

A QUASI-NEWTON ALGORITHM FOR LQG CONTROLLER DESIGN WITH VARIANCE CONSTRAINTS

Richard Conway

Computer Mechanics Laboratory
Department of Mechanical Engineering
University of California at Berkeley
Berkeley, California 94720-1740
Email: rconway345@berkeley.edu

Roberto Horowitz

Professor of Mechanical Engineering
University of California at Berkeley
Berkeley, California 94720-1740
Email: horowitz@me.berkeley.edu

ABSTRACT

In this paper, we present a new algorithm for solving the LQG control problem with variance constraints which utilizes derivative information about the relevant \mathcal{H}_2 costs to achieve quasi-Newton convergence. Using a lifting procedure, this algorithm is then generalized to work with linear periodically time-varying systems. This algorithm is then applied to the design of controllers for hard disk drives in order to assess the limits of performance of a particular setup. It is demonstrated that just by utilizing multirate sampling and actuation characteristics (i.e. without changing the hardware), the performance of this particular setup can be improved by more than 39%.

1 INTRODUCTION

In the field of optimal control, control design problems are typically formulated as the minimization of a scalar cost function involving the closed loop system. However, in real-world applications, it is often difficult to capture the tradeoffs inherent in controller design with a scalar cost function; the controller design process is inherently multiobjective in nature. One example of such a control design problem is the minimization of the variance of one closed loop signal subject to variance constraints on several other closed loop signals. This problem is known as the LQG control problem with variance constraints. In addition to being a useful tool for practical controller design, it is also useful as a mechatronic design tool to determine the limitations of performance of a given system. This can be used, for example, to compare several different hardware setups by determining the best possible closed loop performance that can be achieved for each one [1].

The current approaches to solving this problem fall into two categories: ones which invoke the Lyapunov shaping paradigm to transform the problem into an linear matrix inequality (LMI) optimization problem [2] and ones which use Lagrange multipliers to solve the optimality conditions [3–5]. In the first approach, it is well-known that the time required to find the optimal controller scales very poorly with increasing plant dimension. This makes the latter approach more attractive from the standpoint of computational efficiency. However, the approaches using the latter method only use heuristics to update the Lagrange multipliers. In this paper, we will present a new methodology for solving this control problem which uses derivative information to update the Lagrange multipliers and achieve quasi-Newton convergence.

To demonstrate the effectiveness of this methodology, we then consider track-following control of hard disk drives (HDDs). For several decades, the areal storage density of hard disk drives (HDDs) has been doubling roughly every 18 months, as predicted by Kryder's law. As the storage density is pushed higher, the concentric tracks on the disk which contain data must be pushed closer together, which necessitates more accurate control of the read/write head.

Currently, it is a goal of the magnetic recording industry to achieve an areal storage density of 1 terabit/in², which is expected to require the 3σ value of the closed loop position error signal (PES) to be less than 4.6nm. To help achieve this goal, the use of a secondary actuator has been proposed to give increased precision in read/write head positioning. In this paper we use a microactuator (MA) which directly actuates the head/slider assembly with respect to the suspension tip and generates MA displacement (RPES) measurements [6]. With this in mind, we

would like to find controllers which minimize the PES variance subject to keeping the variance of each control input smaller than prescribed bounds. In this paper, we use the proposed control design methodology to evaluate the limits of performance of a particular hard drive under various sampling and actuation conditions. We then compare the results to ones achieved by solving the equivalent LMI optimization problem and provide evidence that the proposed approach is superior in a number of ways, including numerical accuracy and computational efficiency.

2 CONTROL DESIGN

2.1 Preliminaries

To begin, we denote the linear time-invariant (LTI) system we want to control as G . For a given controller, K , we will denote the closed-loop system as $G_{cl}(K)$. With this in mind, the control problem we will be considering is one of the form

$$\begin{aligned} & \min_K \|L_1 G_{cl}(K)\|^2 \\ \text{subject to: } & \|L_i G_{cl}(K)\|^2 \leq \gamma_i, \quad i = 2, \dots, n \end{aligned} \quad (1)$$

where the γ_i 's and L_i 's are respectively scalars and matrices chosen by the control designer and the norm being considered is the \mathcal{H}_2 norm. Without loss of generality, we assume that the nullity of $[L_1^T \dots L_n^T]^T$ is zero. An equivalent (unconstrained) optimization problem, called the primal optimization problem, is

$$\begin{aligned} & \min_K \max_{\lambda \geq 0} \mathcal{L}(K, \lambda) \\ \mathcal{L}(K, \lambda) := & J_1 + \sum_{i=2}^n \lambda_i (J_i - \gamma_i), \quad J_i(K) := \|L_i G_{cl}(K)\|^2 \end{aligned}$$

where λ is the vector containing the λ_i 's and inequalities are element-wise. By definition, $\mathcal{L}(K, \lambda)$ is the Lagrangian of Eq. (1). Any controller which satisfies the constraints in Eq. (1) will be called primal feasible. Note that for any primal feasible controller, $\mathcal{L}(K, 0)$ is an upper bound on the optimal cost in Eq. (1).

A related optimization problem, called the dual optimization problem, is

$$\begin{aligned} & \max_{\lambda \geq 0} g(\lambda) \\ g(\lambda) := & \min_K \mathcal{L}(K, \lambda). \end{aligned}$$

By the properties of the dual optimization problem, $g(\lambda)$ is a lower bound for the optimal cost in Eq. (1). A controller, K , will be called dual feasible if $\exists \lambda$ such that $g(\lambda) = \mathcal{L}(K, \lambda)$. We will call (K, λ) which satisfy this relationship a dual feasible pair.

