OPTIMAL FRACTIONAL ORDER PROPORTIONAL AND INTEGRAL CONTROLLER FOR PROCESSES WITH RANDOM TIME DELAYS

by

Varsha Bhambhani

A thesis submitted in partial fulfillment
of the requirements for the degree

of

MASTER OF SCIENCE

in

Electrical Engineering

Approved:

Dr. YangQuan Chen Major Professor	Dr. Wei Ren Committee Member
Dr. Edmund Spencer Committee Member	Dr. Byron R. Burnham Dean of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2008

Abstract

Optimal Fractional Order Proportional and Integral Controller for Processes with Random Time Delays

by

Varsha Bhambhani, Master of Science

Utah State University, 2008

Major Professor: Dr. YangQuan Chen
Department: Electrical and Computer Engineering

This thesis developed a new practical tuning method for fractional order proportional and integral controllers (FO-PI / PI$^{\alpha}$) for varying time-delay systems like networked control systems (NCS), sensor networks, etc. Based on previously proposed FO-PI controller tuning rules using fractional Ms constrained integral gain optimization (F-MIGO), simultaneous maximization of the jitter margin and integrated time weighted absolute error (ITAE) performance for a set of hundred gain delay time-constant (KLT) systems having different time-constants and time-delay values are achieved. A multi-objective optimization algorithm is used to simultaneously maximize the ITAE factor and jitter margin of the plants at initial F-MIGO gain parameters. The new values of controller gain parameters are generalized to give a new set of optimal fractional order proportional integral (OFOPI) tuning rules such that the jitter margin and system performance of closed-loop KLT systems are maximized and yet the closed-loop feedback system is stable. This is further tested and verified by simulation techniques. Comparisons are made with other existing proportional integral derivative (PID) and fractional order proportional integral (PI) tuning rules to prove the efficiency of the new technique. It is further shown that OFOPI tuning rules perform better than traditional tuning methods for lag-dominated FOPDT systems, because it can take the varying time-delay better into account. The tuning method is modified to work with discrete-time controllers in the context of NCSs. Furthermore, experimental results in a NCS platform, Stand-alone Smart Wheel (omnidirectional networked control robot wheel), are reported using the tuning rules developed in this thesis. The optimization tuning method performed almost equally well in practice as in simulations. The thesis also shows that the tuning rule development procedure for OFOPI is not only valid for FOPDT systems but is also applicable for other general classes of plants which could be reduced to first order plant systems. Temperature control in heat flow apparatus and water-level control in a coupled tank system using FO-PI tuning rules are other major contributions of this thesis work.

(105 pages)

To my family and friends.

Acknowledgments

I wish to express my deep gratitude and sincere thanks to everyone who contributed to making the dissertation “Optimal Fractional Order Proportional and Integral Controller for Processes with Random Time Delays” a reality. I must single out my advisor, Dr. YangQuan Chen (associate professor, Electrical and Computer Engineering Department, USU, Logan, UT), whose valuable suggestions, guidance, and training were instrumental in creating this document. I am highly obliged to him for the time he spent in leading, supporting, and encouraging me, which helped me to shape my thoughts in the right direction not only on this project, but also for my future endeavor. His knowledge, wisdom, guidance, patience, passion for challenges, perseverance in research, and constant encouragement have boosted up my morale and given me the competence in the subject of controls. His thoughtful insights and endless efforts, sharp perceptiveness, and penetrating questions have always showed me the right direction at the right time. For this and much more, I will forever be indebted to him.

I would like to extend my sincere thanks to my associate advisors, Dr. Wei Ren (assistant professor, Electrical and Computer Engineering Department, USU, Logan, UT) and Dr. Edmund Spencer (assistant professor, Electrical and Computer Engineering Department, USU, Logan, UT) for their timely help and encouragement. Their happy faces and smiles have always brightened my spirits. I am highly obliged to them for sharing their knowledge and wisdom. Help from Dr. Don Cripps is greatly appreciated.

I would like to thank my friends for their support and valuable suggestions. Special thanks go to my colleagues and friends, Shayok Mukhopadhyay, Calvin Coopmans, and Yiding Han, for their immense help in the lab (especially with the Smart Wheel project). All the time spent with them will be cherished for a long time. Efforts, cooperation, and help from other important members of CSOIS, Yan Li, Ying Luo, Lizabeth Lee, Haiyang Chao, Christophe Tricaud, Shelley Rounds, Daniel Scott Stuart, Austin Jensen, and Yongcan Cao

(in no particular order), are gratefully acknowledged. Their wonderful company helped maintain a jovial atmosphere in the lab while doing serious research at the same time.

I am grateful to my friends and family members who have been always been my supporters. I am thankful to them for their sacrifices. First, sincere and genuine thanks to my brother, Dr. Akhilesh Bhambhani, and to Melissa Peterson and Sravanthi Venigalla for their immense patience, support, and listening to the control engineering, despite having no actual interest in the subject. I am indebted to my family for being a constant source of inspiration. I admire the cooperation and understanding of my brother Mr. Kamal K. Bhambhani, my sister-in-law Mrs. Priya Bhambhani, and Laksh. My parents, Mr. K. L. Bhambhani and Mrs. Rashmi Bhambhani, taught me the importance of hard work and being faithful on all the endeavors, and it is to them that I dedicate this dissertation.

Varsha Bhambhani

Contents

Page

Abstract . . . . **iii**

Acknowledgments . . . . **vi**

List of Tables . . . . **x**

List of Figures . . . . **xi**

Acronyms . . . . **xiv**

1 Introduction **1**
1.1 Motivation 1
1.2 Contribution 2
1.3 Outline of the Thesis 3

2 Fractional Order Proportional Integral (FO-PI) Controller and Coupled Tank Experiments **5**
2.1 Introduction 5
2.2 Basics 6
2.2.1 Controller Design and Practical Tuning Rules 6
2.2.2 The Coupled Tank System (CTS) PP-100 7
2.2.3 Hardware-in-the-loop (HIL) and Real-time Control 8
2.3 Experimental Study on the Coupled Tank System (CTS) PP-100 10
2.3.1 Case I: The First Order SISO CTS Configuration 10
2.3.2 Case II: The Second Order SISO CTS Configuration 15
2.3.3 Case III: The Cascaded Control CTS Configuration 21
2.4 Conclusion 26

3 Optimal Fractional Order Proportional and Integral (OFOPI) Tuning Rules for Varying Time-delay Systems **27**
3.1 Introduction 27
3.2 Preliminaries 28
3.2.1 Multi-objective Optimization Problem 28
3.2.2 Optimization Criteria 29
3.3 Test Bench Simulation and Optimization 31
3.4 Optimal FO-PI Tuning and Verification 34
3.5 Comparison between OFOPI and OPID Controllers 37
3.6 Conclusion and Future Work 39

4 Networked Control of Smart Wheel using OFOPI Tuning Rules 42

4.1 Introduction 42

4.2 Hardware Description 43

4.2.1 Mechanical Hardware Architecture 43

4.2.2 Wheel-node Electronic Hardware Architecture 45

4.2.3 NPort DE-311 1-Port Serial Server 45

4.2.4 DCS-5300 Internet Camera 46

4.3 Software Description 47

4.3.1 Software Architecture Overview 48

4.3.2 Arbitration Mechanism 49

4.4 Smart Wheel Networked Steering Speed Control 51

4.4.1 System Identification using Step Response Analysis 52

4.4.2 Laboratory Set up and Hardware-in-the-loop Control 54

4.4.3 Simulation Illustration 56

4.4.4 Experiment on the Real-time Smart Wheel Speed Control System . 58

4.5 Conclusion 59

5 Fractional-order Integral and Derivative Controller Design for Temperature Profile Control 60

5.1 Introduction 60

5.2 Temperature Control 61

5.3 Fractional Order $I^{\alpha}D^{\beta}$ Control 63

5.3.1 Fractional Order Calculus 63

5.3.2 Merit of Using Fractional Order Controller 64

5.4 Analytical Tuning Method 65

5.5 Experimental Test 68

5.6 Comparisons and Simulation Results 73

5.7 Conclusions 76

6 Conclusion 78

6.1 Summary 78

6.2 Future Work 79

References 81

Appendices 86

Appendix A Optimal PID Tuning Rules 87

Appendix B Setpoint Conversion Formulae 89

Appendix C Code 90

List of Tables

Table Page

2.1 Frequency response of the first order SISO CTS. 12

2.2 Controller gain parameters for water-level control in first order SISO CTS. . 13

2.3 Frequency response of the second order SISO CTS configuration. 17

2.4 Controller tuning parameters for the second order SISO CTS configuration. 19

2.5 PI controller gain parameters for cascade control configuration. 24

4.1 Gain parameters for different controllers. 56

List of Figures

Figure Page

2.1 Front and back sides of the coupled tank system (CTS) pp-100 at CSOIS. . 8

2.2 Schematic of the real-time hardware-in-loop. 8

2.3 Feedback control of the coupled tank system (CTS) PP-100. 9

2.4 Data acquisition control board (DACB) from Quanser. 9

2.5 System model for first order SISO coupled tank system (CTS) PP-100. . . . 10

2.6 Frequency response of the first order SISO CTS. 12

2.7 Simulink block diagram of water-level control in the first order SISO CTS configuration. 13

2.8 Simulated control of water-level in first order SISO configuration. 14

2.9 Real-time control of water-level in the first order SISO CTS configuration. . 15

2.10 System model for the second order SISO CTS configuration. 15

2.11 Frequency response of the second order SISO CTS configuration. 17

2.12 Comparison of the actual second order CTS and the approximated first order CTS configuration. 19

2.13 Simulink block diagram of water-level control in the second order SISO CTS configuration. 20

2.14 Simulated control of water-level in the second order SISO CTS configuration. 20

2.15 Real-time control of water-level in the second order SISO CTS configuration. 21

2.16 System model for the cascade control CTS configuration. 22

2.17 Open-loop step response of two cascaded tanks. 22

2.18 Comparison of the actual cascaded CTS and the approximated first order CTS. 24

2.19 Simulink block diagram of water-level control in the cascaded control configuration. 25

2.20 Simulated control of water-level in the cascaded control configuration. . . . 25

2.21 Real-time control of water-level in the cascaded control configuration. . . . 26

3.1 Block diagram of closed-loop system with delay. 29

3.2 Block diagram of the equivalently transformed system. 30

3.3 Improvement of jitter margin of KLT systems after optimization of FO-PI controller. 34

3.4 Performance of KLT systems before and after optimization of FOPI controller. 34

3.5 Step response of lag dominated system $(K = 1, L = 2, T = 8)$ at different delays. 35

3.6 Step response of an intermediate system $(K = 1, L = 8, T = 8)$ at different delays. 36

3.7 Step response of delay dominated system $(K = 1, L = 8, T = 2)$ at different delays. 36

3.8 Jitter margin of KLT systems controlled by OPID and OFOPI controllers. . 38

3.9 Performance of KLT systems controlled by OPID and OFOPI controllers . 38

3.10 Step response of lag dominated system $(K = 1, L = 2, T = 8)$ at different delays. 39

3.11 Step response of an intermediate system $(K = 1, L = 8, T = 8)$ at different delays. 40

3.12 Step response of delay dominated system $(K = 1, L = 8, T = 2)$ at different delays. 40

4.1 Stand-alone enhanced Smart Wheel assembly at CSOIS. 43

4.2 The mechanical hardware architecture of Smart Wheel demonstrator. . . . 44

4.3 The wheel node electronic hardware architecture. 45

4.4 Interfacing details of the Neteon Net NPort DE-311 1-Port serial server. . . 46

4.5 The DCS-5300 Internet camera from DLink. 46

4.6 Software architecture of stand-alone Smart Wheel demonstration system. . 47

4.7 Flowchart describing working mechanism in Smart Wheel. 49

4.8 Successful telepresence control session. 50

4.9 Session termination when idle for 30 seconds. 51

4.10 Session denied due to multiple operators. 51

4.11 Step response analysis of Smart Wheel. 53

4.12 Laboratory setup and hardware-in-the-loop. 55

4.13 Flowchart describing working of Smart Wheel. 55

4.14 Simulated network control of steering speed of Smart Wheel : China. 57

4.15 Simulated network control of steering speed of Smart Wheel : Australia. . . 57

4.16 Real-time network control of steering speed of Smart Wheel: China. 58

4.17 Real-time network control of steering speed of Smart Wheel: Australia. . . 59

5.1 The set of equipment for heat flow test. 68

5.2 Sensor 1 (Fan Command Voltage constant= 5V, Heat Command Voltage varying). 69

5.3 Sensor2 (Fan Command Voltage constant= 5V, Heat Command Voltage varying). 69

5.4 Sensor3 (Fan Command Voltage constant= 5V, Heat Command Voltage varying). 69

5.5 Step response data from the open-loop heat flow experiment. 71

5.6 Heating effect: Variation of static gain of sensors with distance. 71

5.7 Heating effect: Variation of delay of sensors with distance. 71

5.8 Heating effect: Time constant of sensors with distance. 72

5.9 Cooling effect: Variation of static gain of sensors with distance. 72

5.10 Cooling effect: Variation of delay of sensors with distance. 72

5.11 Cooling effect: Variation of time constant of sensors with distance. 73

5.12 Controller performances for sensor 1 in real-time. 74

5.13 Controller performances for sensor 2 in real-time. 75

5.14 Controller performances for sensor 3 in real-time. 75

5.15 Controller performances for sensor 1 in Simulink. 75

5.16 Controller performances for sensor 2 in Simulink. 76

5.17 Controller performances for sensor 3 in Simulink. 76

Acronyms

FOPDT	first order plus time delay systems
OFOPI	optimal fractional order proportional integral
CSOIS	Center for Self-Organizing and Intelligent Systems
OPID	optimal proportional integral and derivative
NCS	networked control system
CTS	coupled tank system
HFE	heat flow experiment
JM	jitter margin
ITAE	integrated time weighted absolute error
F-MIGO	fractional Ms constrained integral gain optimization
FOC	fractional order control
HIL	hardware-in-the-loop
PID	proportional integral and derivative
FO-ID	fractional order integral and derivative
ZN	ziegler nichol
MZN	modified ziegler nichol
PWM	pulse width modulation
DACB	data acquisition and control board
DAC	digital-to-analog converter
LLC	low-level control
UART	universal asynchronous receiver/transmitter
ASCII	american standard code for information interchange

Chapter 1
Introduction

1.1 Motivation

Growth and development of Internet combined with the advantages provided by networked control system (NCS) has been of great interest to researchers around the globe. Along with the advantages, several challenges also emerged. This triggered research in fields of new control strategies, reliability and security of communications, bandwidth allocation, development of data communication protocols, kinematics of the actuators in the systems, real-time information collection, and efficient processing of sensors data to list a few.

Additional time delays in control loops or possibility of packages loss result due to the insertion of the communication network in the feedback control loop which makes the analysis and design of an NCS complex. Time-delays could impose severe degradation on the system performance and can even make the system unstable. Many researchers have provided solutions using concepts from several control areas such as robust control, optimal stochastic control, fuzzy logic, etc., to eleviate the time-delay effects. Extensive simulation results on how the jitter in the loop can degrade system performance and lead to instability of system can be found in dissertation works, e.g. P. Marti and A. Cervin [1, 2]. Ensuring the stability of systems with varying time-delays has always been an interesting area of research for control engineers like Wu et al., Phat et al., and Kao et al. [3–5].

A controller which could ensure the stability of the plant at large values of network delays will revolutionize the Internet. This thesis work introduces a new jitter-robust controller design by optimizing the gain parameters of fractional order proportional integral (FO-PI) controller based on fractional M_s constrained integral gain optimization (F-MIGO) algorithm [6–9]. This controller design is helpful in finding the maximum value of jitter (variance in time-delay) at which the system remains stable. The integral of time weighted absolute error (ITAE) performance of the proposed controller is better than the best integer order proportional integral derivative (PID) controller.

1.2 Contribution

As mentioned in sec. 1.1, a fractional order controller is designed by optimizing the gain parameters of fractional order proportional integral (FO-PI) controller based on fractional M_s constrained integral gain optimization (F-MIGO) algorithm which ensures stability at large values of delays (jitter margin). For this a test batch of hundred first order plus delay time (FOPDT) systems is considered and FO-PI tuning rules are used to compute the proportional gain, integral gain, and non-integer order α of the integrator for each system. The gain and α values so obtained for each system are then used to compute the integrated time weighted absolute error (ITAE) performance and jitter margin. Finally, multi-objective optimization algorithm is applied to optimize these gains and α values for each system. Jitter margin and ITAE value are calculated at these optimum values as in work of Eriksson et al. [8, 9]. The new gain parameter values and fractional order for the set of plants are then generalized to obtain the optimal fractional order proportional integral (OFOPI) tuning rules. Simulation results show that new OFOPI tuning rules have larger jitter margin and better performance for lag-dominated systems when compared to proportional integral derivation (PID) controllers for varying time delays.

In simple words, the main contribution of this work lies in answering this research question: "*Will a fractional order controller help and do better?*" when there are uncertainties in the time delay due to e.g., jitter in the loop.

Furthermore, experimental results in a NCS platform: Stand-alone Smart Wheel are reported using the tuning rules developed in this thesis. This involves system identification of the plant, implementation of s-function to enable communication and control using a hardware-in-the-loop, simulations and real-time experiments. The optimization tuning method performed almost equally well in practice as in simulations.

Also, a novel analytical tuning method for a fractional order integral derivative ($FO-ID/I^{\alpha}D^{\beta}$) controller is introduced and the results are applied to analyze heat flow in heat flow equipment (HFE) module. The performance of new FO-ID controller is compared to the integer order PI/PID controllers and FO-PI controller. Extensive simulation results are included to illustrate the simple yet practical nature of the developed new tuning rules. Besides experiments are also performed for water level control in the coupled tank PP-100 using FO-PI tuning rules.

1.3 Outline of the Thesis

Chapter 2 reviews the FO-PI tuning rules based on F-MIGO algorithm [10], which will be used as starting point for obtaining the jitter margin and ITAE values of the FOPDT systems in this thesis. Furthermore to illustrate the significance of FO-PI controller, we apply them to the coupled tank fluid level controls. Also, the FO-PI controller is compared with both Ziegler Nichol's (ZN) tuned and Modified Ziegler Nichol's (MZN) tuned conventional integer order PI controllers in terms of load disturbance rejection, changes in plant dynamics, and setpoint tracking.

Chapter 3 aims at development of OFOPI tuning rules using multi-objective optimization. This involves introduction to basic terms like jitter margin, ITAE performance index and the multi-objective optimization method. A section focuses on optimal tuning of FO-PI controller followed by another section on approximation of optimized proportional gain parameter (K_p), integral gain parameter (K_i), and fractional order (α) to get new set of optimal FO-PI tuning rules that ensures the best jitter margin and ITAE performance. Simulation results are provided followed by final section which concludes this chapter with remarks on future research work.

In Chapter 4, experimental results in a NCS platform: Stand-alone Smart Wheel are reported using the tuning rules developed in this thesis. The chapter focuses on the hardware and software architecture description of the Smart Wheel system and networked steering speed control of Smart Wheel using fractional and integer order controller tuning rules at random delays. This involves system identification of Smart Wheel plant, laboratory setup for the networked control system design for the steering axis of the Smart Wheel using hardware-in-the-loop speed control, simulation techniques, real-time experimental results and conclusions.

Chapter 5 is devoted to work on temperature control using fractional order controllers. It introduces a new control scheme for temperature regulation using fractional-order integral derivative ($FO - ID/I^{\alpha}D^{\beta}$) controller for a more accurate temperature profile tracking of the spatially distributed heat flow. An actual experimental task is performed using the heat flow equipment (HFE) from Quanser. This involves in depth description and derivation of proposed analytical tuning method, real-time control and hardware-in-the-loop method, detailed descriptions of the HFE platform, system analysis, and design of a fractional controller based on the analytical tuning approach, an extensive comparison of existing integer order solutions and fractional order solutions, and some remarks on the achieved results and ideas for future work.

Chapter 6 summarizes the thesis work on fractional order controllers for processes with random time delays and also lists the scope for future work in this regards.

Chapter 2

Fractional Order Proportional Integral (FO-PI) Controller and Coupled Tank Experiments

2.1 Introduction

The past decade has witnessed an increase in research efforts related to fractional calculus [11, 12] and its applications to control theory. The fractional order controller is more common in control practice, because the plant model may have already been obtained as an integer order model in the classical sense. Improving or optimizing performance as seen in work of Xue et al. [13] is the major concern from an engineering point of view. Thus, our objective is to apply the fractional order control (FOC) to enhance the integer order dynamic system control performance. Pioneering works in applying fractional calculus in dynamic systems and controls and the recent developments can be found in research works of Manabe, Oustaloup, and Axtell et al. [14–17].

This chapter reviews the fractional order proportional integral (FO-PI) tuning rules based on fractional Ms constrained integral gain optimization (F-MIGO) algorithm [10], which will be used as starting point for obtaining the jitter margin (variance in delay) and integrated time weighted absolute error (ITAE) values of the first order plus delay time (FOPDT) systems in this thesis. Furthermore to illustrate the significance of the FO-PI controller, we apply them to the coupled tank fluid level controls. This is a four-step process consisting of mathematical modeling of the system plant, identification of system plant parameters, water level controller design, and comparisons in simulink [s/w mode] and experimental verification in real time [h/w mode]. Also, the FO-PI controller is compared with both Ziegler Nichol's (ZN) tuned and Modified Ziegler Nichol's (MZN) tuned conventional integer order proportional integral (PI) controllers in terms of load disturbance rejection, changes in plant dynamics, and setpoint tracking. Experimental results prove that the FO-PI is a promising controller in terms of percentage overshoot and system response in liquid level and flow control.

Section 2.2 explains the basics needed to understand the problem and presents the tuning methods used. This involves reviewing FO-PI controller tuning methods, the construction and working of KRi control apparatus coupled tank system (CTS) PP$-$100, and concepts of real-time control and hardware-in-the-loop (HIL). Intensive case studies of experiments on the coupled tank in different configurations are provided in sec. 2.3. It is subdivided into three main sections: first order single input single output (SISO), second order SISO system, and cascade control plant system, respectively. Each subsection involves a brief description of the system modeling, real-time system identification, controller design, experiments, and simulations. Section 2.4 summarizes the results of the experimental study on the coupled tank system.

2.2 Basics

2.2.1 Controller Design and Practical Tuning Rules

Immense work on fractional controllers can be found in literature [11, 13, 18–20]. In time-domain, if $r(t)$ is the setpoint signal, and $u(t)$ is the control input, and $y(t)$ is the output, the FO-PI controller is represented by (2.1) as:

$$u(t) = K_p(r(t) - y(t)) + K_i D_t^{-\alpha}(r(t) - y(t)), \tag{2.1}$$

where $D_t^{\alpha}(x)$ is the fractional differointegral operator.

