View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by University of Kentucky

University of Kentucky
UKnowledge

University of Kentucky Master's Theses

Graduate School

2006

ALL DIGITAL DESIGN AND IMPLEMENTAION OF PROPORTIONAL-INTEGRAL-DERIVATIVE (PID) CONTROLLER

Hui Hui Chin
University of Kentucky, hhchin1@uky.edu

Right click to open a feedback form in a new tab to let us know how this document benefits you.

Recommended Citation

Chin, Hui Hui, "ALL DIGITAL DESIGN AND IMPLEMENTAION OF PROPORTIONAL-INTEGRAL-DERIVATIVE (PID) CONTROLLER" (2006). *University of Kentucky Master's Theses*. 272.
https://uknowledge.uky.edu/gradschool_theses/272

This Thesis is brought to you for free and open access by the Graduate School at UKnowledge. It has been accepted for inclusion in University of Kentucky Master's Theses by an authorized administrator of UKnowledge. For more information, please contact UKnowledge@lsv.uky.edu.

ABSTRACT OF THESIS

ALL DIGITAL DESIGN AND IMPLEMENTAION OF PROPORTIONAL-INTEGRAL-DERIVATIVE (PID) CONTROLLER

Due to the prevalence of pulse encoders for system state information, an all-digital proportional-integral-derivative (ADPID) is proposed as an alternative to traditional analog and digital PID controllers. The basic concept of an ADPID stems from the use of pulse-width-modulation (PWM) control signals for continuous-time dynamical systems, in that the controller's proportional, integral and derivative actions are converted into pulses by means of standard up-down digital counters and other digital logic devices. An ADPID eliminates the need for analog-digital and digital-analog conversion, which can be costly and may introduce error and delay into the system. In the proposed ADPID, the unaltered output from a pulse encoder attached to the system's output can be interpreted directly. After defining a pulse train to represent the desired output of the encoder, an error signal is formed then processed by the ADPID. The resulting ADPID output or control signal is in PWM format, and can be fed directly into the target system without digital-to-analog conversion. In addition to proposing an architecture for the ADPID, rules are presented to enable control engineers to design ADPIDs for a variety of applications.

KEYWORDS: proportional-integral-derivative, microprocessor, analog, digital,
counters, frequency, all-digital, B2 Spice

Hui Hui Chin

11 January 2006

ALL DIGITAL DESIGN AND IMPLEMENTATION OF PROPORTIONAL-INTEGRAL-DERIVATIVE (PID) CONTROLLERS

By

Hui Hui Chin

Dr. Bruce Walcott
Director of Thesis

Dr. YuMing Zhang
Director of Graduate Studies

11 January 2006

RULES FOR THE USE OF THESES

Unpublished theses submitted for the Master's degree and deposited in the University of Kentucky Library are as a rule open for inspection, but are to be used only with due regard to the rights of the authors. Bibliographical references may be noted, but quotations or summaries of parts may be published only with the permission of the author, and with the usual scholarly acknowledgements.

Extensive copying of publication of the thesis in whole or in part also requires the consent of the Dean of the Graduate School of the University of Kentucky.

A library that borrows this thesis for use by its patrons is expected to secure the signature of each user.

Name	Date

THESIS

Hui Hui Chin

The Graduate School

University of Kentucky

2006

ALL DIGITAL DESIGN AND IMPLEMENTATION OF PROPORTIONAL-INTEGRAL-DERIVATIVE (PID) CONTROLLERS

THESIS

A thesis submitted in partial fulfillment of the requirements for
the degree of Master of Science in Electrical and Computer Engineering
in the College of Engineering at the University of Kentucky

By

Hui Hui Chin

Lexington, Kentucky

Director: Dr. Bruce Walcott, Professor of Electrical and Computer Engineering

Lexington, Kentucky

2006

Copyright © Hui Hui Chin 2006

MASTER'S THESIS RELEASE

I do not authorize the University of Kentucky
Libraries to reproduce this thesis in
whole or in part for purposes of research.

Hui Hui Chin

11 January 2006

ACKNOWLEDGMENTS

This thesis, while an individual work, benefited from the insights and direction of several people. The author wishes to express her gratitude and appreciation to Dr. Bruce Walcott for his patience, constant encouragement, invaluable guidance and advice throughout this research work. Dr. Michael Marra and Dr. Walcott are the primary originator of the ADPID idea behind this thesis. Gratitude is also expressed to Dr. YuMing Zhang and Dr. William Dieter for serving as committee members for this thesis defense. Their recommendations and time are truly appreciated. Technical support provided by Beige Bag Software, Inc. was also invaluable in making this thesis a success.

Special appreciation is extended to the Department of Electrical and Computer Engineering at the College of Engineering and the University of Kentucky, for providing facilities to carry out this work and a fabulous graduate study experience. In addition, my graduate study would not have been possible without financial support from the Graduate School – specifically in granting me the Kentucky Graduate Scholarship. Throughout my time as a graduate student, I was supported as a Teaching Assistant by the department which further facilitates my study.

Equally important are the people around me. I am very grateful to my parents for their unwavering support throughout my years at the University of Kentucky. Special thanks also go out to all my colleagues and friends at the college. They are the one making made my experience a truly unique one.

TABLE OF CONTENTS

Acknowledgements iii

List of Tables vii

List of Figures viii

Chapter 1: Introduction

1.1 Background 1

1.2 Scope of Thesis 4

Chapter 2: Literature Review

2.1 Brief History of PID Controller 6

2.2 Systems Involve Encoder Feedback Techniques 8

Chapter 3: Analog and Digital Design of PID Controller

3.1 Introduction 13

3.2 Analog PID 13

3.2.1 Root Locus Method 13

3.2.1.1 Procedures for Designing an Analog PID Controller by the Root Locus Method 15

3.2.1.2 Example of an Analog PID Root Locus Design 15

3.2.2 Frequency Response Method 18

3.2.2.1 Procedures for Designing an Analog PID Controller by the Frequency Response Method 19

3.2.2.2 Example of an Analog PID Frequency Response (Bode Plot) Design 19

3.2.3 Ziegler-Nichols Tuning Method for PID Controllers 21

3.2.3.1 Procedure for an Open Loop Test 22

3.2.3.2 Procedure for an Closed Loop Test 23

3.2.3.3 Example of an Open Loop Ziegler-Nichols Tuning Method 24

3.3 Digital PID 26

3.3.1 Conversion from Analog to Digital PID 26

3.3.2 Direct Root Locus Design... 28
3.3.3 Direct Frequency Design 28
3.3.4 Tuning for Digital PID.. 29

Chapter 4: All-Digital PID

4.1 Introduction 30
4.2 The First Patent on an ADPID Controller.. 30
4.3 The Modification of the Original ADPID in This Thesis 33
4.4 Overview of the Contrast between a Digital PID and an ADPID............... 36
4.5 Design of an ADPID 37
4.5.1 Error Signal (P) 38
4.5.2 Error Directional Signal (D) 39
4.5.3 ADPID Signal 40
4.5.3.1 Base Frequency..41
4.5.3.2 Proportional Error Signal... 41
4.5.3.3 Integral Error Signal 45
4.5.3.4 Derivative Error Signal 48
4.5.3.5 Combination of the Three Separate Terms (P, I, D)...... 52
4.5.3.6 PWM Amplifying gain 55
4.5.3.7 Theoretical Analog and All-Digital PID Signal Comparison 56

Chapter 5: ADPID Simulation and Results

5.1 Introduction.. 63
5.2 Case Study - Inkjet Printer Carriage Motion Control 63
5.2.1 Transfer Function for the Inkjet Printer Carriage Motion System 63
5.2.2 PID Controller for the Inkjet Printer Carriage Motion System..... 66
5.2.3 Simulation and Results on a PID Compensated System Based on B2 Spice 68
5.2.3.1 Analog Simulation 69
5.2.3.2 All-Digital Simulation 70
5.2.3.3 Analog and All-Digital PID Simulation Comparison.... 71

Chapter 6: Conclusion and Future Work

6.1 Conclusion 76
6.2 Future Work 83

References 85

Vita 88

LIST OF TABLES

Table 3.1: Open Loop Ziegler-Nichols Tuning Parameter on Step Response..... 23
Table 3.2: Closed Loop Ziegler-Nichols Tuning Parameter................................ 24
Table 4.1: Truth Table of an EXOR 39
Table 4.2: Summary of Proportional Term Counting Sequence.......................... 45
Table 4.3: Summary of Integral Term Counting Sequence 48
Table 4.4: Summary of Derivative Term Counting Sequence............................. 51
Table 4.5: Summary of Combine Counters Counting Sequence 55
Table 5.1: Parameters of a Cartridge Transport Mechanism [21]........................ 64

LIST OF FIGURES

Figure 2.1: ADPLL Servo Control System with Optical Encoder [24] 11
Figure 3.1: Closed Loop System 14
Figure 3.2: Step Response of the Closed Loop Compensated System by Root Locus 17
Figure 3.3: Simulated steady_state_error_parabola of approximately 0.0001...... 17
Figure 3.4: Open Loop Uncompensated System by Bode Plot 20
Figure 3.5: Open Loop PID System Compensated by Bode Plot.......................... 21
Figure 3.6: Open Loop Ziegler-Nichols Step Response Measurement................. 22
Figure 3.7: Closed Loop Ziegler-Nichols Measurement 24
Figure 3.8: Ziegler-Nichols Tuning Method on an Open Loop System................25
Figure 3.9: Closed loop Compensated System by Ziegler-Nichols Tuning Method 26
Figure 4.1: Counters Structure Illustrated In the Patent 34
Figure 4.2: Counters Structure of the Modified All-Digital PID 35
Figure 4.3: Digital PID Controller in an Encoded system....................................36
Figure 4.4: All-digital PID Controller in an Encoded System 37
Figure 4.5: Generation of an Error Signal 38
Figure 4.6: State Diagram to Control Count Enable and Load Signal for a Proportional Counter 44
Figure 4.7: State diagram to Control Count Enable and Load Signal for an Integral Counter 47
Figure 4.8: State Diagram to Control Count Enable and Load Signal for a Derivative Counter 50
Figure 4.9: State Diagram to Control Multiplexer Select Line and Load Signal for a Combine Counter 54
Figure 4.10(a): PWM of a Proportional Signal.. 57
Figure 4.10(b): All-Digital Simulation of a Proportional Signal 57
Figure 4.10(c): Analog Method of Proportional Gain Multiplied by Input Voltage ... 58
Figure 4.11(a): PWM of an Integral Signal 59
Figure 4.11(b): All-Digital Simulation of an Integral Signal.......................................59
Figure 4.11(c): Analog Method of Integral Gain Multiplied by Input Voltage........... 60
Figure 4.12(a): PWM of a Derivative Signal... 61
Figure 4.12(b): All-Digital Simulation of a Derivative Signal 61
Figure 4.12(c): Analog Method of Derivative Gain Multiplied by Input Voltage 62
Figure 5.1: Analytical Model of a Cartridge Transport Mechanism [21] 64
Figure 5.2: Root Locus for a PID Compensated System.......................................67
Figure 5.3: Step Response for an Uncompensated System and a PID Compensated System 68
Figure 5.4: Analog Simulation Tracking 1 Volt for 250ms 69
Figure 5.5: All-Digital Simulation Tracking 1 Volt for 350ms............................ 71
Figure 5.6: Generated Error during the Simulation... 73
Figure 5.7: Counting Frequency for Integral and Derivative Counters at 10Hz ... 73

CHAPTER 1

INTRODUCTION

1.1 Background

Proportional-Integral-Derivative (PID) controllers have been in existence for nearly two-thirds of a century. They remain a key component in industrial process control as over 90% of today's industrial processes are controlled by PID controllers [1]. Due to its simplicity, versatility, speed, reliability, flexibility and robustness, many industries still rely on this stalwart controller for all types of control. Example includes temperature, engine speed and position control among many others.

PID controllers have evolved from analog controllers using mechanical integrators and differentiator, to digital controllers using microprocessors and encoders. Indisputably, digital controllers using microprocessors dominate industrial control today. Many advantages of microprocessor-based controller can be found in [2-3]. Microprocessor control is less expensive to implement than its analog counterpart, and is capable of utilizing advanced control algorithm. Other advantages of microprocessor-based control include flexibility in changing parameter, lighter weight and greater insensitivity to noisy external signals.

Yet, the majority of industrial dynamical systems utilizing digital control are continuous, rather than discrete. Thus, using digital controllers on such systems typically involves processing an analog sensor signal, in order for the microprocessor to obtain system output information. This process is commonly known as analog-to-digital conversion (ADC). Likewise, the control signal produced by the microprocessor typically requires translation into analog form prior to being fed into the system's input. This process is known as digital-to-analog conversion (DAC). Both ADC and DAC can introduce error, delay or loss of information.

The introduction of programmable logic devices (PLD) has opened a new era in digital implementation. A comparison between PLDs and microprocessors in terms of system design and development can be found in [4]. This report clearly shows that PLDs have the potential to replace custom microprocessors. The reasons given in [4] include the facts that PLDs are less expensive, require shorter time-to-market, have no non-recurring engineering costs, and have faster simulation times. For these reasons, there is an opportunities to replace microprocessors with PLDs. Simultaneously, there exists a similar opportunity to eliminate ADC and DAC when implementing digital PID controllers.

Inkjet printers are one of the many applications that utilize digital PIDs. In this specific application, the objective is to control the speed of the cartridge carriage inside the printer. Inkjet printers have become a popular choice for home users as well as small businesses, costing less than laser printers. The challenge in this application is to continue to reduce cost while maintaining print quality; marketplace pressure to lower cost and improve the quality of printing have pushed printer designers to continually search for better ways to improve the product.

In an inkjet printer, the head that deposits the ink is attached to a carriage which typically houses the ink reservoirs. This carriage moves across the page at a constant speed to deposit the ink uniformly onto the paper. This carriage mechanism can be actuated either in open loop by a stepper motor, or in closed loop by a DC motor [5]. The advantages of a stepper motor and open loop control in this application include: 1) noncumulative error; 2) reliability and greater life span as there are no contact brushes; 3) full torque available at stand-still; and 4) lower costs [6]. However, increasing the speed of stepper motor generally produces unwanted oscillations. Hence, despite the aforementioned advantages, the overall performance under open loop control is limited compared to a DC motor with closed loop control. In this specific application, closed loop control is also preferred, in that media position drift can be compensated for by adjusting the control signal until the speed of the carriage matches that of the reference, thereby improving print quality substantially.

Current inkjet printer systems combine Reduced Instruction Set Computer (RISC) and Application-Specific Integrated Circuit (ASIC) for image processing and printer control. A microprocessor controls the printing process, while an ASIC implements the digital circuitry to support the microprocessor [22]. Improvements in both RISC and ASIC technologies effectively reduce the cost of a printer. Yet, differences in individual microprocessor architecture and clock speed introduce challenges in simulation porting control code from one platform to another [7]. If an Field-Programmable Gate Array (FPGA) can replace the microprocessor in an inkjet printer, lower production costs will ultimately occur.

An all digital PID controller (ADPID) introduced in this thesis is a means of replacing a PID controller in microprocessor with pure digital logic, that can be programmed in a simple FPGA chip. Furthermore, an ADPID eliminates ADC and DAC conversion and the associated problems, such as delay. Through digital logic substitution, the cost of implementing a PID controller can also be minimized.

To prove the concept behind an all-digital PID, we selected a Lexmark Z-52 inkjet printer as a test bed. The first step in ADPID design is to convert the system's desired output into the equivalent pulse train that would be produced by a linear encoder attached to the output of the system. The next step is to produce an error signal by comparing the actual system's encoded output to this reference pulse train. The last step is for the ADPID to process this error signal and produce a control signal in Pulse Width Modulation (PWM) form, which can be sent directly to the system's input with no need of ADC.

1.2 Scope of Thesis

This thesis presents a design and implementation methodology for an All-Digital-PID-Controller (ADPID) that can replace traditional analog and digital PIDs. The proposed ADPID implementation requires only digital logic (*i.e.*, FPGAs, Complex PLDs (CPLDs), *etc*). For an example application, an inkjet printer carriage control system is selected. Typical industry control requirements, such as settling time and overshoot for this application are 0.16sec and 12% overshoot, respectively.

Beyond the introduction, Chapter 2 of this thesis begins with a brief history of PID controllers. Then a literature review of several techniques for controlling the positioning of printhead carriage transportation is presented.

Chapter 3 presents an introduction to analog and digital PID controller design. The standard rules and procedures for designing a PID controller are discussed. Also, two famous design methods, Root Locus and frequency response design, are followed for both analog and digital PID.

In Chapter 4, the All-Digital-PID-Controller is introduced. The theory of the controller is discussed. The components and signals involved in the design are explained, and the procedures are developed and summarized. A step-by-step heuristic design rules are also discussed in detail.

Chapter 5 presents a case study for an ADPID design using an inkjet printer. A transfer function for the printer is derived, and simulation results will be presented and discussed.

Chapter 6 is a summary and conclusion of the thesis; some suggestion for future work to improve this ADPID design will be proposed at the end of the chapter.

CHAPTER 2

LITERATURE REVIEW

2.1 Brief History of PID Controller

PIDs combine proportional-integral-derivative control action. In 1788, James Watt included a flyball governor, the first mechanical feedback device with only a proportional function, into his steam engine. The flyball governor controlled the speed by applying more steam to the engine when the speed dropped lower than a set point, and *vice versa* [8]. In 1933, the Taylor Instrumental Company introduced the first pneumatic controller with a fully tunable proportional controller. However, a proportional controller is not sufficient to control speed thoroughly, as it amplifies error by multiplying it by some constant (Kp). The error generated is eventually small, but not zero. In other words, it generates a steady state error each time the controller responds to the load [9].

Around 1930s, control engineers discovered that steady state error can be eliminated by resetting the set point to some artificial higher or lower value, as long as the error nonzero. This resetting operation integrates the error, and the result is added to the proportional term; today this is known as Proportional-Integral controller. In 1934-1935, Foxboro introduced the first PI controller. However, PI controllers can over-correct errors and cause closed-loop instability. This happens when the controller reacts too fast and too aggressively; it creates a new set of errors, even opposite to the real error. This is known as “hunting” problem [10].

In 1920s, there were suggestions of including the rate of change of error in conjunction with PI controller. In 1940, Taylor Instrument Companies successfully produced the first PID pneumatic controller; the derivative action was called “pre-act”.

With an extra derivative action, problems such as overshoot and hunting are reduced. However, issues like finding the appropriate parameter of PID controllers were yet to be solved.

In 1942, Taylor Instrument Company's Ziegler and Nichols introduced Ziegler-Nichols tuning rules. Their well-known paper "Optimum settings for automatic controllers", presented two procedures for establishing the appropriate parameters for PID controllers. However, the PID controller was not popular at that time, as it was not a simple concept; the parameters the manufacturers required to be tuned did not make much sense to the users.

In the mid 1950's, automatic controllers were widely adopted in industries. A report from the Department of Scientific and Industrial Research of United Kingdom state, "Modern controlling units may be operated mechanically, hydraulically, pneumatically or electrically. The pneumatic type is technically the most advanced and many reliable designs are available. It is thought that more than 90 percent of the existing units are pneumatic." [11] The report indicated the need to implement controllers in electrical and electronic form.

In 1951, The Swartwout Company introduced their first electronic PID controller, based on vacuum tube technology. Around 1957, the manufacturers started to realize the possibility of implementing the controllers in transistors. In 1959, the first solid-state electronic controller was introduced by Bailey Meter Co. The advantage of using electronic instrument to implement PID controller was explored more deeply years later. They are not only capable of including the functions available in pneumatic instruments, but even more complicated mathematical operations can be carried out as well [12]. Electronic PID controllers became more common and more acceptable since then.

The digital computer became involved in process control in the 1960s. The first instance in which closed loop control was implemented by a digital computer in an industrial plant was done by Texaco's Port Arthur plant on March 15th, 1959. By 1960, many control instrument companies responded to this new technology and offered computer-based systems. "Analog controllers should gradually evolve into digital devices, providing accuracy at low cost. These controllers will be relatively simple to combine into multipoint configurations, which can be applied to optimize unit processes on a local basis." [13]. More discoveries concerning digitizing PID controllers were made, and arguments for implementing controllers on microprocessors were brought up as microprocessors could handle calculations directly in engineering units [14-15].