A controller will be called primal-dual feasible if it is both primal and dual feasible. We will call (K, λ) a primal-dual feasible pair if it is a dual feasible pair and K is primal feasible. We

now define for any dual feasible pair

$$v(K, \lambda) := \mathcal{L}(K, 0) - \mathcal{L}(K, \lambda).$$

In the case that (K, λ) is a primal-dual feasible pair, v is the duality gap; i.e. it represents the difference between the upper and lower bounds for the optimal cost in Eq. (1). The important property here is that $\forall \epsilon > 0$, there exists a primal-dual feasible pair such that the duality gap is less than ϵ . This fact can be seen, for instance, by examining the central path of the optimization problem in Eq. (1) and the associated dual variables [7]. Thus, we can justify reducing the set of controllers over which we are optimizing to ones which are primal-dual feasible.

2.2 Dual Feasible Controllers

In this section, we show how to construct a dual feasible pair given a positive value of λ . To begin, we define

$$W := [L_1^T \sqrt{\lambda_2} L_2^T \dots \sqrt{\lambda_n} L_n^T]^T$$

and use the properties of \mathcal{H}_2 norms to express

$$\begin{aligned} g(\lambda) &= \min_K \left\{ \|L_1 G_{cl}(K)\|^2 + \sum_{i=2}^n \lambda_i \left(\|L_i G_{cl}(K)\|^2 - \gamma_i \right) \right\} \\ &= - \sum_{i=2}^n \lambda_i \gamma_i + \min_K \|W G_{cl}(K)\|^2 \end{aligned}$$

Thus, we can see that finding $g(\lambda)$ and the associated minimizing controller is equivalent to solving an \mathcal{H}_2 optimal control problem.

Now we let G have the realization

$$\begin{bmatrix} x_{k+1} \\ z_k \\ y_k \end{bmatrix} = \begin{bmatrix} A & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & 0 \end{bmatrix} \begin{bmatrix} x_k \\ w_k \\ u_k \end{bmatrix} \quad (2)$$

where z , y , u , are w are respectively the vectors of performance outputs, measurements, control inputs, and disturbances. The subscript k refers to the time index. The following assumptions are made about this model:

- $D_{12}^T D_{12}$ and $D_{21} D_{21}^T$ are nonsingular
- $[A, B_1]$ and $[A, B_2]$ are stabilizable
- $[A, C_1]$ and $[A, C_2]$ are detectable

When $\lambda > 0$, there exists a unique optimal \mathcal{H}_2 controller for this system, which we will denote as $K_*(\lambda)$. This controller can be decomposed into a Kalman filter and an optimal full information (FI) controller. Defining

$$Q_{KF} := B_1 B_1^T, \quad R_{KF} := D_{21} D_{21}^T, \quad S_{KF} := B_1 D_{21}^T$$

the Kalman filter can be written

$$\begin{bmatrix} \hat{x}_{k+1}^o \\ \hat{x}_k \\ \hat{w}_k \end{bmatrix} = \begin{bmatrix} A+LC_2 & -L & B_2 \\ I+F_x C_2 & -F_x & 0 \\ F_w C_2 & -F_w & 0 \end{bmatrix} \begin{bmatrix} \hat{x}_k^o \\ y_k \\ u_k \end{bmatrix}$$

where

$$M = AMA^T + Q_{KF} - (AMC_2^T + S_{KF}) \times (C_2MC_2^T + R_{KF})^{-1} (C_2MA^T + S_{KF}^T) \quad (3a)$$

$$L = -(AMC_2^T + S_{KF}) (C_2MC_2^T + R_{KF})^{-1} \quad (3b)$$

$$F_x = -MC_2^T (C_2MC_2^T + R_{KF})^{-1} \quad (3c)$$

$$F_w = -D_{21}^T (C_2MC_2^T + R_{KF})^{-1}. \quad (3d)$$

Here, \hat{x}_k^o and \hat{x}_k respectively represent the a priori and a posteriori state estimates and \hat{w}_k represents the a posteriori disturbance estimate. We now turn our attention to the FI controller. Defining

$$\begin{aligned} Q_{FI} &:= C_1^T W^T W C_1, & R_{FI} &:= D_{12}^T W^T W D_{12} \\ S_{FI} &:= C_1^T W^T W D_{12}, & S_{Kw} &:= D_{11}^T W^T W D_{12} \end{aligned}$$

the FI controller can be written

$$u_k = [K_x \ K_w] \begin{bmatrix} \hat{x}_k \\ \hat{w}_k \end{bmatrix}$$

where

$$P = A^T P A + Q_{FI} - (A^T P B_2 + S_{FI}) \times (B_2^T P B_2 + R_{FI})^{-1} (B_2^T P A + S_{FI}^T) \quad (4a)$$

$$K_x = -(B_2^T P B_2 + R_{FI})^{-1} (B_2^T P A + S_{FI}^T) \quad (4b)$$

$$K_w = -(B_2^T P B_2 + R_{FI})^{-1} (B_2^T P B_1 + S_{Kw}^T). \quad (4c)$$

Note that the assumption on the L_i 's and the nonsingularity of $D_{12}^T D_{12}$ guarantees that $W^T W$ is nonsingular, which in turn guarantees the existence of the FI controller whenever $\lambda > 0$. Unlike the Kalman filter, the FI controller is a function of W . This fact will become important in the next subsection. Combining these two results, a realization of $K_*(\lambda)$ is given by

$$\begin{bmatrix} x_{k+1}^o \\ u_k \end{bmatrix} = \begin{bmatrix} A+LC_2+B_2K_x+B_2HC_2 & -L-B_2H \\ K_x+HC_2 & -H \end{bmatrix} \begin{bmatrix} x_k^o \\ y_k \end{bmatrix} \quad (5)$$

$$H := K_x F_x + K_w F_w. \quad (6)$$

Now that we have an expression for $K_*(\lambda)$, we need to determine a convenient way of finding the costs involved in Eq. (1).

