Efficient Robust Predictive Control

B. Kouvaritakis, J. A. Rossiter, and J. Schuurmans

***Abstract*—Predictive constrained control of time-varying and/or uncertain linear systems has been effected through the use of ellipsoidal invariant sets [6]. Linear matrix inequalities (LMI's) have been used to design a state-dependent state-feedback law that maintains the state vector inside invariant feasible sets. For the purposes of prediction however, at each time instant, the state feedback law is assumed constant. In addition, due to the large number of LMI's involved, online computation becomes intractable for anything other than small dimensional systems. Here we propose a new approach that deploys a fixed state-feedback law but introduces extra degrees of freedom through the use of perturbations on the fixed state-feedback law. The problem is so formulated that all demanding computations can be performed offline leaving only a simple optimization problem to be solved online. Over and above the very significant reduction in computational cost, the extra degrees of freedom allow for better performance and wider applicability.**

***Index Terms*—Constrained, predictive control, robust.**

I. Introduction

Predictive control [3] is a popular strategy and algorithms that handle constraints and have guaranteed stability have been given (e.g., [9], [8], and [13]). Computationally efficient extensions to the uncertain and/or nonlinear case are harder to develop. This paper, which considers linear uncertain and/or time-varying systems, addresses this issue by deploying the dual mode paradigm [5] according to which a fixed control law is used as soon as the system state enters a neighborhood of the origin. The fixed law can be selected [6] so that the current state belongs to an invariant set, thereby providing a guarantee of stability even for uncertain linear time varying (LTV) systems. This scheme allows the control law to vary from sample to sample; however, at each sample the predicted control trajectory is generated by a fixed state feedback law. This assumption imposes: i) limits on achievable performance; ii) restrictions on the size of the allowable set of initial conditions; and iii) significant online computational demands.

A convenient way to remove these limitations is to allow the first N control moves to become degrees of freedom that can be deployed to improve the transient predicted performance [13], [10]. The condition that the predicted control law becomes fixed and that the state vector enters the invariant set is delayed by N samples into the future [12]. The difficulty however is that for uncertain systems, predictions are not single-valued and computations can be conservative and still very demanding. Here we overcome these problems by using invariant sets that capture the system predictions, even during the transients when the control is not a fixed state-feedback law. It is shown that the appropriate invariant set can be significantly larger than that obtained with a fixed feedback law (e.g., [6]). In addition our formulation is such that the online computational burden is reduced dramatically thereby making practical implementation realistic even for high order systems.

Our development is based on the MPC strategy introduced in [7], [10], and [11] which optimizes closed-loop rather than open-loop predictions; predicted performance is not optimized over the system input $\mathbf{u}$, but rather a new free variable $\mathbf{c}$, which forms the input to a prestabilized loop. This scheme affords numerical advantages [10] but also enables a significant reduction in the online optimization. The idea here is as follows: in the absence of constraints let the prestabilized loop be robustly stable (over the uncertainty class) and optimal in some appropriate sense, say it optimizes worst case performance, or as per H_∞/H_2 design it minimizes a measure of performance subject to a suitable robustness constraint on the nominal sensitivity function. Then to retain optimality, the new free variable $\mathbf{c}$ should be zero whenever constraints are inactive. During transients however, the predicted behavior is nonlinear (due to constraint violations) and this implies that the prestabilized loop is no longer optimal and indeed may not even be robustly stabilizing. The variable $\mathbf{c}$ is the instrument through which this can be avoided by ensuring that the predicted trajectories reach but do not exceed the limits imposed by constraints. Feasibility, i.e., the existence of predicted $\mathbf{c}$ trajectories which avoid constraint violations over the whole uncertainty class, would ensure that the prestabilized dynamics remain linear and would be optimal had it not been for the nonzero values of $\mathbf{c}$. The obvious strategy therefore is to minimize the norm of $\mathbf{c}$ under the condition that constraints be satisfied over the whole uncertainty class. This formulation shifts the bulk of computational burden from online *optimization* to *feasibility* (which can be handled offline).

II. Earlier Work [6]

Consider a polytopic set of state space systems described as

$$\mathbf{x}_{k+1} = A(k)\mathbf{x}_k + B(k)\mathbf{u}_k, \qquad \mathbf{y}_k = C(k)\mathbf{x}_k,\ \mathbf{x} \in R^m,\ \mathbf{u} \in R^p \tag{1}$$

where $[A(k),\, B(k)] \in \Omega = Co\{[A_1,\, B_1], \cdots, [A_n,\, B_n]\}$ with $Co\{\cdot\}$ denoting the convex hull. The polytopic set can be taken to define an uncertain LTI (linear time invariant) or an uncertain LTV (linear time varying) system. Predicted performance at sample time k is measured by the cost

$$J = \sum_{i=0}^{\infty} \mathbf{x}_{k+i+1}^T Q \mathbf{x}_{k+i+1} + \mathbf{u}_{k+i}^T R \mathbf{u}_{k+i} \tag{2}$$

where $\mathbf{x}_{k+j},\ \mathbf{u}_{k+j}$ denote predicted values of the state and input vectors.

