Nonlinear Q-Design for Convex Stochastic Control

Joëlle Skaf and Stephen Boyd, *Fellow, IEEE*

***Abstract*—In this note we describe a version of the Q-design method that can be used to design nonlinear dynamic controllers for a discrete-time linear time-varying plant, with convex cost and constraint functions and arbitrary disturbance distribution. Choosing a basis for the nonlinear Q-parameter yields a convex stochastic optimization problem, which can be solved by standard methods such as sampling. In principle (for a large enough basis, and enough sampling) this method can solve the controller design problem to any degree of accuracy; in any case it can be used to find a suboptimal controller, using convex optimization methods. We illustrate the method with a numerical example, comparing a nonlinear controller found using our method with the optimal linear controller, the certainty-equivalent model predictive controller, and a lower bound on achievable performance obtained by ignoring the causality constraint.**

***Index Terms*—Convex optimization, nonlinear control, Q-parameter, stochastic control.**

I. Introduction

We consider the stochastic control problem for a finite-horizon discrete-time linear time-varying system with convex objective and constraints. The optimal controller can be described recursively via dynamic programming, but this gives a practical method for implementing the controller only in a few special cases, such as when the state-space dimension is very low (say, no more than two or three), or when the objective is quadratic and there are no constraints. On the other hand there are many methods for finding a suboptimal controller, including classical control techniques, model predictive control, and approximate dynamic programming (which will be described in more detail later).

In this note we show how the Youla or Q-parametrization, suitably extended, can be used to convert the controller design problem into an equivalent stochastic convex optimization problem. By choosing a large enough basis of nonlinear functions, and approximate solution of the resulting stochastic convex optimization problem, we can (at least in principle), solve the controller design problem to any desired accuracy, using standard (finite-dimensional) convex optimization techniques. In any case, our method can be used to find a suboptimal controller, using standard convex optimization methods. We illustrate our method with an example, in which the synthesized controller yields substantially better performance than a linear controller, or a model predictive controller, despite an ad hoc choice of basis. A more detailed version of this note, including the data and source code for the numerical example, can be found in [1].

II. System Model

A. Signals and Plant

We consider a discrete-time linear time-varying plant, over the time interval $t = 1, \ldots, T$, with control or actuator signal $u_1, \ldots, u_T \in \mathbf{R}^m$, sensor signal $y_1, \ldots, y_T \in \mathbf{R}^p$, disturbance or exogenous input signal $w_1, \ldots, w_T \in \mathbf{R}^n$, and output signal $z_1, \ldots, z_T \in \mathbf{R}^q$. As explained in [2, Chap. 2], the exogenous input signal can represent plant and measurement noise, as well as command, reference, or tracking signals; the exogenous output signal can represent the regulated variables or tracking errors. We will use w, u, y, and z to denote the associated trajectories:

$$\begin{aligned} u &= (u_1, \ldots, u_T) \in \mathbf{R}^{mT} \\ y &= (y_1, \ldots, y_T) \in \mathbf{R}^{pT} \\ w &= (w_1, \ldots, w_T) \in \mathbf{R}^{nT} \\ z &= (z_1, \ldots, z_T) \in \mathbf{R}^{qT}. \end{aligned}$$

The plant is linear, and so can be described by

$$\begin{bmatrix} z \\ y \end{bmatrix} = \begin{bmatrix} P^{\mathrm{zw}} & P^{\mathrm{zu}} \\ P^{\mathrm{yw}} & P^{\mathrm{yu}} \end{bmatrix} \begin{bmatrix} w \\ u \end{bmatrix} \tag{1}$$

where P, the matrix above, is the plant input-output matrix.

B. Causality

To describe causality assumptions and restrictions we will need the idea of block lower triangularity. Suppose we write a matrix $A \in \mathbf{R}^{rk \times sl}$ in (k, l) block form, where $A_{ij} \in \mathbf{R}^{k \times l}$ denotes the i, j block. If $A_{ij} = 0$ for $i < j$, we say that A is (k, l) *block lower triangular*. If in addition the diagonal blocks vanish, i.e., $A_{ij} = 0$ for $i \le j$, we say the matrix is (k, l) *block strictly lower triangular*. We extend these concepts to (possibly nonlinear) functions as follows. Suppose that $F : \mathbf{R}^{sl} \to \mathbf{R}^{rk}$, with $F(x) = (F_1(x), \ldots, F_r(x))$, where $F_i(x) \in \mathbf{R}^k$, and $x = (x_1, \ldots, x_s)$, with $x_j \in \mathbf{R}^l$. We say that F is (k, l) block lower triangular if $F_i(x)$ depends only on $x_1, \ldots, x_i$, and (k, l) block strictly lower triangular if $F_i(x)$ depends only on $x_1, \ldots, x_{i-1}$. When F is block lower triangular, we write F_i as $F_i(x_1, \ldots, x_i)$, to emphasize that it depends only on $x_1, \ldots, x_i$. When the indices represent time, block lower triangularity corresponds to causality, and block strict lower triangularity corresponds to strict causality.