Due to advances of technology, the PID controller is widely and commonly used in process control, aircraft systems, automobiles, home equipment and appliances as well as portable devices nowadays. Since the introduction of many modern control theories to complement the PID controller, things have not been the same, although the fundamental theory for designing one remains the same. Hence, we are greatly indebted to those who laid the foundation for developing PID control theory.

2.2 Systems Involve Encoder Feedback Techniques

Sensors play an important role in mechanical motion. Sensors detect motion, such as velocity, shaft angle and position, from stepper or servo motors, and output the useful data to the controller. Traditionally, analog transducers are widely used in analog control.

As the technology advanced from analog to digital, analog transducers were replaced by digital transducers. Some analog transducers are still employed with digital controllers, by using an analog-digital converter chip; the analog-digital conversion is eliminated when a digital transducer is used. By doing so, the digital signal from the transducers can be directly transmitted into the controller, and noise level is reduced. More attractively, optical sensors can operate under a wide temperature range, and are resistant to magnetic fields. Such sensors are economical devices that are able to provide very high levels of resolution, accuracy and repeatability [16].

Digital encoders are optical sensors within the family of digital transducers. They are commonly used to measure linear and rotary position. Generally, the digital encoder has a light source, such as a LED, on one side of the disk, and a photodetector on the other side of the disk. The resolution of the encoder is determined by the distance between the slots in the disk. As the disk rotates, the slots in the disk interrupt the light source, and the photodetector sends a pulse train series to the computer. Thus, incremental position can be measured by counting the pulses occurring during rotation. The velocity can be determined by finding the frequency of the pulse train [17].

In 1996, Lin *et al.* successfully controlled the speed of an inkjet print head transport system using a phase-locked loop (PLL) [18]. Characteristics such as high speed response, insensitivity to noise, and commercially cheap integrated chips make PLL highly recommended for motor speed control. A PLL is composed of a phase frequency detector (PFD), loop filter, and voltage-controlled oscillator (VCO). The PFD in the model was based on the tri-state PFD presented by Best [19]. A lead-lag compensator was designed using classical root locus methods as a loop filter. It not only filters out and smoothes the output of the phase-frequency detector, but also improves the transient response of the system, according to the design specification. The VCO represents mechanical and sensor subsystems, composed of a DC motor, belt pulley transmission subsystem, linear strip, and optical sensor. In the experiment, Lin managed to regulate the speed at steady state to within 10% error when the carriage moved at 33 inches/sec. A 10% error is relatively large, but Lin's performance can definitely be improved if the closed-loop system is better modeled.

In 1997, Adkins came out with an all-digital phase-locked loop (ADPLL) [20], and successfully reduced the microprocessor load in operating a Lexmark inkjet printer. In most inkjet printers, a microprocessor and an Application-Specific Integrated Circuit (ASIC) coordinate to form a controller. A microprocessor controls the printing process, while the ASIC is programmed to support the digital circuitry needed by the microprocessor. By integrating an ASIC with the controller, the bandwidth of the microprocessor is reduced, and a more economical microprocessor can replace it.

PLL controllers to date are either all analog, or a combination of analog and digital configuration (DPLL). In [20], an ADPLL is proposed with a different design methodology than Lin's. First, Adkins analyzed the entire PLL motor system as a sophisticated non-linear system. Then, an accurate closed-loop model was derived. Following that, he designed a DPLL control system using classical control techniques in order to meet design specifications. Lastly, the analog loop filter was converted into a digital loop filter. By doing so, ADPLL can now be implemented in an ASIC. The output of the control system is connected to the optical encoder, where the frequency of the digital pulse signal is generated proportional to the velocity. The digital output is then compared with the phase-frequency detector, in order to generate the error signal. Figure 2.1 shows the implementation of an ADPLL servo control system with an optical encoder [21].

The report shows that the author meets all the design specifications. The steady state error is ±5%, overshoots are less than 20%, and the carriage attains 90% of the desired print speed before the print head traverses 0.5 inches.

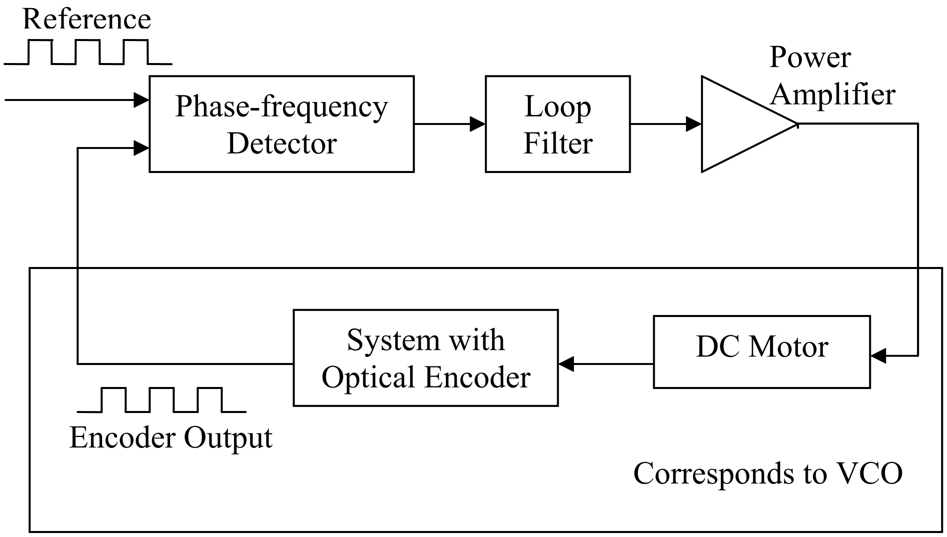

Figure 2.1: ADPLL servo control system with optical encoder [24]

Deshpande [22, 23] designed and implemented *Dynamic Print Mode Control* (DPMC) on an inkjet printer motion control system, using a Digital Signal Processing (DSP), in his master thesis in 2001. DPMC is a method that optimizes the tradeoff between print quality and print speed. The system in current commercial inkjet printers is based on RISC and ASIC architecture, for image processing and printer control. However, due to the high performance of its real-time execution and compilers on a real-time operating system, the author claimed that image processing and printer engine control as well as time critical functions can be done on a single DSP. The motivation behind a single DSP is that it reduces production cost and yet provides high performance, and can be leveraged to suit all different kinds of market.

For the cartridge motion control system, the author designed a Zero-Phase-Error-Tracking (ZPET)-based feed-forward controller for system stability, and a Disturbance Observer feedback controller to handle disturbance and uncertainty (*i.e.* friction, un-modeled parameter) while controlling the tracking motion. As a result, the author analytically obtained a maximum carriage velocity of about 40 inches per second (ips), with a steady state error of approximately ±3%.

CHAPTER 3

ANALOG AND DIGITAL DESIGN OF PID CONTROLLER

3.1 Introduction

In this chapter, typical methods used to design analog and digital PID controllers are discussed. First, PID compensator design based upon root locus is introduced, and the procedure for designing the compensator is explained. Next, PID design based on a frequency response method is discussed. Finally, the Ziegler-Nichols tuning method is briefly introduced.

3.2 Analog PID

Analog PID controllers are common in many applications. They can be easily constructed using analog devices such as operational amplifiers, capacitors and resistors. They are reliable in mechanical feedback systems, and able to satisfy many control problems.

3.2.1 Root Locus Method

Root locus is one of the methods used to design control systems. It is a technique that plots closed-loop poles in the complex plane as the gain varies from zero to infinity. It is a method that analyses the relationship between the poles, gain and the stability of the system. By understanding the root locus plot, one can design a controller to novel specifications, and understand clearly how different controller architectures affect the system.

In a root locus, the imaginary component of a pole corresponds to damped natural frequency, while the radius from the origin to the pole corresponds to natural frequency. The settling time for a system is determined by the slowest response among all responses. The least settling time can be achieved if the roots fall to the far left on the left-hand plane; overshoot can be prevented by placing the poles on the real axis.

In order to design a PID controller using the root locus method, the system must be first transformed into a transfer function. In general, root locus technique analyzes only single input single output (SISO) systems. However, an appropriate approximation of transforming a multi input multi output (MIMO) system into a SISO model can produce a close estimation of the characteristics of the system. A root locus that passes through the right-hand plane is considered unstable, whereas one that remains in the left-hand plane implies a stable system. A root locus that falls in the jω axis (between the right- and left-hand planes) is considered marginal stable.

Figure 3.1 is an example of a close loop system. K represents the PID controller, G represents the transfer function of the system, and H represents the feedback parameter.

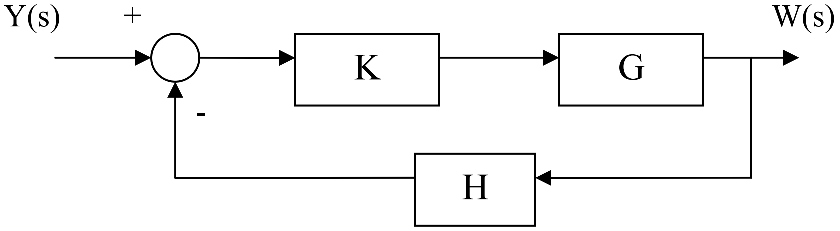

Figure 3.1: Closed loop system

3.2.1.1 Procedures for Designing an Analog PID Controller by the Root Locus Method

I. Develop a set of reasonable transient specification based upon the particular application. From the specifications, find a pair of closed-loop dominant poles which meet these specifications, s_1 and s_1*.

II. Find K_I term from steady-state error, e_{ss}.

III. Lump $\frac{K_I}{s}$ term into the G_{PID} together with G(S).

IV. Solve for K_P and K_D by using

$$G_{PID}G(s_1) = -1$$ Equation 3.1

V. Equation 3.1 is rearranged such that

$$K_P + K_D s_1 = -\frac{1}{G(s_1)} - \frac{K_I}{s_1}$$ Equation 3.2

VI. Hence, K_P and K_D can be solved by equating the real and imaginary term on the left and right side of the equation.

VII. Sketch the resulting root locus for the compensated system.

3.2.1.2 Example of an Analog PID Root Locus Design

A set of specification such as settling time, overshoot and steady-state error is required to design a PID controller. Settling time is the time required for the process variable to settle to within 2% of the target value. Overshoot represents the maximum percentage of the process variable overshoots the target value. Steady-state error expresses the final difference between the process variable and the set point.

The example will be designing a PID controller by root locus method, with the following specification:

Settling time = 0.137 second

Overshoot < 30% (Damping ration, ζ=0.377)

$\text{Steady\_state\_error}_{\text{parabola}} = 1/3070$

$$transfer\ \ function = \frac{6.49}{s(s+47.579)} \qquad \text{Equation 3.3}$$

From the design specification, the desired closed-loop dominant poles are -29.14+j47.02. By going through procedure III to VI in section 3.2.1.1, proportional, integral and derivative gains are found 900.12, 22507 and 9 respectively. The step response of the closed loop compensated system by root locus is plotted in Figure 3.2.

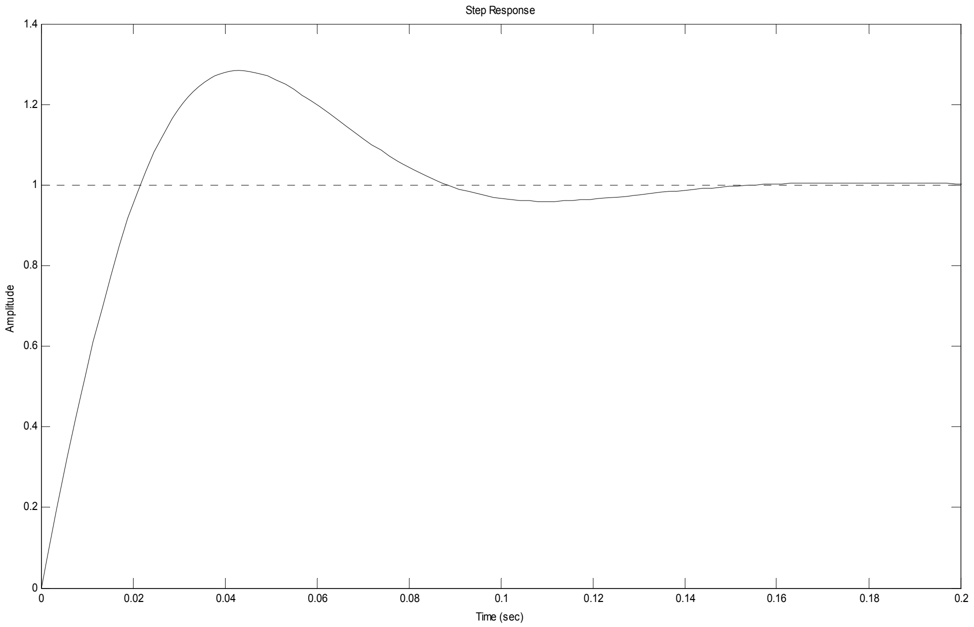

Figure 3.2: Step response of the closed loop compensated system by root locus

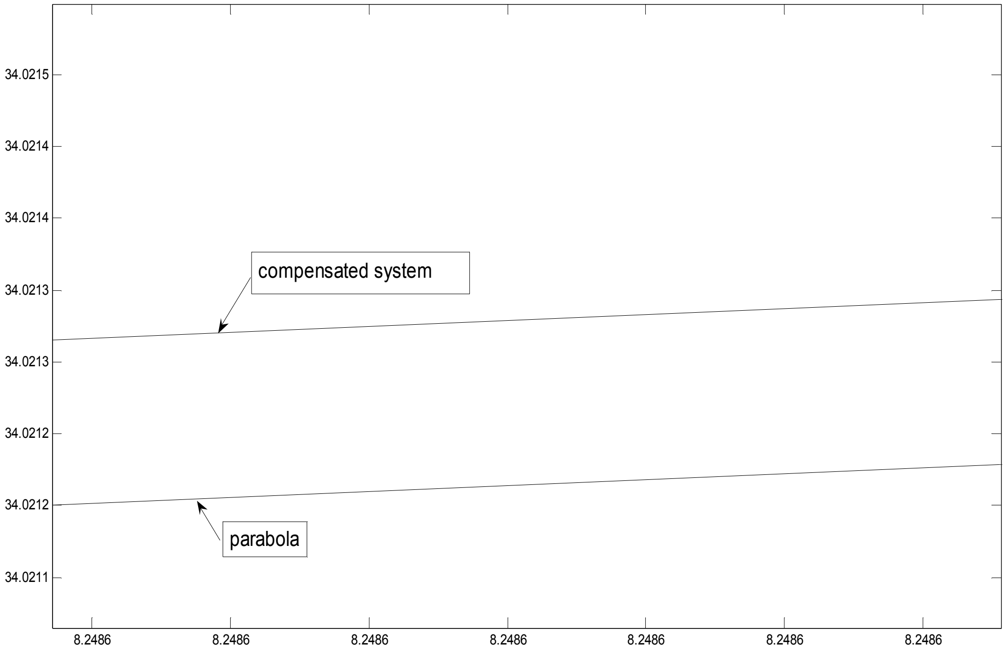

Figure 3.3: Simulated steady_state_error_parabola of approximately 0.0001

3.2.2 Frequency Response Method

Frequency response is another method commonly used to design a PID controller. Unlike root locus for the s-domain, using poles and zeros, frequency response uses the magnitude and phase of the controller to shape the curve in order to meet the specifications.

Each individual term of a PID controller is defined differently in Bode plots. As in root locus, the proportional term does not change the shape of the plot; it adjusts the gain and phase margins by shifting the magnitude of the Bode plot up or down. The integral term adds a slope of -20dB/dec to the phase; it tends to destabilize the system by adding a constant -90 degrees to the phase angle of the system. The derivative term increases the phase margin by adding a +90 degree phase angle into the system, which corresponds to the damping ratio; also, a slope of +20dB/dec is contributed to the phase.

Two important parameters in determining the system stability in the frequency response method are gain margin (GM) and phase margin (PM). Phase margin can be found by finding the crossover frequency when the phase angle is -180 degrees, and measuring the magnitude distance below 0 dB. Similarly, gain margin can be found by finding the crossover frequency when the magnitude plot is 0 dB, and measuring the angle distance above -180 degrees. The system is unstable if the magnitude plot is not below the 0 dB line when the system is at -180 degrees, or if the phase plot is not above -180 degrees when the system is at 0 dB

3.2.2.1 Procedures for Designing an Analog PID Controller by the Frequency Response Method

I. Make sure the open loop system is stable.

II. Draw the Bode plot of the open loop system.

III. From the design specification, phase margin is related to damping ratio, ζ as in Equation 3.4. Also, the ratio of the crossover frequency and the natural frequency is related as in Equation 3.5.

$$\text{Phase Margin} = \tan^{-1}\frac{2\zeta}{\sqrt{-2\zeta^2+\sqrt{1+4\zeta^4}}}$$

Equation 3.4

$$\frac{\omega_c}{\omega_n} = \sqrt{-2\zeta^2+\sqrt{1+4\zeta^4}}$$

Equation 3.5

IV. By understanding the contribution of P, I and D on a Bode plot, they can be specified such that the design specification for a closed loop system, such as phase margin and crossover frequency can be fulfilled.

3.2.2.2 Example of an Analog PID Frequency Response (Bode Plot) Design

Design a PID controller with the specification as in Equation 3.4.

The Bode plot of the open loop system is shown in Figure 3.4.

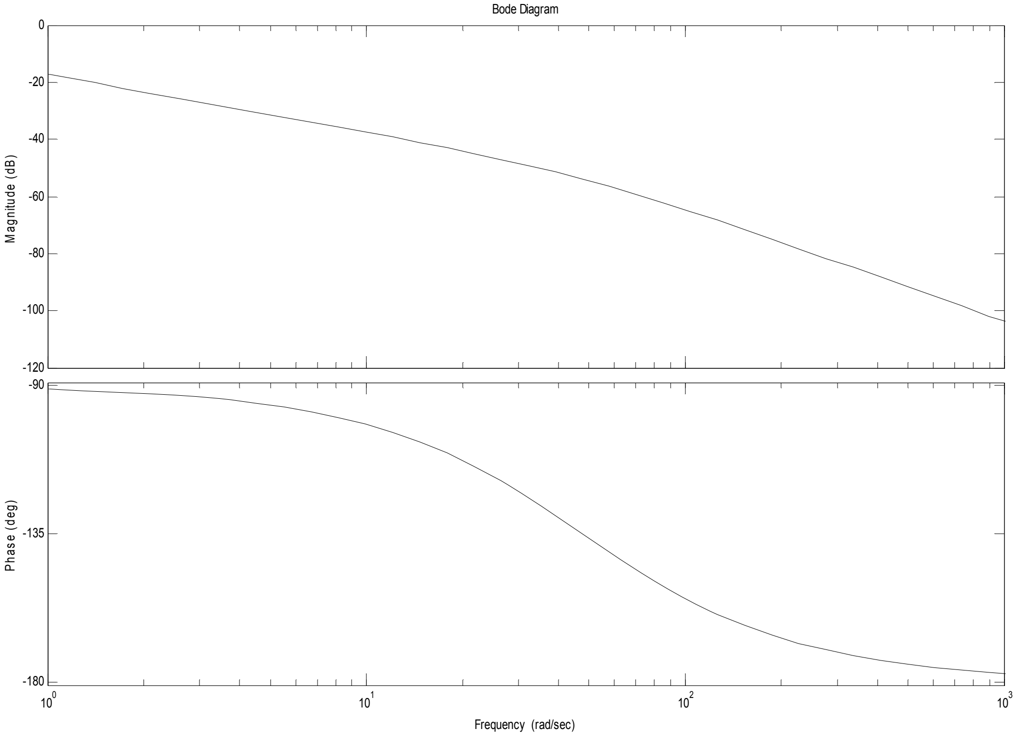

Figure 3.4: Open loop uncompensated system by Bode plot

From Equation 3.4 and Equation 3.5, the phase margin is found 41^o, and the crossover frequency is 66.47 rad/sec. Thus the P, I and D gain are 850, 22000 and 10 respectively. The corresponding open loop Bode plot is shown in Figure 3.5.

