Technical Notes and Correspondence

An Interpolation Strategy for Discrete-Time Bilinear MPC Problems

H. H. J. Bloemen, M. Cannon, and B. Kouvaritakis

***Abstract*—Input–output (I–O) feedback linearization suffers from a number of restrictions which have limited its use in model-based predictive control. Some of these restrictions do not apply to the case of bilinear systems, but problems with input constraints and unstable zero dynamics persist. The present note overcomes these difficulties by means of an interpolation strategy. Involved in this interpolation is a feasible and stabilizing trajectory, which is computed through the use of invariant feasible sets, and a more aggressive trajectory, which can be chosen to be either the unconstrained optimal trajectory or any alternative one.**

***Index Terms*—Control Lyapunov function, discrete-time bilinear systems, interpolation.**

I. Introduction

Feedback linearization is a design technique that, through the use of an appropriate nonlinear state transformation and feedback law, yields an input–output map (I–O feedback linearization), or an input–state map (I–S feedback linearization) which is equivalent to that of a linear system, see, e.g., [1], [2]. I–O feedback linearization is less restrictive than I–S feedback linearization. I–O feedback linearization is a powerful tool that has been investigated within the context of model-based predictive control (MPC), e.g., [3]. It has not found wide acceptance because its use is restricted to the class of models which: i) are affine in the input; ii) have a definite relative degree; and iii) have stable zero dynamics [4]. Bilinear models meet the first two requirements, but of course these could have unstable zero dynamics. A further objection to the applicability of feedback linearization arises on account of the fact that iv) original linear input constraints become nonlinear state dependent constraints.

The aim of this note is to show that for bilinear systems, it is possible to use a linear interpolation strategy between two trajectories in order to handle constraints and to cope with unstable zero dynamics. One trajectory is designed as a "cautious" trajectory which is feasible and stabilizing for all initial conditions within a given set. The other trajectory can be the unconstrained optimal trajectory, i.e., the "desired" trajectory. Difficulties arising in connection with unstable zero dynamics can be overcome by the imposition of a convergence constraint. It is then possible to interpolate between the two trajectories so as to get as close to the unconstrained optimum as possible, while still ensuring nominal closed-loop stability (irrespective of the stability of the zero dynamics) and feasibility of the input constraints. Alternative "desired" trajectories can be incorporated as well. The online computations of the algorithm presented in this note consist of solving a one-dimensional linear program, which is less demanding than using the interpolation strategy proposed in [5]. The note starts with a discussion of the potential of I–O feedback linearization in Section II, and describes the interpolation strategy in Section III. The results of the note are illustrated by means of a numerical example in Section IV. Finally, the note is concluded with a discussion in Section V.

Manuscript received March 2, 2001; revised August 30, 2001 and December 20, 2001. Recommended by Associate Editor G. De Nicolao. This work was supported by the Dutch Technology Foundation (STW) and the Engineering and Physical Research Council, U.K.

H. H. J. Bloemen is with the Department of ITS, Delft University of Technology, 2600 GA Delft, The Netherlands (e-mail: H.Bloemen@its.tudelft.nl).

M. Cannon and B. Kouvaritakis are with the Department of Engineering Science, Oxford University, OX1 3PJ Oxford, U.K. (e-mail: Mark.Cannon@eng.ox.ac.uk; Basil.Kouvaritakis@eng.ox.ac.uk).

Publisher Item Identifier S 0018-9286(02)04765-7.

II. Input–Output Feedback Linearization

Given the stabilizable bilinear model

$$\begin{aligned} x(k+1) &= Ax(k) + G(x)u(k) \\ G(x) &= [B + [F_1x(k), \ldots, F_mx(k)]] \\ y(k) &= Cx(k) \end{aligned} \tag{1}$$

where $u \in \mathbb{R}^m$ is the input, $x \in \mathbb{R}^n$ is the state, $y \in \mathbb{R}^p$ is the output, and $A, B, F_1, \ldots, F_m, C$ are state-space matrices of conformal dimensions. Consider the input constraints

