Robust Servo Control Design for Self-Servo-Write Hard Disk Drives with Non-Uniform Samples of Position Error Signal

Behrooz Shahsavari, Richard Conway

University of California, Berkeley
Computer Mechanics Lab.

Adviser:
Professor Roberto Horowitz

January 2015

Contents

List of Figures

List of Tables

Chapter 1

abstract

This paper considers robust controller design for track-following in hard disk drives (HDD) with irregular sampling of the position error signal (PES) but regular (clock-driven) control updates. This sampling and actuation behavior is modeled by applying a novel discretization method to a continuous-time model of a HDD, resulting in a discrete-time linear periodically time-varying model. Then, the controller design is performed using optimal H_∞ control for periodic systems, and uses a generalization of the disk margin criterion to quantify the robustness of the closed-loop system. To show the effectiveness of the proposed method, the design methodology is applied to a hard disk drive model and the resulting controller is validated by examining its nominal performance in terms of the root mean square of the standard deviation of the PES, and robustness in terms of disk margin. Since the proposed controller has too many parameters to be implementable on an HDD due to memory limitations, we use a vector quantization method to approximate the entire parameters set of the designed controller by a smaller set of parameters.

Chapter 2

Introduction

Sampling time in hard disk drive (HDD) servo systems is not always constant over a revolution of the disk. The position of the read/write head is attained when it passes over the servo sectors written on the disk **al2007hard** Hence, when the servo sectors are not placed equidistantly on the tracks the sampling will be time-varying during one revolution of the disk. However, since the disk is rotating, the irregularity in sampling intervals will be N-periodic, where N is the number of servo sectors on each track. There are different factors that can make the sampling time irregular. For instance, when the center of servo tracks does not exactly coincide with the center of disk rotation, there can be large variations in the sampling rate. Another factor resulting in irregular sampling rate is the existence of missing sectors as a result of false PES demodulation .**ehrlich2005methods** Furthermore, during the manufacturing and testing processes of some HDDs, PES sampling rates can be more than 10 times that of the normal servo loop bandwidth **guo2010servo** and, because track accuracy is not strongly gated, this may result in the servo system missing half of the PES samples.

When HDDs areal density is low, ignoring the variation in sampling rate and using a linear time invariant (LTI) controller usually results in acceptable performance and robustness **hirata1992head** However, the areal density of HDDs has had 40% annual increase during the last 6 years, and is expected to reach 2.5 Tb/in^2 by 2014, which will require writing data on 19×13.5 nm^2 bit cells **fontana2012technology** For this high areal density, ignoring the sampling time irregularity and utilizing an LTI controller might not lead to the desired level of performance and robustness **shahsavari2013limits** Therefore, a design methodology that deals with irregularity in the sampling rate is necessary. Two meaningful schemes for updating the control signal can be considered for a system with irregular sampling **shahsavari2013limits** In the first scheme, the controller is event-driven, i.e. the control is updated as quickly as possible after receiving a new measurement. In the second scheme, the controller is clock-driven and the control action is updated at a regular rate regardless of the sampling time variation. As illustrated in **shahsavari2013limits** a system with irregular but periodic sampling time can be modeled as a linear periodically time-varying (LPTV) system and exploiting the latter scenario results in better limits of performance in such a system. Moreover, since the control action is usually passed through one or more oversampled notch filters before being applied to the actuator(s), in order to prevent the excitation of actuator high frequency resonance modes **semba2001method** the second control scheme is more attractive because it decouples the design of the notch filters from the sampling rate variation. Due to these two significant advantages, our design methodology is based on the latter control scheme. Using this control method, there might be arbitrary number of measurements between two successive control updates **shahsavari2012robust** In section 4 it is presented how our proposed controller accommodates the varied number of measurements between successive control updates.

It is worth noting that this type of servo mechanism can be considered as a network control system (NCS) with two main sources of delay. Firstly, updating the control signal may be computationally expensive and time consuming. Secondly, since the control update and sampling time are not synchronized, there usually exists time differences between the measuring instants and the scheduled control update instants. NCSs with event-driven (irregular) control are considered in **yu2005sampled** where it is assumed that the control is updated at the sampling instants. The same control update scheme for NCSs when the control is updated with a delay after each sampling is considered in **gao2008new** In both of these works it is assumed that the sampling is clock-driven and the case when the feedback signal arrives irregularly is not considered.

