Available online at www.sciencedirect.com
SCIENCE DIRECT
Systems & Control Letters 55 (2006) 618–621
SYSTEMS & CONTROL LETTERS
www.elsevier.com/locate/sysconle

Comments on: "Efficient robust constrained model predictive control with a time varying terminal constraint set" by Wan and Kothare

Z. Wan [a], B. Pluymers [b], M.V. Kothare [c,*], B. de Moor [b]

[a] GE Water & Process Technologies, 4636 Somerton Road, Trevose, PA 19053, USA
[b] Katholieke Universiteit Leuven, Department of Electrical Engineering, Kasteelpark Arenberg 10, B-3001 Heverlee, Belgium
[c] Chemical Process Modeling and Control Research Center, Department of Chemical Engineering, Lehigh University, 111 Research Drive, Bethlehem, PA 18015, USA

Received 3 February 2006; accepted 21 February 2006
Available online 18 April 2006

Abstract

We present an algorithm that modifies the original formulation proposed in Wan and Kothare [Efficient robust constrained model predictive control with a time-varying terminal constraint set, Systems Control Lett. 48 (2003) 375–383]. The modified algorithm can be proved to be robustly stabilizing and preserves all the advantages of the original algorithm, thereby overcoming the limitation pointed out recently by Pluymers et al. [Min–max feedback MPC using a time-varying terminal constraint set and comments on "Efficient robust constrained model predictive control with a time-varying terminal constraint set", Systems Control Lett. 54 (2005) 1143–1148].

Keywords: Robust control; Constrained control; Model predictive control; Time varying system; Moving horizon

1. Introduction

The survey paper [1] on constrained finite horizon MPC reveals the presence of three 'ingredients'—a terminal cost $\mathscr{F}(\cdot)$, a terminal constraint set $\mathscr{X}_f$, and a local controller $\kappa_f(\cdot)$—that have been found useful in developing stabilizing MPC. In general, a stabilizing MPC steers the state into $\mathscr{X}_f$ over a finite horizon. Inside $\mathscr{X}_f$, a local stabilizing controller $\kappa_f(\cdot)$ is employed over the remaining infinite horizon, and the terminal cost is bounded by $\mathscr{F}(\cdot)$. A modified infinite horizon optimal control problem [1] is formulated to minimize the performance cost over the finite horizon plus the terminal cost $\mathscr{F}(\cdot)$. The decision variables are the control moves over the finite horizon.

For an N-step fixed control horizon, at time k, if the system is linear time invariant, there is only one realization of the system evolution over the N-step control horizon. The optimization solves a single sequence of N-step control moves. Only the first control move is implemented at time k. At time $k+1$, the remaining $N-1$ control moves solved at time k can move the system into $\mathscr{X}_f$, the Nth control move at time $k+1$ can be constructed as the first control action of the local controller $\kappa_f(\cdot)$.

If the system is linear time varying (LTV) within a polytope Ω with L vertices, the optimization at time k solves one move at time $k|k$, L moves for the L vertices of the polytope Ω at time $k+1|k$, and so on, L^{N-1} moves for the L^{N-1} vertices of the polytope Ω^{N-1} at time $k+N-1|k$. We then implement only the control move at time $k|k$. At time $k+1$, since the state $x(k+1)$ is a linear combination of the L predicted state $x(k+1|k)$, a feasible solution for the optimization at time $k+1$ can be constructed by linear combination of the control moves solved at time k (see [2] for details). For both linear time invariant and LTV systems, once a feasible solution is constructed from the optimization solved at a previous time, proving feasibility and asymptotic stability is straightforward [1,3,2].

For a linear varying system, the above formulation with a fixed control horizon will lead to high computational complexity. In this paper, we propose an alternative approach, which uses a variable control horizon and can significantly reduce the

* Corresponding author. Tel.: +1 610 758 6654; fax: +1 610 758 5057.
E-mail addresses: zhaoyang.wan@ge.com (Z. Wan), bert.pluymers@esat.kuleuven.be (B.Pluymers), mayuresh.kothare@lehigh.edu (M.V. Kothare), bart.demoor@esat.kuleuven.be (B. de Moor).

0167-6911/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.sysconle.2006.02.006

computational complexity. This approach is a modified version of the approach in [4]. The proposed algorithm can be proved to be robustly stabilizing and preserves all the advantages of the original algorithm of [4], thereby overcoming the limitation of the original algorithm pointed out recently by Pluymers et al. [2].

