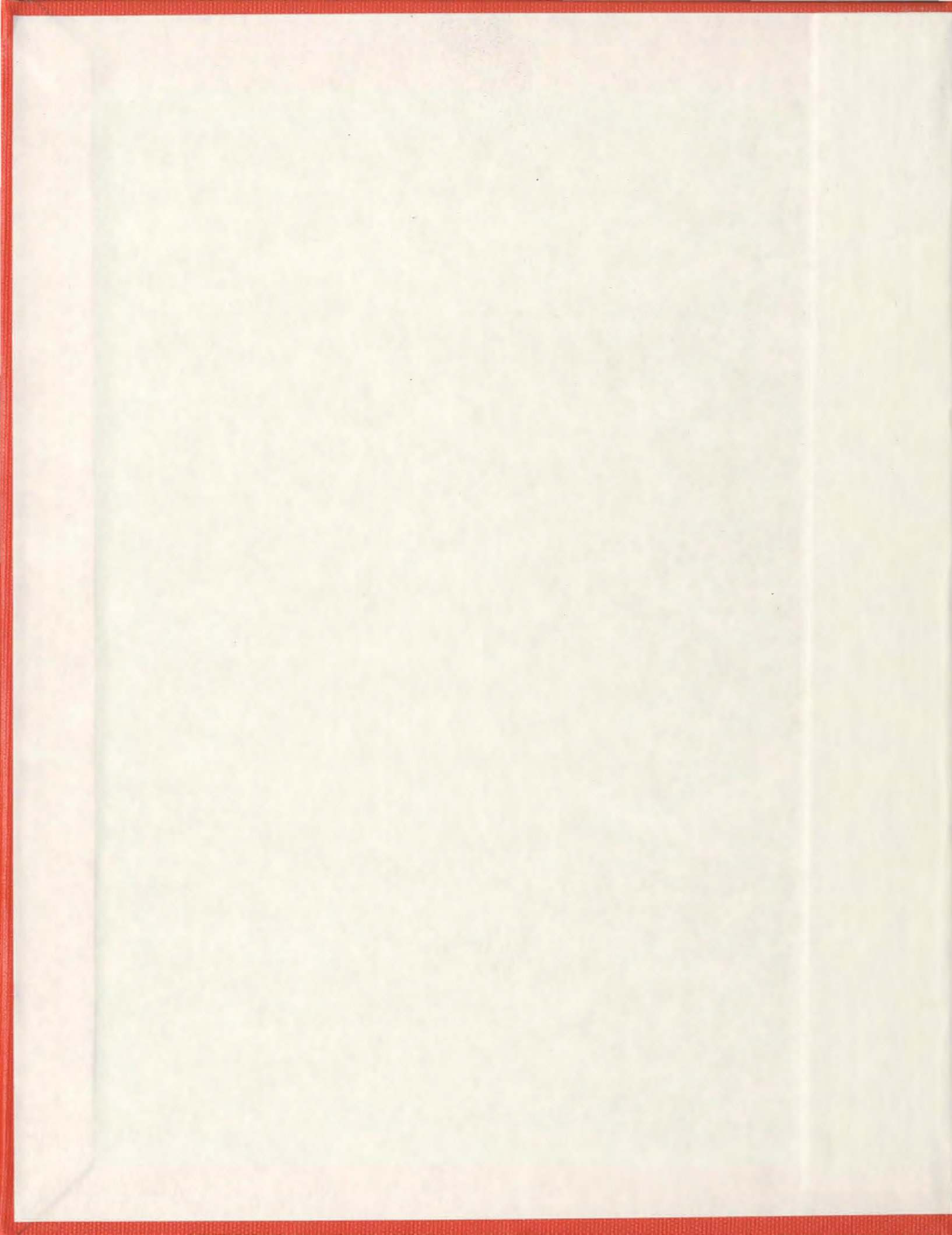


OPTIMIZATION OF CONTROLLER GAINS FOR
FPGA-BASED MULTIVARIABLE MOTION CONTROLLER
USING RESPONSE SURFACE METHODOLOGY

HARI PRIYAI SEKARAN



**Optimization of Controller Gains for
FPGA-Based Multivariable Motion Controller
Using Response Surface Methodology**

by

© Hari Priyai Sekaran

A Thesis submitted to the
School of Graduate Studies
in partial fulfillment of the
requirements for the degree of
Master of Engineering

Faculty of Engineering and Applied Science
Memorial University of Newfoundland

May 2013

St. John's

Newfoundland

Abstract

Field Programmable Gate Arrays (FPGA) has become increasingly popular in recent years for control applications. Using contemporary FPGA technology, a powerful virtual processor can be synthesized and integrated with custom hardware to create a dedicated controller that outperforms conventional microcontroller and microprocessor based designs. The FPGA based controller takes advantage of both hardware features and virtual processor technology. This study details the development of a cascaded type Proportional-Derivative (PD) controller for an inverted pendulum system implemented on a single FPGA device. The controller includes a hardware based implementation of the Input Output (IO) modules including quadrature decoders/counters and a Pulse Width Modulation (PWM) controller for the motor driver. The NIOS II processor was synthesized to implement the cascaded PD controller algorithm. This study also proposes a novel method for obtaining the optimal controller gains for the system. It uses Design of Experiments (DoE) techniques for obtaining optimal gain values. In this study three Response Surface Methodology (RSM) designs: Central Composite, Box-Behnken and Uniform Design are used for obtaining optimal gain values. Based on the results of this study, the uniform design approach yielded the most satisfactory results. The gains provided by the response surface model from the uniform design experiment are verified experimentally to validate the proposed controller tuning method. A classic inverted pendulum system was selected to demonstrate the applicability of the proposed approach

Acknowledgements

I would like to express my sincere gratitude and thanks to my supervisor Dr. Nicholas Krouglicof, who helped, encouraged and offered his guidance in this research. I also thankfully acknowledge his persistent help, advice and valuable guidance at every stage of my research work.

I extend my appreciation to my colleague Mr. Migara Liyanage for his help and support during my program of study.

Last but not least, I gratefully thank my family, father Mr. Sekaran and mother Mrs. Lalitha Sekaran and my sisters Mrs. Manju Bhargavi and Mrs. Kamini Kousalya for their moral support, constant encouragement, help, immense love and affection shown throughout my research life.

Table of Contents

ABSTRACT	i
ACKNOWLEDGEMENTS	ii
List of Tables	vi
List of Figures.....	vii
List of Symbols andAbbreviations	ix
Chapter 1 Introduction.....	1
1.1 FPGA- an overview	4
1.1.1 Evolution of programmable logic devices.....	4
1.2 Motivation.....	8
1.3 Problem statement.....	9
1.4 Proposed methodology for obtaining optimal gains.....	9
1.5 Research methodology overview.....	10
1.6 Thesis outline.....	10
Chapter 2 Literature review and recent trends	12
2.1 Overview	12
2.2 FPGA based control system: existing methods and techniques.....	14
2.2.1 FPGA based control	14

2.2.2 Tuning the system.....	16
2.2.3 Design of Experiment techniques.....	21
Chapter 3 Experimental apparatus.....	23
3.1 Introduction	23
3.2 Description of the system.....	24
3.3 System modeling	25
3.3.1 Modeling the motor	25
3.3.2 Modeling the pendulum.....	35
Chapter 4 Architecture of proposed FPGA based controller	40
4.1 Overview	40
4.2 FPGA architecture	41
4.2.1 IO modules	42
4.2.2 Processor module.....	49
4.2.3 Hardware implementation	53
Chapter 5 Novel method for tuning controller gains.....	57
5.1 Introduction.....	57
5.2 Classical approach using Ziegler-Nichols method.....	58
5.3 Proposed methodology for the controller	60
5.4 DOE based tuning method-an overview	64

5.5 Why Design of Experiment (DOE).....	65
5.6 Response Surface Methodology(RSM).....	67
5.6.1 Brief summary of BBD and CCD.....	68
5.7 Experimental methodology	69
5.7.1 BBD results and analysis.....	71
5.7.1.1 Statistical test of significance using ANOVA-BBD.....	71
5.7.1.2 Regression analysis-BBD	75
5.7.2 CCD results and analysis.....	75
5.7.2.1 Statistical test of significance using ANOVA-CCD.....	75
5.7.2.2 Regression analysis-CCD	80
5.7.2.3 Experimental validation of system model-CCD.....	82
5.7.3 Uniform Design Experiment (UD)-overview.....	83
5.7.4 Uniform design results and analysis.....	84
5.7.4.1 Statistical test of significance using ANOVA-UD.....	84
5.7.4.2 Regression analysis-UD	90
5.7.4.3 Experimental validation of system model-UD.....	90
5.7.4.4 Optimization technique	92

Chapter 6 Conclusions and recommendations.....	100
6.1 Summary of results and conclusions.....	101
6.2 Contributions.....	106
6.3 Limitations and recommendations	107

List of Tables

4-1 State Transitions	44
4-2 Decoding logic.....	46
5-1 Initial range of gain values	59
5-2 Range considered for primary factors	71
5-3 Analysis of variance table (BBD)	72
5-4 Summary of Statistics (BBD)	73
5-5 Analysis of variance table (CCD)	76
5-6 Summary of Statistics (CCD)	77
5-7 Predicted optimal solution and observed value (CCD)	82
5-8 Analysis of variance table (UD)	85
5-9 Summary of Statistics (UD)	86
5-10 Predicted versus observed data points (UD).....	90
5-11 Predicted optimal solution and average observed values (UD).....	98

List of Figures

1-1	Configurable logic blocks in FPGA.....	6
1-2	Logic elements	7
2-1	FPGA starter board	13
2-2	Block diagram of PID type controller.....	18
3-1	Diagram of an inverted pendulum	24
3-2	Mechanical system with pendulum.....	25
3-3	General motor model	26
3-4	Block diagram representation of equation (3.4).....	28
3-5	Block diagram representation of equation (3.7).....	30
3-6	Control system	31
3-7	Block diagram representation of equation (3.12).....	31
3-8	Block diagram representation of equation (3.13).....	32
3-9	Simulink model of motor	33
3-10	Motor PID controller.....	34
3-11	Response to a Step input	34
3-12	Free body diagram of the system	35
3-13	Pendulum model in Simulink.....	37
3-14	Simulink model of the system with initial gain values	38
3-15	Position of the table	38
3-16	Pendulum angle in radiance	39
4-1	Internal architecture of the proposed controller.....	42

4-2	Quadrature decoder	43
4-3	Quadrature decoder timing diagram	44
4-4	Edge detection circuitry	45
4-5	PWM Module.....	47
4-6	Modulated duty cycle.....	48
4-7	SOPC system with Nios II processor.....	52
4-8	Design entry flowchart.....	56
5-1	Diagram of the controller	58
5-2	Diagram of proposed controller architecture	60
5-3	Integral squared error	66
5-4	Response Surface Designs.....	67
5-5	Residuals versus predicted (BBD).....	74
5-6	Box-Cox plot for power transforms (BBD)	74
5-7	Normality probability plots of residuals (CCD)	78
5-8	Residuals versus predicted (CCD)	79
5-9	Residual versus run (CCD)	79
5-10	Predicted versus actual (CCD).....	80
5-11	Box-Cox plot for power transforms (CCD)	80
5-12	Predicted data versus observed data	83
5-13	Normality probability plots of residuals (UD).....	87
5-14	Residuals versus predicted (UD).....	87
5-15	Predicted versus actual (UD)	88

5-16	Box-Cox plot for power transforms (UD)	88
5-17	Residual versus run (UD).....	89
5-18	Interaction graph (UD).....	89
5-19	Experimental validation of optimal gains	92
5-20	Numerical optimizations of factors.....	96
5-21	Numerical optimization of response	97
5-22	Predicted confirmation report	99

List of Symbols and Abbreviations

m	Inverted pole of mass
θ	Angle from the vertical axis on a cart
M	mass
x	Direction along the track
f	Force required to push the cart horizontally
I_a	Armature current
R_a	Resistive component
L_a	Inductive component
V_B	Back EMF voltage
EMF	Electromotive Force
K_B	Back EMF constant
K_m	Torque constant
ω	Angular velocity of the motor
J_M	Moment of inertia of motor
J_L	Moment of inertia of load
f_M	Viscous damping friction of motor
$\dot{\theta}$	Angular velocity of the motor
τ_E	Electrical time constant of a DC motor
τ_M	Mechanical Time Constant of a DC motor
f	Force exerted on the pendulum
K_p	proportional gain

T_i	Integral gain
T_d	Derivative gain
K_{cr}	Critical value
P_{cr}	Sustained oscillation Period
α	Axial points
y	Response
β_0	Overall average
β_i	Regression coefficient
$x_i x_j$	Represents the factor i and j respectively
k	Number of factors,
β_{ij}	Interaction terms
d_i	Desirability Index
Kp_one	Proportional gain of the table.
Td_one	Derivative gain of the table.
Kp_two	Proportional gain of the pendulum.
Td_two	Derivative gain of the pendulum.
VLSI	Very-Large-Scale-Integration
EDA	Electronic Design Automation
FPGA	Field Programmable Gate Array
ASIC	Application-Specific Integrated Circuit
IP	Intellectual property
SoPC	System-On-a-Programmable-Chip

PID	Proportional Integral Derivative
DC	Direct current
PWM	Pulse width modulation
HDL	Hardware Description Languages
ZN	Ziegler and Nichols
SISO	Single Input Single Output
MIMO	Multiple Input Multiple Output
DOE	Design of experiments
RSM	Response Surface Methodology
PROM	Programmable Read-Only Memory
PLD	Programmable Logic Devices
PLA	Programmable Logic Array
PAL	Programmable Array logic
CPLD	Complex Programmable Logic Devices
CLB	Configurable Logic Block
LE	Logic Element
LUT	Look-Up Table
LAB	Logic array block
PCB	Printed Circuit Board
VHDL	Very High Speed Integrated Circuit Hardware Description Language
MATLAB	Matrix Laboratory

FLC	Fuzzy Logic Controller
LQR	Linear Quadratic Control
FSF	Full State Feedback
DA	Distributed Arithmetic
ISE	Integral Square Error
LPM	Library of Parameterizable Megafunctions
IO	Input output module
PLL	Phase locked loop
LED	Light emitting diode
AHDL	Altera Hardware Description Language
MMU	Memory Management Unit
MPU	Memory Protection unit
ISC	Instruction Set Architecture
JTAG	Joint Test Action Group
IRQ	Interrupt Request
SRAM	synchronous Random Access Memory
DDR SDRAM	Double Data Rate Synchronous Dynamic Random Access Memory
OFAT	one-factor-at-a-time
ANOVA	Analysis of Variance
BBD	Box-Behnken Design
UD	Uniform Design Experiment

Chapter 1

Introduction

Prompted by the development of sophisticated techniques of automatic control theory, the process of designing digital control hardware has changed dramatically with the introduction of many novel control algorithms over the past few years. For a digital motion controller to work efficiently, the hardware structure and control algorithm go hand in hand. Most controller structures are based on a Digital Signal Processor (DSP) or a microprocessor, in combination with additional memory and interface circuits. This requirement demands a complex hardware structure, increased system size, high power consumption and high cost. Hence with the progress in Very-Large-Scale-Integration (VLSI) technology and Electronic Design Automation (EDA) techniques, Field Programmable Gate Arrays (FPGA) are now widely used because of their simplicity, fast time to market, programmability, short design cycle, low power consumption, and performance. FPGAs have gained acceptance over Application-Specific Integrated Circuit (ASIC) for industrial applications because FPGAs can be reconfigured to meet the design requirements and end user's preferences.

FPGAs also offer embedded processor Intellectual Property (IP) and application IP that can be deployed to construct a System-On-a-Programmable-Chip (SoPC). System-on-a-Programmable-Chip (SoPC) architectures are increasingly popular for robotic and motion control applications. This architecture integrates the tasks of different chips on to one device which yields a more compact, reliable design. Within the SoPC, the functions

requiring fast processing can be implemented in hardware and the highly sophisticated, computationally intensive algorithms can be realized in software on a “soft” processor which makes a complete computer system that runs on a single FPGA chip. With the advent of dedicated parallel architectures, the execution time of algorithms has been drastically reduced; however, much work remains to be done to develop fast and reliable motion control devices with FPGAs.

A virtual processor integrated with the hardware design provides greater flexibility to designers. Designers can now integrate the powerful features of hardware designs with complex software algorithms. In this research work, an FPGA-based system on a chip motion controller for an inverted pendulum system was developed. Utilizing the hardware/software re-configurability and the computational capability, FPGA based controllers offer several advantages over the conventional microprocessor-based approach. In this study, a highly integrated, cascaded type Proportional-Integral-Derivative (PID) controller was implemented in a dedicated FPGA. Generally PID controllers have a most impressive record in terms of the number of successful industrial deployments and are the most extensively used controller class in modern industrial control systems. FPGAs offer features for a much more compact implementation of the hardware interfaces required for the feedback and command signals. Utilizing the hardware/software partitioning approach in a single FPGA, the quadrature decoders/counters acquiring the encoder feedback and the Pulse-Width-Modulation (PWM) module for driving the DC motor were implemented in hardware. The development tool used was the Quartus II Web Edition in which designs are created using

a Hardware Description Languages (HDL). The controller, which is essentially a cascaded PID based algorithm, was soft coded in a virtual processor called the Nios II, all configured on the same FPGA.

System performance indices are an important element in tuning the system. Tuning of PID controllers for a satisfactory system performance has been widely discussed topic in control engineering. Many novel techniques exist for obtaining optimal controller gains, most of which are computationally intensive and difficult to implement in real time systems. One of the most popular and widely adopted techniques is the Ziegler and Nichols (ZN) method for obtaining controller gains. In general for a Single Input Single Output (SISO) system conventional ZN tuning results in a stable, well-behaved system, whereas for a Multiple Input Multiple Output (MIMO) system the ZN method often does not yield optimum dynamic performance. The available SISO-based PID design techniques have limitations when extended to MIMO systems. There is no systematic design procedure available to design and tune PID controllers for MIMO systems. The main objective of this study is to obtain optimal controller gains for a multivariable system. The study proposes a novel method for obtaining optimal controller gain values using Response Surface Methodology (RSM) and commercially available Design of Experiments (DOE) software, Design Expert 8 by Statease Inc.

RSM is a statistical and mathematical technique for developing, improving and optimizing a process or system under study. RSM has become common practice in engineering problems to examine and determine how the input parameters or variables influence the performance of a process or system. The performance aspect is referred to

as the response of the system. In RSM, the optimization of the response is achieved by factors or variables that are subjected to simultaneous testing over a limited number of experiments. RSM which develops a functional relationship that quantifies the effects of each variable and their interactions could potentially prove a promising technique for obtaining optimal controller gains for a multivariable system.

1.1 FPGA- an overview

FPGA's belong to the class of programmable digital devices called Programmable Logic Devices (PLD's).

1.1.1 Evolution of programmable logic devices

PROM (Programmable Read-Only Memory): PROM was the first type of user-programmable chip that could implement logic circuits. In PROMs, address lines are used as logic circuit inputs and data lines as outputs.

Disadvantage:

- Although useful for implementing look up tables PROMS are an inefficient architecture that are very rarely used in practice by Floyd (2009).

PLA: The Programmable Logic Array was a specifically developed device for implementing logic circuits. It consists of two levels of logic gates: a programmable “wired” AND-plane followed by a programmable “wired” OR-plane. Hence PLA's are well suited for implementing logic functions in sum-of-product forms by Floyd (2009).

Disadvantage:

- Very expensive to manufacture.

- Poor speed performance.
- Two-level of configurable logic introduces significant propagation delay.

PAL (Programmable Array logic): To overcome the drawbacks of two-level programmable logic, PALs implement a single level of programmability consisting of “wired” AND-plane that feeds fixed OR-gates. Flip flops are connected to OR-gate outputs so that sequential circuits can be realized. The development of PALs had a profound effect on digital hardware design, and formed the basis for some newer and more sophisticated architecture. PLD’s including PLA’s, PAL’s, and PAL-like devices are grouped into single category called Simple PLD’s (SPLD’s).

CPLD (Complex PLD): With advancement in technology it has become possible to produce devices with higher capacity than SPLDs. The integrating of multiple SPLD’s onto a single chip is collectively referred to as a CPLD.

Disadvantage:

- CPLD’s provide logic capacity equivalent of 50 typical SPLD devices, but it is somewhat difficult to extend these architectures for higher densities. Hence a different approach is required.

FPGA: General-purpose logic chips available today which offer the highest capacity are traditional gate arrays. FPGAs are composed of an array of Configurable Logic Blocks (CLBs), input/output blocks and programmable interconnects. Figure 1-1 depicts the logic block and programmable interconnects.

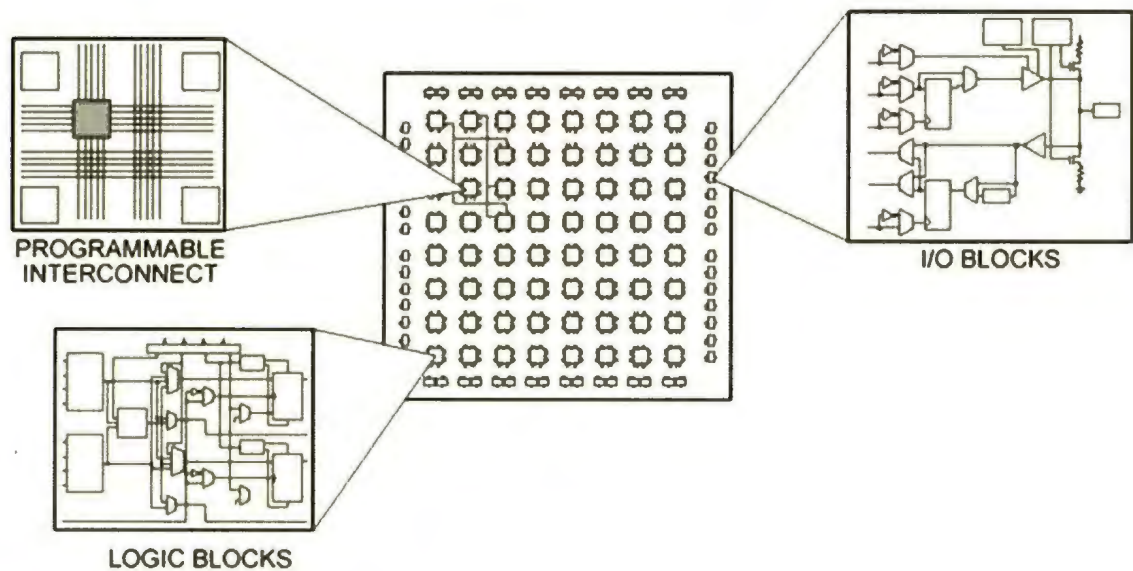


Figure 1-1 Configurable logic blocks in FPGA

(<http://www.sei.cmu.edu/reports/05tr016.pdf>)

Configurable Logic Blocks (CLB)

Generally, the CLB consists of a few logical cells called Logic Elements (LEs). LEs are the smallest units of logic in the FPGA architecture. Each LE consists of a 4-input Look-Up Table (LUT), a full adder and a D-type flip-flop. Sixteen LEs comprise a logic array block (LAB) which is the hierarchical array in the Cyclone III device family as shown in Figure 1-2. LEs operate in two modes namely normal and arithmetic modes. Logical applications and combinational functions are implemented in normal mode. Arithmetic mode is ideal for implementing adders, counters, accumulators and comparators by Floyd (2009).

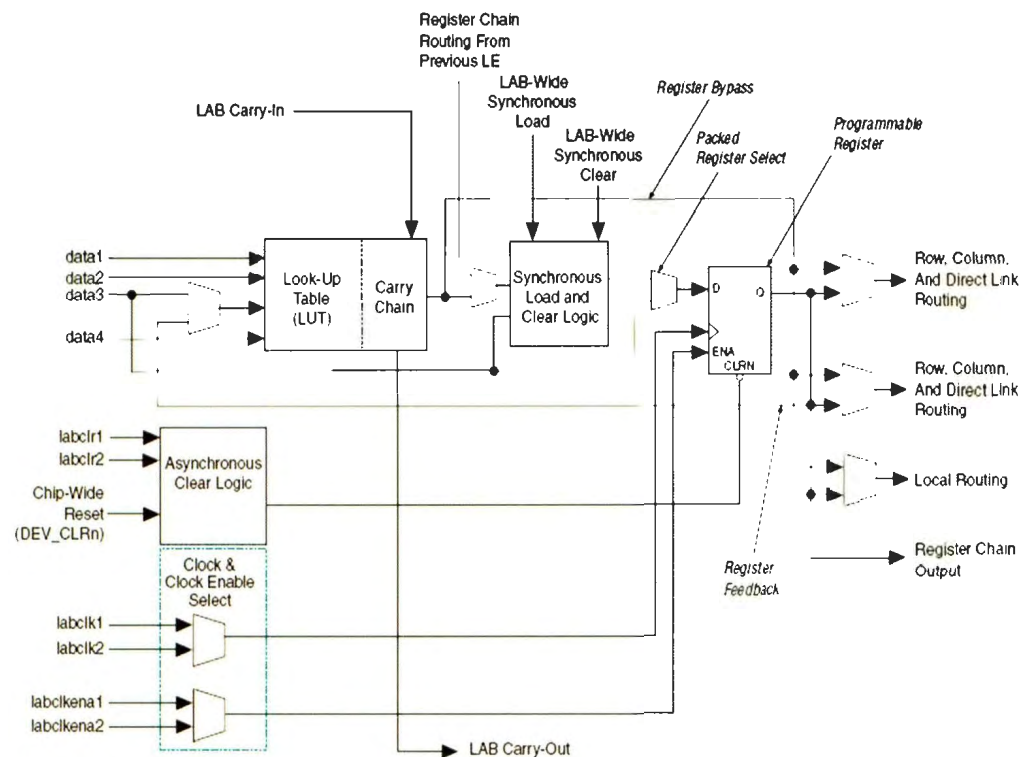


Figure 1-2 Logic Elements

(http://www.altera.com/literature/hb/cyc3/cyc3_ciii51002.pdf)

Programmable Interconnect

The salient feature influencing FPGA performance is the programmable interconnect. Each logic block (subsystem) and the I/O pads are routed through switch matrices. In an FPGA each wiring segment spans only one logic block before it terminates into a switch box. Within the switch box longer paths are constructed by turning on some of the programmable switches. This results in general purpose interconnects, adjacent interconnects and long lines. The pattern, or topology, of switches used in this architecture is the planar or domain-based switch box topology.