Performance and robustness are the most important specifications that should be considered in control design for HDDs. In particular, since the resulting controller must achieve a high level of performance when applied to any unit in the product line, it must have sufficient robustness margins. Traditional H_∞ control design techniques for linear time invariant (LTI) systems can attenuate error well even under a set of plant variations **huang2005robust** One goal of this paper is to extend the H_∞ control design methodology to systems with periodic irregular sampling. In **nie2012optimal** the H_∞ control problem for a HDD with irregular sampling rate caused by missing PES data is considered. The time interval between any two subsequent successful measurements in **nie2012optimal** is a positive integer multiplier of the nominal sampling time, while in this work it can be any positive value. Therefore, the control design proposed in that work can only be exploited for a specific type of sampling time irregularity (e.g. the results of **nie2012optimal** cannot be utilized for a HDD having disk eccentricity, in which the sampling time is a trigonometric function of time). To quantify the robustness of our closed-loop system, which will be modeled as a LPTV system, we extend the idea of disk margins **blight1994practical** to LPTV systems by proposing a novel idea based on H_∞ norm of a modified open loop system. We then maximize the disk margin of the closed loop system by choosing a few shaping functions and solving an optimal H_∞ control problem for the LPTV model of the servo system.

It is illustrated in **aruga2007study** that almost all disturbances in higher rpm HDDs can be reduced by sealing the drive and filling it with a light weight gas (e.g. Helium). This advantage of using light weight gases has guided the HDD industry toward producing Helium filled drives. The H_∞ control problem presented in this work includes performance weighting functions which are usually chosen based on the system noise model. Therefore, the proposed control design method can be exploited for the next generation of HDDs by knowing basic information about their noise model. Furthermore, it is shown in the paper that the complexity of the controller depends on the complexity of the performance weighting functions. Hence, using light weight gases can benefit this control design since smaller disturbances can be ignored or modeled with simpler dynamics.

Chapter 3

Modeling of the HDD

In this section, we will describe a sampling and actuation scheme presented in **shahsavari2013limits** in which the time interval between two subsequent successful measurements is time-varying while the control update rate is constant (i.e. the controller is clock-driven). Since the servo sectors are written during the manufacturing process, the time interval between reading any two particular consecutive servo sectors is constant, but it changes from sector to sector. Hence, the time intervals between two consecutive samplings can be known and used in control design. Throughout this section, the continuous-time state space model of the system is given by

$$\dot{x}_c(t) = A_c x_c(t) + B_c u_c(t) \tag{3.1}$$

$$y_c(t) = C_c x_c(t) + D_c u_c(t). \tag{3.2}$$

The signals, $x_c(t)$, $u_c(t)$, and $y_c(t)$ respectively have dimension n_x, n_u, and n_y. We assume that an event-driven zero-order holder (ZOH) is used to hold the output of the discrete-time controller for the plant. For a positive real value of period, say T, the state dynamics matrices for the discrete time system with period T are

$$A_d(T) = e^{A_c T}, \quad B_d(T) = \left(\int_0^T e^{A_c \tau} d\tau \right) B_c. \tag{3.3}$$

We denote the time that the control signal will be updated at time step k as $t_{u,k}$. Assuming that the control update rate is regular with period T, and $t_{u,0} = 0$, the control update instances, $t_{u,k}$, can be characterized as

$$t_{u,k} = kT, \quad k \in \mathbb{Z}. \tag{3.4}$$

The computational delay associated with the controller will be denoted as δ, and we will assume for simplicity that $\delta < T$. In particular, since the controller has to update the control signal at scheduled moments rather than as soon as receiving a feedback signal, we need to treat the computational delay as the constraint that the controller cannot use the measurements arriving in the time interval $(t_{u,k}-\delta,\infty)$ in the calculation of u_k. Equivalently, the value of u_k will be updated at time step $k+1$ by measurements that arrive in the time interval

$$S_k := (t_{u,k}-\delta, t_{u,k+1}-\delta]. \tag{3.5}$$

We now define

$$\begin{aligned} x_k &:= x_c(t_{u,k}) & (3.6)\\ u_k &:= u_c(t_{u,k}). & (3.7) \end{aligned}$$