The fractional derivative of order α of function $f(t)$ can be defined as in works of Bhaskaran et al. [7, 21]:

$$\frac{\mathrm{d}^{\alpha}}{\mathrm{d}t^{\alpha}} f(t) \doteq \begin{cases} f^{(n)}(t) \quad \text{if } \alpha = n \in N \\ \frac{t^{n-\alpha-1}}{\Gamma(n-\alpha)} * f^{(n)}(t) \quad \text{if } n-1 < \alpha < n \end{cases}, \tag{2.2}$$

where $*$ denotes the time convolution between two functions.

In frequency-domain, the FO-PI controller $C(s)$ is expressed as:

$$C(s) = K_p + \frac{K_i}{s^{\alpha}}, \tag{2.3}$$

where K_p and K_i are the proportional and integral gain values of the fractional controller and α is the fractional order of the integrator. Furthermore, work of Bhaskaran et al. [21] explains how to tune the gains K_p, K_i, and the non-integer order α whereas in Bohannan et al. [7] experimentally validates the tuning rules. The tuning rules developed in Bhaskaran et al. [21] are restated as:

$$K_p = \frac{0.2978}{K(\tau + 0.000307)}, \tag{2.4}$$

$$K_i = \frac{K_p(\tau^2 - 3.402\tau + 2.405)}{0.8578T},$$

$$\alpha = \begin{cases} 0.7, & \text{if } \tau < 0.1 \\ 0.9, & \text{if } 0.1 \le \tau < 0.4 \\ 1.0, & \text{if } 0.4 \le \tau < 0.6 \\ 1.1, & \text{if} \tau \ge 0.6 \end{cases}.$$

These FO-PI tuning rules are based on the Fractional M_s constrained Integral Gain Optimization (F-MIGO) method and further explanation can be found in Bhaskaran et al. [21] and Bohannan et al. [7].

2.2.2 The Coupled Tank System (CTS) PP-100

The back and front sides of the coupled tank system PP-100 at Center of Self Organizing and Intelligent Systems (CSOIS) are shown in fig. 2.1. The coupled tank system (CTS) PP-100 is a compact bench top instrument made of two water tanks made of perspex placed on a water reservoir [22–24]. The reservoir has two pulse width modulated (PWM) operated motor pumps. These pumps use either $0 - 5$ V analog voltage or external PWM sources for their operation. Internal signal conditioning system converts analog to PWM digital

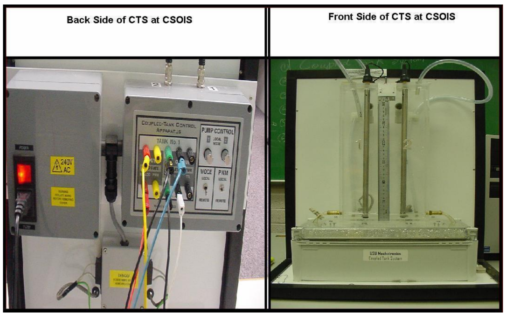


Fig. 2.1: Front and back sides of the coupled tank system (CTS) pp-100 at CSOIS.

signals. Water flow rates into tanks vary according to change in these pump voltages. To vary interaction or coupling dynamics between the two tanks, there is a baffle plate separating the two tanks which can be slided up and down. Two capacitive probes, one in each tank, measure water-level. Output signals from these probes are conditioned to give $0-5$ V DC analog output. A water outlet on the side near the base of each tank, connected by a flexible tube, returns water to the reservoir. Two potentiometers on the back side of the CTS are provided for manual operation of the motors.

2.2.3 Hardware-in-the-loop (HIL) and Real-time Control

Figure 2.2 shows the schematic of the real-time,hardware-in-the-loop configuration provided by Quanser. A typical feedback control system is shown in fig. 2.3. It consists of the plant, sensors, actuators, and a controller. Furthermore, a data acquisition and control board (DACB) provided by Quanser [25] provides feedback to the digital controller (computer) from the plant in the appropriate format. The HIL multi-$Q4$ board shown in fig. 2.4 has four analog inputs, four voltage outputs, four quadrature encoder inputs, 16 programmable digital input/output (I/O) channels, two counters/timers, and several other

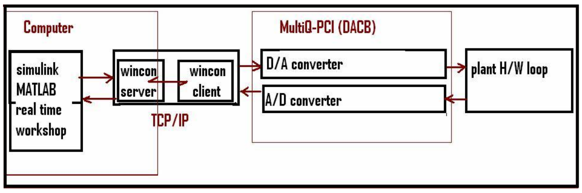


Fig. 2.2: Schematic of the real-time hardware-in-loop.

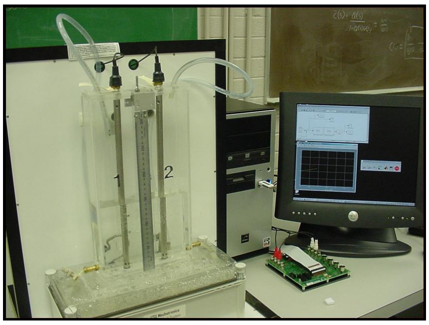


Fig. 2.3: Feedback control of the coupled tank system (CTS) PP-100.

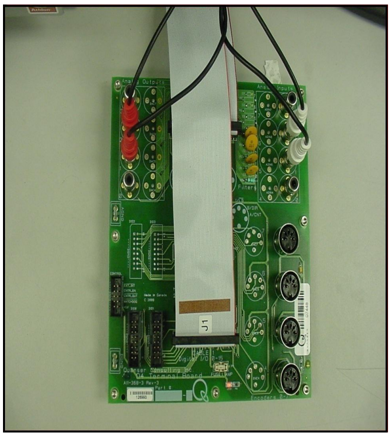


Fig. 2.4: Data acquisition control board (DACB) from Quanser.

useful features. Analog sensor signals from the plant are converted to digital mode and then sent back to the computer by these HIL boards. The next set of control signals, obtained as a result of user program implementation on data from the sensor, sends digital data to the DACB, which converts it to an analog signal, sent to the actuators. WinCon implements this in real-time. WinCon is a real-time Windows2000/XP application that runs in real-time the C code generated for the control law implemented in MATLAB/Simulink Real Time workshop. It consists of two basic parts: WinCon Client, the real component of the software that runs at a period specified by the user, and WinCon Server, that interfaces with the DACB unit and provides a graphical user interface for the user to record the signals returned from the sensors.

2.3 Experimental Study on the Coupled Tank System (CTS) PP-100

There are three subsections, each focusing on a specific coupled tank configuration. The three configurations of the CTS, which will be considered in following subsections are as follows:

- Case I - First order SISO CTS,
- Case II - Second order SISO CTS,
- Case III - Cascaded control CTS.

In-depth explanation of how system identification of the CTS is accomplished using frequency response or step response in each subsection. It also explains how the controller is designed and gain parameters computed using the tuning rules. This is done in real-time and then compared with simulation results.

2.3.1 Case I: The First Order SISO CTS Configuration

When the baffle plate is lowered completely, the two tanks operate independently as first order single input single output (SISO) systems.

Mathematical Modeling of the System Plant

The system model for the first order SISO coupled tank system (CTS) is shown in fig. 2.5 which is similar to model shown in work of Laubwald [26]. Here, the relation

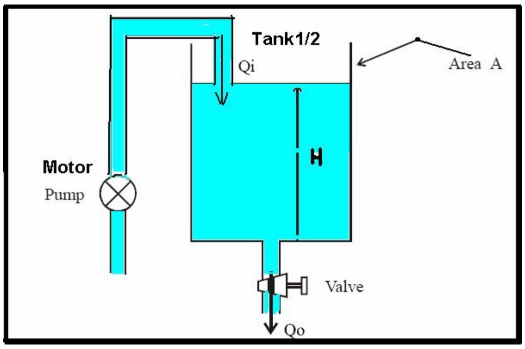


Fig. 2.5: System model for first order SISO coupled tank system (CTS) PP-100.

between water entering and leaving tank is expressed as:

$$Q_i - Q_o = A\frac{dH}{dt}, \tag{2.5}$$

where Q_i is the rate of water flow in the tank, Q_o is the rate of water flow out of the tank, A is the cross-sectional area of the tank, and H is the height of water in tank. Substituting the value of Q_o in (2.5), we get:

$$Q_i - Ca\sqrt{2gH} = A\frac{dH}{dt}, \tag{2.6}$$

where C is the discharge coefficient of the outlet valve, a is the cross-sectional area of orifice, and g is the gravitational constant equal to 9.8 m$/s^2$. The nonlinear equation above describes the system behavior of first order SISO coupled tank system.

Identification of the System Plant Parameters

In terms of transfer function, in real-time, the manipulated variable/plant input is the pump input voltage and the process variable/plant output is the water-level in the tank [26–28]. The transfer function of first order SISO system is given by (2.7).

$$G(s) = \frac{K}{Ts+1}e^{-Ls} \tag{2.7}$$

System parameters, such as the time constant and the gain, can be found using the step response or frequency response (bode plot). In the present case we consider frequency response analysis due to reliability and accuracy of method. Frequency response is the measure of any system's spectrum response at the output to a signal of varying frequency (but constant amplitude) at its input. Different frequencies are considered and gain in decibels and phase shift in degrees of the respective sinusoidal output is noted at steady state. The results of the frequency response are shown in fig. 2.6 and summarized in the form of Table 2.1.

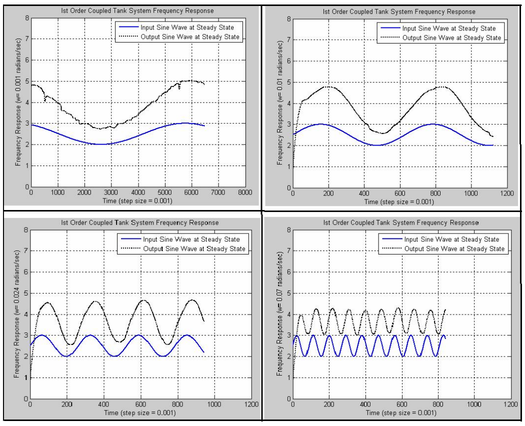


Fig. 2.6: Frequency response of the first order SISO CTS.

System parameters are computed from the frequency response data as:

$$K = 10^{\frac{M(\omega_{min})}{20}} = 10^{\frac{7.224}{20}} = 2.2972, \tag{2.8}$$

$$T = \frac{1}{\omega(-45)} = \frac{1}{0.024} = 41.667sec, \tag{2.9}$$

$$L = 1.75sec, \tag{2.10}$$

where

- K is the DC gain of the plant system,
- $M(\omega_{min})$ is the gain at minimum frequency,
- T is the time constant,
- $\omega(-45)$ is the frequency when the phase plot is -45^o,

Table 2.1: Frequency response of the first order SISO CTS.

Frequency (rad/sec)	Magnitude (decibels)	Angle (degrees)
0.001	7.224	-0
0.005	6.7408	-8
0.024	6.4208	-45
0.07	1.8852	-80

- L is the time lag of the plant system.

Substituting values of K, T, and L obtained experimentally as described above in the transfer function of first order system, we identify our first order SISO CTS as:

$$G = \frac{2.2972}{41.667s + 1} e^{-1.75s}. \tag{2.11}$$

Water-Level Controller Design and Comparison in Simulink [S/W Mode]

The next step in level control are controller design and tuning. Table 2.2 shows the gain parameters for different controllers that control water-level and fluid flow for single order SISO configuration. The three controller tuning methods compared are FO-PI/F-MIGO, PI/ZN and PI/MZN. The simulation block diagram of water-level control in the first order CTS is shown in fig. 2.7. In fig. 2.8 is shown the result of a simulation of different controllers performing water-level control in tank system.

Table 2.2: Controller gain parameters for water-level control in first order SISO CTS.

	K_p	T_i	α
PI-ZN	6.6245	5.5071	
PI-MZN	1.4641	6.9443	
FOPI-FMIGO	3.1919	15.7488	0.7

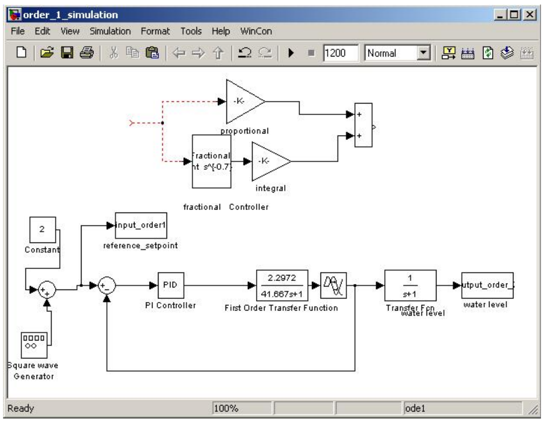


Fig. 2.7: Simulink block diagram of water-level control in the first order SISO CTS configuration.

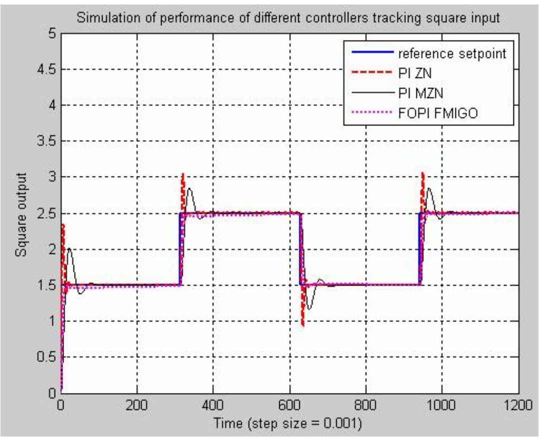


Fig. 2.8: Simulated control of water-level in first order SISO configuration.

As can be seen from the simulation results, PI/ZN accounts for the largest amount of overshoot, but has a fast response. PI/MZN has a relatively less percentage overshoot whereas the new controller, i.e., the FO-PI/F-MIGO controller, results in a much less percentage overshoot. Percentage overshoot is defined as the maximum fraction by which the response overshoots the steady state value expressed as a percentage. In general, quick response and small overshoot are desirable in most of the target-tracking control problems. However, it is well-known that quick response results in a large overshoot. Thus, most of the design schemes must make a trade-off between these two transient performance indices. In fact, from the simulation results, it can be said that the FO-PI/F-MIGO controller is a promising controller in fluid level and flow control in tanks.

Experimental Verification in Real-Time [H/W Mode]

The final step in the design process of any control system is the control experiment. For control plants like inverted pendulum, CTS, robots, etc., real-time control is required. Furthermore, as can be seen from fig. 2.9, the controller performances confirm the simulated results. However, a small deviation from the simulated results can be accounted for due to uncontrollable real-time environmental disturbances such as dynamics of fluid, modeling uncertainties, etc.

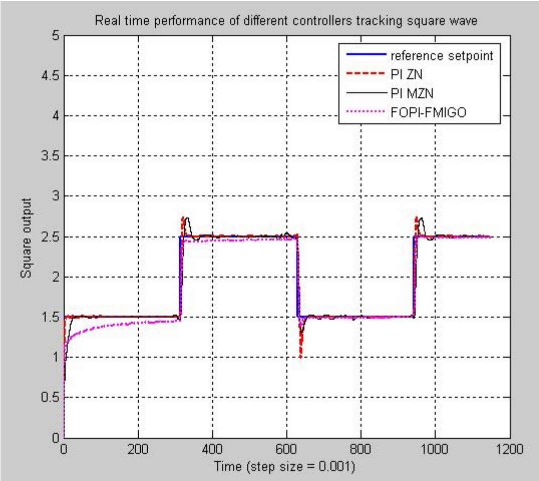


Fig. 2.9: Real-time control of water-level in the first order SISO CTS configuration.

2.3.2 Case II: The Second Order SISO CTS Configuration

A second order SISO CTS can be obtained by raising the baffle plate so that water flows from one tank to another as shown in help manuals on coupled tank system by KRi [26, 27, 29]. The aim of this experiment is to design a controller that maintains a fixed water-level in tank 2 by varying voltage input to the motor of tank 1.

Mathematical Modeling of the System Plant

Figure 2.10 shows the system model for second order SISO coupled tank system. For

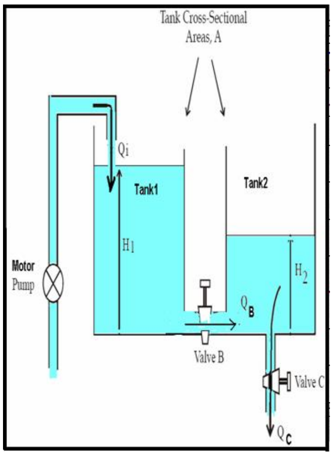


Fig. 2.10: System model for the second order SISO CTS configuration.

a second order SISO system, there are two states, the water-levels in the two tanks, i.e. H_1 in the tank 1 and H_2 in the tank 2. The control input to the system is the rate of pump flow to the tank 1, represented as Q_i, and the output is the water-level H_2 in the tank 2. Additionally, there are two valves for the flow of water. Valve B allows flow of water between tank 1 and tank 2 and valve C allows fluid out of tank 2. Fluctuation in the rate of water flow through these two valves account for load disturbances. The rate of change of water volume in tank 1 is given by (2.12) as:

$$Q_i - Q_b = A\frac{dH_1}{dt}, \tag{2.12}$$

where Q_b is the rate of flow of water through valve B and A is cross-sectional area of tank 1. Similarly, the water flow balance equation for tank 2 is given by:

$$Q_b - Q_c = A\frac{dH_2}{dt}, \tag{2.13}$$

where Q_c is flow of water out of tank 2 through valve C. Assuming orifices to be ideal, the nonlinearities are computed by square law and substituted in (2.12) and (2.13) to give the following equations:

$$Q_i - C_b a_b \sqrt{2g(H_1 - H2)} = A\frac{dH_1}{dt}, \tag{2.14}$$

$$C_b a_b \sqrt{2g(H_1 - H2)} - C_c a_c \sqrt{2gH_2} = A\frac{dH_1}{dt}.$$

These nonlinear equations describe the system behavior of second order SISO CTS and can be linearized further to obtain state equation of the CTS.

System Plant Parameters Identification

The transfer function for second order CTS is given by (2.15) as:

$$G = \frac{K\omega_n^2}{s^2 + 2\zeta\omega_n + \omega_n^2} = \frac{K}{\tau^2 s^2 + 2\zeta\tau s + 1}, \tag{2.15}$$

where

- K is the process gain of the system plant,
- ζ is the damping ratio and is defined as degree of oscillation in the process response after a perturbation,
- ω_n is the natural frequency of the system and is the inverse of time constant τ which determines the speed of response of the system.

Frequency response analysis is done by feeding sinusoidal signals at different frequencies to the second order coupled tank configuration and recording output at steady state. Different frequencies are considered and gain in decibels and phase shift in degrees of the respective sinusoidal output is noted at steady state. This is shown in fig. 2.11 and results are listed in Table 2.3.

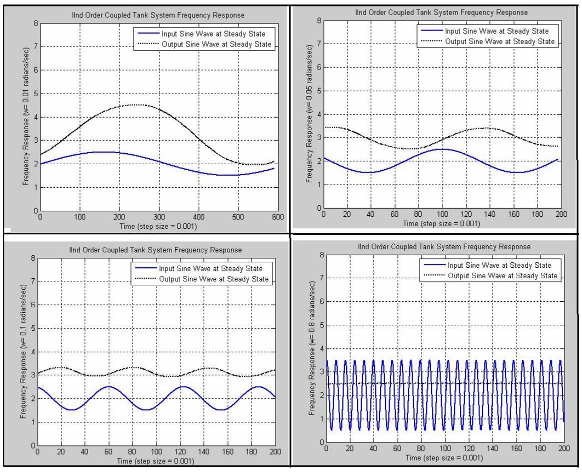


Fig. 2.11: Frequency response of the second order SISO CTS configuration.

Table 2.3: Frequency response of the second order SISO CTS configuration.

Frequency (rad/sec)	Magnitude (decibels)	Angle (degrees)
0.01	6.3	-23.8
0.02	5.3	-48
0.04	1.93	-90
0.05	-0.283	-108
0.1	-10.5	-143
0.8	-30.0063	-180

Second order plant parameters are obtained as:

$$K = 10^{\frac{M(\omega_{min})}{20}} = 10^{\frac{6.3}{20}} = 2.06, \tag{2.16}$$

$$T = \frac{1}{\omega(-90)} = T = \frac{1}{0.04}, \tag{2.17}$$

$$\zeta = \frac{K}{2 * 10^{\frac{M(-90)}{20}}} = \frac{2.06}{2 * 10^{\frac{1.93}{20}}} = 0.824, \tag{2.18}$$

where

- K is the DC Gain of a system,
- $M(\omega_{min})$ is the gain at minimum frequency,
- T is the reciprocal of the natural frequency ω_n,
- $\omega(-90)$ is the frequency at which phase shift is -90^o,
- ζ is the damping ratio,
- $M(-90)$ is magnitude of the bode plot when the phase is -90^o.

Since the value of $\zeta < 1$, it is an underdamped system. Substituting values of K, ω_n and ζ obtained experimentally as described above in the transfer function of the second order system. The second order SISO CTS is identified as:

$$G = \frac{2.06 * 0.04^2}{s^2 + 2 * 0.824 * 0.04 + 0.04^2}. \tag{2.19}$$

Since a first order plant system is required for computing ZN, MZN and F-MIGO tuning parameters, a MATLAB file called "getfod.m" is used to approximate a second order system by first order plant system. The approximated first order SISO system obtained is given by (2.20) as:

$$G = \frac{2.06}{21.1528s + 1} e^{-20.0472s}. \tag{2.20}$$

To further justify how approximate is this new first order transfer function, the step responses of the second order coupled tank and the approximated first order system obtained by “getfod.m” file are plotted as shown in fig. 2.12. The approximated system has larger delay as compared to the actual system. This will be further noticed in real-time second order experiments.

Water-Level Controller Design and Comparison in Simulink [s/w mode]

Based on approximated first order transfer function (2.20) for the second order system, Table 2.4 shows the gain parameters for different controllers that control water-level and fluid flow for the second order SISO CTS configuration. The simulink block diagram for water-level control in the second order CTS is shown in fig. 2.13. Figure 2.14 shows the result of a simulation of different controllers performing water-level control in the tank system.

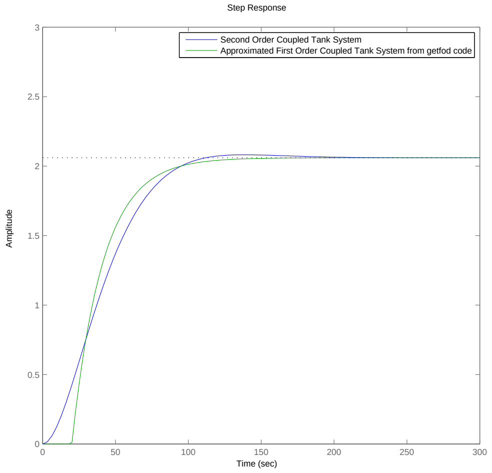


Fig. 2.12: Comparison of the actual second order CTS and the approximated first order CTS configuration.

Table 2.4: Controller tuning parameters for the second order SISO CTS configuration.