To begin, we define $\tilde{x}_k := x_k - \hat{x}_k$ and express the closed loop system as

$$\begin{bmatrix} \tilde{x}_{k+1}^o \\ \tilde{x}_{k+1} \\ z_k \end{bmatrix} = \begin{bmatrix} \bar{A}_{11} & 0 \\ \bar{A}_{21} & \bar{A}_{22} \\ \bar{C}_1 & \bar{C}_2 \end{bmatrix} \begin{bmatrix} B_1+LD_{21} \\ -LD_{21}-B_2HD_{21} \\ \bar{D} \end{bmatrix} \begin{bmatrix} \tilde{x}_k^o \\ \tilde{x}_k \\ w_k \end{bmatrix} \quad (7)$$

$$\begin{aligned} \bar{A}_{11} &:= A+LC_2, & \bar{C}_1 &:= C_1-D_{12}HC_2, \\ \bar{A}_{21} &:= -LC_2-B_2HC_2, & \bar{C}_2 &:= C_1+D_{12}K_x, \\ \bar{A}_{22} &:= A+B_2K_x, & \bar{D} &:= D_{11}-D_{12}HD_{21}. \end{aligned}$$

It is straightforward to verify that the controllability gramian for this system, X_c , is given by

$$X_c = \begin{bmatrix} M & 0 \\ 0 & Y \end{bmatrix} \quad (8a)$$

$$Y = \bar{A}_{22} Y \bar{A}_{22}^T + (L+B_2H) (C_2MC_2^T + R_{KF}) (L+B_2H)^T. \quad (8b)$$

Thus, for this controller,

$$J_i(K_*(\lambda)) = \text{tr} \left\{ L_i \left[\bar{C}_1 M \bar{C}_1^T + \bar{C}_2 Y \bar{C}_2^T + \bar{D} \bar{D}^T \right] L_i^T \right\}. \quad (9)$$

2.3 Cost and Controller Derivatives

In this section, we form a first-order model for the costs in Eq. (9) as λ varies. Since the Kalman filter does not change as λ (i.e. W) varies, we just have to worry about how the FI controller changes as we change λ . First, we look at the derivatives of Eq. (4a). For convenience, we now define

$$\begin{aligned} \Pi_i &:= L_i^T L_i \\ \Phi &:= B_2^T P B_2 + R_{FI}. \end{aligned} \quad (10)$$

Since the stabilizing solution of an algebraic Riccati equation is analytic [8], we can implicitly differentiate both sides of Eq. (4a) with respect to λ_i to get, after some simplification,

$$P^{[i]} = \bar{A}_{22}^T P^{[i]} \bar{A}_{22} + \bar{C}_2^T \Pi_i \bar{C}_2 \quad (11)$$

where the superscript $[i]$ denotes the partial derivative with respect to λ_i . Since the closed loop system in Eq. (7) is stable, \bar{A}_{22} is Schur, which in turn implies that this Lyapunov equation always has a unique solution for $P^{[i]}$. Once we have this, we can express the derivatives of Eqs. (4b), (4c), and (6) respectively as

$$\begin{aligned} K_x^{[i]} &= -\Phi^{-1} \left(B_2^T P^{[i]} \bar{A}_{22} + D_{12}^T \Pi_i \bar{C}_2 \right) \\ K_w^{[i]} &= -\Phi^{-1} \left(B_2^T P^{[i]} (B_1 + B_2 K_w) + D_{12}^T \Pi_i (D_{11} + D_{12} K_w) \right) \\ H^{[i]} &= K_x^{[i]} F_x + K_w^{[i]} F_w. \end{aligned} \quad (12)$$

With this in place, we can now express the derivatives of the matrices in Eq. (7). In particular

$$\begin{aligned}\bar{A}_{22}^{[i]} &= B_2 K_x^{[i]}, & \bar{C}_1^{[i]} &= -D_{12} H^{[i]} C_2 \\ \bar{D}^{[i]} &= -D_{12} H^{[i]} D_{21}, & \bar{C}_2^{[i]} &= D_{12} K_x^{[i]}. \end{aligned} \quad (13)$$

Using these expressions, the derivative of Eq. (8b) is given by

$$\begin{aligned}Y^{[i]} &= \bar{A}_{22} Y^{[i]} \bar{A}_{22}^T + \left(\bar{A}_{22}^{[i]} Y \bar{A}_{22}^T + \bar{A}_{22} Y \left(\bar{A}_{22}^{[i]} \right)^T \right. \\ &\quad + B_2 H^{[i]} (C_2 M C_2 + R_{KF}) (L + B_2 H)^T \\ &\quad \left. + (L + B_2 H) (C_2 M C_2 + R_{KF}) \left(B^2 H^{[i]} \right)^T \right). \end{aligned} \quad (14)$$

Again, because \bar{A}_{22} is Schur, this Lyapunov equation for $Y^{[i]}$ is always guaranteed to have a unique solution. Finally, the derivative of Eq. (9) is given by

$$\begin{aligned}\frac{\partial J_j}{\partial \lambda_i} &= \text{tr} \left\{ L_j \left(\bar{C}_1^{[i]} M \bar{C}_1^T + \bar{C}_1 M \left(\bar{C}_1^{[i]} \right)^T + \bar{C}_2^{[i]} Y \bar{C}_2^T \right. \right. \\ &\quad \left. \left. + \bar{C}_2 Y^{[i]} \bar{C}_2^T + \bar{C}_2 Y \left(\bar{C}_2^{[i]} \right)^T + \bar{D}^{[i]} \bar{D}^T + \bar{D} \left(\bar{D}^{[i]} \right)^T \right) L_j^T \right\} \end{aligned} \quad (15)$$