For the sake of brevity, we will refer to the contents of the original paper [4] with notation (.)*.

2. A modified robust constrained MPC with a time varying terminal constraint set

Consider a LTV system

$$x(k+1) = A(k)x(k) + B(k)u(k), \tag{1}$$

where $x(k) \in \Re^n$ is the state of the plant, $u(k) \in \Re^m$ is the control input subject to

$$|u_r(k)| \leqslant u_{r,\max}, \quad r = 1, 2, \ldots, m, \tag{2}$$

and $[A(k) \;\; B(k)] \in \Omega = \mathrm{Co}\{[A_1 \;\; B_1], \ldots, [A_L \;\; B_L]\}$ with Co denoting the convex hull. The nominal model $[\hat{A} \;\; \hat{B}] \in \Omega$ can be defined as the model that is most likely to be the actual plant. $\hat{x}(k+\mathrm{i}|k)$ is the state of the nominal model. $\mathscr{X}(k+N|k)$ is the uncertain terminal state set of the LTV system (1).

In Algorithm 1*, we construct a continuum of terminal sets $(\mathscr{X}_f(\theta), \kappa_f(\theta, \cdot), \mathscr{F}(\theta, \cdot))(0 \leqslant \theta \leqslant 1)$, which is a convex combination of the largest terminal set $(\mathscr{X}_f(1), \kappa_f(1, \cdot), \mathscr{F}(1, \cdot))$ and the smallest terminal set $(\mathscr{X}_f(0), \kappa_f(0, \cdot), \mathscr{F}(0, \cdot))$.

The main algorithm is summarized as follows.

Theorem 1. *Given a dynamical system* (1). *Off-line, construct a continuum of terminal constraint sets by using Algorithm* 1*. *On-line, given* $x(0|0)$ *at time* $k = 0$, $N > 0$.

Step 1: *If* $N > 0$, *minimize the nominal infinite horizon cost* $\hat{J}_\infty(k)$

$$\hat{J}_\infty(k) = \sum_{i=0}^{N-1} [\hat{x}(k+\mathrm{i}|k)^{\mathrm{T}} \mathscr{Q} \hat{x}(k+\mathrm{i}|k) + u(k+\mathrm{i}|k)^{\mathrm{T}} \mathscr{R} u(k+\mathrm{i}|k)] + \mathscr{F}(1, \hat{x}(k+N|k))$$

subject to (1), (2) *and the terminal constraint*

$$\mathscr{X}(k+N|k) \subset \mathscr{X}_f(1).$$

Apply the first control move $u(k|k)$. *Let* $k := k+1$, $N := N-1$. *If* $N > 0$, *go to Step* 1; *otherwise*, *let* $\theta = 1$ *and go to Step* 2.

Step 2: *If* $N = 0$, *i.e.*, $x(k|k) \in \mathscr{X}_f(1)$ *and* $\theta > 0$, *minimize* θ *to find the smallest terminal set such that* $x(k|k) \in \mathscr{X}_f(\theta)$. *Apply* $u(k|k) = \kappa_f(\theta, x(k|k))$. *Let* $k := k+1$. *If* $\theta > 0$, *go to Step* 2; *otherwise*, *go to Step* 3.

Step 3: *If* $\theta = 0$, *i.e.*, $x(k|k) \in \mathscr{X}_f(0)$, *apply* $u(k|k) = \kappa_f(0, x(k|k))$. *Let* $k := k+1$. *Go to Step* 3.

Suppose the algorithm is initially feasible, *then it robustly asymptotically stabilizes the system.*

Proof. Suppose the algorithm is initially feasible in Step 1. Consider a feasible *N*-step control sequence solved at time k, which steers $\mathscr{X}(k+N|k)$ into $\mathscr{X}_f(1)$ for all models in Ω.