	K_p	T_i	α
PI-ZN	0.4556	50.0966	
PI-MZN	0.4692	63.171	
FOPI-FMIGO	0.2969	18.3949	1

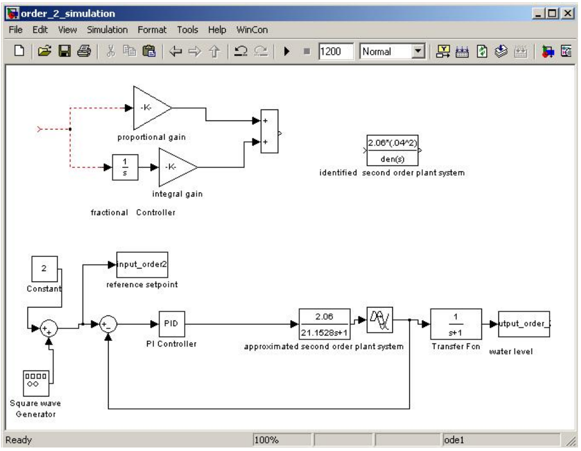


Fig. 2.13: Simulink block diagram of water-level control in the second order SISO CTS configuration.

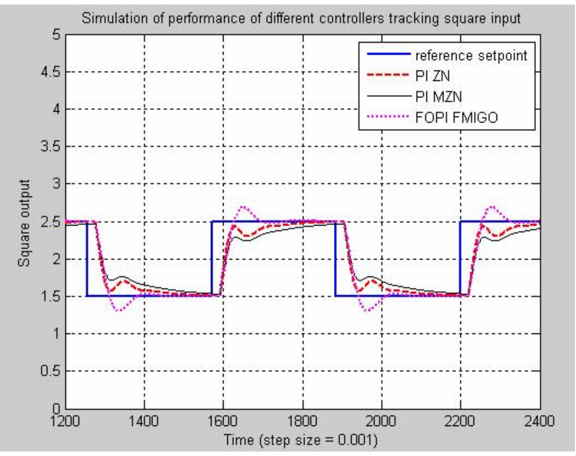


Fig. 2.14: Simulated control of water-level in the second order SISO CTS configuration.

As can be seen from the simulation results, PI/ZN accounts for relatively less overshoot as compared to the fractional controller, but this is on the account of slow system performance. PI/MZN has minimum overshoot and slowest response among the three controller designs. Thus, we see there is a trade off between percentage overshoot and system response.

Experimental Verification in Real-time [H/W Mode]

To validate the simulation results for water-level controller design for the second order system, real-time control system is a must. This section focuses on reproduction of simulation results in real-time. As can be seen from fig. 2.15, the controller performances confirm the simulated results. However, a small deviation from the simulated results does exist due to uncontrollable real-time environmental disturbances such as dynamics of fluid, modeling uncertainties, etc.

2.3.3 Case III: The Cascaded Control CTS Configuration

The baffle plate is raised so that water flows from one tank to another and dynamic coupling takes place between the two tanks. This type of control system has two cascaded controllers namely primary and secondary controllers as shown in help manuals on coupled tank system [26, 27].

Mathematical Modeling of the System Plant

For the cascade control coupled tank configuration, the controlled variable is water flow to tank 1. The master controller decides the setpoint of the slave controller, while the slave controller attempts to track the setpoint. The master controller uses the water-level in tank 2 as the process variable by varying water-levels in tank 1. Suitable baffle opening between the two tanks, introduces significant time separation between the two controllers which minimizes the effect of disturbance in the water-level of tank 1 to the water-level of tank 2. The system model for cascaded control coupled tank is shown in fig. 2.16.

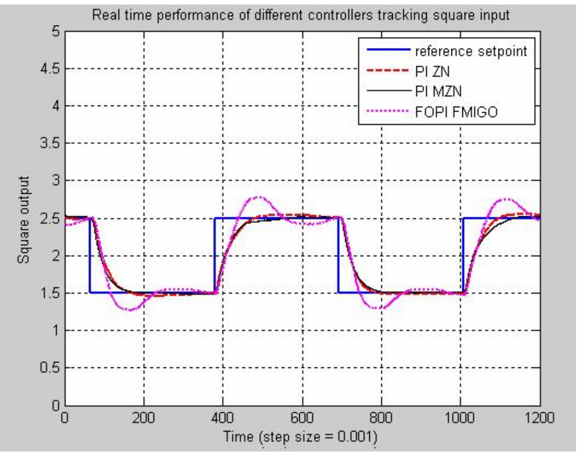


Fig. 2.15: Real-time control of water-level in the second order SISO CTS configuration.

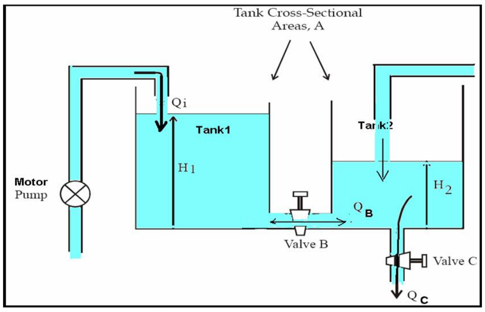


Fig. 2.16: System model for the cascade control CTS configuration.

Identification of System Plant Parameters

Both cascaded plants are configured as first order transfer function; namely, primary and secondary plants having transfer function in general form as:

$$G = \frac{K}{Ts+1}e^{-Ls}, \tag{2.21}$$

where K is the gain of the system, T is the time constant, and L is the delay of the system. One can use either frequency response analysis or step response analysis to identify transfer function. Instead of using frequency response twice for each plant (which is time consuming), one can use step response, in which step input is applied to the plant and the response recorded. This is performed in real-time. The step response for the two tanks so obtained are plotted as shown in fig. 2.17.

This system appears to be a first order system, because the response does not oscillate

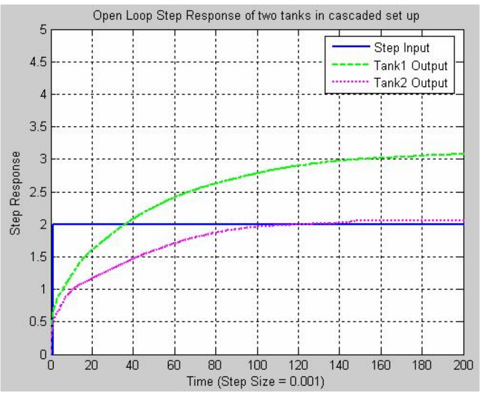


Fig. 2.17: Open-loop step response of two cascaded tanks.

and has a nonzero slope when $t = 0$. For this reason we will model this system as a first order system. The DC gain is the ratio of the steady state step response to the magnitude of a step input. From fig. 2.17, we have:

$$K_1 = \frac{3.14}{2} = 1.57, \tag{2.22}$$

$$K_2 = \frac{2.06}{2} = 1.03.$$

Time constant T is the time when output reaches 63% of the final steady state value, which is 35 seconds for tank 1 (secondary controller) and 25 seconds for tank 2 (primary controller).

Delay L is the time after which the system responds, once the input is provided. This is 2 seconds for each tank. In summary, the tank systems have the following transfer function:

$$G_1 = \frac{1.57}{35s+1}e^{-2s}, G_2 = \frac{1.03}{25s+1}e^{-2s}. \tag{2.23}$$

Water-Level Controller Design and Comparison in Simulink [S/W Mode]

The next step is controller design and controller tuning. Here, the secondary controller which controls the water-level in tank 1 is a proportional controller, having proportional parameter K_p^* equal to half of gain margin of tank 1 which is equal to 8.9595. Furthermore, the primary controller controls the cascaded plant given by:

$$G_3 = (\frac{K_p^* G_1}{1 + K_p^* G_1})G_2 = \frac{14.49}{875s^2 + 411.7s + 15.07}. \tag{2.24}$$

Again this is a second order transfer function, which is approximated to first order transfer function by the “getfod.m” file giving first order transfer function for the above cascaded arrangement as:

$$G_4 = \frac{0.9616}{25.1077s+1}e^{-2.2153s}. \tag{2.25}$$

Justification is made by making a comparison between actual and approximated plant as shown in fig. 2.18. Also, a delay of 4 seconds is added to this, accounting for an original

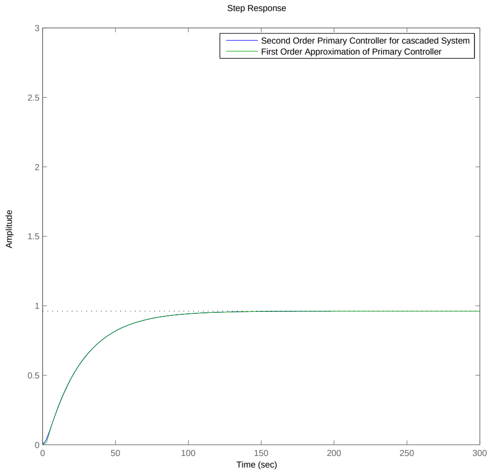


Fig. 2.18: Comparison of the actual cascaded CTS and the approximated first order CTS.

delay of 2 seconds in each tank. Thus, the final plant to be controlled by primary controller is given by (2.26) as:

$$G_4 = \frac{0.9616}{25.1077s + 1} e^{-6.2153s}. \tag{2.26}$$

Based on the first order transfer function obtained in (2.26), Table. 2.5 shows the gain parameters for different controllers for controlling water-level and fluid flow for cascade control configuration. Figure 2.19 shows the simulink block diagram of water-level control in the cascade CTS configuration. Figure 2.20 shows the result of a simulation of different controllers performing water-level control in the tank system. As can be seen from the simulation results, PI/ZN accounts for the largest amount of overshoot, but has a quick response. PI/MZN has a relatively less percentage overshoot whereas the new controller, i.e., the FO-PI/F-MIGO controller, results in a much smaller percentage overshoot. Percentage

Table 2.5: PI controller gain parameters for cascade control configuration.

	K_p	T_i	α
PI-ZN	2.911	18.224	
PI-MZN	3.4465	6.1765	
FOPI-FMIGO	1.5583	12.1726	0.9

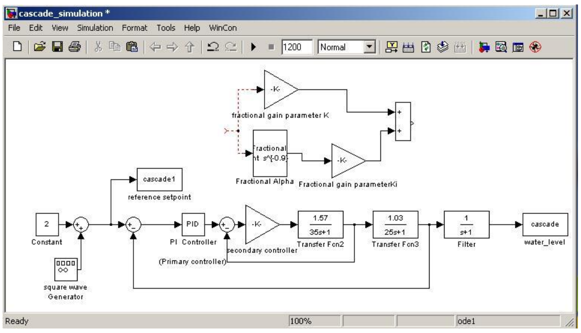


Fig. 2.19: Simulink block diagram of water-level control in the cascaded control configuration.

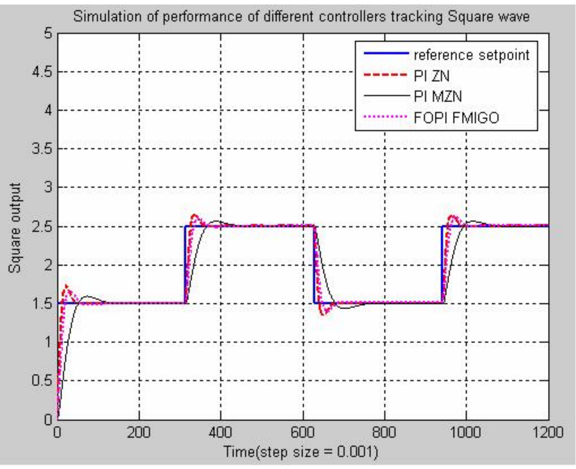


Fig. 2.20: Simulated control of water-level in the cascaded control configuration.

overshoot is defined as the maximum fraction by which the response overshoots the steady state value expressed as a percentage. In general, quick response and small overshoot are desirable in most of the target tracking control problems. However, it is well known that quick response results in a large overshoot. Thus, most of the design schemes have to make a trade-off between these two transient performance indices. In fact, it can be said from the simulation results, that the FO-PI/F-MIGO controller is a promising controller in fluid level and flow control in tanks.

Experimental Verification in Real-time [H/W Mode]

The final step in the design process of any control system is the control experiment. As can be seen, a PID block shows the controller (FOPI controller block is shown in case

2). Further as seen in fig. 2.21, the controller performances confirm the simulated results, although a small deviation from the simulated results can be accounted for due to uncontrollable real-time environmental disturbances such as fluid dynamics, modeling uncertainties, etc.

2.4 Conclusion

An intensive study of experimental work on water-level and fluid control in coupled tank PP-100 in three different configurations is provided. A comparison between different PI controller tuning methods has been made for each tank configuration. This is a four-step procedure comprising of system plant modeling, identification, validation, simulation, and real-time control experiments. For each case simulation results confirm real-time control results though disturbances due to water dynamics, modeling uncertainties, nonlinearities in the system, etc., are still responsible for differences between the two. Furthermore, it is observed that FO-PI(F-MIGO) is a promising controller in process industries and can perform even better when compared with integer order PI controllers.

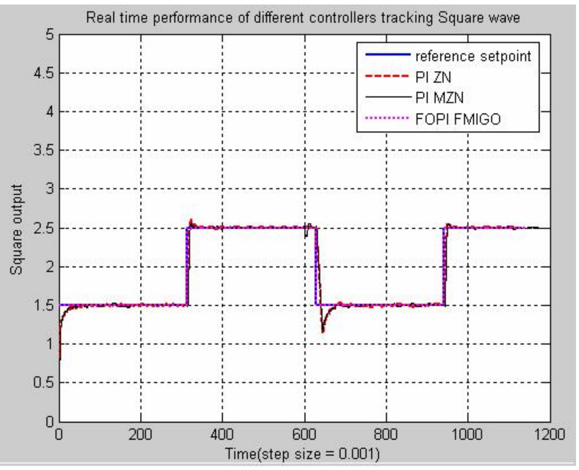


Fig. 2.21: Real-time control of water-level in the cascaded control configuration.

Chapter 3

Optimal Fractional Order Proportional and Integral (OFOPI) Tuning Rules for Varying Time-delay Systems

3.1 Introduction

Time-delays may cause poor performance, controller complexity, and even instability of system in many chemical, biological, mechanical, and transportation processes. Thesis works of Marti and Cervin [1, 2] provide extensive simulation results on how the jitter (variance in delay) in the loop can degrade system performance and lead to instability of system. Ensuring the stability of systems with varying time-delays has always been an interesting area of research for control engineers like Wu et al., Phat et al., and Kao et al. [3–5]. This chapter introduces a novel jitter-robust controller design by optimizing the gain parameters of fractional order proportional integral (FO-PI) controller based on fractional M_s constrained integral gain optimization (F-MIGO) algorithm [6–9]. This controller design is useful in finding the maximum value of jitter at which the system remains stable. The integral of time weighted absolute error (ITAE) performance of the proposed controller is better than the best integer order proportional integral derivative (PID) controller for lag dominated systems.

For this, a set of hundred first order plus delay time (FOPDT) systems is considered and FO-PI tuning rules are used to compute the proportional gain, integral gain, and non-integer order α of the integrator for each system. The proportional gain (K_p), integral gain (K_i) and fractional order (α) values so obtained for each system are then used to compute the ITAE performance and jitter margin. Finally, multi-objective optimization algorithm is applied to optimize these gains and α values for each system. Jitter margin and ITAE value are calculated at these optimum values as in works of Eriksson et al. [8, 9].

The main contribution of this work lies in answering this important research question: "*Will a fractional order controller help and do better?*" when there are uncertainties in the time delay due to e.g., jitter in the loop.

This chapter is organized as follows: Section 3.2 briefly defines the jitter margin, ITAE performance index, and the multi-objective optimization method. Section 3.3 focuses on optimal tuning of FO-PI controller followed by sec. 3.4 which aims at approximation of optimized gain parameters K_p, K_i, and α to get new set of optimal FO-PI tuning rules that ensures the best jitter margin and ITAE performance. Finally, sec. 3.5 concludes this chapter with remarks on future research work.

3.2 Preliminaries

This work is established on design of optimum FO-PI controller to control a class of systems which can be approximated by FOPDT model, also called gain delay time-constant (KLT) model.

3.2.1 Multi-objective Optimization Problem

Multi-objective optimization method is applied which simultaneously minimizes n objective functions $O(x)$ which are functions of decision variables x bounded by some nonlinear equality and inequality constraints. This is expressed mathematically as:

$$\min_{x} O(x), \tag{3.1}$$

subjected to the following equality and inequality constraints:

$$\sigma = \begin{cases} c_i(x) \leq 0, & i = 1, \cdots, n_1, \\ ceq_i(x) = 0, & i = 1, \cdots, n_2, \end{cases}, \tag{3.2}$$

where $O(x) = [O_1, O_2, \cdots, O_n]^T$ and $x = [x_1, x_2, \cdots, x_k]^T$.

3.2.2 Optimization Criteria

This work is based on optimization of two important controller performance indices, namely, jitter margin and ITAE, which are briefly explained in this section.

Jitter Margin

Let $G(s)$ be an first order plus delay time (FOPDT) plant system as shown in (2.1) and $C(s)$ be the FO-PI controller given by (2.3). Let $\Delta(t)$ be the time-varying delay of the system as shown in fig. 3.1.

The jitter margin is defined as the maximum delay that a closed-loop system can have while still maintaining its stability and performance. Furthermore, the condition of stability for continuous-time varying delay systems can be verified by (3.3). This work takes into account the form of equation given in Eriksson et al. [8,9] for finding the stability condition for single input single output (SISO) continuous systems, though information provided in thesis work of Marti [1] is also quite useful in regard to jitter margin.

$$|\frac{G(j\omega)C(j\omega)}{1+G(j\omega)C(j\omega)}| < \frac{1}{\delta_{max}\omega} \tag{3.3}$$

For the purpose of proving the above condition of stability, consider the transformed system shown in fig. 3.2 which is same as system represented by fig. 3.1 as shown in work of Kao and Lincoln [30], where signals $m(t)$ and $n(t)$ are marked between two dashed blocks Δ_F and G_{nm}.

Following Kao et al. [30], let us denote the operator Δ as

$$\Delta m(t) = m(t - \delta(t)) \text{ s.t. } 0 \leq \delta(t) \leq \delta_{max}, \tag{3.4}$$

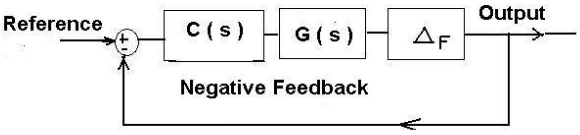


Fig. 3.1: Block diagram of closed-loop system with delay.

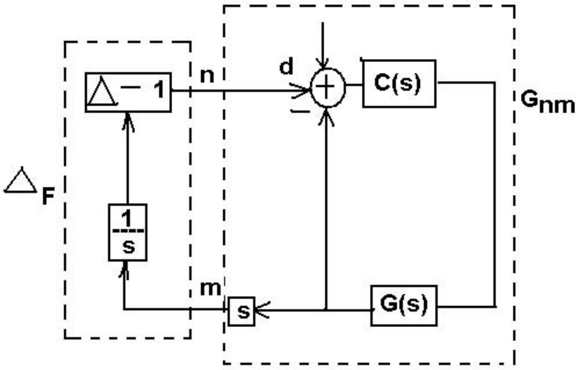


Fig. 3.2: Block diagram of the equivalently transformed system.

and obviously, via the operator Δ_F of the left dashed box in fig. 3.1,

$$n(t) = \Delta_F m(t) = (\Delta - 1)\frac{1}{s}. \tag{3.5}$$

Then, $y(t)$, the output signal of the plant $G(s)$, can be expressed as:

$$y(t) = \int_0^t m(\nu)\mathrm{d}\nu. \tag{3.6}$$

Therefore, Δ_F can be expressed as:

$$n(t) = \Delta_F m(t) = y(t - \delta(t)) - y(t) = \int_{t-\delta(t)}^t m(\nu)\mathrm{d}\nu. \tag{3.7}$$

Thus,

$$\Delta_F m(t)^2 \leq \delta(t) \int_0^t m(\nu)^2 \mathrm{d}\nu \leq \delta_{max} \int_0^t m(\nu)^2 \mathrm{d}\nu. \tag{3.8}$$

Now, the L_2 norm of $n(t)$ is bounded as:

$$\| \Delta_F \|_{L_2}^2 \leq \int_0^\infty \delta_{max} (\int_{t-\delta(t)}^t m(\nu)\mathrm{d}\nu)\mathrm{d}t \tag{3.9}$$

$$= \delta_{max} \int_0^\infty \int_{-\delta_{max}}^0 m(t+s)^2 \mathrm{d}s\mathrm{d}t$$

$$= \delta_{max} \int_{-\delta_{max}}^0 \int_0^\infty m(t)^2 \mathrm{d}t\mathrm{d}s$$

$$= \delta_{max} \parallel m(t) \parallel_{L_2}^2 \int_{-\delta_{max}}^{0} \mathrm{d}s = \delta_{max}^2 \parallel m(t) \parallel_{L_2}^2 .$$

Thus, the L_2 gain of operator Δ_F is bounded by δ_{max}. The stability criterion in (3.3) is from the small gain theorem applied to transformed system block diagram. Thus, one can conclude that the transformed system is stable if L_2 induced norm of linear part of system from n to m is bounded by $1/\delta_{max}$, i.e,

$$\parallel G_{nm} \parallel_{L_2} = \sup_{\omega \epsilon [0,\infty]} |\frac{G(j\omega)C(j\omega)}{1+G(j\omega)C(j\omega)}| < \frac{1}{\delta_{max}\omega}. \tag{3.10}$$

This work deploys above stability condition for computing jitter margin of KLT systems.

The Integrated Time-weighted Absolute Error (ITAE) Criterion

ITAE stands for integral of time weighted absolute error, that is,

$$ITAE = \int_0^\infty t|e(t)|\mathrm{d}t. \tag{3.11}$$

Optimum ITAE is used as a deciding factor in design and tuning of controllers by many researchers, mainly Caceres et al. [31] and Shrivastava [32]. Note that lower ITAE value implies better performance of the system.