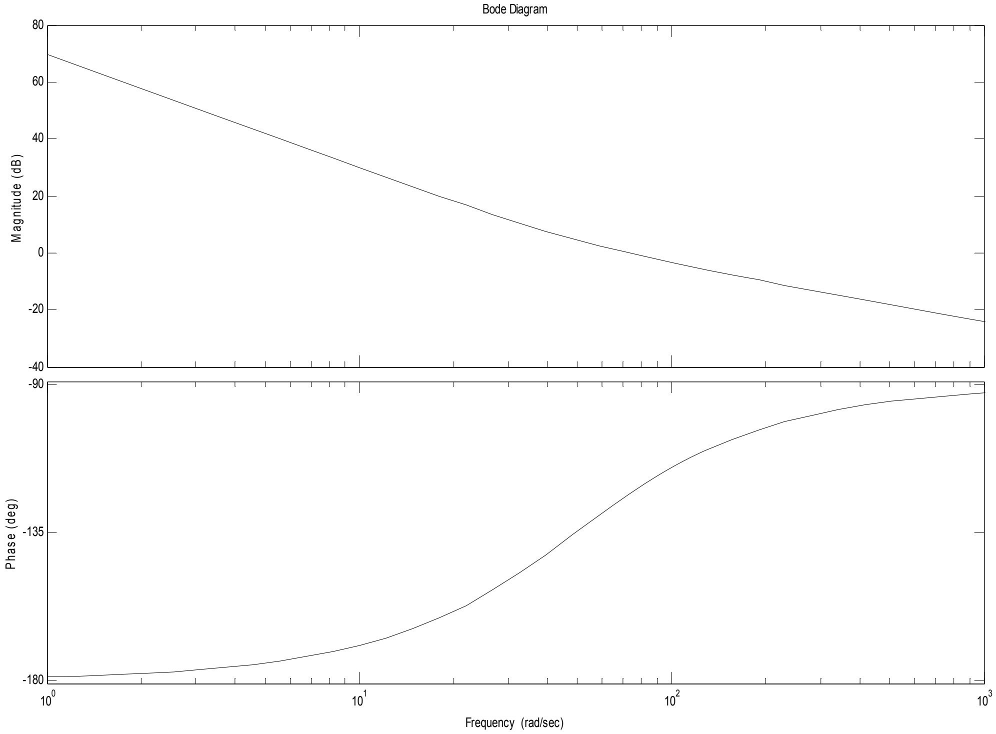

Figure 3.5: Open loop PID system compensated by Bode plot

3.2.3 Ziegler-Nichols Tuning Method for PID Controllers

The Ziegler-Nichols tuning method is based on both open and closed loop testing. This tuning method is useful if one purchases a generic PID controller and wants to tune it to control a complex system. In order to do so, system parameters needed to be found experimentally; the results are interpreted into proper information for the Ziegler-Nichols PID controller equation, Equation 3.8.

$$Gc = Kp(1 + \frac{1}{T_i s} + T_d s) \qquad \text{Equation 3.8}$$

In Equation 3.8, integral time, T_i and derivative time, T_d replace integral gain and derivative gain; they are more commonly used when the Ziegler-Nichols method is applied.

3.2.3.1 Procedure for an Open Loop Test

I. Make a step test on an open loop plant. Graph like Figure 3.6 should be attained.

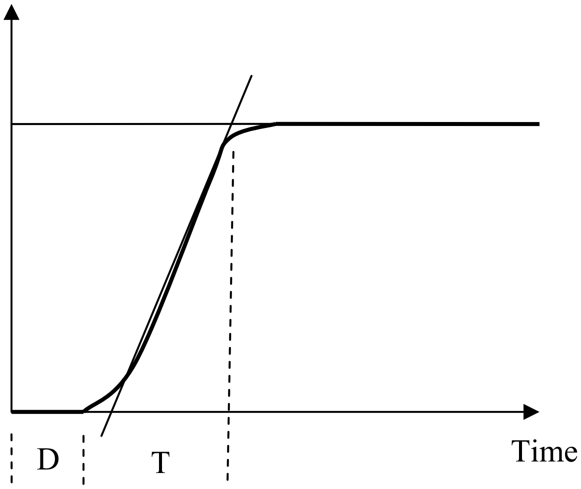

Figure 3.6: Open loop Ziegler-Nichols step response measurement

II. Determine dead time, D and rise time, T as in Figure 3.6.

III. Table 3.1 lists the Ziegler-Nichols tuning parameters for P, PI and PID controller. Substitute D and T found from step II into the table to calculate the gains.

IV. Substitute the K_p, T_i and T_d found in Table 3.1 into Equation 3.8 to obtain proportional gain, integral gain and derivative gain.

Table 3.1: Open loop Ziegler-Nichols tuning parameter on step response

Controller Type	K_p	T_i	T_d
P	$\frac{T}{D}$	-	-
PI	$0.9*\frac{T}{D}$	$\frac{D}{0.3}$	-
PID	$1.2*\frac{T}{D}$	2*D	0.5*D

3.2.3.2 Procedure for a Closed Loop Test

I. Disable integral and derivative action of the controller so that the controller only has proportional control action.

II. Make a set point test (*i.e* step response) and increase the proportional gain until stable oscillation is achieved. The proportional gain at stable oscillation is recorded as K_u.

III. Read the oscillation period, T_u as shown in Figure 3.7.

IV. Substitute both K_u and T_u into Table 3.2 to obtain K_p, T_i and T_d. Finally, proportional, integral and derivative gain can be found from Equation 3.8.

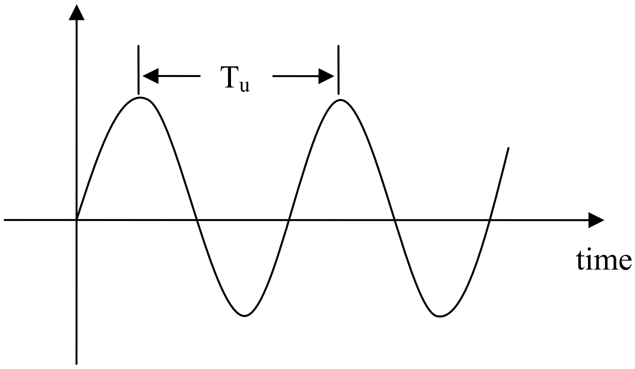

Figure 3.7: Closed loop Ziegler-Nichols measurement

Table 3.2: Closed loop Ziegler-Nichols tuning parameter

Controller Type	K_p	T_i	T_d
P	$0.5 * K_u$	-	-
PI	$0.45 * K_u$	$\frac{T_u}{1.2}$	-
PID	$0.6 * K_u$	$0.5 * T_u$	$0.125 * T_u$

3.2.3.3 Example of an Open Loop Ziegler-Nichols Tuning Method

The specification in Equation 3.3 will be used to find the proper P, I and D gains by using Ziegler-Nichols tuning method.

$$\frac{6.49}{s(s+47.579)} \qquad \text{Equation 3.9}$$

The open loop step test for 50 seconds was made on the model and Figure 3.8 was obtained.

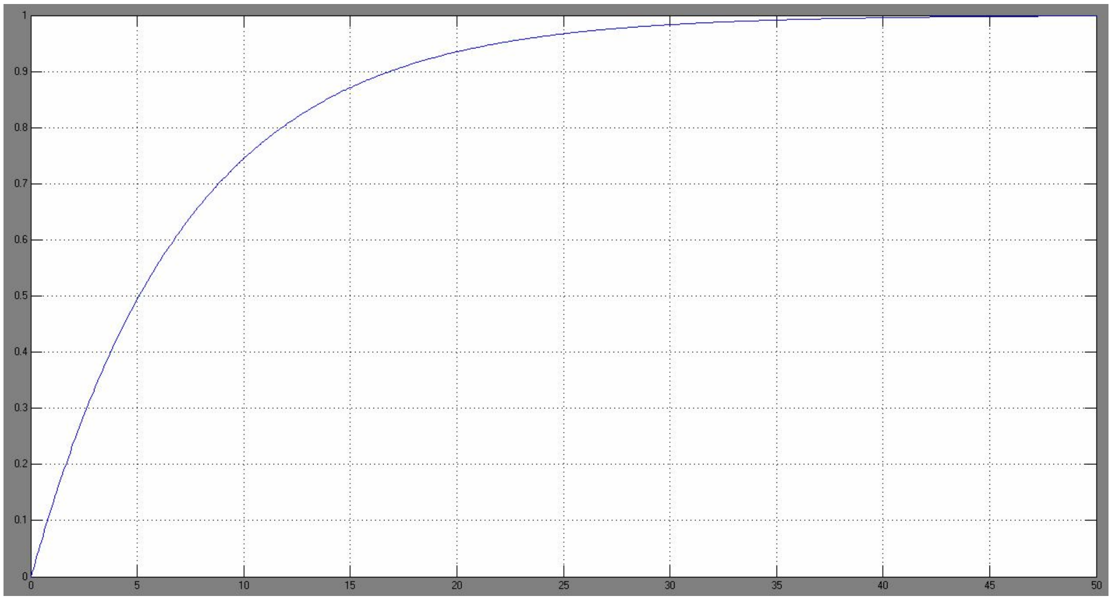

Figure 3.8: Ziegler-Nichols tuning method on an open loop system

The dead time, D = 0.02 second, and rise time, T = 15 second. Thus, $K_P = 900$, $K_I = 22500$, $K_D = 9$. The compensated closed loop system for 1 second is shown in Figure 3.9.

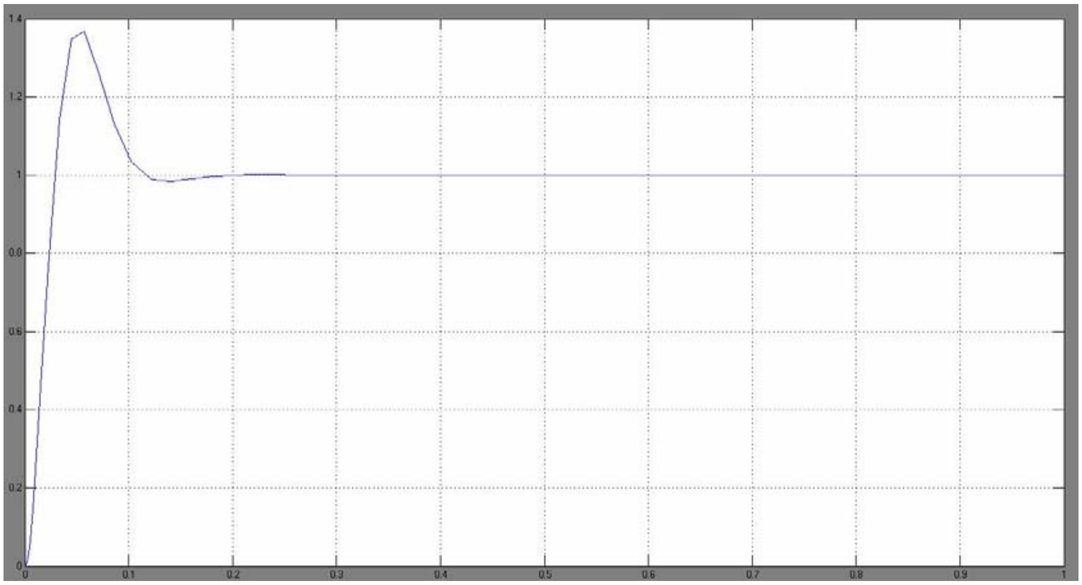

Figure 3.9: Closed loop compensated system by Ziegler-Nichols tuning method

3.3 Digital PID

Digital PID is commonly used because it is more suitable to design for a complex system for the purpose of reducing cost, and is more immune to noise than an analog PID. Several methods can be used to design a digital PID. One of the methods is to design an analog PID first, then convert the s-domain into the z-domain with appropriate approximation. A digital PID can also be directly designed by the root locus and direct response methods.

3.3.1 Conversion from Analog to Digital PID

The conversion from s-domain into z-domain is quick and easy. The conversion can be done by using difference approximation, ZOH (zero-order hold), bilinear transformation or first-order hold. In this section, the difference approximation equation is derived.

The proportional term in PID can be approximated as:

$$K_P e(k) \qquad \text{Equation 3.10}$$

The backward rectangular rule approximation of integral term in PID:

$$K_I T e(k-1) \qquad \text{Equation 3.11}$$

Also, the backward difference approximation of derivative term in PID:

$$\frac{K_D}{T}[e(k)-e(k-1)] \qquad \text{Equation 3.12}$$

However, the integral term requires previous information. Thus, the summation of the three terms becomes, where T denotes the sample period:

$$\begin{aligned} u(k) &= K_P e(k) + a(k) + \frac{K_D}{T}[e(k)-e(k-1)] \\ a(k) &= a(k-1) + K_I T e(k-1) \end{aligned} \qquad \text{Equation 3.13}$$

Equation 3.13 is the position algorithm of the present control output. The velocity algorithm for the PID is:

$$\begin{aligned} u(k-1) &= K_P e(k-1) + a(k-1) + \frac{K_D}{T}[e(k-1)-e(k-2)] \\ a(k-1) &= a(k-2) + K_I T e(k-2) \end{aligned} \qquad \text{Equation 3.14}$$

By subtracting Equation 3.14 from Equation 3.13, the digital PID is approximated as:

$$u(k) - u(k-1) = K_P [e(k) - e(k-1)] + K_I T e(k-1) + \frac{K_D}{T}[e(k) - 2e(k-1) + e(k-2)]$$

Equation 3.15

3.3.2 Direct Root Locus Design

Root locus design for a digital PID is similar to an analog PID. Basically, the rules for drawing the root locus for both are the same except that stability, frequency and damping ratio are changed.

In terms of stability, it is suggested that the poles be placed in the right-hand plane, and inside the unit circle. The closer the poles are to the origin, the faster the settling time will be. The procedure to design a digital PID is exactly the same as an analog PID, where the poles and zeros work together to shape the root loci to the desired location.

Even though there is no need to physically build a controller algorithm as the analog PID, one needs to consider whether the digital PID is realizable (*i.e.* the controller does not requires future variables). If the controller is not programmable, the digital PID needs to be redesigned. Modification such as adding another pole inside the unit circle can possibly make the controller realizable.

3.3.3 Direct Frequency Design

Direct frequency design is useful especially in deadbeat control, a method to make the system meet commands one sample time later than the desired time.

Using direct frequency design, system requirements are first considered, and written in the form of a transfer function. The controller and system transfer function is set equal to the desired transfer function. Then, the proportional, integral and derivative terms can be solved. This is illustrated in Equation 3.16.

$$T(z) = \frac{C(z)}{R(z)} = \frac{D(z)G(z)}{1 + D(z)G(z)} \qquad \text{Equation 3.16}$$

In Equation 3.16, T(z) represents the desired transfer function, C(z) represents sampled system output, R(z) represents sampled system desired input, D(z) represents a controller transfer function, and G(z) represents a discrete system transfer function.

Again, the digital PID must be programmable, so that it does not require the knowledge of future variables.

3.3.4 Tuning for Digital PID

The procedure of Ziegler-Nichols tuning for a digital PID is the same as tuning an analog PID, explained in section 3.2.3.1. The main difference between them is the sampling time. If the sampling time designed for the digital PID is small compared to system response, an analog tuning method like Ziegler-Nichols works well in a digital PID. However, if the sampling time is larger than the system response, the tuning becomes inaccurate. Thus, it is important to select and design the sampling time wisely, in order to achieve optimum performance.

CHAPTER 4

ALL-DIGITAL PID

4.1 Introduction

ADPID is an all-digital implementation of a PID controller. PID controllers currently available on the market are implemented by programmable controller (PC) or DSPs. In 1998, Professors Marra and Walcott proposed the ADPID in their patent application, as a means to implement PIDs using only digital logic.

In this chapter, the theory of designing an ADPID is discussed. Each component used, as well as the role of each signal involved, is explained. Also, the procedures and basic rules for designing an ADPID are clearly listed and walked through step by step. The objective of the design is to minimize the hardware and to speed the execution process.

4.2 The First Patent on an ADPID Controller

The idea of an ADPID was first introduced by Professors Bruce Walcott and Michael Marra from the University of Kentucky. A patent application titled "Asynchronous Digital Implementation of PID controllers" was submitted in 1998. In the application, the authors contrasted the common approach and the all-digital way of PID implementation. They realized the importance of PLDs, that they may replace customized microprocessors in the near future as they become less expensive, require shorter time-to-market, and do not suffer from obsolescence issues.

The authors considered the problems introduced by DAC and ADC in digital control of continuous dynamical systems. Both ADC and DAC produce error, delay, or loss of information. Thus, the authors saw the need to implement PIDs with an ADPID, without either ADC or DAC. The resulting ADPID implementation would be far more cost effective.

In the patent application, the authors included the requirements to implement this ADPID. For instance, an ADPID requires a digital signal proportional to the error between the reference frequency and the frequency of the pulse train from the output of the encoder. An integral term is required, representing a digital signal proportional to the integral over the time of the error between the reference frequency and the frequency of the pulse train from the output of the encoder. A derivative term is also needed, representing a digital signal proportional to the change in error between the reference frequency and the frequency of the pulse train from the output of the encoder. Also, ADPID requires a means to combine these three digital signals into one single control signal to generate a control signal.

In addition to the list of requirements, the authors proposed some helpful solutions to make this controller realizable. First, the authors represented each P, I and D term with two sets of counters, C_{P1} and C_{P2}, C_{I1} and C_{I2}, C_{D1} and C_{D2} respectively. Then, the authors varied the weighting on the P, I and D terms by varying the frequencies of the counters. The weighting on the proportional term is suggested such that

$$K_P = \frac{f_{P1}}{f_{P2}} \qquad \text{Equation 4.1}$$

In the patent application, the authors explained how each counter works. For the proportional term, if the error signal is activated and the direction of the error is in the high state, C_{P1} counts up at the frequency of f_{P1}. Likewise, C_{P1} counts down at frequency f_{P1} when the direction of the error is in the low state. As the error signal transitions from high to low, C_{P1} stops counting and resets its own counters to zero. C_{P2} loads the current state of C_{P1} and counts the state at the frequency of f_{P2} to zero.

The weighting on the integral term is suggested

$$K_I = \frac{f_{I1}}{f_{I2}} \qquad \text{Equation 4.2}$$

For the integral term, if the error signal is activated and the direction of the error is in the high state, C_{I1} counts up at the frequency of f_{I1}. Likewise, C_{I1} counts down at frequency f_{I1} when the direction of the error is in the low state. As the error signal transitions from high to low, C_{I1} stops counting and holds at the current state. C_{I2} loads the current state of C_{I1} and counts the state at the frequency of f_{I2} to zero. Then, C_{I1} resumes from the previous state if the error goes to high state.

The weighting on the derivative term is proposed such that

$$K_D = \frac{f_{D1}}{f_{D2}} \qquad \text{Equation 4.3}$$

For the derivative term, if the error signal is activated and the direction of the error is in the high state, C_{D1} counts up at the frequency of f_{D1}. Likewise, C_{D1} counts down at frequency f_{D1} when the direction of the error is in the low state. As the error signal transitions from high to low, C_{D1} stops counting and subtracts the present state from the current state of the register. The value after the subtraction will be loaded into

C_{D2}. The register is refreshed with the current state of C_{D1}. Then, C_{D1} resets its own counters. C_{D2} counts the state at the frequency of f_{D2} to zero.

Nevertheless, the authors did not mention how the three signals can be combined together to become a control signal. The generation of the control signal in PWM, and the direction of the control signal, were not defined. The authors also did not explain how the error signal and the error directional signal can be generated.

