ELSEVIER

IFAC

automatica

A Journal of IFAC the International Federation of Automatic Control

Volume 43 Number 11 ISSN 0005-1098 November 2007

CONTENTS

Continued on outside back cover

Available online at www.sciencedirect.com
ScienceDirect
automatica
Automatica 43 (2007) 1945–1953
www.elsevier.com/locate/automatica

Brief paper

A stable block model predictive control with variable implementation horizon☆

Jing Sun[a,*], Ilya V. Kolmanovsky[b], Reza Ghaemi[a], Shuhao Chen[a]

[a]Department of Naval Architecture and Marine Engineering, University of Michigan 2600 Draper Road, Ann Arbor, MI 48109, USA
[b]Powertrain Control Research and Advanced Engineering, Ford Motor Company 2101 Village Road, Dearborn, MI 48121, USA

Received 9 January 2006; received in revised form 10 December 2006; accepted 14 March 2007
Available online 22 August 2007

Abstract

In this paper, we present a stable model predictive control method for discrete-time nonlinear systems. The standard MPC scheme is modified to incorporate (1) a block implementation scheme where a sub-string of the optimized input sequence is applied instead of a single value; (2) an additional constraint which guarantees that a Lyapunov function will decrease over time; (3) a variable implementation window that facilitates the stability constraint enforcement. Stability of the closed-loop system with the proposed algorithm is established. Examples are given to illustrate the effectiveness of the control scheme. The impacts of several key design parameters on the overall performance are also analyzed and discussed.

Keywords: Model predictive control; Stability; Constraints

1. Introduction

Model predictive control (MPC) (Maciejowski, 2002), despite the computational intensity associated with its on-line implementation, has found many successful industrial applications (Qin & Badgwell, 1997). Its intuitively appealing and flexible formulation, together with its capability for dealing with constraints, nonlinearities, and hybrid systems has been a major advantage. The main challenges of MPC, which include how to ease the computational requirements and how to guarantee stability, have also attracted attention of many researchers and control practitioners (Bemporad & Morari, 1999, Chen & Allgower, 1998, Lee, Kouvaritakis, & Cannon, 2002, Mayne, Rawlings, Rao, & Scokaert, 2000).

In the standard MPC implementation, a finite horizon open-loop optimization problem is solved at each sampling instant, using the current state as the initial condition. The optimization results in a control sequence, the first element of which is selected and then applied as the control input to the plant. In repeating the process, the state used in the optimization is re-initialized at each sampling instant, thereby providing a feedback mechanism for disturbance rejection and reference tracking. The designer can choose different cost functions and receding horizon length in the optimization problem formulation in order to meet different design objectives. State and input constraints, whether they are pointwise-in-time or accumulative, can also be accommodated with an added computation burden.

The two major challenges associated with MPC schemes are the computational intensity and stability. For systems with nonlinear constraints, the numerical difficulties in solving the optimization problem often represent road blocks to the real-time implementation of MPC. However, advances in computing technology and efficiency improvements in optimization algorithms are easing the computational burden, and the real-time implementation of MPC schemes is becoming more affordable

☆ This paper was not presented at any IFAC meeting. This paper was recommended for publication in revised form by Associate Editor Zhihua Qu under the direction of Editor Hassan Khalil.

* Corresponding author. Tel.: +1 734 615 8061.
E-mail addresses: jingsun@umich.edu (J. Sun), ikolmano@ford.com (Ilya V. Kolmanovsky), ghaemi@umich.edu (R. Ghaemi), shchen@umich.edu (S. Chen).

0005-1098/$ - see front matter
doi:10.1016/j.automatica.2007.03.026

as the cost of computer hardware decreases. The issue of stability, on the other hand, has been recognized as a more fundamental problem. Algorithms and mechanisms that assure stability for MPC schemes have been actively pursued by the control engineering community (Mayne et al., 2000). Thus far, the key mechanisms used to guarantee stability for MPC fall into two main categories: one is to extend the prediction horizon, and another is to incorporate a proper penalty and/or to impose certain constraints on the final state at the end of the prediction horizon. Other strategies have also been proposed, such as the dual mode control, which uses the MPC to steer the trajectory into a terminal set inside which the control is switched to a local stabilizing controller (Michalska & Mayne, 1993).