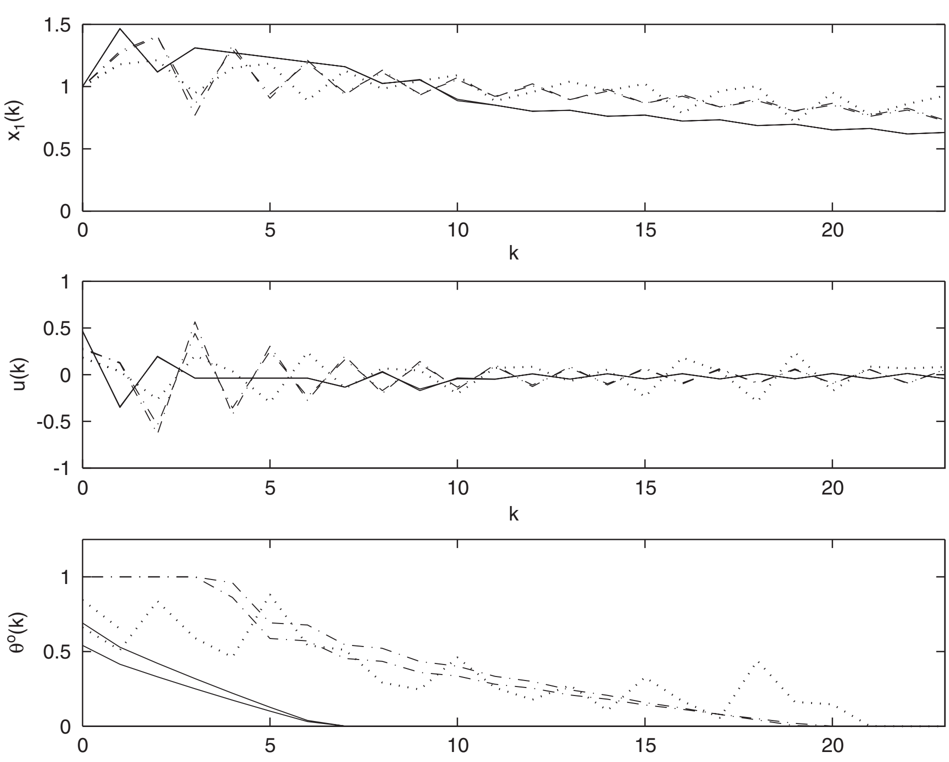

Fig. 1. Closed-loop response comparison on the example in [2] using the algorithm of [4] (dotted), the algorithm of [2] (solid) and the proposed algorithm in this paper (dash-dotted) for the cases $r_1 = 0.35$ and $r_1 = 0.34$.

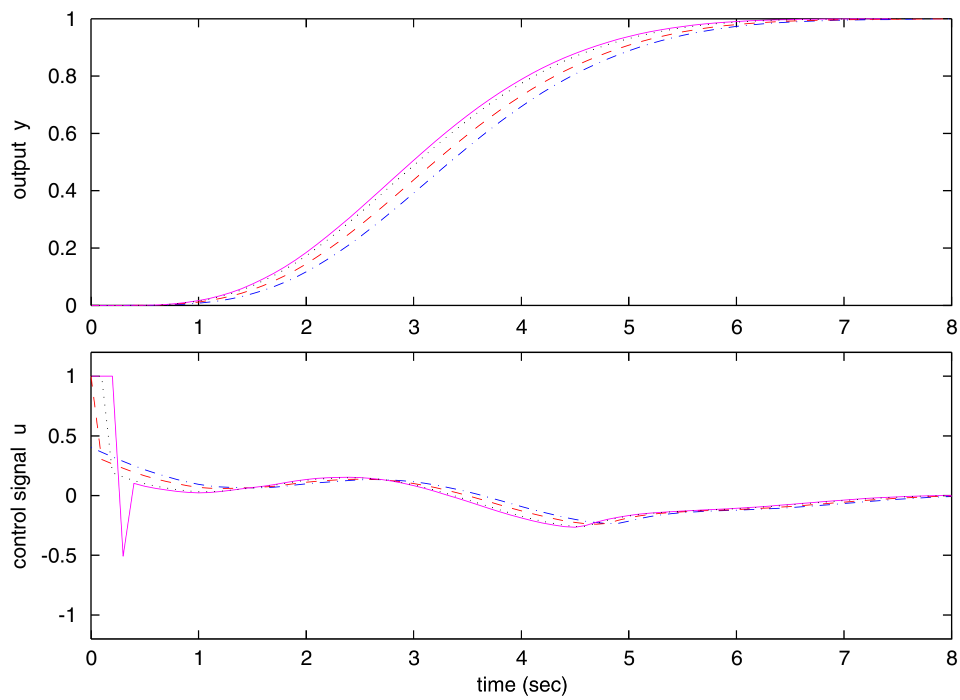

Fig. 2. Closed-loop responses. (Solid line with initial $N = 4$, dotted line $N = 2$, dashed line $N = 1$, dash-dot line $N = 0$.)