3.3 Test Bench Simulation and Optimization

The objective of this study is to design an optimum FO-PI controller such that the jitter margin and system performance are maximized and yet the closed-loop feedback system is stable. For our numerical simulation and optimization studies, a set of 100 first order plus delay time (FOPDT) systems are used with 10 delay values $L = [1, 2, \cdots, 10]^T$ and 10 time-constant values $T = [1, 2, \cdots, 10]^T$ and $K = 1$. These values are substituted in equation of first order plus time delay systems as in (2.1) to get 100 different systems. Further two objective functions are targeted as:

$$O_1(x) = ITAE = \int_0^\infty t|e(t)|\mathrm{d}t, \tag{3.12}$$

and

$$O_2(x) = \frac{1}{\delta_{max}}.$$

Note that, here δ_{max} can be computed from (3.3) as:

$$\delta_{max} = \min_{\omega} \epsilon[0, \infty]|\frac{1 + G(j\omega)C(j\omega)}{j\omega G(j\omega)C(j\omega)}|. \tag{3.13}$$

Hence, the multi-objective optimization problem takes the following form:

$$\min_{x=[K_p,K_i,\alpha]} O(x), \tag{3.14}$$

where $O(x) = [O_1, O_2]$ subject to the following equality and inequality constraints,

$$\sigma : \begin{cases} |C_0 + G(j\omega)C(j\omega)|^2 \geq R_0^2, & i = 1, \cdots, n_1 \\ \frac{\partial|C_0+G(j\omega)C(j\omega)|^2}{\partial\omega} = 0, & i = 1, \cdots, n_2, \end{cases} \tag{3.15}$$

where the objective function $O_1(x)$ is the ITAE criterion and $O_2(x)$ is the inverse of jitter margin. These values should be minimized while still ensuring robust stability of the system. The set of constraint equations defined by σ ensures the robustness and stability. The inequality constraint $|C_0+G(j\omega)C(j\omega)|^2$ is the sensitivity constraint which is also a function of K_p, K_i, α and ω and must be greater than R_0^2. Here, C_0 and R_0 are the center and radius of the circle which encloses both the M_s and M_p circles described by (3.17):

$$C_0 = \frac{M_s - M_sM_p - 2M_sM_p^2 + M_p^2 - 1}{2M_s(M_p^2 - 1)}, \tag{3.16}$$

$$R_0 = \frac{M_s + M_p - 1}{2M_s(M_p^2 - 1)}, \tag{3.17}$$

where M_s and M_p are the maximum absolute values of sensitivity and complementary sensitivity functions, respectively. Furthermore, as remarked in works of Bohannan et al. and Eriksson et al. [7–9], $|1 + G(j\omega)C(j\omega)|^2 = 0$ is the stability region of the sensitivity constraint and satisfies the boundary condition at critical point or the point at which $C_0 = 1$ and $R_0 = 0$. In our implementation, a MATLAB command `fgoalattain` is used to get optimized values of x by multi-objective goal attainment. `fgoalattain` command finds the minimum of a multi-objective optimization problem by minimizing γ such that $O(x) - W\gamma \leq O_{goal}$, where x are the optimized gain parameter values, W is the weight (generally equal to absolute value of goal function), and O_{goal} is the target values of the objective functions. In our case, goal and weight values are given by :

$$\mathbf{O_{goal}} = \left[\begin{array}{cc} J_{FMIGO} & \frac{1}{T+L} \end{array} \right], \tag{3.18}$$

$$\mathbf{W} = \left[\begin{array}{cc} J_{FMIGO} & \frac{2T}{T+L} \end{array} \right],$$

where J_{FMIGO} is the ITAE cost criterion value if F-MIGO tuning rules are used and since $(T + L)$ gives relatively large jitter margin for both delay dominated and lag dominated systems, it is chosen as goal for jitter margin. For systems with $T > L$, setting goal equal to weight results in poor performance because of trade-off between the objectives. To avoid this situation, weight is different from goal. Optimization is run for many iterations until minimum values of Ox are obtained. The final value of x which is the result of `fgoalattain` command is the optimized gain parameters and α value. At these values, the system has maximum jitter margin and good performance ensuring robustness and stability.

Furthermore, extensive simulations were performed to investigate the behavior of fractional order proportional integral controller after optimization. A plot of jitter margin as a function of τ, before and after optimization, of FO-PI controller is shown in fig. 3.3. The jitter margin of KLT systems after optimization is comparatively larger (up to two fold for $\tau \leq 0.8$) than the jitter margin of systems prior to optimization. Similar increase in jitter margin for delay dominated systems ($\tau > 0.6$) is accompanied with very slight decrease in performance. This is in contrast to lag-dominated systems that show an increase in the jitter margin without adversely affecting the system performance, as shown in fig. 3.4. Nonetheless, the jitter margin is improved in all the cases studied above. Thus, it could be

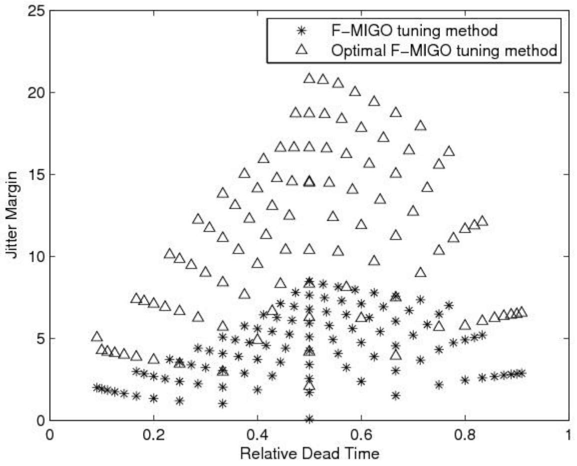


Fig. 3.3: Improvement of jitter margin of KLT systems after optimization of FO-PI controller.

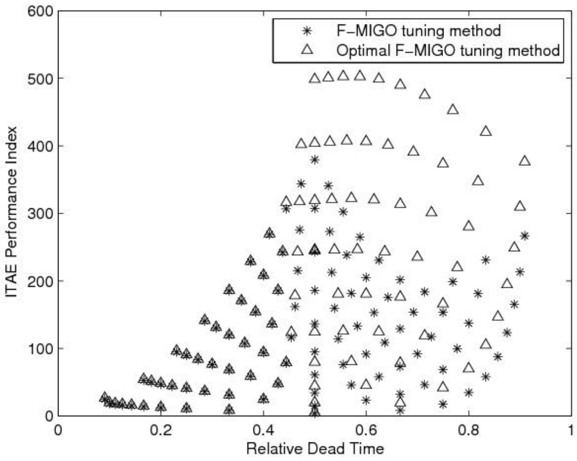


Fig. 3.4: Performance of KLT systems before and after optimization of FOPI controller.

inferred that the optimal F-MIGO tuning is a better option over simple F-MIGO tuning in increasing the jitter margin of closed-loop systems controlled by fractional order controllers.

3.4 Optimal FO-PI Tuning and Verification

This section describes the methods used for derivation of optimal FO-PI tuning rules. The optimized gain parameters and α obtained as result of `fgoalattain` command in previous section were plotted in MATLAB and analyzed carefully to find any hidden pattern or there dependence on the delays L and the time-constants T of the systems. For example, it was found that the optimal proportional gain parameter K_p^o increases with increasing T and decreases with increasing L. Whereas, optimal integral gain parameter K_i^o shows a decrease with increasing values of T and L, though this decrease is more profound for small values of L and becomes almost constant for large values of L. Furthermore, it should be noted here that the integral gain parameters so obtained from multi-objective optimization method ensure the stability of the systems, but do not result in true jitter margin when tested in Simulink. Thus, to tighten the constraint, the optimized K_i values were increased by some integer factor which was determined by simulation results. The optimal fractional order α^o of the integrator was a function of the relative dead time τ and delay L of the system. These optimal tuning rules are expressed mathematically as:

$$\begin{aligned} K_p^o &= \frac{0.2T}{L} + 0.16, && (3.19)\\ K_i^o &= \frac{0.25}{TL} + \frac{0.19833}{L} + 0.09, && (3.20)\\ \alpha^o &= \tau - 0.04L + 1.2399. && (3.21)\end{aligned}$$

To verify the tuning rules obtained above, three different types of systems are considered and step response plotted as shown in figs. 3.5, 3.6, and 3.7. These are a lag dominated system with $\tau = 0.2$ $(K = 1, L = 2, T = 8)$, an intermediate delayed system with $\tau = 0.5$

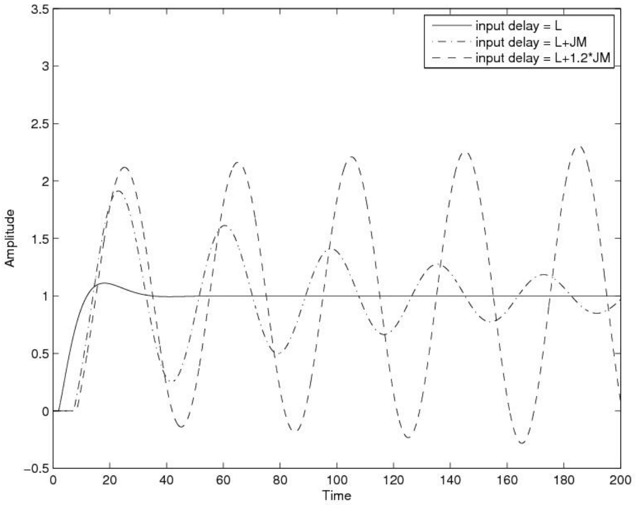


Fig. 3.5: Step response of lag dominated system $(K = 1, L = 2, T = 8)$ at different delays.

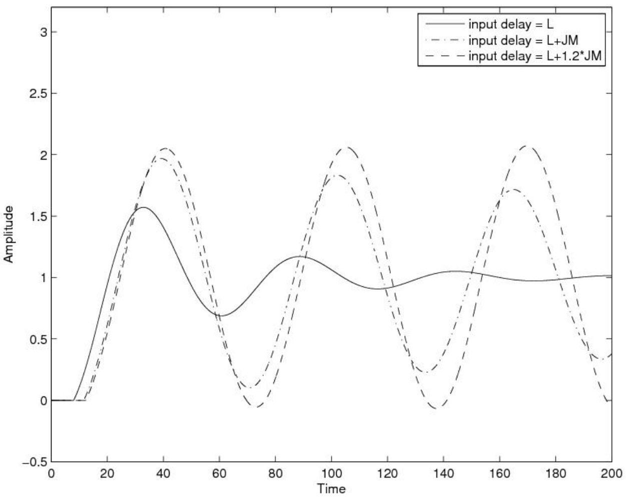


Fig. 3.6: Step response of an intermediate system $(K = 1, L = 8, T = 8)$ at different delays.

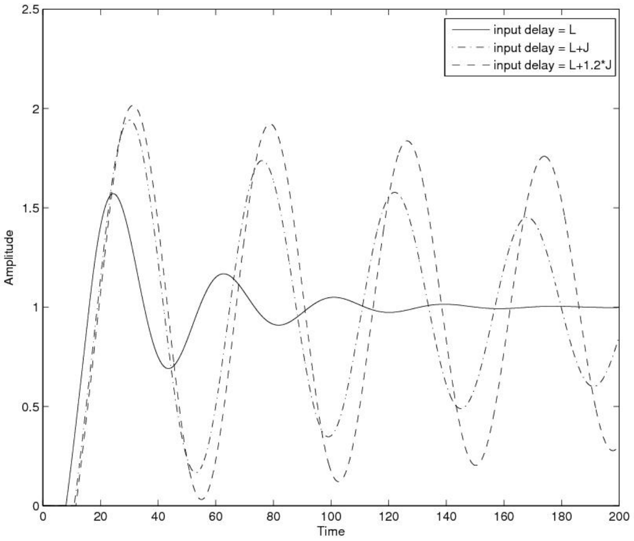


Fig. 3.7: Step response of delay dominated system $(K = 1, L = 8, T = 2)$ at different delays.

$(K = 1, L = 8, T = 8)$ and a delay dominated system with $\tau = 0.8$ $(K = 1, L = 8, T = 2)$. The optimal gain parameters K_p^o, K_i^o, and α^o are computed using (3.19), (3.20), and (3.21), respectively. These are then used to compute the jitter margins using stability criteria in (3.3).

It can be observed that for all the three cases considered, systems are stable at jitter margin (shown by `JM` in the figures) and become unstable if the jitter margin is increased by just 20 per cent. Several other systems were simulated to see the validity of the tuning rules and it was found that tuning rules are quite accurate.

3.5 Comparison between OFOPI and OPID Controllers

In addition to designing of an optimal FO-PI (OFOPI) controller and developing optimal FO-PI tuning rules, we also compare the OFOPI controller and the optimal PID controller (OPID) studied in Eriksson et al. [8,9] and further stated briefly in Appendix A. Briefly summarizing, the OPID controller is represented in time-domain as:

$$\begin{aligned} u(t) &= k(py_r(t) - y_f(t)) + k_i \int_0^t (y_r(\tau) - y_f(\tau))\mathrm{d}\tau \\ &\quad + k_d(q\frac{\mathrm{d}y_r(t)}{\mathrm{d}t} - \frac{\mathrm{d}y_f(t)}{\mathrm{d}t}), \end{aligned} \tag{3.22}$$

where k, k_i, and k_d are the gain parameters of the controller given by AMIGO tuning rules, p and q are the setpoint weights and y_f is the filtered process variable. The output is considered to pass through a low-pass filter having a transfer function $G_f(s)$ given as:

$$G_f(s) = \frac{1}{(T_f s + 1)^2}. \tag{3.23}$$

The other controller parameters are defined mathematically as:

$$p = \begin{cases} 0, & \text{if } \tau \leq 0.5, \\ 1, & \text{if } \tau > 0.5; \end{cases} \tag{3.24}$$

$$q = 0;$$

$$T_f = \begin{cases} \frac{0.05}{\omega_{gc}}, & \text{if } \tau \leq 0.2, \\ 0.1L, & \text{if } \tau > 0.2; \end{cases}$$

where ω_{gc} is the cut off frequency of the filter. Further, the tuning rules proposed in Eriksson et al. [8,9] on PID controller were as:

$$k = \frac{0.4T - 0.04}{K_p L} + \frac{0.16}{K_p}, \tag{3.25}$$

$$k_i = 0.01(\frac{-0.11T^3 + 1.5T^2 - 1.5}{K_p L^2} + \frac{0.35T^2 + 4T + 50}{K_p L}), \tag{3.26}$$

$$k_d = 0.01(\frac{0.4T^2 + 11T}{K_p}). \tag{3.27}$$

Jitter margins and ITAE indices were calculated for the test batch of hundred KLT systems by using these optimal AMIGO tuning rules (OPID controller) and optimal F-MIGO tuning rules (OFOPI controller). These are shown in fig. 3.8 and fig. 3.9, respectively.

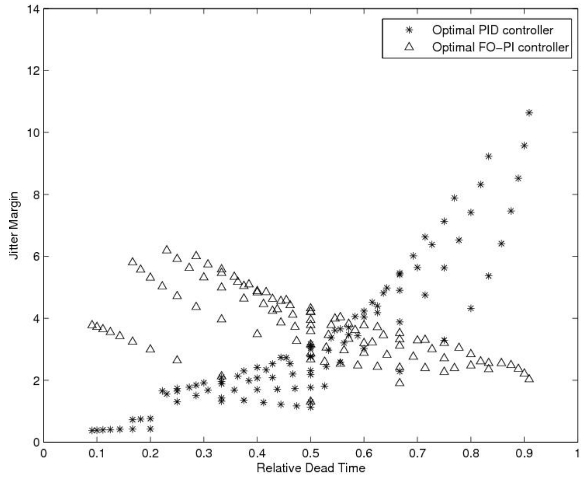


Fig. 3.8: Jitter margin of KLT systems controlled by OPID and OFOPI controllers.

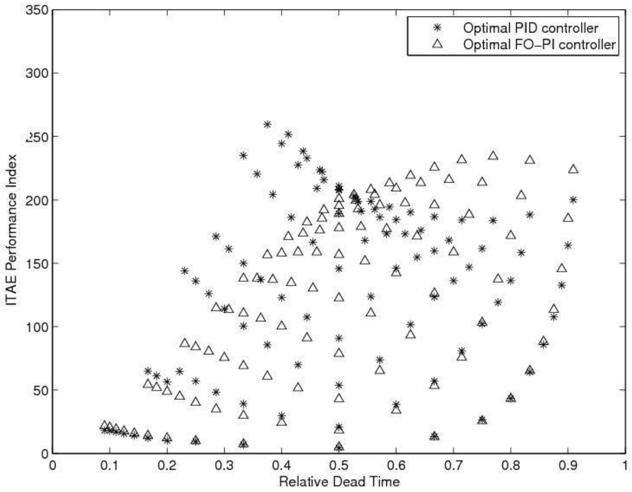


Fig. 3.9: Performance of KLT systems controlled by OPID and OFOPI controllers

.

It should be noted that the OPID controller proposed in Eriksson et al. [8, 9] uses a low-pass filter which enhances the performance of the controlled system, whereas the OFOPI controller designed in this work uses no filter for process output.

Thorough investigation of these figures reveals that OFOPI is a better controller than OPID for systems with $\tau < 0.5$. These systems have larger jitter margin and lower ITAE values than that obtained by OPID controller. This is in contrast to OPID controller which have a better performance for systems with $\tau > 0.5$.

3.6 Conclusion and Future Work

A detailed explanation of design of a robust-jitter controller called optimum fractional proportional integral controller was provided. A set of hundred KLT system was simulated and compared to prove the efficiency of controller in providing higher jitter margin when compared to simple FO-PI controller and PID controller (for $\tau < 0.5$). Finally, tuning rules were given to determine the gain parameters and α of OFOPI controller. This kind of controller could prove to be a better option than OPID controller for systems with small value of τ and when large jitter margin and better controller performance are desirable.

Present work considers a special case when $\delta(t) = \delta_{max}$, for all values of t. This is shown in figs. 3.10, 3.11, and 3.12 by simulating three different systems when $\delta(t) \neq \delta_{max}$

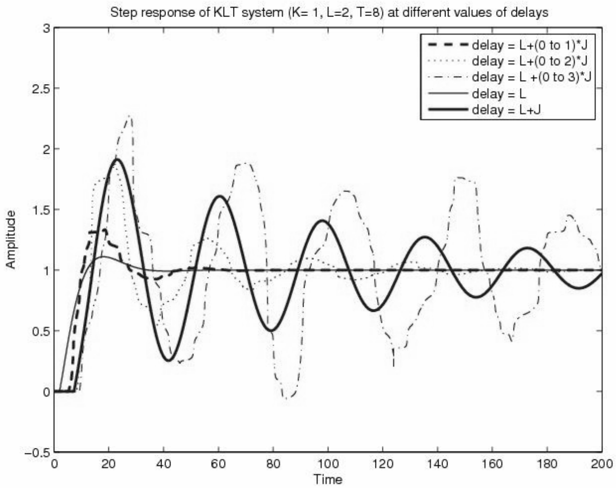


Fig. 3.10: Step response of lag dominated system $(K = 1, L = 2, T = 8)$ at different delays.

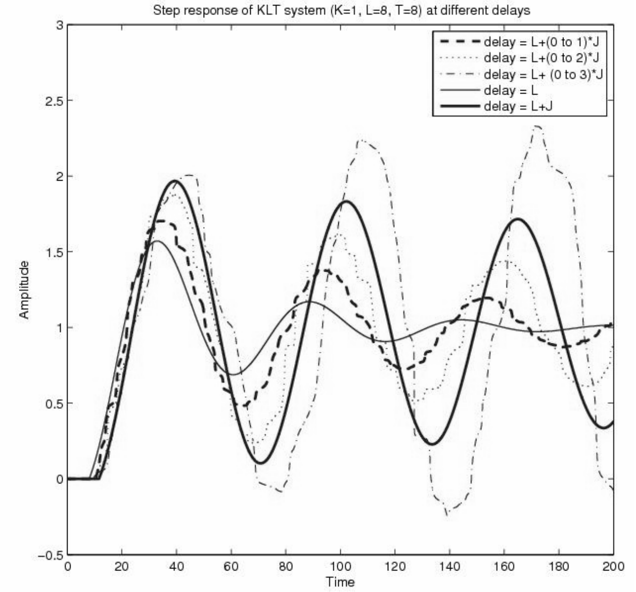


Fig. 3.11: Step response of an intermediate system ($K = 1, L = 8, T = 8$) at different delays.

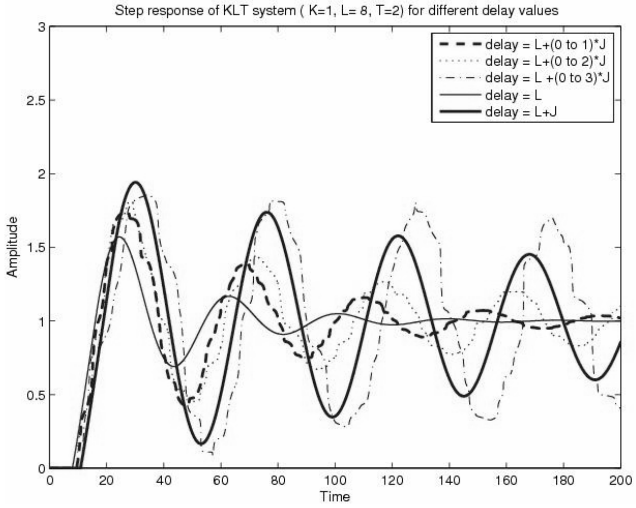


Fig. 3.12: Step response of delay dominated system ($K = 1, L = 8, T = 2$) at different delays.

and $\delta(t)$ is uniformly distributed in a a given range. In other words, $\parallel \Delta_F \parallel_{L_2} = \delta_{max}$. For such a case, the tuning rules give the gain parameters of the OFOPI controller at which the jitter margin is maximized for the system. In other cases when $\delta(t) \neq \delta_{max}$, these tuning rules no longer hold.

Future research work will include design of OFOPI tuning rules for systems when $\delta(t) \neq \delta_{max}$. We also plan to investigate several other sets of 100 KLT test batches for validation purpose and engineering an embedded and telepresence control of a three-axis T2 Stand-alone Smart wheel control at CSOIS using the OFOPI controller/tuning rules.

Chapter 4

Networked Control of Smart Wheel using OFOPI Tuning Rules

4.1 Introduction

As stated in works of Pohjola [33] and Katayama et al. [34], the traditional local control loop is expected to expand to tomorrow's control over large communication network with variable delay times. Random delay has serious impacts on networked control systems (NCS) such as the deterioration of the performance, instability, etc. In this chapter, real random delays are incorporated in our NCS testbed in pinging different remote sites for both hardware-in-the-loop simulation and actual experiments. Furthermore, experimental results in a NCS platform: Stand-alone Smart Wheel are reported using the tuning rules developed in this thesis. The optimization tuning method performed almost equally well in practice as in simulations. The thesis also shows that the tuning rule development procedure for optimal fractional order proportional integral (OFOPI) is not only valid for first order plus delay time (FOPDT) systems but also applicable for other general class of plants.

Center of Self Organizing and Intelligent Systems (CSOIS) at Utah State University has a prototype enhanced Smart Wheel mounted on a stand as shown in fig. 4.1. The Smart Wheel is a self-contained wheel module equipped with a steering motor, a drive motor, and an innovative slip ring, which allows data and power to pass from the chassis to the wheel without a wired connection. The slip ring allows infinite rotation in the steering degree of freedom. When multiple smart wheels are attached to a chassis, the resulting vehicle is called an Omni-directional Vehicle (ODV). This vehicle is capable of driving along a path with independent rotation and motion in the X-Y plane, unlike traditional Ackerman-steered vehicles (automobiles) or a tracked vehicle that uses skid-steering (tanks and forklifts).

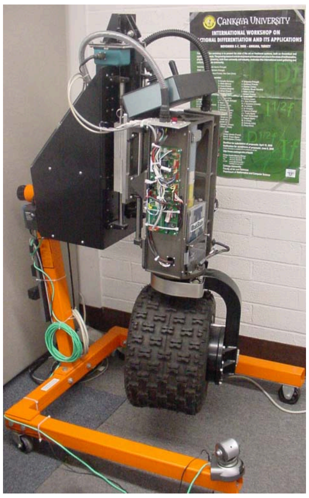

ÇANKAYA UNIVERSITY
INTERNATIONAL WORKSHOP ON
TIONAL DIFFERENTIATION AND ITS APPLICATIONS

Fig. 4.1: Stand-alone enhanced Smart Wheel assembly at CSOIS.