Next we summarize the ideas of invariance and feasibility as applied to ellipsoidal sets [6]

$$E_x = \left\{\mathbf{x} \,\middle|\, \mathbf{x}^T Q_x^{-1} \mathbf{x} \le 1\right\}, \qquad Q_x \ge 0. \tag{3}$$

Consider a state feedback control law which stabilizes all models in (1) for which

$$\mathbf{u}_k = K\mathbf{x}_k; \qquad \mathbf{x}_{k+1} = [A(k) + B(k)K]\mathbf{x}_k = \Phi(k)\mathbf{x}_k. \tag{4}$$

Then E_x is an invariant set for $\mathbf{x}_{k+1} = \Phi(k)\mathbf{x}_k$ if

$$\mathbf{x}_k \in E_x \Rightarrow \mathbf{x}_{k+1} \in E_x. \tag{5}$$

It is easy to show (e.g., [1]) that E_x is invariant for model set (1) if

$$Q_x^{-1} - \Phi_i^T Q_x^{-1} \Phi_i > 0, \qquad i = 1, \cdots, n; \quad \Phi_i = A_i + B_i K. \tag{6}$$

Assumption 1: The set of $[K,\, Q_x]$ satisfying (6) is nonempty.

For simplicity consider only constraints of the form

$$|(u_k)_j| \le d_j, \qquad j = 1, \cdots, p. \tag{7}$$

Manuscript received May 3, 1999; revised October 19, 1999. Recommended by Associate Editor, A. Tesi. This work was supported by EPSRC.

B. Kouvaritakis is with the Department of Engineering Science, Oxford University, Oxford OX1 3PJ U.K. (e-mail: basil.kouvaritakis@eng.ox.ac.uk).

J. A. Rossiter and J. Schuurmans are with the Department of Mathematical Sciences, Loughborough University, LE11 3TU, U.K. (e-mail: j.a.rossiter@lboro.ac.uk).

Publisher Item Identifier S 0018-9286(00)04214-8.

Under the state feedback law of (4) this constraint can be rewritten as

$$\left|\mathbf{K}_j^T \mathbf{x}\right| \le d_j, \qquad j = 1, \cdots, p \tag{8}$$

where $\mathbf{K}_j^T$ denotes the jth row of $\mathbf{K}$. For $\mathbf{x} \in E_x$ the above implies

$$\begin{aligned} \left|\mathbf{K}_i^T \mathbf{x}\right| &= \left|\mathbf{K}_i^T Q_x^{1/2} Q_x^{-1/2} \mathbf{x}\right| \\ &\le \left\|\mathbf{K}_i^T Q_x^{1/2}\right\| \left\|Q_x^{-1/2}\mathbf{x}\right\| \\ &\le \left\|\mathbf{K}_i^T Q_x^{1/2}\right\|. \end{aligned} \tag{9}$$

Therefore input constraints are satisfied if the following LMI's hold:

$$\left\|\mathbf{K}_i^T Q_x^{1/2}\right\| \le d_i \Leftrightarrow d_i^2 - \mathbf{K}_i^T Q_x \mathbf{K}_i \ge 0, \qquad i = 1, \cdots, p. \tag{10}$$

Remark 2.1: Given model set (1) and $K,\ Q_x$ such that both (6) and (10) are satisfied, define E_x as in (3). Then under the assumption that the same feedback controller K is used over all future prediction instants, $\mathbf{x}_k \in E_x$ implies that $\lim_{i\to\infty} \mathbf{x}_{k+i} = 0$.

The above forms the basis of a robust stabilizing predictive control strategy [6] which at each instant of time: i) varies K so as to ensure that $\mathbf{x}$ lies inside an invariant set; ii) uses the remaining design freedom to minimize an upper bound over the set Ω on the predicted performance cost of (2). The weakness of this approach is two-fold: i) the online computational burden is prohibitive as a new K has to be selected (involving many LMI's) at each time instant; and ii) as K is assumed fixed over the prediction horizon, only K is available to improve performance and alter the shape/size of the invariant set. Both these difficulties can be overcome through the use of prediction control trajectories which do not correspond to fixed state feedback control laws.

