Design of a probabilistic robust track-following controller for hard disk drive servo systems

Mohammadreza Chamanbaz, Ehsan Keikha, Venkatakrishnan Venkataramanan, Abdullah Al Mamun, Qing-Guo Wang

A*STAR Data Storage Institute, Singapore
A Department of Electrical and Computer Engineering, National University of Singapore, Singapore

Abstract

This paper addresses the design of robust track-following dynamic output feedback controller for hard disk drives (HDDs) in face of parameter uncertainties which can enter into problem description in a possibly non-linear way. The design is performed in a probabilistic framework where the uncertain parameters are treated as random variables and the design specification is met with a given probability level. In particular, a sequential algorithm based on gradient iteration is employed to find a probabilistic robust feasible solution to the formulated problem. The design procedure is computationally tractable and its computational complexity does not depend on the number of uncertain parameters. Our case study allows natural frequency and damping ratio to vary within 8% and 10% from their nominal values for rigid body and all resonance modes. The designed controller achieves robustness in the presence of these uncertainties. Furthermore, the designed controller is implemented in real time on a commercial HDD.

1. Introduction

More than 52% of world wide data storage is dedicated to hard disk drives (HDDs) [11] and with the rapid growth in the amount of generated digital data, the gap between the world wide data and available capacity is increasing dramatically. In order to fill this gap, larger aerial density is necessary. Servo control techniques play a vital role in increasing aerial density by providing positioning algorithms which results in higher track per inch (TPI). On the other hand, since disk drives are produced in batches, one cannot expect exactly the same dynamic for all disk drives in the same batch. The reason lies in some manufacturing tolerances, slightly different material and environmental conditions, etc. On the other hand, since no tuning algorithm of low computational complexity is available currently, it is not feasible to fine tune the servo algorithm in each disk drive after production. Therefore, a single controller should be able to firstly stabilize all plants and secondly, meet the required performance specifications in a robust manner. Consequently, using robust strategy in designing servo controller for the future HDDs seems to be inevitable. Some robust control techniques such as Robust $\mathcal{H}_\infty$, Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ and $\mu$-synthesis have been studied for HDD servo control [17,22]. However, classical robust multi-objective framework suffers from a number of theoretical limitations due to its deterministic nature. These limitations can be generally divided into two main groups, namely computational complexity and the issue of conservatism. Various robust control problems have been proven to belong to the category of “intractable” problems, which is generally denoted as “NP-hard” [18]. In general, the design of robust output-feedback controllers lead to bilinear (or rather bi-affine) matrix inequality terms (BMIs) [28], which are NP-hard in nature [26]. Available algorithms to solve such problems are typically based on the iterative methods which are computationally expensive, specially for higher-order plants. To overcome the conservatism, Conway et al. [9] proposed the approach of utilizing parameter dependant Lyapunov function to design a single robust controller for polytopic parameter uncertain system which led to improvement in the performance of HDD. However, due to the limitations in the existing numerical solvers, this method is just applicable to problems of modest size with few number of uncertainties. In addition to the complexity problem, conservatism is also a challenge for the deterministic robust approach. It is well known that in cases where real parametric
uncertainty enters affinely into plant description, it is possible to compute the robustness margins accurately. However, in real world problems, we usually deal with non-linear parametric uncertainty; for instance, each resonance mode in HDD contains the product term of damping ratio and natural frequency \( \zeta, \omega_n \) as well as natural frequency squared \( \omega_n^2 \) which are non-linear. We highlight that non-linear parametric uncertain system refers to the case where parametric uncertainty enters non-linearly into plant description. Hence, there is a clear distinction between non-linear parametric uncertain system and non-system. In order to design robust controller for non-linear uncertain plant in classical robust paradigm, the non-linear uncertainty will be embedded into affine structure by replacing the original set by a larger one. In other words, multipliers and scaling variables are introduced to relax the problem [3]: which are associated with an evident conservatism.