$$-\bar{u} \le u \le \bar{u} \tag{2}$$

where the aforementioned inequalities apply element wise. Then, the problem to be considered is to optimize a predicted performance criterion, such as (notation: $\|w\|_\Psi^2 = w^T\Psi w$)

$$J(k) = \sum_{i=0}^{\infty} \|y(k+i+1)\|_Q^2 \tag{3}$$

where Q is a positive–definite weighting matrix. Moreover, the goal is to ensure that

$$\lim_{k\to\infty} x(k) = 0. \tag{4}$$

For convenience, it is assumed that the only constraints on the problem are symmetrical input saturation constraints; it is noted that other more general input, output and state constraints can be accommodated just as easily. It is usual to include in the cost of (3) a term $\|u(k+i)\|_R^2$ that penalizes control effort; the rationale is that the positive definite weighting matrix R can be used to avoid highly tuned controllers that could result in violation of constraints with a consequent loss of performance. However, this is an ad hoc and suboptimal way of handling constraints which in this note will be taken care of explicitly; hence the absence of the control effort term in the cost of (3).

It is easy to show that everywhere outside the region $\Pi = \{x\colon \text{rank}(CG(x)) < p\}$ the system of (1) has relative degree 1 and is I–O feedback linearizable. In particular, if one can avoid Π, and if the zero dynamics of (1) are stable, it is easy to show that the unconstrained optimal control law with respect to the cost of (3) is given by the I–O feedback linearizing control law

$$u(k) = -(CG(x(k)))^{-1}CAx(k) \tag{5}$$

(if $p < m$ the inverse refers to the pseudoinverse). However, this control law may violate constraints (2) and so to get feasible input trajectories one must perturb (5) to get (in analogy to [6])

$$\begin{aligned} u(k+i) &= -(CG(x(k+i)))^{-1}[CAx(k+i) - c(k+i)] \\ i &= 0, \ldots, \nu \\ u(k+i) &= -(CG(x(k+i)))^{-1}CAx(k+i) \quad i > \nu \end{aligned} \tag{6}$$

where for the purpose of computational tractability, the sequence $c(k), \ldots, c(k+\nu)$ is of finite length. It is then easy to show that the cost of (3) is equivalent to $J(k) = \sum_{i=0}^{\nu} \|c(k+i)\|_Q^2$. Therefore, the obvious strategy is to make the perturbations as small as allowed by the requirement for feasibility.

Clearly $J(k)$ has to be minimized subject to constraint (2). It is possible to solve this problem in a computationally very efficient way (albeit at the cost of a small degree of suboptimality) by extending techniques deployed in the linear case [6]. This approach will not be explored here, however, because condition (4) must still be satisfied, which is difficult in the case where (1) has unstable zero dynamics. Instead in the sequel we consider an interpolating strategy which is also computationally efficient, and thus, may be the strategy of choice for a fast-sampling application where the online computational complexity has to be kept to a minimum.

III. Interpolating MPC

Let the control law $u(k) = K_{\text{des}}(x(k))$ denote a control law such as that of (5), which gives good (in the case of (5) optimal) performance with respect to $J(k)$. On the other hand, let

$$u(k) = K_{\text{st}}x(k) \tag{7}$$

be a linear state feedback law which is known to be stabilizing. Consider the interpolating control law

$$\begin{aligned} u(k) =& \lambda(k)K_{\text{st}}x(k) + (1-\lambda(k))K_{\text{des}}(x(k)), \\ 0 \leq& \lambda(k) \leq 1. \end{aligned} \tag{8}$$

With respect to (8), it is clear that, at each time instant, $\lambda(k)$ should be as small as possible with a view to improving the output performance while taking into account the following constraints: C1) $\lambda(k)$ is such that constraint (2) is satisfied, C2) $\lambda(k)$ satisfies a condition which ensures that in the closed loop the state converges to zero, and C3) at the next time instant a feasible λ exists, namely $\lambda(k)$ exists such that a $\lambda(k+1)$ exists that satisfies C1) and C2) at the next time instant. These issues are handled in this section. First, consideration is given to the design of (7).