Since the controller is clock-driven and it updates the control signal regularly, we can discretized the state dynamics of (3.1) as

$$x_{k+1} = A_d(T)x_k + B_d(T)u_k. \tag{3.8}$$

It is noteworthy that the state dynamics of the discrete-time system is time invariant. In other words, since the states and control of the system are updated regularly in time, regardless of the irregularity in the sampling time, the matrices representing the state dynamics are time-invariant. It will be shown later that this LTI dynamics results in requiring less memory for storing the controller parameters. We now find the representation of a measurement at a time instant $\bar{t} \in S_k$. There are two cases to consider. The first case corresponds to $\bar{t} \geq t_{u,k}$. In this case, we obtain

$$\begin{aligned} y(\bar{t}) &= C_c x_c(\bar{t}) + D_c u_c(\bar{t}) \\ &= [C_c A_d(\bar{t}-t_{u,k})]x_k + [C_c B_d(\bar{t}-t_{u,k}) + D_c]u_k. \end{aligned} \tag{3.9}$$

The second case corresponds to $\bar{t} < t_{u,k}$. In this case, we note that

$$\begin{aligned} x_c(\bar{t}) &= A_d(\bar{t}-t_{u,k-1})x_{k-1} + B_d(\bar{t}-t_{u,k-1})u_{k-1} \\ &= A_d(\bar{t}-t_{u,k-1})A_d^{-1}(T)[x_k - B_d(T)u_{k-1}] \\ &\quad + B_d(\bar{t}-t_{u,k-1})u_{k-1}. \end{aligned} \tag{3.10}$$

For notational convenience, we define

$$\begin{aligned} \bar{A} &:= A_d(\bar{t}-t_{u,k-1})A_d^{-1}(T) \\ \bar{B} &:= B_d(\bar{t}-t_{u,k-1}) - A_d(\bar{t}-t_{u,k-1})A_d^{-1}(T)B_d(T) \end{aligned}$$

so that the previous expression can be written as

$$x_c(\bar{t}) = \bar{A}x_k + \bar{B}u_{k-1}.$$

This yields

$$\begin{aligned} y(\bar{t}) &= C_c x_c(\bar{t}) + D_c u_c(\bar{t}) \\ &= C_c\bar{A}x_k + C_c\bar{B}u_{k-1} + D_c u_k. \end{aligned}$$

Since this expression depends on u_{k-1}, we need to augment the state vector, i.e. we write the state dynamics of the discrete-time system as

$$\begin{bmatrix} x_{k+1} \\ u_k \end{bmatrix} = \hat{A} \begin{bmatrix} x_k \\ u_{k-1} \end{bmatrix} + \hat{B} u_k \tag{3.11}$$

$$\hat{A} := \begin{bmatrix} A_d(T) & 0 \\ 0 & 0 \end{bmatrix}, \quad \hat{B} := \begin{bmatrix} B_d(T) \\ I \end{bmatrix}.$$

Again, these dynamics are LTI and do not depend on the measurement characteristics of the system. The output at time instant $\bar{t} \in S_k$ corresponds to

$$y(\bar{t}) = \bar{C} \begin{bmatrix} x_k \\ u_{k-1} \end{bmatrix} + \bar{D} u_k \tag{3.12}$$

where

$$\hat{C}(k) = \begin{cases} [C_c \bar{A}, C_c \bar{B}], & \bar{t} \in (t_{u,k} - \delta, t_{u,k}) \\ [C_c A_d(\bar{t} - t_{u,k}), 0], & \bar{t} \in [t_{u,k}, t_{u,k+1} - \delta] \end{cases} \tag{3.13}$$

$$\hat{D}(k) = \begin{cases} D_c, & \bar{t} \in (t_{u,k} - \delta, t_{u,k}) \\ C_c B_d(\bar{t} - t_{u,k}) + D_c, & \bar{t} \in [t_{u,k}, t_{u,k+1} - \delta] \end{cases} \tag{3.14}$$