Hence, a fairly new approach, the so called probabilistic robust control [25], was proposed to overcome above mentioned problems. In the new approach, the robustness is determined in probabilistic sense rather than the classical deterministic one. We accept a small risk that the objective function being violated for a set of uncertainty having small probability measure. Performance and robustness are contrary specifications and the probabilistic approach seems to be a useful way to design the controller in a less conservative manner. It turns out that if we allow a small violation of cost function, the performance of the resulting controller will be improved significantly compared to classical approaches. Early appearance of this approach goes back to 1980s in the field of flight control by Stengel [23]. Some papers have been published in the field of flight control during 1980s and early 1990s which mostly deal with analysis problems based on Monte-Carlo simulation. Later on some results based on explicit sample size bound [24] and statistical learning theory Vidyasagar [30] were presented into literature that deal with both synthesis and analysis problems. Finally, randomized algorithms (RAs) are used in a number of control applications including design of truss structure [6], unmanned aerial vehicles [16] and stability and robustness of high speed communication networks [1]. For a comprehensive study, interested readers are referred to Calafiore and Dabbene [5] and Tempo et al. [25].

The main contribution of this paper, which is an extended version of Chamanbaz et al. [8], is to design a probabilistic robust \( \mathcal{H}_\infty \) controller for a plant with parametric uncertainties. Such a controller can handle parametric uncertainties with significantly less conservatism compared to classical deterministic methods, thanks to randomization. In addition, this method is not computationally complex; while, when it comes to parametric uncertainty, classical deterministic schemes suffer from considerable computational complexity. The reminder of the paper is organized as follows: in Section 2.1, a \( \mathcal{H}_\infty \) classical controller is formulated to be used for designing the probabilistic robust controller. The procedure for designing the probabilistic controller is given in Section 2.2. The experimental set-up is reported in Section 3. Section 4 is dedicated to simulation results. The designed controller is testified through experiment in Section 5. Finally, some concluding remarks are given in Section 6.

1.1. Notation

For a square matrix \( X \in \mathbb{R}^{n \times n} \), \( X \succ 0 \) (\( X \succeq 0 \)) means \( X \) is symmetric and positive definite (semidefinite). Trace of the matrix \( X \) is shown as \( \text{Tr}(X) \). \( I_n \) denotes \( n \times n \) identity matrix. The \( \mathcal{H}_\infty \) norm of a transfer function matrix \( Q(s) \in \mathbb{R}^{p \times q} \) is denoted as \( \|Q\|_\infty \) and \( \|X\|_F \) denotes Frobenius norm of \( X \). The symbol \( \lfloor x \rfloor \) is the minimum integer greater or equal to \( x \in \mathbb{R} \). Also, \( i \) denotes the \( i \)th element of vector \( x \). The projection of a real matrix \( A \) onto the closed set \( S \) is defined as

\[
[A]_S = \arg \min_{S \succeq 0} \| A - S \|_F ,
\]

(1)

In the special case where \( S \) is the cone of symmetric positive semi-definite matrices, we use

\[
[A]_S = \arg \min_{X \succeq 0} \| A - X \|_F .
\]

(2)

To indicate that the matrices \((A, B, C, D)\) are a state space realization of a transfer function matrix \( Q(s) \), we use the notation

\[
Q(s) = \begin{bmatrix} A & B \\ C & D \end{bmatrix} = C(sI - A)^{-1}B + D.
\]

2. Controller design

In this section, we discuss the procedure for designing probabilistic controller. In order to do so, first we need to formulate the problem in the classical \( \mathcal{H}_\infty \) framework; next, \( \mathcal{H}_\infty \) design is represented as linear matrix inequality (LMI) optimization and finally, the algorithm to solve uncertain LMI based on gradient iteration is presented.

2.1. Nominal controller formulation

\( \mathcal{H}_\infty \) control is a well known control design methodology for synthesizing control systems that achieve robust performance or stability. Presented materials of this subsection are classical but a summery is instrumental to our further developments. One of the most popular methods for designing \( \mathcal{H}_\infty \) controller is based on LMI [21]. The reason behind this popularity is that solving LMI is a convex optimization problem for which there are several numerically efficient algorithms and softwares. Furthermore, most performance specifications (time and frequency domains) can easily be represented in LMI format.