A. Design of a Stabilizing State Feedback Law

Let V be a polytope in the state space, containing the origin. V can be represented by

$$V = Co\{v_1, \ldots, v_N\} \Leftrightarrow V = \{x : \Gamma x \leq \theta\}. \tag{9}$$

In the left representation, V is defined as the convex hull (Co) of N vertices $v_j \in \mathbb{R}^n$, $j = 1, \ldots, N$. In the right representation, V is defined as the region that satisfies the constraints $\Gamma x \leq \theta$. Both representations will be used in the note.

The following lemma states a sufficient condition which ensures that (8) satisfies condition C2) for $\lambda(k) = 1$.

Lemma 1: The control law of (7) locally asymptotically stabilizes the origin of (1) if there exists a positive–definite P such that for (small) $\epsilon > 0$

$$P - (A + G(v_j)K_{\text{st}})^T\, P\, (A + G(v_j)K_{\text{st}}) \geq \epsilon I \qquad \forall\, j = 1, \ldots, N. \tag{10}$$

Proof: Given the affine dependence of $G(x)$ on x, condition (10) ensures that everywhere inside V the control law of (7) gives $x(k+1)^T Px(k+1) < x(k)^T Px(k)\ \forall\, x(k) \neq 0$, i.e., locally $x^T Px$ is a strictly decreasing Lyapunov function for the origin. □

For $\lambda(k) = 1$, the interpolating control law of (8) will revert to that of (7) and, at such times, one may wish that K_{st} does more than just cause the state to converge to the origin. For example, given that the output (not the state) is the variable of primary interest, one may wish to emphasize the convergence of y to zero. This can be accomplished by replacing (10) by

$$\begin{aligned} P - \|A + G(v_j)K_{\text{st}}\|_P^2 \geq& \epsilon I + \|A + G(v_j)K_{\text{st}}\|_{C^TQC}^2 \\ \forall\, j =& 1, \ldots, N \end{aligned} \tag{11}$$

since under the above condition $x(k)^T Px(k)$ provides an upper bound on the cost of (3) [7]. Note that (11) implies (10) so that lemma 1 still holds.

Lemma 1 provides a local stability result for the origin of the system (1) in closed loop with (7). In order to arrive at an expression for the region of attraction, the state must remain within the polytope V. Moreover, condition C1) needs to be addressed. Sufficient conditions for these requirements are stated below.

Lemma 2: The ellipsoid $E = \{x : x^T Px \leq \eta\}$ is invariant under (1) in closed loop with (7) provided that P and η satisfy (10) or (11) and

$$\Gamma_r^T \frac{1}{\theta_r^2} \Gamma_r \leq \frac{P}{\eta} \quad \forall\, r = 1, \ldots, n_f \tag{12}$$

where the subscript r refers to the rth row, and n_f is the number of rows of Γ and θ. Moreover, feasibility with respect to (2) is guaranteed if the following constraints are satisfied:

$$K_r^T \frac{1}{\bar{u}_r^2} K_r \leq \frac{P}{\eta} \quad \forall\, r = 1, \ldots, m \tag{13}$$

where K_r denotes the rth row vector of K_{st}.

Proof: If x remains within the polytope V, then the value of the Lyapunov function can be bounded by: $x(k)^T Px(k) \leq \eta$. x is an element of V if $\Gamma x \leq \theta$ (9), which is satisfied if $x^T \Gamma_r^T \Gamma_r x \leq \theta_r^2 \Leftrightarrow x^T \Gamma_r^T (1/\theta_r^2) \Gamma_r x \leq 1\ \forall\, r = 1, \ldots, n_f$, which leads to (12) through the use of the ellipsoid $x^T Px \leq \eta$. Next, invariance of E is accomplished by Lemma 1. Feasibility with respect to (2) can be proven along the same lines as feasibility with respect to (9). □