I/O Blocks

I/O blocks act as an interface between logic blocks and external devices. An I/O block consists broadly of an I/O pad. An I/O pad makes a connection with the adjacent channel through any one of the wiring segments. For example, an I/O pad at the top of the chip connects to any of the “W” (W is channel width) wires in the horizontal channel below it.

1.2 Motivation

In the Altera FPGA family of devices, the evolution of FPGA applications have led to higher density devices, IP integration using System-on-a-Programmable-Chip (SoPC) builder, and high speed I/O interconnect technology. Hence with the availability of multimillion gates and features, design structure similar to the traditional ASIC devices can be created in an FPGA using SoPC. SoPC builder connects the entire system using the Avalon bus which is a flexible and intelligent bus system that can be configured to fit the system requirements. Therefore with the resources and architecture that are available in today's FPGAs, an effective system that was once only possible in traditional ASIC devices can now be implemented on a reconfigurable device. In this study, the use of the Cyclone III FPGA board results in a low cost system with low power consumption and extremely high performance. The system is composed of a 32-bit processor called the NIOS II/s standard processor that can access up to 2GB external address space and 20 KB of on-chip memory. In this study, an FPGA-based motion controller for a multivariable system was developed that exploits the advantages of SoPC.

1.3 Problem statement

Ogata (2005) stated that there is no systematic approach for tuning the controller gains to obtain optimal gain values for a multivariable system. The classic Ziegler and Nichols (ZN) technique for tuning controller gains is the most popular and widely adopted empirical method; however it provides a starting point for fine tuning rather than giving the final gain values. A novel approach is proposed herein for obtaining optimal gain values for the system.

1.4 Proposed methodology for obtaining optimal gain

A novel, computationally efficient, offline method for obtaining the controller gains will be investigated. Response Surface Methodology (RSM) technique will be used to tune the system of multiple PID type control loops. In this method the P, I, D controller gains will be considered as the primary factors. An applicable range of gains will be considered with an appropriate performance index as the response of the system. The objective is to minimize the performance index which represents the cost function of the given system. A set of experiments will be carried out for each combination of gains determined by a specific response surface design and implemented by the the Design of Experiments (DOE) software, Design-Expert V8 by Statease Inc. The response is then modeled using regression analysis and then used to calculate the optimal set of gains, which minimizes the cost function. The optimal gain values will be obtained by simultaneous testing of factors or variables through experimentation. Ideally the optimal gains provide the most stable dynamic performance in the system.

1.5 Research methodology overview

- The study starts with the implementation of an FPGA based motion controller for an inverted pendulum. The hardware/software processes were partitioned in the FPGA with the IO modules implemented in hardware. The numerically intensive control algorithms (i.e.; the Cascaded PID type controller) were soft coded on the FPGA using SoPC.
- Initially the system was tuned using the Ziegler and Nichols method in order to obtain the approximate values of controller gains.
- Using a trial and error method the initial gain values were adjusted and a range of gains was determined. Based on these initial gain values, Design of Experiments (DOE) was employed to optimize the gain values. The objective function that was minimized was the integral square error of the pendulum angular position.
- The last part of this work involves the experimental validation of the optimal solutions obtained from DOE.

1.6 Thesis outline

This thesis consists of six chapters and it is organized as follows:

Chapter 1

Provides a synopsis of FPGA based control, the objectives of the study, problem statement, discussion on proposed technique and methodology overview.

Chapter 2

It discusses literature review of existing methods, regarding the FPGA based motion control, tuning techniques in control engineering and a novel approach for optimal controller gains using Design of Experiments.

Chapter 3

It describes the overall experimental setup, the description of the hardware, the mathematical modeling of the system, and the simulation results obtained from the system model.

Chapter 4

Explain the detailed study of the proposed controller architecture. The various modules implemented in hardware and software on the FPGA are described in this chapter.

Chapter 5

Describes a novel computational technique for obtaining optimal controller gains is discussed in detail in this chapter. The results obtained from this technique and the experimental validation incorporating DOE results are summarized.

Chapter 6

Summarizes the results and conclusions are drawn. Contributions and recommendations for future work are also highlighted.

Chapter 2

Literature review and recent trends

2.1 Overview

Progress in Very Large Scale Integration (VLSI) technology has lead to the widespread application of FPGAs in digital control systems. Traditional microprocessors and Digital Signal Processors (DSPs) can no longer keep pace with the new generation of applications which requires more flexibility and higher performance at low power and cost. The option of integrating additional logic onto a single chip is not possible in a microprocessor or DSP. In addition, customizable chips reduce the complexity of PCB (Printed Circuit Board) layouts. Custom chips can be classified as semi custom or fully custom application specific devices that are relatively expensive. Companies including Xilinx and Altera have responded by introducing customizable, low-cost, fast time-to-market FPGAs that offer a range of hardwired features. FPGA hardware programmability enables easy implementation of dedicated high-performance logic circuits. Consequently a single FPGA can replace thousands of discrete components by incorporating millions of logic gates in a single integrated circuit (IC) chip. Today's high-end FPGA's can hold several millions gates and have some significant advantages over ASIC's including ease of design, lower development costs, more product revenue, and the opportunity to speed products to market.

PID controllers are widely used in industry because of their simplicity, robustness, effectiveness, and applicability. Despite the availability of numerous modern control

techniques, PID controllers are still employed in 95% of industrial applications. To exploit the advantages, an FPGA based controller was designed and implemented in this research work.

The FPGA chip adopted here is Altera cyclone III EP3C25

- 24,624 logic elements (LEs)
- 608,256 RAM bits
- 32-bit configurable CPU core
- Four Phase-Lock-Loops (PLLs)
- 215 I/Os



Figure 2-1 FPGA Starter board

http://www.altera.com/literature/manual/rm_ciii_starter_board.pdf?GSA_pos=4&WT.os_s_r=1&WT.oss=parts%20in%20FPGA%20pdf

Following standard design practices, the controller modules are implemented in Very High Speed Integrated Circuit Hardware Description Language (VHDL), synthesized using Quartus II as the foundation tool, and subsequently implemented on the FPGA board. The controller algorithm are soft coded in a virtual processor configured onto the same FPGA.

2.2 FPGA based control system: existing methods and techniques

2.2.1 FPGA based control

Early robotic controllers implement digital control system techniques using DSP chips and programmable logic to realize the software part and hardware part of the control system. With the development of submicron process technology, FPGA technology has greatly improved and thus created the opportunity to develop complex, compact, concurrent control algorithms and fast, reliable controllers for industrial motion control applications providing rapid development at a very low-cost. Nowadays an embedded processor IP and application IP can be developed and downloaded into an FPGA using an SoPC environment. SoPC architectures integrate different tasks on to one chip, and are increasingly popular in robotic applications. Functions requiring fast processing are programmed in hardware and highly sophisticated, computationally intensive algorithms can be realized in a soft processor on the same FPGA. Virtual processor integrated with the hardware design provides greater flexibility to designers. Designers can now integrate powerful hardware designs with complex algorithms on an FPGA chip using SoPC.

Therefore a fully digital motion control environment is possible yielding compact, more reliable designs.

Various published FPGA based motion control systems for robots are proposed in literatures. For example, an FPGA based motion control IC for a robot arm was presented by Kung & Shu (2005). The inverse kinematics scheme and point-to-point motion trajectory control was implemented in an embedded processor IP and the application IP used to realize the five-axis position control. The proposed internal architecture of the motion control IC based on SoPC technology involves functions of position command generation, inverse kinematic computation and point to point motion control implemented in a NIOS processor. The application IP realizes the five-axis position control in hardware. The overall utilization of circuit resources includes a NIOS embedded processor of 25.1 %, application IP of 30.9%, and 46% utilization of logic elements of the Cyclone I board. Another application of FPGA based motion control described by Chakravarthy & Xiao (2006) outlines the technology behind developing an FPGA based control system utilizing the hardware/software reconfigurable feature. By taking advantage of the hardware/software reconfigurable nature, the authors developed “a la carte” fashioned functions in hardware for high-speed performance and several others in software for high flexibility and optimal utilization of logic resources. In a custom designed, high performance, FPGA based multiprocessor acts as the “brain” of the miniature robot while flexible hardware for different tasks increases the processing speed. An FPGA based motion controller for humanoid robot arms is developed by Kim et al (2007). The emphasis is in the implementation of nonlinear PID controller as well as a

conventional PID control algorithm using a hardware description language. A comparative study between the PID controller and nonlinear PID controller are presented with simulation studies obtained for position control of the humanoid robot arms. A design implementation of Distributed Arithmetic (DA) based FPGA control was proposed by Chan et al (2007). A DA based PID controller was implemented, demonstrating an 80% saving of hardware utilization and 40% saving of power consumption in comparison with multiplier based control. In the proposed architecture presented by Chan et al (2007), a various modules built in the FPGA are reused for other applications resulting in cost reduction, less resource usage, high speed and low power consumption. Other related work by Kung et al (2009) is an FPGA-based motion control IC for an X-Y table with a self-tuning PID controller. The motion control IC exploits the advantages of the FPGA by implementing two axis speed and current control in hardware and the motion trajectory control algorithm for the X-Y table in software using a NIOS II embedded processor. Based on the principles of software/hardware co-design, all functionalities are implemented on the same FPGA. Until now, most related research work focused on using FPGAs to achieve a compact form factor, low-cost and high performance.

2.2.2 Tuning the system

PID controllers are the most extensively used controller class in modern industrial control system. They have a very impressive record in terms of the number of successful industrial deployments. They have been used for decades in a variety of applications ranging from slow response temperature controllers to fast acting robotic manipulators. Digital PID controllers based on microprocessor technology have also come into their

own in industry making it straightforward for the designer to implement PID controllers in an industrial environment. PID controllers based on traditional control loop feedback mechanisms are extensively used in industrial automation control system.

In this study, an FPGA based PID controller was designed and implemented to control an inverted pendulum. The inverted pendulum system is an example of a classic control system that is widely used for demonstration purposes. This particular system is useful for demonstrating the application of linear control to stabilize unstable systems. Inverted pendulum systems are inherently nonlinear in nature and thus make the control more challenging. Common control approaches such as Proportional-Integral-Derivative (PID) controller, Fuzzy Logic Control (FLC) and Linear Quadratic Control (LQR) requires a good knowledge of the system dynamics and accurate tuning in order to obtain desired performances.

In a PID controller, the proportional term indicates the response to the current error, the integral value determines the response based on the sum of present and past errors, and the derivative value determines the response based on the rate of change in errors. Hence the weighted sums of the parameters are used to control the plant or process. Though the structure of a typical PID controller as shown in Figure 2-2 is simple, it is not always easy to achieve the desired system behavior including settling time, overshoot, and steady state error. A PID controller can be referred to as a PI, PD, P or I controller in the absence of certain control actions. PI controllers are fairly common, since derivative action is sensitive to disturbance or noise in the system. Absence of an

integral term may prevent the system from reaching its target value; i.e., may result in steady-state error.

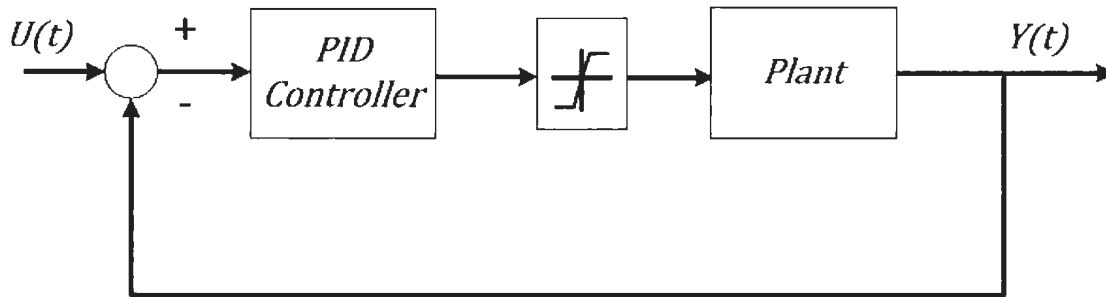


Figure 2-2 Block diagram of PID type controller

The previous section emphasizes the compactness, speed and reliability of industrial controllers based on FPGA technology. This section deals with common control approaches that are prominently and widely used in control engineering. Comparative evaluations of classical and modern control techniques are analyzed by Akhtaruzzaman and Shafie (2010). Three methods including 2DOF PID, Full State Feedback (FSF), and LQR are applied to control a rotary inverted pendulum. Firstly for a Single Input Single Output (SISO) system, 2DOF PID was applied and the root locus method was implemented to design the compensators. Secondly, modern control techniques that include FSF and LQR were implemented to test the up-right and swing up mode of the pendulum. The study explored the efficiency, reliability and accuracy of the system based on classical and modern control techniques. The researchers by Magna and Holzapfel (1998) studied the applicability and limitations of fuzzy logic control techniques for an

inverted pendulum. The fuzzy logic controller used vision feedback in to record the position of the inverted pendulum. To successfully control the pendulum based on vision feedback, the author separates the task into two parts: data acquisition and control. The vision system acquires the data and transmits them to the PC that determines the control command fed to the motor. Another FPGA based control technique for an inverted pendulum system is based on Visual Servoing. An inverted pendulum is balanced using visual servoing by incorporating visual information in the feedback control loops showed by Tu & Ho (2010). The position of the pendulum is measured with a machine vision system with the image processing algorithms implemented on an FPGA device. The FPGA provides real-time performance for computational intensive tasks through inherent architectural parallelism.

Research work by Zhao et al (2005) details an FPGA based controller for a small scale robot offering high reliability, re-configurability, and low power consumption. Zhao et al (2005) made a comparative study of different designs for closed loop PID control algorithms evaluated speed, resources, and power consumption. The PID module, which is the focus of this research, was implemented in both hardware on the FPGA and in software in the microprocessor for a comparative study. Other related work by Siddique et al (2009) presents an efficient implementation of a PID control algorithm on an FPGA. The algorithm was implemented using a Distributed Arithmetic (DA) based scheme, which utilizes Look-Up-Tables (LUTs) in the FPGA. A comparative study was made between the proposed design and designs based on conventional methods with respect to hardware utilization, power consumption and speed. The DA based PID controller saved

80% on hardware utilization and 40% on power consumption. Thus DA based implementations offers good closed loop performance while using less resources and lower power consumption. A very similar work was presented by Chan et al (2004) where the goal was to implement efficient design schemes for PID controllers using FPGA technology. The algorithm was implemented using a Distributed Arithmetic (DA) based scheme using Look-Up-Table mechanism. Two novel DA-based PID controllers were proposed and a comparative study compared power and resources. With the two DA designs only 13% and 4% of logic elements were utilized compared with the multiplier implementation thus reducing the power consumption by 40%. According Sreenivasappa et al (2009) a PID controller was implemented in two different FPGAs and their results compared. Results are compared in terms of power consumption, speed, memory usage, Look up Tables (LUTs), and number of multipliers. Xilinx and Altera FPGAs were the two different FPGA's in which the PID control algorithms were implemented and their results compared. Xilinx FPGA fared better in terms of number of multipliers, power and speed. Altera FPGAs obtained good results for memory usage, counters and LUTs. Hence it can be concluded that FPGAs offer flexibility and higher performance without increasing cost and resources. Most of the published research work emphasized speed, power consumption, and cost reduction for FPGA-based controller hardware.

2.2.3 Design of Experiments (DOE) techniques

In conventional PID controllers, properly tuned gain values are a prerequisite for good system performance. Depending on the dynamic nature of the system, which is time dependent, it is necessary to tune the gains for optimum performance. Obtaining gains for

plants without the mathematical model is haphazard. There are very few novel approaches proposed in the literature; however, most of the techniques are computationally intensive and difficult to implement in real time system. One of the most widely adopted conventional techniques is the Ziegler-Nichols method. It has been reported in some cases the Ziegler-Nichols method may result in excessive overshoot of the response and are not applicable to plants that have long time delays. Optimization is one of the key factor and most discussed topics in applied research and control engineering. Hence in this research work, a computationally efficient, novel method is proposed to obtain optimal controller gains using Response Surface Methodology (RSM). RSM has become common practice in engineering design and is extensively used in industry. In RSM, the input variables (i.e.; factors) affecting the product, system performance, or response must be determined. Subsequently, a simultaneous testing of factors over a set of limited experimental runs is performed and statistically analyzed for obtaining optimized values. RSM also provides quantitative measurement of possible interaction between the factors in the experiment.

Research work reported by Santhakumar & Asokan (2010), Demirtas & Karaoglan (2012) employs the Taguchi method and Response Surface Methodology for obtaining optimal controller gains. In Santhakumar & Asokan (2010) Taguchi method was proposed for self-tuning of an autonomous underwater vehicle. Tuning based on the Taguchi method resulted in optimal tuning of gain values with less computational effort. The proposed method is suitable for both Single Input and Single Output (SISO) systems as well as Multiple Input Multiple Output (MIMO) systems. The simulation results were compared

with the standard Ziegler- Nichols method. The selected performance index, namely the Integral Square Error (ISE) was significantly reduced.

Demirtas and Karaoglan (2012) propose Response Surface Methodology for tuning the Proportional Integral (PI) parameters for a motor. A comparative study was made between the experimental and simulation results for the response of the system. A RSM technique provided optimal values of PI gain parameters. The RSM technique determined the mathematical relationship between the response and the input parameters which were further analyzed using MINITAB optimizer.

Chapter 3

Experimental apparatus

3.1 Introduction

Balancing an inverted pendulum is a classic control system problem. When studying classical control theory, inverted pendulums are often used as an example of stabilizing an inherently unstable system. In practice, the system is inherently nonlinear, which is useful for illustrating concepts in nonlinear control. In this system an inverted link of mass m , forms an angle θ with respect to the vertical axis of a cart of mass M . The cart is free to move in the x direction along the track as shown. A horizontal force F is applied to the cart which is constrained to move in the horizontal direction. The feedback signals include the position of the cart and the angular position of the pendulum. With a properly designed and tuned controller, the pendulum's angle is actively maintained at zero. Figure 3-1 depicts an inverted pendulum system.

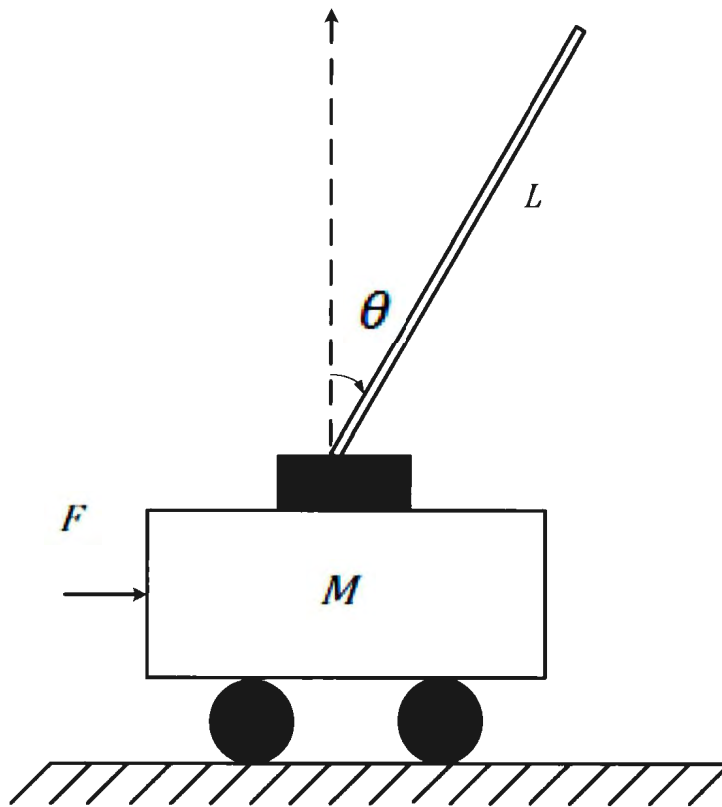


Figure 3-1 Diagram of Inverted Pendulum

3.2 Description of the system

The physical implementation of the inverted pendulum system is shown in Figure 3-2 below. A translation stage consisting of a servomotor, ball screw, table, and guide rails is used to displace the base of the pendulum. A Pittman DC servomotor (Model No. 9232 s003-R1) is directly coupled to a ball screw with a pitch of 0.2" (5.08 mm). The ball screw converts the rotational motion of the motor to translational motion of the table. The motor produces a continuous torque of 2.4 Nm at 24 V. Two optical encoders are used to provide feedback. The first is directly coupled to the servomotor and has a resolution of 500 counts per revolution (2000 counts in quadrature). It is used to measure the angular

position of the ball screw and hence the translation of the table. A second encoder with a resolution of 2500 counts per revolution (10,000 counts in quadrature) is used to measure the angle of the pendulum.

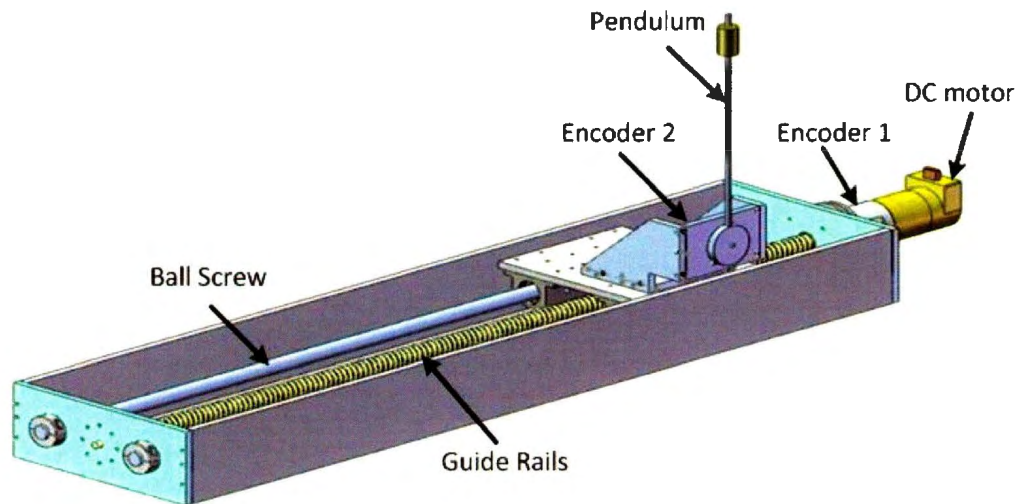


Figure 3-2 Mechanical system with pendulum

3.3 System modeling

3.3.1 Modeling the motor

A classical DC motor includes two sets of windings, i.e., stator windings and rotor windings. This type of motor can be controlled by adjusting the current through either set of windings. In modern servomotors, the stator winding are often replaced by rare-earth permanent magnets and the motor can only be controlled by adjusting the current through the rotor windings. The rotor (attached to the shaft and commutator) constitutes the

rotating armature circuit depicted in Figure 3.3. The stator or stationary part of the motor includes the motor casing along with the permanent magnetic pole pieces that generate a stationary, constant magnetic field (i.e.; the “Field Circuit” in Figure 3-3). Permanent magnet servomotors can only be controlled by varying the armature current, I_a .

