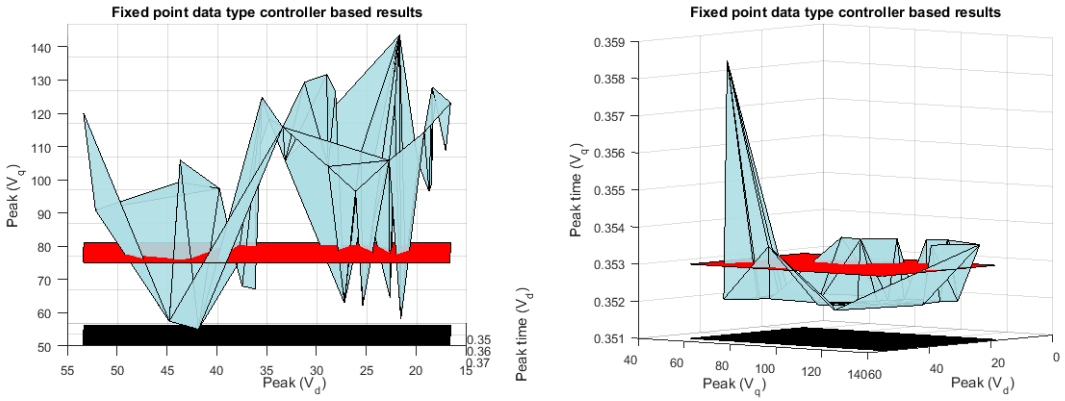
6.2 Alternate ways to visualize Pareto front

In the earlier section, two dimensional matrix plots of the different objective functions were presented. In this section, a three dimensional visualization for the different objective functions are presented. This could be observed from Fig. 6.19 to 6.22. The black surface indicates a potential minimum limit and the red surface indicate a potential maximum surface, which a user can define for an objective function. The task of visualizing objective functions beyond three dimensions is a research undertaking in itself. The visualization methods proposed here are a couple of ways for the user to understand the complex relationship which exist between these different objective functions. Also, the relationship between the different control blocks which exist in the architecture itself.



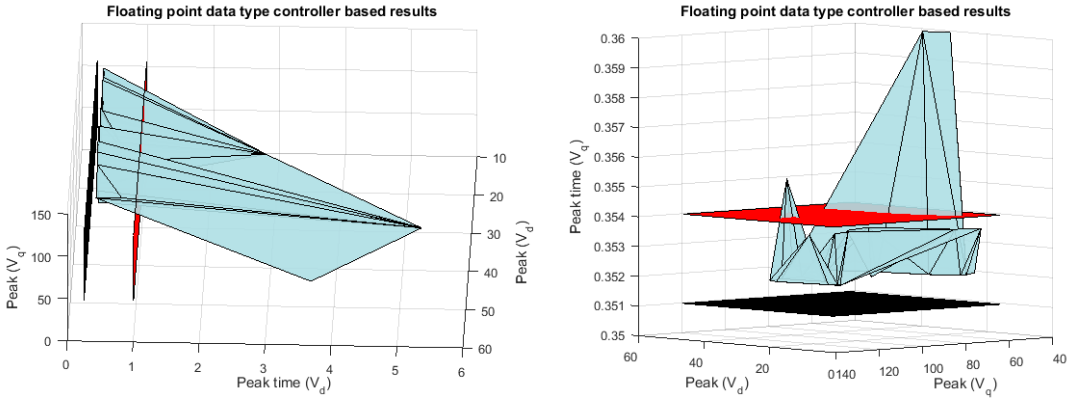
(a) f(1) vs f(2) vs f(3) **(b)** f(4) vs f(5) vs f(6)

Figure 6.20: Objective function results for floating point data type based PQ controller



(a) f(1) vs f(2) vs f(3) **(b)** f(1) vs f(3) vs f(4)

Figure 6.21: Objective function results for fixed point data type based VF controller



(a) f(1) vs f(2) vs f(3) **(b)** f(1) vs f(3) vs f(4)

Figure 6.22: Objective function results for floating point data type based VF controller

6.3 Summary

This chapter presented the results of gain tuning using RTDS and FPGA based hardware. Results indicate that gain values generated by optimization algorithm provides good step response in the simple power system and the microgrid system with multiple inverter case. This proves the reusability of the proposed algorithm. The controller architecture for both grid connected and islanded mode of operation were tuned. This denotes the modularity of the proposed approach. Six objective functions were used and the optimization algorithm found good values. The approach was then used for tuning a multiple inverter case scenario with the inverters at different bus locations. The results indicate that this approach also worked for the multiple inverter case. This proves the scalability of the proposed algorithm. The Pareto front plots shows information regarding the conflicting and non-conflicting objectives in the gain tuning optimization problem.