We will assume in what follows that P^{yu} is (p, m) block strictly lower triangular, which means that the mapping from actuator to sensor is strictly causal. It often occurs that P^{yw}, P^{zw}, and P^{zu} are also block lower triangular, but we do not need this assumption.

C. Causal Controller

We will consider causal feedback controllers (or control policies), for which $u = \varphi(y)$, where $\varphi : \mathbf{R}^{pT} \to \mathbf{R}^{mT}$ is (p, m) block lower triangular. We can write this out as

$$u_t = \varphi_t(y_1, \ldots, y_t), \quad t = 1, \ldots, T.$$

Since P^{yu} is block strictly lower triangular, the system equations are always well-posed. Indeed, the trajectories of u and y are readily found from the recursion

$$\begin{aligned} y_1 &= (P^{\mathrm{yw}} w)_1 \\ u_1 &= \varphi_1(y_1) \\ y_2 &= (P^{\mathrm{yw}} w)_2 + P^{\mathrm{yu}}_{2,1} u_1 \\ u_2 &= \varphi_2(y_1, y_2) \\ &\vdots \\ y_T &= (P^{\mathrm{yw}} w)_T + P^{\mathrm{yu}}_{T,1} u_1 + \cdots + P^{\mathrm{yu}}_{T,T-1} u_{T-1} \\ u_T &= \varphi_T(y_1, \ldots, y_T). \end{aligned}$$

We can then find z from $z = P^{\mathrm{zw}} w + P^{\mathrm{zu}} u$. Once we fix the controller φ, the actuator signal u and output z become functions of the exoge-

Manuscript received July 16, 2008; revised July 17, 2008, February 17, 2009, and June 29, 2009. First published September 18, 2009; current version published October 07, 2009. This was supported by the Focus Center for Circuit & System Solutions (C2S2), by the Precourt Institute on Energy Efficiency, by NSF Award 0529426, by NASA Award NNX07AEIIA, by AFOSR Award FA9550-06-1-0514, and by AFOSR Award FA9550-06-1-0312. Recommended by P. A. Parrilo.

The authors are with the Information Systems Lab, Electrical Engineering Department, Stanford University, Stanford, CA 94305-9510 USA (e-mail: jskaf@stanford.edu; boyd@stanford.edu).

Digital Object Identifier 10.1109/TAC.2009.2029300

nous input w. We refer to these as the closed-loop exogenous input to actuator mapping, and the closed-loop exogenous input to output mapping, respectively.

III. Stochastic Control Problem

We assume that w is random, with known distribution. Via the closed-loop mappings, the actuator signal u and the output z are also random variables. Let $\phi : \mathbf{R}^{qT} \times \mathbf{R}^{mT} \to \mathbf{R}$ be a convex (objective) function. We judge our control performance by the expected value of this objective function, $\mathbf{E}\phi(z,u)$, where the expectation is over w. We treat constraints in a similar way. Let $\psi_i : \mathbf{R}^{qT} \times \mathbf{R}^{mT} \to \mathbf{R}$, $i = 1,\ldots,M$, be a set of convex (constraint) functions. Our control design constraints are $\mathbf{E}\psi_i(z,u) \le 0$, $i = 1,\ldots,M$. Such constraints, which require the expected value of a function to be less than zero (say) are called *stochastic constraints*. But the same form can be used to enforce an *almost-sure constraint*, such as $(z,u) \in \mathcal{C}$ almost surely. Here we simply define $\psi(z,u) = 0$ for $(z,u) \in \mathcal{C}$, and $\psi(z,u) = \infty$ for $(z,u) \notin \mathcal{C}$; the stochastic constraint $\mathbf{E}\psi(z,u) \le 0$ is then equivalent to $(z,u) \in \mathcal{C}$ almost surely.

The stochastic controller design problem can then be expressed as

$$\begin{array}{ll} \text{minimize} & \mathbf{E}\phi(z,u) \\ \text{subject to} & \mathbf{E}\psi_i(z,u) \le 0, \quad i = 1,\ldots,M \\ & \varphi \text{ block } (m,p) \text{ block lower triangular} \end{array} \tag{2}$$

where the expectation is over w, and the (infinite dimensional) optimization variable is the function (control policy) φ.

The stochastic control problem can be solved in a few very special cases. For example, in the linear quadratic Gaussian (LQG) problem (i.e., when there are no constraints, w is Gaussian, and ϕ is (convex) quadratic), the optimal policy φ is an affine function, i.e., a linear function of the past measurements, plus a constant, and can be found from dynamic programming (see [3], [4]) or by the method described below. Another case in which the optimal control policy is known, and affine, is when the cost is the exponential of a quadratic function, and the disturbances are Gaussian, which is the linear exponential quadratic Gaussian (LEQG) or risk-sensitive LQG problem [5, vol. 1, §19].