III. Free Future Control Moves and the Size of Invariant Sets

Enhance the closed-loop formulation of (4) with an additional term $\mathbf{c}_k$

$$\mathbf{u}_k = K\mathbf{x}_k + \mathbf{c}_k; \qquad \mathbf{c}_{k+n_c+i} = 0, \qquad i \ge 0 \tag{11}$$

with $\mathbf{c}_{k+i},\ i = 0, \cdots, n_c - 1$ representing degrees of design freedom, to get the closed-loop system

$$\mathbf{x}_{k+1} = \Phi(k)\mathbf{x}_k + B(k)\mathbf{c}_k. \tag{12}$$

Equivalently, the dynamics of (12) can be described by the autonomous state space model

$$\begin{aligned} &\mathbf{z}_{k+1} = \Psi(k)\mathbf{z}_k; \quad \mathbf{z} \in R^{m+pn_c} \\ &\Psi(k) = \begin{bmatrix} \Phi(k) & [B(k) \quad 0 \quad \cdots \quad 0] \\ 0 & M \end{bmatrix} \\ &\mathbf{z} = \begin{bmatrix} \mathbf{x} \\ \mathbf{f} \end{bmatrix}; \qquad \mathbf{f}_k = \begin{bmatrix} \mathbf{c}_k \\ \mathbf{c}_{k+1} \\ \vdots \\ \mathbf{c}_{k+n_c-1} \end{bmatrix} \\ &M = \begin{bmatrix} 0_{n_c} & I_{n_c} & 0_{n_c} & \cdots & 0_{n_c} \\ 0_{n_c} & 0_{n_c} & I_{n_c} & \cdots & 0_{n_c} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 0_{n_c} & \cdots & \cdots & 0_{n_c} & I_{n_c} \\ 0_{n_c} & 0_{n_c} & \cdots & \cdots & 0_{n_c} \end{bmatrix} \end{aligned} \tag{13}$$

where 0_{n_c} is an $p \times p$ matrix of zeros. Clearly the stability of the autonomous system above is guaranteed by the stability of $\Phi(k)$. Moreover, if there exists an invariant set E_x for (4), then there must exist at least one invariant set, E_z, for (13). To see this consider the definition of E_z

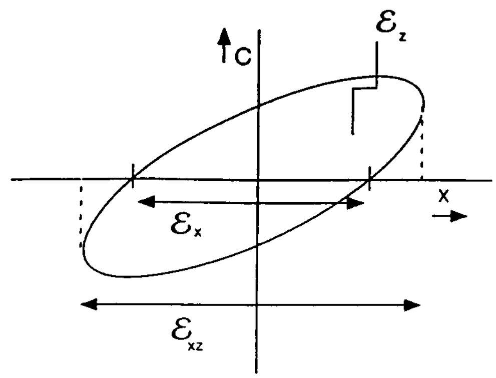

Fig. 1. Illustration of invariant sets $E_x,\ E_{xz},\ E_z$.

$$E_z = \left\{\mathbf{z} \left| \mathbf{z}^T Q_z^{-1} \mathbf{z} \le 1 \right.\right\} \tag{14}$$

and partition Q_z^{-1} in a manner conformal to the partition of $\mathbf{z}$ to get the blocks $\hat{Q}_{11},\ \hat{Q}_{12},\ \hat{Q}_{21},\ \hat{Q}_{22},\ \hat{Q}_{12}^T = \hat{Q}_{21}$. Then the inequality of (14) can be written as

$$\mathbf{x}^T \hat{Q}_{11} \mathbf{x} \le 1 - 2\mathbf{f}^T \hat{Q}_{21} \mathbf{x} - \mathbf{f}^T \hat{Q}_{22} \mathbf{f} \tag{15}$$

which is clearly satisfied by all $\mathbf{z} = [\mathbf{x}^T \quad 0^T]^T$, $\mathbf{x} \in E_x$ providing $\hat{Q}_{11} = Q_x^{-1}$. However nonzero $\mathbf{f}$ can be used to advantage to get invariant ellipsoids for (1) which are larger than E_x. Maximum benefit is obtained for $\mathbf{f} = -\hat{Q}_{22}^{-1}\hat{Q}_{21}\mathbf{x}$ [the maximizer of the RHS of (15)] for which (15) leads to

$$\begin{aligned} E_{xz} &= \left\{\mathbf{x} \left| \mathbf{x}^T Q_{xz}^{-1} \mathbf{x} \le 1 \right.\right\} \\ Q_{xz} &= \left[\hat{Q}_{11} - \hat{Q}_{12}\hat{Q}_{22}^{-1}\hat{Q}_{21}\right]^{-1}. \end{aligned} \tag{16}$$

Given that $Q_{xz}^{-1} \le \hat{Q}_{11}$ it is clear that $\hat{Q}_{11} = Q_x^{-1}$ implies $E_x \subseteq E_{xz}$. Note that a more convenient expression for Q_{xz} is $Q_{xz} = TQ_zT^T$ where T is defined by $\mathbf{x} = T\mathbf{z}$. A graphical illustration of the benefits of $\mathbf{f}$ is given in Fig. 1 for a simple example with $m = p = n_c = 1$ for which both $E_x,\ E_{xz}$ are straight line segments; Fig. 1 shows that the length of E_{xz} is about twice the length of E_x.

