Performance Comparison between LQG and PID Controllers for HDD's Servo Timing Control System

NG YAP HAN

FACULTY OF ENGINEERING
UNIVERSITY OF MALAYA
KUALA LUMPUR

2018

Performance Comparison between LQG and PID Controllers for HDD's Servo Timing Control System

NG YAP HAN

DISSERTATION SUBMITTED IN FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF ENGINEERING

(MECHATRONICS)

FACULTY OF ENGINEERING
UNIVERSITY OF MALAYA
KUALA LUMPUR

2018

UNIVERSITY OF MALAYA

ORIGINAL LITERARY WORK DECLARATION

Name of Candidate: Ng Yap Han

Matric No: KQF160015

Name of Degree: Master of Electrical Engineering

Title of Project Paper/Research Report/Dissertation/Thesis ("this Work"):

"Performance Comparison between LQG and PID Controllers for HDD's Servo Timing Control System"

Field of Study: Mechatronic engineering

I do solemnly and sincerely declare that:

(1) I am the sole author/writer of this Work;
(2) This Work is original;
(3) Any use of any work in which copyright exists was done by way of fair dealing and for permitted purposes and any excerpt or extract from, or reference to or reproduction of any copyright work has been disclosed expressly and sufficiently and the title of the Work and its authorship have been acknowledged in this Work;
(4) I do not have any actual knowledge nor do I ought reasonably to know that the making of this work constitutes an infringement of any copyright work;
(5) I hereby assign all and every rights in the copyright to this Work to the University of Malaya ("UM"), who henceforth shall be owner of the copyright in this Work and that any reproduction or use in any form or by any means whatsoever is prohibited without the written consent of UM having been first had and obtained;
(6) I am fully aware that if in the course of making this Work I have infringed any copyright whether intentionally or otherwise, I may be subject to legal action or any other action as may be determined by UM.

Candidate's Signature Date:

Subscribed and solemnly declared before,

Witness's Signature Date:

Name: Designation

Abstract

The growth of technology in the cloud system namely Artificial Intelligence (AI) and Big Data nowadays has pushed the memory storage capacity to grow rapidly above 10% every year, as IBM has predicted the demand for data scientist will grow 28% by year 2020 [1]. The challenge for HDD maker today is to increase the areal density (the number of bytes of bytes per information to be stored per surface area of the disk), and when the areal density gets higher in the same dimension size on the media, the inch per tracks significantly reduces which in turn becomes a challenge to servo design engineers [19]. The accuracy of control system in HDD needs to be able to reject disturbance efficiently to maintain the least steady-state error. HDD is vastly adopting the technology which uses PID controller to achieve the performance, nevertheless it is believed that there is much room for improvement on the performance using the new method, Linear Quadratic Regulator with Kalman Filter (LQG) [20]. In this thesis, LQG will be analyzed against existing PID control in HDD which sees a **30% improvement on NRO Phase Rejection**.

Abstrak

Perkembangan teknologi dalam sistem 'Cloud' iaitu Kecerdasan Buatan (AI) dan Big Data kini telah meningkatkan kapasiti penyimpanan memori untuk berkembang pesat melebihi 10% setiap tahun, bersamaan dengan ramalam IBM di mana permintaan untuk saintis data akan berkembang 28% menjelang tahun 2020 [1]. Cabaran untuk pembuat HDD hari ini adalah untuk meningkatkan kepadatan kawasan (bilangan bait per bait setiap maklumat yang akan disimpan di dalam setiap kawasan permukaan cakera), dan apabila kepadatan kawasan menjadi lebih tinggi dalam saiz dimensi yang sama pada media, inci setiap trek akan berkurang dengan ketara yang seterusnya akan menjadi cabaran kepada jurutera reka bentuk servo [19]. Ketepatan sistem kawalan dalam HDD perlu menolak gangguan dengan cekap untuk mengekalkan kesilapan paling tipis dalam keadaan mantap. Kebayakan HDD mengamalkan teknologi yang menggunakan pengawal PID untuk mencapai prestasi, tetapi dipercayai bahawa terdapat banyak ruang untuk penambahbaikan prestasi menggunakan kaedah baru, Pengatur Kuasa Linear dengan Penapis Kalman (LQG) [20]. Dalam tesis ini, LQG akan dianalisis terhadap kawalan Proportional-Integral-Derivative (PID) yang sedia ada dalam HDD yang **melihat peningkatan sebanyak 30% pada NRO Phase Rejection.**

Acknowledgement

I would like to take this opportunity to express my heartfelt gratitude to Prof Dr. Yap Hwa Jen as my project supervisor. Prof Dr. Yap Hwa Jen has provided me with a lot of support, knowledge, and freedom to explore things by myself. He has given high commitment and time to ensure the thesis writing goes smooth. It is my great pleasure to be given the honor to be one of his research projects students.

A special thanks to my family members especially my wife who tirelessly takes care of me and my family while I am spending most of my time on my studies and thesis. I would also like to thank my US colleague Dr. Richard Conway for sharing his knowledge on the project.

Last but not least, I would like to thank the people who have directly or indirectly lent a hand on the thesis.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATION

ABS	:	Air Bearing Surface
ADC	:	Analog to Digital Conversion
AGC	:	Automatic Gain Control
ARE	:	Arithmetic Riccati Equation
DSP	:	Digital Signal Processing
ERC	:	Machine to Machine
FFT	:	Fast Fourier Transform
ID	:	Inner Disk
LQ	:	Linear Quadratic
LQE	:	Megabits per second
LQG	:	Linear Quadratic Gausian
LQR	:	Linear Quadratic Regulator
MDW	:	Media Disk Writer
NRRO	:	Non Repeatable Run Out
OD	:	Outer Disk
PCB	:	Printed Circuit Board
PID	:	Proportional Integrator Derivation
PLL	:	Phase Lock Look
PRRO	:	Phase Repeatable Run Out
R/W	:	Read/Write
RPM	:	Revolution Per Minute
RRO	:	Repeatable Run Out
SAM	:	Synchronous Address Marking

SNR	:	Signal to Noise Ratio
SOC	:	System On Chip
TBG	:	Time Base Generator
VCM	:	Voice Coil Magnetic
VCO	:	Voltage Control Oscillatior
ZOH	:	Zero Order Hold

LIST OF APPENDICES

CHAPTER 1: INTRODUCTION

1.1 Introduction

In HDD servo timing system, servo sync mark time plays a key role in determined the spindle motor variation. In spiral copy operational mode, the sync mark signal can be found in servo patterns written on the disk, and the sync mark is stamped on a system-on-chip (SOC) counter that is free running on servo clock [1]. The difference between current timestamp and its preceding timestamp can be used to estimate the spindle motor speed variation between two consecutive servo patterns [1]. The speed error is fed into the timing loop controller to calculate the control output to adjust the clock frequency and compensate the error induce by the spiral copy operational mode. In spiral copy mode, the magnetic head is write signal while seeking at the same time the motion generate drag effect on the media and cause the speed variation [1].

Existing method used Proportional-Integration-Derivative (PID) controller to compensate the speed error induce by the operational mode. The PID controller has few main control components that are the proportional, integral, and derivative elements, it is widely used in feedback control of industrial processes, and some modern applications due to their simple structure, and robustness. PID also gives designer easy way to tune the PID to achieve the desire performance on various systems [2]. The method is tuned the PID values by monitor the system transient response, bandwidth, disturbance rejection, steady sate error, and robustness of the plant. However, PID is poor in dealing various disturbances which need additional filter or compensator to overcome those disturbances, and difficulty to deal with multiple-input and multiple-output systems [3].

On the other hand, a modern optimal control theory method is used as comparison to the PID called Linear-Quadratic-Regulators (LQR). LQR used the Linear Quadratic (LQ) equation approach to minimize/maximize measureable states in cost function to achieve best performance in control [4]. In details LQR minimized the excursion in state trajectories of a system while required minimum controller effort. Kalman Filter is an optimal full state estimator and noise rejection in the system such as measurement noise and process noise. Combined both methods is called "Linear-Quadratic-Gaussian" (LQG). The LQG modified the system characteristic to achieve the performance by changed the feedback gain. The feedback gain can be obtain through "Algebraic Riccati Equations" (ARE). ARE is to calculate the state feedback gains on a chosen set of weighting matrices [5]. These weighting matrices is regulated by the penalties on the deviation in the trajectories of the state variables (x(t)) and control signal (u(t)).

Both PID and LQG controller are implemented to compare the performance during spiral copy operational mode.

1.2 Problem Statement

In spiral copy operational mode, the head is seeked while writing from outer disk to inner disk with high velocity, during seek motion the magnetic head fly height is very close to the media due to air dynamic and Air Bearing Surface (ABS) design on head which created a down force that pushed the head even closer to media. Because the clearance of the head and media plus actuator movement it caused the spindle speed variation occur, the speed difference/error is the root cause of the timing offset written into the disk and especially largest during acceleration. The written-in servo patterns have the timing error which need to be corrected by later processes.

1.3 Objective Research

The objective of this research are stated as below:

1. To analyze the performance between two conventional controllers for servo timing system in HDD during spiral copy mode.
2. To improve the performance of overall 3 sigma Non-Repeatable-Runout (NRRO) phase error.
3. To reduce the cycle time of the process with optimum transient, and settling time during the operational mode.

1.4 Scope Research

The scope of this research project is to focus on analysis the performance of using Linear Quadratic Regulator with Kalman Filter (LQG) against Proportional-Integral-Deviation Controller (PID) servo timing system using single type of HDD. The thesis includes the theory of timing analyze technique on HDD, PID design, and LQG design walkthrough. The analysis and controllers design used Matlab as the platform for all the activities, the thesis excluded firmware coding for HDD but control code will be attached at appendix section.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction of A Hard Disk Drive System

A typical HDD as shown in figure 2.1 [6] consists of components such as spindle motor, Voice Coil Magnet (VCM), flex cable, actuator, E-block, media/disk, read write (R/W) head, printed circuit board (PCB), and more. The empty non-written media/disk center is clamped on the DC brushless 3-phase motor, each media separated with a minimal space by disk spacer. When the HDD is powered on, spindle motor is spun at a fixed Revolution-Per-Minute (RPM) which is predefined by manufacturer and controlled by the firmware. The R/W head is assembled at the edge of the actuator, when HDD in operational modes the head is loaded out from parking area called "Ramp" on to the disk surface with controlled velocity. An actuator is a combination of E-block, suspension and slider. The R/W head positioning servomechanism is controlled by the VCM to move the R/W head to access the data on the disk [16]. The VCM movement is governed by electromagnetic theory to move the actuator and R/W head within a range of angle [16].

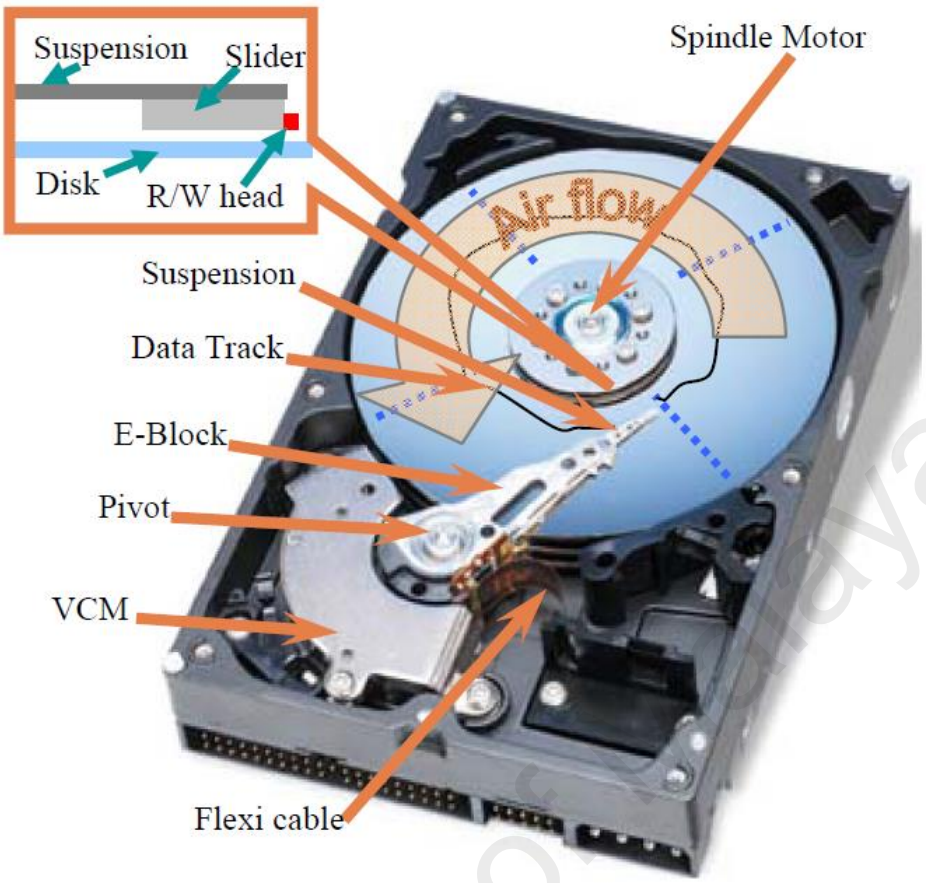


Figure 2.1: A Typical Hard Disk Drive's components diagram

2.2 Spiral Copy Process

What does spiral copy operational mode mean? It is the part of the process of servo writing technique used in HDD industries, and some HDD industries may use different methods [2]. For spiral copy process, shown as the block diagram [figure 2.2], start with the drive with pre-written surface with reference signals using a high precision machine called "Media Writer Machine" (MDW), the reference signals is called "spiral signal" [7] [figure 2.3: blue color spiral pattern].

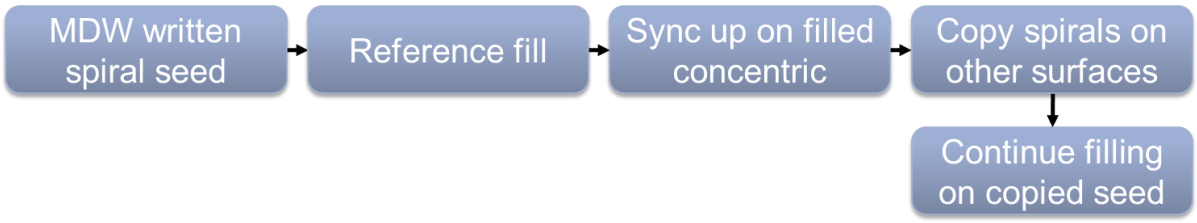


Figure 2.2: Spiral write process sequence block diagram

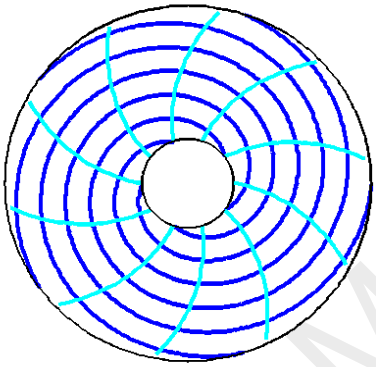


Figure 2.3: Illustration of signals pattern on the HDD's media

Then R/W head positions the head using the spiral signals and starts fill up the surface with servo patterns [figure 2.3: light blue in wedge]. Spiral pattern is a raw reference signal that provides relative position, unlike servo pattern which provides discretize information to the system enable the R/W position precisely without relative to the any start point [15]. The pre-written servo pattern on the surface, R/W head positions using servo sector and performs a process called "spiral copy" process to other blank surfaces or media [21]. In the process R/W head is seeking/sweeping from outer disk (OD) into inner disk (ID) with constant linear velocity [figure 2.4], and write signals is generated to the next set of write head to write the spiral signal [15].