The chapter is divided into six sections. Section 4.2 describes the hardware architecture of the Smart Wheel system. Section 4.3 briefly introduces the software architecture. Section 4.4 describes the networked steering speed control of Smart Wheel using fractional and integer order controller tuning rules at random delays. This is divided into sections on system identification of Smart Wheel plant, laboratory setup for the networked control system design for the steering axis of the Smart Wheel using hardware-in-the-loop speed control, simulation techniques, and real-time experimental results. Section 4.5 summarizes the results and provides concluding remarks.

4.2 Hardware Description

The Smart Wheel at CSOIS is equipped with steering and drive motors, a linear actuator for z-axis movement, drive circuitry for the motors and actuator, encoders for drive and steering feedback, a controller board, and a power distribution unit. The steering motor, drive motor, and linear actuator control the steering axis, drive axis, and z-axis of the Smart Wheel, respectively.

4.2.1 Mechanical Hardware Architecture

The mechanical hardware architecture of the Smart Wheel is depicted in fig. 4.2. There

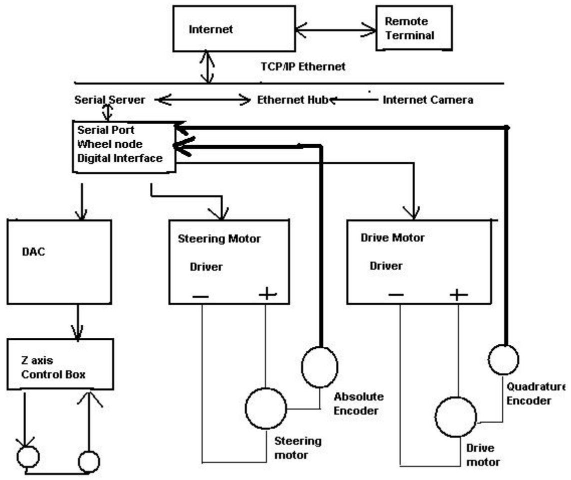


Fig. 4.2: The mechanical hardware architecture of Smart Wheel demonstrator.

is a steering suspension made of spring-over-damper, linear type. It supports the entire steering mechanism and wheel electronic systems. The steering motor is directly coupled to the steering spindle. The yoke fastens the drive assembly to the steering spindle and also houses the power and communications cables, which go down to the wheel. A custom slip ring allows infinite rotation of the wheel about the steering axis. The wheel shell encloses the entire drive assembly including the drive spindle, motor, failsafe brake, and encoder. The drive motor is a Kollmorgen model QT-6407 frameless torqued motor and the steering motor is a MicroMo series GNM 5440 PM DC Motor.

The steering and drive motors are controlled by pulse width modulation (PWM) and direction signals from their respective motor drivers. The drive motor is attached to a quadrature encoder that measures the relative motion of the wheel. The steering motor is coupled to an absolute encoder, which measures the absolute angular position of the wheel. The linear actuator moves the entire steering column up or down along the linear slide. The linear potentiometer provides feedback about the z-position of the steering column. The z-axis control box commands the linear actuator based on the voltage value from the digital-to-analog converter (DAC) and the feedback from the linear potentiometer.

4.2.2 Wheel-node Electronic Hardware Architecture

Smart Wheel has the wheel node digital interface board which consists of the Tattletale TT8 based controller [35] along with the interfaces for serial communication, motor drivers, DAC, Model 40 H absolute encoder, CP560 quadrature encoder, circuits for power supply and regulation, level-shifting, and optical isolation, and the hardware watchdog unit. Figure 4.3 shows the wheel node electronic hardware architecture. The failsafe brake from Warner operated by a brake relay prevents the T-2 vehicle from freewheeling in the event of a system failure. The power distribution unit supplies 5 V to the z-axis control box and 48 V to each of the two motor drivers. It also supplies +12 V, -12 V and +5 V to the wheel node digital interface board.

4.2.3 NPort DE-311 1-Port Serial Server

The remote operator communicates with the Smart Wheel through the Neteon Net NPort DE-311 1-Port Serial Server from Moxa Technologies [35], which converts messages

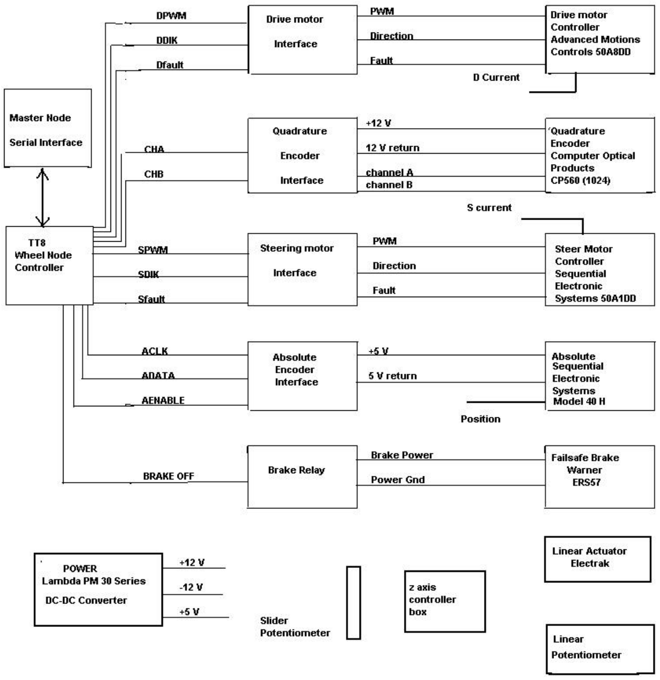

DPWM
DDIK
Dfault
Drive motor Interface
PWM
Direction
Fault
Drive motor Controller Advanced Motions Controls 50A8DD
D Current
Master Node Serial Interface
CHA
CHB
Quadrature Encoder Interface
+12 V
12 V return
channel A
channel B
Quadrature Encoder Computer Optical Products CP560 (1024)
S current
TT8 Wheel Node Controller
SPWM
SDIK
Sfault
Steering motor Interface
PWM
Direction
Fault
Steer Motor Controller Sequential Electronic Systems 50A1DD
ACLK
ADATA
AENABLE
Absolute Encoder Interface
+5 V
5 V return
Absolute Sequential Electronic Systems Model 40 H
Position
BRAKE OFF
Brake Relay
Brake Power
Power Gnd
Failsafe Brake Warner ERS57
POWER Lambda PM 30 Series DC-DC Converter
+12 V
-12 V
+5 V
Slider Potentiometer
z axis controller box
Linear Actuator Electrak
Linear Potentiometer

Fig. 4.3: The wheel node electronic hardware architecture.

from RS232 format into TCP/IP format and vice-versa. The serial server is powered with a 9 V adapter. A straight-through ethernet cable is used to connect one end of the serial server to the nearest network hub or switch. The other end is connected to the RS232 main port of the smart wheel with a straight-through DB 9 serial cable. Figure 4.4 shows the interfacing details of the serial server with the rest of the system.

4.2.4 DCS-5300 Internet Camera

The DCS-5300 Internet camera from D-Link, as explained in thesis work of Bharat [36] located near the Smart Wheel assembly directly sends streaming video and audio of the Smart Wheel motion to the remote operator. It is connected to the nearest network hub or switch, placed facing the Smart Wheel, and positioned for optimum viewing. It is powered with a 12 V power adapter. Figure 4.5 shows the camera and its connections. The remote

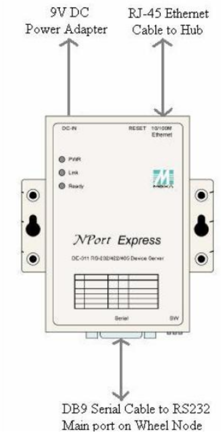


Fig. 4.4: Interfacing details of the Neteon Net NPort DE-311 1-Port serial server.

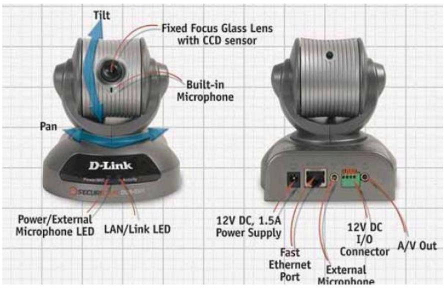


Fig. 4.5: The DCS-5300 Internet camera from DLink.

computer communicates with the serial server using a virtual COM driver.

4.3 Software Description

The wheel node software run on the Aztec C cross-compiler and the Crosscut terminal emulator. Though the TT8 has good libraries of built-in functions and device drivers, some of them needs to be re-written for specific purposes. A real-time kernel and scheduler called uCOS by Jean J. Labrosse was used to schedule and synchronize the wheel node low-level controller (LLC) tasks. The low-level controllers process commands from the master node, for the interface to the electronic hardware, and for control of the actuator systems. Furthermore, they generate odometry and status information for the master node. A steering control task uses a PID/FO-PI controller for wheel positioning. It outputs a PWM and direction signal to the steering motor driver and receives feedback from the absolute encoder. Similarly, a drive control task uses a PID/FO-PI controller for wheel speed. It outputs a PWM and direction signal to the drive motor driver and receives feedback from the relative encoder. A common packet structure called standard packet protocol (SPP) is used for all serial communications between the wheel node and master node. Packets or data that do not follow the SPP are rejected and do not affect wheel behavior. Further references can be found in Bharat's thesis work [36].

Figure 4.6 represents the software architecture of the stand-alone Smart Wheel assembly. It has a hierarchial arrangement of layers namely application layer/user program,

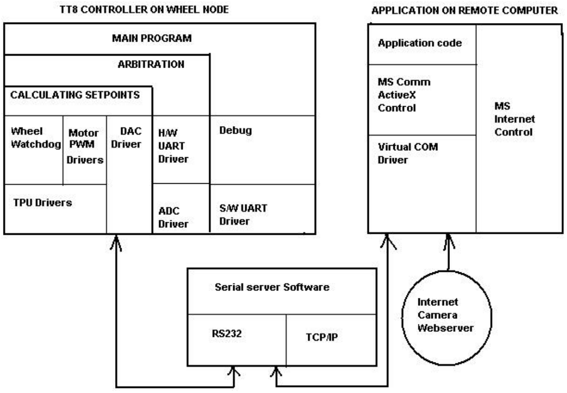


Fig. 4.6: Software architecture of stand-alone Smart Wheel demonstration system.

arbitration program and calculate setpoint program. Each software layer serves the layer above it. The lower-most layers interact directly with the physical hardware. The application program is any user-made program which serves as an interface between the user and the Smart Wheel. This is further described in sec. 4.4. The communication channels between blocks are represented by arrows.

4.3.1 Software Architecture Overview

Software for the stand-alone Smart Wheel demonstration system comprises of:

1. Embedded software on the TT8 board,
2. Video and audio buffering and streaming software on the Internet cameras web server,
3. Embedded software to perform Ethernet-serial conversion in the serial server,
4. Virtual COM driver that allows communication between the remote computer and the serial server,
5. MATLAB/Simulink application program implementing the communication and controller design.

The embedded software on the TT8 board performs a number of functions like:

- Poll the serial port for set-points from the remote operator,
- Send appropriate handshaking signals to the remote operator through the serial port,
- Provide control signals to the motor drivers for steering and drive motors,
- Provide voltage signals to the digital-to-analog converter (DAC) for z-axis setpoint.

Furthermore, the serial data from the remote operator that enters the TT8 board through the hardware UART must be converted to the appropriate set-points for the three wheel axes. The data from the remote computer is in the form of ASCII characters. The ASCII character set from a (97) to t (116) is mapped linearly into 20 different steering speed values in both directions, limited between -0.5 and 0.5. Similarly, the ASCII character set from G (71) to Z (90) is mapped linearly into 20 different drive speed values in both directions (between -1.0 and 1.0) and the sequence from 0 (48) to D (68) into 21 different positions along the z-axis. Appendix B provides the formulae for conversion of serial data into axes set-points.

4.3.2 Arbitration Mechanism

This subsection explains what will happen when two or more remote operators contend for control of the Smart Wheel demonstration system. This is further depicted in form of a flowchart as shown in fig. 4.7. Arbitration has been implemented in the system as per the following rules:

1. At any given time, the wheel node will be in one of two modes of operation, namely, the idle mode or the remote mode.

2. When the Smart Wheel is in the idle mode, it waits for any remote operator to begin controlling it.

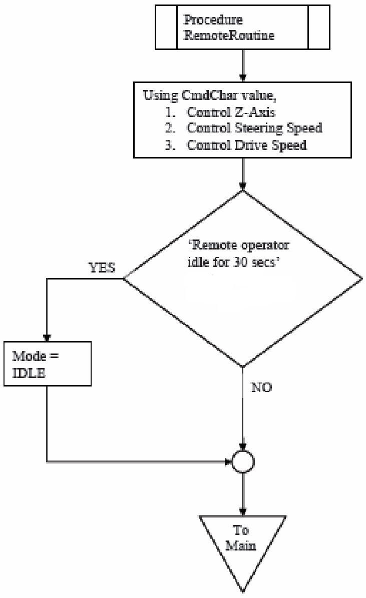


Fig. 4.7: Flowchart describing working mechanism in Smart Wheel.

3. The remote operator requests control of the wheel by issuing a REMOTE_REQ character. Once the wheel node receives this character, it switches to the remote mode and allows commands from the remote operator to affect the set-points of the three axes.

4. When the remote operator is idle for 30 seconds, a timeout occurs and the system slips into the idle mode.

Communication Protocol

The TT8 board on the Smart Wheel communicates asynchronously with the remote application through the serial server. Data are transferred in the form of ASCII characters. The remote user requests access to the wheel by transmitting a REMOTE_REQ character to the wheel. Depending on the status of the wheel node, appropriate action is taken. Figures 4.8, 4.9, and 4.10 depict three different scenarios during communication between the remote computer (Host PC) and the TT8 board via the serial server. Note that the figures are not to scale. In fig. 4.9, when the wheel is in idle mode, the TT8 responds to the remote access request with a REMOTE_ACCEPT character and then goes into remote mode. Any further data from the remote operator denote wheel commands, which are converted into set-points and given to the wheel motor drivers or DAC. At the stop time, the session ends and sends the REMOTE_GIVEUP character, which takes the wheel back into idle mode and disables the control panel. Figure 4.10 shows the scenario when a

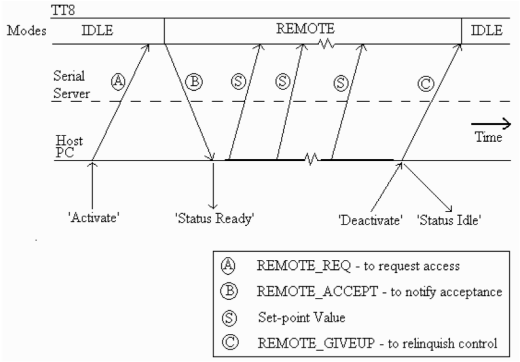


Fig. 4.8: Successful telepresence control session.

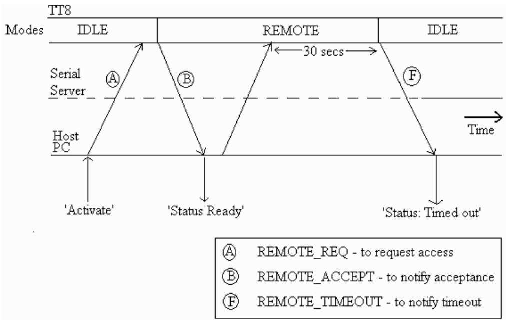

TT8
Modes
IDLE
REMOTE
IDLE
30 secs
Serial Server
A
B
F
Host PC
Time
'Activate'
'Status Ready'
'Status: Timed out'
A REMOTE_REQ - to request access
B REMOTE_ACCEPT - to notify acceptance
F REMOTE_TIMEOUT - to notify timeout

Fig. 4.9: Session termination when idle for 30 seconds.

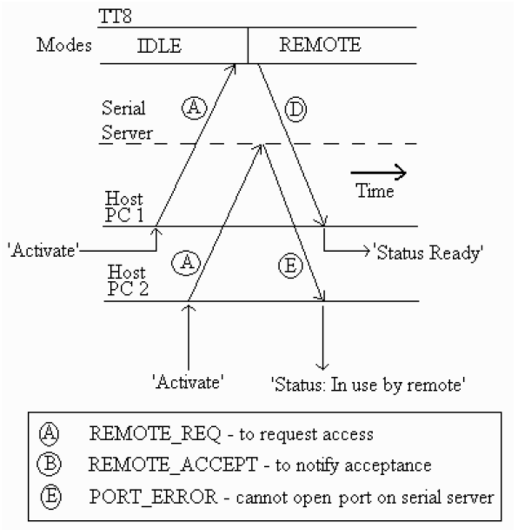

TT8
Modes
IDLE
REMOTE
Serial Server
A
D
Time
Host PC 1
'Activate'
'Status Ready'
Host PC 2
A
E
'Activate'
'Status: In use by remote'
A REMOTE_REQ - to request access
B REMOTE_ACCEPT - to notify acceptance
E PORT_ERROR - cannot open port on serial server

Fig. 4.10: Session denied due to multiple operators.

second operator attempts to access the Smart Wheel while it is already in remote mode. At any given time, the serial server can be used by only one host PC. When the application attempts to open the virtual COM driver to the serial server, an error is generated since the port was already opened by another operator.

4.4 Smart Wheel Networked Steering Speed Control

Random delays have serious effects in networked control systems, which deteriorate the performance and may even cause instability of the system. Hence, a controller which can make the plant stable at large values of delay is always desirable in NCS systems. Simulation results in Chapter 3 show that optimal fractional order proportional integral

(OFOPI) controllers have larger jitter margin (maximum value of delay for which system is stable) for lag-dominated systems as compared to traditional proportional integral derivative (PID) controllers, whereas integer order PID controllers have larger jitter margin for delay-dominated systems. In this section, a telepresence controller based on OFOPI tuning rules is used to obtain the maximum value of delay (jitter margin) at which the system will be stable. To illustrate this, an extensive experimental study on the real-time Smart Wheel networked speed control system is performed using hardware-in-the-loop control.

4.4.1 System Identification using Step Response Analysis

The first step is system identification of the the stand-alone Smart Wheel Assembly at CSOIS. For this a MATLAB code was written to enable communication between the remote computer and the Smart Wheel. The very first step in communication is to connect to the port on the Smart Wheel and obtain access to it. This is done by sending the REMOTE_REQ or ASCII character 47 to the Smart Wheel. The Smart Wheel responds with a REMOTE_ACCEPT or ASCII character 33 in its buffer as explained in sec. 4.3.2. Connection is verified if this character is received. The process of connection is done at the start of the simulation. A count is maintained and connection is performed only on the first ever operation. In the event of timeout or no receipt of connection code, the simulation shuts down. When connected, data is read in from the buffer on the Smart Wheel. Size of the data available is limited by the size of the buffer, longer data produces greater computational delay and noise when differentiated to get velocity, hence a proper data length selection is very important. The Smart Wheel reports data in a particular format, hence data is parsed to get the value of steering angle (θ) and time (t) from it.

Also an s-function is written which implements the same MATLAB code for communication (shown in Appendix C). The s-function block properly sizes the θ and t vectors so they are the same length and are synchronized. At the reach of stop-time, the s-function block issues a REMOTE_REQ command, ASCII character 92, which issues a stop command and the wheel stops spinning. The steering motor rotates in one direction when it receives ASCII values 70 to 90 (70 - Stopped, 90 Maximum Speed). It rotates in the opposite direction when it receives ASCII values 70 to 50 (70 - Stopped, 50 Maximum Speed). The control input lies in the range of 0 - 20 for either direction of rotation. Figure 4.11 shows the steering speed of Smart Wheel corresponding to a control input of 19. X-axis is the time in seconds and Y-axis is the steering speed of Smart Wheel obtained by taking derivative of θ with respect to t in radians/seconds.

Furthermore, it should be noted that the input issued to the motor over the network is in the range of (+/-1 to 19), with the signs representing the direction of rotation. Hence, since the input is not 1 or 5 volts, as is the case in a traditional step response system ID, the value of K obtained for the motor is low. A hardware-in-the-loop implementation of the Smart Wheel is done using an s-function block which handles communication between the plant and the remote computer. The sampling interval for the s-function block in Simulink is set equal to the time needed to read data from the buffer to attain synchronization.

From the step response, the transfer function for the first order plus delay time (FOPDT) model of the Smart Wheel is obtained as:

$$P(s) = \frac{0.1484}{0.045s + 1} e^{-0.592s}. \tag{4.1}$$

It is found that the Smart Wheel is a delay-dominated system with value of ($L > T$). Also for different locations of remote computers, the delay will vary, and hence our plant model. This also depends on the amount of traffic on the network at the time of

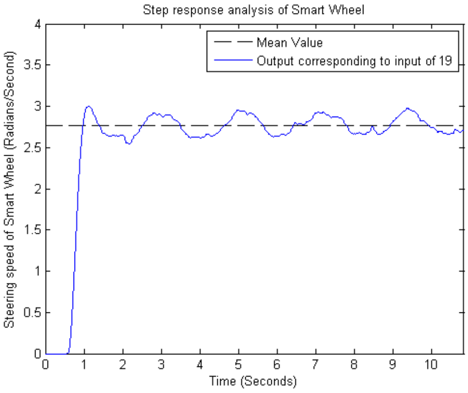


Fig. 4.11: Step response analysis of Smart Wheel.

experimentation.

It should be noted that the optimal fractional order proportional integral (OFOPI) tuning rules derived in Bhambhani et al. [37] were based on a set of FOPDT systems such that their delay values are $L = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]'$, values of the time constant are $T = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]'$ and $K = 1$, whereas the current plant system, i.e. Smart Wheel lies out of the range. Efforts were made to simulate another set of FOPDT systems with time delay and time constant values in the range $L = [0.1 \cdots 1]'$ and $T = [0.1 \cdots 1]'$. However due to computational limitations and frequency mismatch, this still remains a difficult task and is one of the major challenges. Similar difficulties were encountered for PID controller using the approach found in Eriksson and Johansson [38] for the new set of FOPDT systems, although the range of plants considered in that paper were $L = [0.1, 1 \cdots 10]'$, $T = [0.1, 1 \cdots 10]'$ and $Kp = 1$. To see how well the existing OPID/OFOPI tuning rules perform for the Smart Wheel system at different random delays, real-time experiments were conducted. It was noticed that the tuning rules still hold true i.e the networked system is still stable for delays less than jitter margin. However, the upper bound of jitter margin increases by 60 to 70%. This was concluded from simulating several systems lying in the new range.

$$\delta''_{max} = 1.6 * \delta_{max}, \tag{4.2}$$

where δ''_{max} is the jitter margin for new set of FOPDT systems at the same OFOPI/OPID tuning tuning parameters.