Over and above making E_{xz} as large as possible, it is still necessary to ensure its invariance and feasibility which by a simple extension of the earlier work to the model of (11) and (12) imply

$$\begin{aligned} &\Psi_i^T Q_z^{-1} \Psi_i - Q_z^{-1} \le 0 \\ \Leftrightarrow &\begin{bmatrix} Q_z & Q_z\Psi_i^T \\ \Psi_i Q_z & Q_z \end{bmatrix} \ge 0, \qquad i = 1, \cdots, n \end{aligned} \tag{17}$$

$$\begin{aligned} &\left\|[\,K_i^T \quad e_i^T\,]\ Q_z^{1/2}\right\| \le d_i \\ \Leftrightarrow\ &d_i^2 - [\,K_i^T \quad e_i^T\,]\, Q_z\, [\,K_i^T \quad e_i^T\,]^T \ge 0, \qquad i = 1, \cdots, p \end{aligned} \tag{18}$$

where e_i denotes the ith column of the identity matrix. Then, to maximize the volume of E_{xz}, defined by $\det(TQ_zT^T)$, Q_z can be computed from the following.

Algorithm 3.1: Minimize $\log\det(TQ_zT^T)^{-1}$ subject to LMI's (17) and (18).

This is a *Complex Problem* [1] which is tractable and can be solved efficiently in polynomial-time. Also LMI's (17) and (18) do not depend on the current state, so the algorithm can be applied offline.

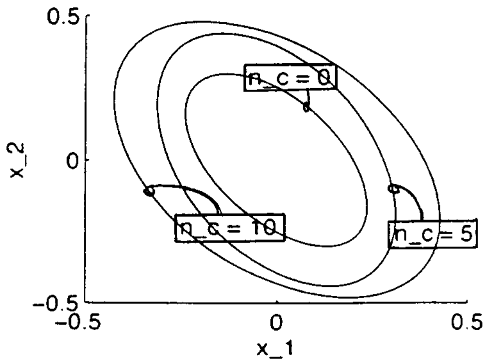

Fig. 2. Positive invariant sets for $n_c = 0$, 5, and 10.

A. Examples of Invariant Sets for Various n_c and K

MPC based on invariant feasible sets clearly cannot be applied unless the set of allowable initial conditions X_o is such that $X_o \subseteq E_x(K, Q_x)$ for some K, Q_x; the notation $E_x(K, Q_x)$ is used here to emphasize the dependence of E_x on both K, Q_x. As explained above, the introduction of the free variables, $\mathbf{c}_i$, $i = 0, \cdots, n_c - 1$, affords d.o.f. with which to widen applicability (to larger sets X_o) through the use of the larger invariant sets, E_{xz}. However it is also possible to adopt an alternative perspective: suppose that for K_b, a highly tuned controller, $X_o \cap E_x(K_b, Q_x) = \emptyset$ over all $Q_x > 0$, but that for K_a, a more detuned controller, $X_o \subseteq E_x(K_a, Q_{xa})$, $Q_{xa} > 0$. In such a case and in the absence of the perturbation signal $\mathbf{c}$ one must settle for the use of a suitably detuned controller K_a. However $\mathbf{f} \neq \mathbf{0}$ enables the use of a more highly tuned controller K_b with better predicted performance. These arguments apply at initial time only, but similar arguments can be used for all subsequent times. We illustrate these points by means of an example described by

$$\begin{aligned} A(k) &= Co\{A_1, A_2\}, \qquad A_1 = \begin{bmatrix} 1 & 0.1 \\ 0 & 1 \end{bmatrix} \\ A_2 &= \begin{bmatrix} 1 & 0.1 \\ 0 & 2 \end{bmatrix}, \qquad B(k) = \begin{bmatrix} 0 \\ 0.0787 \end{bmatrix} \\ C(k) &= [1 \;\; 0] \end{aligned} \tag{19}$$