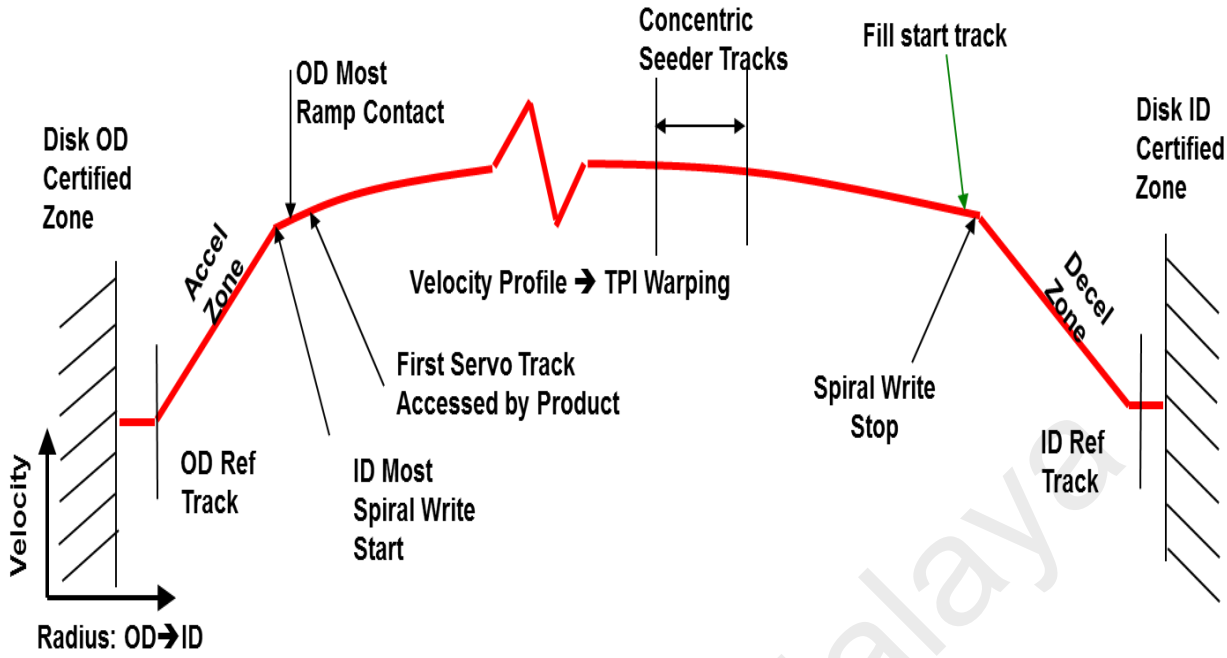
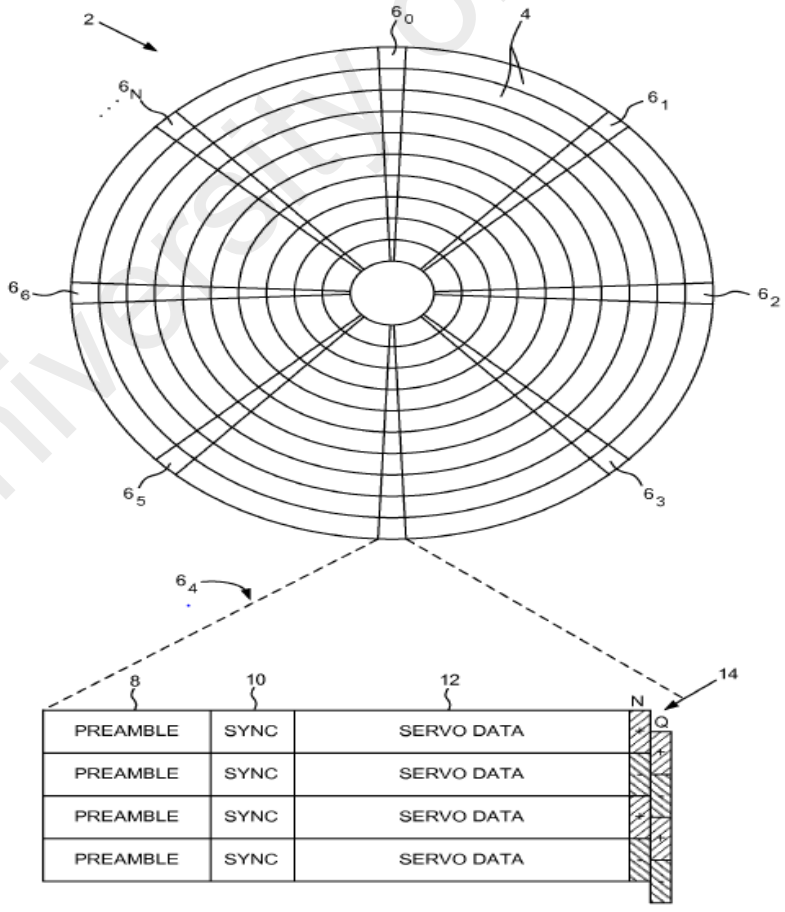


Figure 2.4: Trajectory diagram of spiral pattern written from OD to ID

The performance timing control defines the quality of the written-in phase error. A poor timing control is not robust and sensitive to disturbance, it yields poor written-in phase error on adjacent surface. The result cause poor quality or yield loss issue on next process. The spiral copy process repeated on each surface until all available surfaces fill up with servo patterns [15].

2.3 Phase Error Signal

In order to maintain the R/W head on the same position/radius, the patterns designated by manufacturer called “servo patterns” [8] [figure 2.5] on specific area on disk is decoded, and signals fed into the SOC for control algorithm to position the head. In servo pattern it contains the signal information such as AGC (Automatic Gain Control), Sync Mark (Synchronous Marking), Wedge Identification, Track Number, and Position Signals [8]. Each of the information plays very important role in HDD servo control system. The servo controls in HDD are “Position” and “Timing”. Position controller is to control the location of R/W head. Timing controller is for SOC chip synchronize with disk. Except controllers, HDD also contains signal-to-noise ratio (SNR) control for Digital Signal Processing (DSP) on the user data.



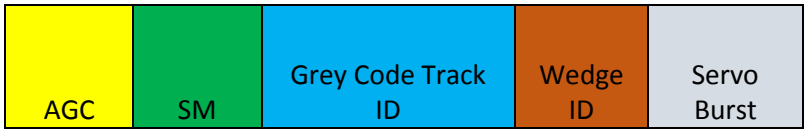


Figure 2.5: An example of servo pattern signal on the disk and zoom in of the servo pattern signal description

HDD's servo system is a combination of position-feedback control, and timing "phase lock loop" (PLL) [9]. Refer the figure 2.6 shows the HDD's timing diagram stamping on the time difference on each servo sector detected, and the timing difference between actual and target phase. For PLL system, the difference between target sync mark timestamp and detected sync mark timestamp of wedge is used to compute the timing error (2.1). The servo pattern's sync mark timestamp is stamped on the SOC free running counter with predefined clock frequency [18]. The SOC counter frequency setting is based on a parameter called "Ts per Rev". The "Ts per Rev" equivalent to digital bits, the parameter is defined by HDD's Channel team based on product basis. The tolerance of speed variation from wedges to wedge is below 15Ts. Example, a product "A" spinning with 5400RPM, and "Ts per rev" is 4665600 which is ~35.722ns (= (60/RPM)/Ts Per Rev * 15Ts).

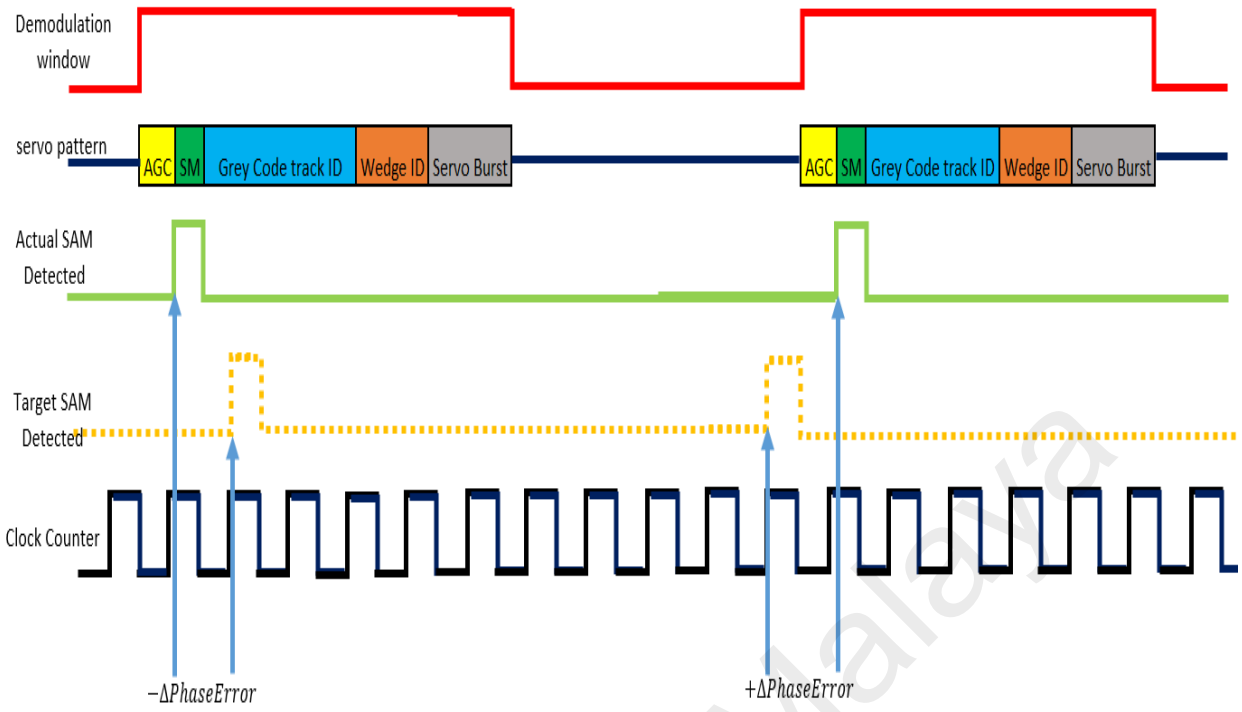


Figure 2.6: A timing diagram how the decode window work for servo pattern, and example of timing error

$$\pm\Delta\text{PhaseError(wedge)} = \text{TargetPhase(wedge)} - \text{DetectedPhase(wedge)} \quad (2.1)$$

Phase error is divided into two categories which is RRO (Repeatable Run Out) and NRRO (Non-repeatable Run Out) [10]. The RRO phase error mainly due to the eccentricity of the disk, spindle motor, and imperfect servo pattern. Other than RRO components, the residual data is called NRRO. NRRO phase error mainly from air turbulence, spindle motor, R/W head, and electronic noise [10]. Air turbulence directly acts to the suspension of the head which causes the intermittent contact between R/W head and media result speed variation. Electronic noise mainly from ADC (Analog to Digital Conversion). Refer to figure 2.7, 2.8 and 2.9 on the examples plots of total phase error plot together with RRO phase error, FFT (Fast Fourier Transform) RRO phase error, and FFT NRO phase error.

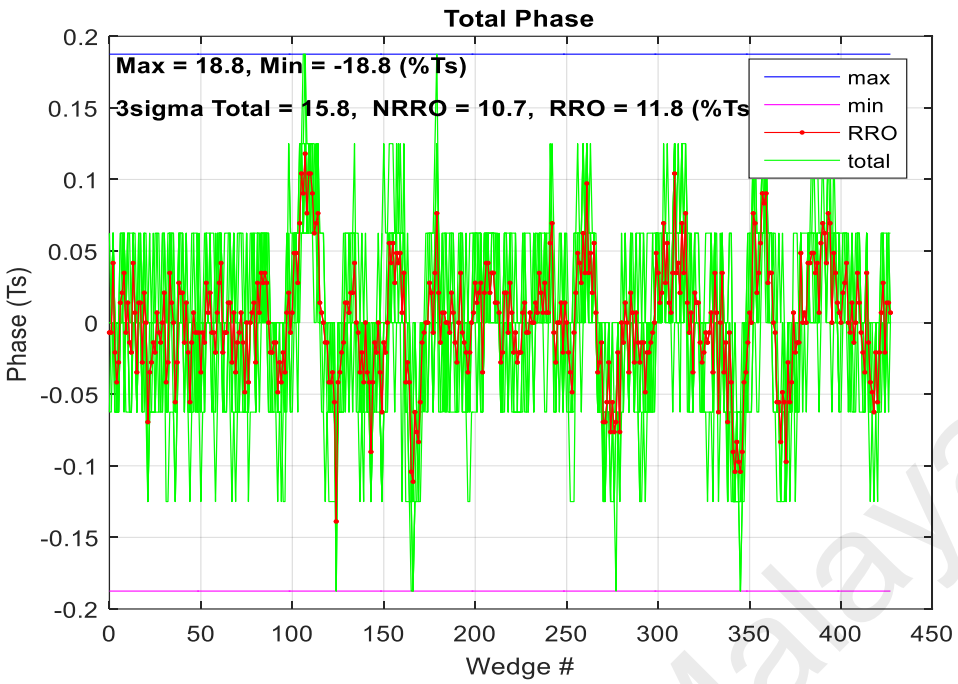


Figure 2.7: Phase/Timing error plot contain RRO component and raw data

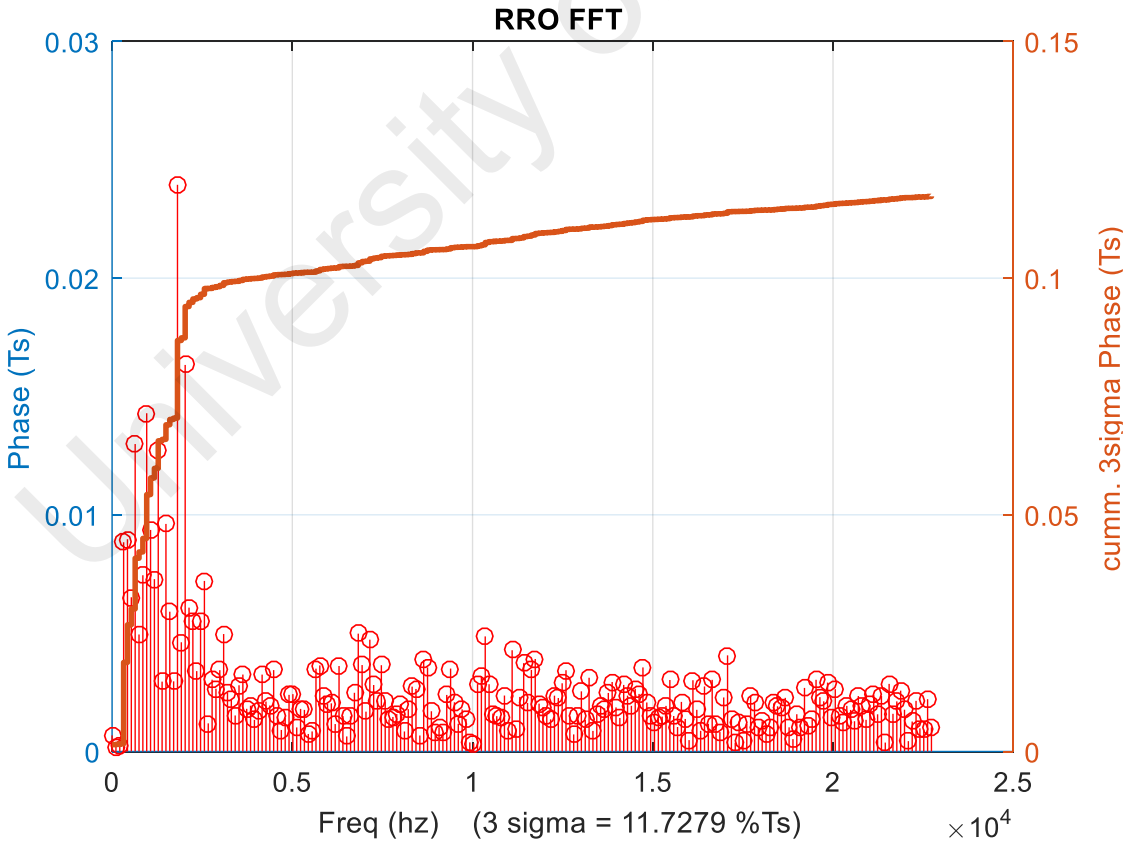


Figure 2.8: Example of graph RRO of phase error plot and FFT of RRO phase error

The RRO value is the median of phase error by revolution, the accuracy of RRO depends on revolutions collected. Refer to equation (2.2) [23].

$$rro_{phase} = \frac{\sum_{i=0}^{i=N-1} err_{phase,i}}{N} \tag{2.2}$$

where err is the phase error per wedge, N is total servo pattern on the disk

Figure 2.9 is the NRRO of phase error, it calculated based on the equation (2.3) [23], by removed the component RRO from the raw phase error data.