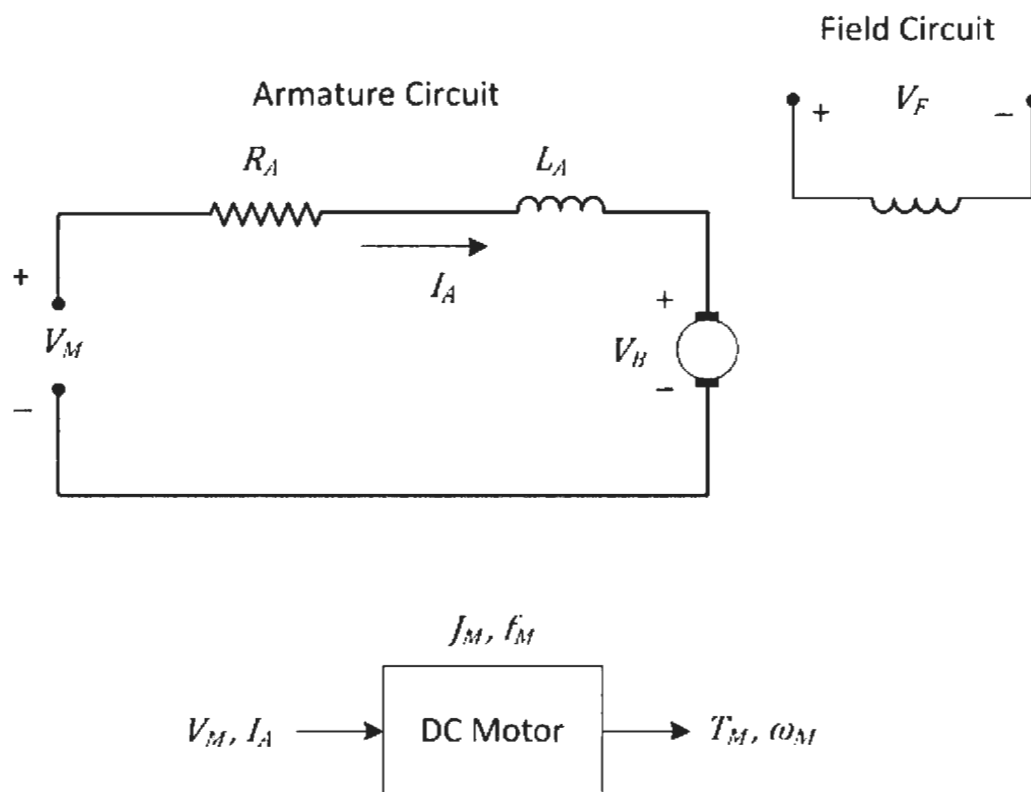


Figure 3-3 General motor model

Electrical model

When current flows through the rotating armature windings, the impedances in the armature circuit are represented by a resistive component, R_a , in series with an inductive component, L_a . By applying Kirchhoff's Voltage Law (KVL) the armature current can be expressed in terms of the applied motor voltage, V_M , as follows:

$$V_M = R_a I_a + L_a \frac{dI_a}{dt} + V_B \quad (1)$$

The voltage V_B represents the generated back Electromotive Force (EMF) of the motor. As the motor shaft rotates, the windings in the rotor move through a fixed magnetic field causing the motor to act as a generator. In accordance with Lenz's Law, as the angular velocity of the motor increases, the voltage generated by the back EMF also increases in direct proportion to the angular velocity of the motor. The back EMF ultimately limits just how fast the motor can turn. In other words, as the motor accelerates, the back EMF begins to approach the applied voltage and both the armature current and motor torque tend towards zero. The back EMF voltage, V_B , can be expressed as a function of the angular velocity of the motor, ω , as follows:

$$V_B = K_B \omega = K_B \frac{d\theta}{dt} \quad (2)$$

K_B is the back EMF constant, which is numerically identical to the torque constant, K_m , of the motor in the SI system of units. The armature current, I_a , can be expressed in terms of the angular velocity of the motor, ω , by substituting $K_B \omega$ for V_B in Equation 1:

$$V_M = R_a I_a + L_a \frac{dI_a}{dt} + K_B \omega \quad (3)$$

Assuming zero initial conditions, the above equation can be expressed in the Laplace domain as follows:

$$0 = -V_M + R_a I_a + L_a s I_a + K_B s \theta \quad (4)$$

Solving for I_a :

$$I_a = \frac{V_M - K_B s \theta}{(R_a + L_a s)} \quad (5)$$

Equation 4 can be expressed in block diagram form along with the torque constant, K_m , and the motor torque, T_M , as follows:

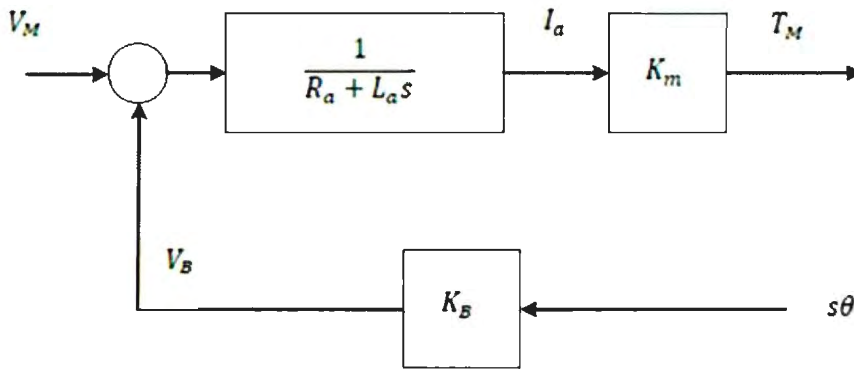


Figure 3-4 Block diagram representation of Equation (5)

Mechanical model

In order to derive a complete motor model, a relationship between the torque and angular velocity of the motor must be formulated. Consider the mechanical model of the motor.

Torque generated by the motor accelerates the armature of the motor as well as the load inertia attached to the shaft. In this model only viscous motor friction is considered. According to Newton's second law the sum of the applied torques is equal to the product of the mass moment of inertia and angular acceleration. The total mass moment of inertia equals the mass moment of inertia of the motor armature, J_M , and the load, J_L .

$$T_M = (J_M + J_L)\ddot{\theta} + f_M\dot{\theta} \quad (6)$$

J_M - Moment of inertia of motor

J_L - Moment of inertia of load

f_M - Viscous damping friction of motor

$\dot{\theta}$ - Angular velocity of the motor

Assuming zero initial conditions, the above equation can be expressed in the Laplace domain as follows:

$$T_M = (J_M + J_L)s^2\theta + f_M s\theta \quad (7)$$

$$T_M = s\theta [(J_M + J_L)s + f_M]$$

The transfer function relating the motor angular velocity to the motor torque can be written as follows:

$$\frac{s\theta}{T_M} = \frac{1}{(J_M + J_L)s + f_M} \quad (8)$$

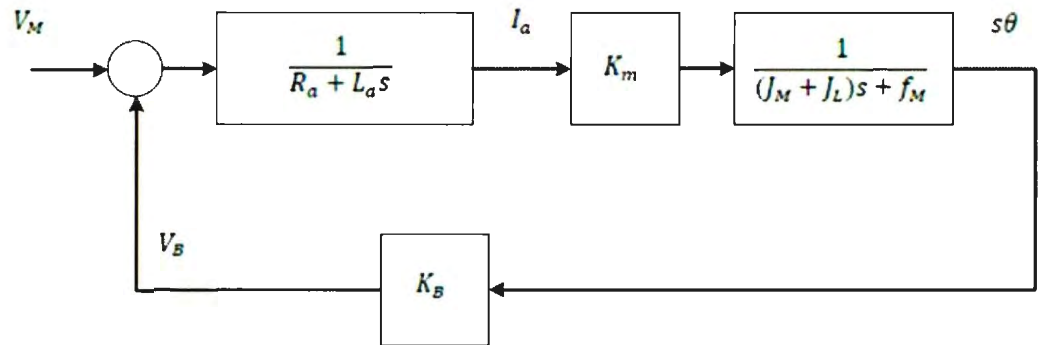


Figure 3-5 Block diagram representation of Equation (8)

Reduction of block diagram models

Block diagram reduction is used to represent complex models into its simpler form i.e., representing the different blocks into single block. Considering the block diagram shown below is represented in Laplace domain as follows:

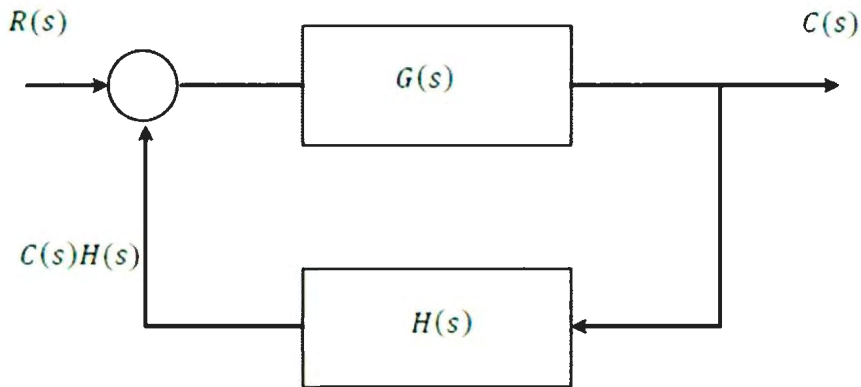


Figure 3-6 Control system

$$[R(s) - H(s) C(s)] G(s) = C(s) \quad (9)$$

$$G(s) R(s) - G(s) H(s) C(s) = C(s) \quad (10)$$

$$G(s) R(s) = C(s) + G(s) H(s) C(s) \quad (11)$$

$$R(s) G(s) = C(s) [1 + G(s) H(s)] \quad (12)$$

$$\frac{C(s)}{R(s)} = \frac{G(s)}{1 + G(s) H(s)} \quad (13)$$

Therefore the block diagram below is the representation of equation (13)

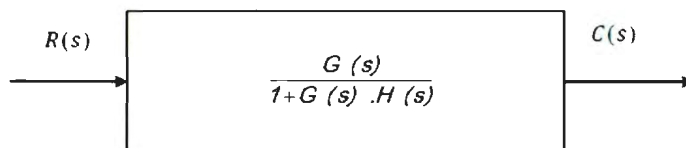


Figure 3-7 Block diagram representation of Equation (14)

In case of the DC servomotor $G(s)$ is given by,

$$G(s) = \frac{1}{R_a + L_a s} \cdot K_B \cdot \frac{1}{(J_M + J_L)s + f_M} = \frac{K_B}{(R_a + L_a s)[(J_M + J_L)s + f_M]}$$

The transfer function relating the angular velocity to the applied motor voltage

$$\frac{s\theta}{V_M} = \frac{\frac{K_M}{(R_a + L_a s)(J_M + J_L s + f_M)}}{1 + \frac{K_M K_B}{(R_a + L_a s)(J_M + J_L s + f_M)}}$$

One can simplify the above expression by multiplying both numerator and denominator by:

$$[(R_a + L_a s)(J_M + J_L s + f_M)]$$

$$\frac{s\theta}{V_M} = \frac{K_M}{(R_a + L_a s)[(J_M + J_L)s + f_M] + K_M K_B} \quad (14)$$

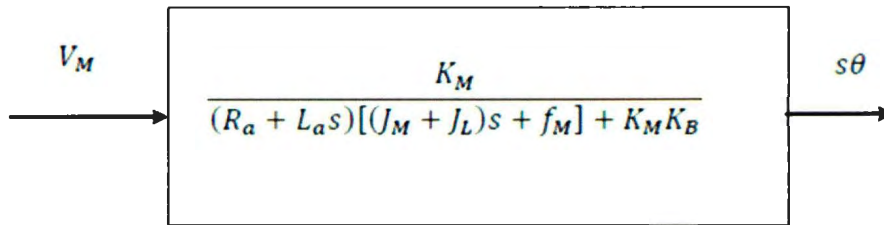


Figure 3-8 Block diagram representation of Equation (14)

Calculating electrical time constant of a DC servomotor

The electrical time constant of a DC motor, τ_E , can be determined by considering the electrical part of the motor model.

$$I_A = \frac{V_M - K_B s \theta}{(R_a + L_a s)} \quad (15)$$

Rearranging the above equation (15)

$$\frac{I_A}{V_M - K_B s \theta} = \frac{1}{R_a + L_a s} \quad (16)$$

$$\frac{1}{R_a + L_a s} = \frac{\frac{1}{R_a}}{1 + \frac{L_a}{R_a} s} \quad (17)$$

The Electrical time constant is therefore given by

$$\tau_E = \frac{L_a}{R_a}$$

In this case the motor used was a Pittman 9232s003-R1 with the armature resistance, R_a , provide by the manufacturer in the datasheet as 7.38Ω , and the armature inductance, L_a , as 4.64 mH. This results to the electrical time constant as 1.5ms.

Calculating Mechanical Time Constant of a DC motor

Considering the transfer function i.e., equation (14)

$$\frac{s \theta}{V_M} = \frac{K_B}{(R_a + L_a s)[(J_M + J_L)s + f_M] + K_M K_B}$$

In general mechanical time constant is larger than τ_E because the armature inductance L_a is quite small. Hence in most practical cases τ_E is neglected. Therefore the expression yields:

$$\frac{s \theta}{V_M} = \frac{K_B}{R_a[(J_M + J_L)s + f_M] + K_M K_B} \quad (18)$$

Therefore mechanical time constant is

$$\tau_M = \frac{R_a(J_M + J_L)}{R_a f_M + K_M K_B}$$

Where

$$R_a = 7.38 \, \Omega$$

$$J_M = 1.9 \times 10^{-6} \, \text{N-m-s}^2$$

$$J_L = 0 \, (\text{assuming no load})$$

$$f_M = 1.8 \times 10^{-6} \, \text{N-m-s/rad}$$

$$K_m = 3.11 \times 10^{-2} \, \text{N-m/A}$$

$$K_B = 3.11 \times 10^{-2} \, \text{V-S/rad}$$

Which yields a mechanical time constant, $\tau_M = 5.5 \, \text{ms}$.

Simulink model of the motor:

Motor and Ball screw model

A dynamic model for the motor was derived in earlier section. It consists of two first order functions; one for the armature circuit and the other for inertia/friction of the platform. The motor back emf acts as feedback in the circuit, opposing the input voltage. The friction torque which opposes motion is represented using a sign function. The system can be modeled using Matlab – Simulink as shown in figure 3-9.

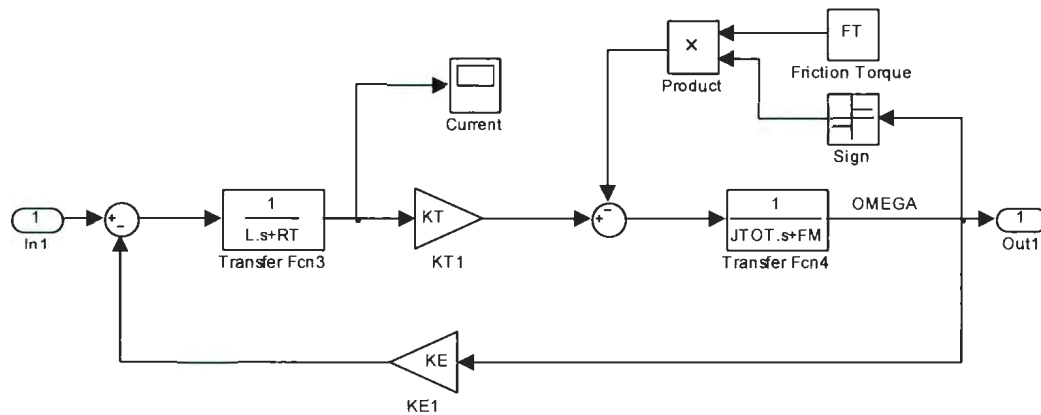


Figure 3-9 Simulink model of motor

Figure 3-10 shows the Matlab-Simulink model of the motor with the control system and feedback from the encoder. A PID type controller is implemented and a saturation block is used in order to represent the 10 bit H-bridge driving the motor.

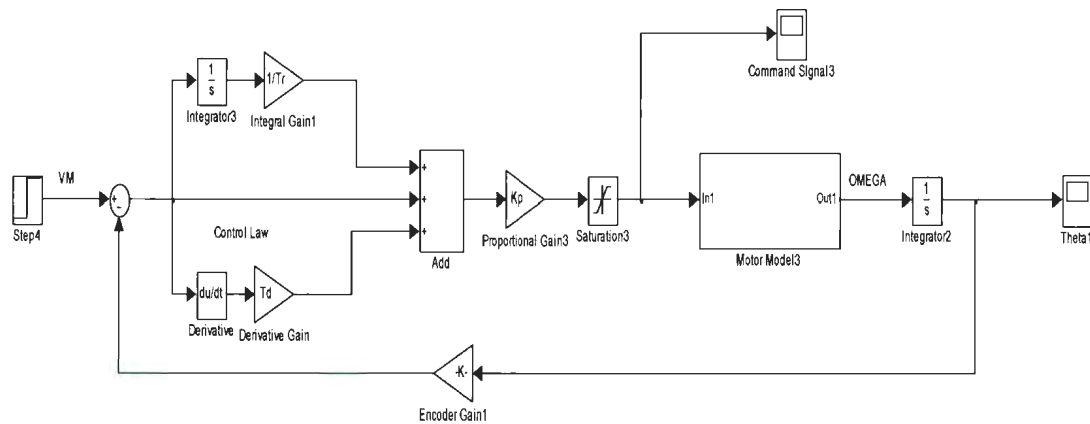


Figure 3-10 Motor PID controller

The response of the motor to a step input of 2000 pulses of the encoder is shown in Figure 3-11. The system reaches steady state in 23 ms.

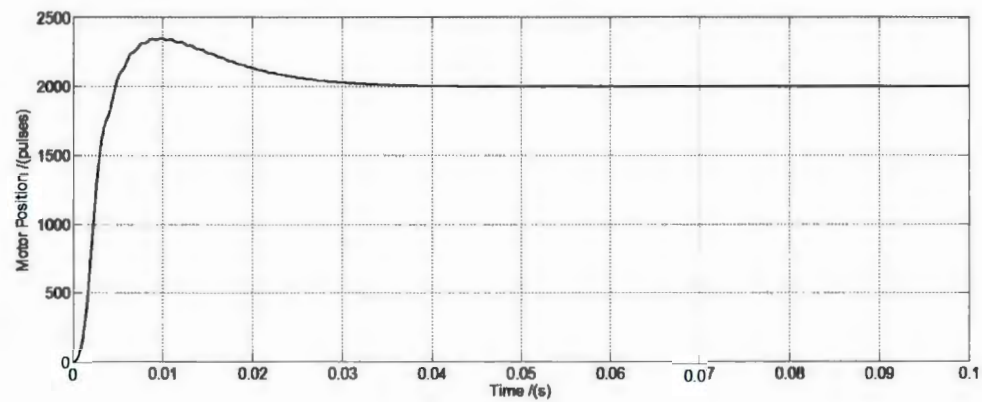


Figure 3-11 Response to a step input

3.3.2 Modeling the pendulum

Setup description:

The pendulum is mounted on a moving cart. A servomotor controls the translational motion of the cart by means of a ball screw. The encoder connected to the servomotor provides feedback on the position of the cart while a second encoder measures the angular motion of the pendulum. Movement of the cart applies the moments on the pendulum to maintain it upright.

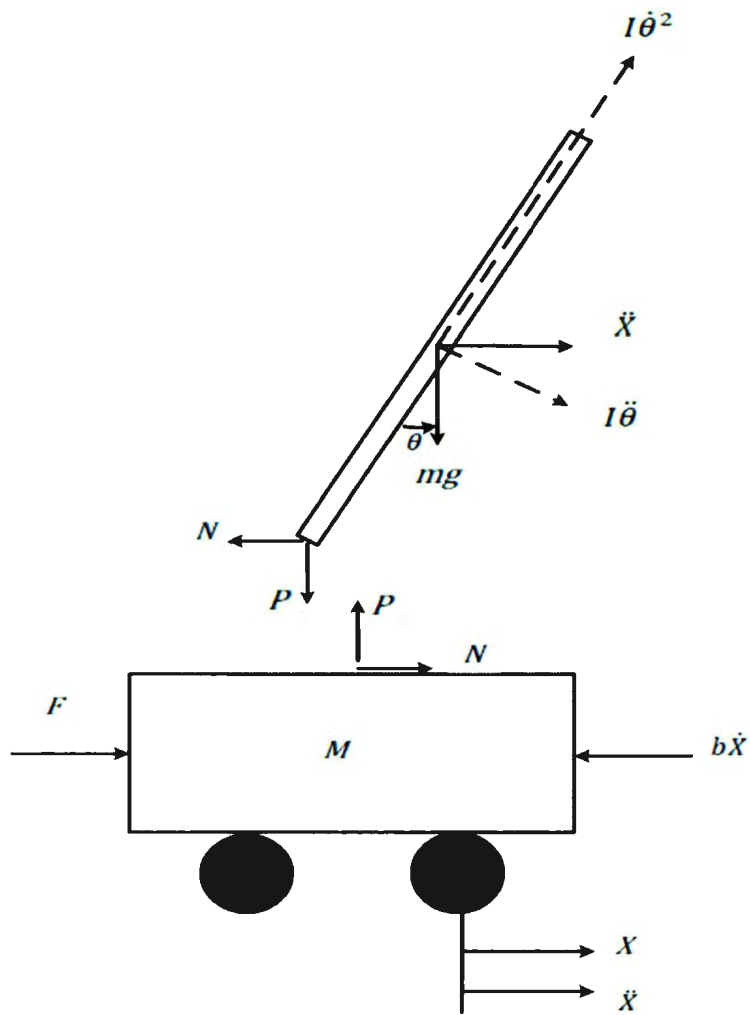


Figure 3-12 Free body diagram of the system

System equations:

Considering the free body diagram of the system shown above, the equations of motion can be determined.

Summing the forces acting on the cart in horizontal direction, the equation of motion obtained is:

$$M\ddot{X} + b\dot{X} + N = F \quad (19)$$

Summing of the forces in vertical direction is not necessary as the earth's reaction force balances it.

Force exerted on the pendulum due to moment is expressed as

$$\tau = r * f = I\ddot{\theta}$$

$$f = ml\ddot{\theta}$$

Component of this force in the direction of N is $ml\ddot{\theta} \cos \theta$

The inertial force acting along the horizontal direction is:

$$F = \frac{l\dot{\theta}^2}{r} = ml\dot{\theta}^2$$

The component of this force in the direction of N is $ml\dot{\theta}^2 \sin \theta$

Summing the forces in the free body diagram of the pendulum in horizontal direction is:

$$N = m\ddot{X} + ml\ddot{\theta} \cos \theta - ml\dot{\theta}^2 \sin \theta$$

Substituting the above expression of N in equation (19)

$$(M + m)\ddot{X} + b\dot{X} + ml\ddot{\theta} \cos \theta - ml\dot{\theta}^2 \sin \theta = F \quad (20)$$

To obtain second equation of motion sum of forces along the perpendicular of pendulum is considered

$$P \sin \theta + N \cos \theta - mg \sin \theta = ml\ddot{\theta} + m\ddot{X} \cos \theta \quad (21)$$

Summing the moments around the centroid of the pendulum

$$-Pl \sin \theta - Nl \cos \theta = I\ddot{\theta} \quad (22)$$

Combining equations (21) and (22)

$$(I + ml^2)\ddot{\theta} + mgl \sin \theta = -ml\ddot{X} \cos \theta \quad (23)$$

Therefore the set of equations defining the dynamics of inverted pendulum are:

$$(M + m)\ddot{X} + b\dot{X} + ml\ddot{\theta} \cos \theta - ml\dot{\theta}^2 \sin \theta = F$$

$$(I + ml^2)\ddot{\theta} + mgl \sin \theta = -ml\ddot{X} \cos \theta$$

A Matlab-Simulink model (Figure 3.13) can be determined by isolating the highest order derivative of θ and integrating twice.

$$\ddot{\theta} = \frac{-ml\ddot{X} \cos \theta - mgl \sin \theta}{I + ml^2}$$

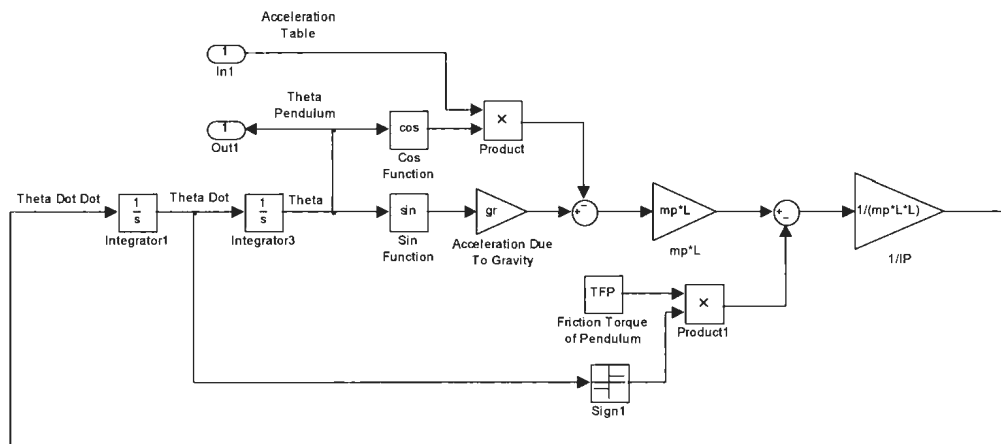


Figure 3-13 Pendulum model in simulink

Simulink model of the inverted pendulum system:

Figure 3-13 depicts the complete model of the Inverted Pendulum System which incorporates both the servomotor model and pendulum models. The controller implemented in the Simulink model is a cascaded PD type controller. A detailed study of PD type controller is explained in the proceeding chapters. Initially the controller gains

were tuned using a trial and error method as in Figure 3-14 to achieve the responses shown in Figures 3-15 and 3-16.