Although (3.13) and (3.14) describe the output at an arbitrary time instance, these relationships do not fully describe the system output corresponding to a given time index. In particular, since there is no fixed relationship between the times at which measurements are obtained and the times at which the control is updated, the number of samples in the time interval S_k is not necessarily constant over k. For simplicity, we will consider a situation in which 0, 1, or 2 measurements may be made in any time interval S_k. We thus have three cases to consider. In all three cases, we will force the discrete-time model to have two outputs. We begin by considering the case when, for a particular value of k, there are two measurements made in the time interval S_k; we denote the time instances corresponding to these measurements as $\bar{t_1}$ and $\bar{t_2}$. In this case, we choose

$$y_k = \begin{bmatrix} y(\bar{t_1}) \\ y(\bar{t_2}) \end{bmatrix}. \tag{3.15}$$

Note that y_k captures all of the information that the controller can use to update the value of u_k to u_{k+1}. We now consider the case when, for a particular value of k, there are no measurements made in the time interval S_k. In this case, the controller should accept no inputs at time step k. Equivalently, the input into the controller should be zero. This motivates choosing

$$y_k = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{3.16}$$

Choosing this form for y_k allows the system to have a time-invariant structure; y_k is acting here as a placeholder so that the discrete-time model has two outputs, even at time steps when the controller has no inputs. Finally, we consider the case when, for a particular value of k, there is one measurement made in the time interval S_k; we denote the time instance corresponding to this measurement as $\bar{t}$. In this case, we choose

$$y_k = \begin{bmatrix} y(\bar{t}) \\ 0 \end{bmatrix}. \tag{3.17}$$

As in the previous case, we are using zeros as placeholders so that the discrete-time model has two outputs, even at time steps when the controller only has only one input. Under the assumption that the sampling of the system is periodic, (3.11)-(3.17) define an LPTV discrete-time state space model. This model describes the continuous-time model (3.1)-(3.2) under a zero-order hold on the input with sampling and actuation conditions described at the beginning of this section. However, there is one subtle detail that remains: with the convention we have chosen for defining y_k, we cannot use any arbitrary control scheme to control this model. In particular, for continuous-time causality to hold, u_k can only depend on, y_{k-1}, y_{k-2}, ..., y_0, i.e. a discrete-time controller satisfies continuous-time causality if and only if it is strictly causal in the discrete-time domain.

Chapter 4

Control Design

In this section, we present an optimal H_∞ control design for a HDD model that is discretized by using the method discussed in section 3. Our control objective throughout this paper is to maximize the stability margins while keeping the position error signal, PES, as small as possible, in order to achieve high areal densities and low-readout error rates. We first define the metrics for performance evaluation of a LPTV system and specifically a HDD that is discretized as a LPTV system, and then proposes a method for robustness evaluation of such a system. Finally, the architectures used for control design and performance calculation are presented.

Performance Analysis

If the entire system is adequately modeled as a stochastic system, i.e., all external disturbances can be considered as random signals with Gaussian distribution, the tracking performance is normally characterized by the 3σ value of the PES **huang2005robust** When all the disturbance sources are normalized through proper weighting functions, this root-mean-square (RMS) value is then equivalent to the H_2 norm of the transfer function from those normalized disturbances to the PES, i.e.,

$$RMS(PES) = \|G_{d\to PES}\|_2$$

where $G_{d\to PES}$ is the transfer function from the Gaussian input disturbance to the PES. For a LPTV system although the output signals will not be stationary, the second-order statistics will be periodic, with period N. Hence, the 3σ value of PES at each sector of a HDD modeled as a LPTV system is constant and can be computed by

$$\sigma_k(PES) = \|G^k_{d\to PES}\|_2, \quad k \in \{1, \cdots, N\}$$

where $G^k_{d\to PES}$ is the closed loop system from N input disturbance vectors during one revolution to the PES at sector k. A good way to capture the performance of a controller in a

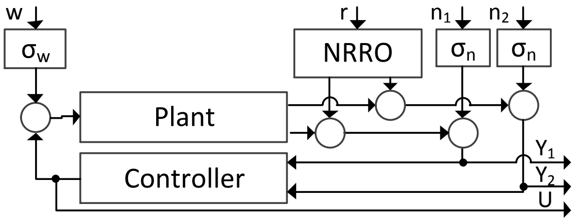

Figure 4.1: Closed loop system for performance calculation.