4.4.2 Laboratory Set up and Hardware-in-the-loop Control

This section gives a brief introduction about the implementation of optimal fractional order proportional integral (OFOPI) control of the steering axis speed of the Smart Wheel over the network loop.

Figure 4.12 shows a simple implementation of a OFOPI controller with the Smart Wheel in the loop. The block labeled smartwheel is the actual block which handles the entire process of communicating remotely with the hardware. Operational details of that block are illustrated with help of a flow chart shown in fig. 4.13.

A feedback control loop is used. The current velocity and the error between the setpoint

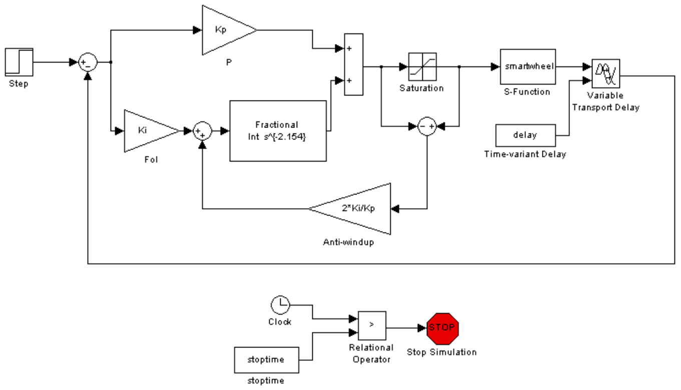


Fig. 4.12: Laboratory setup and hardware-in-the-loop.

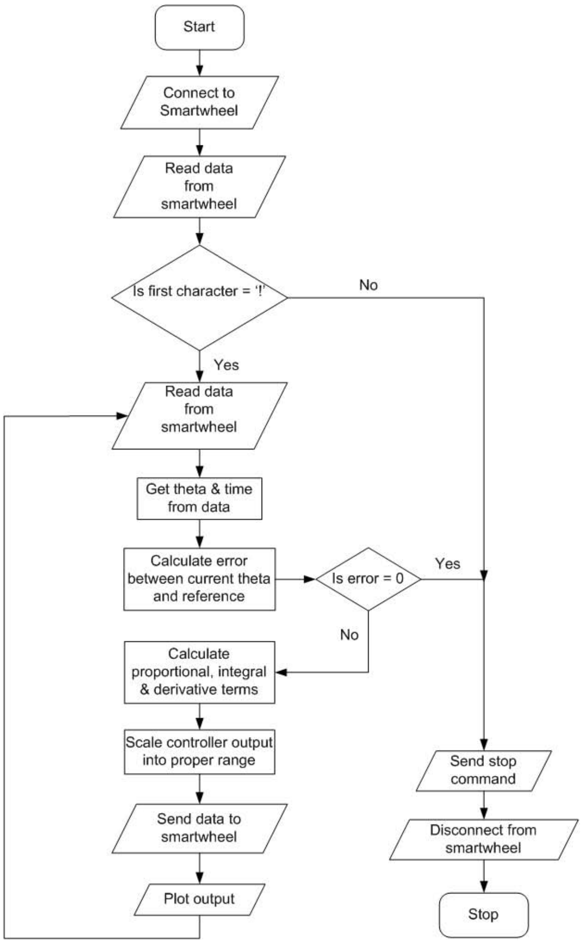

Start
Connect to Smartwheel
Read data from smartwheel
Is first character = '!'
No
Yes
Read data from smartwheel
Get theta & time from data
Calculate error between current theta and reference
Is error = 0
Yes
No
Calculate proportional, integral & derivative terms
Scale controller output into proper range
Send data to smartwheel
Plot output
Send stop command
Disconnect from smartwheel
Stop

Fig. 4.13: Flowchart describing working of Smart Wheel.

and the current velocity is obtained next. If the error is zero then the s-function block issues a $REMOTE\_REQ$ command, $ASCII$ character 92, which issues a stop command and the wheel stops spinning. For nonzero error, the controller gives some output. And then the results are plotted in MATLAB. As seen in fig. 4.12, a saturation block is placed in the loop which modulates the controller output to acceptable values, this necessitates the use of an anti-windup feedback block to the integrator which prevents integrator saturation. The correct values are fed into the s-function block which communicates with the Smart Wheel and sends appropriate commands. In each cycle, the s-function block reads data, parses data, calculates error, takes input from controller, sends command to Smart Wheel, and plots current received data to the screen.

4.4.3 Simulation Illustration

The next step in network control is the computation of controller gain parameters for the two controllers taken into consideration, i.e. OFOPI and OPID tuning rules. Two different sets of random time delays are obtained by pinging two different places located far apart from the Smart Wheel platform. These are Hua Zhong University of Science and Technology in China and The Australian National University in Australia. The gain parameters so obtained for different controllers are listed in Table. 4.1.

It should be noted that the jitter margin values for OPID are larger as compared to OFOPI controller, which is true as the Smart Wheel is a delay dominated system. Based on

Table 4.1: Gain parameters for different controllers.

			K_p	K_i	$\frac{\alpha}{K_d}$	JM
Aus	PID	AMIGO	1.524	0.43189	0.009323	1.8696
		OPID	0.8866	4.2004	0.03341	1.5596
	FOPI	FMIGO	2.1227	4.5671	1.1	0.2168
		OFOPI	0.1716	7.524	2.154	0.5550
China	PID	AMIGO	1.5015	0.38166	0.00952	2.1424
		OPID	0.91103	3.684	0.03341	1.7739
	FOPI	FMIGO	2.1079	3.9876	1.1	0.2186
		OFOPI	0.17015	6.5769	2.1561	0.6384

the above parameters, OFOPI and OPID controllers are used to control the steering speed of the Smart Wheel for different delays. Figure 4.14 show the simulated result for random delays obtained by pinging the university in China, whereas fig. 4.15 is for random delays obtained by pinging the Australian university.

The X-axis is the time in seconds and Y-axis is the speed of the wheel. Also $L(t)$ denotes the random delay and includes both the network delays and inherent delays due to computation and other miscellaneous factors, JM represents δ_{max} and JM'' represent δ''_{max}. It can be said that the OFOPI controller has a faster response as compared to OPID

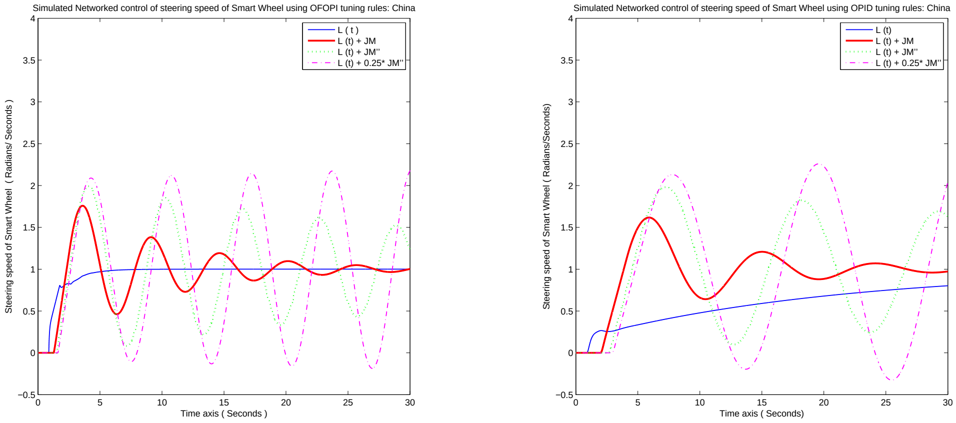


Fig. 4.14: Simulated network control of steering speed of Smart Wheel : China.

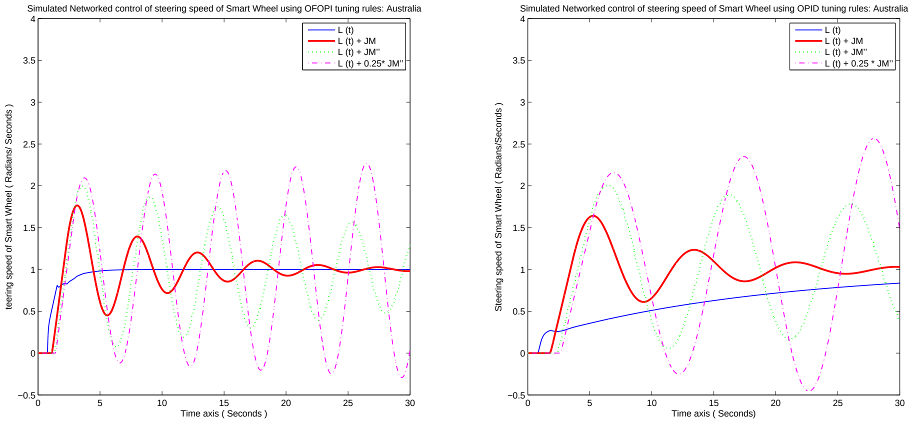


Fig. 4.15: Simulated network control of steering speed of Smart Wheel : Australia.

controller. Also the simulation results are as expected.

4.4.4 Experiment on the Real-time Smart Wheel Speed Control System

The final step in the design process of any control system is the control experiment. Figure 4.16 show the real-time results for random delays obtained by pinging the university in China, whereas fig. 4.17 is for random delays obtained by pinging the Australia university. Noise due to sensors and actuators are an inherent part of real-time network control unlike simulation results. Furthermore, oscillations are observed due to delays and physical phenomenon like suspension movements, frictional changes, and motor-cogging. Also, the motor velocity has a lower bound due to deadzone.

As can be seen, the OFOPI controller has a faster response as compared to the PID controller. Also for every case considered, the system becomes more and more unstable as the delay is increased and finally reaches instability at delay greater than δ''_{max}. Moreover, the results obtained in real-time are in confirmation with simulation results obtained in sec. 4.4.3.

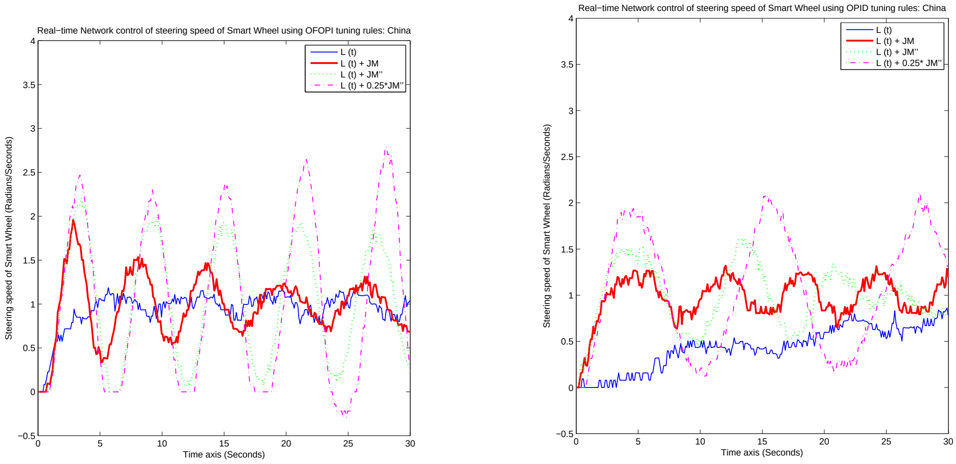


Fig. 4.16: Real-time network control of steering speed of Smart Wheel: China.

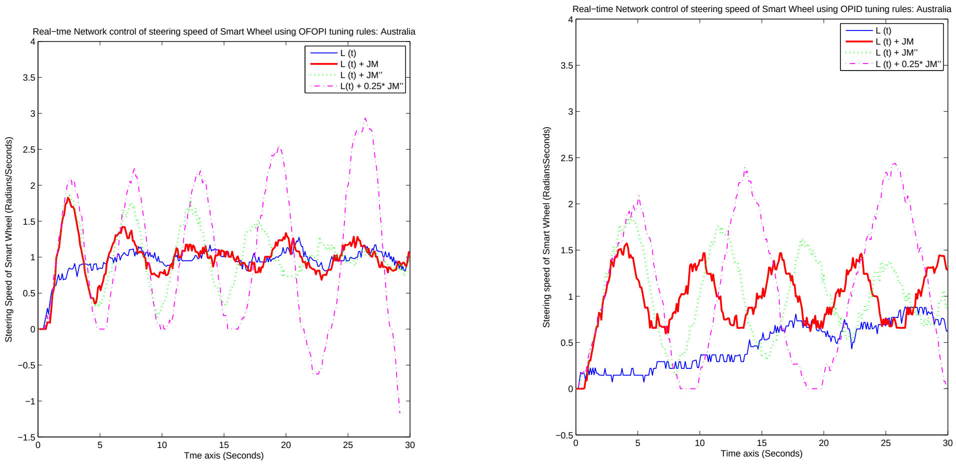


Fig. 4.17: Real-time network control of steering speed of Smart Wheel: Australia.

4.5 Conclusion

This chapter has presented an intensive study and experimental work on networked control of the steering speed of the stand-alone Smart Wheel platform at Center of Self Organizing and Intelligent Systems (CSOIS), Utah State University. Optimal fractional order proportional and integral (OFOPI) tuning rules are used to determine the parameters at which system has maximum delay and yet is stable. An s-function block is used in Simulink to implement the Smart Wheel (an omnidirectional robot) in the network loop. It is noticed that since the Smart Wheel is a delay-dominated system, it will result in higher jitter margin using an optimal proportional integral derivative (OPID) controller. This is in confirmation with the results obtained in the work of Varsha et al. [37]. However, for systems with L and T in the range $[0.1 \cdots 1]$, the upper bound on jitter margin is increased manyfold. Also, the OFOPI controller has a faster performance as compared to OPID controller. Hence, the work presented in this paper justifies that the OFOPI tuning rules work well in the networked control setting having random delays.

Chapter 5

Fractional-order Integral and Derivative Controller Design for Temperature Profile Control

5.1 Introduction

Closed-loop feedback temperature control is considered one of the most challenging design problems in control engineering. Temperature control arises in many engineering fields. For example in cryogenic applications [8], the precision spatial temperature control via spatial heating and spatial temperature sensing is important, and in process industry [39], the most common control task is to achieve the precise temperature profile. There are basically three types of temperature control tasks: temperature setpoint regulation, temperature profile tracking, and temperature uniformity control. For temperature setpoint control, single-loop commercial off-the-shelf temperature controllers can usually be used. Temperature profile tracking is to raise the temperature according to a prescribed temperature time history, which is required for applications such as precision heat treatment for materials, batch chemical reactors, etc. The temperature uniformity control is to achieve uniformly spatially distributed temperature profile. The objective of this work is to design a controller for temperature profile tracking at the spatially distributed places, thus, to regulate the temperatures at the given sensor locations to ensure the temperatures be close to the desired values.

For temperature control, usually, on/off control, proportional control, and traditional proportional integral derivative (PID) controller, have been widely used.[1] In this chapter, as a new control scheme for temperature regulation, we suggest using fractional-order PID controller for a more accurate temperature profile tracking of the spatially distributed heat

[1] See “http://www.omega.com/prodinfo/temperaturecontrollers.html”

flow. An actual experimental task is performed using the heat flow equipment (HFE) of Quanser.[2]

The organization of this chapter is as follows. Section 5.2 briefly summarizes role of integer order PID controller in temperature control and sec. 5.3 lists basic definitions in fractional order calculus and merits of fractional controller. This is followed by in-depth description and derivation of proposed analytical tuning method in sec. 5.4. Then, sec. 5.5 introduces the concepts of real-time control and hardware-in-the-loop. Detailed descriptions of the HFE platform, system analysis, and design of a fractional controller based on the analytical tuning approach are given. In sec. 5.6, an extensive comparison of existing integer order solutions and fractional order solutions is made. Finally, sec. 5.7 concludes this chapter with some remarks on the achieved results and ideas for future work.

5.2 Temperature Control

In temperature control, it is difficult to find the response time constant. Thus, usually, for the temperature control, on/off control scheme is used so as to regulate the output temperature within the dead band. However in this case, the output will be oscillatory around the setpoint, so an accurate temperature profile is not achieved. For a more accurate temperature regulation, closed-loop feedback control schemes are required. Most popular control scheme is PID control, because it does not require the plant model and practically it is easily implementable. In traditional PID control, the proportional term, integral term, and derivative term have different effects to the heat flow, temperature, and speed. For the temperature control, it is usually recommended to use full PID control, but with the accurately tuned control gains.[3] For the reliable temperature control, however there are some basic environmental requirements. For examples, the heater should apply enough power and the temperature sensors should be spatially distributed in appropriate places. Tuning the controller means that we select the proportional, integral, and derivative gains with a particular purpose. In fact, tuning PID gains for the temperature control requires

[2] See“www.quanser.com”

[3] See “http://www.w-dhave.inet.co.th/index/”

some physical interpretation about the system. Thus, it is necessary to understand the effects of the proportional, integral, and derivative terms to the system. In temperature control, individual PID gain has the following characteristics.

- Proportional gain: It requires more power proportional to the error between sensor temperature and the desired trajectory profile. The proportional gain is used for on/off control. That is, when the output is within the proportional band, the power is off, but when it is out of the dead band, the power is on. If the temperature is below setpoint, the output will be on longer; if the temperature is too high, the output will be off longer.[4]

- Integral gain: It provides a control signal that is proportional to the accumulated error. So, this integral term is for slow mode reaction and forces the steady-state error to zero for a step response. In temperature control, it adjusts the temperature to setpoint after stabilization.

- Derivative gain: It provides the control force proportional to the rate of change of the output error. So, this derivative term is for fast mode reaction and yields large signal with the high-frequency control errors and with the rise or fall of system temperature.

From a literature survey, it is shown that there have been numerous applications of PID controller or fuzzy/neural network-based PID controller for the temperature control of various engineering objects. As some examples, Peter Galan showed that a fuzzy logic for enhancing PID controller is necessary for the satisfactory temperature profile tracking of injection moulding processes[5] and Dihac et al. used PID controller for a rapid thermal processor control [40]. Lin et al. proposed a neural fuzzy inference network for the temperature control of a water bath system and compared the performance with the PID control [41]. Juang and Chen proposed TSK-type recurrent neural fuzzy network based on the direct inverse control configuration, which does not require the priori knowledge of the plant order [42] and Ramos et al. used PID controller to control the temperature of

[4] See“http://www.omega.com/prodinfo/temperaturecontrollers.html’

[5] See“http://www.manufacturing.net/ctl/article/CA408369.html”

the bath [43]. However, there has been no trial of using fractional-order PID controller for the temperature control of the spatially distributed system. In the next section, we briefly summarize fractional-order $I^\alpha D^\beta$ control and its benefits in temperature control.

5.3 Fractional Order $I^\alpha D^\beta$ Control

After Newton and Leibniz discovered calculus in the 17th century, fractional order calculus has been studied as an alternative calculus in mathematics [7]. As claimed in Chen et al. [20], fractional order calculus will play an important role in mechatronic and biological systems.

Fractional order dynamic system and controls are relatively new research areas in control engineering. From early 90's, there has been steadily research in these areas as shown in Oustaloup et al., Raynaud et al., Vinagre et al., Podlubny et al., Ortigeira et al. [44–51] even though some pioneering works can be traced back to Manabe, Oustaloup and Axtell et al. [14–17].

Traditional PID control method is a most popular control approach where integrator and derivative are integer order. Recently, in fractional order calculus community, a trend of using non-integer integrator or non-integer derivative for the accurate profile tracking in controlled-output has appeared, which is so-called fractional order PID control. In the following subsections, we briefly review this fractional order PID control.

5.3.1 Fractional Order Calculus

In this chapter, fractional integral is defined as:

$$I^\alpha f(t) = \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} f(\tau) d\tau, \ t > 0, \ \alpha \in \mathbf{R}^+, \tag{5.1}$$

and for the fractional derivative, Caputo derivative is used, which is defined as:

$$\begin{aligned} D^\alpha f(t) &= \frac{d^\alpha f(t)}{dt^\alpha} = \frac{1}{\Gamma(\alpha - n)} \int_0^t \frac{f^{(n)}(\tau)}{(t-\tau)^{\alpha+1-n}} d\tau, \\ & \qquad (n-1) < \alpha \le n, \end{aligned} \tag{5.2}$$

where Euler's Gamma function is given as:

$$\Gamma(x) = \int_0^\infty e^{-t} t^{x-1} dt,\ x > 0, \tag{5.3}$$

with the special case when $x = n$:

$$\Gamma(n) = (n-1)(n-2)\cdots(2)(1) = (n-1)!.$$

Now, using the Laplace transformation, we have $s^{-\mu}F(s) = I^\alpha f(t),\ \mu = \alpha$, and $s^\lambda F(s) = \frac{d^\lambda f(t)}{dt^\lambda}$. Then, the fractional PID controller can be written as:

$$C(s) = K_p + K_i s^{-\mu} + K_d s^\lambda. \tag{5.4}$$

If we take $\mu = \lambda = 1$, then we obtain the classic PID controller; with $\mu = 0$, it is the PD controller; and if $\lambda = 0$, it is the PI controller. Also, we will consider fractional order $I^\alpha D^\beta$ controller which is an special case of fractional PID controller with $K_p = 0$.

5.3.2 Merit of Using Fractional Order Controller

As explained in Kostial et al. [52], the idea of using fractional order controllers for the dynamic system control belongs to Oustaloup et al. [46, 53] and generalized fractional order proportional integral derivative (PID) controller was proposed by Podlubny [54]. Advantages of using fractional order PID controller have been introduced in a number of publications. In Vinagre et al. [55], some observations about fractional order PID control were given. Particularly, by varying fractional order of integrator (α) from 1 to ∞, it was shown that there could be a constant increment in the slope of the magnitude curve varying between -20 *db/dec* and 0 *db/dec* and could be a constant delay in the phase response varying between $-\frac{\pi}{2}$ and 0. Similarly, by changing fractional order of derivative (λ) from zero to 1, we can change the amount of phase lead and the slope of magnitude response. Vinagre et al. provided frequency domain analysis to illustrate the superiority of the fractional order PID controller applied to both the fractional dynamic system and the integer dynamic system [56]. In Podlubny et al. [52], it was claimed that fractional order PID controller is an adequate controller for the fractional order mathematical models and it is less sensitive to shifts of parameters of a controlled system and to variations of parameters of the controller. Particularly, in work of Podlubny [45], it was illustrated that the fractional order PID controller is a suitable way for the control of the fractional system. Hwang et al. [57] proposed the fractional order band-limited compensator, which has the similar response as the fractional-order PD controller but has the less sensitivity to the high frequency noise. Leu et al. [58] provided a differential evolution algorithm to search for optimal fractional order PID parameters to meet the phase margin and the gain margin specifications. Thus, fractional order controller could be beneficial for temperature control since there are often variations in parameters in heat flow systems, and most of desired specifications, which are not readily achieved simultaneously by traditional PID controller, can be ensured by fractional order closed-loop dynamics.