Chapter 7

Conclusions and future work

This research work presented a multi-objective optimization approach to obtain optimal gains for PI controller. Step response based multiple objectives were used as objective function for the optimal gain tuning problem. The results from multi-objective optimization were compared with the single objective based optimization. Genetic algorithm was used to solve both the single and multiple objective optimization problem. Simulated algorithm was used to solve only the single objective optimization problem. For the plants that could be modelled in the multiple processors, parallel computing based optimization was proposed. Genetic algorithm had the inherent advantage of the ability to perform optimization through embarrassingly parallel mode. This advantage was utilized for solving both single and multiple objectives when genetic algorithm was used.

7.1 Proposed gain tuning methodology

The proposed methodology has three parts. First, generation of gain values. The optimization algorithm will generate k_p and k_i values for the PI control block. Second, the simulation of micro grid in RTDS/plant simulator with the generated gain values. The output is then stored in a common database that is accessible by both RTDS and the optimization algorithm. And, finally ranking the gain values based on step response. If the step response is within the required performance characteristics, the algorithm will stop. Otherwise, the algorithm will generate new gain values and continue the process again.

Parallel computing is proposed to perform the optimization when using genetic algorithm. The use of parallel computing supported genetic algorithm to perform computation faster than serial computing. The improvement was observed using the speed up and the efficiency plot.

7.2 Controller hardware in loop based gain tuning

There are two main reasons which support the use of RTDS for the gain tuning here: *a*) The ability to perform simulation in real time. This will help the tuning process when a large power system model is used for the gain tuning, *b*) Perhaps the important feature which supports the use of RTDS is its ability to perform CHIL simulation and the potential to perform power HIL simulation. This enables the PI controller to be programmed in an actual hardware and use it in CHIL simulation along with the RTDS. Gain tuning of PI controller in the hardware will help in moving the controller from CHIL set up to actual inverter.

7.3 Summary of the advantages of the proposed method

A multi-objective optimization method based on proportional integral controller was proposed here. Some of the advantages of the proposed multi-objective optimization method are:

- The Pareto front for different functions provide information about the characteristics of different functions (conflicting or non conflicting).
- The conflicting or non conflicting nature of the functions depend on the plant model and is not unique for the optimization problem itself.
- The k_p and k_i values for the Pareto front will also be provided by the genetic algorithm based optimization.
- The same approach provides good gain values over three different software, and hardware platform.
- Parallel computing speeds up the genetic algorithm based optimization problem.
- Real time digital simulator based tuning also calculates the optimal solution for both software and hardware based PI controller.
- The optimization algorithm calculates the gain value even in the presence of error generated by RTDS.
- The CHIL test setup was used to tune multiple inverter case and the results show that the approach could tune multiple inverters.
- The RTDS based approach was tested for multiple inverter case and the optimization algorithm was able to set good gain values.

7.4 Future work

In this work, two PI controllers are used for the controller hardware-in-loop simulation based tuning. Proportional gain, and integral gain values are generated using the optimization algorithm. However, this was an off-line gain tuning. In the future work, this will be moved to a powerful FPGA hardware which is capable of performing the optimization tuning in itself. One such powerful hardware is PXI chassis, in which seven FPGA cards can be used. This should provide enough computation power to perform initial development for optimization algorithm.