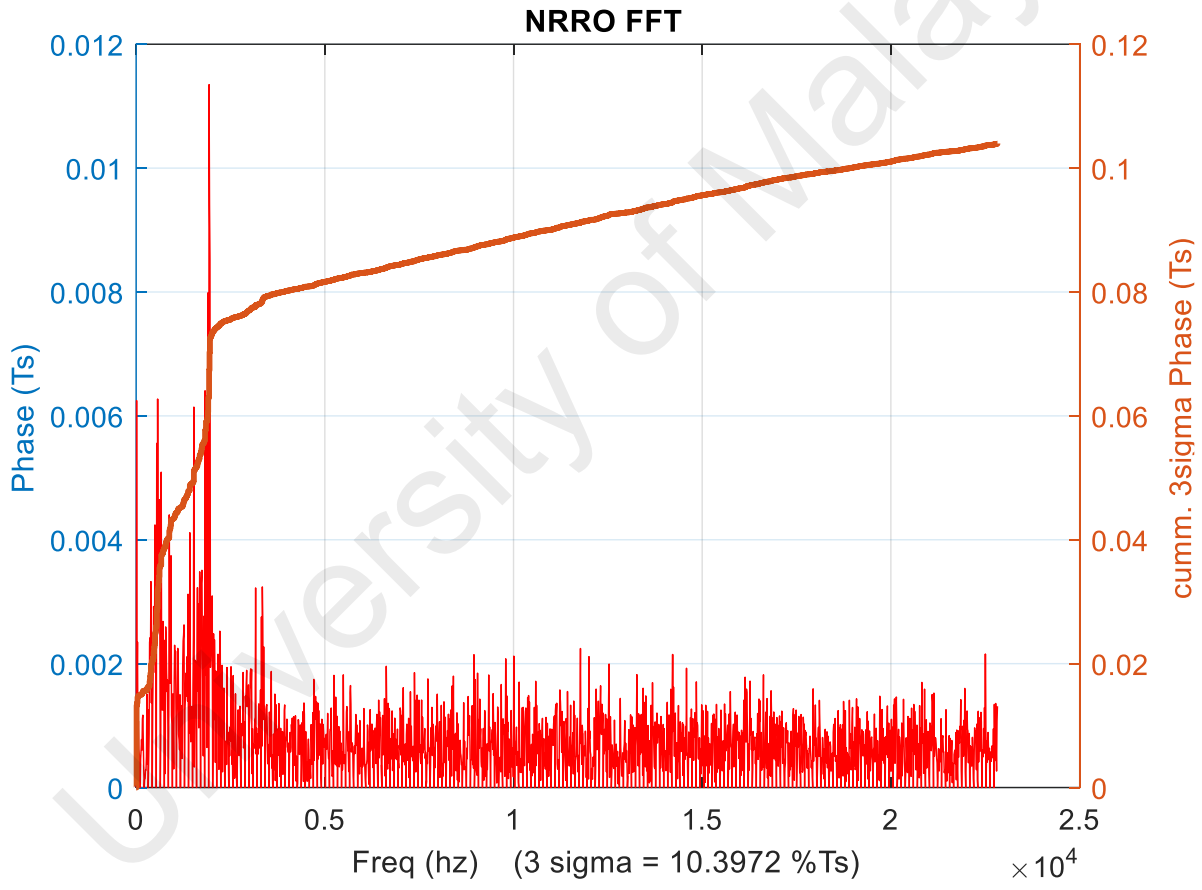


Figure 2.9: NRRO FFT phase error plot

$$\text{nrro}_{\text{phase,i}} = \text{rro}_{\text{phase,i}} - \text{err}_{\text{phase,i}} \tag{2.3}$$

where i is the wedge number, $\text{rro}_{\text{phase,i}}$ from equation (2.2)

2.4 HDD Phase Lock Loop System

The highlighted blocks in figure 2.10 are the timing system in HDD, it contains two time base generators (TBG) which is read clock/servo clock and write clock [11]. The both TBG are run in different frequency. In operation mode, write clock with multiple sets of frequency are configured for writing purpose. The read clock is preset with unique frequency when power up HDD, the setup is to demodulate read servo pattern on the disk.

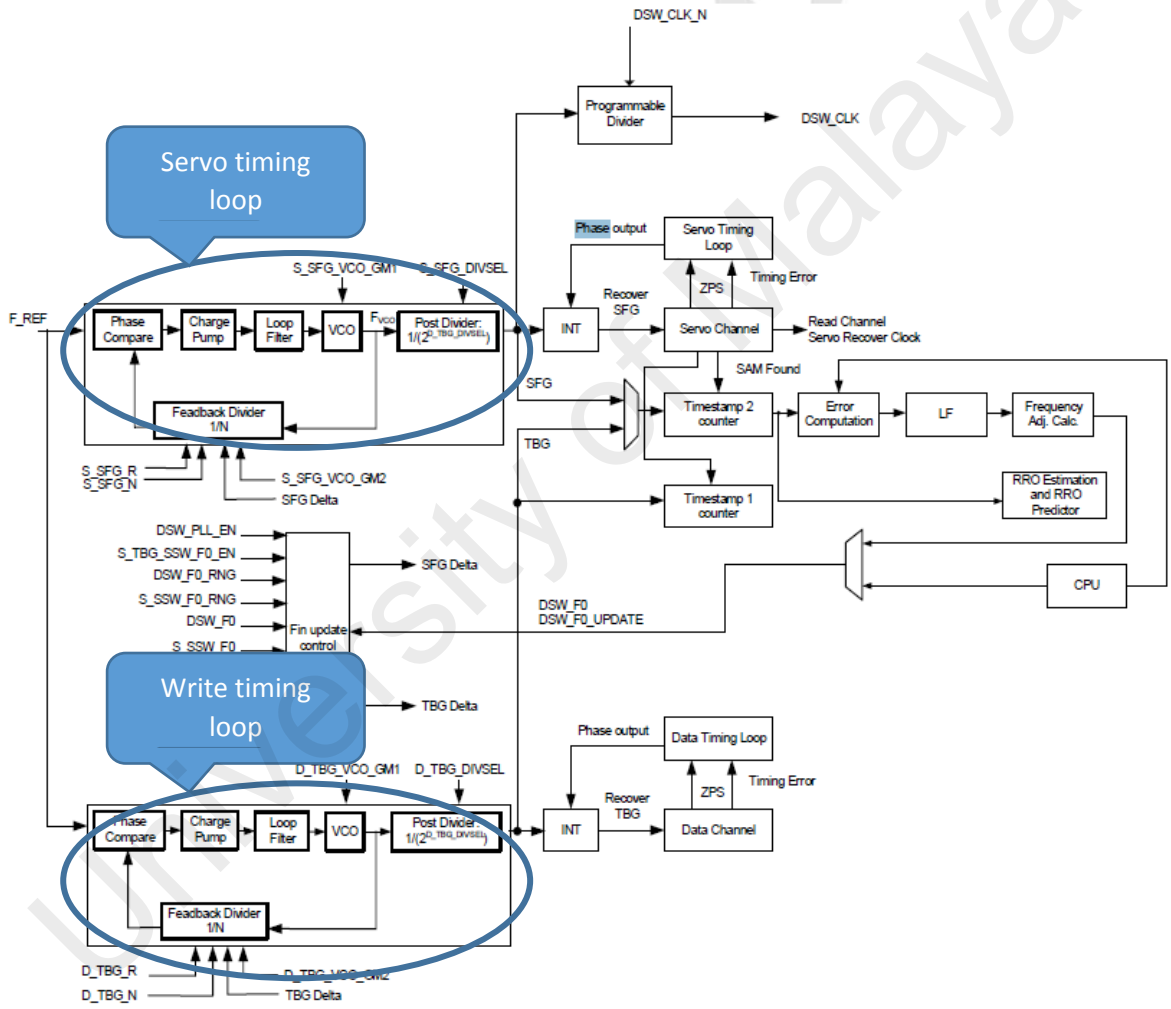


Figure 2.10: Frequency Generator block diagram of the HDD timing system

The PLL's system contains components such as phase compare, charge pump, loop filter and VCO (voltage control oscillator) [11] [figure 2.11]. The phase compare in HDD is a general PLL's phase detector, the voltage difference is generated for the reference and output

of VCO. To maintain the VCO output same as reference frequency, the phase compare voltage is used to output the clock tick to control the VCO voltage.

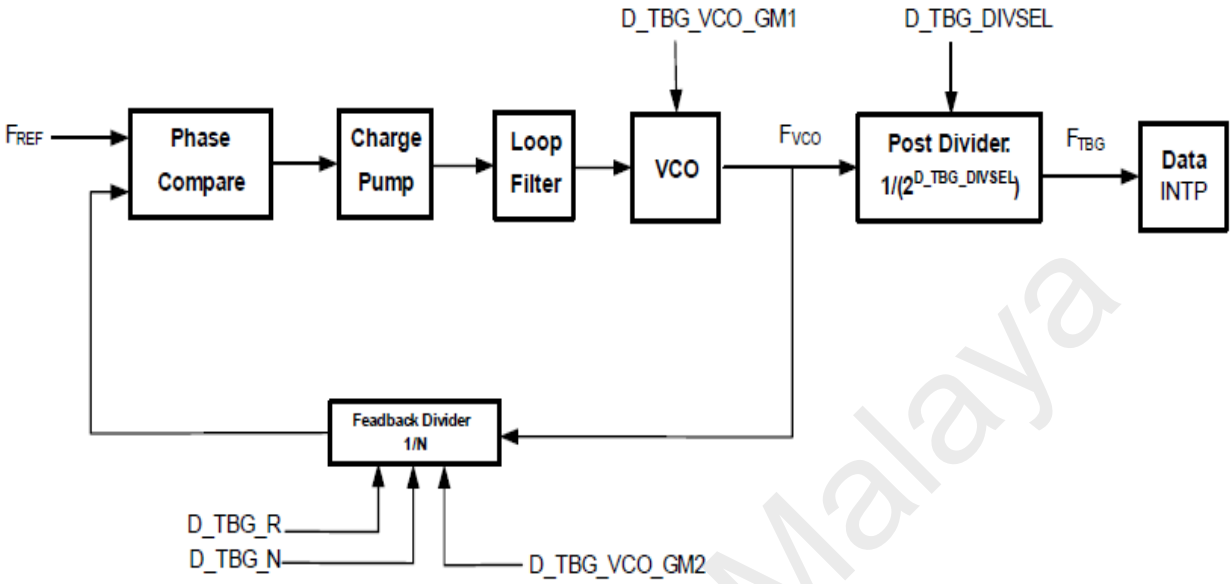


Figure 2.11: Block diagram of a PLL system in HDD

The loop filter is a configurable second order bandwidth filter can be enabled through programmable register DSW_LF_ALPHA and DSW_LF_BETA, the loop filter is equipped with magnitude clamp feature to limit the signal range without saturating the VCO. The bandwidth is adjusted to filter the noise generated by the charge pump. The noise is the root cause of the jittering on the signal to VCO [12]. The hardware loop filter is enabled by setting the register DSW_CLOSE_LOOP_EN, and DSW_LF_EN, the hardware loop filter is only updated when there is successful SAM detected.

A VCO (Voltage Controlled Oscillator) is an oscillator which generate output signal that varied over a range depending on the configuration. The output signal is controlled by the input DC voltage from loop filter [11]. The output frequency is directly related to the input from loop filter. For HDD the oscillation frequency varies from MHz to GHz, by varying the input DC voltage, the output frequency of the signal produced is adjusted.

The PLL's clock frequency adjustment is designed to be able to control by hardware or firmware. DSW_CLOSE_LOOP_EN and DSW_LF_EN control registers are set to 1 to enable the hardware PLL close loop correction system. The downside of the hardware control timing correction is correction only happens when there is two consecutive success detected SAM. At the event when HDD encounter missing samples, it can cause the PLL update delay due to the hardware downside. Firmware close loop control is the option to prevent delay happens. The normalized frequency error from phase error computation is computed by the firmware during the operational mode by reading the servo pattern wedge to wedge SAM timestamp. When encounter any missing sample, the system used the controller estimator value to compute the control. The normalized frequency error information is then written to the register DSW_F0, at least once per servo pattern.

The method to adjust clock frequency offset by write the offset value into the register DSW_F0 to offset the F_{DELTA}. The F_{DELTA} formula written as below:

$$\text{FPLL} = \text{FREF x } 2 - \text{D\_TBG\_DIVSEL x } (\text{D\_TGB\_N} + \text{D\_TBG\_R}/128)/1 + \text{TBG Delta} \quad (2.4)$$

$$\text{PLL Delta} = \text{DSW\_F0 x } 2 - 22 \text{ x } 2\text{DSW\_F0\_RNG} \quad (2.5)$$

Equation 2.4 is calculated the PLL frequency need to be generated and equation 2.5 is the clock frequency offset need to be update every servo pattern to prevent timing variation.

2.5 PID Servo Control for PLL

The design of PLL servo control system has a single mode which is "on track following", unlike the position servo control which has "on track following", "settle control", and "seek control" modes. Figure 2.12 is HDD's PLL servo control system representation in S-domain. The P represents the plant of the PLL system, an integrator (2.6) with ZOH (zero order hold). PID is the feedback controller and PRRO is the feedforward compensation on the RRO error during operation mode (this paper will not discuss details of PRRO) [10]. In timing control design, the NRRO phase error is fed into the timing controller and RRO phase error is removed by PRRO. R is the reference input, which is target timing of N^{th} servo sector during on track control mode; d represents all of disturbances including RRO and NRRO contributing to phase error; y_h is phase value in terms of Ts read back from channel register.

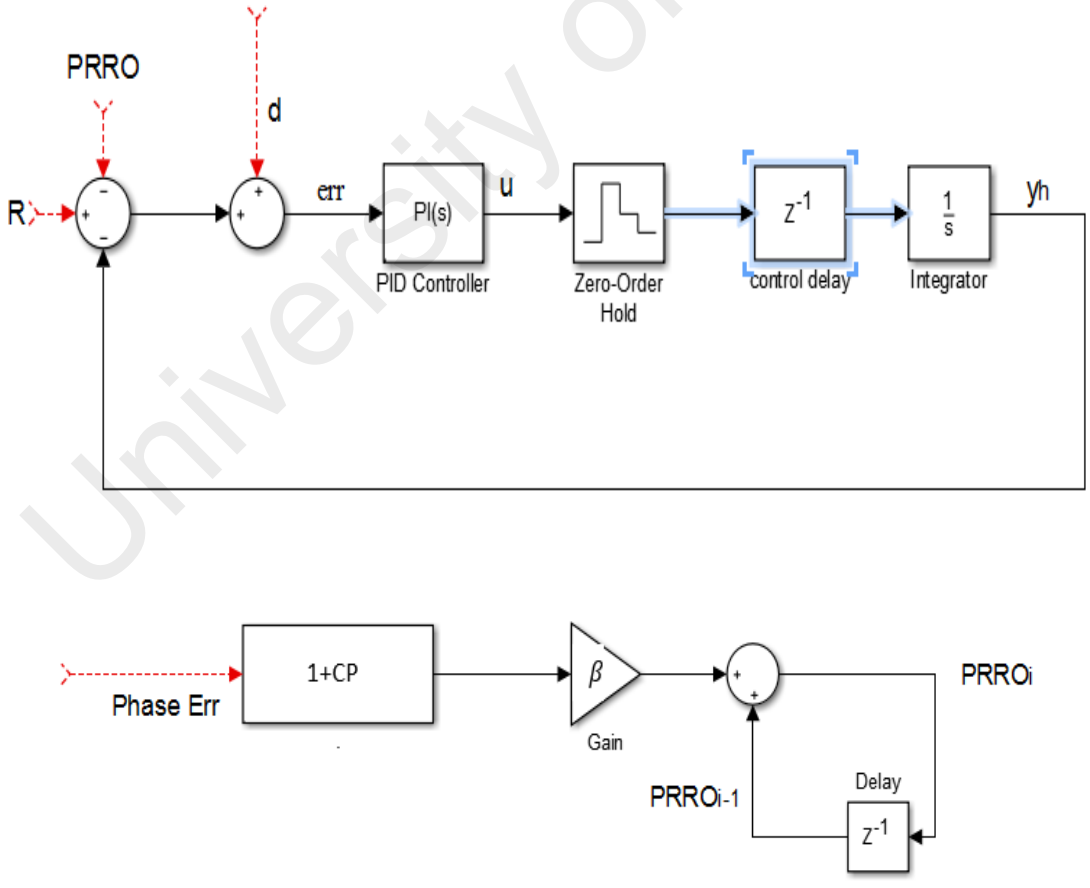
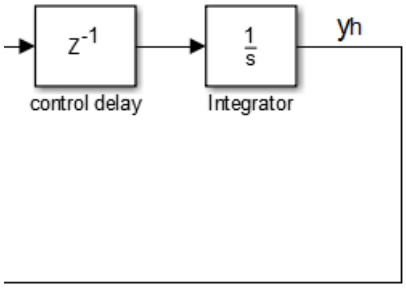


Figure 2.12: HDD PLL servo control architecture and PRRO architecture



$$P(s) = e^{-Ts} * \frac{\alpha}{s} \tag{2.4}$$

where z^{-1} is the control delay of PLL system, α is the scaled factor convert to register value

Existing implementation is used standard PID controller [figure 2.12] [13] for the phase error correction. $U(z)$ is the output control to update the frequency synthesizer, and $E(z)$ is the phase error from servo pattern to servo pattern. U_{max} is the maximum control output for phase correction, the boundary condition is based on the system-on-chip (SOC) design.

$$U(z) = \left(K_p + K_i \frac{1}{1 - z^{-1}}\right) * E(z), \qquad |U(z)| \leq U_{max} \tag{2.5}$$

where, K_p = Proportional gain, and K_i = Integrator gain

The calculated U(z) value is normalized and written into the SOC register, and the SOC PLL register is updated for the error frequency to synchronize the servo sector and spindle speed variation on each interval servo sector [17].