4.3 The Modification of the Original ADPID in This Thesis

The primary modification was made to the placement of the summation of the three signals. Instead of adding the signals after their second counters, the signals are now combined immediately after their first counters. A digital adder adds the three signals simultaneously when the error signal transitions from high to low state. This result is transferred to a combined counter, where the signals now share one counting frequency, f_A. In other words, only four counting frequencies, f_{P1}, f_{I1}, f_{D1} and f_A, are involved, instead of six. Figure 4.1 is the counter structure illustrated in the patent. Figure 4.2 shows the counter structure of the modified ADPID.

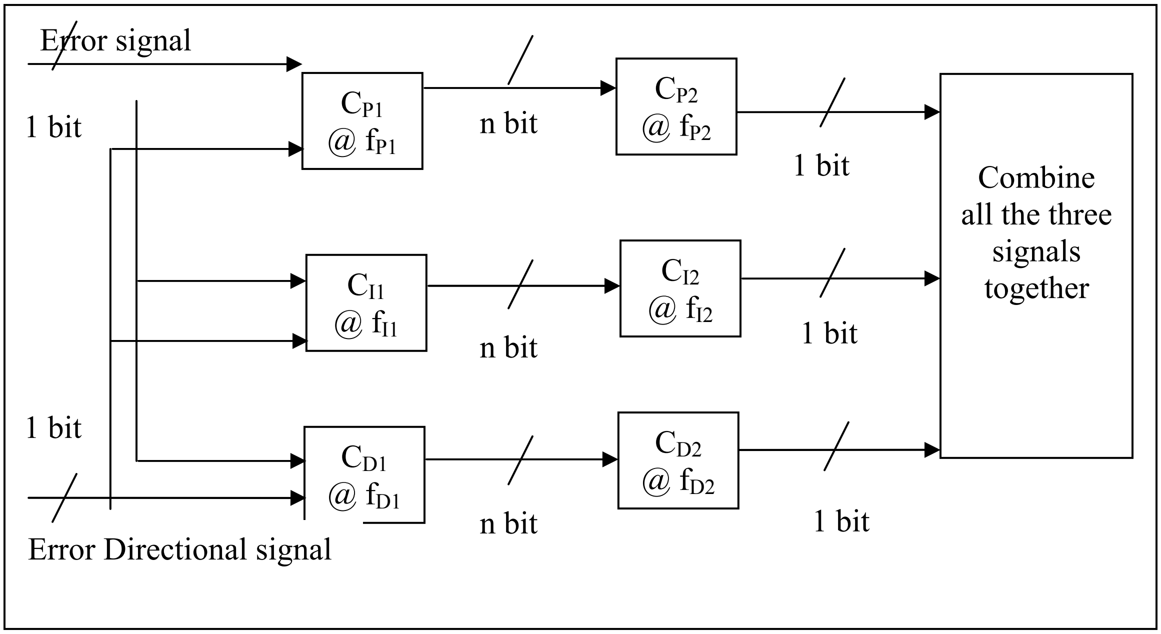

Figure 4.1: Counters structure illustrated in the patent

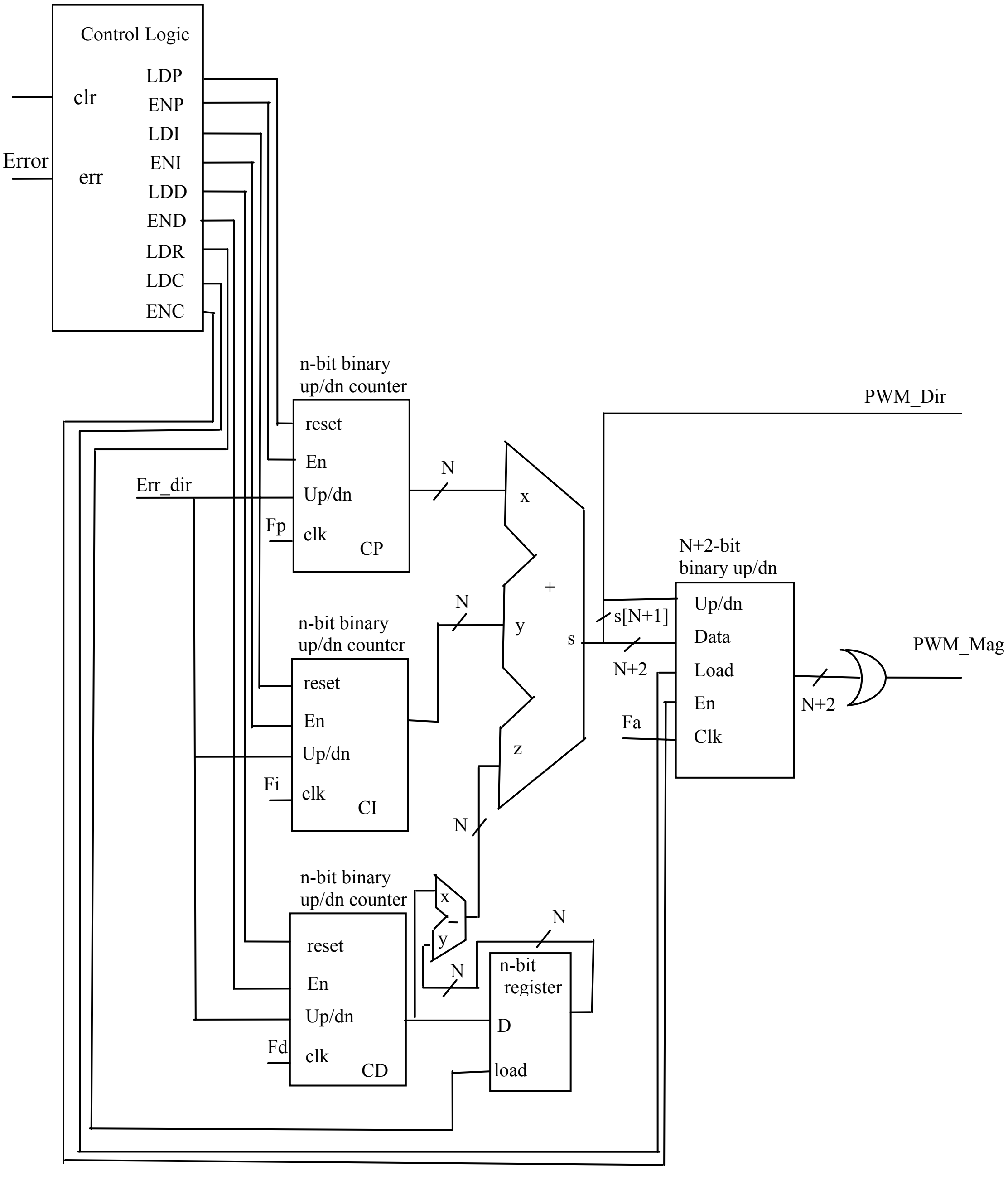

Figure 4.2: Counters structure of the modified all-digital PID

Implementing the ADPID with four counting frequencies reduces both hardware and cost. As the hardware is reduced, the time delay is possibly minimized. In the patent, no counters share a common counting frequency. As a result, the summation is always changing until they all reach zero state. This summation will be processed before it can generate a PWM signal.

Through the present proposal, a static summation can be loaded into the counter as soon as the error has been counted by the first set of counters. Also, the PWM output from the common counter can be directly understood by the system without further interpretation.

4.4 Overview of the Contrast between a Digital PID and an ADPID

In a digital PID control, the encoded signal feedback from the system is converted into usable form for PC/DSP, where the digital PID controller is programmed into. For instance, the encoded output needs to be first translated into analog voltage, in order to be compared with the reference voltage. Then this signal is manipulated by digital PID to produce a control signal. This is illustrated in Figure 4.3.

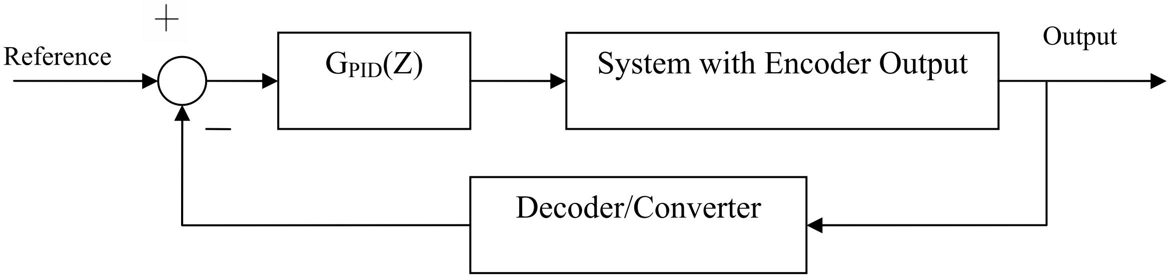

Figure 4.3: Digital PID controller in an encoded system

In contrast, an ADPID control can directly utilize the unaltered and raw data from the output of an encoder as a feedback signal, by means of all digital logic devices. After defining a pulse train to represent the desired output of the encoder, an error signal can be formed by comparing the pulse train with the encoded system output. This is shown in Figure 4.4.

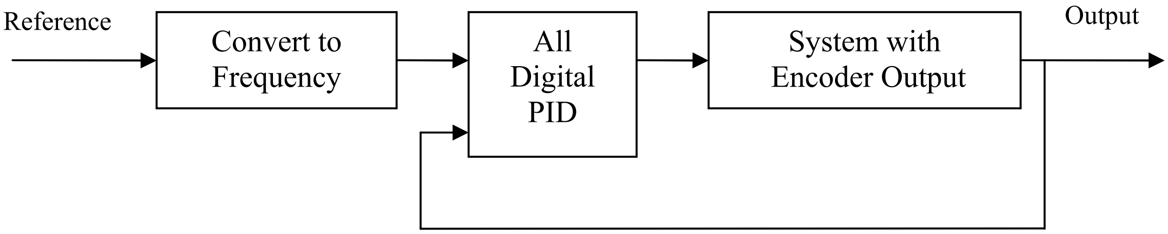

Figure 4.4: All-digital PID controller in an encoded system

4.5 Design of an ADPID

To design an ADPID requires the knowledge to generate a proportional digital signal of the error, an integral digital signal for the errors over time, and a derivative digital signal for the change in error. These three separate signals need to be combined together to form a PWM control signal.

An ADPID is mainly constructed by digital logic devices, such as 4-bit up/down counters, JK flip-flop, D flip-flop, multiplexer, and digital gates (*i.e.* AND, OR, EXOR…). In general, the error between the reference and encoded output is counted up or down by counters. The speed of counting depends on frequencies f_P, f_I and f_D. Then, the summation of the counters is counted to zero at frequency f_A. The control signal generated by an ADPID is in PWM form, and the corresponding direction is determined by the most significant bit of the adders.

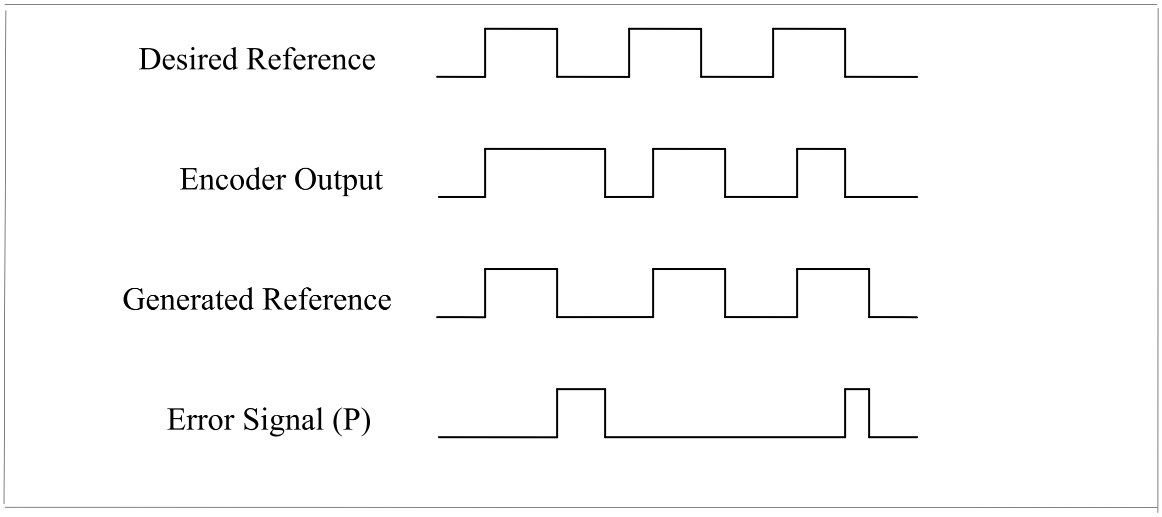

Figure 4.5: Generation of an error signal

4.5.1 Error Signal (P)

I. Generated reference

Generated reference is a set of pulse trains that represents the desired output of the encoder. The resolution of the encoder used in the system determines the frequency of the pulses. For example, if a linear encoder represents 1 Volt with one hundred and fifty pulses, the frequency of the reference will be the desired voltage multiplied by one hundred fifty. The conversion from analog voltage to digital pulse train can be done by crystal, 555 Timer, or a Voltage-Controlled-Oscillator (VCO).

II. Encoded output

Encoded output is the raw system output transformed into digital pulses by an encoder. Among the many encoders, encoder strips and encoder disks are commonly installed in systems.

From Figure 4.5, error is generated when the “high” or “low” of the reference signal and the encoded output do not match. An EXOR is the simplest device to implement this task. Table 4.1 shows the truth table of an EXOR.

Table 4.1: Truth table of an EXOR

Generated Reference	Encoded Output	Result
0	0	0
0	1	1
1	0	1
1	1	0

Whenever the generated reference and the encoded output differ, the output from EXOR is “1”. On the other hand, when both generated reference and encoded output are the same, EXOR outputs a “0”. This implies that the error is zero.

4.5.2 Error Directional Signal (D)

An error directional signal is necessary to indicate the counting direction of K_P, K_I and K_D counters. This is the direction for the error generated by EXOR. One way to implement this task is to compare the desired voltage with the analog output from the system. When the desired voltage is greater than the system output, the counters are set to count up mode. The counters switch to count down mode when the system output overshoots the desired voltage. Since this is an analog way of implementation, the comparator can be done by an operational amplifier.

A digital way to implement the task is to have two sets of counters with the same counting frequency. Each counter is responsible to count the width of the pulses generated by reference and encoded output respectively. The counter with the higher count indicates that the corresponding signal is longer and thus, it is slower than the other. Thus, the direction of the K_P, K_I and K_D counters can be switched accordingly.

4.5.3 ADPID Signal

An all-digital PID signal is composed of:

1. P = Proportional error signal
2. I = Integral error signal
3. D = Derivative error signal

Each K_P, K_I and K_D term needs to be first designed by either root locus or frequency response methods, explained in Chapter 3. Then these terms are translated into frequency format to be used by an ADPID.

From Figure 4.2, P, I and D have their own distinct counting frequencies, f_P, f_I and f_D respectively. The three separate signals are combined together, and the result is fed into one counter that counts at base frequency f_A.

In this thesis, the determination of the frequencies is made primarily based on experimental experience. A few simulations were run, and it was concluded that the optimum frequency for the largest gain among the terms K_P, K_I or K_D is fifty to one hundred times larger than the reference signal. After the first counting frequency is known, the combined frequency, f_A, can be calculated. Then the other two frequencies with lower gain can be obtained.

4.5.3.1 Base Frequency

In order to vary the weighting on each proportional, integral and derivative term, the frequencies of the counters are carried so that:

$$K_Z = \frac{f_Z}{f_A}, Z = P, I, D$$

Equation 4.4

The base frequency, f_A, is the counting frequency for the combined counters. It is directly related to f_P, f_I and f_D. It can be determined after any one of the proportional, integral or derivative frequencies is known. Technically, low frequencies cause inaccurate counting, whereas extremely high counting frequencies create problems such as integrator windup. As a result, more hardware is needed to prevent counting overflow. Hence it is crucial to design a proper counting frequency, to obtain an accurate model yet reduce the hardware to the minimum.

4.5.3.2 Proportional Error Signal

The proportional counter is implemented by SN74LS169, counting at frequency f_P. In order to vary the weighting on the proportional terms, the frequencies of the counters are carried such that:

$$K_P = \frac{f_P}{f_A}$$

Equation 4.5

K_P is the proportional term designed for PID controllers from Chapter 3. If K_P is the largest gain among the three, f_P can be estimated as one hundred times larger than the reference frequency. Thus, f_A is known. If K_P is not the largest gain, f_A is calculated through the other terms before f_P is known. Inputs involve in the proportional counter (C_P) are:

I. up/down ($U/\overline{D}$)

   This input signal is the error directional signal explained in Section 4.5.2. This signal controls the counting direction for the counter. When the error directional signal is in high state (D=1), the counter counts up. When the error directional signal is in low state (D=0), the counter counts down.

II. clock (CLK)

   This input signal is proportional to frequency (f_P). This frequency controls the counting speed for the proportional counter.

III. inputs (A, B, C, D)

A, B, C and D load data from 2-line to 1-line data selectors/multiplexers, SN74LS157. The multiplexer selects either 1 or -1 based upon the directional error signal. If the directional error signal is in low state, the multiplexer chooses -1 and starts counting downwards. Likewise, when the directional error signal is in high state, the multiplexer chooses 1 and count upwards. The counting starts from 1 or -1 instead of 0 (the default value for a counter), because the counting should take place once the error is generated.

IV. count enable($\overline{ENP}\, and\ \overline{ENT}$)

The first counter's count enable is activated ($\overline{ENP}\, and\ \overline{ENT} = 0$) when the error signal is in high state (P=1) and when the JK flip-flop's clear is not zero. Otherwise the proportional counter does not count.

V. load ($\overline{LOAD}$)

Load is activated ($\overline{LOAD} = 0$) for 1 clock cycle when the error signal is in high state (P=1) and when JK flip-flop's clear is not equal to zero. Otherwise the counter does not load any values.

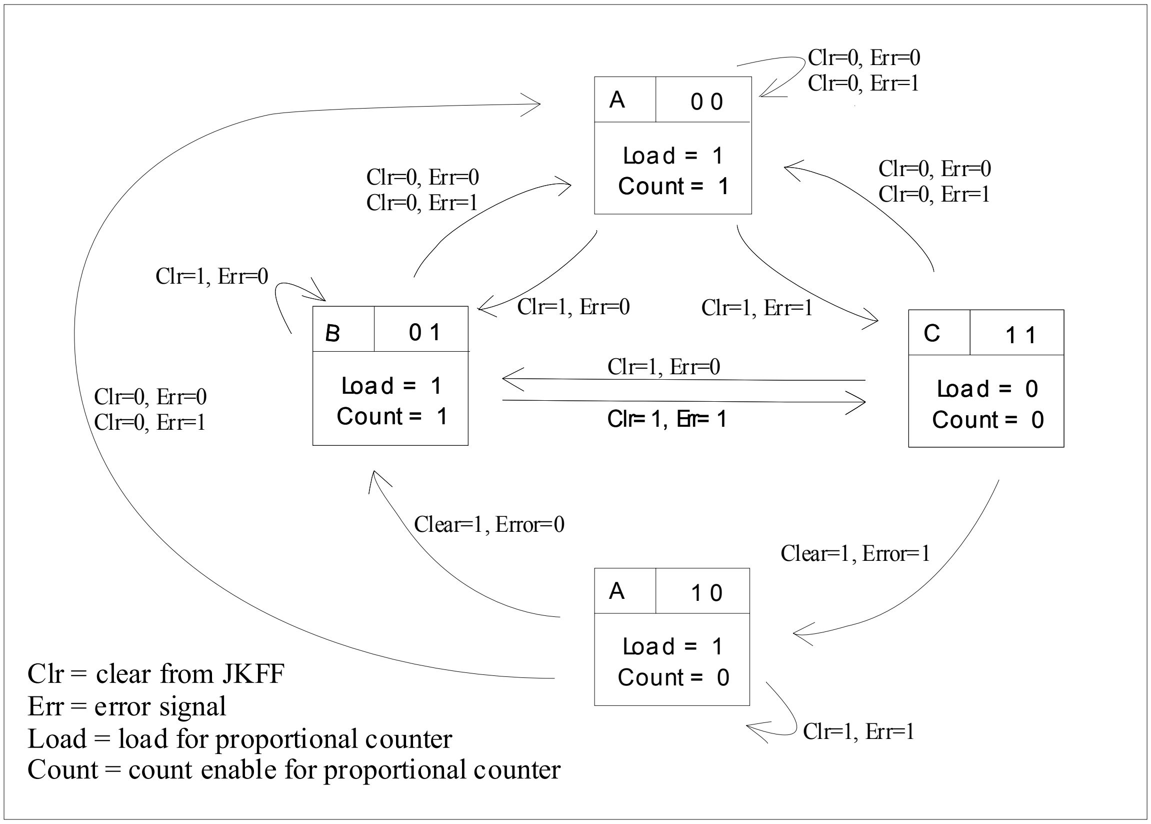

Figure 4.6: State diagram to control count enable and load signals for a proportional counter

If the error signal is in high state (P=1) and the directional signal is also in the high state (D=1), the proportional counter (C_P) counts up at frequency f_P. If the error signal is in high state (P=1) and the directional signal is in the low state (D=0), C_P counts down at frequency f_P. If the error signal is in low state (P=0), C_P stops counting and holds the current values. C_P resets and starts counting from zero when the next error signal is in high state (P=1). These are concluded in Table 4.2.