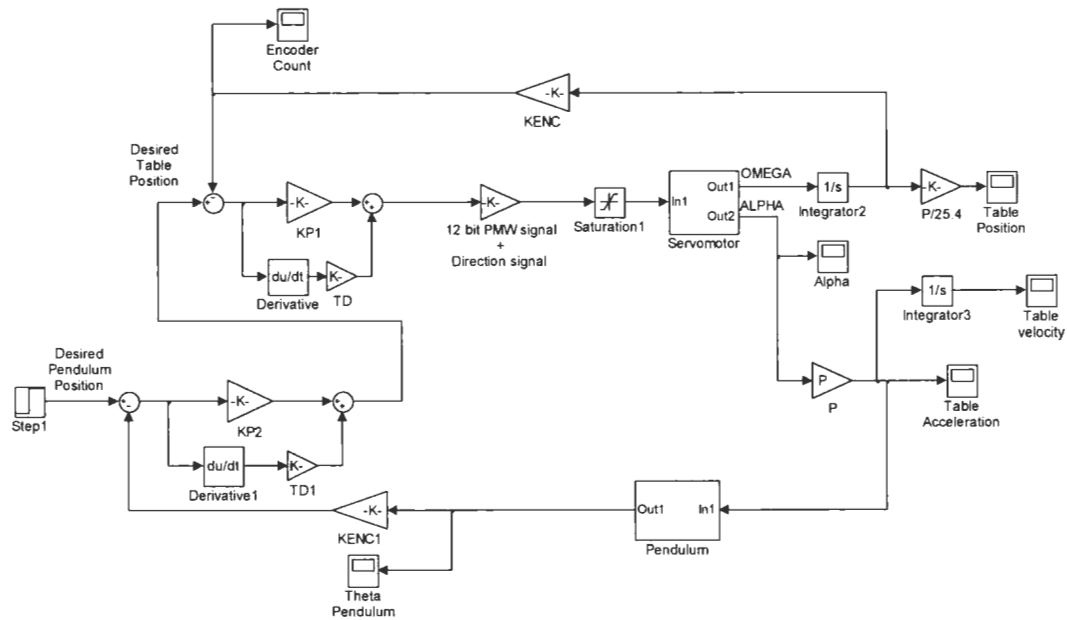


Figure 3-14 Simulink model of the system with initial gain values

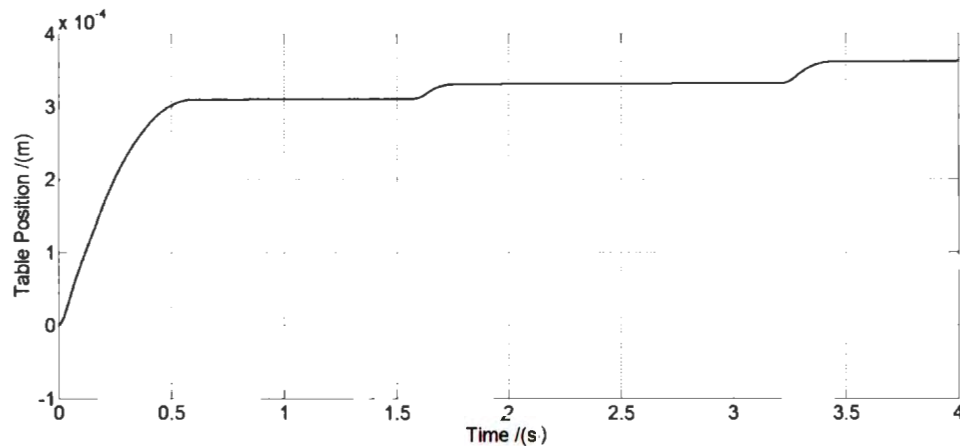


Figure 3-15 Position of the table

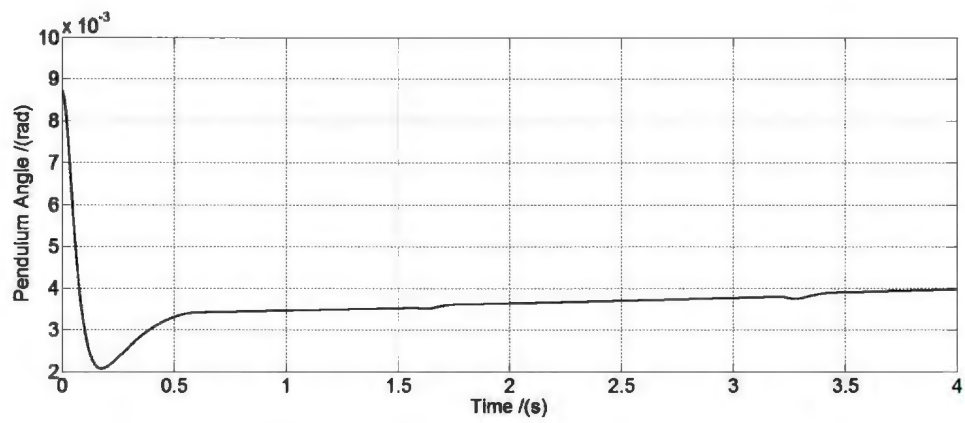


Figure 3-16 Pendulum angle in radiance

Chapter 4

Architecture of proposed FPGA based controller

4.1 Overview

This chapter explains a high-performance architecture designed for controlling an inverted pendulum system based on an FPGA. The main objective of this architecture is to utilize the ultra-high-speed hardwired logic of the FPGA. A detailed description of each module of the controller is presented. The design is partitioned into reconfigurable hardware and reprogrammable software on a single FPGA chip. If the system specifications are met the design is complete, else either the hardware development or software needs to be redesigned. Once the system is designed, the system prototype is usually tested on a development board featuring an FPGA device and other components useful for prototyping. In generating designs to be implemented onto FPGAs, there are multiple methods of developing the design. The methods can be divided into graphical, code, or a combination of both. The graphical methods, such as schematic capture, provide a drag and drop approach, which allows specific components to be connected to form a design. The behavior of some of these components can be modified, and new components created, to allow addition design flexibility. The advantage of a graphical method is that allows a visual layout and provides the user a means to visualize the actual hardware being designed. In our design a graphical design method was implemented. As discussed, the hardware design was accomplished using VHDL coding techniques in the

Quartus II environment provided by Altera whereas software development was accomplished by C programming in the Eclipse IDE tool.

4.2 FPGA architecture

The proposed controller architecture for the system is illustrated in Figure 4-1. A graphical interface was assigned as the top-level entity in the project. Design specifications for each module were entered using the schematic level, VHDL, AHDL, and Megawizard plug-in manager. Altera provides a Library of Parameterizable Megafunctions (LPM) that implements standard building blocks for circuit design. LPM functions are predefined library functions that can be used in the project. By using LPM blocks the design time can be reduced. The controller architecture modules include: i) IO modules and ii) Processor Modules.

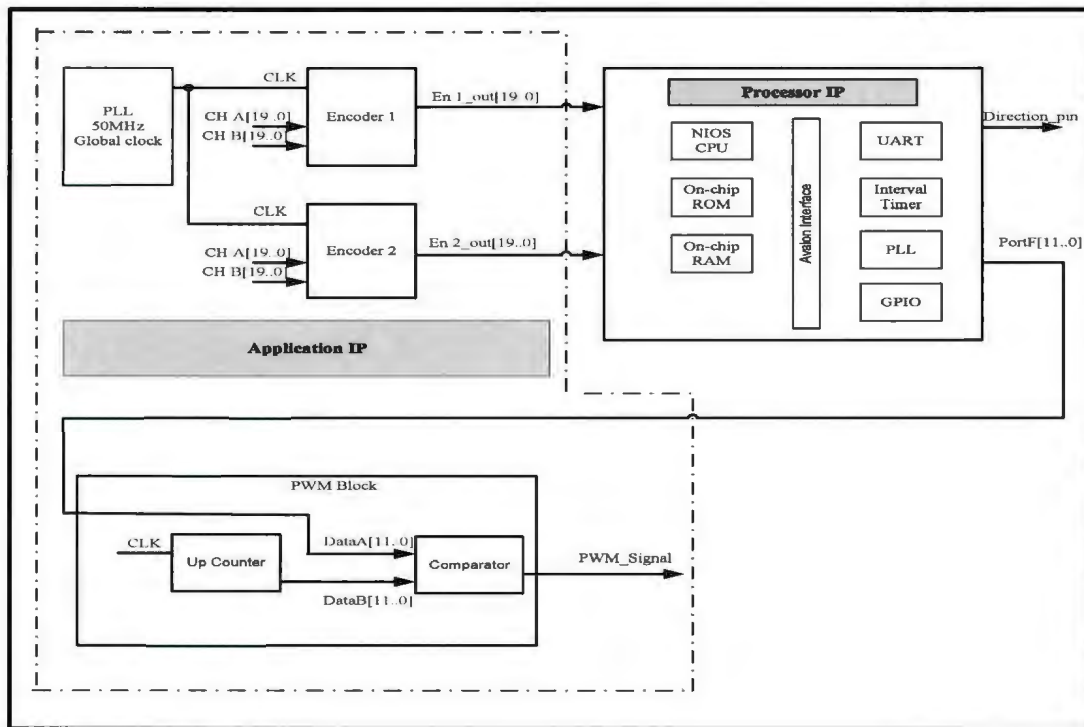


Figure 4-1 Internal architecture of the proposed controller

4.2.1 IO Modules

IO modules include PLL, encoder and PWM modules. PLL and encoder blocks act as inputs to the system whereas the PWM block acts as an output to the H-bridge which drives the motor of the system.

PLL:

One of the primary uses of a PLL is to synchronize the phase and frequency of the 50 MHz internal clock to an input reference clock. The Phase-Locked-loop (PLL) is a closed-loop frequency-control system that compares the difference between the input signal and output signal of an oscillator. Using the Megawizard-plugin manager the

Altpll megafunction was created which generates and customizes clock signals and distributes clock signals to different blocks in a design.

Encoder:

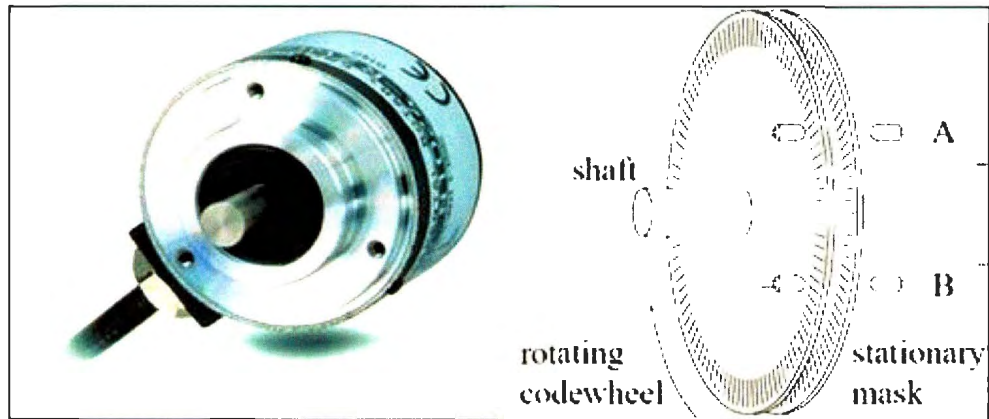


Figure 4-2 Quadrature decoder

http://hades.mech.northwestern.edu/index.php/File:Encoder_diagram.png

The encoder block is an input to the system which serves the purpose of controlling the DC servomotor (eg: position of the cart) and angle of the pendulum (pendulum mounted on the cart). In digital motion control systems encoders are used to translate the rotary motion of a shaft into digital form. Optical encoders have a Light Emitting Diode (LED) as emitter and photodiode as detector. As the code wheel rotates between the emitter and the detector, light from the emitter is interrupted by the slots in the code wheel as shown in figure 4-2. The position of the shaft is evaluated by counting the pulses generated by the detector. A second emitter/detector pair is placed on the circumference of the code wheel in such a way that when the first detector (Channel A) reads the slot the second detector (Channel B) reads the bar. Channel A and Channel B are quadratic to each other; i.e., 90° phase shift. Thus the encoder can accommodate clockwise and counterclockwise

rotation. When Channel A leads Channel B, the counter increments and when Channel B leads Channel A, the counter decrements (Krouglicof, 2004).

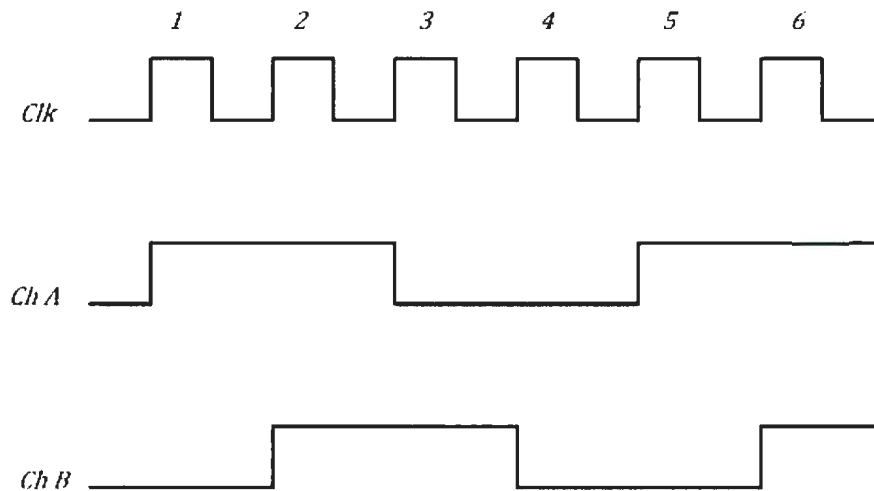


Figure 4-3 Quadrature decoder timing diagram

Figure 4-3 illustrates that channels A and B can be in one of four possible states. Depending on the past and present state, the counter is either incremented or decremented as in table 4-1. Therefore the resolution of the counter corresponds to four times the basic resolution of the code wheel. A 2,500 slot code wheel yields an effective resolution of 10,000 counts per revolution.

Table 4-1 State Transitions

State	Channel A	Channel B
1	1	0
2	1	1
3	0	1
4	0	0

In this design a 20-bit encoder block was designed using AHDL (Altera Hardware Description Language) in the hardware process. The logic implemented in AHDL involves the edge detectors to detect transitions between the states in channels A and B by employing D-type flip flops as in Figure 4-4. For example, in order to detect the rising edge on channel A, *Enc_D0* and *Enc_D1* must be “1” and “0” respectively. Table 4-2 illustrates the transitions between the states and their respective outputs.

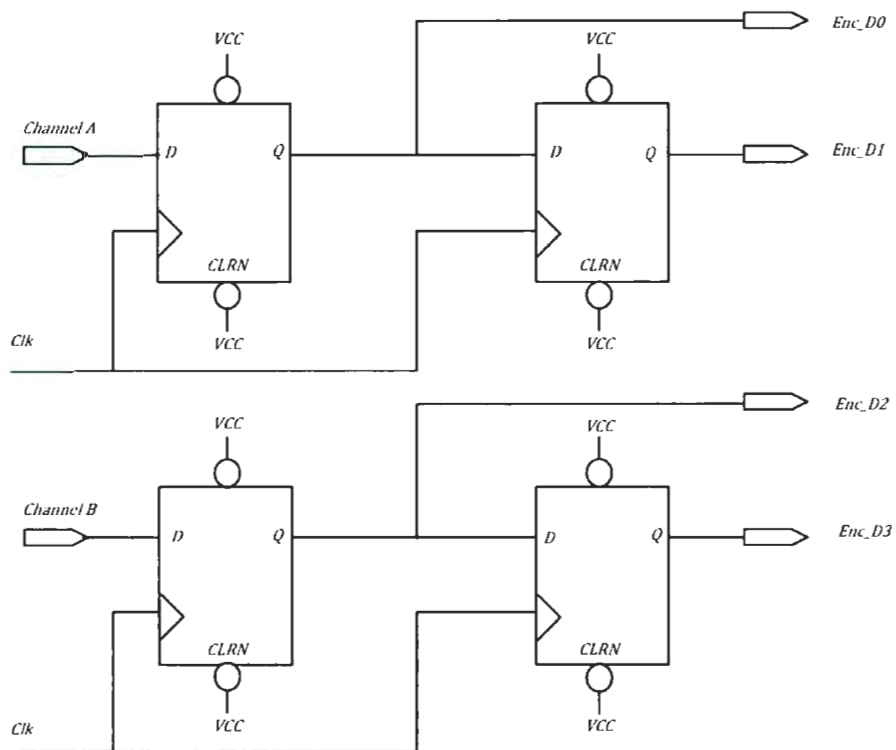


Figure 4-4 Edge detection circuitry

Table 4-2 Decoding Logic

States		Count	Edge Detection		Enc_D0	Enc_D1	Enc_D2	Enc_D3
Past	Present		Channel A	Channel B				
1	2	Up	1	Rising	1	1	1	0
2	3	Up	Falling	1	0	1	1	1
3	4	Up	0	Falling	0	0	0	1
4	1	Up	Rising	0	1	0	0	0
1	4	Down	Falling	0	0	1	0	0
2	3	Down	0	Rising	0	0	1	0
3	2	Down	Rising	1	1	0	1	1
4	1	Down	1	Falling	1	1	0	1

Based on the truth table above, the encoder block was implemented in Quartus. In NIOS using *encoder1_input_BASE* and *encoder2_input_BASE* the base addresses of the encoders are read. Error is estimated from the difference of the actual encoder's values "*enccount*" and "*enccount_one*" and the desired values. Thus the output to the controlled device is a function of error.

PWM module:

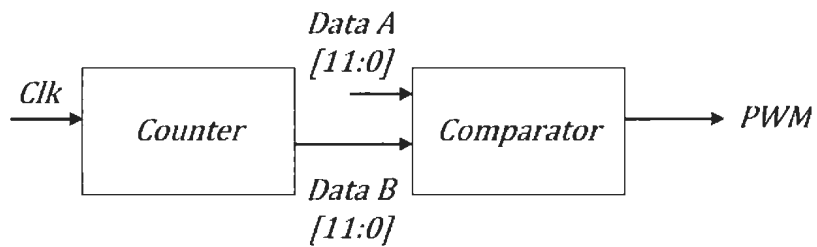


Figure 4-5 PWM module

The motor was driven using a PWM signal comprising of a 5V signal with a frequency of 12 kHz. The hardware PWM module was implemented using a counter and a comparator as in figure 4-5. To obtain a PWM signal the counter was compared with the desired duty cycle value, giving a PWM signal at the output of the comparator. The PWM signal for various duty cycles is shown in figure 4-6

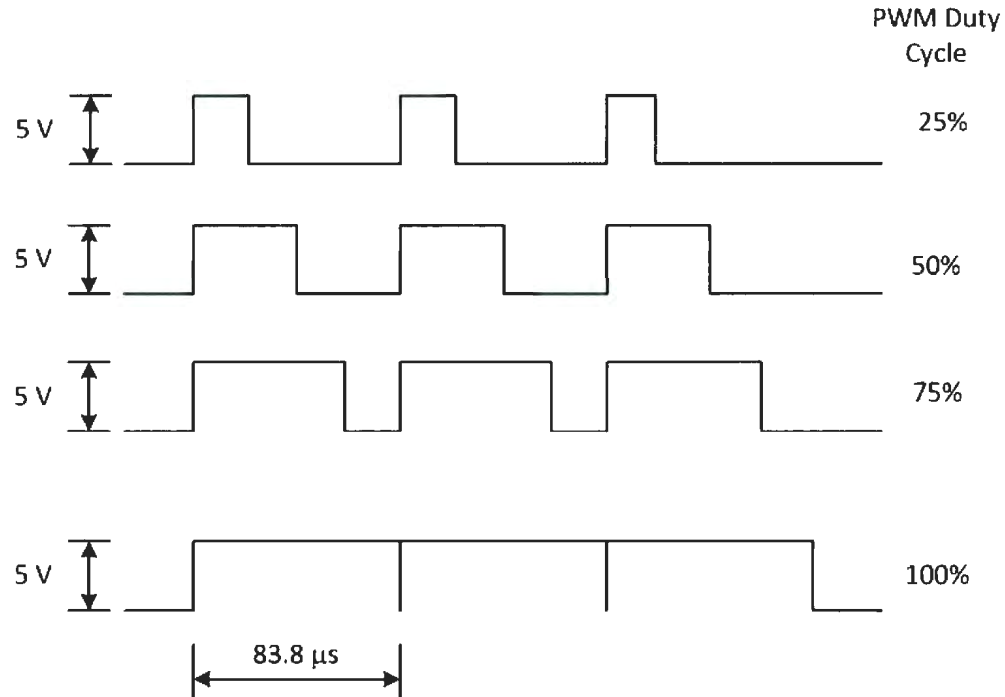


Figure 4-6 Modulated duty cycle

Counter Block:

Through parameterizable functions provided by Altera, LPM functions were generated and used in the project. Specifically a 12 bit binary counter was added to the design with a clock input of 50 MHz provided by the PLL. A 12-bit counter running at 50 MHz yield an output frequency of 12 KHz which is appropriate for driving DC servomotors. The output of the counter is fed to the comparator block.

Comparator Block:

LPM_COMPARE megafunction compares two sets of 12-bit data to determine the relationship between them. Thus it determines if the data's are equal and produces the PWM signal. The ON and OFF period of the duty cycle varies the speed of the motor or the torque of motor.

4.2.2 Processor module**Nios II Processor Core:**

The Nios II processor is a general purpose RISC processor with the following features:

- Full 32-bit instruction set, data path and address space.
- 32 general-purpose register.
- 32 interrupt sources.
- External interrupt controller interface for more interrupt sources.
- Hardware assisted debug module enabling processor to start, stop, step and trace under the control of Nios II development tool.
- Optional Memory Management Unit (MMU).
- Optional Memory Protection unit (MPU).

- Optional shadow Registers.
- Instruction Set Architecture (ISC) compatible with Nios II processor.
- Floating Point instructions.
- Performance up to 250 DMIPS.

The Nios II processor is equivalent to a microcontroller or a “computer on-chip” which includes processor, combination of peripherals and memory on a single chip. There are three processor cores.

Nios II/f: The Nios II/f fast core is designed for fast performance. The Nios II/f core can be fine tuned for performance.

Nios II/s: The Nios II/s standard core is designed for small size and reasonable performance.

Nios II/e: The Nios II/e economy core yields the smallest core size. By choosing this we are limited to only certain features.

The processor core chosen is a Nios II/s standard core which is optimal for cost sensitive and good performance applications.

Overview of Nios II/s:

- Can access upto 2 GB of external address space.
- Employs a 5 stage pipeline.
- Provides hardware multiply, divide and shift to improve arithmetic performance.
- Supports JTAG debug module.
- Supports custom instructions.

JTAG UART:

The JTAG Uart core with the Avalon Interface implements a method to communicate between the host PC and a SoPC builder system on an FPGA as shown in figure 4-7. In most of the designs JTAG eliminates the need of RS-232 serial connection for character I/O. The Nios II processor communicates with the Jtag core by reading and writing control and data registers. Altera provides JTAG terminal software to the host PC that manages the connection to the target, decodes the JTAG data stream and displays the character on screen.

The FPGA has built-in JTAG control circuitry between the devices built-in pins and the logic inside the device. During the process of logic synthesis and fitting, the Quartus II software automatically generates the JTAG hub logic. No manual design effort is required to connect the JTAG circuitry. The host PC connects to the FPGA via the Altera JTAG download cable known as the USB Blaster.

Interval Timer:

The Nios II processor has an Avalon based interval timer with the following features:

- 32-bit and 64-bit counters.
- Controls start, stop and reset timers.
- Two counting modes: countdown once and continuous count down.
- Option to enable and disable the Interrupt Request (IRQ) when the timer reaches zero.
- Optional watchdog timer feature.
- Optional periodic pulse generator feature.

- Compatible with 32-bit and 16-bit processors.

A simple periodic interrupt was implemented by enabling the Preset Configuration option pre-defined in the Hardware options. Simple periodic interrupt are useful in real-time systems where certain function must be executed at predetermined intervals. In this design, the period is fixed to every 1 Millisecond (ms) and the timer cannot be stopped but the IRQ can be disabled.

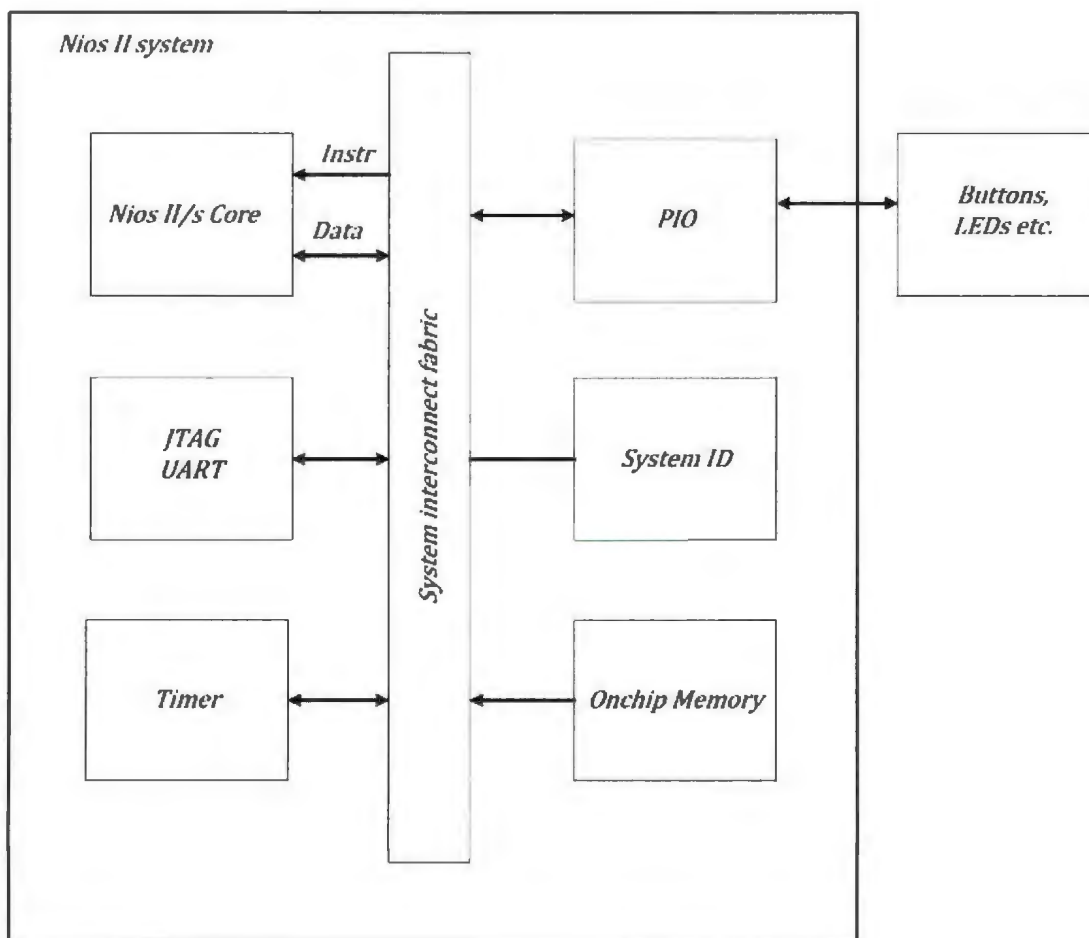


Figure 4-7 SoPC system with Nios II processor

Avalon Interface:

The Avalon bus supports dynamic bus sizing, so the peripherals with different data widths can be used on a single bus. It is designed for interconnection of on and off-chip processors and peripherals into a system on a programmable chip.