with $|u(k)| \leq 2$. The stabilizing law $\mathbf{u} = K\mathbf{x}$, $K = [-8.48 \;\; -5.73]$ was computed using the algorithm of [6] for an initial state $\mathbf{x}_o = [0.38 \;\; 0]^T$. Fig. 2 shows how the size of E_x increases with n_c. Next let the nominal system description be $\mathbf{x}(k+1) = A_2\mathbf{x}(k) + B\mathbf{u}(k)$. For this model and for $Q = C^TC$, $R = 10^{-4}$, minimization of the quadratic performance criterion (2) yields the highly tuned state feedback controller $K_b = [-49 \;\; -16.5]$. On the other hand setting $R = 10^{-2}$ results in the more detuned controller $K_a = [-7.4 \;\; -7.7]$. For $n_c = 0$, the set E_x for K_a is almost identical (and hence is not plotted) to that of Fig. 2 (for $n_c = 0$), as is the set E_{xz} for K_b providing that $n_c = 10$. It is noted that the area of E_x for K_b very much smaller than that for K_a. However, use of $\mathbf{c}$ allows the definition of an invariant set E_{xz} for K_b which recovers the area of the invariant set E_x for the detuned controller K_a. The advantage then is clear: a more highly tuned control law can be used without a consequent reduction in the size of the invariant set.

IV. Predictive Robust Control Algorithm

The strategy of [6] is to constrain (online) K and Q_x, at each instant, so that $\mathbf{x} \in E_x$ and to use the remaining design freedom to minimize an upper bound (over Ω) on the predicted cost associated with the fixed control law $\mathbf{u} = K\mathbf{x}$. Hence the following implicit assumption is made.

Assumption 4.1 [6]: There exist K and Q_x, such that $\mathbf{x}_o \in E_x$.

The earlier discussion implies that this assumption can be relaxed by requiring assumption.

Assumption 4.2: There exist K, Q_z, and n_c such that $\mathbf{x}_o \in E_{xz}$.

Moreover, our development also implies that for n_c sufficiently large, feasibility and invariance can be handled by the d.o.f. in $\mathbf{f}$. In particular the role of $\mathbf{f}$ will be to: i) ensure $\mathbf{x} \in E_{xz}$; ii) ensure that the predicted trajectories of (11), (12) reach (if necessary) but do not exceed limits imposed by constraints; and iii) deploy any available degrees of freedom to optimize predicted performance. Task ii) provides the key to our development because it implies that during prediction the dynamics of the system (11), (12) will be linear, i.e., will not be affected by constraints. Thus, K can be chosen without regard to constraints, and can be designed using any of a plethora of robust control methods (e.g., H_∞, H_2/H_∞) to be optimal in whatever sense is appropriate for the particular application. Then, given the optimality of K, there is no reason to retune it at each subsequent instant. Indeed, if constraints were inactive, then the optimal policy would be to set $\mathbf{f} = 0$; thus $\mathbf{c}$ can be thought as an unwanted perturbation on the optimal control law $\mathbf{u} = K\mathbf{x}$ whose role is merely to ensure the feasibility of the predicted trajectories over Ω. Hence for optimally tuned K, the cost of (2) can be replaced by $J_f = \|\mathbf{f}\|_2^2$. The effect of this is to remove the excessive computational burden involved in the online tuning of K and replace it by a simple optimization subject to feasibility constraints. Invoking these constraints over Ω can still be demanding, however the autonomous system formulation of (13), combined with the use of invariant sets, implies that all time consuming computations (i.e., invoking constraints at each vortex of Ω) can be performed offline. All one has to do is take n_c to be large enough so that: i) Q_z can be chosen to satisfy (17) and (18) and ii) the resulting E_{xz} is large enough such that $X_o \subseteq E_{xz}$. This will guarantee an initial $\mathbf{f} = \mathbf{f}_o$ exists for which $\mathbf{x}_o \in E_{xz}$ so that by invariance $\mathbf{x}_1 \in E_{xz}$. Hence updating $\mathbf{f}$ as per (13), i.e., $\mathbf{f}_{k+1} = M\mathbf{f}_k$, all future $\mathbf{x}$ will also lie in E_{xz}. This update law will not necessarily yield the smallest allowable $\|\mathbf{f}\|_2^2$ and so performance can be further optimized through the minimization of J_f.

On the basis of these ideas we propose the following overall strategy. For most real life application $\mathbf{x}_o$ will belong to a known allowable set X_o. Thus it is possible to compute K, Q_z, and n_c offline so that Assumption 4.2 is satisfied. The way to proceed here is via any conventional (non-MPC) methodology for computing a robustly stabilizing K which is in some sense optimal (i.e., minimizes a worst case cost, or optimizes nominal performance). For example one could let K be the nominal LQ optimal controller (providing that the weight R in the LQ cost can be selected so that K provides the required robust stability margins). Alternatively one could use a state feedback realization of an H_2/H_∞ controller which optimizes an H_2 measure of nominal performance while respecting an H_∞ bound on the sensitivity function.