HDD modeled as a LPTV system is to compute the RMS and maximum 3σ values of PES over a revolution of disk

$$RMS(PES) = \underset{k=1,\cdots,N}{RMS}[\sigma_k\,(PES)] \tag{4.1}$$

$$\sigma_{max} = \max_{k\in\{1,\cdots,N\}}[\sigma_k\,(PES)]. \tag{4.2}$$

Equations (4.1)-(4.2) can be computed by using the solution of a periodic Lyapunov equation. Fig. 4.1 shows a block diagram of the closed-loop system with all relevant disturbances. The signals Y_1, Y_2, U, r, w, n_1 and n_2 are respectively first and second measurement signals, control signal, independent white noises with unit variance that respectively model the effect of the non-repeatable run-out (NRRO), windage, and measurement noise for the two measurement signals. NRRO is the random lateral movement of the disk caused by the mechanical contacts in the bearing motor, and windage is the off track motion at the head caused by the turbulent nature of the air between the disk and the actuator **al2007hard** The NRRO, σ_n and σ_w blocks in Fig. 4.1 are respectively the NRRO model, the standard deviation of measurement noise, and the standard deviation of the windage. We use the same idea as section 3 to discretize the NRRO model, i.e. we find a continuous-time model for the NRRO and then discretize it as a LPTV system. Accordingly, the NRRO model has two outputs similar to the plant model.

Control Design

To quantify the robustness of the closed-loop systems, we would like to use gain and phase margins. However, since computing the gain and phase margins of a LPTV system is difficult (or impossible), we quantify the robustness of the system by generalizing an idea known as disk margin **blight1994practical** For LTI systems, the disk margin is a more rigorous measure of robustness than gain and phase margins because it quantifies how close the open-loop frequency response (e.g. the Nyquist plot) comes to -1 in all directions, rather than just along the real axis and the unit circle (as quantified respectively by the gain and phase margins). For a particular plant and a designed controller the disk margin can be found by adding a new input and output signal to the system and calculating the H_∞ norm from the new input to the new output channel. This idea is shown in Fig. 4.2. In this figure,

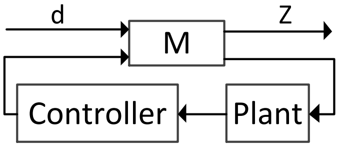

Figure 4.2: Modified close loop system for disk margin calculation.

the block M is a matrix given by

$$M := \begin{bmatrix} 1 & \sqrt{2} \\ \sqrt{2} & 1 \end{bmatrix}.$$

This matrix M is included in the block diagram so that the H_∞ norm from d to Z represents the disk margin of the closed-loop system by the following relations

$$DGM = \begin{cases} \infty & \beta \leq 1 \\ (\beta+1)/(\beta-1) & 1 < \beta < \infty \\ 1 & \beta = \infty \end{cases} \tag{4.3}$$

$$DPM = \begin{cases} \frac{\pi}{2} & \beta \leq 1 \\ \cos^{-1}\left(\frac{\beta^2-1}{\beta^2+1}\right) & 1 < \beta < \infty \\ 0 & \beta = \infty \end{cases} \tag{4.4}$$

where DGM, DPM and β are respectively the disk gain and phase margins of the closed loop system, and the H_∞ norm from d to Z. Since the computation of the disk margin only requires the computation of a single H_∞ norm, the concept and method of computation easily generalizes to LPTV systems.

The architecture we use for control design is shown in Fig. 3. The blocks W_{p1}, W_{p2}, W_u and τ are design parameters; they are respectively the performance weighting functions for the sensitivity functions corresponding to the first and second measurements, a control effort weighting value, and a static uncertainty scaling parameter, as appears in the D-K iteration heuristic for μ-synthesis. Note that the H_∞ norm from d_3 to z_3 represents the disk margin of the closed-loop system.

According to (4.3) and (4.4) minimizing the H_∞ norm from d_3 to z_3 maximizes the disk margins of the closed loop system. However, we need to keep the performance of the system larger than predefined values. In other words, we want to satisfy the following two constraints

$$3\sigma^{PES} \leq 3\sigma^{req}$$
$$3\sigma^{PES}_{max} \leq 3\sigma^{req}_{max}$$

where $3\sigma^{PES}$, $3\sigma^{PES}_{max}$ are the RMS and maximum 3σ values of PES, and $3\sigma^{req}$ and $3\sigma^{req}_{max}$ are the required values, e.g. the RMS and maximum 3σ values achieved by existing controllers.

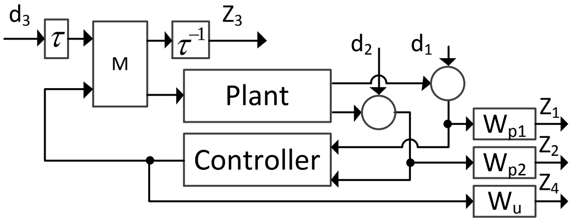

Figure 4.3: Block diagram for control design.