Table 4.2: Summary of proportional term counting sequence

Error Signal (P)	Directional Error Signal (D)	Reaction on proportional counter (C_P)
0	0	Stop counting. Resume from 0 when error signal (P) = 1
0	1	Stop counting. Resume from 0 when error signal (P) = 1
1	0	Reset C_P. Count Down @ f_P
1	1	Reset C_P. Count Up @ f_P

4.5.3.3 Integral Error Signal

The integral counter is implemented by SN74LS169, counting at frequency f_I. In order to vary the weighting on the integral terms, the frequencies of the counters are carried such that:

$$K_I = \frac{f_I}{f_A} \qquad \text{Equation 4.6}$$

K_I is the integral term designed for PID controllers. From Section 4.3.3.1, if K_I is the largest gain among the three, f_I is one hundred times larger than the reference frequency. Thus, f_A is known. If K_I is not the largest gain, f_A is calculated from other terms before f_I is known. Inputs involved in the integral counter (C_I) are:

I. up/down ($U/\overline{D}$)

This input signal is the error directional signal explained in Section 4.5.2. This signal controls the counting direction for the counter. When the error directional signal is in high state (D=1), the counter counts up. When the error directional signal is in low state (D=0), the counter counts down.

II. clock (CLK)

This input signal is integral frequency (f_I). This frequency controls the counting speed for the integral counter.

III. inputs (A, B, C, D)

A, B, C and D load data from 2-line to 1-line data selectors/multiplexers, SN74LS157, during the very first error. The multiplexer selects either 1 or -1 based upon the directional error signal. If the directional error signal is in low state, the multiplexer chooses -1 and starts counting downwards. Likewise, when the directional error signal is in high state, the multiplexer chooses 1 and counts upwards. Counting starts from 1 or -1 instead of 0 (the default value for a counter) because the counting should take place once the error is generated.

IV. count enable($\overline{ENP}\,and\,\overline{ENT}$)

The first counter's count enable is activated ($\overline{ENP}\,and\,\overline{ENT} = 0$) when the error signal is in high state (P=1) and JK flip-flop's clear is not zero. Otherwise the integral counter does not count.

V. load ($\overline{LOAD}$)

Load is activated ($\overline{LOAD} = 0$) for 1 clock cycle at the first appearance of an error. The counter does not load until the clear of the JK flip flop goes low.

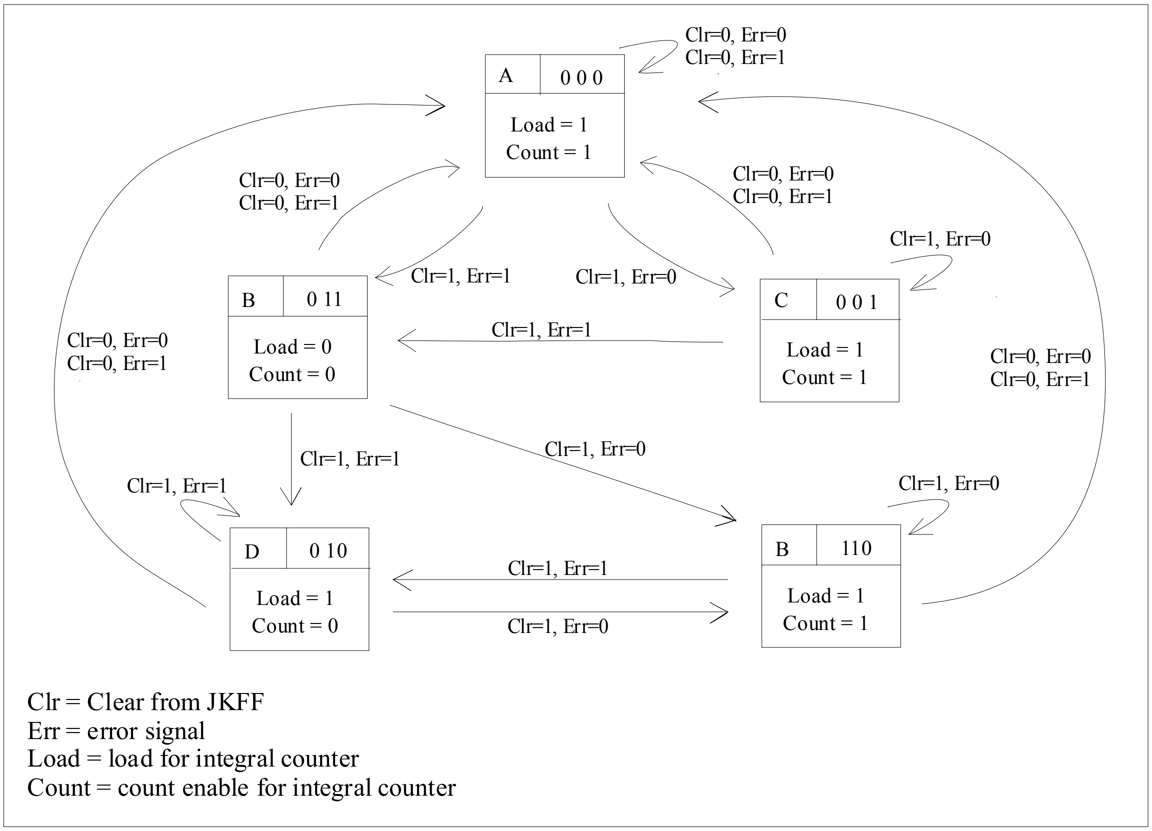

Figure 4.7: State diagram to control count enable and load signals for an integral counter

If the error signal is in high state (P=1) and the directional signal is in the high state (D=1), the integral counter (C_I) counts up at frequency f_I. If the error signal is in high state (P=1) and the directional signal is in low state (D=0), C_I counts down at frequency f_I. If the error signal is in low state (P=0), C_I stops counting and holds its current value. C_I continues counting from previous states when the next error signal is in high state (P=1). These are concluded in Table 4.3.

Table 4.3: Summary of integral term counting sequence

Error Signal (P)	Directional Error Signal (D)	Reaction on integral counter (C_I)
0	0	Stop counting & hold at current value. Continue from previous states when error signal (P) = 1
0	1	Stop counting & hold at current value. Continue from previous states when error signal (P) = 1
1	0	Count Down @ f_I
1	1	Count Up @ f_I

4.5.3.4 Derivative Error Signal

The derivative counter is implemented by SN74LS169, counting at frequency f_D. In order to vary the weighting on the derivative terms, the frequencies of the counters are carried such that:

$$K_D = \frac{f_D}{f_A} \qquad \text{Equation 4.7}$$

K_D is the derivative term designed for PID controllers. From Section 4.3.3.1, if K_D is the largest gain among the three, f_D is one hundred times larger than the reference frequency. Thus, f_A is known. If K_D is not the largest gain, f_A is calculated from other terms before f_D is known. Inputs involve in the derivative counter (C_D) are:

I. up/down ($U/\overline{D}$)

This input signal is the error directional signal explained in Section 4.5.2. This signal controls the counting direction for the counter. When the error directional signal is in high state (D=1), the counter counts up. When the error directional signal is in low state (D=0), the counter counts down.

II. clock (CLK)

This input signal is the derivative frequency (f_D). This frequency controls the counting speed for the derivative counter.

III. inputs (A, B, C, D)

A, B, C and D load data from 2-line to 1-line data selectors/multiplexers, SN74LS157. The multiplexer selects either 1 or -1 based upon the up/down directional error signal. If the directional error signal is in low state (D=0), the multiplexer chooses -1 and starts counting from there. Likewise, when the directional error signal is in high state (D=1), the multiplexer chooses 1. The counting starts from 1 or -1 instead of 0 (the default value for a counter) because the counting should start taking place as the load signal is activated.

IV. count enable ($\overline{ENP}\, and\, \overline{ENT}$)

Count is activated ($\overline{ENP}\, and\, \overline{ENT}$ = 0) when error signal is in high state (P=1) and when JK flip-flop's clear is not zero. Else, derivative counter does not count.

V. load ($\overline{LOAD}$)

Load is activated ($\overline{LOAD}$ = 0) for 1 clock cycle when the error signal is in high state (P=1) and when JK flip-flop's clear is not equal to zero. Otherwise the counter loads no values.

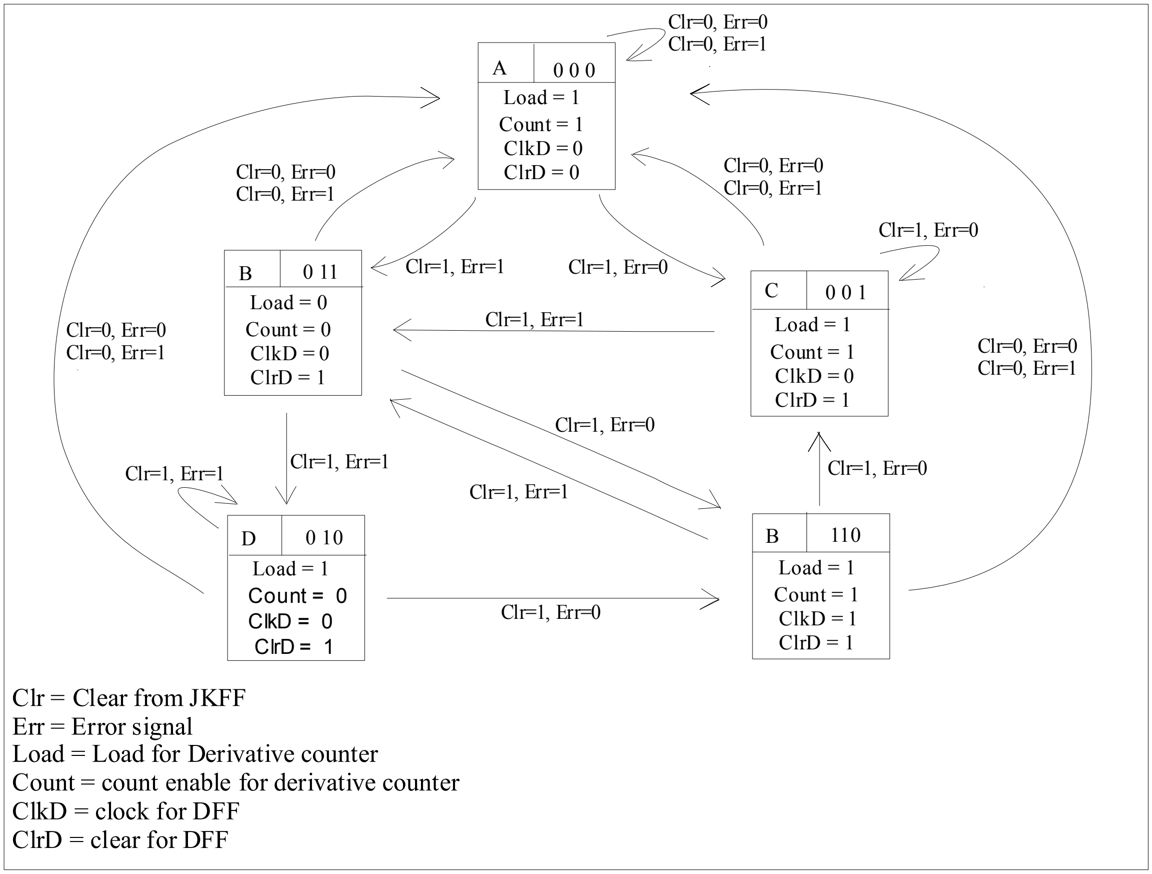

Figure 4.8: State diagram to control count enable and load signals for a derivative counter

If the error signal is in high state (P=1) and the directional signal is in the high state (D=1), the derivative counter (C_D) counts up at frequency f_D. If the error signal is in high state (P=1) and the directional signal is in low state (D=0), C_D counts down at frequency f_D. As the error signal transitions from high to low, the state of C_D is subtracted from the current state of DFF, C_D-R. C_D-R is loaded into the adder, while the current state of C_D is loaded and stored into DFF. C_D resets and starts counting from zero when the next error signal is in high state (P=1). These are concluded in Table 4.4.

Table 4.4: Summary of derivative term counting sequence

Error Signal (P)	Directional Error Signal (D)	Reaction on derivative counter (C_D)
0	0	Stop counting. Subtract C_D from current state of DFF, C_D-R. Load C_D-R into adder. Load and store current state of C_D into DFF. Resume counter from zero when error signal (P) = 1
0	1	Stop counting. Subtract C_D from current state of DFF, C_D-R. Load C_D-R into adder. Load and store current state of C_D into DFF. Resume counter from zero when error signal (P) = 1
1	0	Reset C_D. Count Down @ f_D
1	1	Reset C_D. Count Up @ f_D

4.5.3.5 Combination of the Three Separate Terms (P, I, D)

From Figure 4.2, the separate signals from the proportional, integral and derivative counters are added together by the digital adder, SN74283. The summation of the three is loaded into a combined counter, C_A. Then, the digital counter SN74LS169 counts them to zero at base frequency, f_A. Inputs involved in combined counter (C_A) are:

I. up/down ($U / \overline{D}$)

This input signal depends on the result from the adder. If the most significant bit (MSB) of the adder is “1”, (meaning that the summation of the three terms is negative), the count down counter (C_A) counts up to zero. On the other hand, if the most significant bit (MSB) of the adder is “0”, (meaning that the summation of the three terms is positive), the count down counter (C_A) counts down to zero. Hence, the magnitude for the combination of the three signals is derived.

II. clock (CLK)

This input signal is the base frequency (f_A). This frequency controls the counting speed for the combined counter.

III. inputs (A, B, C, D)

A, B, C and D load data from 2-line to 1-line data selectors/multiplexers, SN74LS157. The multiplexer selects either “0” or outputs from adder. As the counter is not initialized during the first cycle of the error, the multiplexer selects “0” so that the input for PWM can be initialized. As the error counting is finished, the inputs from (A, B, C, D) are loaded for counting. Figure 4.8 shows the activity of the multiplexer select line.

IV. count enable ($\overline{ENP}\, and\, \overline{ENT}$)

Count is activated ($\overline{ENP}\, and\, \overline{ENT} = 0$) when the output of the combined counter is not zero, so counting continues regardless of the error state. Whenever the counter finishes counting (*i.e.* output = 0), this signal is deactivated.

V. load ($\overline{LOAD}$)

Load is activated ($\overline{LOAD} = 0$) for 1 clock cycle when the error signal transitions from high state (P=1) to low state (P=0). It is also activated during the first error, where the counter is not yet initialized. The state diagram in Figure 4.9 shows the activation of the load signal.

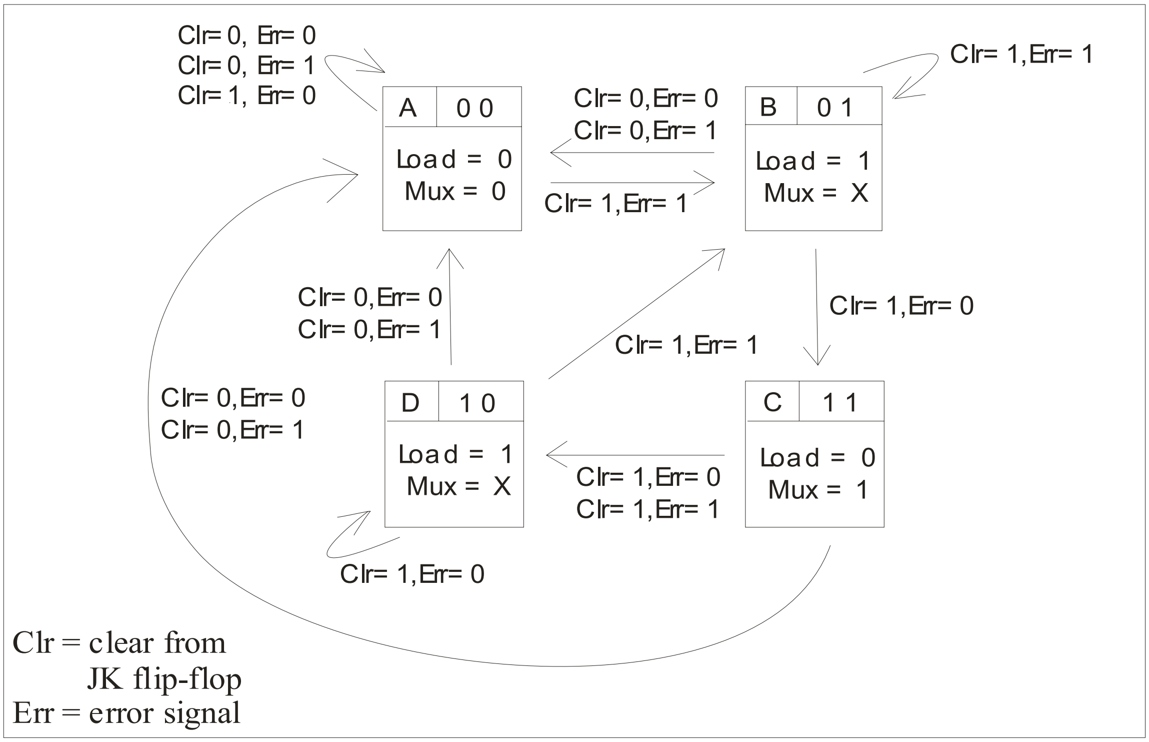

Figure 4.9: State diagram to control multiplexer select line and load signal for a combined counter

I. Adder:

The states of the proportional counter (C_P), integral counter (C_I) and derivative counter (C_D) are added simultaneously by the adder. Since the adder is an asynchronous device, it performs counting when the three counters change state.

II. Combined counters:

As the error signal transitions from high to low, the count-down counter (C_A) loads the results from the adder (C_P + C_I + (C_D-R)). The counting direction depends on the result from the adder. If the most significant bit (MSB) of the adder is "1", the summation of the three terms is negative; thus the combined counter will count up at frequency f_A to zero. On the other hand, if the MSB of the adder is "0", the summation of the three terms is positive; then the combined counter counts down at frequency f_A to zero. However, the counting sequence is interrupted when load is activated. In this case, the combined counters discontinue the previous counting and pick up the new loaded number. The summary of the process to combine the three signals is listed in Table 4.5.

Table 4.5: Summary of combine counters counting sequences

Error Signal (P)	MSB from Adder	Reaction on combine counter
0	0	Load the summation from the adders. Count down @ f_A until zero is reached
0	1	Load the summation from the adders. Count up @ f_A until zero is reached
1	0	Continue count down if zero is not reached
1	1	Continue count up if zero is not reached

4.5.3.6 PWM Amplifying Gain

The output of the combined counters is fed into an OR gate to generate the magnitude of the PWM control signal. The direction of PWM is determined by the MSB of the adder.

The PWM amplifying gain is derived by first obtaining the highest control voltage in an analog simulation; this voltage is divided by the magnitude of the PWM. The final result is the amplifying gain for the PWM signal.