A. Controller Design Methods

There is a large number of heuristic methods for solving the stochastic control problem, also called stochastic optimization with recourse [5]–[9]. Perhaps the simplest methods are those from classical linear feedback control techniques, such as PID (proportional-integral-derivative) control [10]. One very effective technique that can be used when a noiseless measurement of the state is available is model predictive control (MPC) [5], [11]–[15], which also goes by many other names, including dynamic matrix control [16], rolling horizon planning [17], and dynamic linear programming (DLP) [18]. MPC is based on solving a convex optimization problem at each step, with the unknown future disturbances replaced with some kind of estimates available at the current time (such as conditional means); but only the current action or input is used. At the next step, the same problem is solved, this time using the exact value of the current state, which is now known from the measurement. Another approach goes under the name approximate dynamic programming [7], [19], [20], in which some estimate of the optimal value function, or optimal policy, is found.

We also mention that there is a large literature on multistage stochastic linear programming, which can be used to solve (exactly or approximately) some versions of our problem (with piecewise linear objectives and polyhedral constraints); see [21]–[29]. The proposed methods range from decomposition and partitioning methods to sampling-based approximation algorithms, and are usually limited to short horizons.

IV. Nonlinear Q-Parametrization

We define the signal e as

$$e = y - P^{\mathrm{yu}}u = P^{\mathrm{yw}}w.$$

We can think of e as the sensor signal, with the effect of the actuator signal removed, i.e., e is the direct effect of the exogenous input w on the sensor signal, after compensating for its effect via feedback.

Now let $\mathcal{Q} : \mathbf{R}^{pT} \to \mathbf{R}^{mT}$ be any (m,p) block lower triangular function. We define a causal control policy φ via the relation

$$u = \varphi(y) = \mathcal{Q}(e) \tag{3}$$

which can be more explicitly written, using $e = y - P^{\mathrm{yu}}u$, as the recursion

$$\begin{aligned} u_1 &= \mathcal{Q}_1(y_1) \\ u_2 &= \mathcal{Q}_2\left(y_1, y_2 - P^{\mathrm{yu}}_{2,1}u_1\right) \\ &\vdots \\ u_T &= \mathcal{Q}_T\left(y_1,\ldots,y_T - P^{\mathrm{yu}}_{T,1}u_1 - \cdots - P^{\mathrm{yu}}_{T,T-1}u_{T-1}\right). \end{aligned}$$

This recursion shows that u_t is a function of $y_1,\ldots,y_t$, i.e., φ is (m,p) block lower triangular.

We have seen that from any (m,p) block lower triangular function $\mathcal{Q}$, we can construct an (m,p) block lower triangular controller φ. We will now show that *every* causal controller φ can be realized by some choice of $\mathcal{Q}$. Let φ be a given (m,p) block lower triangular controller. Using (3) and $y = e + P^{\mathrm{yu}}u$, we obtain a recursion that defines $\mathcal{Q}(e) = u$ as

$$\begin{aligned} u_1 &= \varphi_1(e_1) \\ u_2 &= \varphi_2\left(e_1, e_2 + P^{\mathrm{yu}}_{2,1}u_1\right) \\ &\vdots \\ u_T &= \varphi_T\left(e_1,\ldots,e_T + P^{\mathrm{yu}}_{T,1}u_1 + \cdots + P^{\mathrm{yu}}_{T,T-1}u_{T-1}\right). \end{aligned}$$

This recursion shows that $\mathcal{Q}$ is (m,p) block lower triangular. When the construction above is applied to $\mathcal{Q}$, we obtain the original causal controller φ.

We conclude that the correspondence between φ and $\mathcal{Q}$ is a bijection: For each (m,p) block lower triangular function φ, there is exactly one (m,p) block lower triangular $\mathcal{Q}$, given by the recursion above. It follows that we can optimize over $\mathcal{Q}$ in the stochastic control problem, instead of φ. In other words, we parametrize φ by $\mathcal{Q}$.

We can express u and z in terms of w and $\mathcal{Q}$, using $e = P^{\mathrm{yw}}w$:

$$u = \mathcal{Q}(P^{\mathrm{yw}}w), \quad z = P^{\mathrm{zw}}w + P^{\mathrm{zu}}\mathcal{Q}(P^{\mathrm{yw}}w).$$

These expressions show that z and u are (in general) nonlinear functions of w, but they are *affine* functions of $\mathcal{Q}$, for each w. Since expectation preserves convexity, it follows that the objective and constraint functions:

$$\mathbf{E}\phi(z,u), \quad \mathbf{E}\psi_1(z,u),\ldots,\mathbf{E}\psi_M(z,u)$$

are convex functions (or, more formally, functionals) of $\mathcal{Q}$. By using the variable $\mathcal{Q}$ instead of φ, we now have an infinite-dimensional stochastic convex optimization problem.