4.2.3 Hardware implementation

The development board chosen for this work was the Cyclone III Starter Kit with the main feature being low power consumption. The main reason for selecting this board is familiarity with Altera FPGAs and Altera design software products. A thorough understanding of the software tools is critical for efficient FPGA design. A second reason for selection of this FPGA was the size of the chip in terms of the number of logic gates as well as the number of off chip ports and chip speed. The main features of this board are:

- 256 Megabits (Mb) DDR SDRAM
- 1 Megabytes (MB) of synchronous SRAM
- 50 MHz onboard oscillator
- HSMC and USB type B connectors

The development environment used is the Quartus II web edition. The development tool is used to design the system using Hardware Description Languages or block diagrams provided by Altera. Quartus II design software is a comprehensive multiplatform design environment for specific design needs. It provides complete environment for System-on-Programmable-Chip (SoPC) design. The Quartus II software allows both Quartus II

graphical user interface and command-line interface for each phase of the design flow. Anyone of the interfaces can be adopted for the entire flow, or different options for different phases.

Design entry Methods:

Basic designs in Quartus are created by a project including design files, software source files and other related files in Quartus II Block Editor or Text editor.

Synthesis:

Once the design files are created the next process was to synthesis the design. By opting for analysis and synthesis, project database are created and the database are examined for logical completeness, consistency in the project, checks for boundary connectivity and syntax errors.

Place and Route:

In Quartus II, the “Fitter” places and routes the design hence referred to as “Fitting”. Generally the Fitter matches the logic and timing requirements of the project from the database created by Analysis and Synthesis tool to the available resources in the target devices.

Simulation:

Simulation compares the output of the design to the expected results. Quartus II supports two modes of simulation: timing and functional simulation. During the functional simulation, functionality of the designs obtained from the synthesis is verified. Timing simulation extracts the timing information from the Fitter and uses it to simulate the design.

Timing Analysis:

Timing analysis provides information about critical paths in the design by analyzing the netlist obtained from the Fitter.

Device Programming:

With the successful compilation of the project with the Quartus II software, the FPGA device can be configured. The assembler automatically converts the Fitters device, logical cells, and pin assignments to a programming image in the form of Programmer Object Files (.pof) or SDRAM Object Files (.sof). The device can be subsequently configured using a downloading tool; e.g., MasterBlaster, ByteBlaster, USB-Blaster or an Ethernet Download cable.

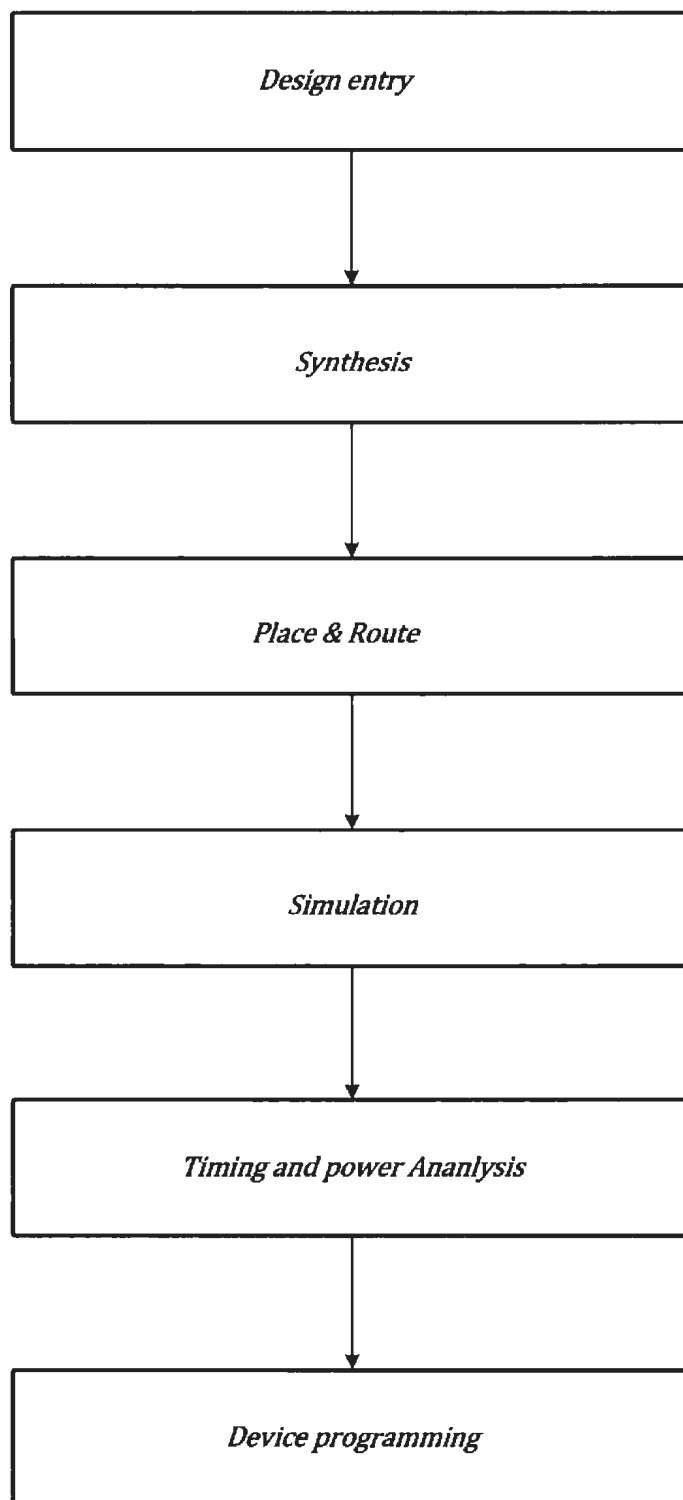


Figure 4-8 Design entry flowchart

Chapter 5

Novel method for tuning controller gains

5.1 Introduction

In a closed loop feedback system, if the mathematical model of a plant is derived, then it is possible to apply various design strategies to determine the parameters of the controller which meets the transient and steady state specifications. In some cases where the plant is so complicated that no mathematical model can be obtained, then an analytical or computational approach to the design of a PID controller is not possible. In such cases, in a classic paper, Ziegler and Nichols suggested a semi-empirical controller tuning method in which the controller parameters are tuned in such a way that it meets the given performance specifications. According to the Ziegler and Nichols method, tuning of a PID controller is based on experimental step responses or on the value of proportional gain, K_p , which results in marginal stability. The Ziegler and Nichols method can also be applied to the design of a system of known mathematical model. It suggests a set of values of gain K_p , T_i , T_d resulting in stable operation of the system; however, the system may yield a large overshoot in response to a step response which may be unacceptable. In general, additional fine-tuning is required until an acceptable output is obtained. The Ziegler and Nichols method suggests a starting point rather than giving the optimal values of gains for a stable system. Ziegler and Nichols suggested two tuning methods: First Method and Second Method. In this project, the second method of tuning is more applicable.

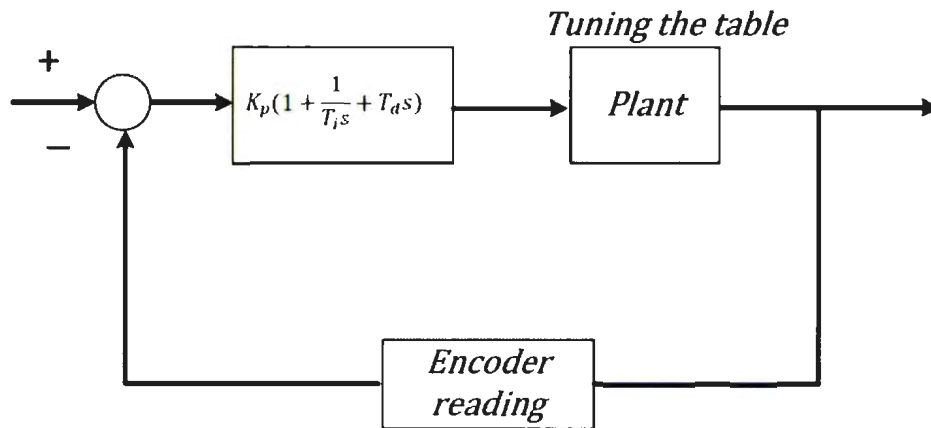


Figure 5-1 diagram of controller

5.2 Classical approach using Ziegler-Nichols method

In the second Ziegler and Nichols method (Ziegler & Nichols, 1942), using proportional control action alone, the proportional gain, K_p , is varied from 0 to a critical value, K_{cr} , that yields a sustained oscillatory output of period, P_{cr} . If the output does not result in sustained oscillations for any value of K_p then this method does not apply. Based on the K_{cr} and P_{cr} values, the gain parameters can be obtained from the formula table. If the system has a known transfer function, the root-locus method can be applied to determine the critical gain and the frequency of the sustained oscillations. In this study, the values obtained are:

Frequency = 50Hz

Critical Gain (K_{cr}) = 75

Sustained oscillation Period (P_{cr}) = 0.02

Table 5-1 Initial range of gain values

Type of Controller	K_p	T_i	T_d
P	$0.5K_{cr}$	-	0
PI	$0.45 K_{cr}$	$\frac{1}{1.2}P_{cr}$	0
PID	$0.6 K_{cr}$	$0.5 P_{cr}$	$0.125P_{cr}$

From the above table, PID gain values obtained were $K_p=45$, $T_i=0.01$, $T_d=0.0025$. In practical cases nonlinear effects are encountered that are not accounted for by the Ziegler and Nichols method. In any control system, the actuators have several limitations such as saturation; e.g., a valve fully opened or fully closed. Under the operating conditions when a control variable reaches the actuator limits, the feedback loop is broken and the system runs open loop. In other words, the actuator remains at its limit independent of the process output. Thus a controller with the integrating action becomes too large or results in integral wind-up, causing the error to be integrated continuously. Therefore the integral gain was ignored in this study and a PD type controller was implemented.

5.3 Proposed methodology for the controller

In this study dual PD type controller was implemented as a cascaded type controller. In Figure 5-2, the plant corresponds to the pendulum and table connected to a motor driven ball screw. In the cascaded controller, the outer feedback loop is based on pendulum angle (θ) as measured by the encoder attached to the pendulum. The inner loop is based on the position of the table (X) as measured by the encoder attached to the motor. The logic implemented for the cascaded controller is as follows:

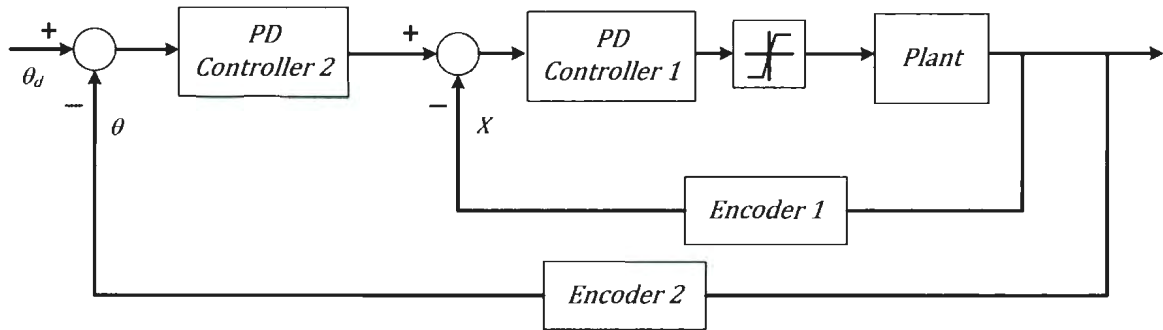


Figure 5-2 Diagram of proposed controller architecture

Algorithm:

Pendulum encoder reading = Read Encoder 2 ()

If (Pendulum encoder reading > 32767)

Pendulum encoder reading = Pendulum encoder reading - 65536

Pendulum error = Pendulum set point – Pendulum encoder reading

Pendulum derivative error = (Pendulum error – Pendulum Previous error)/time

Pendulum Previous error = Pendulum error

Motor set point = Pendulum error * K_{p1} + Pendulum derivative error * T_{d1}

```

Motor encoder reading = Read Encoder 1 ()

If (Motor encoder reading > 32767)

Motor encoder reading = Motor encoder reading - 65536

Motor error = Motor set point - Motor encoder reading

Motor derivative error = (Motor error - Motor Previous error)/time

Motor Previous error = Motor error

Control Signal = Motor error * Kp2 + Motor derivative error * Td2

If (Control Signal > 0)
{
Direction Signal = 0;

If (Control Signal > 4095)
{
Control Signal = 4095
}
}

If (Control Signal < 0)
{
Control Signal = - Control Signal

Direction Signal = 1;

If (Control Signal > 4095)
{
Control Signal = 4095

```

}

PWM = Control Signal

}

5.4 DOE based tuning method- an overview

In early stage engineering design, an often used approach is “Best-Guess” based on experience or engineering knowledge. Another prevalent experimental strategy is the one-factor-at-a-time (OFAT) approach. These approaches are inefficient and often result in an inappropriate solution. OFAT was once considered a standard and systematic approach until the early 1920’s when Ronald A. Fisher discovered the more efficient methods of experimentation based on factorial designs. Classes of experimental design include general factorial, two-level factorial, fractional factorial, response surface methodology and others. These statistical based experimental design techniques are referred to as Design of Experiments (DOE).

Basically DOE is a formal mathematical methodology to relate the input variables (i.e. factors) affecting the process and their possible interactions on the output (i.e. responses) of the process. This approach involves a series of tests carried out with planned changes made to the input variables of a process or system. The effects of these changes to the output of the process are analyzed using Analysis of Variance (ANOVA). DOE can be used widely in all fields of engineering, science and even in marketing studies. By using DOE proposed by Montgomery (2005) we can:

- Understand the process or system
- Screen the important factors

- Build a mathematical model
- Obtain prediction equations
- Optimize the response if necessary

DOE is based on statistical principles and methods such as analysis of variance and regression analysis. ANOVA is used to test the statistical significance of the model by the mathematical process of separating the variability of a group of observations into assignable causes. The prediction model is obtained using regression analysis.

5.5 Why Design of Experiment?

Statistical design of experiments (DoE) is an efficient means to simultaneously study the effect of several input factors on an output and determine the optimum setting for them. Design of experiments allows the main contribution of factors to a problem to be determined, how well does the system perform in the presence of noise and suggest the best optimal solution for the system. As stated earlier, in order to obtain the optimal gain values for a multivariable system, concepts of design of experiments are applied and analyzed. The response of the system is the Integral Squared Error (ISE) as shown in Figure 5-3 and optimizing implies, minimizing, maximizing or getting closer to the target.

The procedure for applying DOE is summarized as follows:

- Choose the process variables (input factors) and the response variables.
- Find a suitable experimental design depending on the objective of the experiment.
- Execute the design and analyze the model by model reduction, finding the significant factors in the model using ANOVA analysis and analyzing residuals for model adequacy.

- Validation is done by comparing the predicted versus observed values.

A detailed discussion of the experimental design chosen, interpretation of the results and validation of the model is described in the following sections.

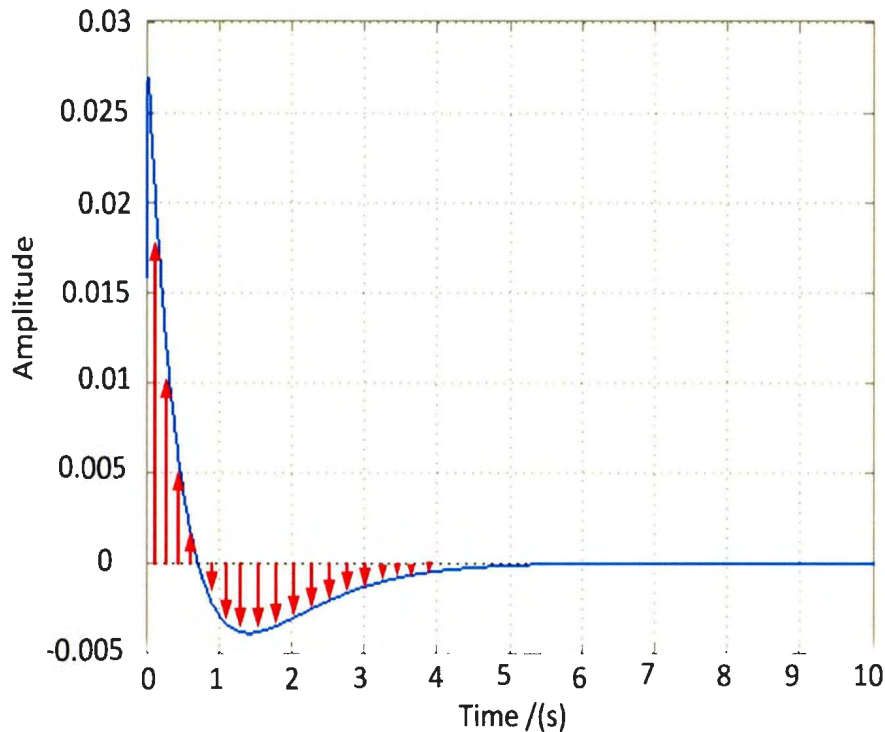


Figure 5-3 Integral squared error

5.6 Response Surface Methodology (RSM)

Response Surface Methodology (RSM) is a mathematical and statistical technique useful for modeling and analyzing the response of interest which is influenced by several variables with an objective of optimizing the response. The RSM technique is performed to establish a mathematical relationship between the responses and the input parameters.

Depending on the possible behaviors of response as a function of factor settings, the results could be a linear function, quadratic function or higher order functions.

Linear:
$$y = \beta_0 + \sum_{i=1}^k \beta_i x_i + \varepsilon$$

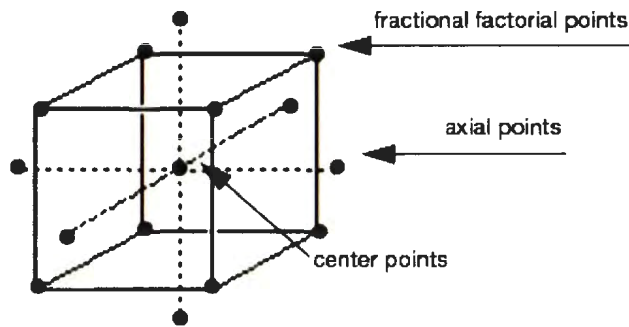
Interaction (2FI):
$$y = \beta_0 + \sum_{i=1}^k \beta_i x_i + \sum_{j < i} \sum_{i=1}^k \beta_{ij} x_i x_j + \varepsilon$$

Quadratic:
$$y = \beta_0 + \sum_{i=1}^k \beta_i x_i + \sum_{j < i} \sum_{i=1}^k \beta_{ij} x_i x_j + \sum_{i=1}^k \beta_{ii} x_i^2 + \varepsilon$$

where y represents the response, β_0 is the overall average, β_i is the regression coefficient, $x_i x_j$ represents the factor i and j respectively, k is the number of factors, β_{ij} are the interaction terms, β_{ii} represents pure second order or quadratic effects and ε is the error estimate.

If the response behaves as a linear function then a simple two-level factorial or fractional factorial designs can be adopted. In our system a logical assumption is made that there may be an existence of slight curvature in the system ruling out the factorial design technique. Generally RSM is a sequential procedure carried out in steps to locate the optimum point if that's the objective. The analogy of climbing a hill is an appropriate example illustrating that when we are far from the optimum there is a slight curvature in the system and it is appropriate to fit a first order model. Once in the vicinity of the optimum solution, a more elaborate second order model may be deployed and further analysis is performed by Montgomery (2005). If the curvature is significant more elaborate design techniques like the classical Central Composite Design (CCD) and Box-Behnken Design (BBD) can be carried out to fit a second order model to capture the optimum.

Central Composite Design



Box-Behnken Design

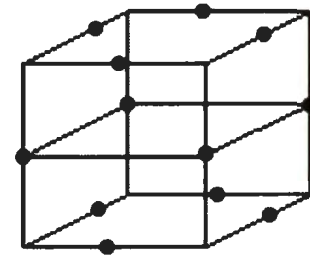


Figure 5-4 Response surface Designs

http://www.jmp.com/support/help/Response_Surface_Designs.shtml

5.6.1 Brief summary of BBD and CCD

Box Behnken Design

The Box-Behnken is an independent quadratic design that does not contain an embedded factorial or fractional factorial design. In BBD the treatment combinations are at the vertices of the cube and at the center as shown in Figure 5-4. This is sometimes useful when it is desirable to avoid these points due to system considerations. Each factor is varied with three levels alternative to CCD which requires 5 levels. There is always higher uncertainty of prediction near the vertices compared to the central composite design (Montgomery, 2005).

Central Composite Design

One of the most popular RSM techniques is CCD. It is an embedded factorial or fractional factorial design with center points in conjunction with axial or star points that

facilitate the estimation of curvature. CCD allows the user to fit a second order (quadratic) model to the response. A CCD design with K factors consists of:

$$\text{CCD} = 2^K \text{ factorial} + 2k \text{ star or axial points} + n_C \text{ center points}$$

- Factorial or fractional factorial design involves the factors under study with levels considered for each factors; i.e., the range of the factors.
- Center points are points at the center value of factor ranges. These are often replicated in the design to improve the precision of the experiment by calculating the pure error.
- Star or axial points (α) are the points on the coordinate axes at α distance from center.

5.7 Experiment methodology

This section of the chapter explains the methodology adopted for obtaining the optimal gain values using Design-Expert 8 to create and analyze the effects of factors affecting the system. Classical response surface methodology designs namely BBD and CCD techniques, were first carried out to analyze the effects. Four factors were considered in the design with two levels and five center points without any replications. The ranges considered for the gains are shown in Table 5-2. Five replications were performed since the experimental setup is subject to uncertainties thus subsequent replications would not result in identical results. Center points along with the star points facilitate the check for curvature and hence permits the software to fit a second order model. Normally we test at $\alpha = 5\%$ significance level which means there is 5 % chance that we are wrong.

Table 5-2 Range Considered for Primary Factors

Factors	Level(-)	Level(+)
Proportional Gain of motor (K_{p_one})	250	450
Derivative gain of motor (T_{d_one})	10	20
Proportional gain of pendulum (K_{p_two})	500	800
Derivative gain of pendulum (T_{d_two})	10	30

The result of BBD and CCD technique is discussed in the next sections. In both, a transformation was suggested by the design software (Design Expert). Assuming the higher uncertainty prediction at the vertices of a cube, a central composite design is carried out. But the results of central composite were not convincing; i.e., firstly a transformation was recommended, secondly the observed data and predicted data did not show a good fit. Hence a Uniform Design (UD) was carried out. In Uniform Design the design points are scattered uniformly in the design space yielding a better fit. Results and analysis of the three techniques are discussed in the next sections.

5.7.1 BBD results and analysis

5.7.1.1 Statistical test of significance using ANOVA-BBD

The BBD technique is carried with four factors, 2 levels and 5 center points resulting in 29 combinational runs. Table 5-3 indicates that the model is significant with “Model F-Value” of 7.00. There is only a 0.01% chance that a “Model F-Value” this large could occur due to noise. In general if the F-Value $\gg 1$ then the effect is significant which means the factors significantly affects the response. The approach taken compares the calculated F-Value to 5% F-table using Degrees of Freedom (DoF) of main effects and Degrees of Freedom of error. If the p-value calculated by the DOE software is less than 0.05, then the effect is significant. The “Lack of Fit F-Value” of 1.08 implies that the lack of fit due to “Pure Error” is not significant. Error occurs when the analysis omits one or more important terms or factors from the process model. Finally, the “Lack of Fit F-Value” indicated that the model fit is significant.