Algorithm 4.1:

(Offline Computation): Ignoring constraints, compute K so as to optimize robust performance, and for this K choose Q_z so as to maximize the volume of E_{xz}. If $X_o \in E_{xz}$ proceed to the online (receding horizon) application of the algorithm, otherwise increase n_c and repeat the offline design.

(Online Computation): At each sampling instant perform the minimization

$$\min_f \mathbf{f}^T\mathbf{f} \quad \text{s.t.} \quad \mathbf{z}^T Q_z^{-1} \mathbf{z} \leq 1. \tag{20}$$

Of the optimizing $\mathbf{f}$, implement the 1st element $\mathbf{c}_k$ in (11) and move onto the next sample.

Theorem 4.1: Under Assumption 4.2, the receding horizon application of Algorithm 4.1 is robustly stabilizing.

Proof: By assumption there exists K, Q_z, n_c such that $\mathbf{x}_o \in E_{xz}$ so that there exist $\mathbf{f}$ such that $\mathbf{x}_1 \in E_{xz}$. Let $\mathbf{f}_o$ be the solution of (20) at initial time. Then by invariance, at the next sampling time $\tilde{\mathbf{f}}_1 = M\mathbf{f}_o$ provides a feasible choice for $\mathbf{f}$ which results in a smaller cost. The decrease of cost is due to the definition of M according to which $\|M\mathbf{f}_o\|_2 \le \|\mathbf{f}_o\|_2$; of course $\tilde{\mathbf{f}}_1$ need not be the smallest feasible solution at the current time and so the cost may be reduced further still. These arguments can be applied recursively to establish that the minimization problem of (10) will be feasible at all future time (providing that it is feasible at initial time as guaranteed by Assumption 4.2) and that the cost $J_f = \|\mathbf{f}\|_2^2$ is monotonically decreasing. Thus Algorithm 4.1 generates a sequence of $\mathbf{c}_k$ which converges to zero and which ensures that constraints are met. Given constraint satisfaction, the dynamics of (11) and (12) describe those of a robustly stable closed-loop system driven by a reference signal $\mathbf{c}_k$ which converges to zero, which in turn implies that $\mathbf{x}$ itself will go to zero.

Remark 4.1: Given the stabilizing nature of K and the monotonic decrease of cost, it is apparent that $\mathbf{x}$ is progressively steered toward the origin, and that there will come a finite time when $\mathbf{u} = K\mathbf{x}$ becomes feasible. At such an instant the optimal solution for $\mathbf{f}$ will be zero. Thus the sequence of $\mathbf{c}_k$ generated by Algorithm 4.1 will go to zero in finite time.

A. Computational Advantages

The online optimization of (20) is clearly convex and in contrast to that of [6] involves just one LMI. Moreover, given the simple nature of the cost it is easy to show that (20) requires the solution of a univariate problem, namely that defined by the computation of the shortest distance of an ellipsoid from the origin. To see this rewrite (20)

$$\begin{aligned} &\min_f \mathbf{f}^T\mathbf{f} \quad \text{s.t.} \quad \mathbf{f}^T S_1 \mathbf{f} + \mathbf{f}^T S_2 \mathbf{x} + \mathbf{x}^T S_3 \mathbf{x} \le 1 \\ &S_1 = T_1 Q_z^{-1} T_1^T, \qquad S_2 = T_1 Q_z^{-1} T^T, \\ &S_3 = T Q_z^{-1} T^T, \qquad \mathbf{f} = T_1 \mathbf{z}. \end{aligned} \tag{21}$$

At any time instant $\mathbf{x}$ is assumed known and hence the inequality above defines an ellipsoid, say E_f, within which $\mathbf{f}$ must lie. To compute the optimal $\mathbf{f}$ note that: i) if E_f includes the origin, then the minimizer is zero; ii) otherwise the minimizing $\mathbf{f}$ will exist on the boundary of E_f and will be normal to the hyperplane that is tangent to boundary of E_f at $\mathbf{f}$

$$\mathbf{f}^T S_1 \mathbf{f} + \mathbf{f}^T S_2 \mathbf{x} + \mathbf{x}^T S_3 \mathbf{x} = 1, \quad \mu \mathbf{f} = (2S_1 \mathbf{f} + S_2 \mathbf{x}) \tag{22a,b}$$

where $\mu < 0$. Solving (22b) and substituting in (22a) gives a simple polynomial equation (of degree $2pn_c - 2$) which can only have two real solutions, one positive corresponding to the largest distance from the origin of the boundary of E_f and the other negative corresponding to the shortest distance. Hence the optimization problem of (20) is univariate and can be solved efficiently via any of a number of well known techniques. This in contrast to the optimization problem of [6] whose solution is several orders of magnitude more demanding and indeed becomes intractable for large n.