LTI model of the plant can be described in state space form as

\[
P : \begin{bmatrix} z \\ y \end{bmatrix} = \begin{bmatrix} A & B_w & B_e \\ C_w & D_w & D_w \\ C_y & D_y & 0 \end{bmatrix} \begin{bmatrix} w \\ u \end{bmatrix} ,
\]

(3)

where \( u \in \mathbb{R}^p \) is the control input vector, \( w \) is a vector of exogenous signals such as disturbance, reference signal and sensor noise, \( y \in \mathbb{R}^q \) and \( z \) are measurement and output vectors respectively. The goal is to design a dynamic output feedback controller of the form

\[
u = K(s)y = \begin{bmatrix} A_k & B_k \\ C_k & D_k \end{bmatrix} y ,
\]

(4)

that stabilizes the plant and minimizes the infinity norm of closed-loop transfer function from exogenous signal \( w \) to the output \( z \), assuming \((A, B_k)\) being stabilizable and \((C_k, A)\) detectable. \( \mathcal{H}_\infty \) optimal control design objective is to minimize \( \| F_r(P(s), K(s)) \|_\infty \),

\[
\min_{K \in \mathcal{K}} \| F_r(P(s), K(s)) \|_\infty ,
\]

(5)

where \( \mathcal{K} \) is the set of all stabilizing controllers and \( F_r \) denotes lower fractional transformation. This problem can be reformulated in LMI form [21]. Hence, alternatively instead of solving (5), we can solve the following optimization problem

\[
\min_{\gamma} \gamma \text{ subject to } \begin{bmatrix} A^TP + PA & PB \\ B^TP & -\gamma I \end{bmatrix} < 0 ,
\]

(6)

where \( A, B, C \) and \( D \) are closed loop matrices. Substituting closed loop matrices in (6) results in a non-linear optimization problem.
However, using change of variables and non-linear transformations we can settle the nonlinearity. In the next subsection, we will design the probabilistic controller which can handle parametric uncertainty in an efficient way.

### 2.2. Probabilistic controller design

The main idea behind the probabilistic framework is to use randomization to handle uncertainty and convex optimization to compute the design parameters. In this approach, we assume that uncertainty is a random variable $\theta$ with probability density function (pdf) $f_\theta(\theta)$ and support $B_\theta(\rho)$ where $B_\theta(\rho)$ is uncertainty ball of radius $\rho$. For simplicity of notation denote all design parameters in (6) by $\theta \in \Theta \subseteq \mathbb{R}^{n_\theta}$; hence, the set of LMI s obtained from (6) after non-linear transformation and change of variables is in the form

$$ F(\theta, \theta^\prime) = F_\theta(\theta) + \sum_{i=0}^{n_\theta} \theta_i F_i(\theta^\prime) < 0. \quad (7) $$

The aim is to find design parameter $\theta^\prime$ which satisfies the LMI constraint (7) in a probabilistic fashion. In other words, by introducing a (small) risk term $\varepsilon$ we want to find $\theta^\prime$ such that

$$ \Pr[F(\theta, \theta^\prime) < 0] \geq 1 - \varepsilon. $$

It turns out that this problem is extremely difficult to solve exactly; since it requires the computation of multi-dimensional integrals associated with the probability. However, we can estimate this probability using randomization. Because of introducing randomization, there will be another probabilistic confidence parameter $\delta$ which indicates the probability by which the randomized algorithm may come up with an erroneous result. Therefore, the full objective is to find $\theta^\prime$ such that

$$ \Pr_{\theta \sim \Theta \; \forall \; \theta^\prime \sim \Theta}[ \Pr[F(\theta, \theta^\prime) < 0] \geq 1 - \varepsilon ] \geq 1 - \delta. \quad (8) $$

where $\Pr_{\theta \sim \Theta \; \forall \; \theta^\prime \sim \Theta}$ is the product probability measure $\Pr \times \Pr \times \cdots \times \Pr$ (N times). Mainly, there are three different methodologies to handle this problem. The first approach is based on Vapnik–Chervonenkis theory of learning [29]. This approach requires computation of a combinatorial parameter called Vapnik–Chervonenkis dimension (VC-dimension) which is not trivial in practice. The second paradigm is based on scenario approach [4]. In this approach, a prespecified number of randomly selected independent scenarios are chosen and optimization is solved subject to finitely many constraints. However, for even moderate sized problems, the resulting optimization problem is very computationally complex. Finally, the third method is sequential approximation method based on gradient, ellipsoid and cutting plane iterations [25]. In this paper, we use the sequential approximation method based on gradient iteration. In order to solve the problem (8), a scalar function $\tau(\theta, \theta^\prime)$, related to the constraint (7), is introduced. The function $\tau(\theta, \theta^\prime)$ measures the level of violation of performance function and is called performance violation function. The following properties hold for performance violation function:

1. $\tau(\theta, \theta^\prime) \geq 0$ for all $\theta \in B_\theta(\rho)$ and $\theta^\prime \in \Theta$ (The performance violation function is a non-negative function).
2. $\tau(\theta, \theta^\prime) = 0$ if and only if $F(\theta, \theta^\prime) = 0$ (The performance violation function is zero if and only if the LMI (7) holds true; otherwise, it is positive).

There are two performance violation functions introduced in the literature: (i) largest eigenvalue of the LMI and (ii) norm of projection of the LMI on the cone of symmetric positive semi-definite matrices. In this paper we use the latter due to its better numerical performance compared to largest eigenvalue. Hence, the performance violation function can be expressed as

$$ \tau(\theta, \theta^\prime) = \|[F(\theta, \theta^\prime)]_*\|_F, \quad (9) $$

where $[.]_*$ is the projection into the cone of symmetric positive semi-definite matrices. The violation function (9) is convex in $\theta$ for any fixed value of $\theta^\prime$ and its sub-gradient satisfies [20]

$$ \partial_\theta \tau(\theta, \theta^\prime) = \begin{cases} \frac{1}{\ln |\theta|} \left[ \begin{array}{c} \mathrm{Tr}(F_1(\theta)[F(\theta, \theta^\prime)]_*) \\ \vdots \\ \mathrm{Tr}(F_{n_\tau}(\theta)[F(\theta, \theta^\prime)]_*) \end{array} \right]_+, & \text{if } \tau(\theta, \theta^\prime) \neq 0, \\ 0, & \text{otherwise} \end{cases} \quad (10) $$

where $F_i(\theta), i = 1, \ldots, n_\tau$ are defined in (7). We remark that, the computation of the sub-gradient $\partial_\theta \tau(\theta, \theta^\prime)$ amounts to solving a symmetric eigenvalue problem associated with the projection [10] which can be performed efficiently. The new objective is to minimize the performance violation function (9). Algorithm 1 presents a randomized strategy to find a probabilistic robust feasible solution to (7).

**Algorithm 1. A randomized strategy for solving uncertain LMI (7)**

- **INITIALIZATION**
  - Set the iteration counter to zero ($k = 0$). Choose the desired accuracy $\varepsilon \in (0, 1)$ and confidence $\delta \in (0, 1)$. Select an initial condition $\theta_0$.
  - **ORACLE**
    - Draw
      $$ N(k) = \left( \frac{\ln \frac{1}{\varepsilon} + 1.11 \ln k + 2.27}{\ln \frac{1}{\delta}} \right) $$
      (11)
      independent and identically distributed (iid) samples $\{\theta_1, \ldots, \theta_{N(k)}\}$ from the uncertainty set based on $f_\theta(\theta)$.
      - For $i = 1 : N(k)$
        - If $\tau(\theta_i, \theta_0) > 0$, then, goto Update step.
        - **RETURN** $\theta_i$ as the probabilistic solution to (7) and exit.
  - **UPDATE**
    - Having “violation certificate” $A_i$ for which $\tau(\theta_i, \theta_0) > 0$, update $\theta_0$ by
      $$ \theta_{0+} = \theta_0 - \delta \delta_0 \left[ \tau(\theta_i, \theta_0) \right]_+ $$
      (12)
    - $\delta_0$ is the subgradient with respect to $\theta_0$ and $\left[ \tau(\theta_i, \theta_0) \right]_+$ is the projection defined in (1) and
      $$ \delta_0 = \frac{\tau(\theta_i, \theta_0) + \|[\tau(\theta_i, \theta_0)]_+\|_F}{\|\tau(\theta_i, \theta_0)\|_F} $$
      (13)
  - **OUTER ITERATION**
    - Increase the iteration counter by one ($k = k + 1$) and goto Oracle step.