Clearly, Lemma 2 restricts the applicability of K_{st} to initial conditions which belong to the ellipsoid E. Therefore, to increase applicability one may seek to choose P so as to maximize the volume of E. Via the variable transformations $P = \eta S^{-1}$, $K_{\text{st}} = YS^{-1}$ and through the use of Schur complements the inequalities (10)–(13) can be rewritten as linear matrix inequalities (LMIs), see [7] for details. Then, the volume of E is maximized by [8]

$$\min_{\eta, S, Y} -\log(\det(S)), \quad \text{subject to the appropriate LMIs} \tag{14}$$

Such an approach usually leads to a very conservative K_{st}, or a large upper bound on the cost in case of (11). A compromise between a large volume of E and a small η can be reached either by replacing the objective function in (14) by $\rho\eta - \log(\det(S))$, where ρ is a suitable positive user-defined scalar, or by fixing the value for η (usually to 1, e.g., [9]).

The solution to the optimization problem (14) yields the Lyapunov function $x^T Px$ for the bilinear model in closed loop with the feedback law $u = K_{\text{st}}x$, and with a guaranteed region of attraction given by the ellipsoid E, which is an inner approximation of the polytope V due to (12). Therefore, aiming at a large region of attraction requires a large volume of V. On the other hand, a large volume of V may cause inequalities (10) or (11) to be more stringent and thus may even rule out the existence of a feasible P and K_{st}. A systematic procedure to maximize the region of attraction is given by the following steps: a) Choose an initial polytope V, for example a hypercube in $\mathbb{R}^n$, that admits a solution to (14). Under the assumption that the linearized bilinear model at the origin is stabilizable, it is possible to find a feasible solution to (14) by choosing V small enough, since then $G(x)$ approaches

B. b) Based on the solution to (14), construct a new polytope V that fits tightly around (e.g., is tangential to) E in the directions of the principal axes of the ellipsoid. This can be done easily through a singular value decomposition. c) Increase the volume of the new polytope V (since E is an inner approximation of V this step is necessary in order to allow for a larger E), and re-solve (14). d) Repeat steps b) and c) until E no longer increases in size. Thus, the polytope V is a design variable. The Lyapunov function $x^T Px$ essentially is a control Lyapunov function since within E a feasible input is guaranteed to exist (i.e., $u = K_{\rm st} x$) that satisfies the decrease of $x^T Px$. Under the restriction of this convergence constraint the input may deviate from control law (7). This property is used in the interpolation strategy, see Section III-B.

Theorem 1: For $x(k) \in E \subseteq V$, the choice $\lambda(k) = 1, k = 0, 1, \ldots$ satisfies conditions C1)–C3) and hence guarantees asymptotic stability of the origin of the model of (1), under the control law of (8), subject to (2).

Proof: This is a direct result of Lemmas 1 and 2. □

B. Interpolation With a Desired Control Law

With respect to (3), an obvious choice for a "desired" control law is the I–O feedback linearizing law (5). For $\lambda = 0$, the control law of (8) in general does not necessarily satisfy C1)–C3), in which case there is no guarantee of closed-loop stability. This difficulty can be removed through the use of the following strategy.

Offline: For (small) $\epsilon > 0$ and using (10), (12), and (13) design $K_{\rm st}$, P, η as described in Section III-A.

Online:

1) Perform the following optimization:

$$\min_{\lambda} \lambda \text{ subject to:} \tag{15}$$

$$(2), (8), K_{\rm des} = (5) \tag{16}$$

$$\|x(k)\|_P^2 - \|Ax(k) + G(x(k))u(k)\|_P^2 \geq \|x(k)\|_{\epsilon I}^2. \tag{17}$$

2) For the optimal $\lambda(k)$, compute from (8) and implement the corresponding value of $u(k)$; at the next time instant, go to Step 1).

Constraint (17) is quadratic in the one-dimensional variable λ, and thus, under the implicit assumption that $x(k)$ is known, defines two explicit bounds for λ. On the other hand, the input constraints (2) in conjunction with the control law of (8) are linear in λ and thus define a further two explicit bounds on λ. Besides, λ is bounded between zero and one, see (8). Therefore, the optimization of (15) subject to (16) and (17) reduces trivially to selecting out of the six bounds the greatest lower bound.