1. “A-Kp_one” in the model represents the proportional gain of the table.
2. “B-Td_one” in the model represents the derivative gain of the table.
3. “C-Kp_two” which is a significant factor in the model represents the proportional gain of the pendulum.
4. “D-Td_two” in the model represents the derivative gain of the pendulum.

Table 5-3 Analysis of variance table (BBD)

Source	Sum of Squares	df	Mean Square	F Value	p-value Prob>F
Model	2.35	6	0.39	7.00	<0.0003 Significant
A-Kp_one	3.723E-003	1	3.723E-003	0.067	0.7986
B- Td_one	0.42	1	0.42	7.47	0.0121
C- Kp_two	0.97	1	0.97	17.31	0.0004
D- Td_two	0.14	1	0.14	2.57	0.1232
B ²	0.42	1	0.42	7.43	0.0123
C ²	0.29	1	0.29	5.18	0.0329
Residual	1.23	22	0.056		
Lack of fit	1.02	18	0.057	1.08	0.5306 not significant
Pure Error	0.21	4	0.053		
Cor Total	3.57	28			

Table 5-4 Summary of statistics (BBD)

Std.Dev.	0.24	R-Squared	0.6563
Mean	4.38	Adj R-Squared	0.5626
C.V.%	5.39	Pred R-Squared	0.3852
PRESS	2.20	Adeq Precision	9.810

The quantity “R-squared” is interpreted as the proportion of the variability in the data explained by the ANOVA model. The “Pred R-Squared” of 0.3852 is in good agreement with “Adj R-Squared” of 0.5626. Adeq precision measures signal to noise ratio. A ratio greater than 4 is desirable. In this case, 9.810 indicate the adequate signal and therefore this model can be used to navigate the design space.

Model adequacy checking was performed by examining of residuals especially by graphical analysis of the residuals. Any violations of model assumptions (normality of residuals, constant variance and independence) can be investigated by the graphical analysis of residuals. In general if the model is adequate, the residuals will not follow any obvious pattern. A normality check was done by constructing a normal plot of residuals. If the error distribution is normal then plot resembles a straight line. If the model is correct and assumptions are satisfied the residual will be structureless; i.e., it will not follow any obvious pattern. In residual versus predicted graph Figure 5-5, residuals follow an obvious pattern and the Box-Cox plot recommending an inverse transformation as in Figure 5-6. Refer Appendix A for additional information.

Design-Expert® Software
Ln(Sigmaerror)

Color points by value of
Ln(Sigmaerror):
5.31812
3.68888

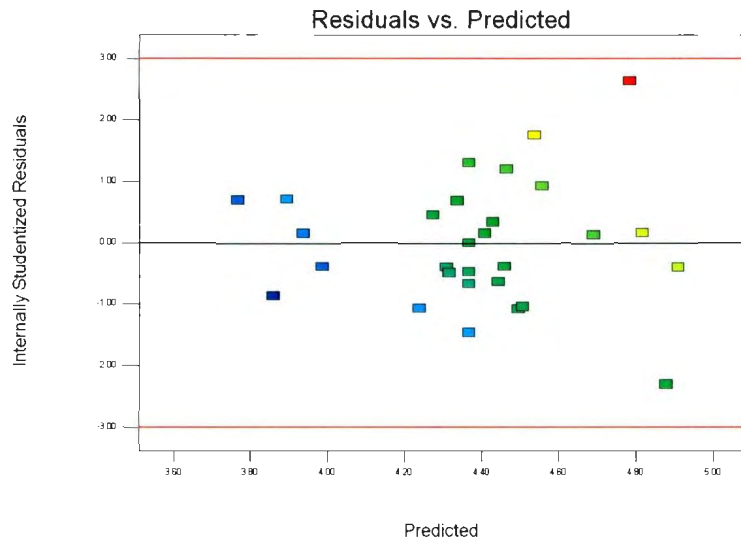


Figure 5-5 Residuals versus predicted (BBD)

Design-Expert® Software
Ln(Sigmaerror)

Lambda
Current = 0
Best = -0.93
Low C.I. = -1.78
High C.I. = -0.06

Recommend transform:
Inverse
(Lambda = -1)

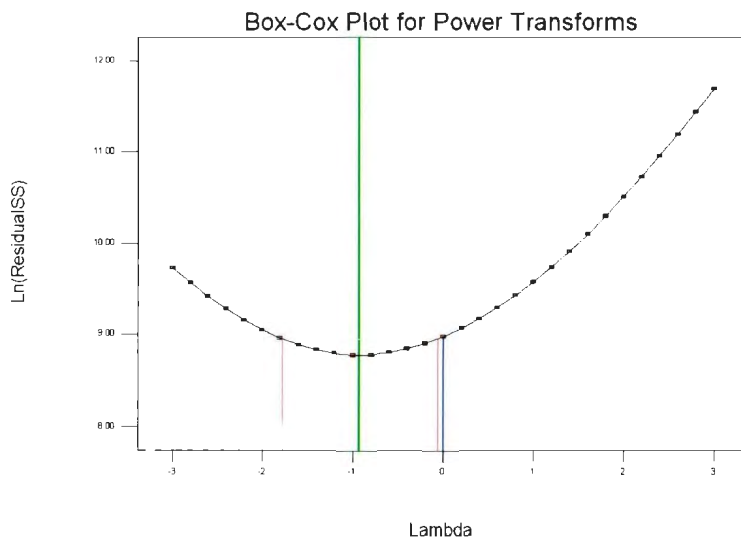


Figure 5-6 Box-Cox plot for power transforms (BBD)

5.7.1.2 Regression analysis-BBD

A quadratic model fit was obtained in terms of coded and actual factors respectively. These models are second order response models with the given factor variables.

Final Equations in Terms of Coded Factors:

$$\begin{aligned} \ln(\text{Sigmaerror}) = & 4.37 - 0.018 * A - 0.19 * B + 0.28 * C + 0.11 * D + 0.25 * \\ & B^2 - 0.20 * C^2 \end{aligned} \quad (24)$$

Final Equations in Terms of Actual Factors:

$$\begin{aligned} \ln(\text{Sigmaerror}) \\ = & 2.10654 - 3.522292E - 004 * K_{p\text{one}} - 0.11673 * T_{d\text{one}} \\ & + 0.017168 * K_{p\text{two}} + 0.010935 * T_{d\text{two}} + 2.4521E - 003 \\ & * T_{d\text{one}}^2 - 2.04707E - 005 * K_{p\text{two}}^2 \end{aligned} \quad (25)$$

5.7.2 CCD results and analysis

5.7.2.1 Statistical test of significance using ANOVA-CCD

The experiment is carried with 4 factors, 2 levels and 5 center points resulting in 29 combinational runs. From Table 5-5 the design indicates that the model is significant with “Model F-Value” of 11.06. There is only a 0.01% chance that a “Model F-Value” this large could occur due to noise. In general if the F-Value $\gg 1$ then the effect is significant which means the factors significantly affects the response. The approach taken compares

the calculated F-Value to 5% F-table using Degrees of Freedom (DoF) of main effects and Degrees of Freedom of error.

Table 5-5 Analysis of variance table (CCD)

Source	Sum of Squares	df	Mean Square	F Value	p-value Prob>F
Model	4.77	8	0.60	11.06	<0.0001 Significant
A-Kp_one	0.27	1	0.27	4.94	0.0380
B- Td_one	0.15	1	0.15	2.74	0.1133
C- Kp_two	2.69	1	2.69	49.85	<0.0001
D- Td_two	0.061	1	0.061	1.13	0.3007
CD	0.32	1	0.32	5.96	0.0240
A ²	0.60	1	0.60	11.09	0.0033
B ²	0.21	1	0.21	3.92	0.0617
C ²	0.32	1	0.32	6.01	0.0235
Residual	1.08	20	0.054		
Lack of fit	0.75	16	0.047	0.57	0.8150 not significant
Pure Error	0.33	4	0.083		
Cor Total	5.84	28			

Alternately, if the p-value which is calculated by the DOE software is less than 0.05, then the effect is significant. The “Lack of Fit F-Value” p value of 0.8150 implies that the lack of fit due is not significant. Error occurs when the analysis omits one or more important terms or factors from the process model.

Table 5-6 Summary of statistics (CCD)

Std.Dev.	0.23	R-Squared	0.8156
Mean	4.63	Adj R-Squared	0.7418
C.V.%	5.02	Pred R-Squared	0.5643
PRESS	2.55	Adeq Precision	15.050

The “Pred R-Squared” of 0.5643 is in reasonable agreement with “Adj R-Squared” of 0.7418. Adeq precision measures signal to noise ratio. A ratio greater than 4 is desirable. In this case, 15.050 indicate the adequate signal and therefore this model can be used to navigate the design space. Figures 5-7 to 5-11 show the result of analysis of residuals for model assumptions. From the Box-Cox plot it is obvious that a transformation is recommended, therefore a natural log transformation is considered.

Design-Expert® Software
Ln(Sigmaerror)

Color points by value of
Ln(Sigmaerror):

5.61677
3.68888

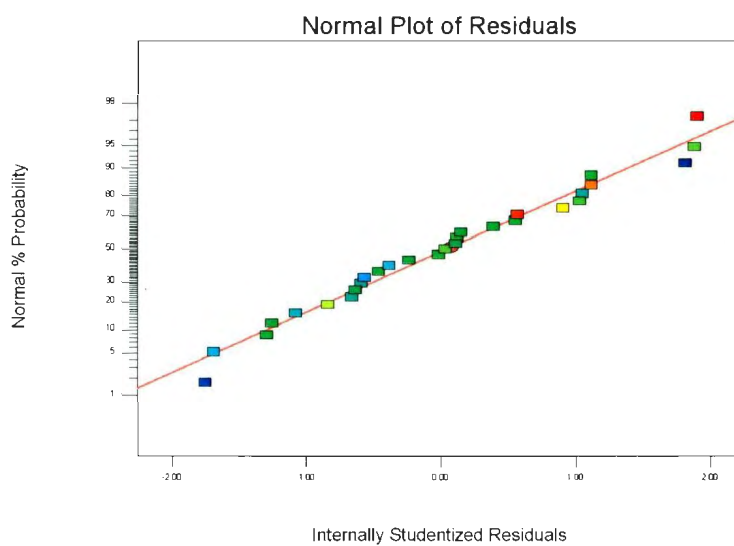


Figure 5-7 Normality probability plots of residuals (CCD)

Design-Expert® Software
Ln(Sigmaerror)

Color points by value of
Ln(Sigmaerror):

5.61677
3.68888

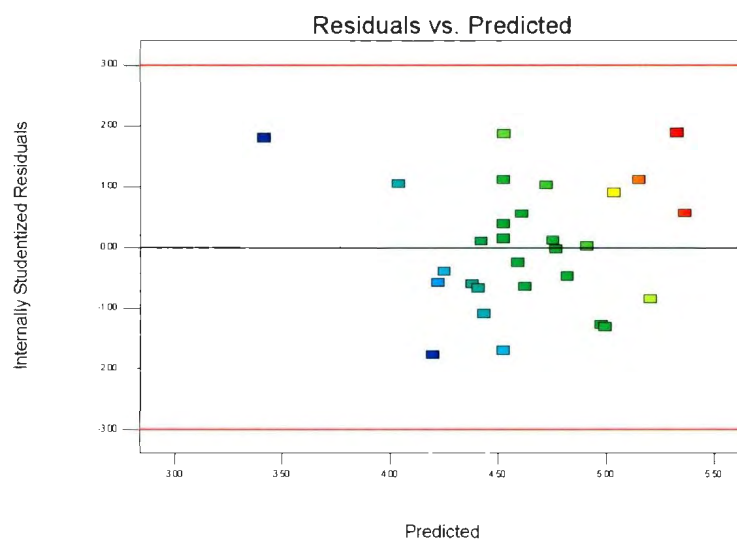


Figure 5-8 Residuals versus predicted (CCD)

Design-Expert® Software
Ln(Sigmaerror)

Color points by value of
Ln(Sigmaerror):
5.61677
3.68888

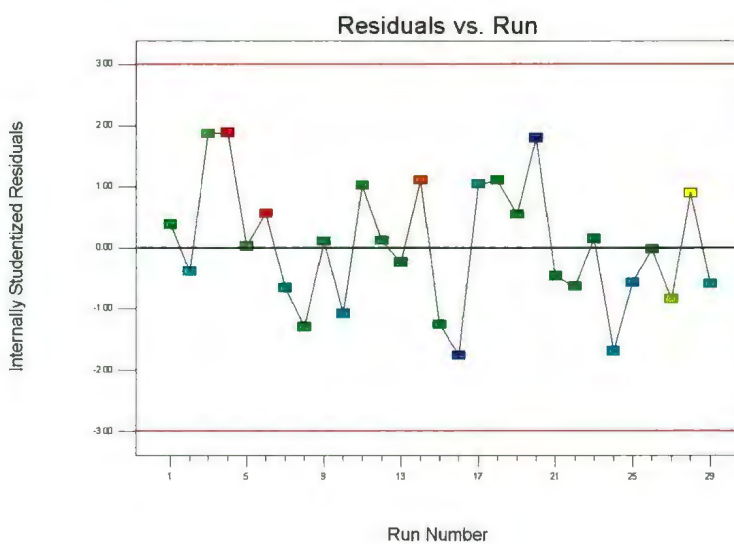


Figure 5-9 Residual versus run (CCD)

Design-Expert® Software
Ln(Sigmaerror)

Color points by value of
Ln(Sigmaerror):
5.61677
3.68888

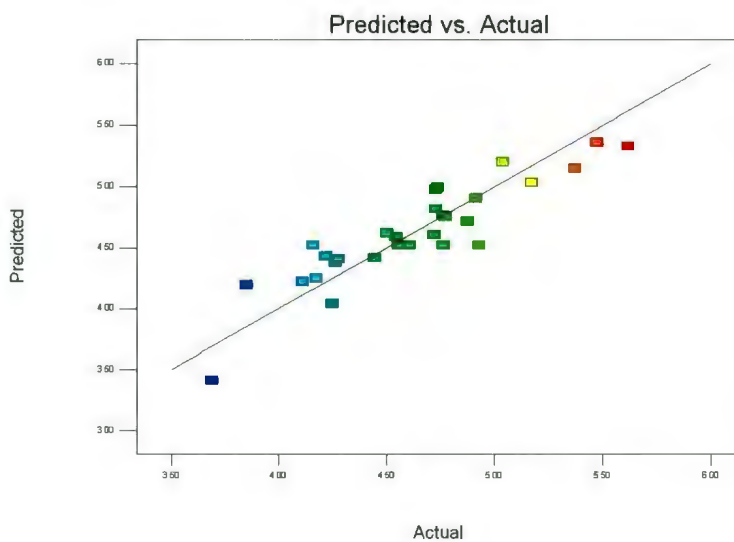


Figure 5-10 Predicted versus actual (CCD)

Design-Expert® Software
Ln(Sigmaerror)

Lambda
Current = 0
Best = -0.49
Low C.I. = -1.17
High C.I. = 0.22

Recommend transform:
Inverse Sqrt
(Lambda = -0.5)

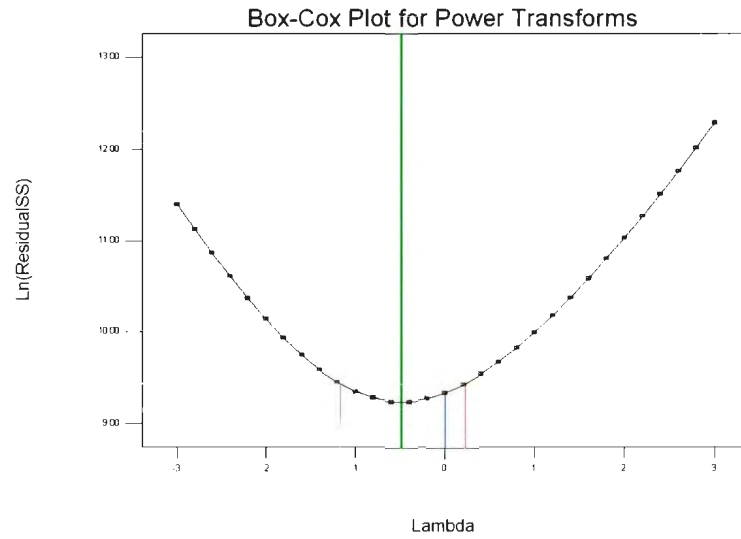


Figure 5-11 Box-Cox plot for power transforms (CCD)

5.7.2.2 Regression analysis-CCD

Regression analysis in design of experiments is quite straight forward. Regression as such is an interpolation and not extrapolation technique. Predictions from the regression model are made within the confines of data rather than rough approximation of the model. Equations shown below are the final model equations obtained from the design of experiments in terms of coded and actual factors. These models are second order response surface model.

Final Equations in Terms of Coded Factors:

$$\begin{aligned} \ln(\text{Sigmaerror}) = & 4.52 + 0.11 * A - 0.078 * B + 0.33 * C + 0.050 * D + \\ & 0.14 * CD + 0.15 * A^2 + 0.089 * B^2 - 0.11 * C^2 \end{aligned} \quad (26)$$

Final Equations in Terms of Actual Factors:

$$\begin{aligned} \ln(\text{Sigmaerror}) = & 8.14344 - 0.033661 * K_{p_one} - 0.043271 * T_{d_one} + \\ & 8.19088E - 003 * K_{p_two} - 0.044562 * T_{d_two} + 1.41704E - 004 * T_{d_two} * \\ & K_{p_two} + 5.96101E - 005 * K_{p_one}^2 + 8.85572E - 004 * T_{d_one}^2 - 1.09707E - \\ & 005 * K_{p_two}^2 \end{aligned} \quad (27)$$

Optimal Solution

The optimal solution is obtained by choosing the factors to be in range as selected and the output response targeted to minimal which is referred to as numerical optimization. For more on numerical optimization see section 5.7.4.4. From the predicted optimal solution and the gain values the experiment is carried out for the system and the responses are observed. An average of 10 trail observations is noted and the resulted is compared with the model prediction value. From the Table 5-7 it is noted that the observed value is not close to the predicted value.

Table 5-7 Predicted optimal solution and observed value

Kp_one	Td_one	Kp_two	Td_two	Predicted value	Observed value
282	24	250	30	51.983	61.5

5.7.2.3 Experimental Validation of the system Model-CCD

To validate the model generated by Design expert, using Minitab a set of 50 different combinations of gain values other than DOE combinational runs is generated. Using the

final equation obtained from the regression analysis the predicted value for each combinational gain is predicted. From the response of the system the observed value is obtained. A graph is plotted between predicted versus the observed value as shown in Figure 5-12. The slope of the regression line is 0.6054 which is not close enough to 1. See Appendix A for more information on data points. Therefore a different experimental design approach known as uniform design approach is carried out for obtaining more accurate results. In Uniform design the design points are uniformly scattered on the design domain.

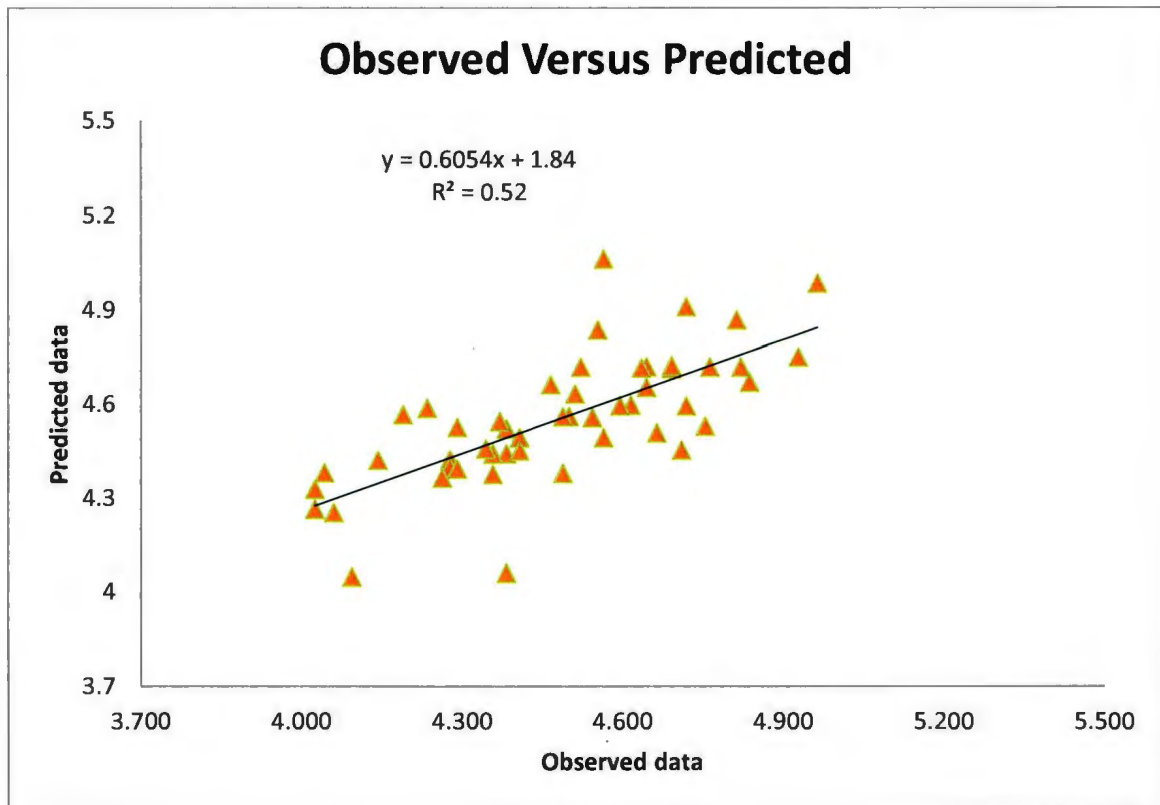


Figure 5-12 Predicted data versus observed data

5.7.3 Uniform Design Methodology-Overview

Uniform experimental designs are one kind of space filling designs that can be used for industrial experiments when the underlying model is unknown. Uniform design was in fact motivated by engineering projects. In the past two decades uniform design has been successfully applied in other fields such as pharmaceuticals and natural sciences. The uniform design seeks its design points to be uniformly scattered on the experimental domain. In many cases the experimenter does not know the underlying model. A space filling design becomes the best choice in such cases. A uniform design table has a notation $U_n(q^s)$, where 'U' stands for Uniform Design, n for the number of runs, s for the number of factors and q for the number of levels as explained by C.R.Rao (2003). Uniform designs are implemented as follows:

- Choose the factors and the experimental domain as well as determine the suitable number of levels for each factor.
- By visiting the Uniform Design (UD)-web a suitable UD table is chosen (<http://uic.edu.hk/isci/UniformDesign/UD%20Tables.html>)
- From the uniform design table randomly determine the run order of experiments and conduct the experiment.

The experiment is conducted with 4 factors, 3 levels and a response variable resulting in 33 combinational runs which included 3 replications of the center points. For each combinational runs the response of the system is noted and a computer generated model is obtained. The following section explains the resulting model turned out to be the satisfactory one.

5.7.4 Uniform Design method- results and analysis

5.7.4.1 Statistical test of significance using ANOVA-UD

From Table 5-8 the obtained model indicates that the model is significant with “Model F-Value” of 7.71. There is only a 0.01% chance that a “Model F-Value” this large could occur due to noise. In general if the F-Value $\gg 1$ then the effect is significant which means the factors significantly affects the response. The approach taken compares the calculated F-Value to 5% F-table using Degrees of Freedom (DoF) of main effects and Degrees of Freedom of error. Alternately, if the p-value which is calculated by the DOE software is less than 0.05, then the effect is significant. Therefore the main effect C considered to be the important factor is significant in the model. The “Lack of Fit F-Value” p value of 0.1549 implies that the lack of fit is not significant.

Table 5-8 Analysis of variance table (UD)

Source	Sum of Squares	df	Mean Square	F Value	p-value Prob>F
Model	7747.62	6	1291.27	7.71	<0.0001 Significant
A-Kp_one	2.86	1	2.86	0.017	0.8970
B- Td_one	54.78	1	54.78	0.33	0.5724
C- Kp_two	6182.40	1	6182.40	36.90	<0.0001 Significant
D- Td_two	72.21	1	72.21	0.43	0.5173
BC	791.47	1	791.47	4.72	0.0390
D ²	1509.20	1	1509.20	9.01	0.0059
Residual	4356.26	26	167.55		
Lack of fit	4295.60	24	178.98	5.90	0.1549not significant
Pure Error	60.67	2	30.33		
Cor Total	12103.88	32			

Table 5-9 Summary of statistics (UD)

Std.Dev.	12.94	R-Squared	0.6401
Mean	86.61	Adj R-Squared	0.5570
C.V.%	14.95	Pred R-Squared	0.4143
PRESS	7089.06	Adeq Precision	11.178

The “Pred R-Squared” of 0.4143 is in very good agreement with “Adj R-Squared” of 0.5570. Adeq precision measures signal to noise ratio. A ratio greater than 4 is desirable. In this case, 11.178 indicate the adequate signal and therefore this model can be used to navigate the design space. Figures 5-13 to 5-18 show the result of analysis of residuals for model assumptions.