2.6 LQG Servo Control for PLL

A Linear-Quadratic-Regulator [figure 2.13] is known as full state feedback controller which uses the ARE (Arithmetic Ricatti Equation) to compute the feedback gain value K for the poles eigenvalues, the controller uses state space model (2.8) to obtain the desired behavior of the system [14].

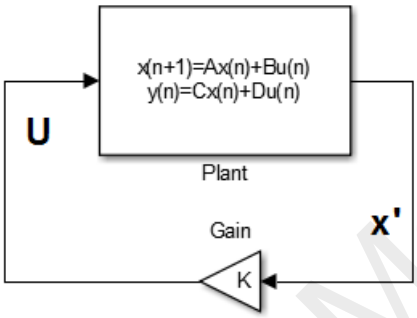


Figure 2.13: A full state feedback block diagram

Where A_k is called system matrix (n x n), B_k is the control/input matrix (n x p), C_k is the output matrix (q x n), and D_k is the feedforward matrix (q x p). x_k is denoted as "state vector" with $x(t) \in R^n$ as real numbers in matrix form n; u is denoted as "control vector" with $u(t) \in R^p$ as real numbers in matrix form p; y is denoted as "output vector" with $y(t) \in R^q$ as real numbers in matrix form q [14].

$$x_{k+1} = Ax_k + B_k u \tag{2.6}$$

$$y = C_k x + D_k u$$

$$u = -K_R x_k \tag{2.7}$$

In state feedback it uses the state measurement to determine the control action. For simplicity, the target reference is set to zero for theory proving. By substituting the equation 2.7 into equation 2.6 the state feedback turn out to be:

$$x_{k+1} = (A - BK_R)x_k \tag{2.8}$$

The ($A-BK_R$) eigenvalues is now different compare to A_k system matrix eigenvalues. The poles now are dependent on selection of K_R values, so in design the K_R values able to modify the eigenvalue location [14]. The objective is to minimize the performance index equation as below (2.9):

$$J_N = \sum_{k=0}^{N} x^T(k)Q(k)x(k) + u^T(k)R(k)u(k) \tag{2.9}$$

The goal is to minimize the total phase error, the performance index of cost function of the system is as minimal as possible. $\mathbf{x^T}$ is the transpose of state vector. Q uses positive definite value and R uses semi-positive definite value. By weighing the Q and R matrix which defines state vectors and control to be penalized; u^T is the transpose of control input to the plant [14].

Before proceed on calculating the feedback gain, K_R, the system needs to be checked on the controllability and observability [14]. The controllability of a system means the state is fully controllable and it is possible to transfer system arbitrary initial state to any desired state in finite time period. Given the state space plant model equation (2.4) to check the controllability (2.10) as below:

$$[B \quad AB \quad \cdots A^{N-1} \quad B] \tag{2.10}$$

Second is test the observability (2.11) as below

$$\begin{bmatrix} C \\ CA \\ \vdots \\ CA^{N-1} \end{bmatrix} \tag{2.11}$$

The rank of a matrix is defined by the number of linearly independent rows and/or the number of linearly independent columns. By using Matlab command "rank(ctrl(C,A))" to check system is a full rank system. If the system is fully controllable and the eigenvalue of the system can be placed at any desire location. Same goes to observability by Matlab command

"rank(ctrb(A,B))" to check system full observable. If the system is fully observable, estimator design is able to fully observe all the states. If both criteria are satisfied, it means the system is controllable and observable in any given finite time. Next is to find the K_R gain for the states vector using ARE method. (Refer to equation (2.12) obtained as below:

$$K = PB^T XA \tag{2.12}$$

$$P = (B^T XB + R)^T \tag{2.13}$$

and **P** is found by following discrete ARE equation below:

$$A^T XA - X - A^T XB(B^T XB + R)^{-1}(B^T SA) + Q = 0 \tag{2.14}$$

To eliminate the tedious long calculation and derivation using Matlab command "dlqr" (Discrete Linear Quadratic Equation) to calculate the K_R gains. The input parameters for "dlqr" are the system matrix A, control matrix B, Q, and R matrix. For Q matrix initial value can be computed using the equation below:

$$Q = C^T * C \tag{2.15}$$

$$R \in \mathbb{R}^{1xn} \tag{2.16}$$

As normal practice R value is selected as 1 as for single output system, this gives maximum effort for control output response to the states change. The Q and R value need to be tuned by iterative way to get the best performance by user. Next is to implement optimal full state estimator for the system.

Kalman Filter estimator also known as Linear Quadratic Estimator (LQE), and the system must be observable using the equation (2.11) [22]. Kalman filter technique adopts the iterative mathematical process that uses a set of equations and consecutive data inputs to quickly estimate the true values, when the values contain unpredicted of random error, uncertainty, or variation. Refer figure 2.14 is an example how a Kalman Filter connect between the controller and system.

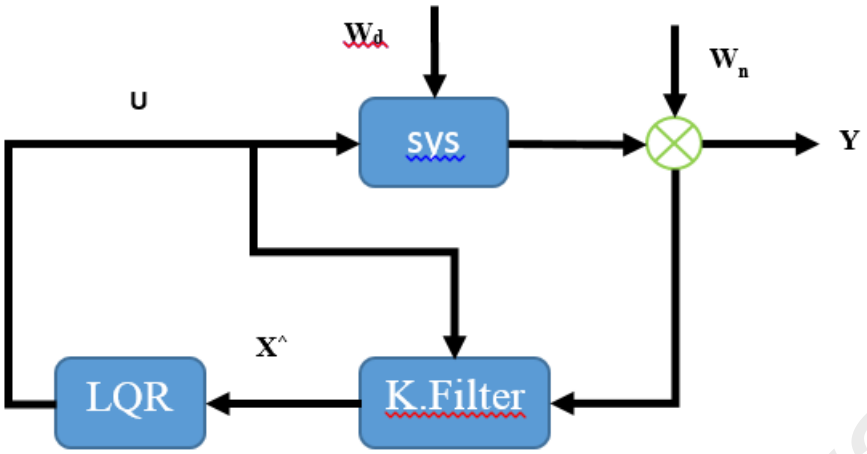


Figure 2.14: A block diagram of LQR with Kalman Filter in the system

Kalman Filter [figure 2.14] assumes there are two noise disturbances act on the system, the noises are called "measurement noise" and "process noise". Refer to equation (2.17), the process noise is added on the state vector equation, and measurement noise is added in the output matrix equation. Since Kalman Filter is an estimator which needs two inputs to the Kalman Filter equations to estimate the states value. Refer to the equation (2.18):

$$x_{k+1}=\mathrm{A}x_k+B_k\mathrm{u}+w_d \tag{2.17}$$

$$\mathrm{y}=C_k\mathrm{x}+D_k\mathrm{u}+w_n$$

$$\check{x}(k+1) = A\check{x} + Bu + K_f(y - \check{y}) \tag{2.18}$$

$$\check{y} = C\check{x}$$

where K_f is the update gain for the difference between the real output of system **y** and the estimator $\check{y}$**.** Every time the system gets the new measurement **y** will compare with the estimator $\check{y}$**,** then multiply the K_f to correct the full state vector. The equation can be formed in another way as below:

$$\check{x}(k+1) = (A - K_f C)\check{x} + [B \quad K_f]\begin{bmatrix} u \\ y \end{bmatrix} \tag{2.19}$$

The equation (2.19) is similar like a LQR system which has [B K_f] as B matrix and [u y]^T as the input matrix of system. When the dynamic system is stable $\check{\boldsymbol{x}}$ is converge to the desired state. Meaning the K_f gain is properly placed it will yield a stable dynamic system same as LQR system. Next is to look at the error dynamic of the estimator versus the system. Given the equation of error dynamic as below:

$$error, \varepsilon = x - \check{x} \tag{2.20}$$

$$error\ dynamic, \frac{d}{dt}\varepsilon = \frac{d}{dt}x - \frac{d}{dt}\check{x} \tag{2.21}$$

By substituting the standard system state space into (2.21) it yield as below:

$$\frac{d}{dt}\varepsilon = A(x - \check{x}) - K_f C(x - \check{x}) \tag{2.22}$$

$$\frac{d}{dt}\varepsilon = (A - K_f C)\varepsilon \tag{2.23}$$

The equation 2.23 is similar to LQR equation (2.8) which means the K_f gain is changing the eigenvalue of the error dynamic system matrix that converge error in finite time. The K_f gain can be obtain through ARE computation same as LQR. By using Matlab command "K_f=lqe(A,V_d,C,V_d,V_n)" it generate the gain for the Kalman Filter system.

CHAPTER 3: METHODOLOGY

3.1 The Procedures to Setup Apparatus

This chapter shows step-by-step procedure on how the setup of the experiments were done (refer to the figure 3.1), controller design steps in calculation and Matlab, and implementation firmware code. The experiments are tested on bench with a multi platters HDD and the parameters (refer Table 3.1) for both control scheme. The requirement of apparatus to run the test are listed as below:

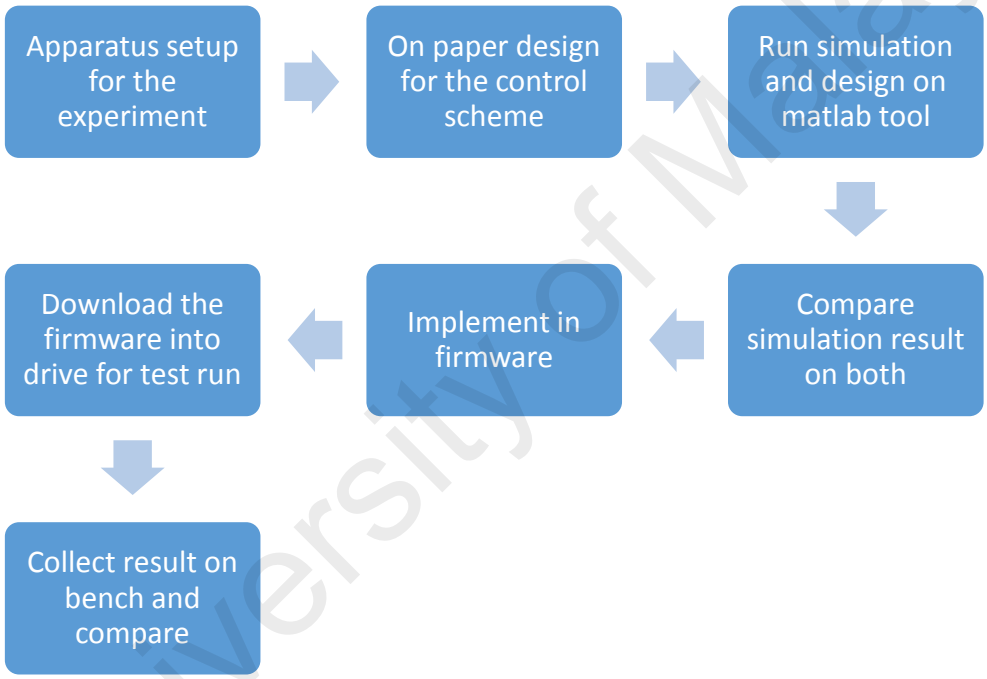


Figure 3.1: The process flow of methodology on designing the control scheme for HDD

Table 3.1: HDD drive parameters

Parameters	Value
RPM	6000
Servo Patterns	324
Platters	1
Heads	2

1. Refer to figure 3.2, USB to serial device, come with USB connector to micro USB. The device is soldered with two wire as output which is Tx, and Rx for drive communication.

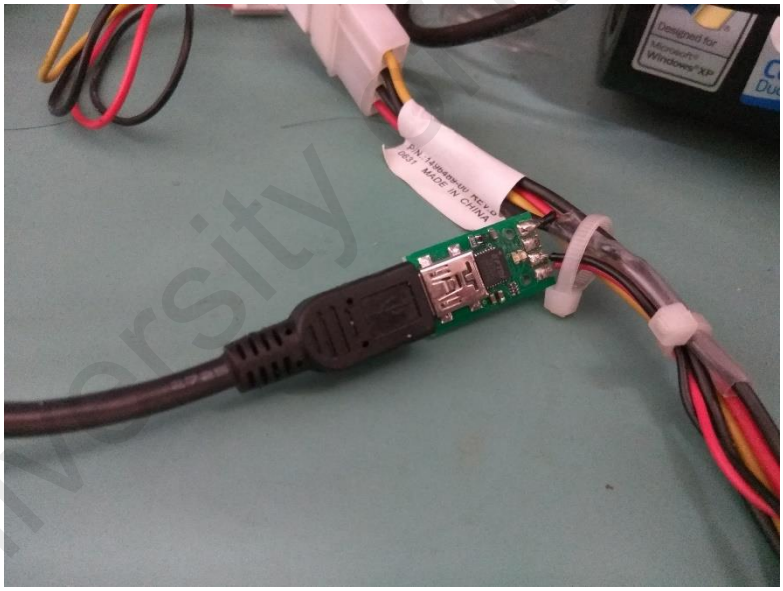


Figure 3.2: USB to serial for drive and computer communication

2. Power Supply can use either the PC power supply or laboratory DC power supply. Refer to figure 3.3.

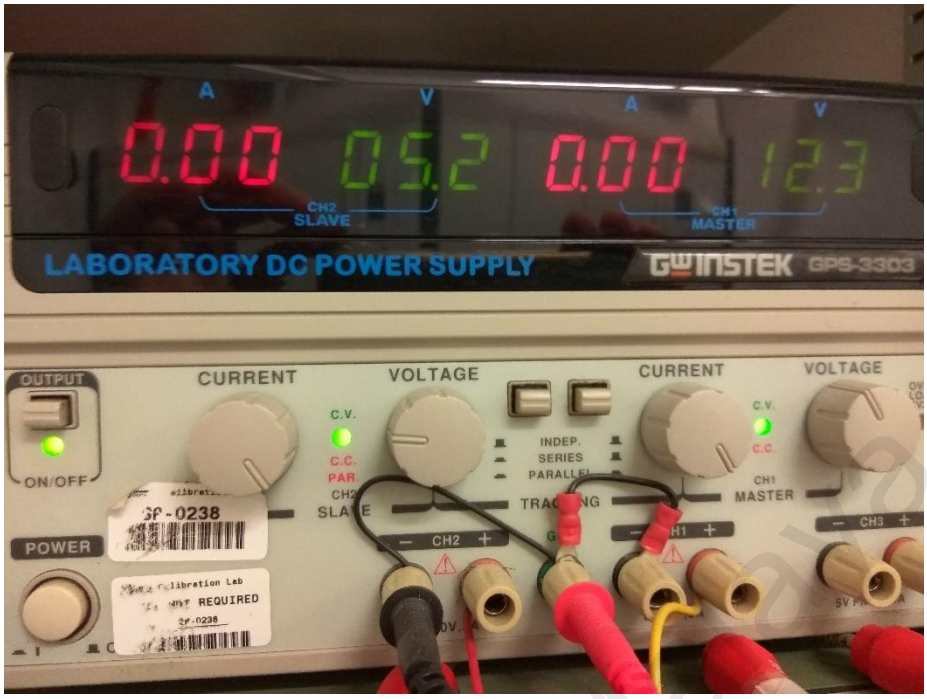


Figure 3.3: Power supply with 12 Volt and 5 Volt supply needed for HDD to operate

3. A Hard Disk Drive with servo pattern written. Refer figure 3.4.

Figure 3.4: A pre-written servo patterns HDD drive needed for the testing

4. SATA extension connector for easy connection due to HDD pins are too small for the socket to push in. Refer figure 3.5.