Bibliography

[1] P. Kundur, J. Paserba, V. Ajjarapu, G. Andersson, A. Bose, C. Canizares, N. Hatziargyriou, D. Hill, A. Stankovic, C. Taylor *et al.*, "Definition and classification of power system stability ieee/cigre joint task force on stability terms and definitions," *Power Systems, IEEE Transactions on*, vol. 19, no. 3, pp. 1387–1401, 2004. 1

[2] X. Yusheng, "The way from a simple contingency to system-wide disaster-lessons from the eastern interconnection blackout in 2003 [j]," *Automation of Electric Power Systems*, vol. 18, p. 000, 2003. 1

[3] C. Marnay, N. Zhou, M. Qu, and J. Romankiewicz, "International microgrid assessment: Governance, incentives, and experience (imagine)," 2012. 1

[4] R. Torquato, Q. Shi, W. Xu, and W. Freitas, "A monte carlo simulation platform for studying low voltage residential networks," *IEEE Transactions on Smart Grid*, vol. 5, no. 6, pp. 2766–2776, 2014. 2

[5] B. R. Bredesen and P. van Lieshout, "Wind-generators in hybrid systems powering remote telecommunication sites in the far north," in *Telecommunications Energy Conference, 1987. INTELEC'87. The Ninth International*. IEEE, 1987, pp. 437–442. 3

[6] J. H. Kueffner, "Wind hybrid power system for antarctica inmarsat link," in *Telecommunications Energy Conference, 1986. INTELEC'86. International*. IEEE, 1986, pp. 297–298. 3

[7] P. McCawley, "Rural electrification in indonesia–is it time?" *Bulletin of Indonesian economic studies*, vol. 14, no. 2, pp. 34–69, 1978. 3

[8] T. Abdallah, R. Ducey, C. A. Feickert, R. S. Balog, W. Weaver, A. Akhil, and D. Menicucci, "Control dynamics of adaptive and scalable power and energy systems for military micro grids," DTIC Document, Tech. Rep., 2006. 3

[9] T. J. Hartranft, "Sustainable energy for deployed military bases." ASME, 2008. 3

[10] D. Sater, "Military energy security," 2011. 3

[11] M. Callahan, K. Anderson, S. Booth, J. Katz, and T. Tetreault, "Lessons learned from net zero energy assessments and renewable energy projects at military installations," *Contract*, vol. 303, pp. 275–3000, 2011. 3

[12] J. Stamp, "The spiders project-smart power infrastructure demonstration for energy reliability and security at us military facilities," in *Innovative Smart Grid Technologies (ISGT), 2012 IEEE PES*. IEEE, 2012, pp. 1–1. 3

[13] T. Ersal, C. Ahn, I. A. Hiskens, H. Peng, and J. L. Stein, "Impact of controlled plug-in evs on microgrids: A military microgrid example," in *Power and Energy Society General Meeting, 2011 IEEE*. IEEE, 2011, pp. 1–7. 3

[14] H. Sanborn and T. Abdallah, "Spiders microgrids for enhanced mission assurance and renewable energy utilization," DTIC Document, Tech. Rep., 2011. 3

[15] A. Kwasinski, "Lessons from field damage assessments about communication networks power supply and infrastructure performance during natural disasters with a focus on hurricane sandy," 2013. 3

[16] K. Hyer, L. M. Brown, J. J. Christensen, and K. S. Thomas, "Weathering the storm: challenges to nurses providing care to nursing home residents during hurricanes," *Applied Nursing Research*, vol. 22, no. 4, pp. e9–e14, 2009. 3

[17] D. E. Dismukes, R. F. Cope III, and D. Mesyanzhinov, "Capacity and economies of scale in electric power transmission," *Utilities Policy*, vol. 7, no. 3, pp. 155–162, 1998. 3

[18] J. Blanchard, "When the grid goes out: Backup power in disaster areas," in *Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology, 2009. Wireless VITAE 2009. 1st International Conference on*. IEEE, 2009, pp. 515–519. 4

[19] A. Kwasinski, "Technology planning for electric power supply in critical events considering a bulk grid, backup power plants, and micro-grids," *Systems Journal, IEEE*, vol. 4, no. 2, pp. 167–178, 2010. 4

[20] L. Che, M. Khodayar, and M. Shahidehpour, "Adaptive protection system for microgrids: Protection practices of a functional microgrid system." *Electrification Magazine, IEEE*, vol. 2, no. 1, pp. 66–80, 2014. 4

[21] S. Mirsaeidi, D. M. Said, M. W. Mustafa, M. H. Habibuddin, and K. Ghaffari, "Progress and problems in micro-grid protection schemes," *Renewable and Sustainable Energy Reviews*, vol. 37, pp. 834–839, 2014. 5

[22] C. Brunner, "Iec 61850 for power system communication," in *Transmission and Distribution Conference and Exposition, 2008. T&# x00026; D. IEEE/PES*. IEEE, 2008, pp. 1–6. 5