Thus, assuming that we have already solved Eqs. (4a)-(4c), we can find the derivative of J_1, \dots, J_n with respect to λ_i using the following methodology:

1. Compute $P^{[i]}$ by solving the Lyapunov equation, Eq. (11).
2. Compute $K_x^{[i]}$, $K_w^{[i]}$, and $H^{[i]}$ using Eq. (12).
3. Compute $\bar{A}_{22}^{[i]}$, $\bar{C}_1^{[i]}$, $\bar{C}_2^{[i]}$ and $\bar{D}^{[i]}$ using Eq. (13).
4. Compute $Y^{[i]}$ by solving the Lyapunov equation, Eq. (14).
5. Compute $(\partial J_j)/(\partial \lambda_i)$ using Eq. (15) for $j = 1, \dots, n$.

The computation time in this algorithm will be dominated by computing the two Lyapunov equations solutions. Thus, we can see that computing the Jacobian of $[J_1 \dots J_n]^T$ will be dominated by computing the solution of $2n - 2$ Lyapunov equations.

2.4 Quasi-Newton Control Design

With the results of the previous section, we can formulate a quasi-Newton algorithm for performing LQG control design with variance constraints. First, we restrict the set of controllers to ones which are primal-dual feasible and recognize that since a primal-dual feasible controller is uniquely determined by λ , we can treat the controller as a function of λ , which leaves the vector λ as the only optimization parameter. With this in mind,

an equivalent formulation of Eq. (1) is

$$\begin{aligned} &\inf_{\lambda} v^2(K_*(\lambda), \lambda) \\ \text{subject to: } &\begin{cases} -\lambda_i < 0 \\ J_i(K_*(\lambda), \lambda) - \gamma_i \leq 0 \end{cases}, \quad i = 2, \dots, n. \end{aligned} \quad (16)$$

Here, we have used the fact that v can be made arbitrarily small via choice of positive λ . Thus, the optimal cost of this optimization problem is always zero. Note that since we have restricted λ_i to be strictly positive, the optimal cost might only be achieved in the limit.

Given positive values of λ_i , we can evaluate the relevant expressions in Eq. 16 using the results of Section 2.2. Also, given the results of Section 2.3 and noting that

$$\frac{\partial v^2(K_*(\lambda), \lambda)}{\partial \lambda_i} = 2v \left(\gamma_i - J_i - \sum_{k=2}^n \lambda_k \frac{\partial J_k}{\partial \lambda_i} \right)$$

we can evaluate the derivatives of all of the relevant expressions in Eq. (16) with respect to λ_i . Thus, using these results, we can use the `fmincon` command in the Optimization Toolbox for MATLAB to solve this problem. This approach gives quasi-Newton convergence without requiring computation of the Hessian of any expressions with respect to λ .

Although the optimization problem in Eq. (16) is a non-linear, nonconvex optimization problem, this approach tends to work well because it balances decreasing the cost in the primal optimization problem and increasing the lower bound on the primal optimization problem. In addition, we get a ‘‘free’’ certificate of optimality when we solve the problem this way, i.e. if ϵ is small, we know that we are close to the optimal solution.

3 GENERALIZATION TO PERIODIC SYSTEMS

In this section, we will consider the control of linear periodically time-varying (LPTV) systems, i.e. ones of which have a periodic realization given by

$$\begin{bmatrix} x_{k+1} \\ z_k \\ y_k \end{bmatrix} = \begin{bmatrix} \mathcal{A}(k) & \mathcal{B}_1(k) & \mathcal{B}_2(k) \\ \mathcal{C}_1(k) & \mathcal{D}_{11}(k) & \mathcal{D}_{12}(k) \\ \mathcal{C}_2(k) & \mathcal{D}_{21}(k) & 0 \end{bmatrix} \begin{bmatrix} x_k \\ w_k \\ u_k \end{bmatrix} \quad (17)$$

where the state space entries are periodic with period N . Since we are no longer dealing with LTI systems, we need to use an appropriate generalization of the \mathcal{H}_2 norm. In the time domain, the \mathcal{H}_2 norm of a system can be interpreted as $\sqrt{\text{tr} \Lambda}$ where Λ is the covariance of the system when driven by zero-mean white Gaussian noise with unit covariance. However, since the second-order statistics of the system output vary with time for a LPTV

system, we are interested in the root-mean-square (RMS) value of $\sqrt{\text{tr}\Lambda}$, which corresponds to the ℓ_2 semi-norm definition

$$\|H^{LPTV}\|_{\ell_{2sn}}^2 := \limsup_{l \rightarrow \infty} \frac{1}{2l+1} \sum_{k=-l}^l \text{tr} \mathcal{E} \left[z_k^d \begin{pmatrix} z_k^d \\ z_k^d \end{pmatrix}^T \right].$$

Thus, we are considering a control design problem of the form

$$\begin{aligned} \min_K & \|\bar{L}_1 G_{cl}(K)\|_{\ell_{2sn}}^2 \\ \text{subject to: } & \|\bar{L}_i G_{cl}(K)\|_{\ell_{2sn}}^2 \leq \gamma_i, \quad i = 2, \dots, n \end{aligned} \quad (18)$$

where $G_{cl}(K)$ is, as before, the closed loop system for a particular controller, K .