A. Q-Design Procedure

Our method is related to the classical Q-design procedure, or Youla parametrization [30], [31] for time-invariant, infinite-horizon linear controller design [32]–[36]. The book [2] and survey paper [37] describes this method, and the use of convex optimization to design continuous-time, time-invariant controllers, in detail; the Notes and References trace the ideas back into the 1960s. The books [38], [39] use these methods to minimize the worst-case output, with unknown but bounded input, which can be cast (after Q-parametrization) as an ℓ_1 norm minimization problem, and then solved using linear programming. The Ph.D. thesis [40] and the article [41] use the Q-design procedure to formulate the controller design problem as a constrained convex optimization problem. Our method is also related to the more recent purified output control method [42], [43].

Although not directly related to our topic, we mention some papers in which extensions of Q-design are used to design stabilizing controllers for nonlinear plants; our method, in contrast, concerns linear plants and nonlinear controllers. In [44], Desoer and Liu show the existence of a parametrization of stabilizing controllers for stable nonlinear plants. For unstable nonlinear plants, only partial results have been obtained (e.g., see [45]–[47]).

V. Approximate Solution

A. Finite-Dimensional Restriction

We can obtain an approximate solution of the stochastic controller design problem, with variable $\mathcal{Q}$, by choosing a basis $\mathcal{Q}^{(1)}, \ldots, \mathcal{Q}^{(K)}$ of (m,p) block lower triangular (and generally nonlinear) functions, and expressing $\mathcal{Q}$ as

$$\mathcal{Q} = \sum_{i=1}^{K} \alpha_i \mathcal{Q}^{(i)} \tag{4}$$

where $\alpha_1, \ldots, \alpha_K$ are the design variables. We now have a finite-dimensional stochastic convex optimization problem. If we were to solve this problem, we would have a suboptimal solution of the stochastic controller design problem, obtained by expressing φ in terms of the $\mathcal{Q}$ found.

We could also (very roughly) claim that, if the basis were large or rich enough, that we have 'nearly' solved the stochastic controller design problem; the only limit to our finding the optimal controller is our choice of basis for $\mathcal{Q}$. But we will focus here on the less ambitious claim that this method simply produces a suboptimal controller, just like the many other methods listed above.

B. Solving the Stochastic Optimization Problem

Once a basis for $\mathcal{Q}$ has been chosen, our problem becomes a finite-dimensional stochastic convex optimization problem. There is a large literature on this topic; see, e.g., [5], [7], [8], [48]–[50]. In a few very special cases we can solve such problems exactly. For example, when ϕ and ψ_i are quadratic, and the mean and covariance of $\mathcal{Q}(P^{\rm yw}w)$ can be computed, the controller design problem reduces to a (convex) quadratic program. But in general, we have to solve the stochastic problem approximately. Typical methods involve a parameter which trades off computational effort and accuracy; arbitrary accuracy can (in principle) be obtained as the parameter (and computational effort) grows.

We will describe here the simplest approximation method for stochastic optimization, based on sampling. Choose $w^{(1)}, \ldots, w^{(N)}$ from the distribution of w, and form

$$z^{(i)} = P^{\rm zw} w^{(i)} + \sum_{j=1}^{K} \alpha_j P^{\rm zu} \mathcal{Q}^{(j)} \left(P^{\rm yw} w^{(i)}\right)$$

$$u^{(i)} = \sum_{j=1}^{K} \alpha_j P^{\rm zu} \mathcal{Q}^{(j)} \left(P^{\rm yw} w^{(i)}\right)$$

for $i = 1, \ldots, N$. These are affine functions of α. Our approximate problem is then

$$\begin{array}{ll} \text{minimize} & \frac{1}{N} \sum_{i=1}^{N} \phi\left(z^{(i)}, u^{(i)}\right) \\ \text{subject to} & \frac{1}{N} \sum_{i=1}^{N} \psi_l\left(z^{(i)}, u^{(i)}\right) \leq 0,\ l = 1, \ldots, M \end{array} \tag{5}$$

with variable $\alpha \in \mathbf{R}^K$. This is a standard finite dimensional convex optimization problem, which can be solved using standard techniques; see, e.g., [51].

Once we solve the convex optimization problem (5), we can check whether we have taken enough samples, i.e., whether N is large enough, by validation, i.e., evaluating the solution found on *another* set of (typically, more) samples. If the empirical means of the objective or constraint functions substantially differ, between the original sample set and the validation sample set, we must increase N; if they are near each other, it gives us confidence that the sampling is adequate.

We note that when sampling is applied to an almost sure constraint, we are guaranteed that the constraint will hold for the original set of samples; but with some of the validation samples of w, the constraint can be violated. If the sample size is large enough, however, we expect that the constraint will hold with very high probability, and that when the violations occur, they will be small.