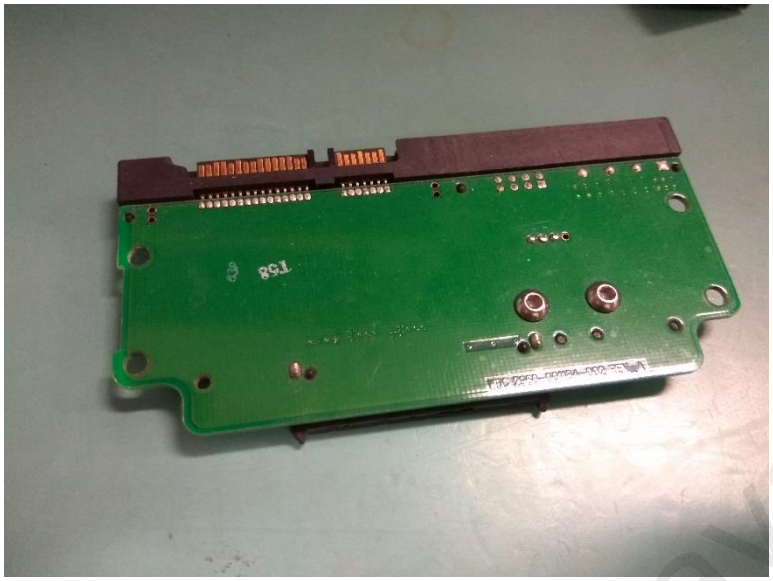


Figure 3.5: A SATA connector for extension purpose

5. ARM compiler software is used for firmware compilation into bin file for execution. HDD drive is using proprietary SOC by Marvel. The chip architecture is based on ARM Cortex R series, and arm compiler is needed for code compilation. Refer to figure 3.6.

```
D:\Work\FillerFirmware\167562.1.1.12>armcc -vsn
Warning: C3910W: Old syntax, please use '--vsn'.
ARM/Thumb C/C++ Compiler, RVCT3.1 [Build 862]
For support please contact Mentor by emailing embedded_support@mentor.com or at www.mentor.com/supportnet
Software supplied by: ARM Limited
```

Figure 3.6: Licensed ARM Compiler

6. Matlab software needed because the communication command is send through Matlab, and the data can direct use Matlab to retrieve and plot at the same time. Refer to figure 3.7.

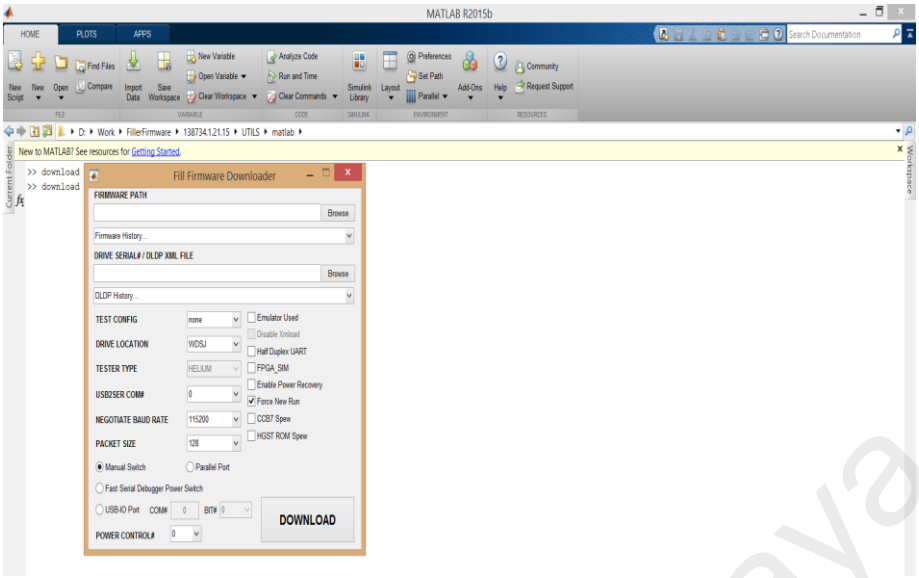


Figure 3.7: The Matlab tools for drive communication purpose

7. With all the components assembly together a complete set of single drive testing on bench is show in figure 3.8.

Figure 3.8: Complete view of HDD test bench look like

The following procedures/steps to run the control scheme using apparatus display above.

a. First, the HDD is connected to USB2Serial interface using any windows desktop with Matlab installed.
b. The HDD is downloaded with new firmware and new control scheme implemented (in binary file).
c. A special command "spiral copy" is send by Matlab to the drive.
d. During the spiral copy mode the phase error data is saved in the HDD's RAM memory and extracted out through USB2Serial interface by Matlab Software.
e. The steps **d.** is tested on different location on the disk. Each test location will collect 10 revolution data then normalize and average, to observe the result is not affected by disk location.

The drive is tested with two control scheme as explain on "Chapter 2: Literature Review" which is the PID control and LQG control. The next section shows how the PID and LQG calculation and simulation is done, before implemented into code and run in HDD.

3.2 PID Controller Calculation for PLL

The PID was implemented on one of the HDD products name "Spyglass" with 4 platter of media, spinning on 6400RPM during its operational mode. The Matlab tool is used to generate the transfer function, gain margin, phase margin, state space model and etc. The set of K_P, K_I, and T_S are chosen as show in table 3.2.

Table 3.2: PID controller setting using platform called “Spyglass”

Parameters	Value
K_P	2.3
K_I	0.2188
T_{sample}	21.9042ns/sample
α	80284
Control delay	3.1µs

*T_{sample} is uses the product “Spyglass” as test platform with 428 servo patterns, and spinning at 6400RPM.

The plant with control delay transfer function as below:

$$\mathrm{P(s)} = \mathrm{e}^{-3.1e^{-6}*\mathrm{s}} * \frac{80284}{\mathrm{s}} \tag{3.1}$$

Convert s-domain transfer function (2.6) to discrete transfer function

$$P(z) = \frac{0.1509z + 0.02489}{z^2 - z} \tag{3.2}$$

For discrete PID controller (2.7) it will be

$$\mathrm{U(z)} = \left(2.1562 + 0.2188\frac{1}{1-\mathrm{z}^{-1}}\right) * \mathrm{E(z)} \tag{3.3}$$

Simplified the equation (3.3) to,

$$\frac{\mathrm{U(z)}}{\mathrm{E(z)}} = \frac{\text{2.156-2.375}}{\text{z-1}} \tag{3.4}$$

The plant without control delay is shows in state space form (3.5), the control delay is removed and separated because it will give a better performance in simulation.

$$\mathrm{A}_d = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}, \mathrm{B}_d = \begin{bmatrix} 0.5 \\ 0 \end{bmatrix}, \mathrm{C}_d = [0.3018 \quad 0.04978]$$

$$x_{k+1} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} x + \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} u \tag{24}$$

$$\mathrm{y} = [0.3018 \quad 0.04978] C$$

For the open loop transfer function CP of the PLL's system it shows as below (3.6):

$$\frac{Y(z)}{E(z)} = \frac{-0.3584z^2 + 0.2662z + 0.05366}{z(z^2 - 2z + 1)} \tag{3.6}$$

As in state space form is shows as below (3.7):

$$\widehat{\mathrm{A}}_{\mathrm{ol,d}} = \begin{bmatrix} 2 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \widehat{\mathrm{B}}_{\mathrm{ol,d}} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \hat{\mathrm{C}}_{\mathrm{ol,d}} = [-0.3584 \quad 0.2662 \quad 0.05366] \tag{3.7}$$

For the close loop transfer function $\frac{\mathrm{CP}}{1+\mathrm{CP}}$ of the PLL's system it shows as below (3.8):

$$\frac{Y(z)}{R(z)} = \frac{-0.3584z^2 + 0.2662z + 0.05366}{z^3 - 1.642z^2 + 0.7338z - 0.05366} \tag{3.8}$$

Translate the z transfer function (3.8) to state space data form as below (3.9):

$$\widehat{\mathrm{A}}_{\mathrm{cl,d}} = \begin{bmatrix} 1.642 & -0.7338 & 0 \\ 1 & 0 & 0 \\ 0 & 0.25 & 0 \end{bmatrix}, \widehat{\mathrm{B}}_{\mathrm{cl,d}} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \hat{\mathrm{C}}_{\mathrm{cl,d}} = [0.3584 \quad -0.2662 \quad -0.2417] \tag{3.9}$$

Using Matlab command "pzmap" to check the pole and zero location of the transfer function (Refer to appendix A for Matlab code). All the poles are within the unit circle [figure 3.9], which indicate the PLL system is a stable system.

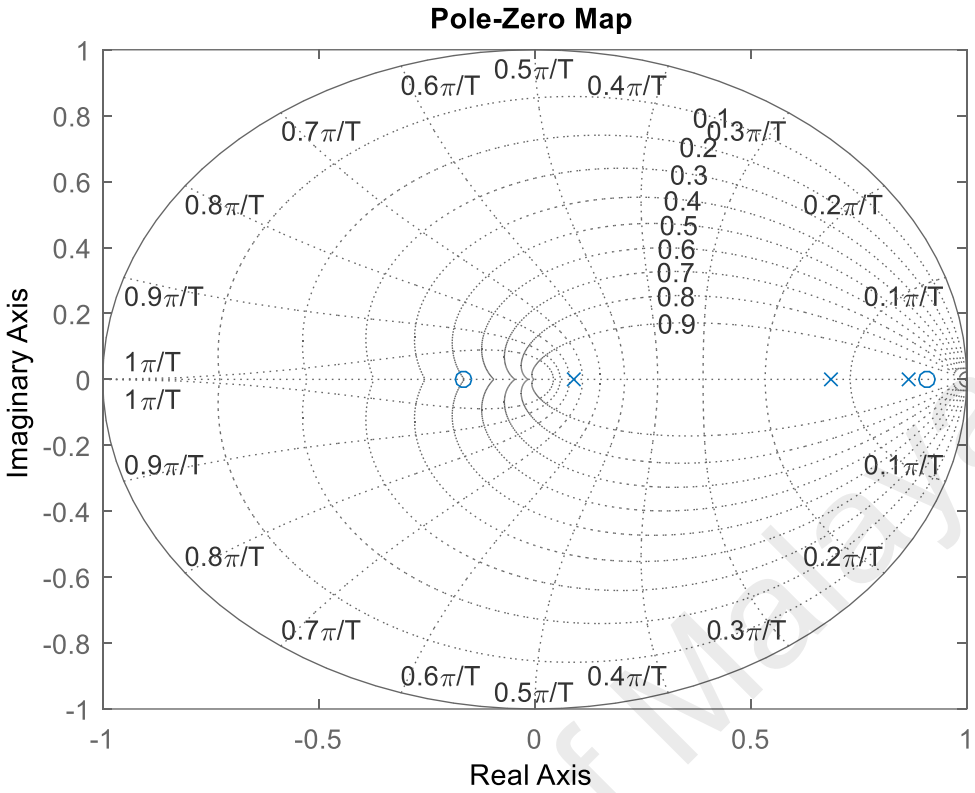


Figure 3.9: pole-zero mapping for PLL's transfer function

In PID controller design except using the bandwidth, gain margin, and phase margin to determine the system performance, the ERC (error rejection curve) (3.10) plays very important role to know the rejection and amplification of disturbance region at any frequency range. The ERC equation is shows as below:

$$\text{ERC/output sensitivity} = {}^{1}/_{1+\text{CP}} \tag{3.10}$$

For PLL system,

$$\text{ERC} = \frac{z^3 - 2z^2 + z}{z^3 - 2.358z^2 + 1.266z + 0.05366} \tag{3.11}$$

3.3 LQG Controller Calculation for PLL

The system dynamic equation (3.5) is checked on the observability and controllability using the formula (2.10) and (2.11). The two parameters R and Q are the weights to optimize the cost function, so it is important that balance the relative of the control effort (R) and state variable (Q). The R initially select as 1 for the control effort, and Q using the equation below:

$$\mathrm{Q} = \begin{bmatrix} Q_1 & Q_3 \\ Q_3 & Q_2 \end{bmatrix} = C^T * C \tag{3.12}$$

$$\mathrm{Q} = \begin{bmatrix} 0.3018 \\ 0.04978 \end{bmatrix} [0.3018 \quad 0.04978] = \begin{bmatrix} 0.0911 & 0.0150 \\ 0.0150 & 0.0025 \end{bmatrix} \tag{3.13}$$

Refer to the cost function (2.9) change it to linear equation for the state variable it will become:

$$\check{X}^T \mathrm{Q} \check{\mathrm{X}} = Q_1 x_1^2 + Q_2 x_2^2 + 2Q_3 x_1 x_2 \tag{3.14}$$

From the linear quadratic equation above shows Q is the weight of the x variables that user want to minimize. For PLL system x1 is the phase error and x2 is the control delay, the weight matrix is focus on the phase error,x1, and not control delay, and removing the $2Q_3 x_1 x_2$ to prevent the term give extra weighting to the LQR equation. The table below is the parameter for "dlqr" command in order to generate the K_R gain.

Table 3.3: LQR Parameters

Parameters	Value
Q	$\begin{bmatrix} 0.0911 & 0 \\ 0 & 0.0025 \end{bmatrix}$
R	1

Final step, using the Matlab script to generate the parameters for the PID controller or LQG controller for firmware. Then implement the parameters into the HDD firmware PLL control code.

For HDD firmware is uses C-programming as the programming language. The controller programming codes can refer to appendix C and D section on both control schemes on the implementation. Firmware is downloaded [figure 3.10] into the drive and uses Matlab to command the spiral copy operation and collected the data at the same time.

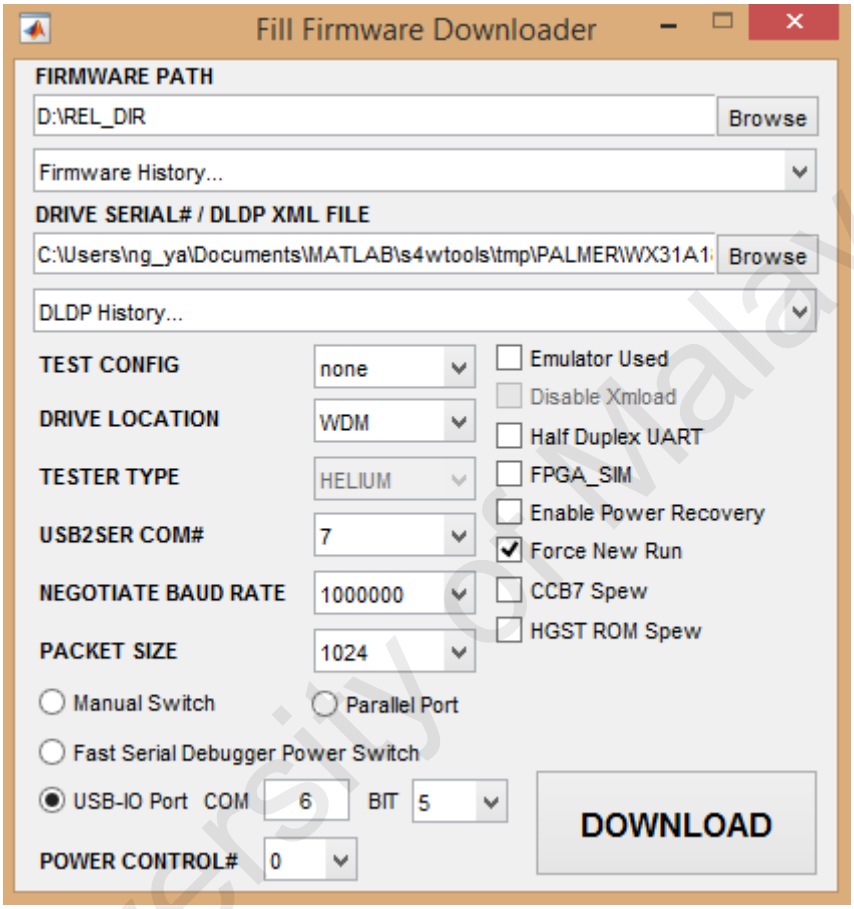


Figure 3.10: GUI for downloading the firmware into the HDD

When process is finished, the data is extracted out using the Matlab script developed by internal engineers. The data is plotted and compared side-by-side to analyze which method gives better result in "Chapter 4: Result". Next chapter is discussed about the result after implementing the new scheme.