The approach that we will take here is to use a lifting procedure [9] to convert the ℓ_2 semi-norm of a system to the \mathcal{H}_2 norm of a related system. Suppose that a given periodic system, G_{LPTV} has the realization

$$G_{LPTV} \sim \begin{pmatrix} \mathcal{A}(k) & \mathcal{B}(k) \\ \mathcal{C}(k) & \mathcal{D}(k) \end{pmatrix}.$$

Now define the LTI system, G_{LTI} by the realization

$$\begin{aligned} G_{LTI} & \sim \begin{pmatrix} Z^T A & Z^T B \\ C & D \end{pmatrix} \\ A & := \text{blkdiag} \{ \mathcal{A}(1), \dots, \mathcal{A}(N) \} \\ B & := \text{blkdiag} \{ \mathcal{B}(1), \dots, \mathcal{B}(N) \} \\ C & := \text{blkdiag} \{ \mathcal{C}(1), \dots, \mathcal{C}(N) \} \\ D & := \text{blkdiag} \{ \mathcal{D}(1), \dots, \mathcal{D}(N) \} \\ Z & := \begin{bmatrix} 0 & I_{n_x} & 0 \\ & \ddots & \ddots \\ & & \ddots & I_{n_x} \\ I_{n_x} & & & 0 \end{bmatrix} \end{aligned}$$

where n_x is the number of states in G_{LPTV} , I_{n_x} is the $n_x \times n_x$ identity matrix, and ‘blkdiag’ represents the operator which forms a block diagonal matrix whose blocks are respectively given by its arguments. Note that Z is an orthogonal matrix. The lifting procedure states that uniform exponential stability of G_{LPTV} is equivalent to stability of G_{LTI} and

$$\|G_{LPTV}\|_{\ell_{2sn}}^2 = \frac{1}{N} \|G_{LTI}\|^2.$$

(This is proved, for example, in [10]). Thus, we can see that

$$\|\bar{L}_i G_{LPTV}\|_{\ell_{2sn}}^2 = \|L_i G_{LTI}\|^2 \quad (19)$$

$$L_i := \frac{1}{\sqrt{N}} I_N \otimes \bar{L}_i \quad (20)$$

where \otimes represents the Kronecker product.

With this in mind, we now consider the controller design for the LTI system \bar{G} , which has the realization

$$\bar{G} \sim \begin{pmatrix} Z^T A & Z^T B_1 & Z^T B_2 \\ C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & 0 \end{pmatrix}$$

where A , B_1 , B_2 , C_1 , C_2 , D_{12} , and D_{21} are the lifted (i.e. block diagonal) counterparts of the state space entries of Eq. (17). To design a controller for this plant, we just apply the methods of Section 2 to the plant \bar{G} with the choice of L_i 's defined in Eq. (20).

Now we examine the structure of the optimal controller for \bar{G} . It can be shown [10] that the solution of Eq. (3a) must have the same block diagonal pattern as A . This implies that M , L , F_x , and F_w respectively have the form

$$\begin{aligned} M & = \text{blkdiag} \{ \mathcal{M}(1), \dots, \mathcal{M}(N) \} \\ ZL & = \text{blkdiag} \{ \mathcal{L}(1), \dots, \mathcal{L}(N) \} \\ F_x & = \text{blkdiag} \{ \mathcal{F}_x(1), \dots, \mathcal{F}_x(N) \} \\ F_w & = \text{blkdiag} \{ \mathcal{F}_w(1), \dots, \mathcal{F}_w(N) \}. \end{aligned}$$

Note that we have used the orthogonality of Z . Similarly, P , K_x , K_w , and Y respectively have the form

$$\begin{aligned} P & = \text{blkdiag} \{ \mathcal{P}(1), \dots, \mathcal{P}(N) \} \\ K_x & = \text{blkdiag} \{ \mathcal{K}_x(1), \dots, \mathcal{K}_x(N) \} \\ K_w & = \text{blkdiag} \{ \mathcal{K}_w(1), \dots, \mathcal{K}_w(N) \} \\ Y & = \text{blkdiag} \{ \mathcal{Y}(1), \dots, \mathcal{Y}(N) \}. \end{aligned}$$

Since these matrices are all compatibly block diagonal, a realization for the optimal LPTV controller is given by

$$\begin{aligned} \begin{bmatrix} x_{k+1}^o \\ u_k \end{bmatrix} & = \begin{bmatrix} (\bar{\mathcal{A}}_{11} + \mathcal{B}_2 \mathcal{K}_x + \mathcal{B}_2 \mathcal{H} \mathcal{C}_2) |_k & | & (-\mathcal{L} - \mathcal{B}_2 \mathcal{H}) |_k \\ \mathcal{K}_x + \mathcal{H} \mathcal{C}_2 |_k & | & (-\mathcal{H}) |_k \end{bmatrix} \begin{bmatrix} x_k^o \\ y_k \end{bmatrix} \\ \mathcal{H}(k) & := \mathcal{K}_x(k) \mathcal{F}_x(k) + \mathcal{K}_w(k) \mathcal{F}_w(k) \\ \bar{\mathcal{A}}_{11}(k) & := \mathcal{A}(k) + \mathcal{L}(k) \mathcal{C}_2(k). \end{aligned}$$

In the MATLAB implementations of the solvers for algebraic Riccati equations and Lyapunov equations, there is no way to specify the sparsity structure of the solution. This means that although these solvers will tend to give accurate results, they will not, in general, have the desired sparsity structure. To deal with this, we use a three step process. First, we solve the equation using the built-in solver. Then, impose the block diagonality constraint, i.e. we set all of the off-block-diagonal entries to zero. Finally, we use an iterative refinement process (such as the one in [11]) to recover the accuracy lost when imposing the block diagonality constraint. This process is applied when solving Eqs. (3a), (4a), (8b), (11), and (14).