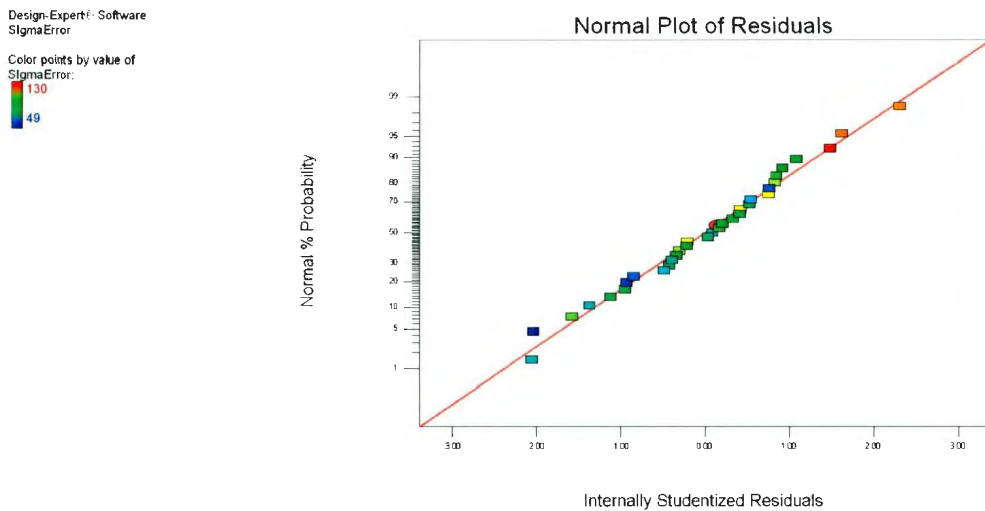


Figure 5-13 Normality probability plots of residuals (UD)

Design-Expert® Software
SigmaError

Color points by value of
SigmaError:

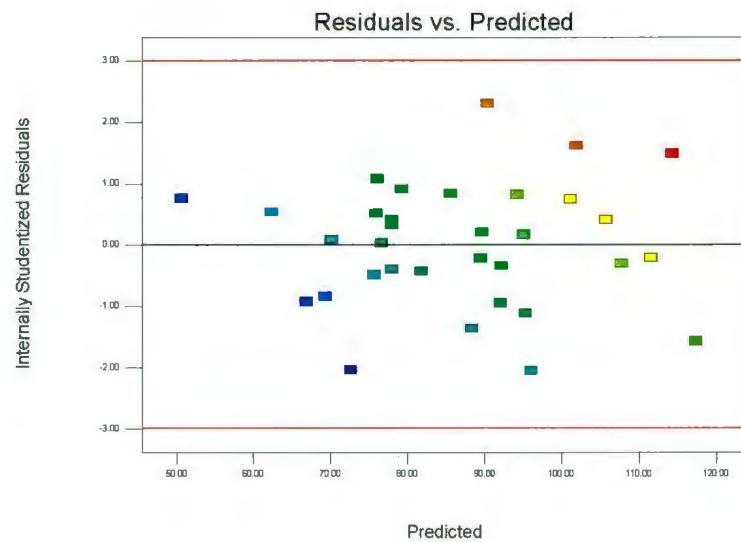


Figure 5-14 Residuals versus predicted (UD)

Design-Expert® Software
SigmaError

Color points by value of
SigmaError:

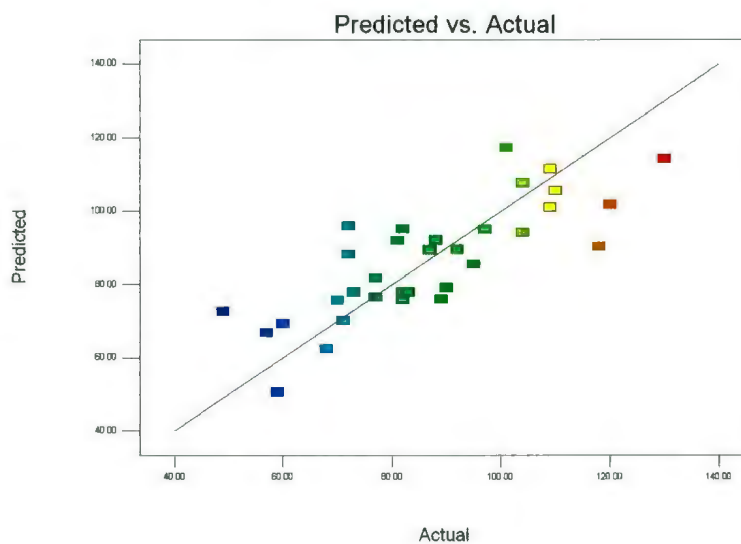


Figure 5-15 Predicted versus actual (UD)

Design-Expert® Software
SigmaError

Lambda
Current = 1
Best = 0.51
Low C.I. = -0.75
High C.I. = 1.86

Recommend transform:
None
(Lambda = 1)

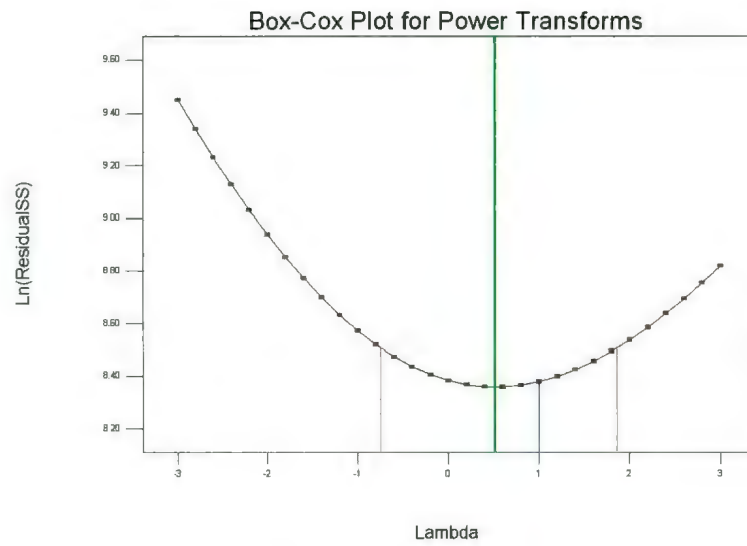


Figure 5-16 Box-Cox plot for power transforms (UD)

Design-Expert® Software
SigmaError

Color points by value of
SigmaError:
130
49

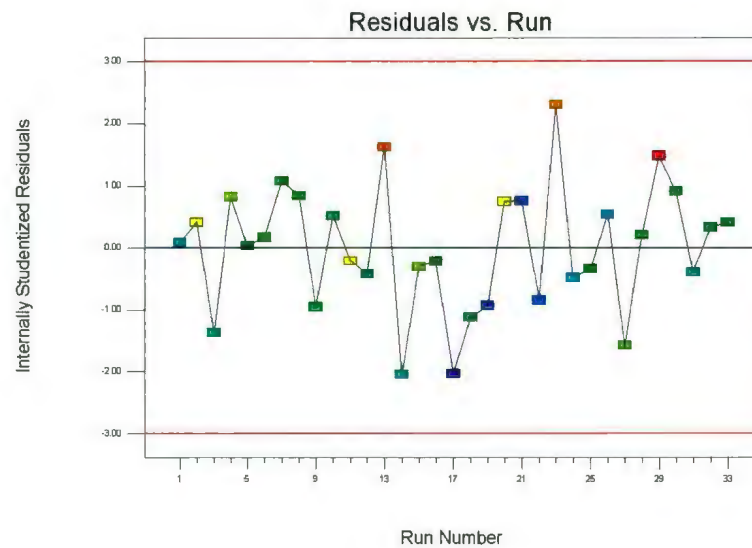


Figure 5-17 Residuals versus run (UD)

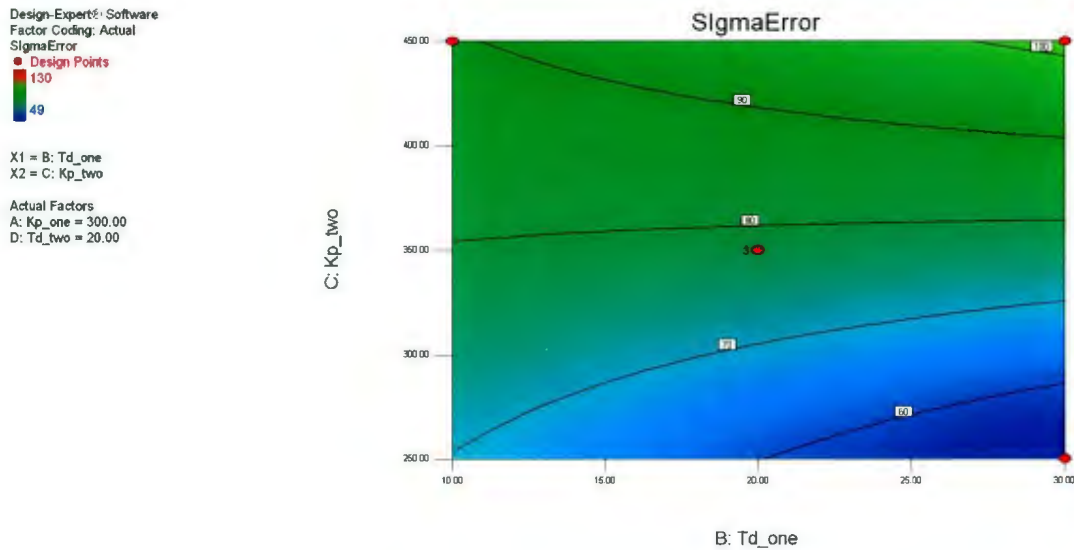


Figure 5-18 Interaction graph (UD)

5.7.4.2 Regression analysis-UD

Equations shown below are the final model equations obtained from the design of experiments in terms of coded and actual factors. These models are second order response surface model.

Final Equations in Terms of Coded Factors:

$$\text{Sigmaerror} = 77.91 + 0.33 * A - 1.66 * B + 17.72 * C - 1.90 * D + 7.86 * BC + 13.96 * D^2 \quad (28)$$

Final Equations in Terms of Actual Factors:

$$\begin{aligned} \text{Sigmaerror} = & 131.58180 + 7.6081E - 003 * K_{p\text{one}} - 2.91638 * T_{d\text{one}} + \\ & 0.020039 * K_{p\text{two}} - 5.77409 * T_{d\text{two}} + 7.85762E - 003 * T_{d\text{one}} * K_{p\text{two}} + \\ & 0.13959 * T_{d\text{two}}^2 \end{aligned} \quad (29)$$

5.7.4.3 Experimental Validation of the system Model-UD

To validate the generated model, using Minitab a 50 set of random combinational gain values different from Design expert 33 combinational runs is generated as shown in Table 5-11. A graph is plotted between predicted and observed values. Predicted values are calculated from the regression model Final Equations of Actual factors. So for each combinational gain values corresponding predicted value is determined. The experiment is carried out for the corresponding gain values whose response is also observed. Thus with predicted and observed values a graph is generated using Minitab as in Figure 5-22. A linear equation can be written is $y = mx+b$. This is called the slope-intercept form, where m is the slope of the line and b is the y-intercept. The slope is interpreted to be amount by which the y-value will increase for a one-unit of increase in the x-value. The scatter plot below shows a linear regression line superimposed on it displaying the value of R^2 and the equation of the line. Observe that the y-intercept value is 1.85 and the value of the slope is 1.015 depicting an angle of 45° .

Table 5-10 Predicted versus observed data points

Kp_one	Td_one	Kp_two	Td_two	Observed	Predicted
276	24	320	18	68	72
259	30	407	17	100	92
304	23	336	11	75	88
328	11	367	25	84	84
271	13	260	13	63	75
259	22	323	20	84	72
317	14	328	30	74	88
257	28	404	13	108	97
281	22	348	11	84	90
343	28	307	12	83	78
282	11	295	23	78	74
348	15	263	13	59	75
328	24	325	22	89	72
313	19	390	12	100	97
338	21	437	30	100	106
318	27	412	27	91	98
289	19	251	21	49	61
273	16	342	19	82	78
271	26	414	19	99	92
297	11	426	28	83	94
342	19	331	10	95	91
292	13	325	21	87	76
339	26	397	12	103	99
347	13	259	29	53	77
265	16	370	16	72	84
274	11	402	13	111	93
283	12	300	13	71	82
281	11	394	16	87	87
325	15	257	12	89	77
325	18	306	28	87	78
299	26	351	24	78	79
273	14	412	18	94	88
284	28	340	18	71	75
263	20	347	29	67	87
321	20	338	23	73	77
283	11	285	23	61	73
338	21	275	15	75	68
347	12	385	13	84	92
334	18	355	28	97	87
258	22	424	20	120	92
294	28	259	28	47	63
263	22	317	23	75	72
284	22	410	13	96	97
325	26	262	15	73	62
324	24	290	16	57	68
283	29	348	20	77	76
250	25	343	25	85	78
288	14	357	30	84	92
326	16	450	11	91	106
327	13	348	22	89	79

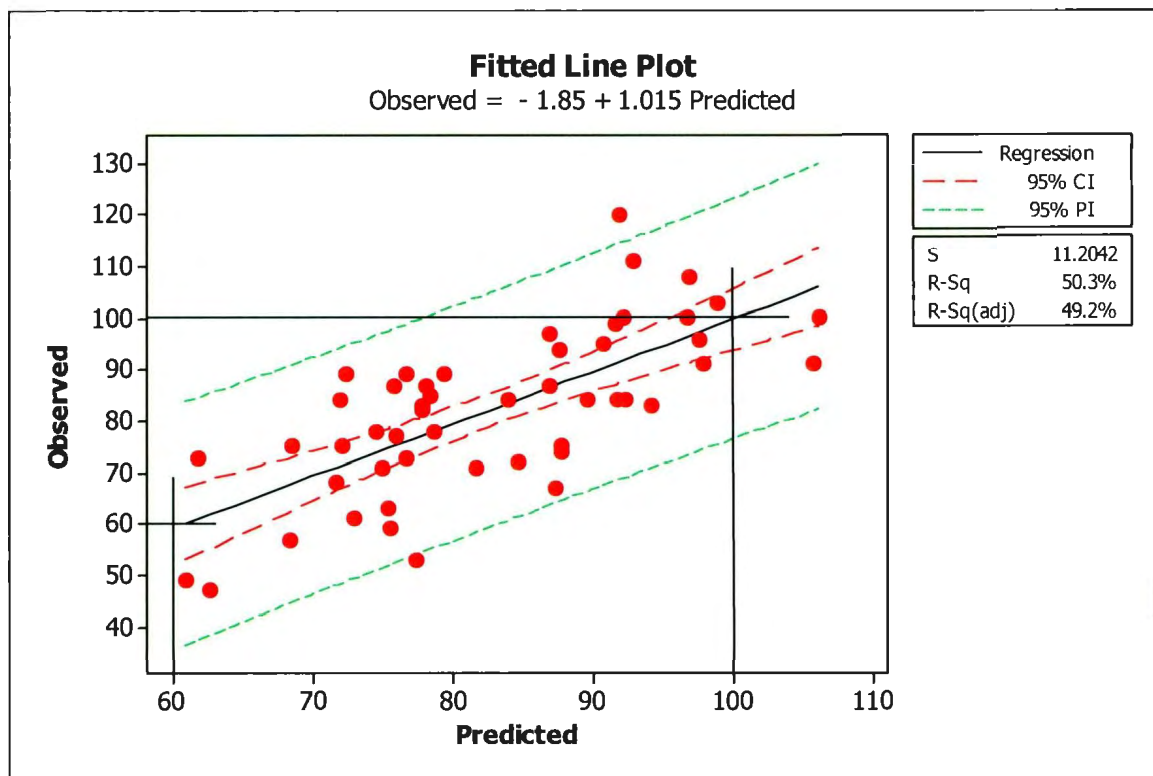


Figure 5-19 Experimental validation of optimal gains

5.7.4.4 Optimization technique

Optimization refers to an approach that optimizes design layout within a given design space, for a given set of factors and boundary conditions such that the resulting layout meets a prescribed set of performance targets. In numerical optimization there exist three “Optimization Parameters” that define each desirability index (d_i). Subject matter knowledge is incorporated in search of the optimum outcome.

(http://www.statease.com/webinars/multiple_response_optimization.pdf)

Desirability function

To determine a best combination of n responses, we use an objective function, D :

$$D = (d_1 * d_2 * \dots * d_n)^{1/n} = (\prod_{i=1}^n d_i)^{1/n}$$

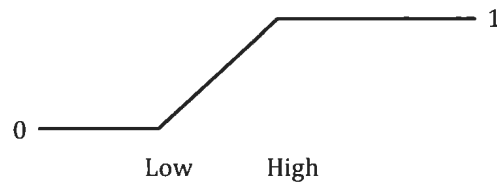
Where d_i reflects the desirability of each response and they range from 0 to 1. A desirability between 0 to 1 then the responses are within the acceptable limits with at least one of the response is not perfect. With desirability of 0 indicates that one or more responses fall outside acceptable limits.

Goals and limits:

1. "None" the response is ignored during optimization.
2. "Maximize" $d_i=0$ $Y < \text{low value}$

$0 \leq d_i \leq 1$ Y varies from low to high

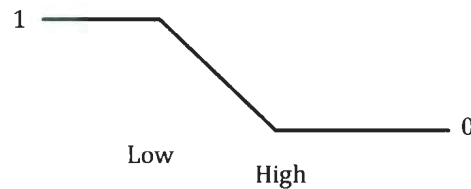
$d_i=1$ $Y > \text{high value}$



3. "Minimum" $d_i=1$ $Y < \text{low value}$

$1 \geq d_i \geq 0$ Y varies from low to high

$d_i=0$ $Y > \text{high value}$

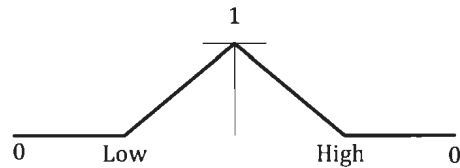


4. "Target" $d_i=0$ if $Y < \text{low value}$

$0 \leq d_i \leq 1$ Y varies from low to target

$1 \geq d_i \geq 0$ Y varies from target to high

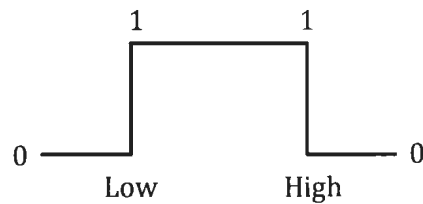
$d_i = 0$ Y > high value



5. "In range" $d_i = 0$ if Y < low value

$d_i = 1$ if Y varies from low to high

$d_i = 0$ if Y > high value



Weights:

For simultaneous optimization there exists an additional parameter called Weights.

Weights give an added emphasis to upper and lower bounds and to target value

(http://www.statease.com/webinars/multiple_response_optimization.pdf)

- Weight value of 1 the desirability vary from 0 to 1 in linear fashion
- Weights greater than 1 give more emphasis to the goal.
- Weights less than 1 give less emphasis to the goal.

In general default setting for factors is “in range” the criteria “none” cannot be applied to factors. There exist additional criteria known as “is equal to” applies only to factors not to responses. If “equal to” is set for the factors, then factors are set to a constant and this reduces the searchable space by one dimension. Numerical optimization is a hill climbing technique. More than one hill may exist therefore in such cases finding multiple optimums can be done by several optimization using different starting points.

In our model the four factors are optimized by choosing “in range” criteria and “minimize” criteria for the response as shown in Figures 5-19-5-20. The four factors chosen with “in range” criteria assign the values from low factor range to high factor range respectively. From Figure 5-19 it is clearly shown how the goal is set to “in range” shown for four factors. The main objective of numerical optimization is to minimize the response of the system which is sigma error. Hence a minimal range is chosen for the response resulting with a desirability of 0.888.

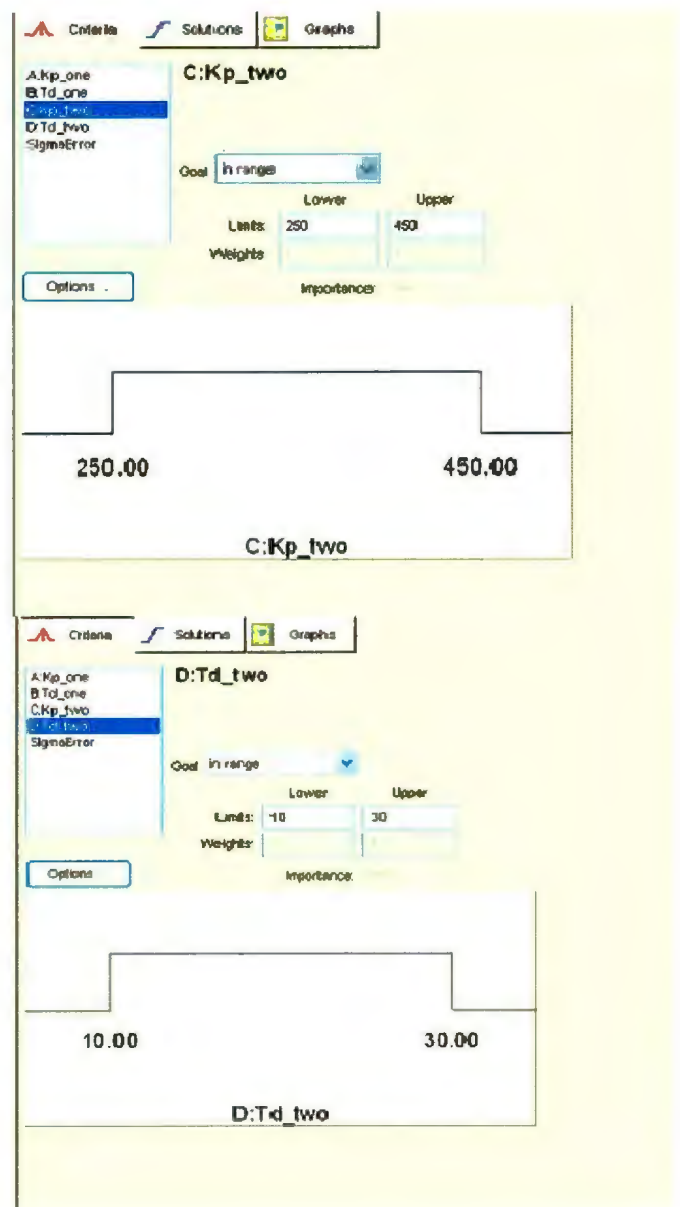
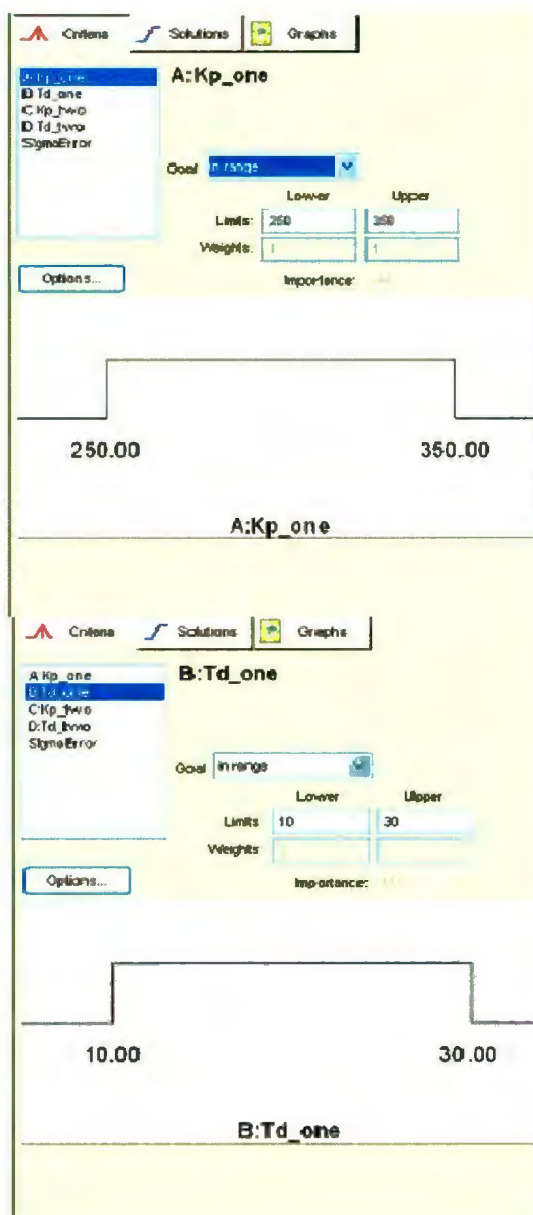


Figure 5-20 Numerical optimizations of factors

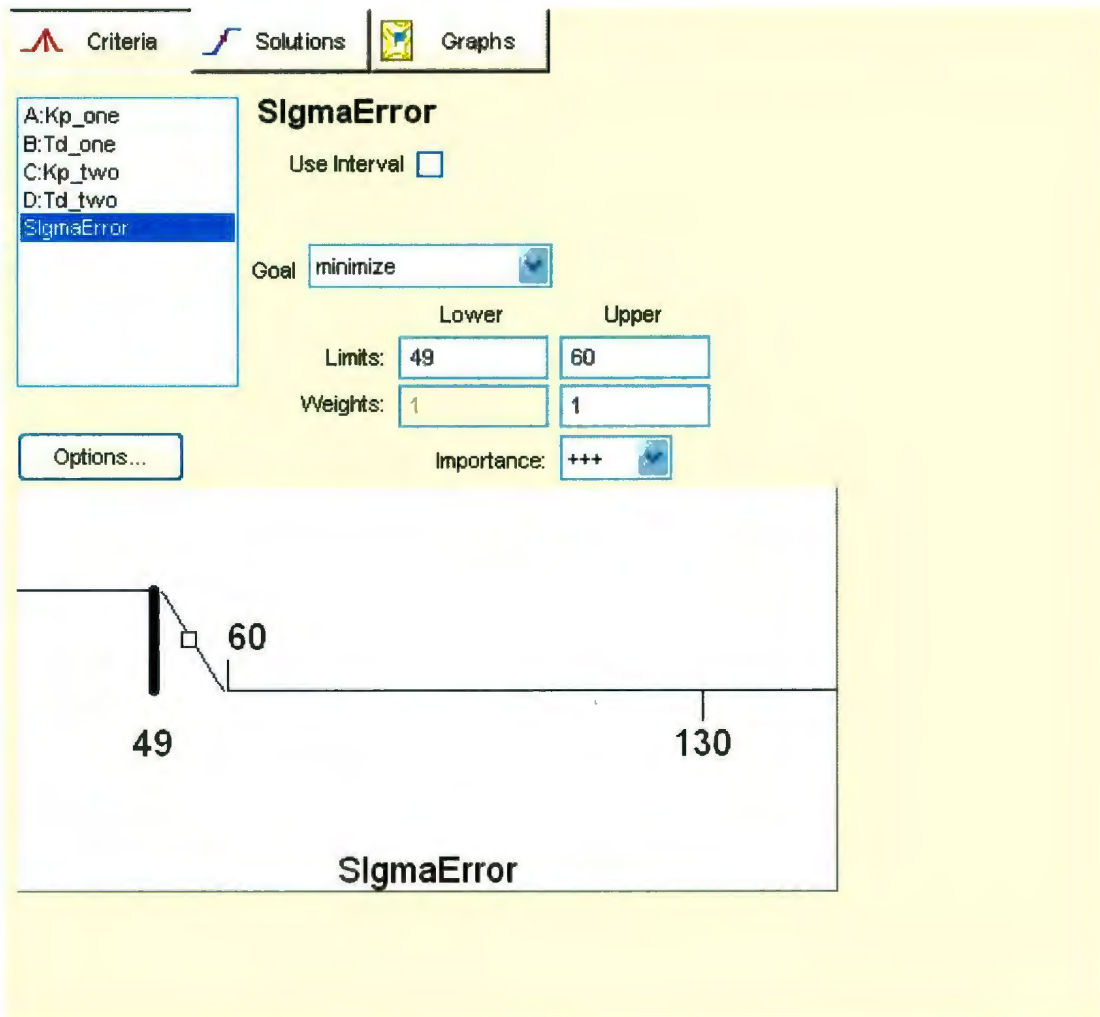


Figure 5-21 Numerical optimization of response

Optimal Solution

Thus the optimal solution for the model is obtained using numerical optimization technique with the factors “in range” and the output response targeted to “minimal” goal. From the predicted optimal solution the experiment is carried out with the given gain values and the response is observed. An average of 10 runs observation is noted and the resulted is compared with the model predicted value. From the Table 5-10 it is noted that

the observed value is closer to the predicted value. In Figure 5-21 a model predicted confirmation report explains the response “in range” lower and higher value prediction varying from 32.8 to 67.6 rounded off value. Thus the obtained result from the 10 observations is 52.8 which falls under the predicted range and closer to the model predicted value of 50.23.