5.4 Analytical Tuning Method

If the system model is given, it is easy to design a controller to control the system. However, it is not easy to design an appropriate controller for a system whose transfer function is time varying and is highly dependent on the environmental change like temperature control. Hence, it is actually quite tough to find a transfer function of the output temperature change from the power input, because the temperature of the environment is time-dependent. Furthermore, as shown in work of Petras et al. [59], the transient unit response of the heat solid system could be fractional order and as noted in Aoki et al. [60] also, heat transfer coefficients may be fractional or non-integer order. Thus, the use of fractional order differential model may approximate well the time-dependent behavior of conductive systems of complex geometries with convective heat transfer. In fact, since thermal loads involve thermal diffusion, a half-order controller represents a better match to the physics of the plant to be controlled as shown in work of Bohannan [61]. Therefore for the temperature control, it may not a good idea to design a controller with a fixed integer model, and since the response of the temperature output from the power input could be fractional, the traditional integer-order controller may not provide an accurate temperature profile tracking. Thus, this chapter proposes using the fractional-order $I^{\alpha}D^{\beta}$ controller for the accurate temperature profile tracking, which will be experimentally demonstrated by the Quanser heat flow equipment (HFE).

Let the plant be represented by a first order plus time delay (FOPTD) transfer function as:

$$G(s) = \frac{K}{Ts+1}e^{-Ls}, \tag{5.5}$$

where K is the static gain or steady state gain of the system, L is the time-delay, and T is the time-constant of the system. These can be computed from the open-loop step response of the system.

The fractional controller is represented by $C(s)$ as:

$$C(s) = K_i \frac{Ts+1}{s^{\alpha}}. \tag{5.6}$$

Also,

$$C(s) = K_i(\frac{1}{s^{\alpha}} + Ts^{1-\alpha}). \tag{5.7}$$

Thus, the above equation reveals the fractional integral derivative nature of the controller. Hence; the controller is $(FO - ID/I^{\alpha}D^{\beta})$ controller where $\beta = 1 - \alpha$.

The open-loop transfer function as Bode's ideal loop transfer function is given by:

$$C(s)G(s) = K_i K \frac{e^{-Ls}}{s^{\alpha}}......1 < \alpha < 2. \tag{5.8}$$

Now, the gain and phase of the open-loop transfer function are given by:

$$|C(j\omega)G(j\omega)| = \frac{K_i K}{\omega^{\alpha}}, \tag{5.9}$$

$$\phi = -L\omega - \alpha\frac{\pi}{2}.$$

For a user-defined gain margin A_m and phase margin ϕ_m, above relations are expressed as:

$$|C(j\omega_p)G(j\omega_p)| = -\frac{1}{A_m}, \tag{5.10}$$

$$\phi_m = \pi + \angle C(j\omega_g)G(j\omega_g),$$

where ω_p and ω_g are the phase and gain crossover frequencies of the open-loop system with controller, respectively.

On substitution, the gain relations are:

$$\frac{KK_i}{\omega_g^\alpha} = 1, \tag{5.11}$$

$$\frac{\omega_p^\alpha}{KK_i} = A_m,$$

and the phase relations are:

$$\phi_m = \pi - L\omega_g - \alpha\frac{\pi}{2}, \tag{5.12}$$

$$\pi = \alpha\frac{\pi}{2} + L\omega_p.$$

Further,

$$A_m = (\frac{\omega_p}{\omega_g})^\alpha, \tag{5.13}$$

$$\frac{\omega_p}{\omega_g} = \frac{\pi - \alpha\frac{\pi}{2}}{\pi - \phi_m - \pi\frac{\pi}{2}}.$$

Simplifying one obtains the nonlinear equation represented by 13 as:

$$A_m = (\frac{\pi - \alpha\frac{\pi}{2}}{\pi - \phi_m - \pi\frac{\pi}{2}})^\alpha. \tag{5.14}$$

Given a gain margin/ phase margin, above equation is solved for fractional order α using numerical classical approach in MATLAB.

Once the value of α is obtained, corresponding values of $(\omega_g, \omega_p, K_i, K_d)$ are obtained as:

$$\omega_g = \frac{\pi - \phi_m - \alpha\frac{\pi}{2}}{L}, \tag{5.15}$$

$$\omega_p = \frac{\alpha - \alpha\frac{\pi}{2}}{L},$$

$$K_i = \frac{\omega_p^\alpha}{KA_m} = \frac{\omega_g^\alpha}{K},$$

$$K_d = TK_i.$$

Thus, given K, L, T, A_m, ϕ_m, one can determine $\alpha, \omega_g, \omega_p, K_i, K_d$ using above analytical approach.

5.5 Experimental Test

Quanser HFE system (see fig. 5.1) consists of a duct equipped with a heater and a blower at one end and three temperature sensors located along the duct. The power delivered to the heater is controlled using an analog signal. For the analog signal generation and measurement, we use Quanser analog input/output libraries via MATLAB/Simulink/RTW. The fan speed of heat flow equipment (HFE) is controlled using an analog signal. Fan speed is measured using a tachometer and is an input signal. Figure 5.1 shows the set of test equipment: computer, software, analog-to-digital/digital-to-analog (AD/DA) converter and Quanser HFE. HFE system includes built-in power module, analog signals for fan speed

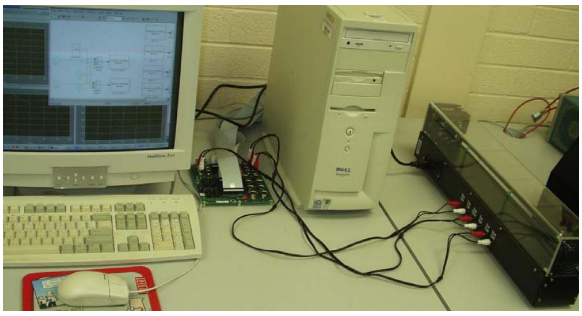


Fig. 5.1: The set of equipment for heat flow test.

and power, an onboard tachometer to design speed control, and fast settling platinum temperature transducers (three sensors along the duct) to measure the temperature. This section summarizes the results of open-loop step response analysis. Figures 5.2, 5.3, and 5.4 are the open-loop step response of sensor 1, 2, and 3 at varying heater (VQ) voltages but constant fan (VF) voltage of $5V$ DC. The fan (VF) and heater (VQ) voltages are applied in the range [0-5 Volts] and the temperature readings at three sensor positions are recorded.

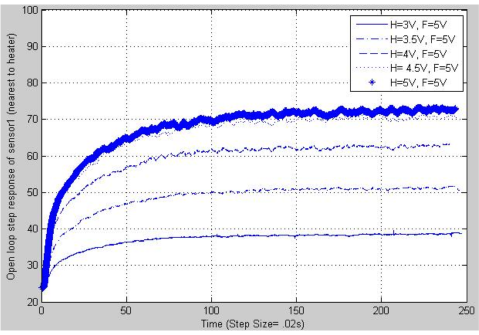


Fig. 5.2: Sensor 1 (Fan Command Voltage constant= 5V, Heat Command Voltage varying).

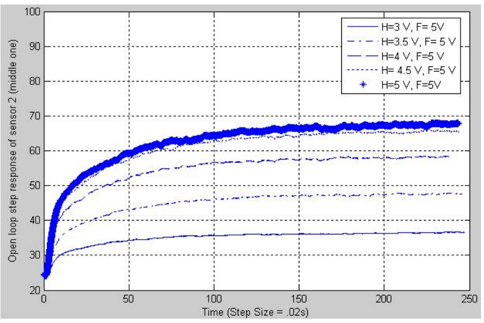


Fig. 5.3: Sensor2 (Fan Command Voltage constant= 5V, Heat Command Voltage varying).

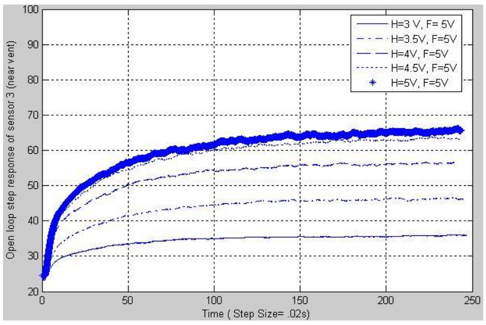


Fig. 5.4: Sensor3 (Fan Command Voltage constant= 5V, Heat Command Voltage varying).

The heating operation takes effect only when $VQ > 3VDC$. The open-loop response shows an initial delay, indicating that it takes finite time for the sensors to detect the change in the temperature.

A first order plus delay time (FOPDT) model at a desired sensor is given by:

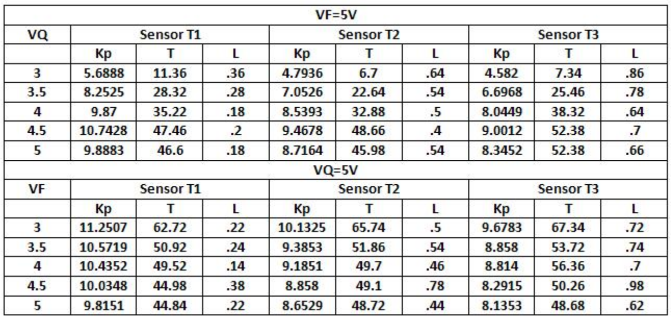
$$\frac{T_n}{V_q} = \frac{K}{Ts+1}e^{-Ls}, \tag{5.16}$$

where K is the open-loop gain given by $(Max - Min)/VQ$ in (C/V).
Min: Temperature in (C) inside the tube before the experiment begins or the room temperature.
Max: Maximum temperature in (C) reached after a run of 60 seconds. This is, strictly speaking, not the steady state value as the temperature is increasing. However, the rate of increase considerably slows down after 20 seconds.
L : The delay in the response in seconds.
T : The time constant is given by the time it takes to reach 63% of the change in temperature or the time it takes for the response to reach $Min + .63 * (Max - Min)$.
τ :The relative dead time of the response given by $L/(L+T)$.

Experiments were conducted to study the open-loop step response of three sensors first at constant fan voltage ($5V$) and varying heater voltage ranging from $3 - 5$V. The above experiment was repeated for constant heater voltage ($5V$) and varying fan voltage ranging from $3 - 5$V. The step response values so obtained are listed in fig 5.5.

Firstly when heating effect is analyzed by keeping the fan speed constant, it is seen that as the distance of the sensors from heater increases, the corresponding gain values of the first order transfer function models of the sensors decrease and delay time increase i.e., $K1 > K2 > K3$ and $D1 < D2 < D3$, where $K1$, $K2$, and $K3$ are the static gain values and $D1$, $D2$, and $D3$ are the delay values of first order transfer function models of sensor 1, sensor 2, and sensor 3, respectively. This is shown in fig. 5.6 and fig. 5.7. There is no trend in time constant of the three sensors with distance. However for every sensor, time constant values show an increase with increase in heater voltage. This is shown in fig. 5.8.

Cooling effect is comparatively different from heating effect. This is shown in figs. 5.9, 5.10, and 5.11. As can be seen, gain values decrease with increase in distance from the fan and with increase in fan voltage, i.e. $K1 > K2 > K3$, whereas the delay values show an increase with distance from the blower, i.e. $D1 < D2 < D3$. Also, there is an increase in time constant values of sensors with increase in distance from fan, i.e. $T1 < T2 < T3$.



VF=5V									
VQ	Sensor T1			Sensor T2			Sensor T3		
	Kp	T	L	Kp	T	L	Kp	T	L
3	5.6888	11.36	.36	4.7936	6.7	.64	4.582	7.34	.86
3.5	8.2525	28.32	.28	7.0526	22.64	.54	6.6968	25.46	.78
4	9.87	35.22	.18	8.5393	32.88	.5	8.0449	38.32	.64
4.5	10.7428	47.46	.2	9.4678	48.66	.4	9.0012	52.38	.7
5	9.8883	46.6	.18	8.7164	45.98	.54	8.3452	52.38	.66
VQ=5V									
VF	Sensor T1			Sensor T2			Sensor T3		
	Kp	T	L	Kp	T	L	Kp	T	L
3	11.2507	62.72	.22	10.1325	65.74	.5	9.6783	67.34	.72
3.5	10.5719	50.92	.24	9.3853	51.86	.54	8.858	53.72	.74
4	10.4352	49.52	.14	9.1851	49.7	.46	8.814	56.36	.7
4.5	10.0348	44.98	.38	8.858	49.1	.78	8.2915	50.26	.98
5	9.8151	44.84	.22	8.6529	48.72	.44	8.1353	48.68	.62

Fig. 5.5: Step response data from the open-loop heat flow experiment.

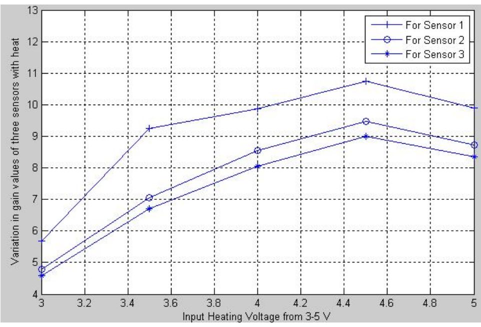


Fig. 5.6: Heating effect: Variation of static gain of sensors with distance.

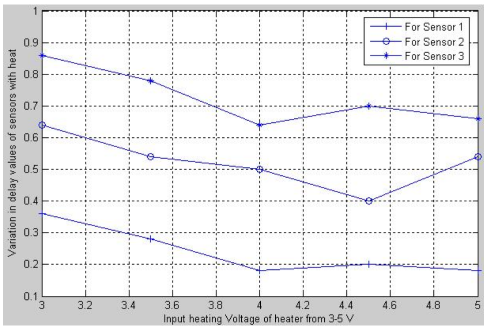


Fig. 5.7: Heating effect: Variation of delay of sensors with distance.

Here $T1$, $T2$, and $T3$ are time constant of sensor 1, sensor 2, and sensor 3, respectively. Thus, one can see that dynamics of HFE is really very complex and permits the need of a better controller than a simple integer order proportional integral/ proportional integral derivative (PI/PID) controller. The next section aims at justifying the efficiency of the new method over already existing integer and fractional order controllers.

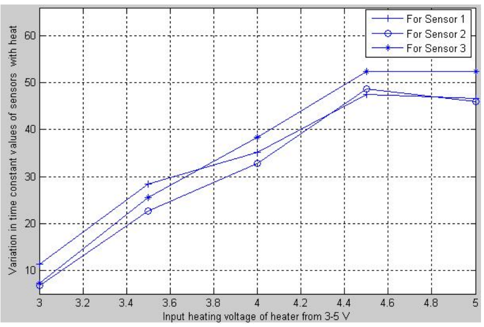


Fig. 5.8: Heating effect: Time constant of sensors with distance.

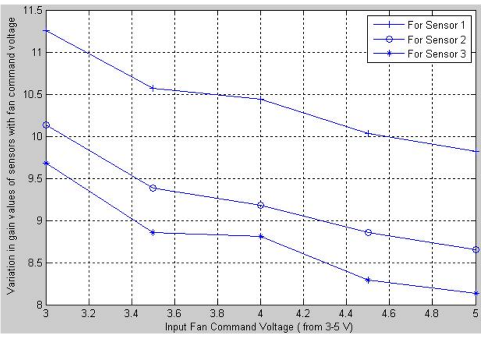


Fig. 5.9: Cooling effect: Variation of static gain of sensors with distance.

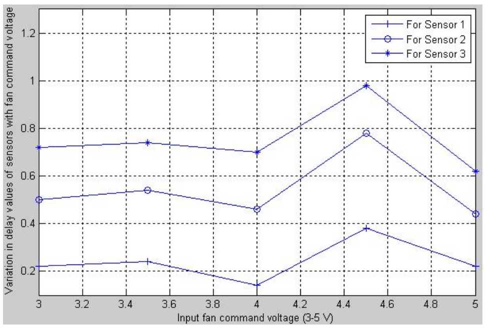


Fig. 5.10: Cooling effect: Variation of delay of sensors with distance.

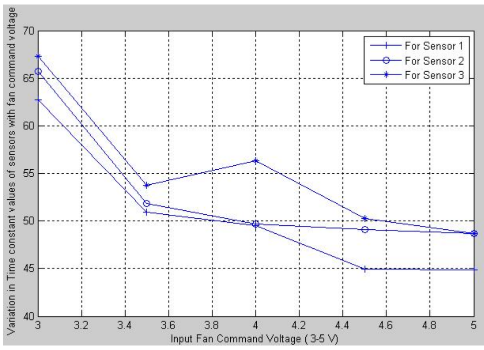


Fig. 5.11: Cooling effect: Variation of time constant of sensors with distance.

5.6 Comparisons and Simulation Results

In this section an extensive comparison of existing integer order solutions and fractional order solutions is made. For integer order case both PI and PID controllers are considered based on Ziegler Nichol's tuning method. The ziegler nichol's tuning formulae for a PI controller is given as:

$$K_p = \frac{0.9T}{KL}, \tag{5.17}$$

$$K_i = \frac{K_p}{3L},$$

whereas for a PID controller the gain values are:

$$K_p = \frac{1.2T}{KL}, \tag{5.18}$$

$$K_i = \frac{K_p}{2L},$$

$$K_d = K_p * \frac{L}{2}.$$

The other controller, which is compared, is a fractional PI controller based on fractional Ms integral gain optimization (F-MIGO) tuning method introduced in Kostial et al. and Bhaskaran et al. [6, 7] as:

$$C(s) = K_p^{\clubsuit} + \frac{K_i^{\clubsuit}}{s^{\alpha}}, \tag{5.19}$$

where $K_p^{\clubsuit}$ and $K_i^{\clubsuit}$ and α values are given by:

$$K_p^{\clubsuit} = \frac{0.2978}{K(\tau + .000307)}, \tag{5.20}$$

$$K_i^{\clubsuit} = T\frac{.8578}{\tau^2 - 3.402\tau + 2.405},$$

$$\alpha = \begin{cases} 0.7, & \text{if } \tau < 0.1 \\ 0.9, & \text{if } 0.1 \le \tau < 0.4 \\ 1.0, & \text{if } 0.4 \le \tau < 0.6 \\ 1.1, & \text{if } \tau \ge 0.6 \end{cases}.$$

The controller performances for all the three sensors are compared in real-time as shown in figs. 5.12, 5.13, and 5.14. As is clear from the figures, the fractional controllers outperform simple integer order PI/PID controller. The new controller fractional order integral derivative ($FO - ID$) is tuned better in comparison to Ms constrained FO-PI controller for sensor 1, whereas for sensor 2 and sensor 3 both the fractional order controllers perform more or less the same.

Figures 5.15, 5.16, and 5.17 show the same results in Simulink enviornment. The simulation and actual lab results showed some mismatch indicating that the linear model was unable to model the systems nonlinearities. In the experiments the fan and heater

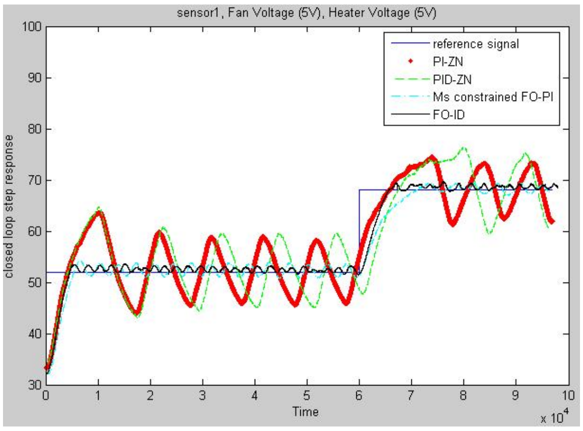


Fig. 5.12: Controller performances for sensor 1 in real-time.

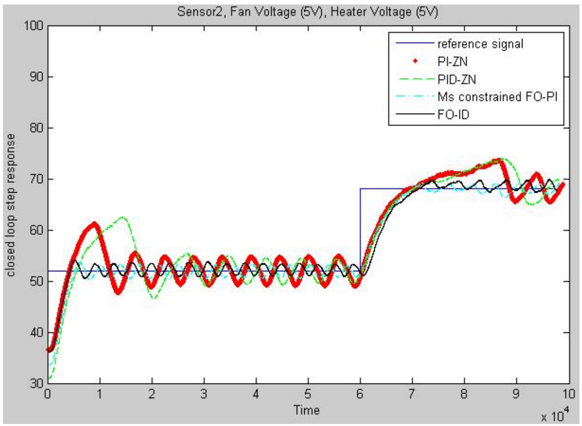


Fig. 5.13: Controller performances for sensor 2 in real-time.

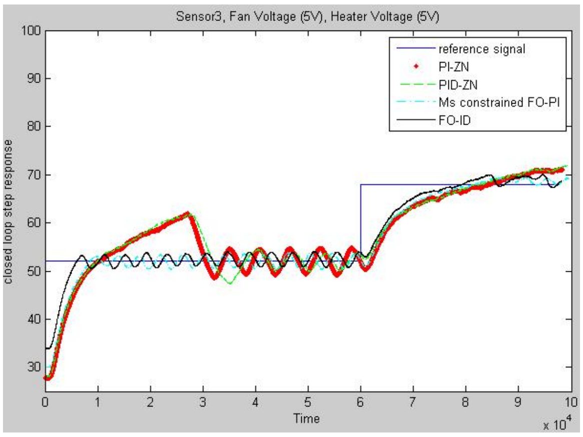


Fig. 5.14: Controller performances for sensor 3 in real-time.

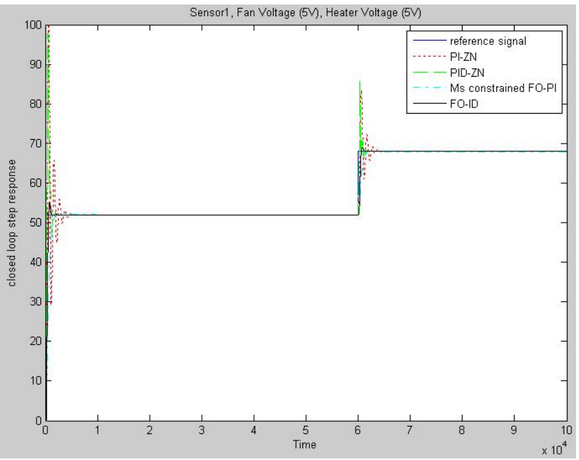


Fig. 5.15: Controller performances for sensor 1 in Simulink.

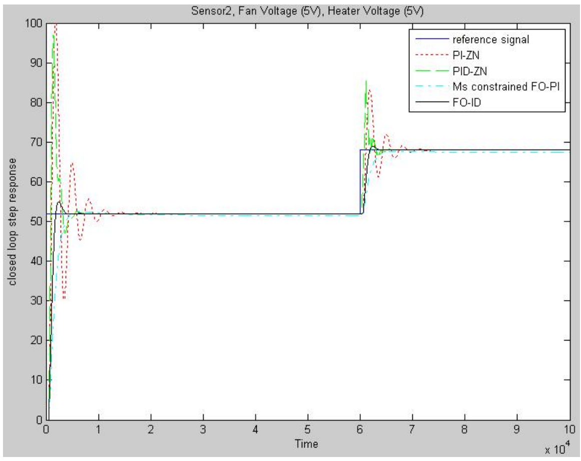


Fig. 5.16: Controller performances for sensor 2 in Simulink.