4.5.3.7 Theoretical Analog and All-Digital PID Signal Comparison

A theoretical signal comparison was made between analog and all-digital PIDs. Each proportional, integral and derivative term for both analog and all-digital was tested and compared in pair. To begin with, an arbitrary set of voltage and gain were first programmed into each term. For the all-digital approach, the arbitrary voltage was changed to frequency format to represent the error signal. The same parameters of voltage and gain were applied to an analog simulation. The theoretical results were compared.

First, a test was run on an all-digital proportional term. Both integral and derivative gains were set to zero. The counters had proportional gain, K_P=1.5, and input voltage = 1V (150 Hz). In Figure 4.10(a), the lower signal shows the PWM corresponding to the error. The upper signal represents the reference waveform of (voltage=1*gain=1). The ratio of the period of PWM to the period of the reference waveform throughout the simulation represents the amplifying gain produced by the proportional counters. In this example, the ratio is 1.5. This result is shown in Figure 4.10(b). Figure 4.10(c) shows the analog version of proportional signal with (voltage=1*gain=1).

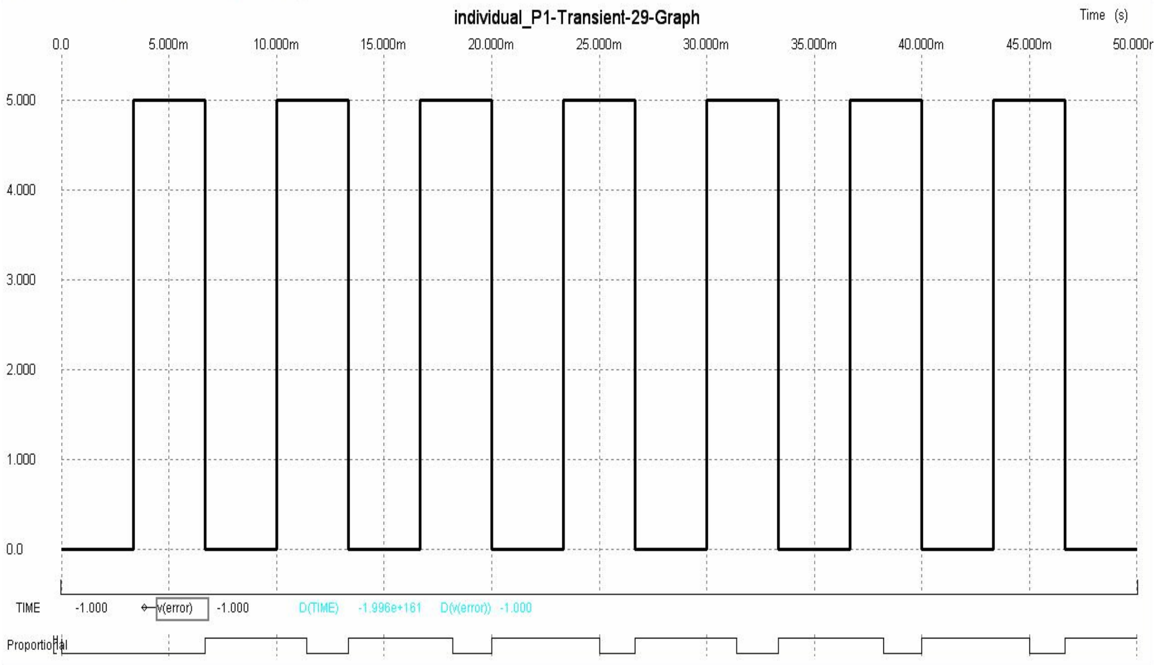

Figure 4.10(a): PWM of a proportional signal

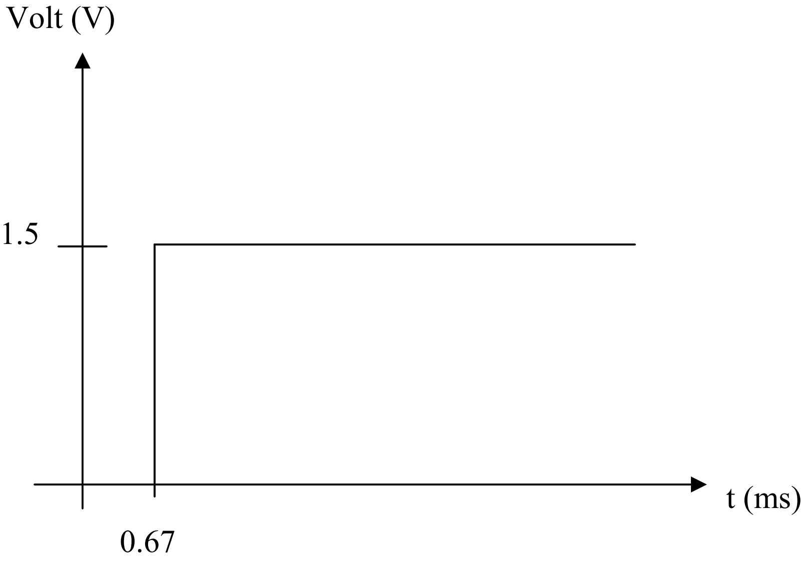

Figure 4.10(b): All-digital simulation of a proportional signal

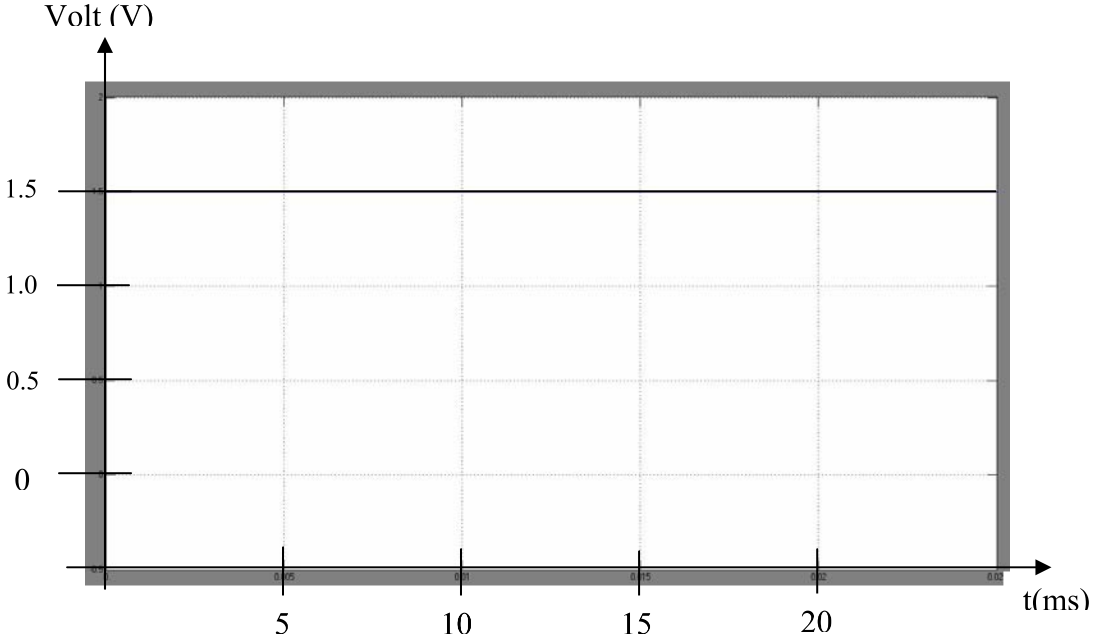

Figure 4.10(c): Analog method of proportional gain multiplied by input voltage

Next, a simulation was run on the integral term to both all-digital and analog PIDs. Both proportional and derivative gains were set to zero. The integral gain and the constant input voltage were set to 1. The upper signal shown in Figure 4.11(a) reflects the reference waveform of (voltage=1*gain=1), while the lower signal represents the PWM signal for the integral term. By finding the ratio of the PWM signal to the reference waveform, Figure 4.11(b) is obtained. The result shown in Figure 4.11(b) differs from Figure 4.11(c) because the all-digital integral counters are saturated after the second cycle. The integral counters are not able to finish counting the previous error, while the next error is already loaded into the counters.

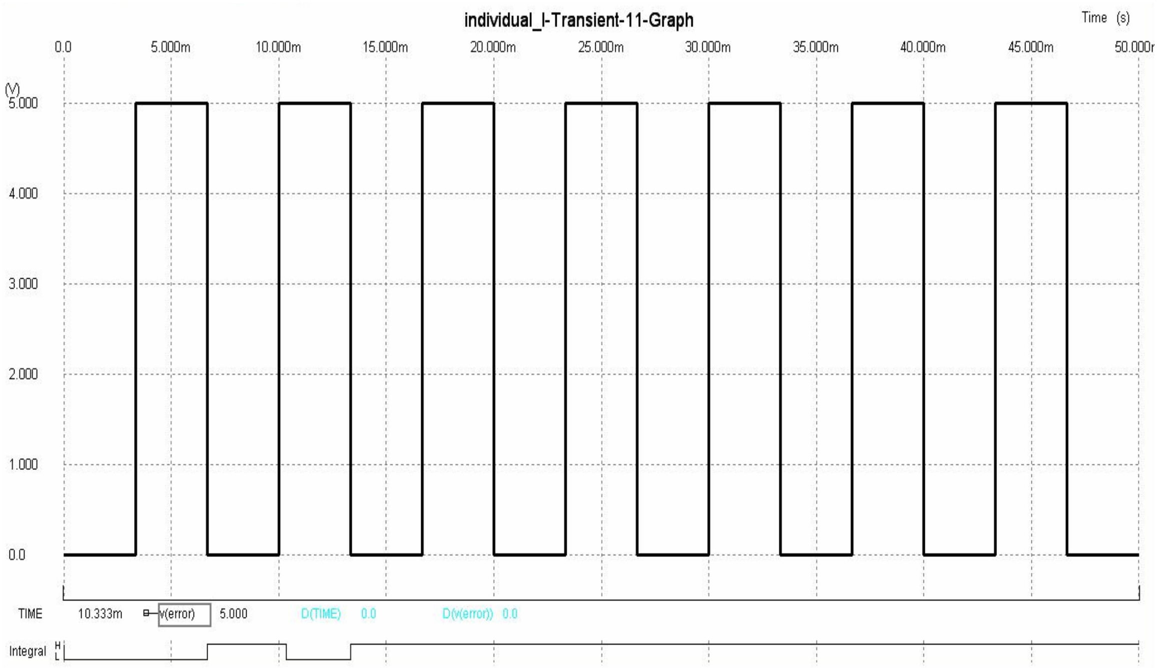

Figure 4.11(a): PWM of an integral signal

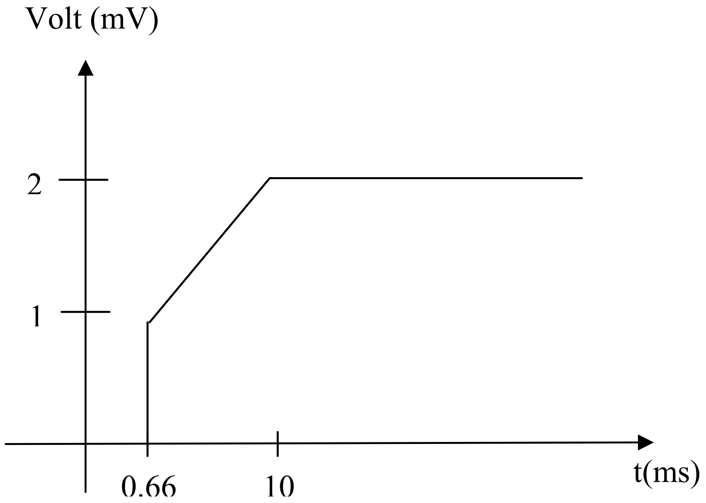

Figure 4.11(b): All-digital simulation of an integral signal

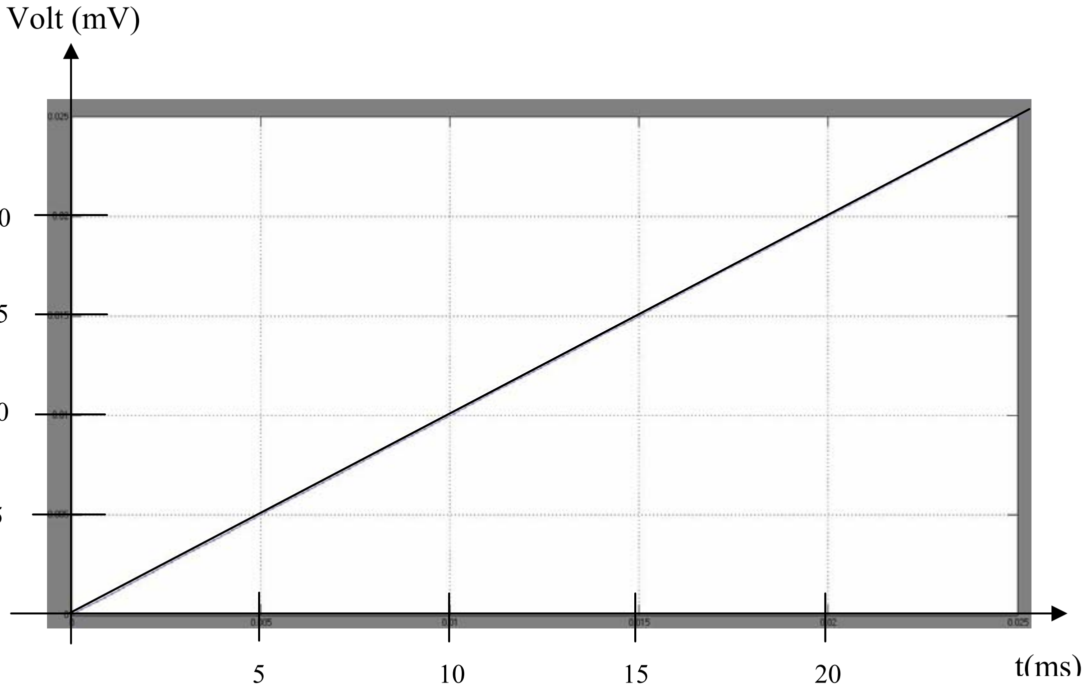

Figure 4.11(c): Analog method of integral gain multiplied by input voltage

Then, a simulation was done on the derivative term on both analog and all-digital PIDs. Both proportional and integral gains were set to zero. The derivative gain was set to 1, and the input was a ramp with slope equal to 1. The upper signal in Figure 4.12(a) shows the reference waveform of (ramp input with slope=1*gain=1), while the lower signal in the same plot is the PWM for the derivative term. As the derivative term detects the change of direction of the system, the ratio of the PWM signal to the reference waveform can be found by first subtracting the current error period from the previous error period. Then the PWM signal is divided by the reference waveform. The all-digital result displayed in Figure 4.12(b) is consistent with the analog result in Figure 4.12(c).

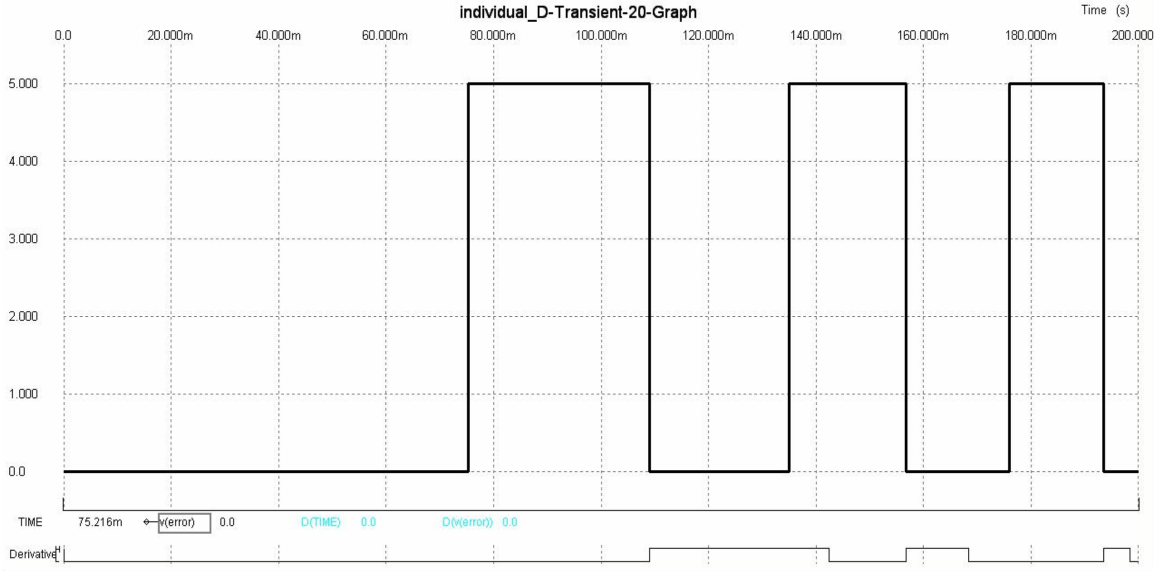

Figure 4.12(a): PWM for a derivative signal

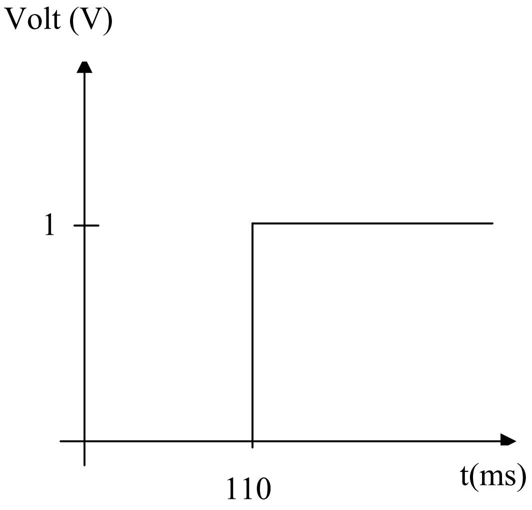

Figure 4.12(b): All-digital simulation of a derivative signal

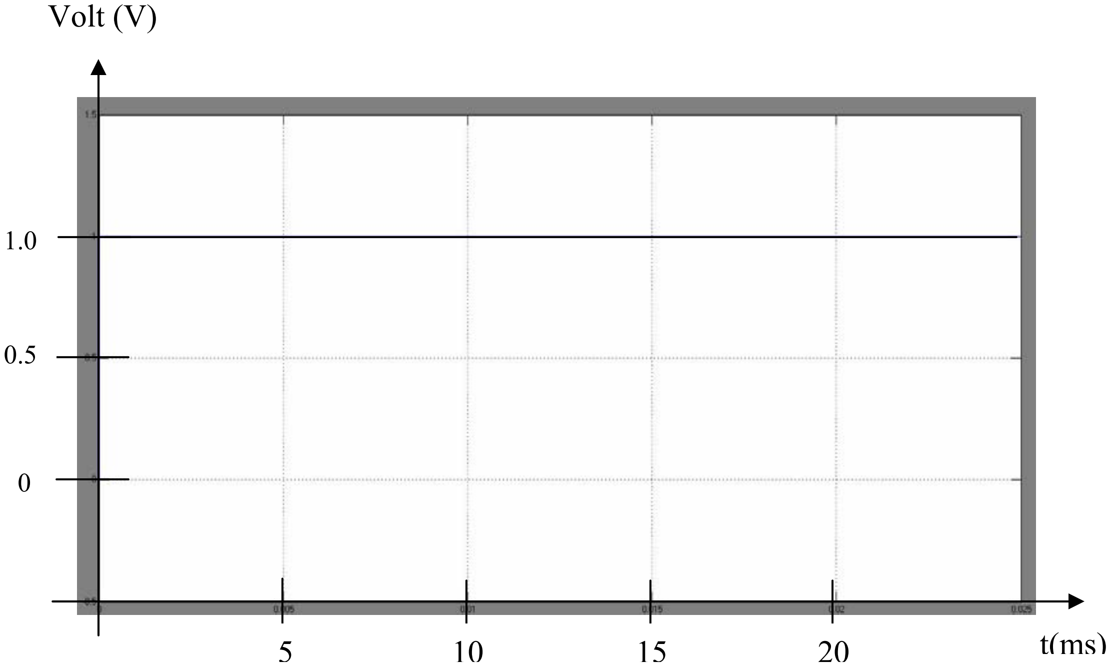

Figure 4.12(c): Analog method of integral gain multiplied by input voltage

From Figures 4.10(b), 4.11(b) and 4.12(b), the outputs do not start from zero milliseconds. This is because the digital pulse train is idle at the beginning of the simulation. A PWM signal is only generated each time the error counting is finished. Hence, the first rising edge of the all-digital simulation can only respond after the first error.