[23] "Ieee standard for electric power systems communications-distributed network protocol (dnp3)," *IEEE P1815/D11, April 2012 - Approved Draft*, pp. 1–866, Oct 2012. 5

[24] Z. Chao, H. Can, Z. Jianyong, and Y. Yun, "Real-time measurement of communication subsystem of intelligent electronic devices," in *International Conference on Electric Information and Control Engineering (ICEICE)*. IEEE, 2011, pp. 626–629. 5

[25] T. Ustun, C. Ozansoy, and A. Zayegh, "Extending iec 61850-7-420 for distributed generators with fault current limiters," in *Innovative Smart Grid Technologies Asia (ISGT), 2011 IEEE PES*, Nov 2011, pp. 1–8. 5

[26] R. H. Lasseter, "Microgrids," in *Power Engineering Society Winter Meeting, 2002. IEEE*, vol. 1. IEEE, 2002, pp. 305–308. 6, 7

[27] N. Hatziargyriou, H. Asano, R. Iravani, and C. Marnay, "Microgrids," *Power and Energy Magazine, IEEE*, vol. 5, no. 4, pp. 78–94, 2007. 6

[28] T. S. Basso and R. DeBlasio, "Ieee 1547 series of standards: interconnection issues," *Power Electronics, IEEE Transactions on*, vol. 19, no. 5, pp. 1159–1162, 2004. 7

[29] B. M. Weedy, B. J. Cory, N. Jenkins, J. Ekanayake, and G. Strbac, *Electric power systems*. John Wiley & Sons, 2012. 7

[30] T. Ohnishi, "Three phase pwm converter/inverter by means of instantaneous active and reactive power control," in *Industrial Electronics, Control and Instrumentation, 1991. Proceedings. IECON'91., 1991 International Conference on*. IEEE, 1991, pp. 819–824. 10

[31] P. Cominos and N. Munro, "Pid controllers: recent tuning methods and design to specification," *IEE Proceedings-Control Theory and Applications*, vol. 149, no. 1, pp. 46–53, 2002. 11, 30

[32] J. Ziegler and N. Nichols, "Optimum settings for automatic controllers," *trans. ASME*, vol. 64, no. 11, 1942. 11, 49

[33] R. Teodorescu and F. Blaabjerg, "Flexible control of small wind turbines with grid failure detection operating in stand-alone and grid-connected mode," *Power Electronics, IEEE Transactions on*, vol. 19, no. 5, pp. 1323–1332, 2004. 11

[34] A. Timbus, M. Liserre, R. Teodorescu, P. Rodriguez, and F. Blaabjerg, "Evaluation of current controllers for distributed power generation systems," *Power Electronics, IEEE Transactions on*, vol. 24, no. 3, pp. 654–664, 2009. 11

[35] M. N. Kumar and K. Vasudevan, "Bi-directional real and reactive power control using constant frequency hysteresis control with reduced losses," *Electric Power Systems Research*, vol. 76, no. 1, pp. 127–135, 2005. 11

[36] S. Mikkili and A. Panda, "Real-time implementation of pi and fuzzy logic controllers based shunt active filter control strategies for power quality improvement," *International Journal of Electrical Power and Energy Systems*, vol. 43, no. 1, pp. 1114–1126, 2012. 11

[37] M. Castilla, J. Miret, J. Matas, L. de Vicua, and J. M. Guerrero, "Linear current control scheme with series resonant harmonic compensator for single-phase grid-connected photovoltaic inverters," *Industrial Electronics, IEEE Transactions on*, vol. 55, no. 7, pp. 2724–2733, 2008. 11

[38] N. Femia, G. Lisi, G. Petrone, G. Spagnuolo, and M. Vitelli, "Distributed maximum power point tracking of photovoltaic arrays: Novel approach and system analysis," *Industrial Electronics, IEEE Transactions on*, vol. 55, no. 7, pp. 2610–2621, 2008. 11

[39] J. M. Carrasco, L. G. Franquelo, J. T. Bialasiewicz, E. Galván, R. P. Guisado, M. A. Prats, J. I. León, and N. Moreno-Alfonso, "Power-electronic systems for the grid integration of renewable energy sources: A survey," *Industrial Electronics, IEEE Transactions on*, vol. 53, no. 4, pp. 1002–1016, 2006. 11