Finally, we can formulate the optimization problem that aims to maximize the robustness subject to the aforementioned constraints as

$$\min_{K} \quad \|G_{cl}(K)\|_\infty \tag{4.5}$$

$$\textit{subject to}$$

$$3\sigma^{PES} \leq 3\sigma^{req} \tag{4.6}$$

$$3\sigma_{max} \leq 3\sigma_{max}^{req}$$

where K and $G_{cl}(K)$ are the controller and the closed loop system shown in Fig. 4.3. Since the solution of the optimization problem strongly depends on the design parameters (e.g. weighting functions) we use the following 3-step algorithm to find the desired controller

1. Choose design parameters (W_{p1}, W_{p2}, W_u and τ)
2. Solve (4.5) (ignore the constraints)
3. If (4.6) is
   a) satisfied: exit and use the K found in 2.
   b) not satisfied: return to step 1.

Once proper values have been chosen for all design parameters in step 1, in step 2 the controller can be designed using optimal H_∞ control in the following form

$$A_K(k) = \hat{A} - \hat{B}\begin{bmatrix} L_{B1}(k) \\ L_{B2}(k) \end{bmatrix} - L_c(k)\hat{C}(k) \tag{4.7}$$

$$B_K(k) = L_c(k), \quad C_K(k) = -L_{B2}(k), \quad D_K(k) = 0$$

where $A_K(k)$, $B_K(k)$, $C_K(k)$ and $D_K(k)$ are state space matrices of the controller. Matrices $L_{B1}(k)$, $L_{B2}(k)$ and $L_c(k)$ can be calculated by the solution of periodic Riccati equations **nie2012optimal**

Controller Parameters Order Reduction

The control synthesis algorithm described in the previous section results in a periodically time varying controller with a large period. The total number of sets of controller matrices (given in (4.7)) to be stored in the memory of servo controller is equal to the number of servo sectors in one data track (mostly between 300 and 400 in current HDDs). In a physical HDD, there is a limited amount of memory available to store the controller parameters and it is almost impossible to reserve so much memory for storing all of these values. As a result, the current designed controller is not directly suitable for implementation.

In this paper we introduce two different approaches to reduce the number of controller parameters required to be stored in memory. The first approach is introducing an innovative discretization method by making the control action regular, which eliminates the time variation in controller parameters $\hat{A}$ and $\hat{B}$. The second approach is to use a vector quantization technique to approximate time-varying controller parameters by a reduced set of matrices, i.e. by replacing $\hat{C}(k)$, $L_{B1}(k)$, $L_{B2}(k)$, and $L_c(k)$ with approximate values.

Vector quantization is a well-known and efficient algorithm in signal processing, which is based on the competitive learning paradigm. It works by dividing a large set of points (vectors) into groups. Each group is represented by its centroid point, as in k-means and some other clustering algorithms.

The basic idea of the algorithm which we have used to approximate the time varying parameters is as follows: the mean of the data set (e.g. the first column of $L_c(k)$) is obtained, and is split into two points, which will respectively be the centroids of two new clusters. The Euclidean distance of each point from these centroids is calculated, and each one is associated with the cluster having the closest centroid. The centroid of each cluster is subsequently replaced by the mean of the vectors in the cluster. If the total distance of vectors from the new centroids is not improved substantially, the centroids are split again. This continues until either the required number of clusters is reached or the improvement remains inadequate.

In our case we would like to get an approximation of $\hat{C}(k)$, $L_{B1}(k)$, $L_{B2}(k)$ and $L_c(k)$ such that the overall performance of the closed loop system remains in predefined limits. Our simulation results show that it is more effective if we apply vector quantization to each column or row of these matrices separately and combine the approximated parameters at the end.

Chapter 5

Results

This section presents numerical results to show the effectiveness of the proposed optimal H_∞ controller for track-following in a hard disk drive with irregular sampling. Our simulation is based on a model for a single stage 3.5-inch hard disk drive.

Plant Model

As mentioned in section 3, the required continuous-time plant model should be discretized by the proposed method in that subsection. To obtain this continuous-time model, we first measured the discrete time frequency response of the voice-coil motor (VCM) in the aforementioned HDD, and then found a continuous-time model such that its discrete time frequency response fits the measured response. The particular setup used in this study has very regular sampling time intervals, which let us discretize the system in the fitting process by using the MATLAB® `c2d` function with the ZOH method. It should be noticed that this discretization is just for the fitting process, and the model for control design or performance analysis should be discretized by the method presented in section II.