Here we describe different parts of the randomized Algorithm 1. First, we initialize the algorithm by choosing the desired probabilistic levels $\varepsilon$ and $\delta$ and any initial condition $\theta_0$. We note that the choice of initial condition directly affects the number of iterations required for convergence, please see Section 4. In “Oracle” step, we extract $N(k)$ iid samples from the uncertainty set and check if the performance violation function is zero for all of them. As soon as we encounter a sample $\theta_i$ for which $\tau(\theta_i, \theta_0) > 0$, we go to the “Update” step otherwise, $\theta_0$ is the probabilistic robust solution and we exit the Algorithm 1. In the “Update” step, the current candidate solution $\theta_0$ is updated with the new one $\theta_{0+}$ based on (12). The convergence of Algorithm 1 is guaranteed under two assumptions.
Assumption 1. A ball with radius \( r \) is contained in the solution set.

Assumption 2. For any design parameter \( \theta \) which is not in the solution set, there is a nonzero probability to generate a sample \( A_i \) for which \( \tau(A_i, \theta) > 0 \).

The first assumption is called "strict feasibility" and the second one is "nonzero probability of detecting unfeasibility". The theoretical properties of Algorithm 1 are summarized in the next theorem [7,25].

Theorem 1. Suppose that Assumptions 1 and 2 hold. For any initial condition \( \theta_0 \), Algorithm 1 finds a probabilistic robust solution \( \theta^* \) in finite number of iterations. Furthermore, \( \Pr(\{F(A, \theta^*) < 0\} \geq 1 - \varepsilon) \) with probability at least 1 – \( \delta \).

The interesting point about the design procedure is that the sample size (11) does not depend on the number of uncertain parameters. As a result, the computational complexity of the designed controller does not depend on the dimension of the uncertainty set. This concept is referred to as breaking the curse of dimensionality which results from the Monte-Carlo simulation in the "Oracle" part of Algorithm 1.

3. Experimental set-up and model identification

In order to verify the designed controller, a track following servo control design of HDD is formulated in this framework. Plant model needs to be identified in order to simulate and implement the designed controller. A widely accepted technique for obtaining the model is frequency domain system identification technique in which the voice coil motor (VCM) is excited using a sweep sine signal and the displacement is measured from written in servo pattern or by using laser doppler vibrometer (LDV). The experimental set-up consists of LDV, 3 HP 35670A, Hewlett Packard Company, Washington, 1 Polytec OFV 5000, Polytec, Waldbronn, Germany, 2 DSpace DS1103, product of dSPACE GmbH, Paderborn, Germany, and 3 LDV is used for measuring the displacement. The identified dynamic model consists of summation of a rigid body and three resonance modes: input and outputs. A commercial disk drive is chosen as the platform to examine the designed controller. Head tip should be accessible to the laser beam; to do so, we made a hole on the casing and covered the hole with a transparent glass. Transparent glass keeps the internal environment isolated from dust and other pollutants; also, it helps to maintain the internal airflow the same as the normal operating condition. As shown in Fig. 1, a touchlight is used as illumination module in order to make read/write head and slider visible. The rotational speed is fixed at 7200 rpm using spindle motor driver and all experiments are carried on a vibration free table.

Fig. 2 shows the measured as well as estimated frequency response of the VCM actuator. The identified dynamic model consists of summation of a rigid body and three resonance modes:

\[
P_{\text{VCM}} = \frac{\sum_{i=1}^{4} \frac{A_i}{s^2 + 2\zeta_i\omega_i s + \omega_i^2}}{s^2 + 2\zeta_0\omega_0 s + \omega_0^2}.
\]

For each mode \( \zeta_i \) is damping ratio, \( \omega_i \) is natural frequency and \( A_i \) is modal constant. Table 1 shows the estimated nominal plant parameters. The parameters \( \zeta_i \) and \( \omega_i \) are assumed to be uncertain by 10% and 8% respectively. The parametric uncertainty is expressed in multiplicative form as

\[
\zeta_i = \bar{\zeta}_i(1 + 0.1\eta_1), \quad \omega_i = \bar{\omega}_i(1 + 0.08\eta_2), \quad i = 1, \ldots, 4
\]