CHAPTER 4: RESULT

4.1 Introduction

This chapter will be discussing the simulation result on Matlab and result after implemented both control scheme on single HDD. The equations derivation on both controllers using Matlab have shown on previous "Chapter 3: Methodology". On this chapter will be focusing on the drives result after implemented the controls scheme, by using Fast-Fourier Transform (FFT) plot and phase error variance across multiple disk location to analyze the performance of the controls scheme.

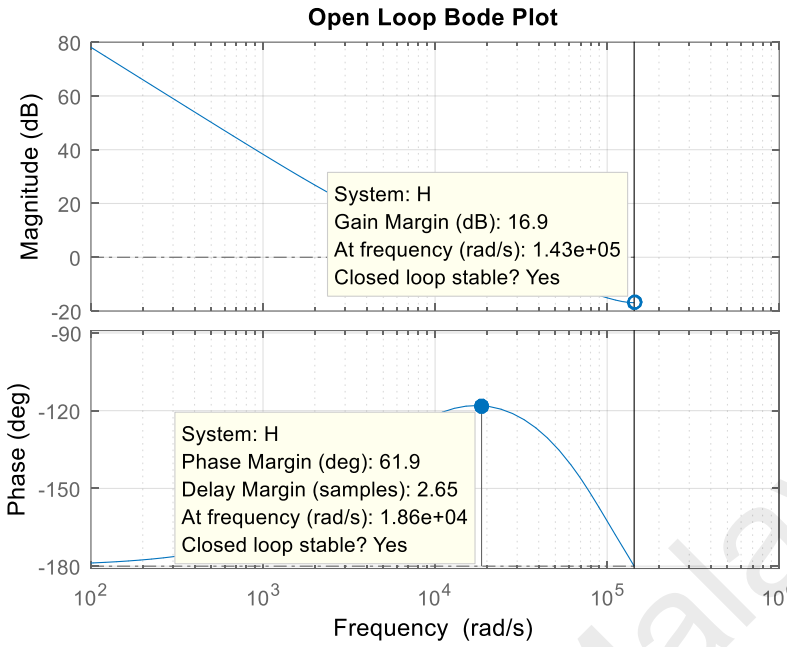
4.2 Simulation Result for PID Controller

Equation (3.6) is the transfer function of the PLL system where the command called "bode" is used in Matlab to plot the bode diagram of the transfer function. The plot is shows on below [figure 4.1] and compared with the HDD bode plot.

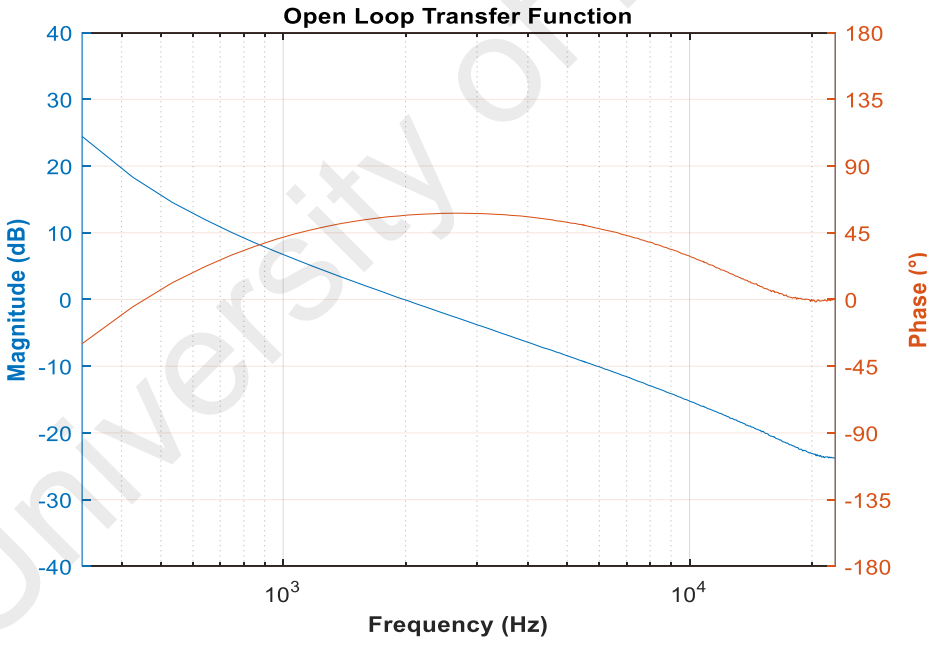
Table 4.1: System response of simulation versus experiment

Parameters	Simulation	Experiment
Gain Margin(dB)	16.9	20
Phase Margin(angle)	61.9	50
Crossover Frequency	~18khz	~2kHz

The open loop and close loop bode plot are generated by Matlab before implementation on firmware code. The figures 4.1(a) and (b) are the simulation and experimental result of open loop; the figures 4.2(a) and (b) are close loop bode plot simulation and experiment result.

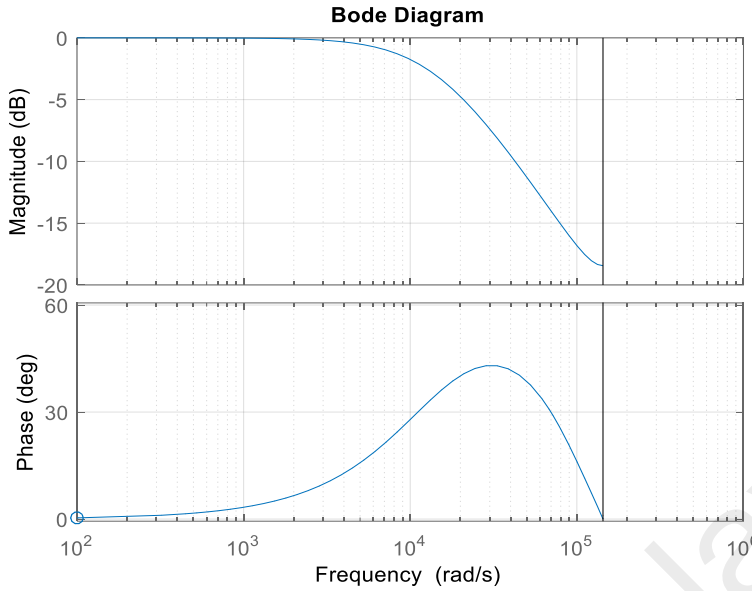


(a)

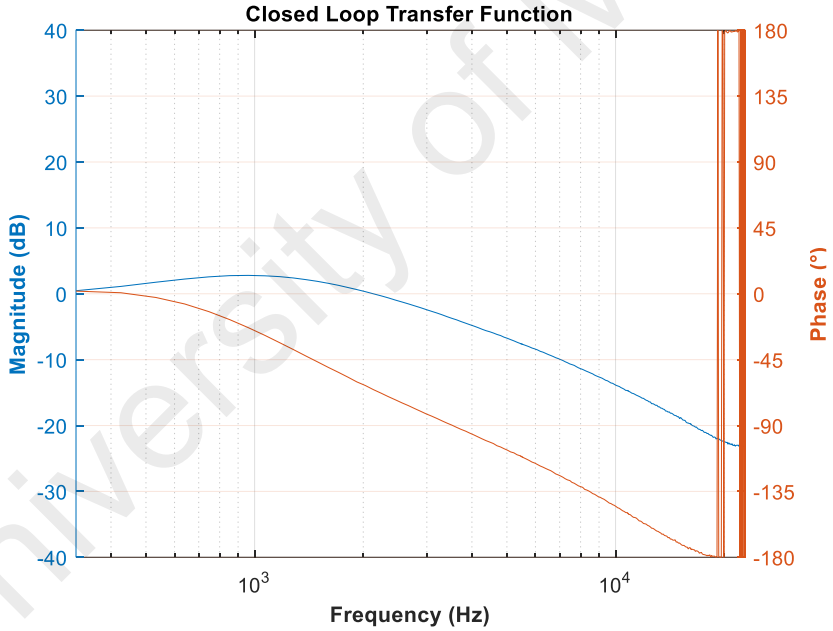


(b)

Figure 4.1: (a) The simulation open loop bode plot of the PLL transfer function. (b) The experimental result of open loop bode of the PLL system



(a)



(b)

Figure 4.2: (a) The simulation close loop bode plot of the PLL transfer function. (b) The experiment close loop bode plot of the PLL transfer function

A unit step response is tested on the system and the result shows on figure 4.3. The control shows slight overshot and settling around ~0.56926ms refer to table 4.2.

Table 4.2: The performance of values of the PID

Parameters	Value
Rise Time	65.685us
Settling Time	0.56926ms
Overshoot	16.4620
Peak	1.1646
Peak Time	0.17516ms

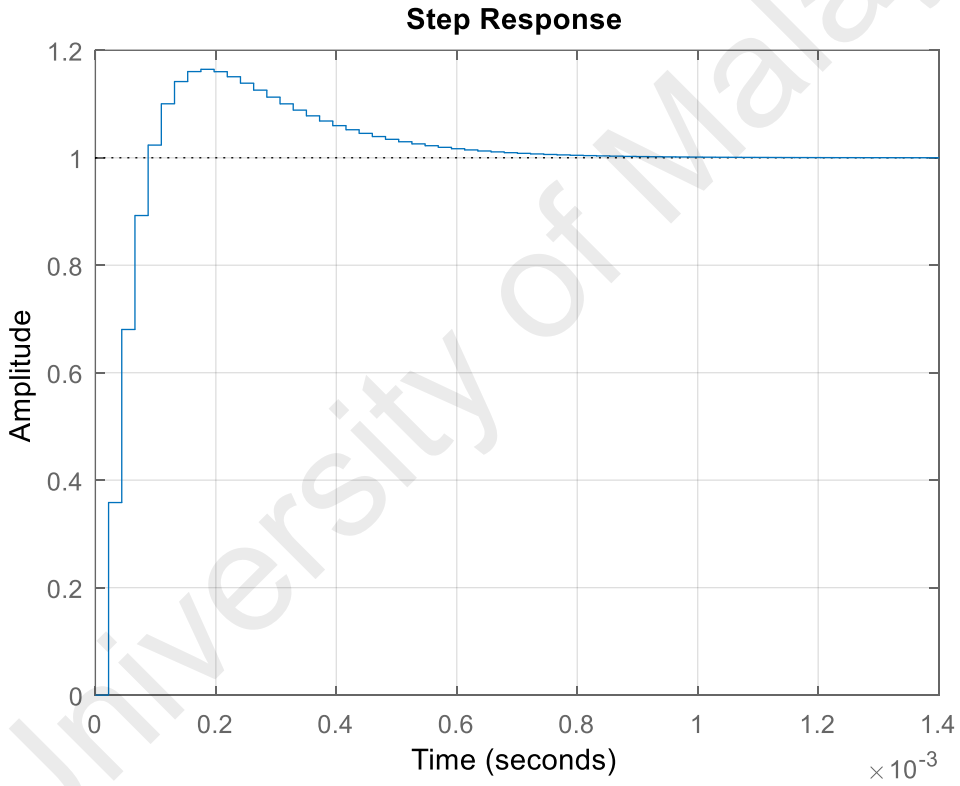
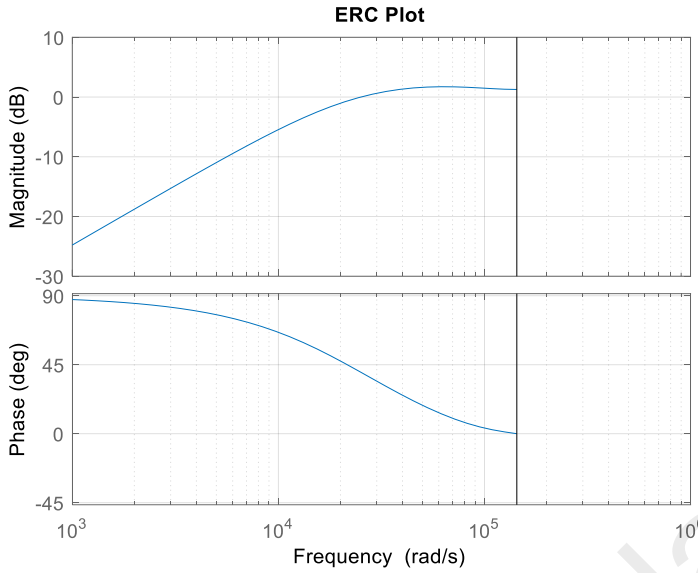
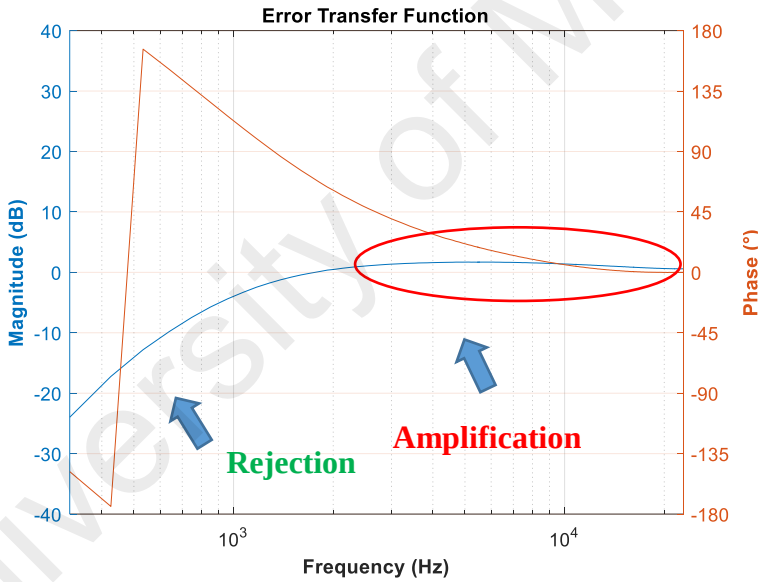


Figure 4.3: Unit steps response simulation on the transfer function

The figure 4.4(a) is the simulation plot of ERC of PLL system, the result shows amplification on certain frequency range. ERC plot acts as a guidance for control tuning since the plant transfer function is using a simple integrator $\frac{1}{S}$ to represent. The PID is implemented with the control parameters and it shows prominent result. The PID firmware code refer to appendix B.