[40] S. B. Kjaer, J. K. Pedersen, and F. Blaabjerg, "Power inverter topologies for photovoltaic modules-a review," in *Industry Applications Conference, 2002. 37th IAS Annual Meeting. Conference Record of the*, vol. 2. IEEE, 2002, pp. 782–788. 11, 48

[41] ——, "A review of single-phase grid-connected inverters for photovoltaic modules," *Industry Applications, IEEE Transactions on*, vol. 41, no. 5, pp. 1292–1306, 2005. 11

[42] Y. Xue, L. Chang, S. B. Kjaer, J. Bordonau, and T. Shimizu, "Topologies of single-phase inverters for small distributed power generators: an overview," *Power Electronics, IEEE Transactions on*, vol. 19, no. 5, pp. 1305–1314, 2004. 11

[43] A. Mohd, E. Ortjohann, D. Morton, and O. Omari, "Review of control techniques for inverters parallel operation," *Electric Power Systems Research*, vol. 80, no. 12, pp. 1477–1487, 2010. 11

[44] H. Li, F. Li, Y. Xu, D. Rizy, and J. D. Kueck, "Adaptive voltage control with distributed energy resources: Algorithm, theoretical analysis, simulation, and field test verification," *Power Systems, IEEE Transactions on*, vol. 25, no. 3, pp. 1638–1647, 2010. 11, 49

[45] K. J. Astrom, "Pid controllers: theory, design and tuning," *Instrument Society of America*, 1995. 13

[46] K. Ogata, *Discrete-time control systems*. Prentice Hall Englewood Cliffs, NJ, 1995, vol. 2. 14

[47] K. J. Aström and R. M. Murray, *Feedback systems: an introduction for scientists and engineers*. Princeton university press, 2010. 16

[48] J. Ziegler and N. Nichols, "Optimum settings for automatic controllers," *trans. ASME*, vol. 64, no. 11, 1942. 17

[49] E. G. Collins Jr, C. Fan, and R. Millett, "Automated pi tuning for a weigh belt feeder via unfalsified control," in *Decision and Control, 1999. Proceedings of the 38th IEEE Conference on*, vol. 1. IEEE, Conference Proceedings, pp. 785–790. 17, 22

[50] C. Hang and K. K. Sin, "On-line auto tuning of pid controllers based on the cross-correlation technique," *Industrial Electronics, IEEE Transactions on*, vol. 38, no. 6, pp. 428–437, 1991. 17, 18

[51] L. Wang and W. Cluett, "Tuning pid controllers for integrating processes," *IEE Proceedings-Control Theory and Applications*, vol. 144, no. 5, pp. 385–392, 1997. 17, 19

[52] G. Cohen and G. Coon, "Theoretical consideration of retarded control," *Trans. Asme*, vol. 75, no. 1, pp. 827–834, 1953. 17, 18

[53] P. S. Fruehauf, I. Chien, and M. D. Lauritsen, "Simplified imc-pid tuning rules," *ISA Transactions*, vol. 33, no. 1, pp. 43–59, 1994. 18

[54] G. E. Rotstein and D. R. Lewin, "Simple pi and pid tuning for open-loop unstable systems," *Industrial and engineering chemistry research*, vol. 30, no. 8, pp. 1864–1869, 1991. 18

[55] E. Grassi and K. Tsakalis, "Pid controller tuning by frequency loop-shaping," in *Decision and Control, 1996., Proceedings of the 35th IEEE Conference on*, vol. 4. IEEE, Conference Proceedings, pp. 4776–4781. 18

[56] L. Wang, T. Barnes, and W. Cluett, "New frequency-domain design method for pid controllers," *IEE Proceedings-Control Theory and Applications*, vol. 142, no. 4, pp. 265–271, 1995. 19

[57] H.-W. Fung, Q.-G. Wang, and T.-H. Lee, "Pi tuning in terms of gain and phase margins," *Automatica*, vol. 34, no. 9, pp. 1145–1149, 1998. 19

[58] W. Ho, K. Lim, and W. Xu, "Optimal gain and phase margin tuning for pid controllers," *Automatica*, vol. 34, no. 8, pp. 1009–1014, 1998. 20

[59] W. Ho and W. Xu, "Pid tuning for unstable processes based on gain and phase-margin specifications," in *Control Theory and Applications, IEE Proceedings-*, vol. 145. IET, Conference Proceedings, pp. 392–396. 20