The continuous-time model, G_c, found by the frequency response fitting process is in the form of

$$G_c = G_e \left(\frac{1}{s^2} + G_{r1} + \cdots + G_{r6} \right)$$

where G_e models a low pass filter embedded in the electronic circuit of the servo amplifier part. G_{r1} to G_{r6} captures the first six important second-order resonance modes appeared in the frequency response, i.e.

$$G_{ri} = \frac{g_i}{s^2 + 2\zeta\omega s + \omega^2}, \quad i = 1, \cdots, 6.$$

In order to validate the accuracy of the continuous-time plant model used for the chosen HDD, we compared the 3σ value of PES computed based on real PES measured in the HDD with the corresponding value simulated based on our plant model. Since we had access to

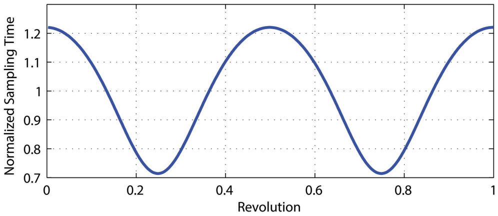

Figure 5.1: Normalized sampling time in one revolution.

the PES during track following in this particular setup, we were able to find the 3σ value of PES in this HDD.

Due to disturbances during servo track writing, the created reference for the track-center deviates from being perfectly circular in shape. These deviations in the track reference constitute the repeatable runout (RRO) for the head positioning servo mechanism. We measured PES at 32000 consecutive sectors during track-following and computed the RRO at each sector by averaging all of the PES values measured at that sector. Once the RRO was computed, we subtracted it from the PES data to form the NRRO, and then found the 3σ value of the NRRO. For this particular setup, this value was 4.95% of the track width. Since we are interested in comparing the actual 3σ value with the simulated one based on our plant model, we used the track-following controller provided by our industry partner to calculate the 3σ value of the closed-loop PES. This value in our model is 8% less than the value calculated based on measurement, which validates the accuracy of our model. We used the ratio of the simulated to measured performance, which is 0.92, to compensate the plant model inaccuracy in the performance analysis. Finally, we discretized G_c by the proposed method in section 3.

Eccentricity and Limit of Performance

As mentioned before, if the center of data tracks, disk and rotation are not exactly coincident, the sampling time over one revolution of the disk will be varying. It is easy to show that when the servo tracks are written radially on the disk and the data tracks are concentric circles having a center coinciding with the rotation center, the time interval between i-th and $(i+1)$-th sampling, denoted as Δt_i, can be calculated by the following two relations:

$$\Delta t_i = \left(\theta_{i+1} - \theta_i\right)/\omega, \quad i \in 0, \cdots, N-1$$

$$\cos\left(\theta_i\right) = \cos\left(\frac{2\pi i}{N}\right)\sqrt{1 - \left(e \sin\left(\frac{2\pi i}{N}\right)\right)^2}$$

where e and ω are respectively the normalized eccentricity between the disk and rotation centers, and the rotation speed. We use $e = 0.7$ to calculate the sampling time interval vector. Although this eccentricity ratio is very large, we use it to show the effectiveness of the proposed methods even though the sampling time intervals vary significantly. For this

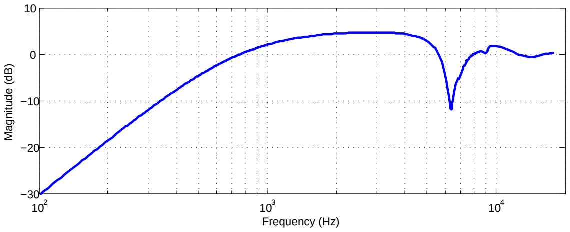

Figure 5.2: Closed loop sensitivity function at the output of the controller.

Table 5.1: Control Signal and PES Performances

Controller Type	$3\sigma^u$	$3\sigma^u_{max}$	$3\sigma^{PES}(nm)$	$3\sigma^{PES}_{max}(nm)$
Limits of Performance	1.00	1.70	2.57	2.85
H_∞ controller	1.30	2.21	2.90	3.04
Quantized H_∞ controller	1.44	2.47	2.93	3.06

particular chosen value, the peak to peak variation of sampling time is equal to 50% of the average value. Fig. 5.1 shows the sampling time vector when it is normalized by the nominal sampling time, which is equal to the average value of the irregular sampling times.