Theorem 2: For $x(k) \in E \subseteq V$, the interpolation algorithm guarantees asymptotic closed-loop stability of the origin of the model (1), subject to constraints (2).

Proof: Under the assumptions of Lemma 1 and Lemma 2, $\lambda = 1$ provides a feasible solution which satisfies conditions C1)–C3), see Theorem 1. To complete the proof, note that λ is chosen so as to satisfy conditions (16) and (17) explicitly, which ensure that the state never leaves the ellipsoid E, that the signal constraints are satisfied, and that $x^T Px$ is strictly monotonically decreasing. □

The benefits of the "desired" control law in terms of output performance can only be derived for small λ. The case of $\lambda = 1$ is catered for by Theorem 1. For interim values of λ, however, the only handle on output performance is provided by (17), and as given this concerns x, not y, and, moreover, for ϵ small can only guarantee slow convergence. Indeed for $\epsilon = 0$, it would be possible that the interpolation algorithm fails altogether to steer the state and even the output to zero. An obvious remedy to this problem is the following: i) in the offline step of the algorithm, replace (10) by (11); ii) in the online step, replace (17) by

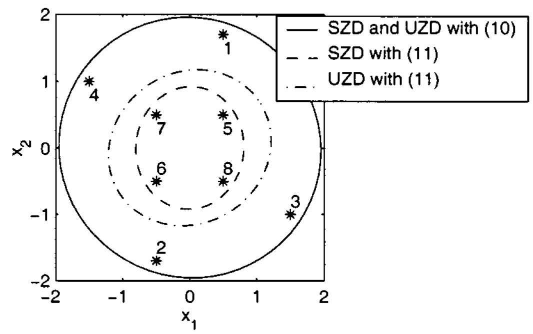

Fig. 1. Ellipsoids for model (20) with C_1 (SZD) and C_2 (UZD) with convergence constraints (10) and (11). The asterisks and numbers refer to the initial states of the simulations in Fig. 2.

$$\|x(k)\|_P^2 - \|Ax(k) + G(x(k))u(k)\|_P^2 \geq \|x(k)\|_{\epsilon I}^2 + \|Ax(k) + G(x(k))u(k)\|_{C^T QC}^2. \tag{18}$$

Clearly, Theorem 2 applies to this adapted algorithm as well.

An alternative for the "desired" controller could be the controller that maximizes the decrease of the Lyapunov function $x^T Px$, which leads to [10]

$$K_{\rm des}(x(k)) = -\left[G(x)^T PG(x)\right]^{-1} G(x)^T PAx(k). \tag{19}$$

Since the stability property in Theorem 2 does not depend on $K_{\rm des}$, the theorem still holds if the $K_{\rm des}(x(k))$ of (5) is replaced by that of (19). If the unforced dynamics are stable at the origin, then there exists a P such that the unconstrained control law (19) is globally stabilizing (excluding the points where $rank(G(x)) < m$) [10]. The interpolation algorithm provides a computationally cheap way to handle input constraints, and moreover, in the offline step it provides a way to design a P for which the unconstrained control law (19) is locally (within E) stabilizing even for bilinear models of which the unforced dynamics are unstable.

IV. Numerical Example

Consider the model of (1) for

$$A = \begin{bmatrix} 0.28 & -0.78 \\ -0.78 & -0.59 \end{bmatrix} \quad B = \begin{bmatrix} 0.71 \\ 1.62 \end{bmatrix}$$
$$F = \begin{bmatrix} 0.34 & 0.36 \\ 0.41 & -0.65 \end{bmatrix} \tag{20}$$
$$C_1 = [\,-0.69 \quad 0.86\,] \quad C_2 = [\,-0.69 \quad 0.20\,] \tag{21}$$

which has unstable unforced dynamics and which locally (around the origin) has stable zero dynamics (SZD) for C_1 and unstable zero dynamics (UZD) for C_2. The input is subject to the constraint given by (2) for $\bar{u} = 0.5$. The parameters $K_{\rm st}$ and P for both these models were optimized so as to maximize the area of the relevant invariant/feasible ellipsoid E, for fixed $\eta = 1$, as described in Section III-A. The resulting ellipsoids are shown in Fig. 1 for $\epsilon = 0.001$, $Q = I$. It is obvious from Fig. 1 that a less restrictive convergence constraint, i.e., (10) compared to (11), leads to a larger region of attraction of $K_{\rm st}$.