(a)



(b)

Figure 4.4: (a) The simulation ERC plot of the PLL system. (b) The experimental result of ERC bode plot of the PLL transfer function

4.3 Simulation Result LQG Controller

With the input parameters input to the command "dlqr" it generates the K_R gain for the LQR system. The commands script shows as below:

```
% LQR Design

dsysPs = ss(dsysP);
Q = dsysPs.c'*dsysPs.c;
R = 1;
k = dlqr(dsysPs.a, dsysPs.b, Q, R);
Ac = [(dsysPs.a-dsysPs.b*k)];
Bc = [ dsysPs.b ];
Cc = [ (dsysPs.c-dsysPs.d*k) ];
Dc = [ dsysPs.d ];
syslqr = ss(Ac,Bc,Cc,Dc,ts);
```

Figure 4.5: The matlab commands for LQR design

Next step is run the "syslqr" the transfer function of LQR with step response to examine the transient response of the new system compare with PID controller. If result shows not promising, Q and R weight need to be adjust to get optimize performance. The process is iterative run until a optimize result is obtained as show as below:

Table 4.3: The performance of values of the LQR

Parameters	Value
Rise Time	284.63us
Settling Time	0.50358ms
Overshoot	0
Peak	1.0899
Peak Time	1.7ms

The step response of the LQR system as below:

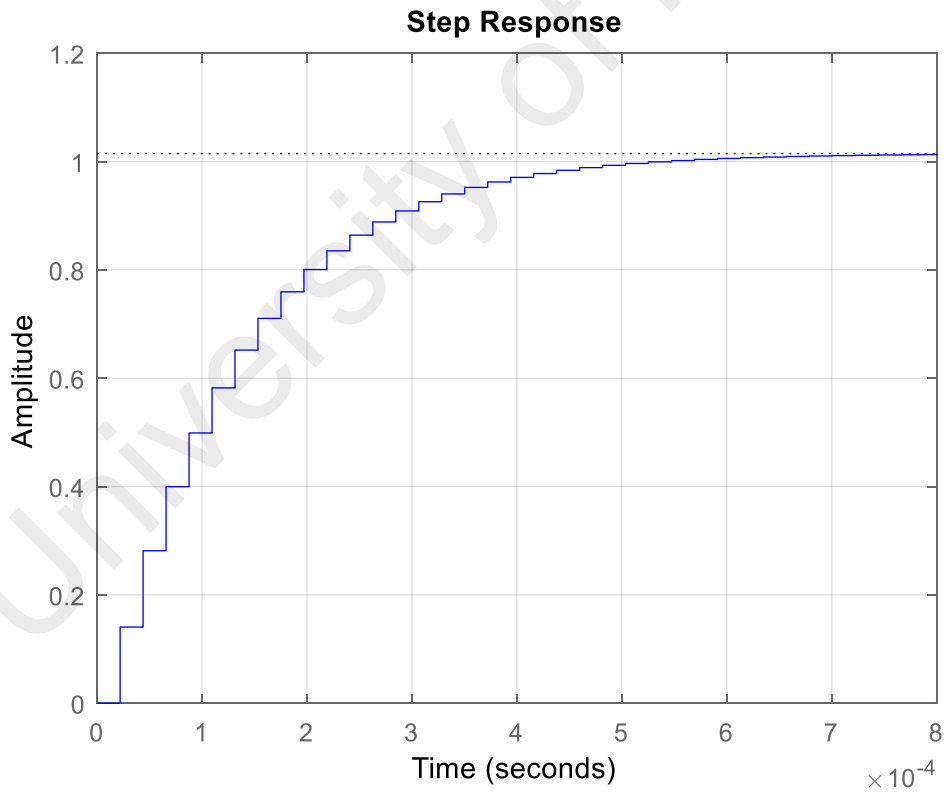


Figure 4.6: The step response of the LQR system

Next LQE (Kalman Filter) design to improve the system robustness. As explained on "Chapter 2: Literature Review" on the Kalman Filter, the design have two disturbances which is "process noise" and "measurement noise". In this case, PLL system process noise is the high frequency NRO baseline noise, and measurement noise is the low frequency range NRO. The input parameters is shows as table below:

Table 4.4: Input parameters for Kalman Filter design

Parameters	Value
A	$\begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}$
C	$[0.3018 \quad 0.04978]$
$\mathbf{V_d}$	$\begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$
$\mathbf{V_n}$	1

With the input parameters set into the "dlqe" it generates the K_F gain for the LQE system. The commands script is shows as below:

```
%kalman filter design
Vd = .1*eye(2);
Vn = 1;
Bf = [dsysPs.b Vd];
Kf = dlqe(dsysPs.a,Vd,dsysPs.c,Vd,Vn);  % design Kalman filter
sysKF = ss(dsysPs.a-Kf*dsysPs.c,[dsysPs.b Kf],dsysPs.c,dsysPs.d);  % Kalman filter estimator
```

Figure 4.7: Matlab commands for Kalman Filter design

The Kalman Filter is tested with white Gaussian noise added into the system, Wn, and output control, Wd as disturbance. The model is tested and results as show as below by plots:

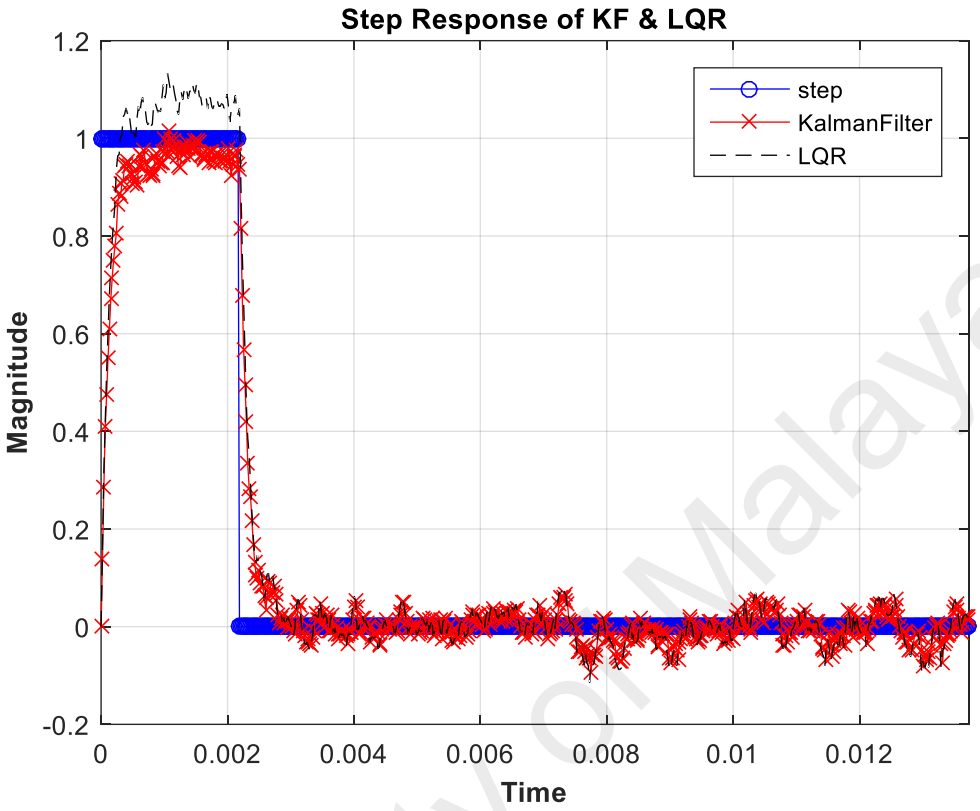


Figure 4.8: Kalman Filter Simulation Plot

4.4 Compare Result on LQG and PID

Both control schemes are tested with step response function using matlab, it yields the plot as below [figure 4.9]. From the plot it shows the PID has overshoot before it settle to the target value, but LQG is gradually rise the magnitude and settle as fast as PID. In term of performance is hard to justify using naked eye, table 4.5 below shows the performance of both controls in step response.

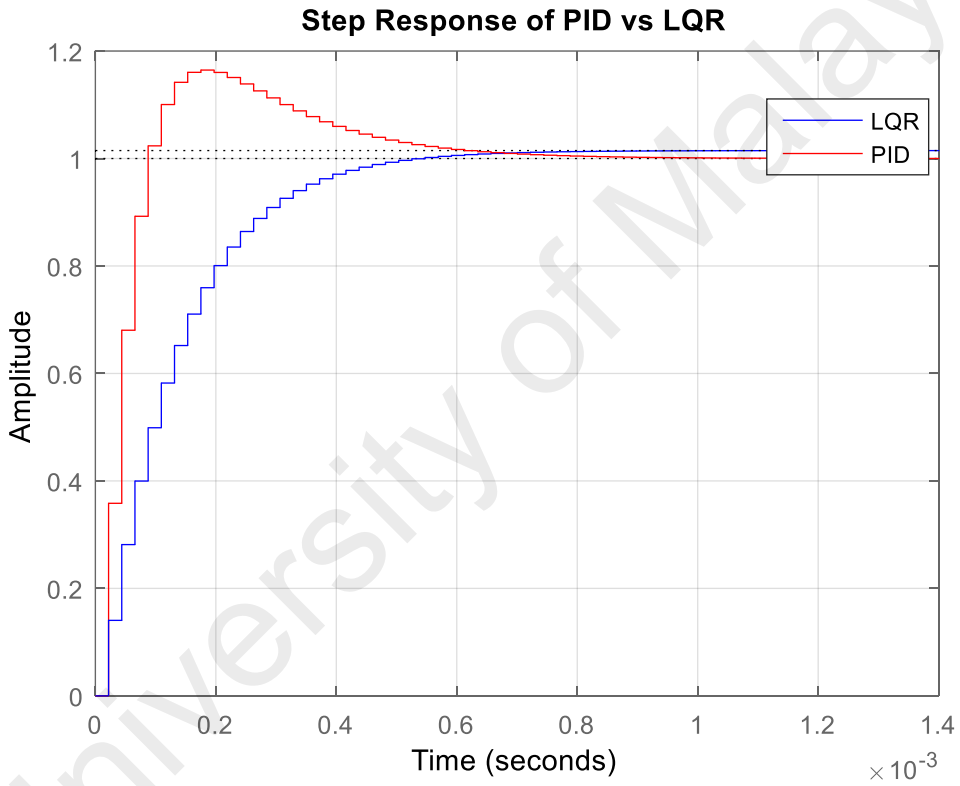


Figure 4.9: Step response of PID and LQG

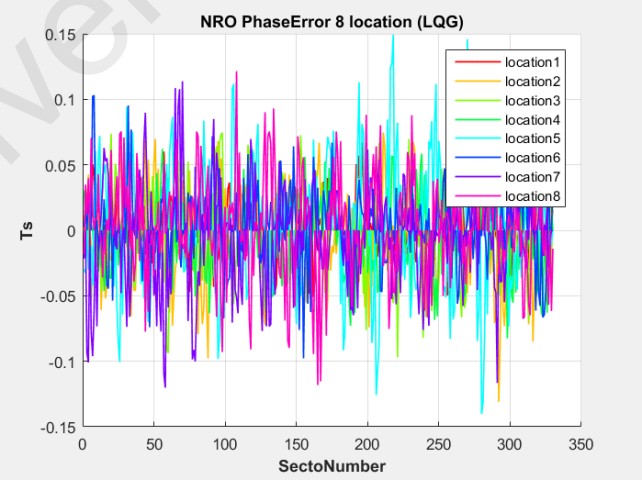
Table 4.5: Performance value of PID Vs LQG

Parameters	PID Value	LQG Value
Rise Time	65.685us	284.63us
Settling Time	**0.56926ms**	**0.50358ms**
Overshoot	16.4620	0
Peak	1.1646	1.0899
Peak Time	0.17516ms	1.7ms

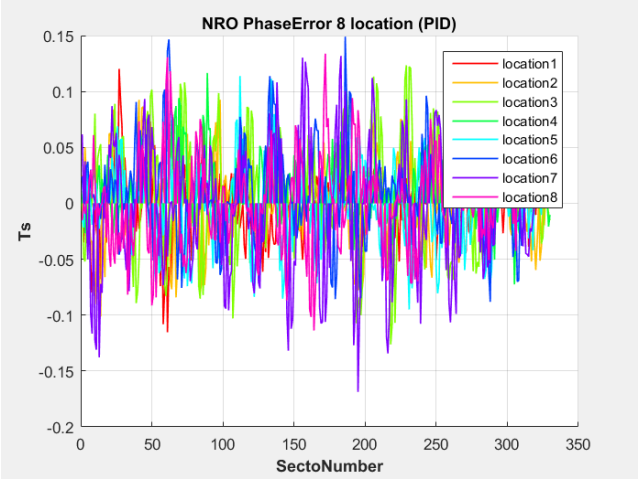
In design aspect, preferable using LQG due overshoot and achieve the target same speed as PID. The LQG control show slightly better in terms on the faster settle than PID. Next it's to analyze the "on-track following" performance on real data.

The following plots are results after implemented the LQG and PID for the PLL system in HDD.

The figure 4.10 (a) and (b) were NRO phase collected across multiple tracks, it is hard to tell any improvement through time domain graph.



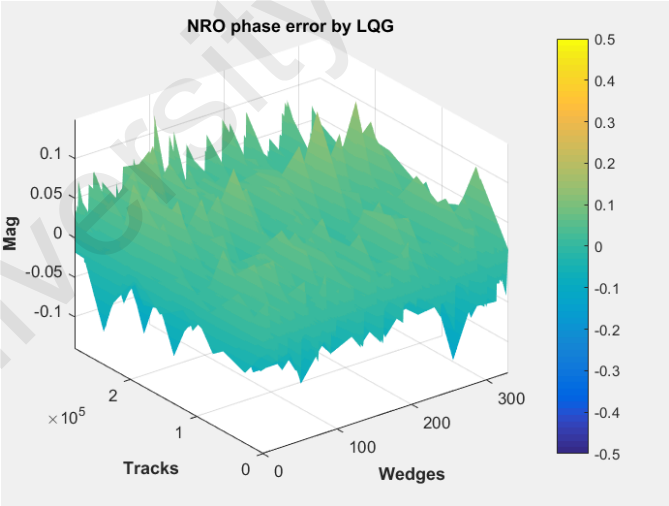
(a)



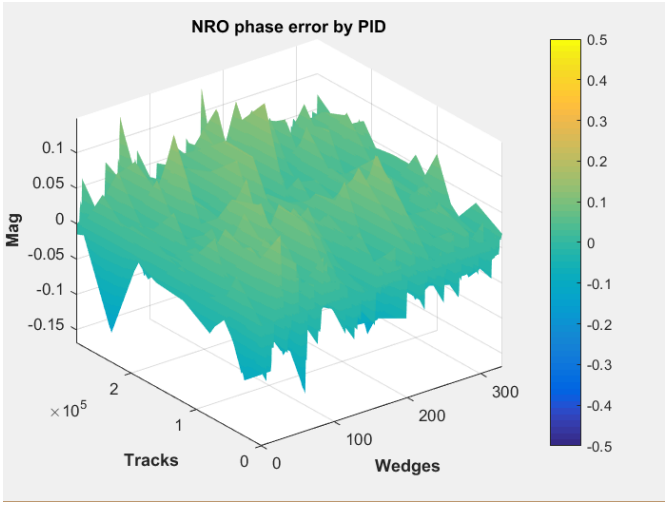
(b)

Figure 4.10: NRO phase error at different location on the disk (a) LQG and (b) PID

Figure 4.11 (a) and (b) are the 3D surface plot using the tracks, servo sectors, and phase error on different tracks. The plot shows no excitation on any location on the disk, either PID or LQG controller.



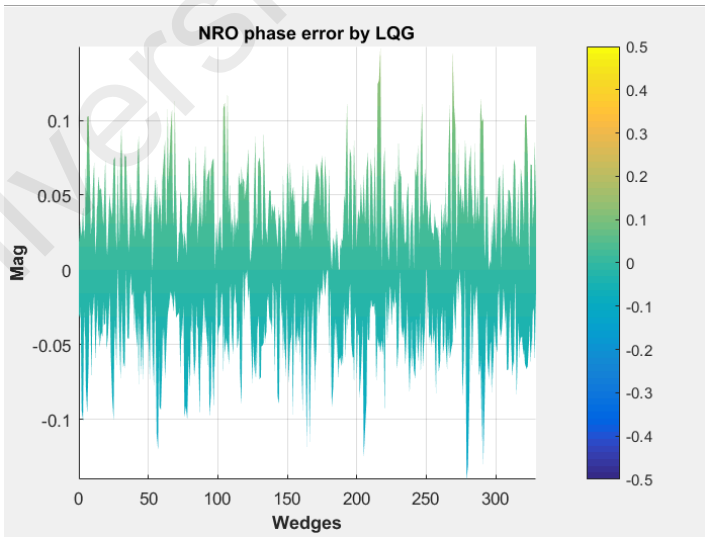
(a)



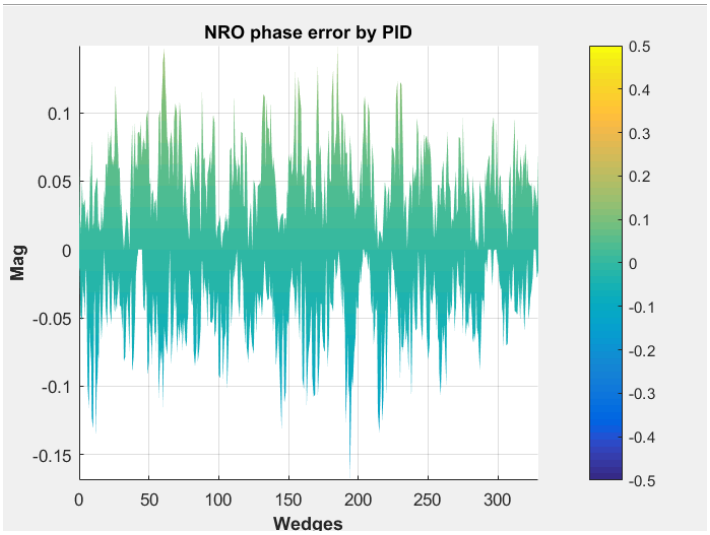
(b)

Figure 4.11: Surface plot for the NRO phase error on different location of the disk. (a) LQG and (b) PID

Surface plot magnitude NRO against servo sectors shows the result both controllers are comparable as shows in figure 4.12 (a) and (b) Plot in time-domain is not suitable to determine the performance of controller in HDD.