[60] Y. Lee, S. Park, M. Lee, and C. Brosilow, "Pid controller tuning for desired closedloop responses for si/so systems," *Aiche journal*, vol. 44, no. 1, pp. 106–115, 1998. 21

[61] D. P. Atherton, "Pid controller tuning," *Computing & control engineering journal*, vol. 10, no. 2, pp. 44–50, 1999. 21

[62] I.-L. Chien, H.-P. Huang, and J.-C. Yang, "A simple multiloop tuning method for pid controllers with no proportional kick," *Industrial & engineering chemistry research*, vol. 38, no. 4, pp. 1456–1468, 1999. 21, 22

[63] M. Jun and M. G. Safonov, "Automatic pid tuning: An application of unfalsified control," *Proc. IEEE CCA/CACSD*, vol. 2, pp. 328–333, 1999. 22

[64] D. Lee, M. Lee, S. Sung, and I. Lee, "Robust pid tuning for smith predictor in the presence of model uncertainty," *Journal of Process Control*, vol. 9, no. 1, pp. 79–85, 1999. 23

[65] G. Liu and S. Daley, "Optimal-tuning pid controller design in the frequency domain with application to a rotary hydraulic system," *Control Engineering Practice*, vol. 7, no. 7, pp. 821–830, 1999. 23, 26

[66] Q.-G. Wang, H.-W. Fung, and Y. Zhang, "Pid tuning with exact gain and phase margins," *ISA transactions*, vol. 38, no. 3, pp. 243–249, 1999. 24

[67] Q.-G. Wang, T.-H. Lee, H.-W. Fung, Q. Bi, and Y. Zhang, "Pid tuning for improved performance," *Control Systems Technology, IEEE Transactions on*, vol. 7, no. 4, pp. 457–465, 1999. 24

[68] E. Grassi and K. Tsakalis, "Pid controller tuning by frequency loop-shaping: application to diffusion furnace temperature control," *Control Systems Technology, IEEE Transactions on*, vol. 8, no. 5, pp. 842–847, 2000. 24

[69] J.-B. He, Q.-G. Wang, and T.-H. Lee, "Pi/pid controller tuning via lqr approach," *Chemical Engineering Science*, vol. 55, no. 13, pp. 2429–2439, 2000. 25

[70] Y.-G. Wang, H.-H. Shao, and J. Wang, "Pi tuning for processes with large dead time," in *American Control Conference, 2000. Proceedings of the 2000*, vol. 6. IEEE, Conference Proceedings, pp. 4274–4278. 25

[71] G. Liu and S. Daley, "Optimal-tuning pid control for industrial systems," *Control Engineering Practice*, vol. 9, no. 11, pp. 1185–1194, 2001. 25

[72] G. Mann, B.-G. Hu, and R. Gosine, "Time-domain based design and analysis of new pid tuning rules," *IEE Proceedings-Control Theory and Applications*, vol. 148, no. 3, pp. 251–261, 2001. 26

[73] C. Pedret, R. Vilanova, R. Moreno, and I. Serra, "A refinement procedure for pid controller tuning," *Computers & chemical engineering*, vol. 26, no. 6, pp. 903–908, 2002. 27

[74] Y.-G. Wang and W.-J. Cai, "Advanced proportional-integral-derivative tuning for integrating and unstable processes with gain and phase margin specifications," *Industrial & engineering chemistry research*, vol. 41, no. 12, pp. 2910–2914, 2002. 27

[75] Y. Chen, C. Hu, and K. L. Moore, "Relay feedback tuning of robust pid controllers with iso-damping property," in *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, vol. 3. IEEE, Conference Proceedings, pp. 2180–2185. 27, 28

[76] W. Tan, J. Liu, F. Fang, and Y. Chen, "Tuning of pid controllers for boiler-turbine units," *ISA transactions*, vol. 43, no. 4, pp. 571–583, 2004. 28

[77] C. Zhao, D. Xue, and Y. Q. Chen, "A fractional order pid tuning algorithm for a class of fractional order plants," in *Mechatronics and Automation, 2005 IEEE International Conference*, vol. 1. IEEE, Conference Proceedings, pp. 216–221. 28

[78] S. Sayedain and I. Boiko, "Optimal pi tuning rules for flow loop, based on modified relay feedback test," in *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*. IEEE, Conference Proceedings, pp. 7063–7068. 29