CHAPTER 5

ADPID SIMULATION AND RESULTS

5.1 Introduction

In the previous chapter, the structure and the design of an ADPID were introduced. In Chapter 5, a case study based on the ADPID theory stated in Chapter 4 is tested. Specifically, an ADPID controller is designed and analytically tested on a Lexmark Z-52 inkjet printer. The goal of the design is to control the system track and stabilize at 1 Volt within 0.16 second.

The transfer function of the printer is derived, and the proper PID gains are designed by the Root Locus method. Next, the gains are transformed into appropriate proportional, integral and derivative frequencies. Then the ADPID simulation is compared with the analog simulation.

5.2 Case Study - Inkjet Printer Carriage Motion Control

The carriage motion model was analytically obtained from [22], [23]. The author approximated the mechanism of the printer by using the pseudo-random binary sequences (PRBS) identification technique. In this model, the belt dynamic is assumed to be stiff, and thus is ignored.

5.2.1 Transfer Function for the Inkjet Printer Carriage Motion System

Figure 5.1 represents an approximate model of the cartridge transport mechanism. The parameters of the symbols used in Figure 5.1 are listed in Table 5.1.

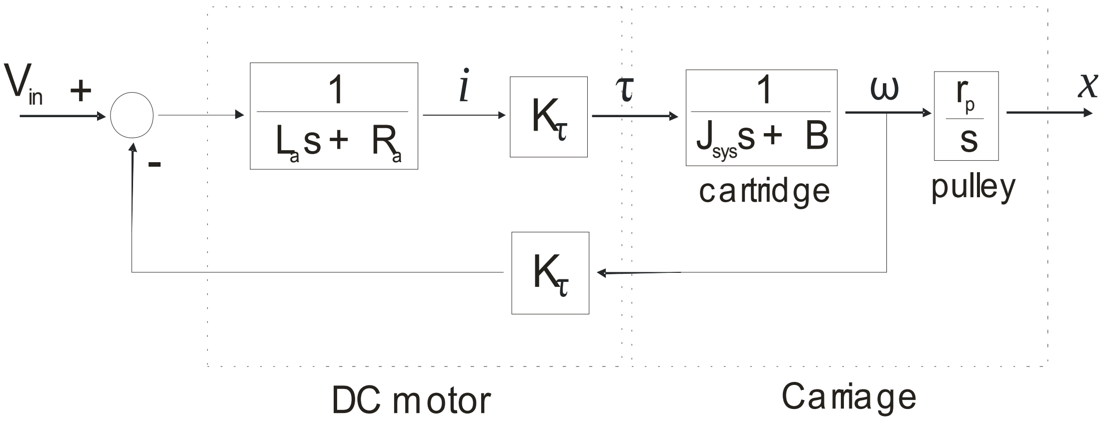

Figure 5.1: Analytical model of a cartridge transport mechanism [21]

Table 5.1: Parameters for a cartridge transport mechanism [21]

Symbol	Parameter	Value
J_{sys}	Effective moment of inertia	$3.973\text{x}10^{-6}$kg·m^2
B	Viscous damping	$1.890\text{x}10^{-4}$N·m·sec
R_a	Motor coil resistance	13.6Ω
L_a	Motor coil inductance	8.16mH
K_τ	Torque constant	0.0452N·m/A
r_p	Motor pulley radius	0.0038m
i	Actual current input to carriage DC motor	
ω	Angular velocity of carriage armature	
x	Carriage position	

From the transport mechanism model shown in Figure 5.1, the transfer function between angular velocity and input voltage for the cartridge transport system is given by

$$\frac{\omega(s)}{V_{in}(s)} = \frac{K_\tau}{L_a s(J_{sys}s + B) + R_a(J_{sys}s + B) + K_\tau^{\,2}} \qquad \text{Equation 5.1}$$

In Equation 5.1, the term K_τ^2 is due to back-emf, and is relatively small compared to other terms. Thus, it is eliminated from the equation. By substituting the values of each parameter from Table 5.1 into Equation 5.1,

$$\frac{\omega(s)}{V_{in}(s)} = \frac{0.0452}{(s + 47.579)(s + 1666.64)} \quad \text{(m·Kg·Sec system of units)} \qquad \text{Equation 5.2}$$

The poles in Equation 5.2 are 0, -47.579, and -1666.64 on the s-plane. The poles that are located closest to the jω-axis on the s-plane will dominate the response, since they have longer settling times than the others. From Equation 5.2, the dominant poles are 0 and -47.579. The least dominant pole can be eliminated by dividing the numerator by 1666.64. Hence the closed-loop transfer function between angular velocity and input voltage for this cartridge transport system is approximated as

$$\frac{\omega(s)}{V_{in}(s)} = \frac{42.94}{s + 47.579} \quad \text{(Inch·g·Sec system of units)} \qquad \text{Equation 5.3}$$

Also, the open-loop transfer function between position and input voltage for the cartridge transport system is given by

$$\frac{x(s)}{V_{in}(s)} = \omega(s) * \frac{r_p}{s} = \frac{6.49}{s(s + 47.579)} \quad \text{(Inch·g·Sec system of units)} \qquad \text{Equation 5.4}$$

5.2.2 PID Controller for the Inkjet Printer Carriage Motion System

The goal of the controller in this project is to track 150 pulses within 0.16 second with an overshoot of less than 12%. By using the root locus method explained in Chapter 3, P, I and D gains for the system are:

$$K_P = 300$$

$$K_I = 0.2$$

$$K_D = 0.2 \qquad \text{Equation 5.5}$$

Thus, the closed loop PID compensated system transfer function becomes

$$\frac{1.298s^2 + 1947s + 1.298}{s^3 + 47.58s^2} \qquad \text{Equation 5.6}$$

A root locus plot for the PID compensated system is shown in Figure 5.2. The step response of the uncompensated system and the PID compensated system are displayed in Figure 5.3.

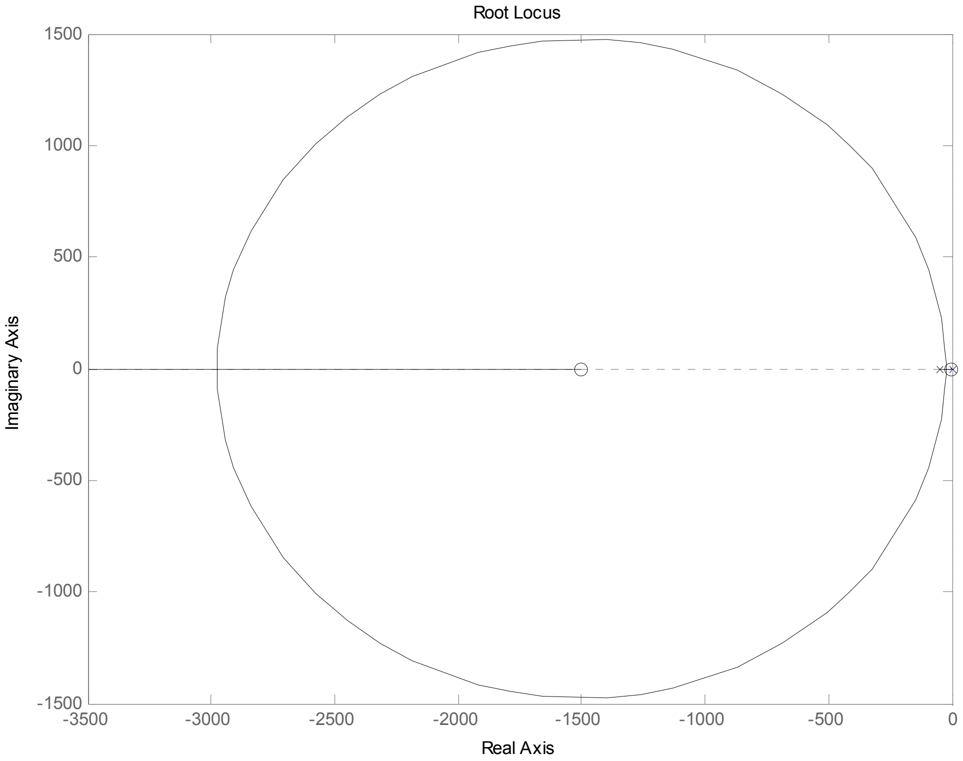

Figure 5.2: Root Locus for a PID compensated system

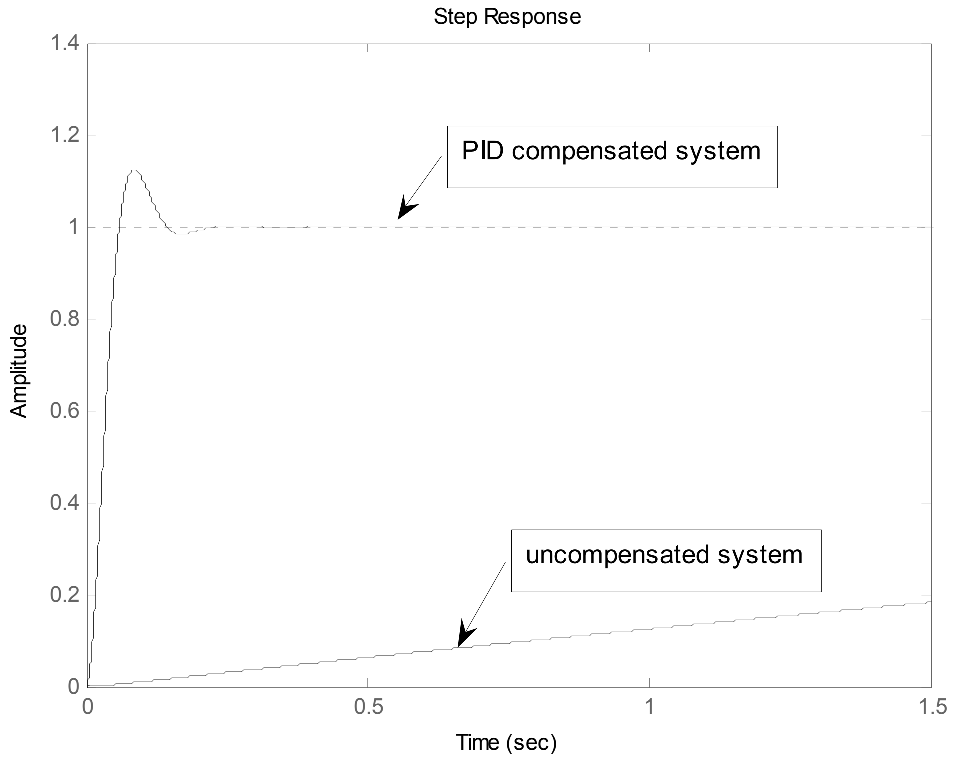

Figure 5.3: Step responses for an uncompensated system and a PID compensated system

5.2.3 Simulation and Results on a PID Compensated System Based on B2 Spice

A PID compensated system was tested on B2 Spice software. Simulink was first considered to implement this simulation task, as it is more suitable and more generally used in system control analysis. However, it was found that the software lacks digital components such as up/down counters. Also, to modify an existing counter was not an option, as the script files are all hidden. Thus, B2 Spice was chosen as a replacement for Simulink.

The simulation was divided into analog and digital categories. The results from analog and all-digital are compared.

5.2.3.1 Analog Simulation

In the analog simulation, the PID controller is programmed as an analog transfer function. The reference voltage and system output voltage are both converted into digital pulses. Then, EXOR compares these two pulses to generate an error signal. In the simulation, switches perform as an H-bridge, as H-bridge is not available in the B2 Spice software. When the output voltage of the system is greater than the reference voltage, the switch selects the negative direction of the PWM control signal, and *vice versa*. Figure 5.4 shows the results of the analog PID tracking 1 Volt for 250 ms.

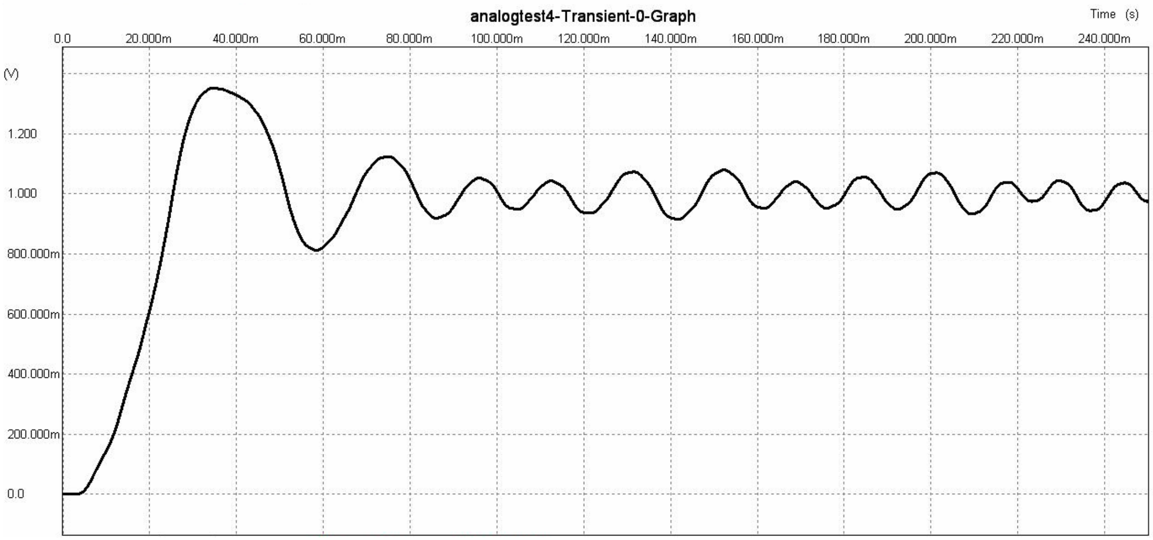

Figure 5.4: Analog simulation tracking 1 Volt for 250 ms

5.2.3.2 All-Digital Simulation

In the all-digital simulation, proportional, integral and derivative terms are translated into f_P, f_I, f_D and f_A. From Equation 5.3, K_P, K_I and K_D are factored out by one hundred because this software consumes intensive computer memory. Later, this factored one hundred is multiplied into the system transfer function. This guarantees that the overall closed loop transfer function before and after the modification are the same. The modified K_P, K_I and K_D, and the system transfer function, are shown in Equations 5.7 and 5.8 respectively.

$$K_P = 3$$
$$K_I = 0.002$$
$$K_D = 0.002$$
Equation 5.6

$$transfer\ \ function = \frac{649}{s(s + 47.579)}$$
Equation 5.7

The magnitude of the PWM is 5V. The amplifier found to amplify the control signal is three. Hence, the final system transfer function for the digital simulation is:

$$system\ transfer\ \ function = \frac{1947}{s(s + 47.579)}$$
Equation 5.8

The largest gain found in Equation 5.6 is K_P. As the system in this simulation is tracking 1 Volt, which is equivalent to 150 pulses-per-inch, the frequency of the reference signal is 150 Hz. Thus, the frequency of f_P is:

$$f_P = 100 \times 150 = 15000\ Hz$$
$$f_A = 15000 \div 3 = 5000\ Hz$$
Equation 5.9

By substituting Equation 5.9 into Equations 4.6 and 4.7, f_I, f_D are found

$$\begin{aligned} f_I &= 0.002 \times 5000 = 10\ Hz \\ f_D &= 0.002 \times 5000 = 10\ Hz \end{aligned} \tag{Equation 5.10}$$

Figure 5.5 is the plot of all-digital PID tracking 1 Volt based on the frequencies found in Equations 5.9 and 5.10.

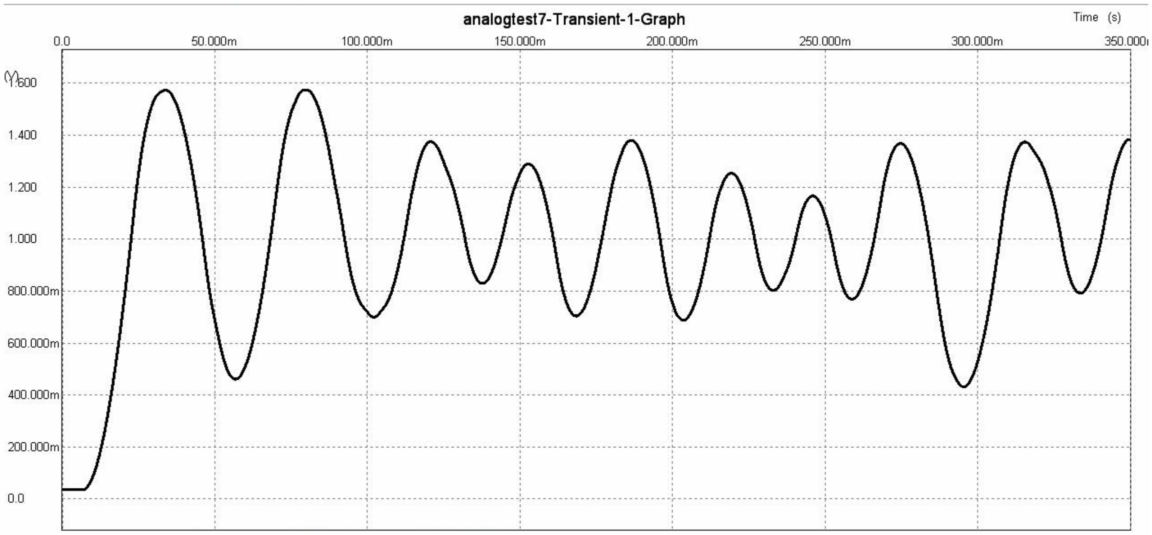

Figure 5.5: All-digital simulation tracking 1 Volt for 350 ms

5.2.3.3 Analog and All-Digital PID Simulation Comparison

Figure 5.5 shows that the ADPID is tracking 1 Volt with approximately ±0.4 Volt of fluctuation. The graph is not as smooth as the analog simulation. In an ADPID, proportional, integral and derivative counters count the error at various frequencies. The three individual signals are added together to generate one summation signal that is counted to zero at the speed of the combined frequency. It would be ideal if the combined counters manage to count the summation to zero before the next error comes. However, this does not always happen because the length of the error differs from time to time. If the summation is large and the next error is ready to be loaded into the combined counter, the counting sequence for the previous error is terminated, and the counter starts a new counting sequence for the new error. As a result, the model becomes less accurate as the settling time is longer and peak voltage is higher.

Another problem encountered during the simulation was that the digital counter SN74LS169 does not work exactly as planned in Chapter 4. In general, a synchronous counter counts when the count enable is low at the rising edge of the clock. In Chapter 4, however, the conditions for proportional, integral and derivative counters to count merely depend on the rising edge of the error signal. This means that regardless of the clock, all the counters should start counting at the very beginning of the error. However, during the digital simulation, the clock is an independent source. As any clock frequencies among proportional, integral and derivative counters are relatively small compared to the generated error frequency, the rising edges of the clock miss the occurrence of the error. As a result, the counters do not count even though the count enable is low.

Figure 5.6 shows the generated error during the first 100 ms of the simulation. Figure 5.7 shows the clock frequency of integral and derivative counters for 100 ms. From Equation 5.10, integral and derivative counters at 10 Hz count only one time within 100 ms. Thus, all the generated error lying between the rising edges of the clock are missed.