Simulations were performed for the initial states indicated in Fig. 1. The closed-loop costs J_s, i.e., (3) for the simulated values of y, normalized with the 2-norm of the initial state, are displayed in Fig. 2 for different choices of $K_{\rm des}$. For the model with stable zero dynamics (SZD) the performance of the controller based on interpolation with

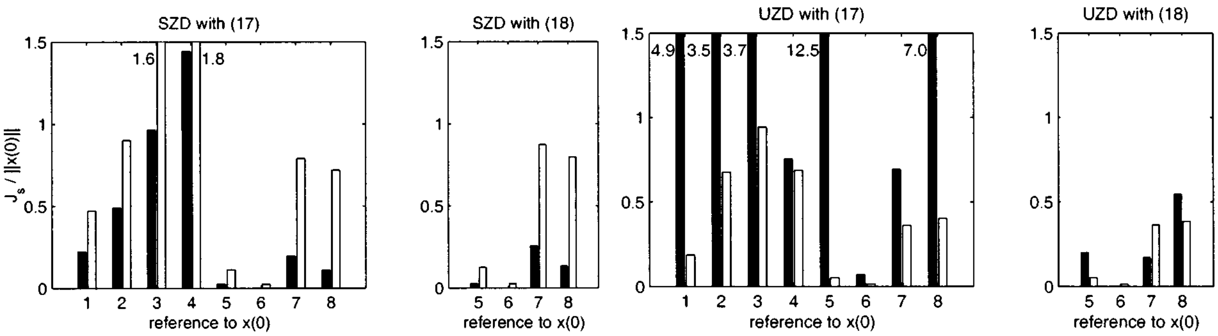

Fig. 2. The normalized closed-loop costs for the initial states indicated in Fig. 1. The black bars correspond to interpolation with (5), the white bars correspond to interpolation with (19). The values in the plots indicate the height of the bars that run off the scale.

the feedback linearizing controller (5) is superior to interpolation with (19). The performance is better due to the fact that the inverse dynamics impose no problem with respect to (4) for model SZD.

For the model with UZD the interpolation with the feedback linearizing controller (5) leads to a worse performance than the interpolation with (19). This is due to the fact that condition (4) is not satisfied by (5) for model UZD. Through the interpolation strategy a stable closed loop can still be obtained. If the convergence constraint is mild, the third plot in Fig. 2, the convergence is slow, which leads to a large (but finite) value of the simulation cost. If the convergence constraint is more stringent, the fourth plot in Fig. 2, the convergence is faster, leading to a reasonable simulation cost even for interpolation with (5). Although the zero dynamics of UZD are unstable, the feedback linearizing controller may still play a role in the transient response.

V. Discussion

Feedback linearization provides a convenient means for computing the optimal unconstrained predicted input trajectories with respect to the output of bilinear systems, but it does not necessarily meet constraints and does not guarantee that the state converges to zero or even that it remains bounded. The bilinear nature of the model considered in this note enables a simple definition of invariant/feasible sets which in turn can be used in the definition of "cautious" stabilizing control laws. Interpolation between these and the unconstrained optimal linearizing trajectories overcomes the problem of unstable inverse dynamics and the problems of feasibility. For models with unstable zero dynamics, close to the origin this algorithm tends to gravitate toward the "cautious" input trajectories. Due to this interpolation, closed-loop stability is guaranteed, and the problems connected to unstable zero dynamics are avoided.