[79] J. J. Gude and E. Kahoraho, "Kappa-tau type pi tuning rules for specified robust levels," in *Preprints IFAC Conference on Advances in PID Control, Brescia, Italy*, Conference Proceedings. 29

[80] ——, "Kappa-tau type pi tuning rules for specified robust levels: The frequency response method," in *Emerging Technologies & Factory Automation (ETFA), 2012 IEEE 17th Conference on*. IEEE, Sept 2012, Conference Proceedings, pp. 1–8. 29

[81] N. Z. Abidin, S. Sahlan, and N. A. Wahab, "Optimization tuning of pi controller of quadruple tank process," in *Control Conference (AUCC), 2013 3rd Australian*. IEEE, Conference Proceedings, pp. 331–335. 30

[82] B. Popadic, B. Dumnic, D. Milicevic, V. Katic, and Z. Corba, "Tuning methods for pi controller-comparison on a highly modular drive," in *Energy (IYCE), 2013 4th International Youth Conference on*. IEEE, Conference Proceedings, pp. 1–6. 30

[83] U. Yildirim, E. Dincel, and M. T. Soylemez, "A symbolic pi tuning method for first order systems with time delay," in *Control System, Computing and Engineering (ICCSCE), 2013 IEEE International Conference on*, Conference Proceedings, pp. 325–328. 30

[84] K. J. strm, T. Hgglund, C. C. Hang, and W. K. Ho, "Automatic tuning and adaptation for pid controllers-a survey," *Control Engineering Practice*, vol. 1, no. 4, pp. 699–714, 1993. 30

[85] G. J. Silva, A. Datta, and S. P. Bhattacharyya, "On the stability and controller robustness of some popular pid tuning rules," *Automatic Control, IEEE Transactions on*, vol. 48, no. 9, pp. 1638–1641, 2003. 30

[86] W. Ho, O. Gan, E. Tay, and E. Ang, "Performance and gain and phase margins of well-known pid tuning formulas," *Control Systems Technology, IEEE Transactions on*, vol. 4, no. 4, pp. 473–477, 1996. 30

[87] W. Tan, J. Liu, T. Chen, and H. J. Marquez, "Comparison of some well-known pid tuning formulas," *Computers and chemical engineering*, vol. 30, no. 9, pp. 1416–1423, 2006. 30

[88] M. Ajmeri and A. Ali, "A comparative study of pi tuning methods for integrating processes," in *Control, Automation, Robotics and Embedded Systems (CARE), 2013 International Conference on*, Conference Proceedings, pp. 1–6. 30

[89] O. Lequin, M. Gevers, M. Mossberg, E. Bosmans, and L. Triest, "Iterative feedback tuning of pid parameters: comparison with classical tuning rules," *Control Engineering Practice*, vol. 11, no. 9, pp. 1023–1033, 2003. 30

[90] A. O'Dwyer, "A summary of pi and pid controller tuning rules for processes with time delay. part 2: Pid controller tuning rules," 2000. 30

[91] R. C. Panda, C.-C. Yu, and H.-P. Huang, "Pid tuning rules for sopdt systems: Review and some new results," *ISA transactions*, vol. 43, no. 2, pp. 283–295, 2004. 30

[92] M. Ajmeri and A. Ali, "Pi tuning for integrating first order plus time delay processes," in *Industrial Technology (ICIT), 2013 IEEE International Conference on*. IEEE, Conference Proceedings, pp. 157–162. 30

[93] M. Zhuang and D. Atherton, "Automatic tuning of optimum pid controllers," in *Control Theory and Applications, IEE Proceedings D*, vol. 140. IET, Conference Proceedings, pp. 216–224. 30

[94] A. Jones and P. de Moura Oliveira, "Auto-tuning of pi smith predictor controllers using genetic algorithms," 1996. 30

[95] S. Jing-Chung, "New tuning method for pid controller," in *Control Applications, 2001. (CCA '01). Proceedings of the 2001 IEEE International Conference on*, Conference Proceedings, pp. 459–464. 31

[96] J. Herrero, X. Blasco, M. Martinez, and J. Salcedo, "Optimal pid tuning with genetic algorithms for non-linear process models," in *15th Triennial World Congress, Barcelona, Spain*, Conference Proceedings. 31