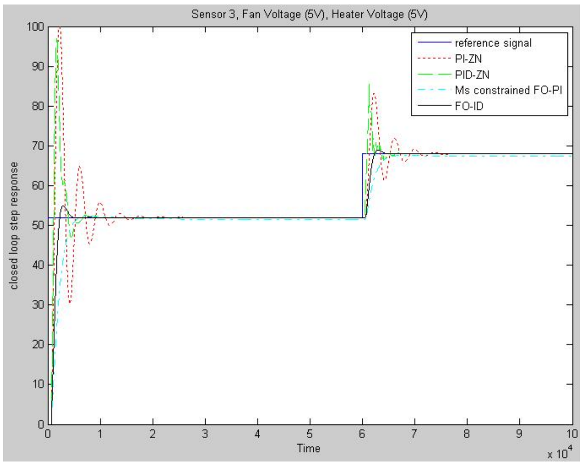


Fig. 5.17: Controller performances for sensor 3 in Simulink.

voltages were $5V$ each.

5.7 Conclusions

The objective of this chapter to introduce a novel analytical tuning method for a fractional integral derivative ($FO - ID/I^{\alpha}D^{\beta}$) controller and apply the results to analyze heat flow in heat flow equipment (HFE) module was successfully met. The performance of new FO-ID controller was compared to the integer order proportional integral/proportional integral derivative (PI/PID) controller and fractional order proportional integral (FO-PI) controller. Our observation was that since the HFE systems have very small relative dead time they do not need a full integrator for their control. FO-PI controller gives a strong competition to FO-ID controller tuned by new analytical method for some cases. For FO-ID method introduced, the resulting closed-loop system has the desirable feature of being robust to gain variations with step responses exhibiting an iso-damping property. Extensive simulation results are included to illustrate the simple yet practical nature of the developed new tuning rules.

Chapter 6

Conclusion

6.1 Summary

In summary, the contributions and results presented in this thesis are as follows.

A detailed explanation of design of a robust-jitter controller called optimum fractional order proportional integral (OFOPI) controller was provided. A set of hundred gain delay time-constant (KLT) systems was simulated and compared to prove the efficiency of controller in providing higher jitter margin when compared to simple fractional order proportional integral (FO-PI) controller and proportional integral derivative (PID) controller (for relative dead time (τ) < 0.5. Finally, tuning rules were given to determine the gain parameters and fractional order of integrator (α) of OFOPI controller. This kind of controller could prove to be a better option than optimal proportional integral derivative (OPID) controller for systems with small value of τ and when large jitter margin and better controller performance are desirable.

Present work considers a special case when varying time delay $\delta(t) = \delta_{max}$, for all values of time t. In other words, $\| \Delta_F \|_{L_2} = \delta_{max}$. For such a case, the tuning rules give the gain parameters of the OFOPI controller at which the jitter margin is maximized for the system. In other cases when $\delta(t) \neq \delta_{max}$, these tuning rules result in higher jitter margin bound. Furthermore, an intensive experimental study for such a system called the Smart Wheel at the Center of Self Organizing and Intelligent Systems (CSOIS) was presented.

For speed control in Smart Wheel experiment, random delays were accounted by pinging different locations and using OFOPI tuning rules to determine the parameters at which system has maximum delay and yet is stable. An s-function block is used in Simulink to implement the Smart Wheel in the network loop. It is noticed that since the Smart Wheel is a delay-dominated system, it will result in higher jitter margin using an OPID controller.

This is in confirmation with the results obtained in the work of Bhambhani et al. [37]. However, for systems with delay (L) and time-constant (T) in the range $[0.1 \cdots 1]$, the upper bound on jitter margin is increased manyfold. Also the OFOPI controller has a faster performance as compared to OPID controller. Hence, the work presented in this paper justifies that the OFOPI tuning rules work well in the networked control setting having random delays.

Another major contribution of this thesis was the experimental work on water-level and fluid control in coupled tank PP-100 in three different configurations. A comparison between different PI controller tuning methods was made for each tank configuration. This was a four-step procedure comprising of system plant modeling, identification, validation, simulation, and real-time control experiments. For each case, simulation results confirmed real-time control results though disturbances due to water dynamics, modeling uncertainties, nonlinearities in the system, etc., are still responsible for differences between the two. Furthermore, it was observed that FO-PI (FMIGO) is a promising controller in process industries and could perform even better when compared with integer order PI controllers.

Besides, a novel analytical tuning method for a fractional integral derivative ($FO-ID/I^{\alpha}D^{\beta}$) controller was introduced and results were applied to analyze heat flow in heat flow equipment (HFE) from Quanser. The performance of new FO-ID controller was compared to the integer order PI/PID controller and FO-PI controller. Our observation was that since the HFE systems have very little relative dead time, they do not need a full integrator for their control. FO-PI controller gave a strong competition to FO-ID controller tuned by new analytical method for some cases. For FO-ID method introduced, the resulting closed-loop system had the desirable feature of being robust to gain variations with step responses exhibiting an isodamping property. Extensive simulation results were included to illustrate the simple yet practical nature of the developed new tuning rules.

6.2 Future Work

It should be noted that the OFOPI tuning rules derived in Bhambhani et al. [37] were based on a set of first order plus delay time (FOPDT) systems such that their delay values are $L = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]'$, values of the time constant are $T = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]'$, and $K = 1$. Efforts were made to simulate another set of FOPDT systems with time delay and time constant values in the range $L = [0.1 \cdots .1]'$ and $T = [0.1 \cdots 1]'$. However due to computational limitations and frequency mismatch, this still remains a difficult task and is one of the major challenges. Similar difficulties were encountered for PID controller using the approach found in Eriksson and Johansson [38] for the new set of FOPDT systems.

Another possible area of research could be to find the upper bound of the controller when δ(t)=δ_{max}, for all values of t. In other words, $\| \Delta_F \|_{L_2}$=δ_{max}.

Use of higher order filters to obtain any desirable jitter-margin using FO-PI design would also prove to be an area to explore as shown in work of Eriksson and Johansson [9]. A tuning scheme based on the simple fractional Ms constrained integral gain optimization (F-MIGO) tuning on an extended plant should be developed. This method opens door for treating the maximum time-delay as a design parameter for the tuning rules.

The tuning rules help in designing a OFOPI controller, extension to include a fractional derivative term is another area of research.

References

[1] P. Marti, *Analysis Design for Real Time Control Systems with Varying Control Timing Constraint.* Ph.D. dissertation, Department de Enginyeria de Sistems, Universitat Politecnicc de Catalunya, Spain, 2002.

[2] A. Cervin, *Towards the Integration of Control and Real Time Scheduling Design.* PhD. dissertation, Department of Automatic Control, Lund Institute of Technology, Sweden, 2000.

[3] M. Wu, Y. He, and J. She, "Delay-dependent criteria for the robust stability of systems with time varying delay," *Journal of Control Theory and Applications*, pp. 97–100, Nov. 2003.

[4] V. N. Phat and P. Niamsup, "Stability of linear time-varying delay systems and applications," *Journal of Computational and Applied Mathematics*, Oct. 2006.

[5] C. Kao and A. Rantzer, "Stability analysis of systems with uncertain time-varying delays," *Automatica*, vol. 43, pp. 959–970, 2007.

[6] I. Koštial, P. Kmetek, J. Prokop, M. Olejár, and L. Dorčák, "On material evaluations and fractality in modelling a blast furnace process," in *Proceedings of the ASRTP'92, The 10th conference with international participation*, pp. 333–336, Zlatá Idka, ČSFR, Dec. 2007.

[7] T. Bhaskaran, Y. Chen, and G. Bohannan, "Practical tuning of fractional order proportional and integral controller (2): experiments," in *Proceedings of ASME 2007 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, IDETC/CIE 2007*, Las Vegas, NV, USA, Sept. 2007.

[8] L. Eriksson and M. Johansson, "PID controller tuning rules for varying time delay systems," in *Proceeding of American Control Conference*, pp. 619–625, New York, USA, 2007.

[9] L. M. Eriksson and M. Johansson, "Simple PID tuning rules for varying time-delay systems," *IEEE Conference on Decision and Control*, 2007.

[10] T. Bhaskaran, *Practical Tuning Method for Fractional Order Proportional and Integral Controlers.* Master's thesis, Utah State University, 2007.

[11] L. Debnath, "A brief historical introduction to fractional calculus," *International Journal of Science and Mathematics Education*, vol. 35, no. 4, pp. 487–501, 2004.

[12] R. L. Magin, "Fractional calculus in bioengineering," *Critical Reviews in Biomedical Engineering*, vol. 32, no. 1-4, 2004.

[13] D. Xue, C. N. Zhao, and Y. Q. Chen, "Fractional order PID control of a DC motor with an elastic shaft: a case study," in *Proceedings of American Control Conference*, pp. 3182–3187, June 2006.

[14] S. Manabe, "The non-integer integral and its application to control systems," *Transactions of Institute of Electrical Engineers of Japan*, vol. 80, no. 860, pp. 589–597, 1960.

[15] S. Manabe, "The system design by the use of non-integer integral and transport delay," *Transactions of Institute of Electrical Engineers of Japan*, vol. 81, no. 878, pp. 1803–1812, 1961.

[16] A. Oustaloup, "Linear feedback control systems of fractional order between 1 and 2," in *Proceedings of the IEEE Symposium on Circuit and Systems*, Chicago, USA, Apr. 1981.

[17] M. Axtell and E. M. Bise, "Fractional applications in control systems," in *Proceedings of the IEEE 1990 National Aerospace and Electronics Conference*, pp. 563–566, New York, USA, 1990.

[18] Y. Q. Chen, "Ubiquitous fractional order controls," in *Proceedings of The Second IFAC Symposium on Fractional Derivatives and Applications (IFAC FDA06, Plenary Paper.)*, pp. 19–21, July 2006.

[19] Y. Q. Chen, B. M. Vinagre, and I. Podlubny, "Continued fraction expansion approaches to discretizing fractional order derivatives - an expository review," *Nonlinear Dynamics (Springer)*, vol. 38, no. 1-2, pp. 155–170, 2004.

[20] Y. Q. Chen, D. Xue, and H. Dou, "Fractional calculus and biomimetic control," *Proceedings of the First IEEE International Conference on Robotics and Biomimetics (RoBio04)*, Aug. 2004.

[21] T. Bhaskaran, Y. Chen, and D. Xue, "Practical tuning of fractional order proportional and integral controller (1): tuning rule development," *Proceedings of ASME 2007 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, IDETC/CIE 2007*, pp. 265–271, Sept. 2007.

[22] V. Ramakrishnan, Y. Zhuang, S. Y. Hu, J. P. Chen, C. C. KO, B. M. Chen, and K. C. Tan, "Development of a web-based control experiment for a coupled tank apparatus," *IEEE Transactions On Educations*, vol. 44, no. 1, Feb. 2001.

[23] T. Sang, M. Khalid, and R. Yusof, "Tuning of neuro-fuzzy controller by genetic algorithm with an application to a coupled-tank level control system," *IEEE Transactions on Systems, Man and Cybernetics*, 1999.

[24] "Coupled-tank control apparatus PP-100" [Online]. Available: `http://www.kri.com.sg/ctank.html`.

[25] "Quanser" [Online]. Available: `http://www.quanser.com/choice.asp`.

[26] E. Laubwald, "Coupled tanks system" [Online]. Available: `http://www.control-systems-principles.co.uk/whitepapers/coupled-tanks-systems.pdf`.

[27] "Application examples of the KRi coupled-tank apparatus PP-100" [Online]. Available: `http://www.kri.com.sg/ct101.pdf`.

[28] "System identification of a first order system" [Online]. Available: `http://www.me.cmu.edu/ctms/modeling/firstorder/systemidentification/mainframes.htm`.

[29] "System identification tutorial" [Online]. Available: `http://www.me.cmu.edu/ctms/modeling/tutorial/systemidentification/mainframes.htm`.

[30] C. Kao and B. Lincoln, "Simple stability criteria for systems with time-varying delays," *Automatica*, vol. 40, pp. 1429–1434, Aug. 2004.

[31] R. Caceres, R. Rojas, and O. Camacho, "Robust PID control of a Buck boost DC-AC converter," in *Twenty-second International Telecommunications Energy Conference, Intelec.*, pp. 180–185, 2000.

[32] R. P. Shrivastava, "Use of genetic algorithms for optimization in digital control of dynamic systems," in *Proceedings of the ACM Annual Computer Science Conference*, pp. 219–224, 1992.

[33] M. Pohjola, "PID Controller Design in Networked Control Systems," *Master's thesis, Helsinki University of Technology*, Jan. 2006.

[34] T. Katayama, T. McKelvey, A. Sano, C. Cassandras, and M. Camp, "Trends in systems and signals," *Proceedings of 16th IFAC World Congress, Prague, Czech Republic*, 2005.

[35] B. Ramaswamy, Y. Chen, and K. Moore, "Omni-directional robotic wheel - a mobile real-time control systems laboratory," *Proceedings of the 2006 American Control Conference*, Jun. 2006.

[36] B. Ramaswamy, *Embedded and Telepresence Control of a 3-Axis Smart Wheel Assembly.* Master's thesis, Utah State University, 2005.

[37] V. Bhambhani, Y. Q. Chen, and D. Xue, "Optimal fractional order proportional integral controller for varying time-delay systems," *17th IFAC World Congress.*, July 2008.

[38] L. M. Eriksson and M. Johansson, "PID controller tuning rules for varying time-delay systems," *in Proceedings of the 2007 American Control Conference*, July 2007.

[39] C.-C. Tsai and C.-H. Lu, "Multivariable self-tuning temperature control for plastic injection-molding process," *IEEE Transactions on Industry Applications*, vol. 34, pp. 310–318, 1998.

[40] J.-M. Dilhac, C. Ganibal, J. Bordeneuve, and N. Nolhier, "Temperature control in a rapid thermal processor," *IEEE Transactions on Electron Devices*, vol. 39, pp. 201–203, 1992.

[41] C.-T. Lin, C.-F. Juang, and C.-P. Li, “Temperature control with a neural fuzzy inference network,” *IEEE Transactions on Systems, Man and Cybernetics, Part C*, vol. 29, pp. 440 – 451, 1999.

[42] C.-F. Juang and J.-S. Chen, “A recurrent neural fuzzy network controller for a temperature control system,” in *Proceedings of the 12th IEEE International Conference on Fuzzy Systems*, pp. 408 – 413, May 2003.

[43] H. Ramos, F. Assuncao, A. L. Ribeiro, and P. M. Ramos, “A low-cost temperature controlled system to test and characterize sensors,” in *Proceedings of the 7th AFRICON Conference in Africa*, pp. 457 – 460, 2005.

[44] B. J. Lurie, “Three-parameter tunable tilt integral derivative (TID) controller,” *US Patent US5371670*, 1994.

[45] I. Podlubny, “Fractional-order systems and PI$^\lambda$D$^\mu$-Controllers,” *IEEE Transactions of Automatic Control*, vol. 44, no. 1, pp. 208–214, 1999.

[46] A. Oustaloup, B. Mathieu, and P. Lanusse, “The CRONE control of resonant plants: application to a flexible transmission,” *European Journal of Control*, vol. 1, no. 2, 1995.

[47] A. Oustaloup, X. Moreau, and M. Nouillant, “The CRONE Suspension,” *Control Engineering Practice*, vol. 4, no. 8, pp. 1101–1108, 1996.

[48] H.-F. Raynaud and A. Zergaïnoh, “State-space representation for fractional order controllers,” *Automatica*, vol. 36, pp. 1017–1021, 2000.

[49] B. M. Vinagre and Y. Chen, “Lecture notes on fractional calculus applications in automatic control and robotics,” in *Proceedings of The 41st IEEE CDC2002 Tutorial Workshop # 2*, B. M. Vinagre and Y. Chen, Eds., pp. 1–310. [Online] `http://mechatronics.ece.usu.edu /foc/cdc02_tw2_ln.pdf`, Las Vegas, Nevada, USA, 2002.

[50] J. A. T. Machado, “Special issue on fractional calculus and applications,” *Nonlinear Dynamics*, vol. 29, pp. 1–385, Mar. 2002.

[51] M. D. Ortigueira and J. A. T. Machado, “Special issue on fractional signal processing and applications,” *Signal Processing*, vol. 83, no. 11, pp. 2285–2480, Nov. 2003.

[52] I. Podlubny, L. Dorcak, and I. Kostial, “On fractional derivatives, fractional-order dynamic systems and $PI^\lambda D^\mu$-controllers,” in *Proceedings of the 36th Conference on Decision and Control*, pp. 4985–4990, San Diego, CA, Dec. 1997.

[53] A. Oustaloup, *La Dérivation Non Entière.* Paris: Hermes, 1995.

[54] I. Podlubny, “Fractional-order systems and fractional-order controllers,” in *UEF-03-94, The Academy of Sciences Institute of Experimental Physics*, Kosice, Slovak Republic, 1994.

[55] B. M. Vinagre, C. Monje, and A. Calderon, “Fractional order systems and fractional order control actions,” in *Proceedings of IEEE Conference on Decision and Control*, 2002.

[56] B. M. Vinagre, I. Podlubny, L. Dorcak, and V. Feliu, *On Fractional PID Controllers: A Frequency Domain Approach*, B. M. Vinagre and Y. Chen, Eds., `http://mechatronics.ece.usu.edu /foc/cc02tw/cdrom/lectures/book.pdf` Las Vegas, NV, USA, Dec. 2002.

[57] C. Hwang, J.-F. Leu, and S.-Y. Tsay, "A note on time domain simulation of feedback fractional systems," *IEEE Transactions on Automatic Control*, vol. 47, no. 4, pp. 625–631, Apr. 2002.

[58] J.-F. Leu, S.-Y. Tsay, and C. Hwang, "Design of optimal fractional-order PID controllers," *Journal of Chinese Institute of Chemical Engineers*, vol. 33, pp. 193 – 202, 2002.

[59] I. Petras and B. M. Vinagre, "Practical application of digital fractional-order controller to temperature control," *Acta Montanistica Slovaka*, vol. 7, no. 2, pp. 131–137, 2002.

[60] Y. Aoki, M. Sen, and S. Paolucci, "Approximation of transient temperatures in complex geometries using fractional derivatives," Department of Aerospace and Mechanical Engineering, University of Notre Dame, Notre, IN, Technical Note, Dec. 2005.

[61] G. W. Bohannan, "Analog fractional order controller in a temperature control application," in *Proceedings of International Federation of Automation and Control, FDA*, pp. 1 – 6, 2006.

Appendices

Appendix A

Optimal PID Tuning Rules

In addition to designing of an optimal FO-PI (OFOPI) controller and developing optimal FO-PI tuning rules, we also compare the OFOPI controller and the optimal PID controller (OPID). Briefly summarizing, the OPID controller is represented in time-domain as

$$\begin{aligned} u(t) &= k(py_r(t) - y_f(t)) + k_i \int_0^t (y_r(\tau) - y_f(\tau))\mathrm{d}\tau \\ &\quad + k_d(q\frac{\mathrm{d}y_r(t)}{\mathrm{d}t} - \frac{\mathrm{d}y_f(t)}{\mathrm{d}t}), \end{aligned} \tag{A.1}$$

where k, k_i, and k_d are the gain parameters of the controller given by AMIGO tuning rules, p and q are the set-point weights and y_f is the filtered process variable. The output is considered to pass through a low-pass filter having a transfer function $G_f(s)$ given as

$$G_f(s) = \frac{1}{(T_f s + 1)^2}. \tag{A.2}$$

The other controller parameters are defined mathematically as

$$p = \begin{cases} 0, & \text{if } \tau \leq 0.5, \\ 1, & \text{if } \tau > 0.5; \end{cases} \tag{A.3}$$

$$q = 0;$$

$$T_f = \begin{cases} \frac{0.05}{\omega_{gc}}, & \text{if } \tau \leq 0.2, \\ 0.1L, & \text{if } \tau > 0.2; \end{cases},$$

where ω_{gc} is the cut off frequency of the filter. Further, the tuning rules proposed on PID controller were:

$$\begin{aligned} k &= \frac{0.4T - 0.04}{K_pL} + \frac{0.16}{K_p}, && \text{(A.4)} \\ k_i &= 0.01(\frac{-0.11T^3 + 1.5T^2 - 1.5}{K_pL^2} + \frac{0.35T^2 + 4T + 50}{K_pL}), && \text{(A.5)} \\ k_d &= 0.01(\frac{0.4T^2 + 11T}{K_p}). && \text{(A.6)} \end{aligned}$$

Appendix B

Setpoint Conversion Formulae

If S_D is drive speed set point, S_S is steering speed set point, P_Z is Z-axis position set point and Z_L is control character that corresponds to lowest position of the wheel assembly along the z-axis, then conversion formulae from Serial Input to Wheel Set-Point are given by:

$$P_Z = \frac{(B - Z_L)}{(Z_U - Z_L)}(Z_U - Z_L) + Z_L, \tag{B.1}$$

$$S_D = \frac{(B - D_{MID})}{(D_{MAX} - D_{MID})}, \tag{B.2}$$

$$S_S = \frac{(B - S_{MID})}{(S_{MAX} - S_{MID})}. \tag{B.3}$$

Appendix C

Code

```
 function F = optim(pid)
 clc
 K = 1;
   L =0.1;
  T= 0.5;
   tau = L/(L+T);
    G = tf(K,[T 1],'OutputDelay',L);
 [Gm,Pm,Wg,Wp] = margin(G);
  if(tau > .6)
  ford= 1.1;
 elseif(.4 <= tau \&\& tau <= .6)
  ford = tau + .5;
  elseif(.2<= tau \&\& tau < .4)
  ford = .9;
  elseif(.1 <= tau \&\& tau<.2)
  ford = tau + .7;
  elseif (.05<= tau \&\& tau < .1)
 ford = tau +.6;
 elseif (tau <.05)
  ford = .7;
  end
   Kp=pid(1);
    Ki=pid(2);
ford=pid(3);
  if (tau>0.2)
   Tf= .1*L;
```

```
else
Tf=.05/Wg;
 end
 if ford == 1
 C = Kp +Ki*tf([1],[1 0]);
 else
Ci = ora_foc(-ford,0,10^-4,10^4);
 C = Kp + Ki*Ci;
 end
  s=tf('s');
  H = G*C;
 w = logspace(-3,3,10000)';
Hf = squeeze(freqresp(H,w));
 [delta_max I]=min(abs((1+Hf)./(j*w.*Hf)));
 w_refined =logspace(log10(w(I-5)),log10(w(I+5)),1000)';
 Hf =squeeze(freqresp(H,w_refined));
 delta_max=min(abs((1+Hf)./(j*w_refined.*Hf)));
 delta_max=delta_max;
 time = 0:.2:200;
 [y t]=step(H/(ss(1)+H),time);
 itae = max(sum(abs(1-y(1:end-1)).*t(1:end-1).*diff(t)));
F = [itae 1/delta_max];
 disp(['Kp: ' num2str(Kp) ', Ki: ' num2str(Ki) ', ford: ' num2str(ford) ',
 ITAE: ' num2str(itae) ', delta_max: ' num2str(delta_max)])
```