After $u(k)$ is implemented, the remaining $N - 1$ control moves provide a feasible solution for the optimization with a control horizon $N - 1$ at time $k + 1$. Using the above argument repeatedly, we can show that the optimization solved from k to $k + N - 1$ with the control horizons strictly decreasing from N to 1 are all feasible. After the last optimization with $N = 1$ is solved and the current control move implemented, the state is brought into $\mathscr{X}_f(1)$ and $N = 0$.

If at time k, the state enters $\mathscr{X}_f(1)$, we go to Step 2. Since $\mathscr{X}_f(\theta(k))$ is positively invariant for the closed-loop system with the control law $\kappa_f(\theta(k), \cdot)$, the control move $\kappa_f(\theta(k), \cdot)$ will move the state further into $\mathscr{X}_f(\theta(k))$. Therefore, there exists a feasible solution of $\theta(k+1)$ such that $\mathscr{X}_f(\theta(k+1)) \subset \mathscr{X}_f(\theta(k))$ which implies $\theta(k+1) < \theta(k)$. The optimal solution of $\theta(k+1)$ is smaller than or equal to the feasible solution of $\theta(k + 1)$. So, the minimization of θ guarantees the monotonic decrease of the optimal θ at each sampling time, and brings the state to the smallest terminal set $\mathscr{X}_f(0)$.

When the state enters $\mathscr{X}_f(0)$, we go to Step 3. Step 3 will be feasible and brings the state to the origin. $\square$

It is straightforward to demonstrate asymptotic stability of the proposed algorithm for the counterexample in [2]. This is also verified in Fig. 1 which shows the closed-loop response for the counterexample system in [2] using the original algorithm of Wan and Kothare [4] (dotted), the modification proposed by Pluymers et al. [2] (solid) and the proposed algorithm in this paper (dash-dotted). Simulations for both $r_1 = 0.35$ and $r_1 = 0.34$ are shown. The corresponding computation times are, respectively, 0.3, 0.58 and 0.16 s for $r_1 = 0.34$, and, respectively, 0.28, 0.56 and 0.14 s for $r_1 = 0.35$, thereby clearly showing the computational advantage of our proposed algorithm. These calculations were done on a Pentium M 1.7 GHz computer, using matlab 6.5 and LMILab 1.0.8.

We also apply the improved algorithm to the original two-mass-spring example from [4]. Fig. 2 shows the closed-loop responses with different initial N's. All the simulations were performed on a Gateway PC with a Pentium III processor (speed 1000 MHz, Cache RAM 256 kB and total memory 256 MB) and using the software LMI Control Toolbox in the MATLAB environment to compute the solution of the linear minimization problem. The following table shows the on-line computational demands for the different optimization problems solved in Theorem 1.

Optimization problem	Time to compute (s)
Step 1 with $N = 4$	0.42
Step 1 with $N = 3$	0.16
Step 1 with $N = 2$	0.08
Step 1 with $N = 1$	0.04
Step 2 and 3	0.02

Since the major contributor to computational complexity is uncertainty propagation over the control horizon and enforcement of terminal constraints over the uncertain set of terminal states, shortening of the control horizon with an enlarged terminal region drastically reduces on-line computation.

Acknowledgments

Bert Pluymers is a research assistant with the IWT Flanders. His research supported by research council KUL: GOA

AMBioRICS, CoE EF/05/006 Optimization in Engineering, Flemish Government: POD Science: I UAP P5/22. The Ph.D. thesis of Z. Wan under the supervision of M. V. Kothare was partially supported by the American Chemical Society's Petroleum Research Fund and by the R. L. McCann Professorship endowment at Lehigh University.

References

[1] D.Q. Mayne, J.B. Rawlings, C.V. Rao, P.O.M. Scokaert, Constrained model predictive control: stability and optimality, Automatica 36 (6) (2000) 789–814.

[2] B. Pluymers, J.A.K. Suykens, B. de Moor, Min–max feedback MPC using a time-varying terminal constraint set and comments on "Efficient robust constrained model predictive control with a time varying terminal constraint set", Systems Control Lett. 54 (2005) 1143–1148.

[3] P.O.M. Scokaert, D.Q. Mayne, Min–max feedback model predictive control for constrained linear systems, IEEE Trans. Automat. Control 43 (1998) 1136–1142.

[4] Z. Wan, M.V. Kothare, Efficient robust constrained model predictive control with a time varying terminal constraint set, Systems Control Lett. 48 (5) (2003) 375–383.