[97] P.-J. Wang, "Pi-tuning methods based on ga," in *Machine Learning and Cybernetics, 2002. Proceedings. 2002 International Conference on*, vol. 1. IEEE, Conference Proceedings, pp. 544–547. 31

[98] G. M. de Almeida, V. V. R. e Silva, E. G. Nepomuceno, and R. Yokoyama, *Application of genetic programming for fine tuning PID controller parameters designed through Ziegler-Nichols technique*. Springer, 2005, pp. 313–322. 32

[99] D. S. Pereira and J. O. Pinto, "Genetic algorithm based system identification and pid tuning for optimum adaptive control," in *Advanced Intelligent Mechatronics. Proceedings, 2005 IEEE/ASME International Conference on*. IEEE, Conference Proceedings, pp. 801–806. 32

[100] N. J. Killingsworth and M. Krstic, "Pid tuning using extremum seeking: online, model-free performance optimization," *Control Systems, IEEE*, vol. 26, no. 1, pp. 70–79, 2006. 32

[101] F. Mesa, J. L. Lozano, and L. Marin, "On the consideration of foptd and soptd responses as bounds of pi tuning," in *Electrotechnical Conference, 2006. MELECON 2006. IEEE Mediterranean*. IEEE, Conference Proceedings, pp. 421–424. 33

[102] S. G. Kumar, R. Jain, N. Anantharaman, V. Dharmalingam, and K. Begum, "Genetic algorithm based pid controller tuning for a model bioreactor," *indian chemical engineer*, vol. 50, no. 3, pp. 214–226, 2008. 33

[103] J. Esch, T. Konings, and S. X. Ding, "A convex optimisation based approach to pi-controller tuning for time invariant integrator and low order lag plus delay plants," in *Control*

Applications (CCA), 2013 IEEE International Conference on. IEEE, Conference Proceedings, pp. 1259–1264. 33

[104] S. Tavakoli, I. Griffin, and P. J. Fleming, "Multi-objective optimization approach to the pi tuning problem," in *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on.* IEEE, Conference Proceedings, pp. 3165–3171. 33

[105] T. Bäck, "Introduction to evolutionary algorithms," *Evolutionary Computation*, vol. 1, pp. 59–63, 2000. 37

[106] A. Popov and K. Filipova, "Genetic algorithms-synthesis of finite state machines," in *Electronics Technology: Meeting the Challenges of Electronics Technology Progress, 2004. 27th International Spring Seminar on*, vol. 3. IEEE, 2004, pp. 388–392. 38

[107] S. Kirkpatrick, M. Vecchi *et al.*, "Optimization by simmulated annealing," *science*, vol. 220, no. 4598, pp. 671–680, 1983. 42

[108] K. Deb, *Multi-objective optimization using evolutionary algorithms.* John Wiley & Sons, 2001, vol. 16. 43

[109] F. Li and R. P. Broadwater, "Distributed algorithms with theoretic scalability analysis of radial and looped load flows for power distribution systems," *Electric Power Systems Research*, vol. 65, no. 2, pp. 169–177, 2003. 43, 45

[110] V. Kumar, A. Grama, A. Gupta, and G. Karypis, *Introduction to parallel computing: design and analysis of algorithms.* Benjamin/Cummings Publishing Company Redwood City, CA, 1994. 45

[111] A. O'Dwyer, *Handbook of PI and PID controller tuning rules.* World Scientific, 2006, vol. 2. 49

[112] P. Persson and K. Astrom, "Dominant pole design-a unified view of pid controller tuning," in *IFAC SYMPOSIA SERIES.* PERGAMON PRESS, 1993, pp. 377–377. 49

[113] N. J. Killingsworth and M. Krstic, "Pid tuning using extremum seeking: online, model-free performance optimization," *Control Systems, IEEE*, vol. 26, no. 1, pp. 70–79, 2006. 49

[114] K. H. Ang, G. Chong, and Y. Li, "Pid control system analysis, design, and technology," *Control Systems Technology, IEEE Transactions on*, vol. 13, no. 4, pp. 559–576, 2005. 49

Vita

Kumaraguru Prabakar was born in India, to the parents of Thanakumari and Prabakar. He obtained his Bachelors degree from SRM University, India and his Masters degree from Arizona State University in 2009 and 2011. He joined University of Tennessee to pursue his doctoral studies in 2011.