To get to the finite-dimensional convex optimization problem (5), we have made two approximations: We have restricted our search to a K-dimensional subspace of the infinite dimensional space of (m,p) block lower triangular functions, and we have approximately solved the stochastic problem by sampling. The second (sampling) approximation is generally good, but the first one (restriction to a finite dimension subspace) is generally not. For this reason we generally would not claim that this method solves the stochastic controller design problem in practice, even when K is large; we can simply claim that the method yields a good controller in a straightforward way, that relies on convex optimization.

Finally, we mention the rough computational complexity of the *design* method. The main effort is in solving the problem (5), which has K variables and M constraints; each of these, in turn, involves N terms. Evaluating the objective and constraints, and their gradients and Hessians, requires on the order of MNK^2 operations. Since this is the dominant effort per iteration of an interior-point method, which in practice require a few or several tens of iterations, we get an overall complexity estimate of MNK^2. We note that this grows linearly with N, the number of samples; this makes is practical to choose a relatively large value of N. The complexity estimate above assumes that no problem structure is exploited; however, if any problem structure is exploited, the complexity can be reduced further [51].

We note that the final controller is *implemented* using the recursions above, and so can run at extremely high rates, assuming the basis functions $\mathcal{Q}_i$ can be rapidly evaluated. In particular, no optimization problem is solved at run time, as in MPC. In other words, the run-time complexity of the method is very low. (But we should mention that the optimization problems that must be solved in each MPC step can be solved quite efficiently; see, e.g., [52].)

C. Standardization

Here we describe a method for standardizing the signal e, i.e., applying a causal linear whitening transformation to it. This transformation is not essential, but it helps in coming up with a reasonable choice for the basis functions. We define

$$e^{\rm std} = L^{-1}(e - \mathbf{E}e)$$

where $\mathbf{E}e = P^{\rm yw}\mathbf{E}w$ is the mean value of e, and L is the (lower triangular) Cholesky factor of the covariance of e

$$LL^T = \mathbf{E}(e - \mathbf{E}e)(e - \mathbf{E}e)^T = P^{\rm yw}\Sigma_w (P^{\rm yw})^T$$

where Σ_w is the covariance matrix of w. The signal $e^{\rm std}$ is standardized, i.e., its entries have zero mean, unit variance, and are uncorrelated. Note that the standardization mapping, from e to $e^{\rm std}$, is lower triangular. We can (loosely) interpret $e^{\rm std}$ as an innovations signal, i.e., the component of the sensor signal that is not (linearly) predictable from the past.

The standardized signal e^{std} can be expressed in terms of e in a natural feedback form, which is called forward substitution in the context of numerical linear algebra

$$e_t^{\mathrm{std}} = L_{tt}^{-1}\left(e_t - \mathbf{E}e_t - L_{t,t-1}e_{t-1} - \cdots - L_{t,1}e_1\right)$$

for $t = 1, \ldots, T$. Here $L_{t,\tau}$ refers to the (t,τ) block of L, interpreted as a (p,p) block matrix.

We now choose our basis for $\mathcal{Q}$ as

$$\mathcal{Q}^{(i)}(e) = \tilde{\mathcal{Q}}^{(i)}(e^{\mathrm{std}}) = \tilde{\mathcal{Q}}^{(i)}\left(L^{-1}(e - \mathbf{E}e)\right), \quad i = 1, \ldots, K$$

where $\tilde{\mathcal{Q}}^{(i)}$ are (m,p) block lower triangular. The functions $\mathcal{Q}^{(i)}i$ are also (m,p) block lower triangular, since the composition of the (m,p) block lower triangular function $\tilde{\mathcal{Q}}^{(i)}$ and the lower triangular function $L^{-1}(e - \mathbf{E}e)$ is also (m,p) block lower triangular.

Since we know that the entries of e^{std} have zero mean and unit variance, we can choose nonlinear functions $\tilde{\mathcal{Q}}^{(i)}$ appropriate for this range of values. As a simple example, suppose that $\tilde{Q}^{(i)}$ saturates the ith component of e^{std} at the level β

$$\tilde{\mathcal{Q}}^{(i)}(e^{\mathrm{std}}) = \mathbf{sat}\left(e_i^{\mathrm{std}}/\beta\right)$$

where $\beta > 0$ and the saturation function $\mathbf{sat} : \mathbf{R} \to \mathbf{R}$ is defined as

$$\mathbf{sat}(a) = \max\{-1, \min\{a, 1\}\}.$$

A choice of β on the order of one would place the saturation level near the standard deviation of e_i^{std}, and is likely to produce an interesting nonlinear function; in contrast, the choice $\beta = 10$ would result in $\mathcal{Q}^{(i)}(e^{\mathrm{std}}) = e_i^{\mathrm{std}}/\beta$ with probability at least 0.99, by the Chebyshev inequality.