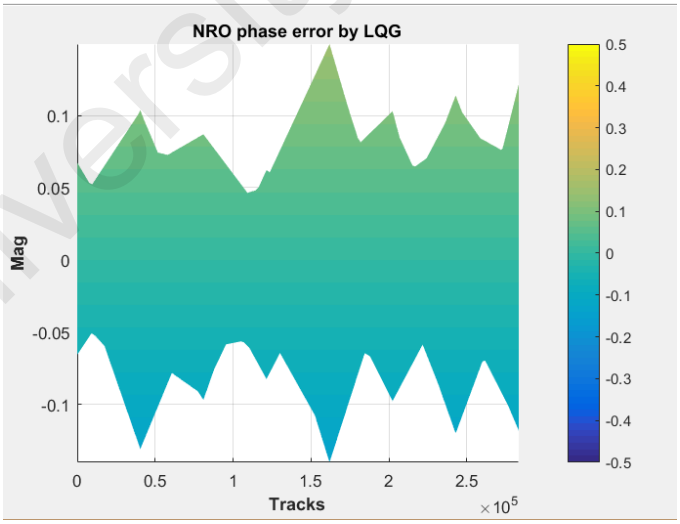
(a)



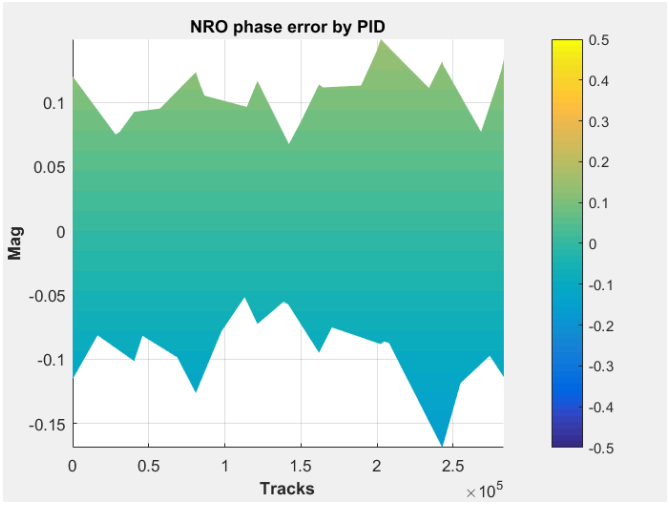
(b)

Figure 4.12: Surface plot view on magnitude vs wedges/servo sectors. (a) LQG and (b) PID

The surface plot magnitude against tracks figure 4.13 (a) and (b) show no sign of excitation on both controllers.

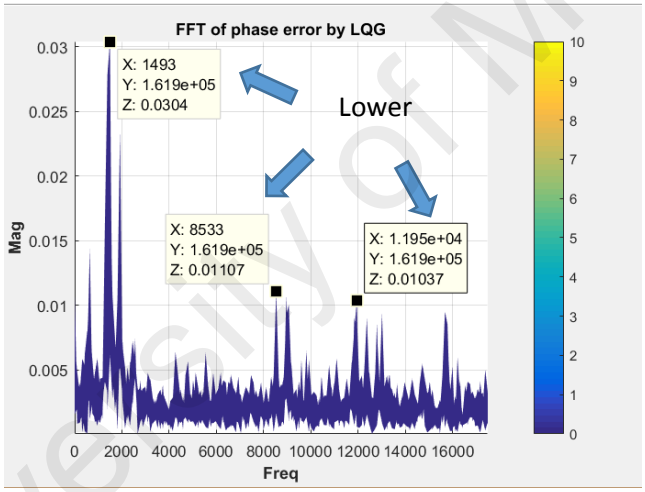


(a)

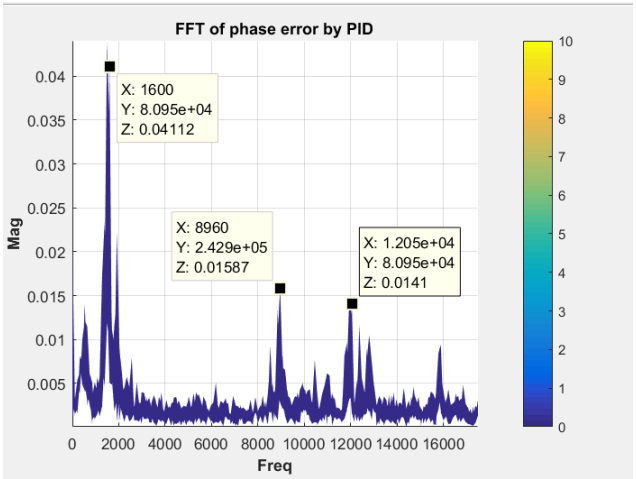


(b)

Figure 4.13: Surface plot view on magnitude vs track. (a) LQG and (b) PID



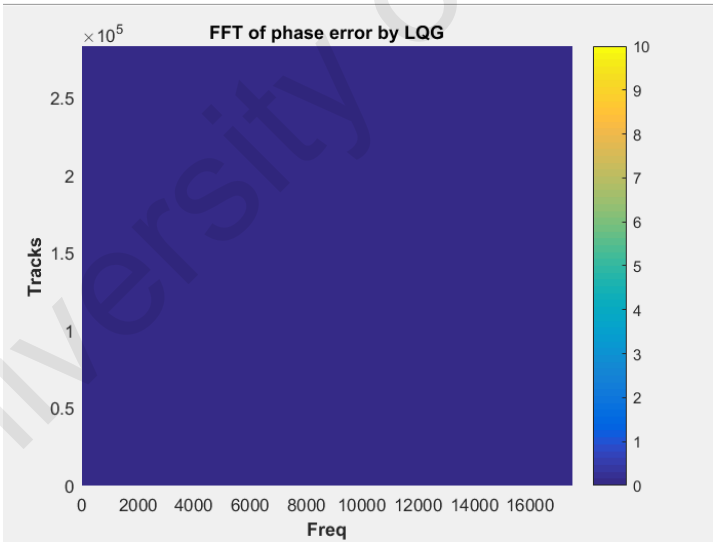
(a)



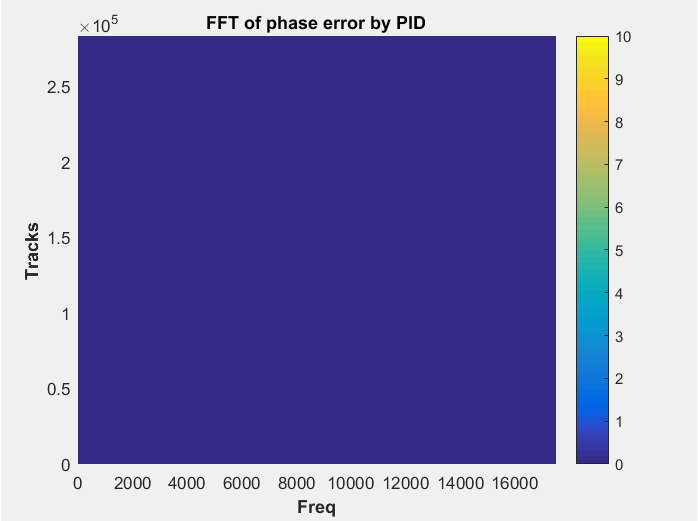
(b)

Figure 4.11: FFT plot NRO phase error view on magnitude vs frequency. (a)LQG and (b)PID

Figure 4.15 (a) and (b) is view from the top of the figure 4.15 tracks against frequency, observed no excitation along the tracks by color map show at the side.



(a)



(b)

Figure 4.15: FFT NRO phase error surface plot view on track vs frequency. (a) LQG and (b) PID

For single drive performance checking it can be done through plots, such as FFT NRO plot [figure 4.15], and 3-sigma NRO plot [figure 4.16]. Figure 4.16 is shows the 3-sigma phase error across the HDD surface, the result shows improvement using LQG. Another method to justify the performance is using a table form rather than plot, because when analysis using plots to analyze the performance for large samples is not a practical way to conclude the quality. The table 4.5 is used for large samples evaluation or production run quality checking guideline. By refer the zone values and mean of the 3-sigma able to tell drive performance.

Table 4.5: 3-sigma NRO phase error for different track location

Drive	Zone1	Zone2	Zone3	Zone4	Zone5	Zone6	Zone7	Zone8	Mean
LQG	0.0629	0.1063	0.0913	0.0804	0.1405	0.0840	0.1121	0.1133	0.0988
PID	0.0925	0.1296	0.1733	0.1132	0.1866	0.1237	0.1788	0.1785	0.1470

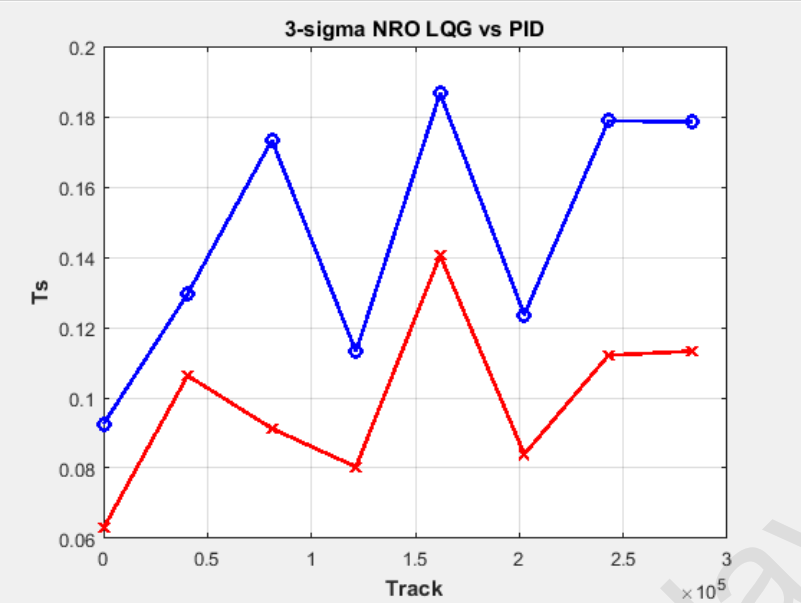


Figure 4.16: 3 sigma of NRO phase error of each location of HDD. (a)LQG and (b) PID

From the experiment data shows that the **LQG performance on rejecting the NRO phase error better than PID approximately 30%** on overall tracks. There are few fall back for using LQG controller in practical case.

- The Q and R weighting is hard to balance by reference the step response plot.
- The model is too simple to represent the system, which cause the LQG need experiment data to weight the Q and R.
- Iterative method together with experiment data to get optimum weighting.
- Indeed, with an arbitrary choice of weighting matrices, the classical state-feedback optimal regulators seldom show good set-point tracking performance due to the absence of integral term unlike the PID controllers.

Although the experiment results show PID performance slightly lower than LQG, but using PID controller it give some advantages to the designer as below:

- PID parameters easy to tune with graphical methods, such as bode plot, step response, Nyquist plot, and pole-zero mapping.
- Without much information from the system, PID implementation can be done just based on system tracking error.

CHAPTER 5: CONCLUSION & FUTURE WORK

5.1 Conclusion

The main objectives of the experiment is to analyze the performance and improve the overall 3-sigma NRO phase error using either PID or LQG controller are successfully demonstrated. The analysis in "Chapter 4: Result" shows LQG has better performance compare to PID control during the HDD operational mode, the performance in terms of steady error and system response. The results show **reducing ~30% in 3-sigma NRRO phase error, and faster response time ~11%** to support the statement that made.

The final objective is to reduce the cycle time on the HDD operational mode is proven successful as well. The cycle time is affected by the capability rejecting the NRRO phase error and response time during HDD operational mode. When NRRO phase error is poor during the operation, the phase error hit the threshold and it will cause retry on the process. The LQG shows that all objectives have significant improvement, as response time reduced it reduced the time from seek to write as well, which means the cycle time able to reduce accordingly.

However, PID is preferable choice of many industry due to its simplicity of implementation.

5.2 Future Work

The process selecting the weight of Q and R for LQG is a tedious and time consuming process. The iterative process need to balance the weighting on Q and R to achieve the optimum performance can be improve using optimization method such as PSO (Particle Swarm Optimization), and GA (Generic Algorithm).

REFERENCE

[1]. Lawrence M. Bryant, Yakov M. Lifchits and Siri S. Weerasooriya. (2008). "Disk drive self-servo writing spiral tracks by propagating bursts" Abstract U.S Patent 7688539B1. Washington, DC: U.S. Patent and Trademark Office.

[2]. Udeh Tochukwu Livinus. (2014). "Effect of A PID Controllroller in Closed Loop Feedback System". Ternopil National Technical University, Ukraine.

[3]. Tore Hagglund. (2000). "Signal Filtering in PID Control". Department of Automatic Control, Lund Univesity, Lund, Sweden.

[4]. R.E. Kalman. (1960), "On the general theory of control systems," in Proc. First Internat. Congress Automat. Contr., pp. 148 - 149, Moscow.

[5]. G Da Prato, I Lasiecka, R Triggiani (1986). "Journal of Differential Equations". URL: https://www.sciencedirect.com/science/article/pii/0022039686900690

[6]. J. Nie and R. Horowitz. (2009). "Design and Implementation of Dual-Stage Track-Following Control for Hard Disk Drives," Proceedings of the 2nd Dynamic Systems and Control Conference, Hollywood, California.

[7]. Su Chen and Masoyoshi Tomizuka (2011). "Spiral Servo Writing in Hard Disk Drives Using Iterative Learning Based Tracking Control". The International Federation of Automatic Control, Milano, Italy.

[8]. Luan Ton-that. (1994). "Data Track Pattern Including Embedded Servo Sectors for Magneto-Resistive Read/Inductive Write Head Structure for A Disk Drive". Abstract U.S Patent US08703561. Washington, DC: U.S. Patent and Trademark Office

[9]. Daniel Abramovitch and Cne Franklin. (2002). "Disk Drive Control: The early years". URL https://www.sciencedirect.com/science/article/pii/S1367578802000287.

[10]. Nestor O. Perez Arancibia, Chi-Ying Lin, Tsu-Chin Tsao, and James S. Gibson. (2007). "Adaptive-Repetitive Control of A Hard Disk Drive". URL https://ieeexplore.ieee.org/document/4434235/

[11]. John Leonard Wallberg. (1997). "A Digital Phase-Locked Loop For Hard Disk Drive Read Channel Applications". URL https://dspace.mit.edu/bitstream/handle/1721.1/42726/38137568-MIT.pdf?sequence=2. Pages 11 – 16.

[12]. Roland E. Best. (2003). "Phase-Locked Loops- Design, Simulation, and Application". Fifth Edition, McGraw-Hill. Page 7 – 26

[13]. Charles L. Philips and H. Troy Nagle. (1995). "Digital Control System Analysis and Design". Third Edition, Prentice Hall. Page 27 – 76

[14]. Donald E. Krik. (1998), "Optimal Control Theory- An Introduction". Englewood Cliffs, N.J. Prentice Hall. Page 53 - 90

[15]. Lawrence M. Bryant, Yakov M. Lifchits and Siri S. Weerasooriya (2008). "Disk drive self-servo writing spiral tracks by propagating bursts" Abstract U.S Patent 7688539B1. Washington, DC: U.S. Patent and Trademark Office.

[16]. J. Nie, (2009) "Control Design and Implementation of Hard Disk Drive Servos," University of California, Berkeley.

[17]. A. Mamun, G. Guo, and C. Bi. (2006). "Hard disk drive: Mechatronics and Control". Boca Raton, FL: CRC.

[18]. Lawrence M. Bryant, Yakov M. Lifchits and Siri S. Weerasooriya (2008). "Disk drive self-servo writing spiral tracks by propagating bursts" Abstract U.S Patent 7688539B1. Washington, DC: U.S. Patent and Trademark Office.

[19]. Dr. Lauro Rizzatti. (2016). "Digital Data Storage is Undergoing Mind-Boggling Growth." URL https://www.eetimes.com/author.asp?section_id=36&doc_id=1330462

[20]. D. Abramovitch and G. Franklin (2002). “A brief history of disk drive control”. IEEE Control Systems Magazine, 22(3):28–42

[21]. Su Chen and Masoyoshi Tomizuka (2011). “Spiral Servo Writing in Hard Disk Drives Using Iterative Learning Based Tracking Control”. The International Federation of Automatic Control, Milano, Italy.

[22]. R. E. Kalman. (1960), “A new approach to linear filtering and prediction problems,” Journal of basic Engineering, vol. 82, no. 1, pp. 35–45,

[23]. Zhi Wan, JengHung, Chen, David Dung, and Tien Nguyen. (2004). ”Disk drive estimating a sinusoidal error in a wedge time period due to eccentricity in disk rotation”. Abstract U.S Patent 7068451B1. Washington, DC: U.S. Patent and Trademark Office.