B. Simulation Example

The introduction of $\mathbf{f}$ not only allows for the dramatic reduction in online computation, but also provides extra d.o.f. that can be deployed to improve performance; such improvements clearly are not possible if constraints are inactive, because then $\mathbf{f} = \mathbf{0}$. However during transients, to cope with constraints [6] detunes K and settles for prediction costs that are achievable through that K. In contrast, Algorithm 4.1 uses the same fixed K throughout the receding horizon application but allows $\mathbf{f}$ to vary in order to meet constraints. For n_c sufficiently large, this allows for a much wider prediction class through which it is possible to improve closed-loop performance. That this is possible is illustrated here by means a simulation study based on the example of Section III. K was taken to be the LQ optimal controller for $R = 10^{-4}$, designed for the nominal model defined as $A = (A_1 + A_2)/2$, $B = B(k)$, where A_1, A_2, $B(k)$ are as given in (19). Fig. 3 shows the simulation results for the nominal (LTI) system and for an initial state $\mathbf{x}_o = [0.1 \quad 0]^T$. Fig. 4 shows simulations of the LTV system controlled by the same algorithms. At each time step of the simulation, the system state matrix was varied within Ω so that

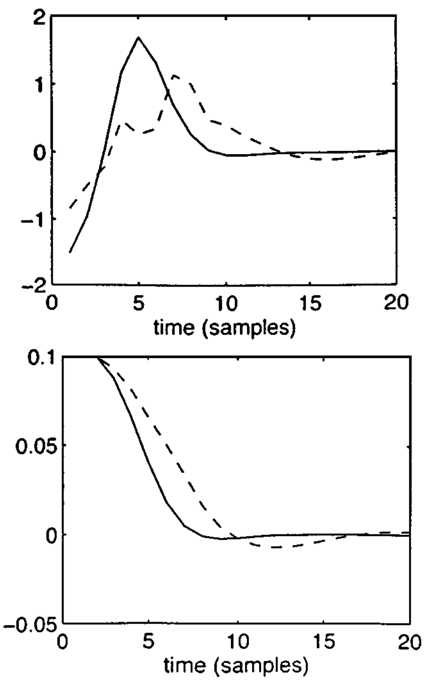

Fig. 3. Simulation with nominal system controlled by: i) algorithm of (21) (solid line) and ii) Kothare's algorithm (dashed line).

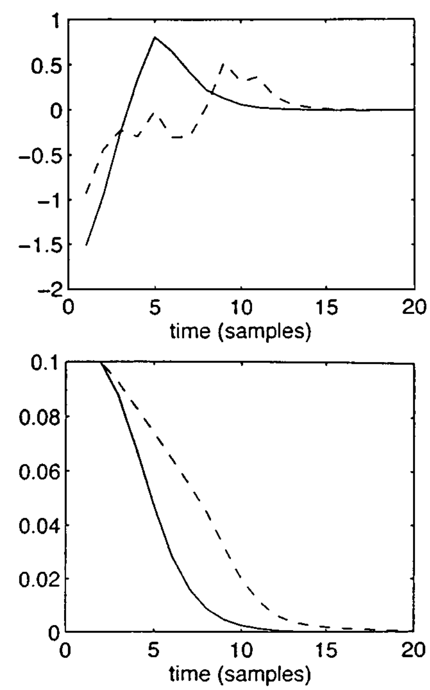

Fig. 4. Simulation with LTV system controlled by: i) algorithm of (21) (solid line) and ii) Kothare's algorithm (dashed line).

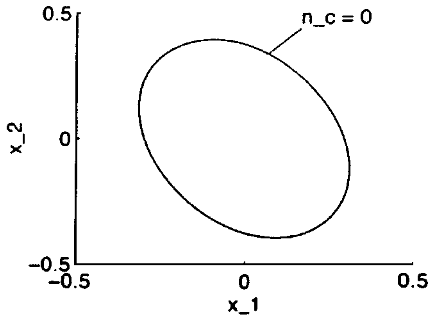

Fig. 5. Invariant sets with controller augmented by Youla parameter.

the performance index at that time step was maximized. It is clear that Algorithm 4.1 outperformed that of [6].