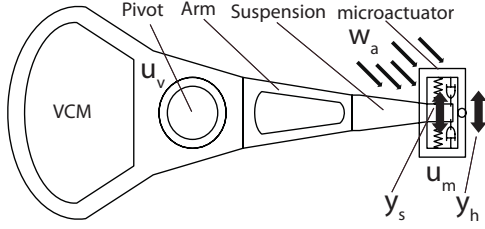


Figure 1. SCHEMATIC OF THE DUAL-STAGE HARD DRIVE

4 CONTROL OF HARD DRIVES

Figure 1 shows the structure of a hard disk drive with dual-stage actuation. The model of our system is given by

$$\begin{bmatrix} z(s) \\ y(s) \end{bmatrix} = G_{gen}(s) \begin{bmatrix} w(s) \\ u(s) \end{bmatrix} \quad (21)$$

where

$$z := \begin{bmatrix} y_h - r \\ y_h - y_s \\ u_v \\ u_m \end{bmatrix} \quad w := \begin{bmatrix} w_r \\ w_a \\ n_{PES} \\ n_{RPES} \end{bmatrix}$$

$$y := \begin{bmatrix} y_h - r + n_{PES} \\ y_h - y_s + n_{RPES} \end{bmatrix} \quad u := \begin{bmatrix} u_v \\ u_m \end{bmatrix}$$

$$\begin{bmatrix} y_h(s) \\ y_s(s) \end{bmatrix} := \sum_{i=1}^4 \begin{bmatrix} 1 & 0 \\ c^{1,i} & 0 \end{bmatrix} \left(sI - \begin{bmatrix} a^{1,i} & 1 \\ a^{2,i} & 0 \end{bmatrix} \right)^{-1} B^i \begin{bmatrix} w_a(s) \\ u_v(s) \\ u_m(s) \end{bmatrix}$$

$$r(s) := \left(\frac{2.8 \times 10^9}{s^2 + 800s + 2.5 \times 10^5} + \frac{1.2 \times 10^5}{s + 1.9 \times 10^3} \right) w_r(s)$$

and the model parameters are summarized in Tab. 1. In this model, the elements of w represent the disturbances to our system (zero-mean white Gaussian noise with unit covariance); w_r , w_a , n_{PES} , and n_{RPES} respectively represent disturbances on the head position (see [12]), airflow disturbance, PES sensor noise, and RPES sensor noise. The elements of u represent the control inputs; u_v and u_m respectively represent the control inputs into the voice coil motor (V) and the MA (V). The elements of z and y respectively represent the signals we would like to keep small in closed loop and our measurements; y_h and y_s respectively represent the head displacement relative to the center of the data track (nm) and the the suspension tip displacement (nm). Note that the first two elements in y are respectively the PES and the RPES. Figure 2 shows the relevant Bode magnitude plots.

In a physical hard drive, the PES is obtained by reading encoded position information on the servo track. Since this information is fabricated directly onto the disk, the PES sampling rate

Table 1. HDD MODEL PARAMETERS

mode i	$\bar{a}^{1,i}$	$\bar{a}^{2,i}$	$c^{1,i}$
1	-377	-1.42×10^5	1
2	-1390	-2.16×10^9	-3.177
3	-2020	-4.52×10^9	-4.246
4	-5650	-2×10^8	0

mode i	B^i
1	$\begin{bmatrix} 0 & -3750 & 0 \\ 0 & -1.99 \times 10^9 & 0 \end{bmatrix}$
2	$\begin{bmatrix} 0 & -5559 & 0 \\ 0 & -1.59 \times 10^8 & 0 \end{bmatrix}$
3	$\begin{bmatrix} 3.48 \times 10^4 & -1817 & 0 \\ 1.07 \times 10^9 & -5.59 \times 10^7 & 0 \end{bmatrix}$
4	$\begin{bmatrix} -3.48 \times 10^4 & 1.11 \times 10^4 & 0 \\ -1.2 \times 10^9 & 2.25 \times 10^9 & -4 \times 10^7 \end{bmatrix}$

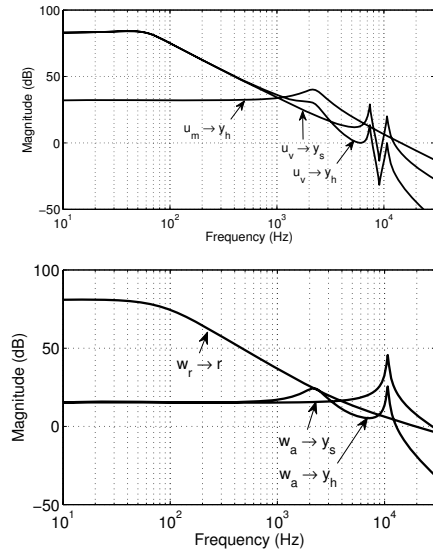


Figure 2. HDD BODE MAGNITUDE PLOTS

cannot be changed. In this paper, we assume that the fixed PES sampling rate is $\omega_0 := 25kHz$. However, unlike the PES, the rate of the RPES measurements and control actuation is not fixed at this rate. In particular, we could sample and actuate these signals at the rate $N\omega_0$, where N is a positive integer.