The computational load of the interpolation strategy is very small since the online computations only involve a one-dimensional linear programming problem. The algorithm presented in this note can be interpreted as a control Lyapunov function (CLF) based approach, where a suitable CLF is calculated in the offline design of $K_{\rm st}$. In this respect the interpolation strategy is related to the pointwise min–norm controller, e.g., [11], in which the squared 2-norm of the input is minimized. This leads to a m-dimensional quadratic programming problem, and thus involves a larger computational load than the algorithm that is presented in this note. The interpolation algorithm presented in this note can be interpreted as a kind of pointwise min–norm controller. To see this, write the state prediction under the interpolating control law as

$$x(k+1) = \phi_{\rm des}(x(k)) + p(x(k)) \cdot \lambda \tag{22}$$

with $\phi_{\rm des}(x) = Ax + G(x)K_{\rm des}(x)$, $p(x) = G(x)K_{\rm st}x - G(x)K_{\rm des}(x)$. Thus the interpolation variable λ can be viewed as an auxiliary input variable, and this allows (15)–(17) to be recast as a pointwise min–norm optimization. However, the aim of the pointwise min–norm strategy in [11] is inverse optimality, which is achieved through the minimization of the norm of the actual input. This approach results in conservative responses since it disregards output performance. For example, in the case of stable unforced dynamics it is likely to lead to $u = 0$, thereby leaving the plant uncontrolled. On the other hand, interpolation provides the means of recharacterizing the degrees of freedom in the input through the definition of $K_{\rm des}$ and $K_{\rm st}$ in (22). Different poles of interpolation lead to different definitions of auxiliary inputs and thus allow for significant improvements in output performance, as demonstrated in Fig. 2 and discussed in Section IV.

Finally, note that for more general control-affine nonlinear models the interpolation strategy of this note can be used as well, in which case the online computational complexity is not affected. The most difficult part for more general control-affine nonlinear models is the design of $K_{\rm st}$. The affine nature of the nonlinearity in bilinear models, i.e., $G(x)$, provides an easy way to define a polytopic uncertain linear model to capture the dynamics of the bilinear model.

References

[1] E. Aranda-Bricair, U. Kotta, and C. H. Moog, "Linearization of discrete-time systems," *SIAM J. Control Optim.*, vol. 34, no. 6, pp. 1999–2023, 1996.

[2] H.-G. Lee and S. I. Marcus, "On input–output linearization of discrete-time nonlinear systems," *Syst. Control Lett.*, vol. 8, pp. 249–259, 1987.

[3] M. J. Kurtz and M. A. Henson, "Feedback linearizing control of discrete-time nonlinear systems with input constraints," *Int. J. Control*, vol. 70, no. 4, pp. 603–616, 1998.

[4] M. A. Henson and D. E. Seborg, "Critique of exact linearization strategies for process control," *J. Process Control*, vol. 1, pp. 122–139, 1991.

[5] B. Kouvaritakis, M. Cannon, and J. A. Rossiter, "Stability, feasibility, optimality and the number of degrees of freedom in constrained predictive control," in *Nonlinear Model Predictive Control*, F. Allgöwer and A. Zheng, Eds. Basel, Germany: Birkhäuser, 2000, pp. 99–113.

[6] B. Kouvaritakis, J. A. Rossiter, and J. Schuurmans, "Efficient robust predictive control," *IEEE Trans. Automat. Contr.*, vol. 45, pp. 1545–1549, Aug. 2000.

[7] M. V. Kothare, V. Balakrishnan, and M. Morari, "Robust constrained model predictive control using linear matrix inequalities," *Automatica*, vol. 32, pp. 1361–1379, 1996.

[8] S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*. Philadelphia, PA: SIAM, 1994.

[9] J.-W. Lee, W. H. Kwon, and J. Choi, "On stability of constrained receding horizon control with finite terminal weighting matrix," *Automatica*, vol. 34, pp. 1607–1612, 1998.

[10] Y. Nurges, "Synthesis of laws of stabilization of discrete bilinear systems," *Soviet J. Comp. Syst. Sci.*, vol. 23, pp. 101–106, 1985.

[11] J. A. Primbs, V. Nevistić, and J. C. Doyle, "A receding horizon generalization of pointwise min–norm controllers," *IEEE Trans. Automat. Contr.*, vol. 45, pp. 898–909, May 2000.