VI. Example

In this section we describe our numerical example. Space limitations do not allow us to give a full discussion of, or specification of, our example here; for much more detail about this example, and all data and code for it, see [1]. We consider a tracking problem with a scalar input, disturbance, sensor, and output signals. The tracking error, which is also the sensor signal, is given by

$$z = y = P^{\mathrm{yu}}u - w.$$

(Thus $P^{\mathrm{zw}} = P^{\mathrm{yw}} = -I$ and $P^{\mathrm{zu}} = P^{\mathrm{yu}}$.) Here we interpret w as the signal to track; we assume it has zero mean and covariance Σ_w. Our objective is the mean-square tracking error

$$\mathbf{E}\phi(z,u) = \mathbf{E}\sum_{t=1}^{T} z_t^2.$$

The actuator input signal u_t must satisfy the (almost sure) constraint

$$|u_t| \le u_{\max}, \qquad t = 1, \ldots, T.$$

Now we describe (our basis for) $\tilde{\mathcal{Q}}$. Let $\beta_1, \ldots, \beta_L$ be positive saturation levels. We take $\tilde{Q}$ be to

$$\tilde{Q}(e^{\mathrm{std}}) = F_1\mathbf{sat}(e^{\mathrm{std}}/\beta_1) + \cdots + F_L\mathbf{sat}(e^{\mathrm{std}}/\beta_L)$$

where F_i are lower triangular coefficient matrices, with last (block) row zero. (The coefficients α_i are the nonzero entries in $F_1, \ldots, F_L$.) The total dimension is thus $K = LT(T-1)/2$.

The particular problem instance we consider has horizon $T = 10$, with the entries of P^{yu} chosen randomly. We take the actuator signal limit to be $u_{\max} = 1/2$. Our basis for $\tilde{\mathcal{Q}}$ will use two saturated versions of the standardized signal e_{std}, with levels $\beta_1 = 0.5$ and $\beta_2 = 1$. The total dimension of our basis is therefore 90. We solve the design problem using the sampling method, with $N = 2000$ (training) samples, and verify the results by simulation on another (validation) set of 10000 samples.

TABLE I
RESULTS

	Training set	Validation set
Control law	$\mathbf{E}\phi(z,u)$	$\mathbf{E}\phi(z,u)$
Optimal linear	8.07	8.04
CE-MPC	7.85	7.81
Nonlinear	7.25	7.24
Prescient	(5.62)	(5.59)

We compare the performance of the nonlinear controller found using our method to the optimal linear controller, designed using the same training set (see [53]), and the certainty-equivalent model predictive control (CE-MPC). We also show the results obtained with a prescient controller, i.e., a controller that is not causal (which, of course, gives us a lower bound on achievable performance).

The results are shown in Table I, along with the performance of the (non-causal) prescient controller, which provides a lower bound on achievable performance. We can see that the nonlinear controller beats the optimal linear controller and CE-MPC. Its performance is around 30% higher than the lower bound given by the (non-causal) prescient controller.

As mentioned in Section V, the almost-sure constraint on $\|u\|_\infty \le u_{\max}$ is guaranteed to hold for all samples in the training set, but not for samples from the validation set. It does however hold with very high probability. For the optimal linear controller, the constraint is violated for 0.16% of the samples in the validation set. For the nonlinear controller, it is violated for only 0.04% of the samples, and when these violations occur, they are very small, with $\|u\|_\infty$ typically a fraction of one percent larger than $u_{\max}$.

VII. Conclusion

We have shown that the problem of finding the optimal nonlinear controller, for a stochastic control problem with linear dynamics and convex cost and constraint functions, can itself be cast as an infinite-dimensional convex stochastic optimization problem, after a nonlinear change of variables. After choosing a finite-dimensional basis, this problem can be approximately solved using standard techniques for numerical solution of stochastic convex optimization problems.

The (big) question that remains is: How should one choose the finite-dimensional basis of nonlinear operators over which to search? Unlike the case of linear controllers, where this question can be answered, we know of no satisfactory general method for choosing this basis. The standardization technique described above at least sets the approximate range of input values for the basis elements, but does not answer the general question of how they should be chosen.

Numerical results show that the method (even when the basis is chosen in an ad hoc way) synthesizes controllers with good, and often competitive, performance.

References

[1] J. Skaf and S. Boyd, Nonlinear Q-Design for Convex Stochastic Control Tech. Rep., 2008 [Online]. Available: http://www.stanford.edu/~boyd/papers/nonlin_Q_param.html

[2] S. Boyd and C. Barratt, *Linear Controller Design: Limits of Performance*. Englewood Cliffs, NJ: Prentice-Hall, 1991.

[3] B. Anderson and J. Moore, *Optimal Control—Linear Quadratic Methods*. Englewood Cliffs, NJ: Prentice-Hall, 1990.

[4] R. Stengel, *Stochastic Optimal Control*. New York: Wiley, 1986.

[5] P. Whittle, *Optimization Over Time: Dynamic Programming and Stochastic Control*. New York: Wiley, 1982.