In this paper, we will consider designing controllers for $N = 1, 2, 4$. To begin, we first discretize the plant using a zero-order hold at the sampling rate $100kHz$, which results in the realization in Eq. (2). At this point, we incorporate the multirate sampling and actuation characteristics of the hard drive, i.e. we impose

the constraints that the PES can only be measured once every 4 time steps, the RPES can only be measured once every $4/N$ time steps, and the values of the control input can only be changed once every $4/N$ time steps.

The typical approach to enforcing the sampling and actuation constraints [13] will not work here because it would result in singular $D_{21}D_{21}^T$ and $D_{12}^TD_{12}$, which violates the conditions stated at the beginning of the paper. To deal with the sampling constraints in this control design methodology, we replace entries of y_k with “useless information” from the standpoint of the Kalman filter whenever they are not measured. Thus, if we define the diagonal periodically time-varying matrix $\Omega_y(k)$ so that its i^{th} diagonal element is 1 when the i^{th} element of y_k is being sampled and zero otherwise, we model this constraint by replacing the controller input with the signal \bar{y}_k defined by

$$\bar{y}_k = \Omega_y(k)y_k + (I - \Omega_y(k))\bar{w}_k$$

where \bar{w}_k is fictitious noise which is uncorrelated with w_k . To deal with the actuation constraints, we first define the diagonal periodically time-varying matrix $\Omega_u(k)$ so that its i^{th} diagonal element is 1 when the value of the i^{th} element of u_k is being changed and zero otherwise. With this, the control input to the plant can be written

$$u_k = \Omega_u(k)\bar{u}_k + (I - \Omega_u(k))u_{k-1}$$

where \bar{u}_k is the controller output. Now, if we define new state, performance, disturbance, measurement, and control vectors respectively as

$$\begin{aligned} x'_k &:= \begin{bmatrix} x_k \\ u_{k-1} \end{bmatrix}, & z'_k &:= \begin{bmatrix} (I - \Omega_u(k))\bar{u}_k \\ z_k \end{bmatrix}, & w'_k &:= \begin{bmatrix} \bar{w}_k \\ w_k \end{bmatrix}, \\ y'_k &:= \bar{y}_k, & u'_k &:= \bar{u}_k \end{aligned}$$

we can form a LPTV realization of our plant model (with incorporated multirate sampling and actuation constraints) which is suitable for controller design by the methods in sections 2 and 3. In this case, it is straightforward to show from the definitions of $\mathcal{L}(k)$, $\mathcal{F}_x(k)$, $\mathcal{F}_w(k)$, and $\mathcal{K}_w(k)$ that $\mathcal{L}(k)(I - \Omega_y(k)) = 0$, $\mathcal{F}_x(k)(I - \Omega_y(k)) = 0$, and $\mathcal{F}_w(k)(I - \Omega_y(k)) = 0$. Therefore, \bar{w}_k has no effect on the closed loop performance of the system for any dual feasible controller, which in turn means that this approach to modeling multirate sampling constraints is equivalent to the typical approach. If we choose our \bar{L}_i 's to have the form

$$\begin{aligned} \bar{L}_1 &= \text{blkdiag}\{W', L'_1\} \\ \bar{L}_i &= [0 \ L'_i], \quad i = 2, \dots, n \end{aligned}$$

where W' is nonsingular, it is straightforward to show that $(I - \Omega_u(k))\mathcal{K}_x(k) = 0$ and $(I - \Omega_u(k))\mathcal{K}_w(k) = 0$. This means

Table 2. RMS 3σ VALUES OF CLOSED LOOP SIGNALS, RICCATI EQUATION ALGORITHM

N	PES (nm)	RPES (nm)	u_v (V)	u_m (V)
1	14.560	426.050	5.000	11.374
2	10.253	64.905	4.975	9.324
4	8.824	49.956	4.218	19.996

that the unused controller output, $(I - \Omega_u(k))\bar{u}_k$, is forced to be zero. Moreover, the values of $\Omega_u(k)\bar{u}_k$ do not change as W' is changed, which implies that the J_i 's do not change. Thus, by adding the extra disturbances and performance outputs into our system model, we have modified the typical approach to incorporating multirate sampling and actuation into control design so that it works for this particular design methodology.

In our system, we would like to find a control scheme which minimizes the RMS 3σ value of the PES. Due to hardware limitations, we would like to keep the RMS 3σ values of u_v and u_m respectively smaller than 5V and 20V so that the actuators are not saturated. It should be noted that the control design being considered here is not practical for implementation due to robustness issues; we are instead using these control designs to determine the limits of PES performance for several choices of sampling and actuation rates. To represent this control design problem in the form of Eq. (18), we choose

$$\begin{aligned} \bar{L}_1 &= \text{blkdiag}\{I_2, [1 \ 0 \ 0 \ 0]\}, & \bar{L}_2 &= [0 \ 0 \ 0 \ 0 \ 1 \ 0], \\ \bar{L}_3 &= [0 \ 0 \ 0 \ 0 \ 0 \ 1], & \gamma_2 &= (5/3)^2, & \gamma_3 &= (20/3)^2. \end{aligned}$$

With these constraints, we designed controllers for $N = 1, 2, 4$. The RMS 3σ values of the relevant signals are summarized in Tab. 2. For each case, the gap between the achieved RMS 3σ PES value and the lower bound on its optimal value was less than $5 \times 10^{-4}nm$. This means that the algorithm didn't get stuck in local minima for these problems and achieved a high degree of numerical accuracy. Thus, by looking at these results, we know that we can obtain more than a 39% improvement in PES value just by sampling the RPES and actuating at a higher rate. Also note that the closed loop RPES, even though it is not a part of the cost function or constraints, decreases dramatically as N is increased.