Table 5-11 Predicted optimal solution and observed value

Kp_one	Td_one	Kp_two	Td_two	Predicted value	Average of observed values
250	30	250	20	50.23	52.8

Confirmation Report						
Two-sided		Confidence = 95%		n = 10		
Factor	Name	Level	Low Level	High Level	Std. Dev.	Coding
A	Kp_one	250.64	250.00	350.00	0.000	Actual
B	Td_one	30.00	10.00	30.00	0.000	Actual
C	Kp_two	250.00	250.00	450.00	0.000	Actual
D	Td_two	20.67	10.00	30.00	0.000	Actual
Response	Prediction	Std Dev	SE (n=10)	95% PI low	95% PI high	
SigmaError	50.2286	12.9441	8.4751	32.8078	67.6494	

Figure 5-22 Predicted confirmation report

Chapter 6

Conclusions and recommendations

In modern control systems, high-speed and high-density FPGA's provides a high performance design solution in which an embedded processor IP and application IP are developed, downloaded into the FPGA resulting in a complete System on a Programmable Chip (SOPC). Designs are partitioned into hardware and software development as reconfigurable hardware and reprogrammable software on the same FPGA chip. Common control approaches such as PID control require a good understanding of the system and accurate tuning in order to obtain the desired performance. In this thesis a PD control approach was adopted to stabilize a custom made inverted pendulum. Firstly a PID controller was implemented for a single variable system; i.e., tuning the table. Secondly a modern, novel based technique based on Design of Experiments (DOE) was carried out for obtaining the optimal gains for the multi variable system. In designing the inverted pendulum the controller architecture was proposed and implemented in an FPGA. There are two inputs to the system; the position of the table and the pendulum angle. The command signal from the controller was a PWM signal that was used to control the motor. In conventional FPGA applications the designs are implemented exclusively in hardware development. With recent advancement in VLSI technology, designs can now also be implemented in software on a synthesized processor to perform various tasks. In this project a highly integrated design approach

was achieved in which the system IO modules were partitioned and implemented in hardware and the control logic was implemented in a soft processor.

6.1 Summary of results and conclusions

The following explains the research work undertaken and the conclusions drawn. The first step in the research work reported here was an FPGA based controller architecture for the proposed system. It was comprised of the following modules: i) IO Module and ii) Processor Module. Secondly with the Design of Experiments an optimal solution was determined and the validation of the obtained model was achieved by comparing the observed versus predicted data.

IO Modules:

The IO Modules constitutes a PLL, Encoder and PWM module. The PLL provides a 50 MHz clock signal that serves as an input clock for the system. The primary feedback signals for the control system are angular position both for controlling the position of the table by means of a DC servomotor as well as the angle of the pendulum. The encoder blocks are used to measure angular position of the encoder blocks. The encoder blocks accept two pulse trains which are used to determine the direction of rotation depending on which signal leads the other. Using quadrature decoding a resolution of four times the fundamental number of pulses per revolution was obtained for the two encoders. The encoder block was implemented using a four flip-flops edge detector in combination with a 20-bit up-down pulse counter that was designed using the AHDL hardware descriptor language.

The PWM block generates a PWM signal to drive the motor. It has a resolution of 12-bits (0 to 4095) that controls the duty cycle. The fundamental frequency of the PWM signal is 12 KHz. The PWM block is implemented as a 12-bit up counter block and a 12 bit comparator block that compares the counter output with the desired duty cycle and generates the ON – OFF PWM output signal.

Processor Module:

In order to implement the control algorithm in software, a Nios II/s standard processor was chosen for the SOPC system. The Nios II/s processor core can address up to 2 GB of external memory. It includes a JTAG debugging module and supports custom instructions. To facilitate real-time programming a simple periodic interrupt was implemented with an interrupt request every 1 millisecond. The processor module has two 20-bit encoder inputs from the encoder blocks as well as a PLL clock input. The outputs from the processor module include a direction bit to control the direction of the cart pole table and 12-bit encoder output. Therefore the output signals from the processor module are connected to the comparator input which compares the two inputs and produce the PWM output.

Conclusions:

- The first step in the research was to design a PID controller for a Single Input Single Output (SISO) system. A ball screw was used for driving the table that was driven by a DC servo motor as a test system. The controller gains for this system

were initially tuned using the Ziegler Nichols method. This method is generally applicable for any closed loop feedback SISO system. The Ziegler Nichols method is a systematic, semi-empirical method for tuning controller parameters that attempts to achieve optimum dynamic performance. The proportional gain K_p , is gradually increased until the system reaches marginal stability. The proportional, integral, and derivative gains, K_p , K_{cr} and P_{cr} , are then derived from the period of oscillation and the “ultimate” gain; i.e., the proportional gain that yields marginal stability. The Ziegler Nichols method produces a stable dynamic response. However, it does not guarantee a set of optimal gains. This method is also not directly applicable for a multivariable system.

- Using a trial and error method, an attempt was made to tune a Multiple Input, Single Output (MISO) PID controller. This exercise was not successful and lead to an unstable system as the responsiveness of the controller to an error resulted in oscillations. The integral term in the PID controller increased the order of the system resulting in system overshoot. The increase in accumulation of the error resulted in integral windup and the system was unstable. Thus the integral action was dropped and the control logic for a cascaded PD controller algorithm for the system was implemented using the eclipse IDE tool.
- The proposed methodology was demonstrated for an inverted pendulum system consisting of a servomotor driven ball screw translation stage and an optical encoder/pendulum assembly. This is a typical multivariable control problem with multiple gain values. A cascaded PD type controller was used to control the

system. It consisted of an outer loop with the feedback from the pendulum and an inner loop with the feedback from the table position. The vertical position was considered as the set point for the pendulum. The set point of the table position was the output of a PD type controller from the outer loop; i.e., the pendulum. In other words, the control signal to the motor was calculated as the output of the second PD type controller. Initially the table was balanced using a trial and error method. The proposed method was subsequently used to find the optimum proportional and derivative gains for the pendulum and table. First a suitable range was considered for the proportional and derivative gains. The performance criterion for the optimization was based on the integral square error of the pendulum position over a fixed time interval. DOE tests were carried out for each of the trial values. The DOE analysis then provided the controller gains that would produce the best performance.

- A set of optimal gains provides a stable dynamic performance for a system. A novel approach for obtaining optimal gains has been proposed in this study based on Design of Experiments. In this method controller gains are considered as primary factors and an applicable range is considered for each factor. A response surface methodology (RSM), namely central composite design (CCD) and Box-Behnken Designs (BBD) was carried out. This generates a series of controller gains based on the number of factors. A cost function associated with each of the gains is considered. The results obtained using BBD and CCD techniques were not satisfactory enough. From the BBD technique the obtained model

recommended a transformation and the residual versus predicted graph followed an obvious pattern resulting in violation of model assumptions and model inadequacies. Therefore with system assumption of design points on the vertices of the cube a Central Composite Design technique was carried out. CCD model was significant with minimal lack of fit and with a natural log transformation recommended in the model. A comparison was made between the model predicted optimal solutions to that of observed value of 10 trial runs resulting in inappropriate value. An experimental validation of the CCD model is done by comparing the observed versus predicted values of 50 data points. A graph is generated with slope value of 0.6 which is not as close to 1. Thus a different technique was approached known as Uniform Design.

- A uniform design experiment was carried out resulted in a significant model with minimal lack of fit. If a lack of fit is found then obtained model would no more be a best model. The residual plots obtained from the model did not violate the model assumptions and model adequacy. From the predicted optimal solution a comparison was made with a 10 trial runs. The observed value was closer to the predicted value from the model. Next, experimental validation was carried out using Minitab software.
- A final step in the thesis was experimental validation. A graph was generated to study the observed versus predicted data. Using Minitab a random 50 set of data points were generated and the experiment was carried out. From the regression final equations the predicted responses for the 50 data points were calculated.

Observed data was calculated by carrying out the actual experiment for the 50 set of gain values which are fairly different from 33 set of actual data points. The generated predicted data versus observed data graph resulted in 45° angle with a slope equals 1.

6.2 Contributions

Summary of the main contributions of the work are:

- A digital controller was implemented in an FPGA using virtual soft processors in which the control algorithm was implemented. Hardware based quadrature decoders and PWM blocks were created using a hardware description language. The implemented controllers include a simple PD controller as well as cascaded controller architecture.
- A novel method of tuning the controller gains based on Design of Experiments was developed. This method is applicable for simple and multivariable systems. It employed an initial set of input gains obtained by using the Ziegler Nichols Method for SISO and by the trial and error method for multi variable systems. Then a specific acceptable range was considered for the gain values. The DOE analysis generated a combination of optimal gains based on uniform design experiment. Initial different DOE technique called RSM technique was implemented for obtaining optimal solutions. But the results were not appropriate hence a Uniform design was used. By DOE analysis it is determined which gains or the factors were significant for control methodology.

6.3 Limitations and recommendations

Limitations:

The following are the limitations of the proposed method:

The output of the proposed method depends on the initial set of gains and the acceptable range of gains. Hence, it does not guarantee a global optimum for the gain values. This method will only provide a local optimum for the range selected. The Ziegler Nichols method could be used for finding the initial gains for the SISO systems. However, there is no clear method in obtaining the initial set of values for the multi variable systems. In some cases it may be a difficult to obtain the initial gains for such systems.

Recommendations:

- The proposed control algorithm was experimentally tested for implementing a cascaded PD type controller for a multi input single output system. This should be tested for other controller architectures with multiple inputs and outputs. In addition to obtaining the optimum gains, this method could be extended for comparing and selecting the best among different controller architectures.
- The initial gain values for a simple system were obtained using the Ziegler Nichols method. However, there is no clear method of obtaining these for a multivariable system. A reliable method for obtaining initial gain values for a multi variable system should be developed.
- The current study considered only one variable (Integral Square of Error) as the performance index. Other factors such as system rise time, setting time, percentage overshoot, minimum error of a combination of feedback parameters

could be considered in the performance index. By considering several factors superior dynamic performance could be achieved.

Bibliography

- Akhtaruzzaman, M., & Shafie, A. A. (2010). Modeling and Control of a Rotary Inverted Pendulum Using Various Methods, Comparative Assessment and Result Analysis. *International Conference on Mechatronics and Automation (ICMA)*, (pp. 1342-1347).
- C.R.Rao, R. (2003). *Handbook of Statistics Volume 22*. Elsevier Science B.V.
- Chakravarthy, N., & Xiao, J. (2006). FPGA-based Control System for Miniature Robots. *IEEE International Conference on Intelligent Robots and Systems*, (pp. 3399-3404).
- Chan, Y. F., Moallem, M., & Wang, W. (2007). Design and Implementaion of Modular FPGA-based PID controllers. *IEEE tansactions on Industrial Electronics*, (pp. 1898-1906).
- Chan, Y. F., Moallem, M., & Wang, W. (2004). Efficient Implementation of PID Control Algorithm using FPGA Technology. *IEEE Conference on Decision and Control*, (pp. 14-17).
- Demirtas, M., & Karaoglan, A. D. (2012). Optimixation of PI parameters for DSbased permanent magnet brushless motor drive using response surface methodology. *Energy Conversion and Management* , 104-111.
- Floyd, T. I. (2009). *Digital fundamental* (tenth ed.). Prentice-Hall, Inc.
- Kim, J. S., Jeon, H. W., & Juang, S. (2007). Hardware implementation of Nonloinear PID Controller with FPGA based on Floating Point operation for 6-DOF Manipulator Robot Arm. *Control International Conference on Automation and Systems (ICCAS '07)*, (pp. 1066-1071).
- Krouglicof, N. (2004). Using Programmable Logic to Introduce Digital Design Methodologies in a Mechatronics Design Course. *International Conference on Mechatronics and Robotics*, (pp. 1267-1272).
- Kung, Y. S., & Shu, G. S. (2005). Development of a FPGA-based motion control IC for Robot Arm. *IEEE International Conference Industrial Technology*, (pp. 1397-1402).
- Kung, Y. S., Wang, M. S., & Chuang, T. Y. (2009). FPGA-based Self Tuning PID Controller using RBF Neural Network and its Appliaction in X-Y Table. *IEEE International Symposium on Industrial Electronics*, (pp. 694-699).
- Magna, M. E., & Holzapfel, F. (1998). Fuzzy-Logic control of an Inverted Pendulum with Vision Feedback. *IEEE Transactions on Education*, (pp. 165-170).

Montgomery, D. C. (2005). *Design and Analysis of Experiments* (sixth ed.). John Wiley & Sons, Inc.

Ogata, K. (2002). *Modern Control engineering* (fourth ed.). Prentice-Hal, Incl.

Santhakumar, M., & Asokan, T. (2010). A Self-Tuning Proportional-Integral-Derivative Controller for an Autonomous underwater Vehicle, Based on Taguchi Method. *The Journal of Computer Science* , 862-871.

Siddique, M. S., Sajid, A. H., & Chougule, D. G. (2009). FPGA based Efficient Implementation of PID Control Algorithm. *Control international Conference on Automation, communication and Energy Conservation*, (pp. 1-5).

Sreenivasappa, B. V., & Udaykumar, R. Y. (2009). Design and Implementation of FPGA Based Low Power Digital PID Controllers. *Fourth International Conference on Industrial and Information systems*, (pp. 568-573).

Tu, Y. W., & Ho, M. T. (2010). Design and Implementation of DSP and FPGA-Based Robust Visual Servoing Control of an Inverted Pendulum. *IEEE International Conference on Control Applications (CCA)*, (pp. 83-88).

Zhao, W., Kim, B. H., Larson, A. C., & Voyles, R. M. (2005). FPGA Implementation of Closed-Loop Control System for Small-Scale Robot. *International Conference on Advanced Robotics*, (pp. 70-77).

Ziegler, J. G., & Nichols, N. B. (1942). Optimum Settings for Automatic controllers. *Transactions of the ASME*, (pp. 759-765).

Appendix

Response Surface Methodology

Response Surface Methodology allows us to estimate the interactions and the quadratic effects therefore give us an idea of the (local) shape of the response surface. For this reason, they are termed RSM designs. RSM designs are used to

- Find improved or optimal process settings
- Troubleshoot process problems and weak points
- Make the process more robust against external and non-controllable influences.

Two very useful popular experimental designs that allow second order model to fit are:

- **Box-Behnken Design**
- **Central Composite Design**

Box-Behnken design:

The Box-Behnken design is rotatable (or nearly so) but it contains regions of poor prediction quality. Missing out the corners may be useful when the experimenter should avoid combined factor extremes. This property prevents a potential loss of data in those cases by Montgomery (2005).

Central Composite Design:

CCD is a commonly used method to fit second order response surface models. There exist three types of central composite designs:

- Central Composite Circumscribed (CCC)

- Central composite Inscribed (CCI)
- Central Composite Face-Centered (CCF)

Table A-1 explains the number of runs required for the given factors for carrying out the above two designs. BBD designs are carried with factors above 3 and for few cases BBD requires fewer runs compare to CCD.

Table A-1

Number of Factors	Central Composite	Box-Behnken
2	13 (5 center points)	-
3	20 (6 centerpoint runs)	15
4	30 (6 centerpoint runs)	27
5	33 (fractional factorial) or 52 (full factorial)	46
6	54 (fractional factorial) or 91 (full factorial)	54

BBD Results:

Design-Expert® Software
Ln(Sigmaerror)

Color points by value of
Ln(Sigmaerror):

5.31812
3.68888

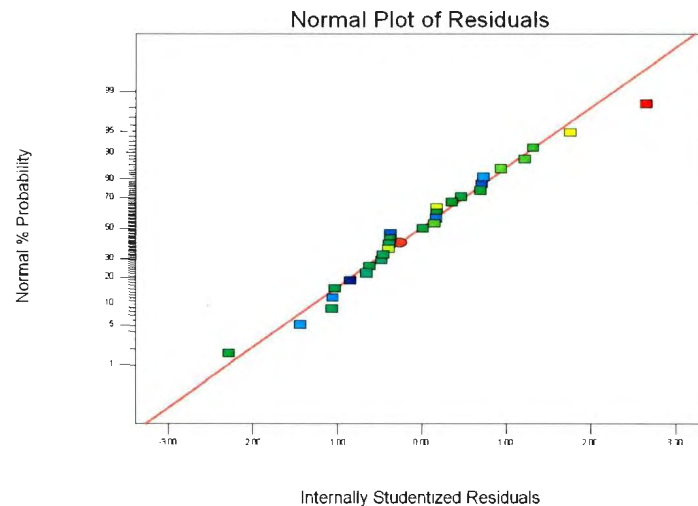


Figure A-1 Normality plot of residuals

Design-Expert® Software
Ln(Sigmaerror)

Color points by value of
Ln(Sigmaerror):

5.31812
3.68888

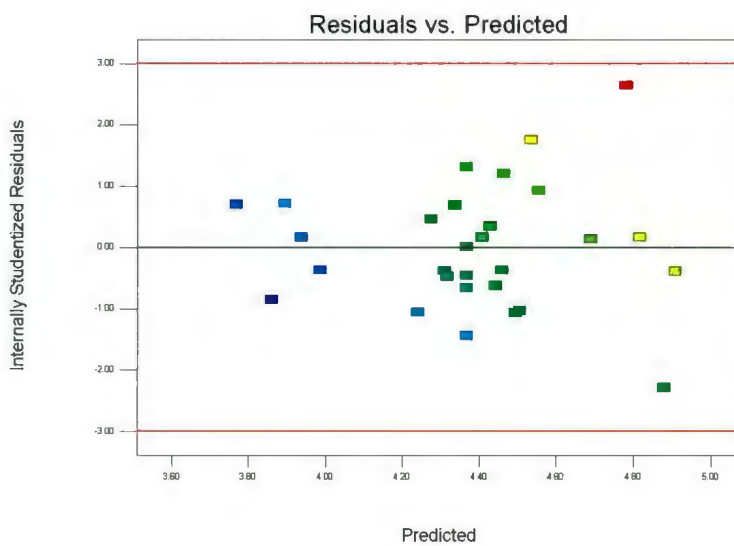


Figure A-2 Residuals versus predicted

Design-Expert® Software
Ln(Sigmaerror)

Color points by value of
Ln(Sigmaerror):

5.31812
3.68888

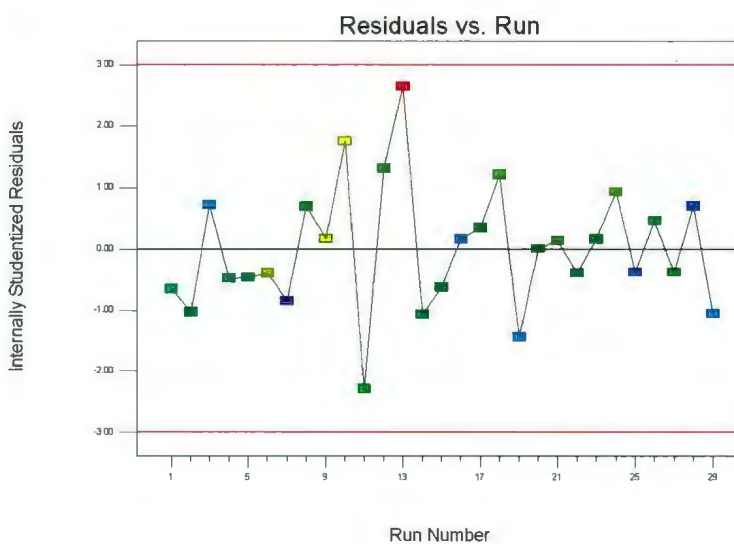


Figure A-3 Residual versus run

Optimal solution

Criteria			Solutions			Graphs											
Solutions	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Constraints																	

Figure A-4 Obtained optimal solution

Experimental Validation for the obtained CCD model:

Minitab software is used to generate a random 50 set of data points and the experiment is carried out whose response is observed. From the obtained Regression equations predicted value is determined and from the experimental result observed data is noted.

Table A-2 Predicted versus observed data points

Kp_one	Td_one	Kp_two	Td_two	Sigmaerror	Observed value	Predicted
276	24	320	18	89	5.770	4.379
259	30	407	17	87	6.008	4.662
304	23	336	11	80	5.818	4.442
328	11	367	25	123	5.906	4.867
271	13	260	13	56	5.559	4.266
259	22	323	20	72	5.777	4.423
317	14	328	30	69	5.794	4.587
257	28	404	13	101	6.002	4.596
281	22	348	11	78	5.851	4.441
343	28	307	12	66	5.727	4.567
282	11	295	23	63	5.688	4.421
348	15	263	13	82	5.571	4.493
328	24	325	22	80	5.784	4.521
313	19	390	12	99	5.966	4.594
338	21	437	30	96	6.081	5.058
318	27	412	27	95	6.022	4.836
289	19	251	21	80	5.524	4.061
273	16	342	19	73	5.835	4.526
271	26	414	19	104	6.027	4.653
297	11	426	28	143	6.055	4.982
342	19	331	10	91	5.801	4.633
292	13	325	21	106	5.782	4.508
339	26	397	12	92	5.984	4.718
347	13	259	29	57	5.556	4.382
265	16	370	16	112	5.914	4.595
274	11	402	13	109	5.997	4.713

283	12	300	13	111	5.705	4.453
281	11	394	16	104	5.976	4.719
325	15	257	12	56	5.550	4.330
325	18	306	28	82	5.723	4.451
299	26	351	24	116	5.862	4.529
273	14	412	18	109	6.020	4.722
284	28	340	18	77	5.828	4.457
263	20	347	29	90	5.850	4.561
321	20	338	23	89	5.822	4.560
283	11	285	23	78	5.651	4.376
338	21	275	15	72	5.618	4.398
347	12	385	13	112	5.952	4.909
334	18	355	28	138	5.871	4.749
258	22	424	20	103	6.051	4.715
294	28	259	28	60	5.558	4.050
263	22	317	23	73	5.759	4.393
284	22	410	13	94	6.015	4.556
325	26	262	15	58	5.568	4.255
324	24	290	16	71	5.670	4.365
283	29	348	20	96	5.851	4.493
250	25	343	25	79	5.837	4.545
288	14	357	30	126	5.879	4.669
326	16	450	11	117	6.108	4.718
327	13	348	22	124	5.852	4.717