[6] K. Åström, *Introduction to Stochastic Control Theory*. New York: Dover, 2006.

[7] D. Bertsekas, *Dynamic Programming and Optimal Control*. Nashua, NH: Athena Scientific, 2005.

[8] J. Birge and F. Louveaux, *Introduction to Stochastic Programming*. New York: Springer, 1997.

[9] A. Prekopa, *Stochastic Programming*. Norwell, MA: Kluwer, 1995.
[10] G. Franklin, J. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*. Reading, MA: Addison-Wesley, 1994.
[11] A. Bemporad, "Model predictive control design: New trends and tools," in *Proc. Conf. Decision Control*, 2006, pp. 6678–6683.
[12] W. Kwon and S. Han, *Receding Horizon Control*. Berlin, Germany: Springer-Verlag, 2005.
[13] R. Bitmead, M. Gevers, and V. Wertz, *Adaptive Optimal Control: The Thinking Man's GPC*. Englewood Cliffs, NJ: Prentice-Hall, 1990.
[14] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, Jun. 2000.
[15] S. Meyn, *Control Techniques for Complex Networks*. Cambridge, U.K.: Cambridge Univ. Press, 2006.
[16] C. Cutler, "Dynamic Matrix Control: An Optimal Multivariable Control Algorithm With Constraints," Ph.D. Dissertation, Univ. Houston, Houston, TX, 1983.
[17] E. Cho, K. Thoney, T. Hodgson, and R. King, "Supply chain planning: Rolling horizon scheduling of multi-factory supply chains," in *Proc. Conf. Winter Simul.: Driving Innovat.*, 2003, pp. 1409–1416.
[18] K. Talluri and G. V. Ryzin, *The Theory and Practice of Revenue Management*. New York: Springer, 2005.
[19] D. Bertsekas, *Neuro-Dynamic Programming*. Nashua, NH: Athena Scientific, 1996.
[20] W. Powell, *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. New York: Wiley, 2007.
[21] J. Birge, "Decomposition and partitioning methods for multistage stochastic linear programs," *Oper. Res.*, vol. 33, no. 5, pp. 989–1007, 1985.
[22] J. Birge and F. Louveaux, "A multicut algorithm for two-stage stochastic linear programs," *Eur. J. Oper. Res.*, vol. 34, pp. 384–392, 1988.
[23] M. Casey and S. Sen, "The scenario generation algorithm for multistage stochastic linear programming," *Math. Oper. Res.*, vol. 30, no. 3, pp. 615–631, 2005.
[24] G. Dantzig and G. Infanger, "Multi-stage stochastic linear programs for portfolio optimization," *Annals Oper. Res.*, vol. 45, pp. 59–76, 1993.
[25] H. Gassmann, "MSLiP: A computer code for the multistage stochastic linear programming problem," *Math. Programming*, vol. 47, pp. 407–423, 1990.
[26] A. Gupta, M. Pál, R. Ravi, and A. Sinha, "What about Wednesday? Approximation algorithms for multistage stochastic optimization," *Approximation, Randomization and Combinatorial Optimization*, vol. 3624, pp. 86–98, 2005.
[27] P. Kall and J. Mayer, *Stochastic Linear Programming: Models, Theory, and Computation*. New York: Springer, 2005.
[28] C. Swamy and D. Shmoys, "Sampling-based approximation algorithms for multi-stage stochastic optimization," in *Proc. 46th Annu. IEEE Symp. Foundations Comp. Sci.*, 2005, pp. 357–366.
[29] R. Van Slyke and R. Wets, "L-shaped linear programs with applications to optimal control and stochastic linear programming," *SIAM J. Appl. Math.*, vol. 17, pp. 638–663, 1969.
[30] D. Youla, J. Bongiorno, and H. Jabr, "Modern Wiener-Hopf design of optimal controllers—Part I: The single-input-output case," *IEEE Trans. Autom. Control*, vol. AC-21, no. 1, pp. 3–13, Feb. 1976.
[31] D. Youla, H. Jabr, and J. Bongiorno, "Modern Wiener-Hopf design of optimal controllers—Part II: The multivariable case," *IEEE Trans. Autom. Control*, vol. AC-21, no. 3, pp. 319–338, Jun. 1976.
[32] C. Desoer and R. Liu, "Global parametrization of feedback systems," in *Proc. IEEE Conf. Decision Control*, 1981, vol. 20, pp. 859–861.
[33] H. Hindi, B. Hassibi, and S. Boyd, "Multi-objective H_2/H_∞-optimal control via finite dimensional Q-parametrization and linear matrix inequalities," in *Proc. Amer. Control Conf.*, 1998, vol. 5, pp. 3244–3249.
[34] C. Gustafson and C. Desoer, "Controller design for linear multivariable feedback systems with stable plants, using optimization with inequality constraints," *Int. J. Control*, vol. 37, no. 5, pp. 881–907, 1983.
[35] V. Kucerǎ, *Discrete Linear Control: The Polynomial Equation Approach*. New York: Wiley, 1979.
[36] M. Vidyasagar, *Control System Synthesis: A Factorization Approach*. Cambridge, MA: MIT press, 1985.
[37] S. Boyd, C. Barratt, and S. Norman, "Linear controller design: Limits of performance via convex optimization," *Proc. IEEE*, vol. 78, no. 3, pp. 529–574, 1990.
[38] M. Dahleh and I. Diaz-Bobillo, *Control of Uncertain Systems: A Linear Programming Approach*. Englewood Cliffs, NJ: Prentice-Hall, 1995.
[39] N. Elia and M. Dahleh, *Computational Methods for Controller Design*. Berlin, Germany: Springer-Verlag, 1998.
[40] S. Salcudean, "Algorithms for Optimal Design of Feedback Compensators," Ph.D. Dissertation, Univ. California, Berkeley, 1986.
[41] E. Polak and S. Salcudean, "On the design of linear multivariable feedback systems via constrained nondifferentiable optimization in H_∞ spaces," *IEEE Trans. Autom. Control*, vol. AC-34, no. 3, pp. 268–276, Mar. 1989.
[42] A. Ben-Tal, S. Boyd, and A. Nemirovski, Control of Uncertainty-Affected Discrete Time Linear Systems Via Convex Programming Minerva Optimization Center, Technion, Haifa, Israel, Tech. Rep., 2005 [Online]. Available: http://www.stanford.edu/~boyd/papers/pur_out_control.html
[43] A. Ben-Tal, S. Boyd, and A. Nemirovski, "Extending scope of robust optimization: Comprehensive robust counterparts of uncertain problems," *Math. Programm.*, vol. B, no. 107, pp. 63–89, 2006.
[44] C. Desoer and R. Liu, "Global parametrization of feedback systems with nonlinear plants," *Syst. Control Lett.*, vol. 1, pp. 249–251, 1982.
[45] V. Anantharam and C. Desoer, "On the stabilization of nonlinear systems," *IEEE Trans. Autom. Control*, vol. AC-29, no. 6, pp. 569–572, Jun. 1984.
[46] C. Desoer and C. Lin, "Two-step compensation of nonlinear systems," *Syst. Control Lett.*, vol. 3, no. 1, pp. 41–45, 1983.
[47] C. Desoer and C. Lin, "Non-linear unity-feedback systems and Q-parametrization," *Int. J. Control*, vol. 40, no. 1, pp. 37–51, 1984.
[48] Y. Ermoliev and R. Wets, *Numerical Techniques for Stochastic Optimization*. Berlin, Germany: Springer-Verlag, 1988.
[49] H. Robbins and S. Monro, "A stochastic approximation method," *Annals Math. Stat.*, vol. 22, pp. 400–407, 1951.
[50] A. Shapiro and A. Ruszczyński, Lectures on Stochastic Programming 2008 [Online]. Available: http://www2.isye.gatech.edu/~ashapiro/publications/html
[51] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[52] Y. Wang and S. Boyd, "Fast model predictive control using online optimization," in *Proc. IFAC World Congress*, Jul. 2008, pp. 6974–6997.
[53] J. Skaf and S. Boyd, Design of Affine Controllers Via Convex Optimization 2008 [Online]. Available: http://www.stanford.edu/~boyd/papers/affine_contr.html