where parameters with overbar e.g. \( \bar{\omega}_i \), denote the nominal values and \( \eta_i \)'s are perturbations which are norm bounded. It is worth noting that in order to solve this problem in classical robust framework, one has to change the original uncertain system into a polytopic uncertain system in which transfer function coefficients depend affinely on the uncertain parameters. Hence, \( \bar{\zeta}_i \), \( \bar{\omega}_i \) and \( \omega_i^2 \) should be considered as new uncertain variables; the procedure is associated with an evident conservatism which results in an over-conservative controller. However, in a probabilistic framework, uncertain parameters are treated as they are. This is one of the superiorities of the probabilistic framework over the classical deterministic one. The block diagram of the augmented open loop system is shown in Fig. 3. \( P \) denotes the nominal plant, \( W_u \), and \( W_p \) are the weighting functions and \( K(s) \) is the dynamic output feedback controller to be designed. There is no general rule for choosing weighting functions \( W_p \) and \( W_u \); however, since the closed loop sensitivity transfer function roughly follows the inverse of the performance weighting function \( W_p \), we should design \( W_p \) in a way that it is large for low frequencies and tends to smaller values for higher frequencies. The crossover frequency can roughly be determined by the frequency in which \( W_u \) crosses the 0 dB line. The control weighting \( W_u \) is chosen to be a high-pass filter.

\footnotetext[1]{Polytec OFV 5000, Polytec, Waldbronn, Germany.}
\footnotetext[2]{HP 35670A, Hewlett Packard Company, Washington.}
\footnotetext[3]{DSpace DS1103, product of dSPACE GmbH, Paderborn, Germany.}
\footnotetext[4]{Spin-Box SPB-NL-1205, DSI, Singapore.}
Therefore, high frequency signals are prevented from being applied to the plant. The performance weighting function $W_p$ is chosen to be in the form

$$W_p = \left( \frac{s^3 + N}{s + W_c} \right)^k$$

for which $W_c$ is the desired crossover frequency, $N$ determines the desired bound on the sensitivity peak, $S$ determines the minimum level of the closed loop sensitivity transfer function in low frequencies and finally, $k$ determines the slope of the closed loop sensitivity transfer function. The crossover frequency should be chosen very carefully as it directly affect the value of track misregistration (TMR). In the presented $H_\infty$ methodology, we can easily achieve the crossover frequency of larger than 2 kHz by the appropriate choice of weightings. However, due to limitations imposed by Bode’s sensitivity integral, pushing the crossover frequency to extremes will increase the peak of the closed loop sensitivity transfer function and hence, increases the value of TMR. Some simulations were carried to find the best weighting functions $W_p$ and $W_c$. We designed the controller for a number of weightings achieving different crossover frequencies and measured the value of TMR in simulation. The best weighting functions $W_p$ and $W_c$ for which the value of TMR is minimized are shown in Fig. 4.

4. Simulation

Simulations are performed to prove the effectiveness of the designed controller. The objective is to design a dynamic output feed-forward controller $K(s)$ which minimizes the worst case (over the uncertainty set) $H_\infty$ norm of the transfer function from reference signal $r$ to outputs $e_u$ and $e_v$. In a classical robust approach, in cases where uncertainty is non-parametric (e.g. high frequency unmodeled dynamics), it can be represented in the form of linear fractional transformation (LFT) and thanks to small gain theorem, the controller objective can be reformulated into controller design in the absence of uncertainty. However, for parametric uncertainty following the same approach is over-conservative and when the number of parametric uncertainties increase, the optimization procedure fails to converge. Hence, one of the novelties in probabilistic controller design is that by accepting a very small risk, the $H_\infty$ controller is designed which robustly stabilizes the closed-loop plant with the desired probabilistic levels. In order to design the controller, the generalized plant should be represented in the state space form. To reduce the number of outer iterations in the algorithm, we solved the $H_\infty$ problem for the nominal plant using YALMIP [15] and the results were given to Algorithm 1 as an initial value. After a number of iterations which depends on the probabilistic levels $e$, $d$ and $r$, algorithm comes up with the design parameters that make the closed-loop plant robustly stable (in probabilistic sense). The probabilistic levels $e$ and $d$ are user defined parameters in the algorithm. However, as it is clear from (11), smaller $e$ and $d$ tend to larger sample bound requiring more computational effort in the “Oracle” to validate the candidate solution. Therefore, there is a tradeoff between computational complexity and smaller probabilistic risk and confidence levels. We chose $10^{-5}$ and $10^{-4}$ for $e$ and $d$ respectively and using Algorithm 1 which was implemented in Matlab [27], we solved the control design problem. We remark that, it is very difficult (if not impossible) to determine the probability density function (pdf) of the uncertain parameters in practice. Nevertheless, uniform pdf exhibits a worst case property [2] and is used in cases where the underlying pdf is unknown. Therefore, in the present paper we used uniform pdf while sampling the uncertainty set.