Two of the required parameters to perform our control design are σ^{req} and σ^{req}_{max} in (4.6). To choose reasonable values for them, we first determined the limits of performance for this particular setup as discussed in **shahsavari2013limits** We then chose σ^{req} and σ^{req}_{max} as 15% larger than their corresponding limits.

The limits of performance are listed in TABLE 5.1, where $3\sigma^u$ and $3\sigma^u_{max}$ are respectively the RMS and maximum 3σ values of control signal. These characteristics of the control signal are also metrics for evaluating controller performances.

Since the system is LPTV, frequency response methods do not apply. Despite this, we quantified the performance of the closed-loop system by using what we call an "empirical Bode magnitude plot," which uses a swept-sine approach to find an approximate frequency response magnitude. The assumption here is that inputting a sine wave into the system produces a negligible response at other frequencies. Note that for a LTI system, the both empirical Bode magnitude plot and Bode magnitude plot are equal.

Table 5.2: Number of Clusters for Controller Parameters Quantization

1^{st} row of $\hat{C}$	2^{nd} row of $\hat{C}$	All rows of L_{B1}	All rows of L_{B2}	1^{st} col. of L_C	2^{nd} col. of L_C
16	2	1	1	16	2

Controller Design

We used the architecture shown in Fig. 4.3 to construct the controller. According to this figure, the open loop system consists of the plant and weighting functions. Hence, the number of states needed to define this system in a state space form is equal to the summation of states of these elements. The H_∞ optimal controller designed by the method discussed in section 3 will have the same number of states as the open loop system. So, it is desirable to select weighting functions with small orders. After some iteration, we chose constant values of W_{p1}, W_{p2} and W_u resulted in a controller that achieved good performance in terms of the 3σ values of PES (i.e. it satisfies (4.6)) and adequate robustness in terms of the disk margin. The RMS and maximum 3σ values of the PES attained by this controller are listed in TABLE 5.1. The disk margin of the close loop system is 0.39, which guarantees a gain and phase margin of at least $7.2dB$ and 43° respectively. Fig. 5.2 shows the empirical Bode magnitude plot for the sensitivity function of this system, measured at the output of the controller.

Quantization of Controller Parameters

Since our controller has a large number of time varying matrices (i.e. $\hat{C}(k)$, $L_{B1}(k)$, $L_{B2}(k)$ and $L_c(k)$), we need to use vector quantization to reduce the number of parameters, as discussed in subsection 4. The number of clusters was chosen such that the quantized controller satisfies the memory constraints and still has a reasonable performance. The performance of this controller and number of clusters are listed in TABLE 5.1 and TABLE 5.2 respectively. Using the novel discretization method we presented in section II and the quantization method with the number of clusters listed in TABLE 5.2, we were able to reduce the number of parameters that needed to be stored by a factor of about 181, while the performance is reduced by only 1%.

Chapter 6

Conclusions and future work

In this paper, we considered robust controller design of HDDs with irregular sampling but regular control updates. We modeled this sampling and actuation behavior by applying a novel discretization method to a continuous-time model of the HDD. Once we had an appropriate discrete-time model of the HDD, we used LPTV H_∞ control to design a robust controller. Unlike other papers using H_∞ to design HDD controllers, we did not use a weighted uncertainty model to quantify the robustness of the closed-loop system. Instead, we used the disk margin. The benefit of using this measure of robustness is that it can be used to compute lower bounds on the gain and phase margins, thus putting our robustness margins into a framework more commonly used by the HDD industry.

Once we designed a controller, we evaluated its robustness by checking the disk margin at the control signal. We also checked the performance using a model of the system that included a detailed model of the stochastic disturbances acting on the system. The performance of the controller we designed is only 13% smaller than the limits of performance, while it achieves high level of robustness in term of disk margin.

Since the designed controller had too many parameters to be implementable on an HDD due to memory limitations, we used vector quantization to approximate the parameters of the designed LPTV controller by a smaller set of parameters. The approximate controller needs 181 times less memory and its performance is only 1% lower than the original controller. In the future, we will use the methods of **nie2012optimal** to do loop-shaping designs for this control architecture. This will be done by simply incorporating an appropriate frequency weighting function into our H_∞ design to directly specify an upper bound on the closed-loop sensitivity function.