For comparison, we also designed a controller for $N = 2$ via LMI optimization [10] using several optimization packages. When using either SeDuMi [14] or SDPT3 [15] with YALMIP [16] as an interface, the resulting controller gave RMS 3σ PES values of about 10^3nm . This highlights the fact that it is difficult in general to achieve a high degree of numerical accuracy using LMI optimization. When the function `mincx` in the Robust Control Toolbox was used, the LMI optimization gave the closed loop results summarized in Tab. 3. Although the resulting controller has comparable PES performance to the one designed using the approach presented in this paper, the values of the other

Table 3. RMS 3σ VALUES OF CLOSED LOOP SIGNALS, LMI ALGORITHM WITH $N = 2$

PES (nm)	RPES (nm)	u_v (V)	u_m (V)
10.507	661.435	5.053	18.140

relevant signals (i.e. RPES, u_v , and u_m) are larger than the ones resulting from the Riccati equation approach. This highlights the fact that, unlike controllers designed using the Riccati equation approach, controllers designed using LMI optimization are not Pareto optimal, i.e. it is possible to decrease the value of one of the J_i 's without increasing the value of any of the remaining J_i 's. Also, the value of the RMS 3σ value of u_v violates its constraint. The reason for this is that small numerical inaccuracies during the controller reconstruction process caused the state space entries of the controller to have small numerical errors, which in turn caused a degradation of closed loop performance. This highlights the fact that the approach in this paper is more favorable from the standpoint of numerical stability. The biggest difference between these two algorithms, though, is the computation time. On a Lenovo Thinkpad with a 2.2GHz Intel Core 2 Duo CPU and 2Gb of RAM, the approach presented in this paper took less than 4.2 seconds of computation, using SeDeMi took more than 80 seconds, using SDPT3 took more than 110 seconds, and using `mincx` took over 16.5 hours.

5 CONCLUSIONS

In this paper, we presented a new algorithm for constrained LQG controller design. We then used this algorithm as a mechatronic design tool to demonstrate that it is possible to improve its closed loop performance by more than 39% just by using multirate sampling and actuation. Also, it was shown through this example that the proposed algorithm has better properties than the LMI optimization in terms of numerical accuracy, numerical stability, the amount of required computation, and Pareto optimality.

ACKNOWLEDGMENT

This work was supported in part by National Science Foundation grant CMS-0428917, the Information Storage Industry Consortium and the Computer Mechanics Laboratory at UC Berkeley.

REFERENCES

[1] Oldham, K., Felix, S., Conway, R., and Horowitz, R., 2008. "Design and control of a dual-stage disk drive servo system with a high-aspect ratio electrostatic microactuator". In Proceedings of the 2008 American Control Conference.
 [2] Scherer, C., Gahinet, P., and Chilali, M., 1997. "Multi-objective output-feedback control via LMI optimization".

IEEE Transactions on Automatic Control, **42**(7), July, pp. 896–911.

[3] Mäkilä, P. M., Westerlund, T., and Toivonen, H. T., 1984. "Constrained linear quadratic gaussian control with process applications". *Automatica*, **20**(1), pp. 15–29.
 [4] Zhu, G., Rotea, M. A., and Skelton, R., 1997. "A convergent algorithm for the output covariance constraint control problem". *SIAM Journal on Control and Optimization*, **35**(1), pp. 341–361.
 [5] Collins, E. G., and Selekwa, M. F., 2002. "A fuzzy logic approach to LQG design with variance constraints". *IEEE Transactions on Control Systems Technology*, **10**(1), January, pp. 32–42.
 [6] Oldham, K., Huang, X., and Horowitz, R., 2005. "Design, fabrication, and control of a high-aspect ratio microactuator for vibration suppression in a hard disk drive". In Proceedings of the 16th IFAC World Congress.
 [7] Boyd, S., and Vandenberghe, L., 2004. *Convex Optimization*. Cambridge University Press, New York, NY, USA.
 [8] Delchamps, D. F., 1983. "Analytic stabilization and the algebraic Riccati equation". In Proceedings of the 22nd IEEE Conference on Decision and Control, pp. 1396–1401.
 [9] Dullerud, G. E., and Paganini, F., 2000. *A Course in Robust Control Theory: A Convex Approach*. Springer-Verlag, NY.
 [10] Conway, R., 2008. Multi-objective control design for discrete time periodic systems via convex optimization. Tech. rep., University of California at Berkeley.
 [11] Arnold, W. F., and Laub, A. J., 1984. "Generalized eigenproblem algorithms and software for algebraic riccati equations". In Proceedings of the IEEE, Vol. 72, pp. 1746–1754.
 [12] Huang, X., Nagamune, R., and Horowitz, R., 2006. "A comparison of multirate robust track-following control synthesis techniques for dual-stage and multi-sensing servo systems in hard disk drives". *IEEE Transactions on Magnetics*, **42**(7), July, pp. 1896–1904.
 [13] Nagamune, R., Huang, X., and Horowitz, R., 2005. "Multirate track-following control with robust stability for a dual-stage multi-sensing servo system in HDDs". In Proceedings of the Joint 44th IEEE Conference on Decision and Control and European Control Conference 2005, pp. 3886–3891.
 [14] Sturm, J. F., 1999. "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones". *Optimization Methods and Software*, **11-12**, pp. 625–653. Special issue on Interior Point Methods (CD supplement with software).
 [15] Toh, K. C., Todd, M. J., and Tutuncu, R., 1999. "SDPT3 — a Matlab software package for semidefinite programming". *Optimization Methods and Software*, **11**, pp. 545–581.
 [16] Löfberg, J., 2004. "YALMIP : A toolbox for modeling and optimization in MATLAB". In Proceedings of the CACSD Conference.