C. Further Robustness Improvements

The closed-loop system of (11) and (12) has input–output realization

$$U(z) = \frac{1}{M_o(z)} E(z), \qquad E(z) = C(z) - N_o(z)Y(z),$$
$$Y(z) = \frac{B(z)}{A(z)} U(z) \tag{22}$$

where the capital letters above are used to denote z-transforms. It is easy to show that replacing $M_o(z), N_o(z)$ by $M(z) = M_o(z) + B(z)Q(z), N(z) = N_o(z) - A(z)Q(z)$, where $Q(z)$ denotes a Youla parameter [14], does not affect the transfer function from $\mathbf{c}$ to $\mathbf{y}$ or from $\mathbf{c}$ to $\mathbf{u}$. Thus the insertion of $Q(z)$ does not affect the performance and feasibility properties of Algorithm 4.1 which now is applied to an augmented closed-loop state space model derived for (22) with $M(z), N(z)$ rather than $M_o(z), N_o(z)$. However, $Q(z)$ can have a significant effect on the size of the invariant set E_{xz}. Systematic design of $Q(z)$ for the purposes of maximizing the value of E_{xz} is possible but will not be discussed here. Instead we illustrate the point by means of a very simple example based on the model of Section III. The LQ controller K for $R = 0.1$ is designed on the matrices $A = A_1 B = B(k)$ of (19). For this controller $E_{xz} = \emptyset$ because there does not exist Q_z which satisfy (17) and (18). However introducing a first order Youla parameter, designed to minimize the H_2 norm of the sensitivity function of the closed-loop system based on the A, B matrices above, yields the invariant set of Fig. 5 which (even for $n_c = 0$) has comparable area to those shown in Fig. 2.

V. Conclusion

A MPC algorithm has been proposed and like that of [6] has been shown to be robustly stabilizing. The new algorithm however requires a modest amount of online computation and introduces extra d.o.f. to enlarge the volume of the relevant invariant set and improve closed-loop response.

Acknowledgment

The authors would like to thank M. Cannon for valuable comments.

References

[1] S. Boyd, L. El Ghaiou, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*. Philadelphia, PA: SIAM, 1966.
[2] D. W. Clarke, C. Mohtadi, and P. S. Tuffs, "Generalized predictive control—Parts 1 and 2," *Automatica*, vol. 23, pp. 137–160, 1987.
[3] C. E. Garcia, D. M. Prett, and M. Morari, "Model predictive control: Theory and practice, A survey," *Automatica*, vol. 25, pp. 335–348, 1987.
[4] E. G. Gilbert and K. T. Tan, "Linear systems with state and control constraints: The theory and practice of maximal admissible sets," *IEEE Trans. Automat. Contr.*, vol. 36, pp. 1008–1020, 1991.
[5] H. Michalska and D. Mayne, "Robust receding horizon control of constrained nonlinear systems," *IEEE Trans. Automat. Contr.*, vol. 38, pp. 1623–1633, 1993.
[6] M. V. Kothare, V. Balakrishnan, and M. Morari, "Robust constrained model predictive control using linear matrix inequalities," *Automatica*, vol. 32, pp. 1361–1379, 1996.
[7] B. Kouvaritakis, J. A. Rossiter, and A. O. T. Chang, "Stable generalized predictive control: An algorithm with guaranteed stability," *Proc. IEE Pt. D*, vol. 140, pp. 364–372, 1992.
[8] J. B. Rawlings and K. R. Muske, "The stability of constrained receding horizon control," *IEEE Trans. Automat. Contr.*, vol. 38, pp. 1412–1516, 1993.
[9] J. A. Rossiter and B. Kouvaritakis, "Constrained generalized predictive control," *Proc. IEE Pt. D*, vol. 140, pp. 233–254, 1993.
[10] J. A. Rossiter, B. Kouvaritakis, and M. J. Rice, "A numerically robust state-space approach to stable predictive control strategies," *Automatica*, vol. 34, pp. 65–73, 1998.
[11] J. A. Rossiter and B. Kouvaritakis, "Youla parameter and robust predictive control with constraint handling," in *Proc. Workshop Nonlinear Predictive Control*, Ascona, 1998.
[12] J. Schuurmans and J. A. Rossiter, "Robust piecewise linear control for polytopic systems with input constraints," in *Proc. IFAC 99*.
[13] P. O. M. Scokaert and J. B. Rawlings, "Infinite horizon linear quadratic control with constraints," in *Proc. 13th IFAC World Congr.*, vol. M, San Francisco, CA, 1996, pp. 109–114.
[14] D. C. Youla and J. J. Bongiorno, "A feedback theory of two-degree-of-freedom optimal Wiener–Hopf design," *IEEE Trans. Automat. Contr.*, vol. 30, pp. 52–664, 1985.