To further validate our design, a posteriori analysis using Monte-Carlo simulation is carried for the designed controller. To do so, 500 random uncertain plants are chosen from the uncertainty set, then we closed the loop for each of them. Fig. 5 shows closed loop sensitivity plots for the designed controller and Fig. 6 demonstrates the same plot for the case where the nominal $H_\infty$ parameters.
controller is designed without considering any uncertainty. The nominal $\mathcal{H}_\infty$ controller tends to instability for some realizations of the uncertainty set. On the other hand, the probabilistic controller degrades much less over the uncertain parameter set. To compare the results obtained using the designed probabilistic controller with the well known classical methods, we designed a PID controller with adaptive notch filters. The PID (or rather PD) controller is designed such that it achieves the same crossover frequency as the nominal $\mathcal{H}_\infty$ design. There are a number of adaptive algorithms in HDD literature such as Kalyanam and Tsao [14,13] Ohno and Hara [19]. The adaptive algorithm in this paper is adopted from Ohno and Hara [19]. Considering three resonance modes in the plant, we need to design three adaptive notch filters. Fig. 7 shows how the estimated resonance frequencies converge to the actual ones. The horizontal axis in Fig. 7 represents the time for which we should wait for the adaptive algorithm to converge. The simulated sensitivity transfer function for 500 random samples extracted from the uncertainty set is shown in Fig. 8. All the sensitivity transfer functions are measured after the adaptive algorithm converges. A modified version of the standard disturbance [12], which includes repeatable as well as non-repeatable runouts (RRO and NRRO), is used in order to further evaluate the track-following performance of the designed controller. The track misregistration (TMR), root mean square (RMS) and peak values of the control input signal are tabulated in Table 2 for nominal as well as worst case scenarios of the 500 randomly selected uncertain plants. The nominal and worst case stability margins (phase and gain margins) are evaluated for the designed controller as well as the PID controller with adaptive notch filter. The results which are shown in Table 3, further demonstrate the robustness of the designed controller. The results obtained using the designed controller is fairly comparable with the obtained results using the PID controller with adaptive notch filter. However, we note that the amount of online computations involved in the adaptive notch filter is significant since it requires frequent computation of big convolutions to estimate the actual resonance frequency which makes it difficult to implement in real time with the available computational power in HDD. On the other hand, the probabilistic robust controller achieves almost the same performance level as the PID controller with adaptive notch filter without any real-time computation. The bode plot of the nominal as well as designed $\mathcal{H}_\infty$ controllers, which are of order 12, along with the PID controller with adaptive notch filter are shown in Fig. 9 for a better comparison.

5. Real time implementation

Simulation results in the previous section confirmed that the designed controller achieves robust stability and performance in the presence of parametric variations in the dynamical system. The validity of the designed controller is testified through experiment. The control algorithm is discretized and implemented in real time using the DSP based system with sampling frequency of 50 kHz. Since the problem is of regulation type, the output sensitivity transfer function, which shows the ability of the system in rejecting different output disturbances, is of vital importance. The transfer function from reference input to the error which represents the output sensitivity transfer function is experimentally measured using DSA for both nominal $\mathcal{H}_\infty$ controller and the designed probabilistic controller. The result of this experiment

![Fig. 6. Closed loop sensitivity plot with nominal $\mathcal{H}_\infty$ controller for 500 random samples from the uncertainty set.](image_url)

![Fig. 7. Error between the actual resonance frequencies and the estimated ones.](image_url)