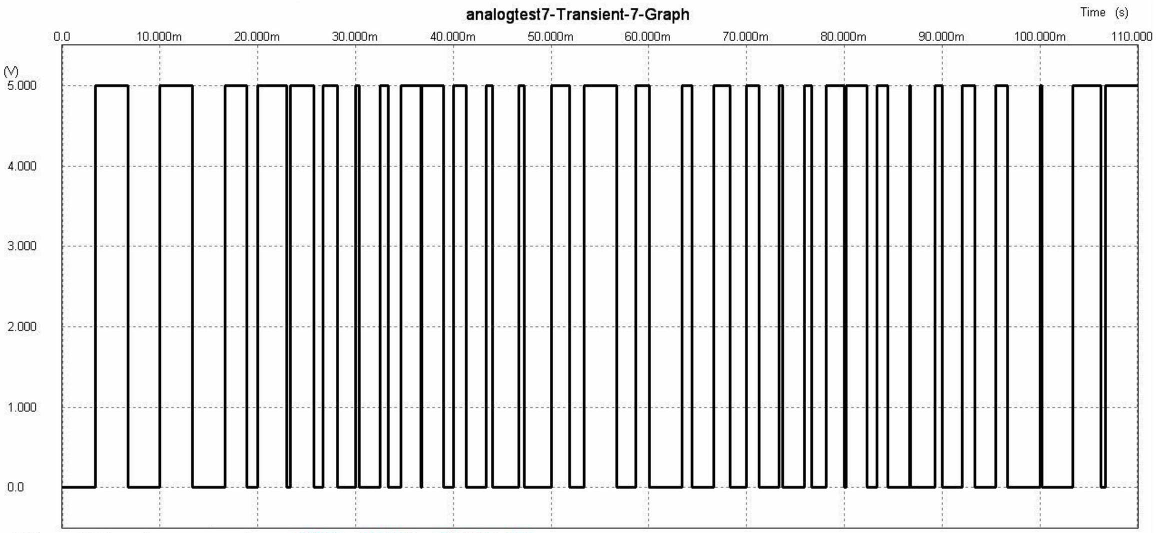

Figure 5.6: Generated error during the simulation

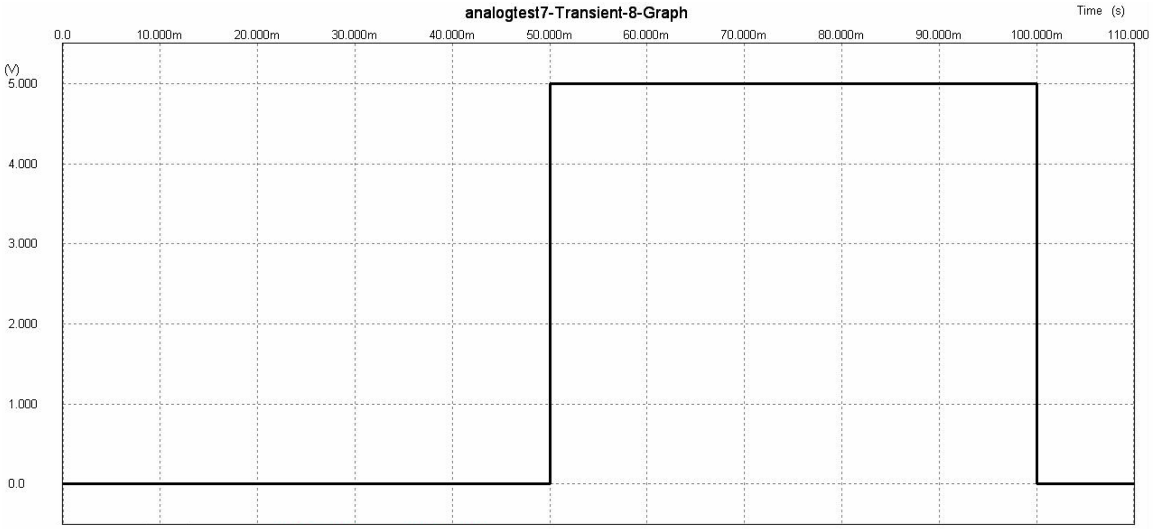

Figure 5.7: Counting frequency for integral and derivative counters at 10 Hz

The accuracy can be improved by replacing the digital counter SN74LS169 with an asynchronous counter. An asynchronous counter's clock rising edge depends on the occurrence of the error as well as the counting frequency. The clock in the all-digital simulation has a rising edge which occurs consistently throughout the simulation. By replacing the synchronous with the asynchronous counter, the counting will start at the rising edge of the generated error. Technically, the rising edge of the clock will align with the rising edge of the generated error. This will solve the problem caused by the synchronous counter.

Also, the all-digital result can be improved by reversing Equation 4.4 to

$$K_Z = \frac{f_A}{f_Z}, Z = P, I, D \qquad \text{Equation 5.11}$$

The new Equation 5.11 provides a large combined frequency f_A with respect to f_Z when the gain is larger than one. From the digital simulation result, there is a problem when the combined counters have insufficient time to count the error to zero. According to Equation 4.4, when the gain is larger than one, f_A is smaller than f_Z. As a result, the time for the combined counters to count the summation to zero will be much longer than the time spent to count the error. By substituting Equation 4.4 with Equation 5.11, the combined counters timing issue can be improved such that the combined counters are now able to count more aggressively than before.

As the combined counters timing issue is improved, both Equations 4.4 and 5.11 agree that the model will be more efficient when the frequencies of all the terms are increased proportionally. As the frequencies increase, the error can be counted more accurately. However, this implementation will cost more in hardware in order to prevent the counters from overflowing.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

The all-digital implementation of a PID controller explores an option in addition to analog and digital PID controllers. The concept of an ADPID stems from the use of PWM, where the controller's proportional, integral and derivative actions are converted into pulses by means of standard up-down digital counters and other digital logic devices. The final product of a controller will be realized in digital programmable logic devices such as FPGAs and CPLDs.

Professors Bruce Walcott and Michael Marra from the University of Kentucky were the pioneers of ADPID controllers. In 1998, the authors introduced an ADPID as a replacement to analog and digital PIDs. The concept was filed in a patent application by the two professors, called "Asynchronous Digital Implementation of PID controllers". In the patent application, the authors contrasted the ADPID to the traditional PID, and listed the requirements to implement this sort of controller. The introduction of ADPID eliminates both ADC and DAC, which reduces cost, error and delay to the system.

In an ADPID, each proportional, integral and derivative term is represented by a pair of frequencies, where the ratio of the frequencies equals the gain of each term. These frequencies determine the counting speeds of the counters. After defining a pulse train to represent the desired output of the encoder, an error signal is formed and processed by the ADPID. The resulting ADPID output or control signal is in PWM format, and can be fed directly into the target system without ADC.

Each term in the PID consists of two sets of counters, C_{Z1} and C_{Z2}, where z represents P, I or D. The weightings on the P, I and D terms are varied by the frequencies of the counters. For instance, the weighting on the proportional term is

$$K_P = \frac{f_{P1}}{f_{P2}} \qquad \text{Equation 6.1}$$

C_{P1} counts at a frequency of f_{P1} when the error signal is in the high state. The counting direction depends solely on the error directional signal. When the voltage of a system output exceeds the voltage of a desired reference, the counters are set to count down mode, and *vice versa*. As the error signal transitions from high to low, C_{P1} stops counting and resets the counters to zero. C_{P2} loads the current state of C_{P1} and counts the state at the frequency of f_{P2} to zero.

In order to represent an integrating action, the weighting on the integral term is

$$K_I = \frac{f_{I1}}{f_{I2}} \qquad \text{Equation 6.2}$$

When the error signal is in the high state, counter C_{I1} counts at the frequency of f_{I1}. Similarly, the counting direction depends on the error directional signal. As the error signal transitions from high to low, C_{I1} stops counting and holds at the current state. C_{I2} loads the current state of C_{I1} and counts the state at the frequency of f_{I2} to zero. The C_{I1} counters resume from the previous state if the error is in high state.

The weighting on the derivative term is represented as

$$K_D = \frac{f_{D1}}{f_{D2}} \qquad \text{Equation 6.3}$$

When the error signal is in the high state, C_{D1} counts at the frequency of f_{D1}. The counting direction depends on the error directional signal. As the error signal transitions from high to low, C_{D1} stops counting and subtracts the present state from the current state of the register, D flip-flop. The value after the subtraction is loaded into C_{D2}. The D flip-flop is refreshed with the current state of C_{D1}. Then, C_{D1} resets the counter. C_{D2} counts the state at the frequency of f_{D2} to zero.

In the patent application, the authors did not mention a procedure to combine the three separate P, I and D signals. The generation of a PWM control signal and the corresponding direction were not stated. There is no detail as to how the error and its directional signal can be determined. Also, no guideline for designing the counting frequencies was explained. Hence, this thesis includes step-by-step "rule of thumb" modifications to complete the picture of an ADPID.

In the thesis, an error signal is generated by comparing the digital-pulsed reference and the encoded system output with an EXOR gate. The direction of an error signal is sensed by an operational amplifier. The separate proportional, derivative and integral signals are summed together after their first set of counting. Specifically, asynchronous adders are placed after C_{P1}, C_{I1} and C_{D1}, and combine the signals when C_{P1}, C_{I1} and C_{D1} change state. At the falling edge of an error signal, the summation is transferred to combined counters, which count at frequency f_A. The output from the combined counters goes through an OR gate, becoming a PWM control signal. The corresponding direction is determined by the MSB of the adders. Thus, by implementing the ADPID with four counting frequencies, f_P, f_I, f_D and f_A, cost and hardware are reduced.

The modified expression for the counting frequency thus becomes

$$K_Z = \frac{f_Z}{f_A}, Z = P, I, D \qquad \text{Equation 6.4}$$

From experimental experience, the counting frequencies are calculated such that the highest gain among proportional, integral and derivative signals has a frequency range of fifty to one hundred times greater than the frequency of the reference signal. Then, f_A and the other two frequencies with lower gain can be found by substituting the appropriate terms into Equation 6.4.

The amplifying gain is necessary to amplify the PWM signal. The gain can be found by first measuring the maximum control voltage generated in an analog version of simulation. Then, this voltage is divided by the magnitude of the PWM signal, emanating from the OR gate of the ADPID. The final result is the amplifying gain for the ADPID, which is placed after the OR gate, or before the PWM is fed into the system.

The complete procedure to implement an ADPID is summarized as:

1. Find P, I and D gains, using one of the methods explained in Chapter 3.
2. Obtain the resolution of the encoder. Find the desired frequency corresponding to the reference signal.
3. Multiply the highest gain among P, I and D to the range of fifty to one hundred times the reference frequency.

4. Find the combined frequency, f_A, from Equation 6.4.
5. Find the other two lower gains' counting frequencies from Equation 6.4.
6. Perform an analog version of the simulation to measure the largest control voltage.
7. Divide the voltage obtained in Step 6 by the magnitude of the PWM signal. This result is the amplifying gain for the ADPID.

Such an ADPID was tested analytically on a Lexmark Z-52 inkjet printer by B2 Spice software. The controller is designed to track and stabilize the system to 1 Volt within 0.16 second. An approximate model of the cartridge transport mechanism for the printer was derived. Using the root locus method, the proportional, integral and derivative gains were determined to be 300, 0.2 and 0.2 respectively. As the resolution of the encoder is 1 Volt, equivalent to 150 pulses per inch, the frequency of the reference signal thus becomes 150 Hz. The largest gain is the proportional signal; the frequency of f_P is calculated to be 15 KHz. f_A, f_I and f_D were found to be 5 KHz, 10 Hz and 10 Hz respectively.

Both analog and all-digital simulation and results were compared and analyzed. Analog simulation shows a peak at around 38 ms, and stabilizes in 80 ms with a ripple of ±0.05 V, while digital simulation ripples steadily at ±0.4 V.

In an ADPID, proportional, integral and derivative signals are combined and counted to zero at frequency f_A by a combined counter. It would be ideal if the combined counters can finish counting before the next error enters; however, this is not always possible, because the length of the errors differs from time to time. If the summation is large and the next error is ready to be loaded into the combined counter, the counting sequence for the previous error will be terminated in order for the counter to start a new counting sequence. As a result, the PWM does not reflect the real control signal for the previous error.

Also, the unwise selection of digital counter SN74LS169 conflicts with the counter proposed in Chapter 4, which are asynchronous rather than synchronous. A synchronous counter counts when its count enable is low at the rising edge of the clock, whereas an asynchronous counter merely depends on the rising edge of its count enable. In the all-digital simulation, the clock in the synchronous counter works as an independent source. As the clock frequencies among proportional, integral and derivative counters are relatively small compared to the generated error frequency, the rising edges of the clock miss the occurrence of the error. As a result, the counters do not count when the count enable is low.

A few attempts have been made to systematize the design of the counting frequencies. One was based on the concept of finding the settling time through bandwidth frequency. This is done by first getting a Bode plot of an open loop compensated system, and then finding the bandwidth frequency at which the magnitude is -3dB. The unit of the frequency is converted from rad/sec to Hz. The time constant is the reciprocal of the frequency. By approximating settling time equals four times of time constant, a frequency corresponding to the settling time can be achieved. In order to obtain a more accurate counting sequence, the settling frequency is increased on the order of fifty to one hundred times greater than the original frequency. This is the combined frequency, f_A.

However after deep consideration, this method is found not always to work, especially when the system tends to have a small time constant. The combined frequency thus becomes small, which effectively reduces the proportional, integral and derivative frequencies. Eventually, the counters will not count accurately because of the low counting frequencies.

Another attempt was to set the combined frequency equal to the reference frequency. Then, proportional, integral and derivative frequencies can be calculated through Equation 6.4. However, this method proves inappropriate because the counting sequence presents poor error estimation when gains are small. This attempt is similar to the method used throughout the thesis when the gain is one hundred.

In addition to previous attempts, a different counters counting sequence was also considered. Instead of using the second complement adding method, the first counting starts from the middle of an n-bit counter. For example, a 4-bit counter starts counting from seven, instead of zero. Ultimately this method was rejected because the counters overflow when the errors are continuously in the counting up direction. Also, the work of adding the three signals can be more complicated when the counters are switching the counting direction.

In this thesis, an ADPID is tested on B2 Spice software. Simulink was first considered to implement the simulation, as it is generally used to analyze system controls. However, Simulink lacks digital components such as up/down counters; in addition, it is impossible to modify an existing counter because script files are not available. Thus, B2 Spice was chosen as a replacement for Simulink.

6.2 Future Work

The error directional signal in this design is implemented by an analog device, an operational amplifier, working as a voltage comparator to compare the desired voltage with the analog output from the system. Since the objective of this ADPID is to implement the controller all-digitally, a digital method should replace this analog implementation, so that the controller can be absolutely programmed into PLDs.

The simulation presented in Chapter 5 shows that an ADPID should be implemented asynchronously rather than synchronously. The rising edge of the clock should depend first on the occurrence of the error, then on the counting frequency. Technically, the rising edge of the clock is aligned with the rising edge of the generated error. Then, the counting rate is determined by the frequencies designed for each of the proportional, integral and derivative counters (*i.e* f_P, f_I, f_D). This helps the counters avoid missing counting the errors.

Also, the simulation result can be improved by rearranging the counting frequencies to

$$K_Z = \frac{f_A}{f_Z}, Z = P, I, D \qquad \text{Equation 6.5}$$

When the gain is larger than one, Equation 6.5 provides a large f_A with respect to f_Z. By letting the combined counters count more aggressively, the insufficient counting time issue caused by the combined counters can be improved.

In this thesis, the method for designing counting frequencies is determined by experimental experience. A mathematical model is yet to be designed, so that ADPIDs can be implemented more systematically.

Even though B2 Spice software was chosen to design and implement the ADPID, some deficiencies were discovered throughout the simulation. For instance, B2 Spice does not provide any H-bridge. The hidden resistance in the components causes voltage fluctuation in the circuit. Also, initially counter SN74LS169 did not work properly, as described in the Texas Instrument TTL data book. This problem was fixed months later, after this issue had been reported many times to Beige Bag Software, Inc. Due to all the inconveniences associated with the software, a more suitable platform should be considered in order to test the ADPID more effectively.

Last but not least, the ADPID should be programmed into a digital PLD such as a FPGA chip. Then it should be utilized on a real application, so that its real world performance characteristics can be gauged.

REFERENCES

[1] Rozsa, L. “Design and Implementation of Practical Digital PID Controllers”, IFAC Proceedings Series, n 15, 1990, pp. 115-121.

[2] Bibbero, R.J. “Microprocessors in Instruments and Control”, New York, NY: Wiley, 1977.

[3] Fassois, S. “Microprocessors and their applications to control systems”, Diploma Thesis, Control Systems Division, National Technical University, Athens, Grece, 1982.

[4] Parnell, K., Bryner, R., “Comparing and Contrasting FPGA and Microprocessor System Design and Development”, White Paper 213, Vol. 1, No. 1, July 21 2004. http://direct.xilinx.com/bvdocs/whitepapres/wp213.pdf

[5] Chen, C.Y., Chen, C.C. and Chiu G.T.-C. “Adaptive Robust Control of Media Advance Systems for Thermal Inkjet Printers”, Mechatronics, v 10, 2000, p111-126.

[6] Stepper Motor Basics, Industrial Circuits Application Note, http://www.ericsson.se/mocroe/apn_ind.html

[7] Roppenecker, G. and O’reilly, J. “Parametric Output Feedback Controller Design”, *Automatica*, 1989.

[8] Bennett, Stuart. “The Past of PID controllers,”, Annual Review in Control, v 25, 2001, p 43-53.

[9] VanDoren, Vance. “Understanding PID control”, *Control Engineering*, June, 2000, p 53-56.

[10] VanDoren, Vance. “PID: Still the One”, *Control Engineering*, December, 2003, p 28-29.

[11] DSIR. “Automation.” Her majesty’s Stationery Office, London. 1956.

[12] Williams, G. P. L. “Trends of development in the British instruments industry”, *Instrument Engineer*, 1957.

[13] Turner, A. “Computer in process control: a panel discussion”, *Instruments & Control Systems*, 1970.

[14] Auslander, D. M., Takahashi, Y. and Tomizuka, M. “The next generation of single loop controllers”, Journal of Dynamic Systems, Measurements, and Control. 1975.

[15] Krikelis J. N. and Fassois D. S. “Microprocessor Implementation of PID controllers and Lead-Lag Compensators.” IEEE Transaction on Industrial Electronics, 1984.

[16] Hayes, L. “Optimizing Optical Encoder Performace.” Sensors. v 12, April 1995, p 65-68.

[17] Lumkes, J.H. Jr., “Control Strategies for Dynamic Systems.” New York: Marcel Dekker, Inc.

[18] Lin, H.Y., Lu, M.C., Horng, J.H., Yang, T.B., Hsieh. S.C., Hsiao, C.Y., Tai, H.M. and Chen, J.S. “DC Servo Speed Control of an Inkjet Print Head Transport System Using a Phase-Locked Loop.” Proceedings of the 1996 4th International Workshop on Advanced Motion Control.” 1996.

[19] Best, E. “Phase-Locked Loops – Theory, Design, and Applications.” New York: McGraw-Hill. 1993.

[20] Adkins, C. "Velocity Control Of An Inkjet Print Head Transport System Using An All-Digital Phase-Locked Loop." M.S. Thesis, University Of Kentucky. 1997.

[21] Adkins, C.A., Marra, M.A. "Modeling of a phase-locked loop servo controller with encoder feedback." Proceedings IEEE Southeastcon'99. Technology on the Brink of 2000. 1999, p 59-63.

[22] Deshpande, A.V., Kamasak, M., Thoon, K.L, Chiu, C.T.-C., Bouman, C., Allebach, J., Fedigan, S., Schafer, D. and Cole, C. "Design and Implementation of a DSP based Inkjet Printer Motion Control System for Dynamic Print Mode Control." International Conference on Digital Printing Technologies. 2001.

[23] Deshpande, A.V. "Printer System for Dynamic DSP Based Inkjet Print Mode Control." M.S. Thesis, Purdue University. 2001.

Vita

Date of Birth:	10 December 1979
Place of Birth:	Miri, Sarawak, Malaysia
Educational Institutions: Attended and Degree Awarded	University of Kentucky, Lexington, USA Bachelors of Science in Electrical Engineering (December 2002)
Scholastic Honor:	Dean's List Spring 2000 - Spring 2001 Tau Beta Pi member Eta Kappa Nu member