Modification of Mikhaylov Criterion for Neutral Time-Delay Systems

Tomáš Vyhlídal and Pavel Zítek

***Abstract*—The main goal of the technical note is to extend the Mikhaylov criterion to the case of neutral time delay systems. The modification consists in determining the vertex angle that bounds the argument oscillations of the Mikhaylov hodograph at high frequency ranges. Utilizing the strong stability concept, the presented stability criterion is examined from the viewpoint of potential fragility with respect to arbitrarily small delay changes. To facilitate the more demanding argument assessment, the Mikhaylov hodograph is converted to Poincaré-like mapping.**

***Index Terms*—Argument principle, linear time-delay system, Mikhaylov criterion, neutral system, Poincaré mapping, strong stability.**

I. Introduction

The class of neutral time delay systems (NTDS) is characterized by the presence of delays not only at the system state, but also at its deriva-

Manuscript received February 03, 2009; revised June 21, 2009. First published September 18, 2009; current version published October 07, 2009. This work was supported by the Ministry of Education of the Czech Republic under Project 1M0567. Recommended by Associate Editor Z. Wang.

The authors are with the Center for Applied Cybernetics and Department of Instrumentation and Control Engineering, Faculty of Mechanical Engineering, Czech Technical University in Prague, Prague 166 07, Czech Republic (e-mail: tomas.vyhlidal@fs.cvut.cz; pavel.zitek@fs.cvut.cz).

Digital Object Identifier 10.1109/TAC.2009.2029301