![Fig. 8. Closed loop sensitivity plot with controller designed using PID and adaptive notch filter for 500 random samples from the uncertainty set.](image_url)

| Design approach            | TMR (nm)  | RMS($u_k$) (mV) | $||u_k||_{L_2}$ (mV) |
|----------------------------|-----------|-----------------|----------------------|
|                            | Nominal   | Worst case      | Nominal   | Worst case     | Nominal | Worst case     |
| Designed probabilistic controller | 9.68      | 9.98            | 8.4       | 8.9            | 39.1    | 39.2            |
| Nominal $\mathcal{H}_\infty$ design | 9.6       | –               | 7.7       | –              | 29.8    | –              |
| PID with adaptive notch filter | 10.39     | 10.84           | 8.1       | 8.6            | 29.1    | 29.3            |

Table 2: Comparison of the nominal and worst case (among 500 scenarios) performance specifications.
The controller and plant are discretized using "tuistin" and "zoh" respectively. The designed controller was also evaluated with a 50 Hz square wave signal. Fig. 12 shows the output displacement for a step responses of 100 nm as well as the corresponding input signal to the VCM driver. Each rise and fall in the reference signal is considered as a step trigger. The fluctuating signals on the step response is mostly due to disk rotation and the air-flow induced vibration.

There are a number of factors which affect the servo performance in our implementation setup and cause the experimental results to deviate slightly from the ones obtained in the simulation. The first, and the most important, factor is the computational delay in the DSP system. Starting from low frequency, the phase of the implemented controller drifts from the actual one; the phase drift increases with the frequency. For instance, the phase drift is 10° at 2 kHz and increases to 30° at 10 kHz. Secondly, in order to perform the experiment some modifications needs to be done to HDD e.g. the hole on the casing, applying transparent glass, etc. which affects the original optimal structural dynamic of the VCM.

### Table 3
Comparison of the nominal and worst case (among 500 scenarios) stability margins.

<table>
<thead>
<tr>
<th>Design approach</th>
<th>Gain margin (dB)</th>
<th>Phase margin (°)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nominal $H_\infty$ design</td>
<td>6.88</td>
<td>37.85</td>
</tr>
<tr>
<td>Designed probabilistic controller</td>
<td>5.44</td>
<td>39.37</td>
</tr>
<tr>
<td>PID with adaptive notch filter</td>
<td>4.14</td>
<td>36.4</td>
</tr>
</tbody>
</table>

for a step responses of 100 nm as well as the corresponding input signal to the VCM driver. Each rise and fall in the reference signal is considered as a step trigger. The fluctuating signals on the step response is mostly due to disk rotation and the air-flow induced vibration.

There are a number of factors which affect the servo performance in our implementation setup and cause the experimental results to deviate slightly from the ones obtained in the simulation. The first, and the most important, factor is the computational delay in the DSP system. Starting from low frequency, the phase of the implemented controller drifts from the actual one; the phase drift increases with the frequency. For instance, the phase drift is 10° at 2 kHz and increases to 30° at 10 kHz. Secondly, in order to perform the experiment some modifications needs to be done to HDD e.g. the hole on the casing, applying transparent glass, etc. which affects the original optimal structural dynamic of the VCM.

### 6. Conclusion

Uncertainty in the plant’s dynamic is inevitable for HDDs; on the other hand, higher performance has always been in demand due to the rapid growth in world wide generated digital information. In the classical robust controller design, performance is sacrificed considerably in order to achieve robustness specially in cases where parametric uncertainty enters non-linearly into plant description. However, in the studied approach which benefits from probabilistic concepts and randomization, no conservatism has been introduced. Furthermore, classical robust design suffers from considerable computational complexity while handling parametric uncertainty. The approach based on probabilistic robust design and randomized algorithms has broken the curse of dimensionality thanks to randomization; as a result, Algorithm 1 runs in polynomial time and its computational complexity does not depend on the dimension of the uncertainty set (the number of uncertain parameters). In our case study, we designed a $H_\infty$ dynamic output feedback controller which achieves robust stability and performance in the presence of various non-linear parametric
uncertainties. Uncertainty was considered as random variable with uniform probability density. The choice of uniform probability density is chosen due to its worst case nature. An iterative algorithm based on gradient iteration was employed in order to find a probabilistic robust feasible solution to the formulated $H_\infty$ problem. Simulation results show the effectiveness of the designed controller. The designed dynamic output feedback controller was practically implemented on a commercial disk drive.

Acknowledgement

We would like to express our gratitude to Prof. Roberto Tempo and Dr. Fabrizio Dabbene for their constructive comments which led to improvement of the paper.

References