In this paper, our attention is mainly focused on the stability of MPC schemes. Our research was motivated by the all-electric ship reconfiguration control problem, where the system has to be moved from one (damaged) state to another (safe operation) with limited available energy resource and within a given time constraint. The unpredictable and adversarial operational scenarios that the naval ships have to face in their reconfiguration stage often render the open-loop based optimal trajectory planning strategy insufficient and inflexible. The MPC, on the other hand, has the capability to incorporate the state feedback to reject disturbances and to ensure performance. Its ability to deal with nonlinear constraints also makes it an ideal candidate for the reconfiguration problem. Given the survival critical nature of the naval ship reconfiguration problem, the stability of the control system is an overriding requirement and cannot be compromised.

It is with the motivation to develop efficient and safe naval ship reconfiguration algorithms that we investigate the performance and stability of a novel MPC scheme, which we refer to as block MPC (Sun, Chen, & Kolmanovsky, 2005). In the proposed block MPC scheme, a string or a subsequence of the control from each optimization run is applied to the system instead of only the first element. An additional contractive constraint, similar to that proposed in (de Oliveira Kothare & Morari, 2000), is also enforced to guarantee that a Lyapunov-like function is decreasing over time. The difference with (de Oliveira Kothare & Morari, 2000) is that by allowing the size of the string, referred to as the implementation window, to vary in our approach, the enforceability of the constraints is greatly enhanced. Stability of the closed-loop system with the proposed MPC scheme is rigorously established in this paper, and numerical simulation examples are provided to illustrate the effectiveness of the proposed scheme. Another approach to contractive NMPC has been proposed in a recent paper (Alamir, 2006), which includes the horizon as an optimization variable and, due to reformulation of the cost function, avoids explicit use of a contraction stability constraint.

2. Block MPC scheme

Consider a class of nonlinear discrete-time systems described by the following equation:

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

where x is the state vector and u the input vector. Without loss of generality, we also assume that $x=0, u=0$ is the equilibrium of the system (1), and consequently $f(0,0)=0$. Suppose that the standard MPC scheme is designed by solving the following optimization problem at each time instant k:

$$\min J = \min_{\{u(\cdot)\}} \left\{ \sum_{i=k}^{k+N_r-1} L(x(i|k), u(i)) + K(x(k+N_r|k)) \right\}$$
$$\text{subject to the dynamic equation (1)}$$
$$\text{and constraints } x(i|k) \in \mathscr{S}, u(i) \in \mathscr{U}, \tag{2}$$

where $x(i|k)$ is the predicted state at i-step ahead using measured (or estimated) state $x(k)$ and input $\underline{\mathbf{u}} = \{u(k), \ldots, u(k+i-1)\}$.[1] $\mathscr{S}$ and $\mathscr{U}$ are admissible sets for the state and input, respectively, and $L(x,u)$ and $K(x)$ are nonnegative functions of (x,u) and x, respectively. In (2), N_r is the prediction horizon over which the performance of the control system is evaluated and optimized, and the term $K(x)$ reflects the penalty on the terminal state. For the remainder of the presentation, we use $z_{[k_1,k_2]}$ to denote the string of variable z in the interval $[k_1,k_2]$, namely, $z_{[k_1,k_2]} := \{z(k_1), z(k_1+1), \ldots, z(k_2)\}$.

To assure a meaningful MPC formulation, we make the following assumptions on the functions L and K:

- (A1) There exists a subset $\mathscr{B} \subset \mathscr{S}$ such that $L(x,u) \geqslant L(x,0)$ for any $x \in \mathscr{B}, u \in \mathscr{U}$.
- (A2) $x=0 \in int\,\mathscr{B}$ and $u=0 \in int\,\mathscr{U}$, where int denotes the interior of a set.
- (A3) There exist monotonically increasing functions l, s, with $l(0)=s(0)=0$, such that $L(x,0) \leqslant l(\|x\|)$, $K(x) \leqslant s(\|x\|)$ for any $x \in \mathscr{B}$.
- (A4) For any $x \in \mathscr{B}$, there exists a function $q(r)$ such that $L(x,0) \geqslant q(\|x\|)$, where $q(\cdot)$ is monotonic in the interval $[0,r']$ with $q(0)=0$ for some $r'>0$.

Assumptions (A1)–(A4) are unrestrictive in general and can be satisfied by most of usual MPC formulations.

Remark 2.1. For linear systems, only assumption (A2) is needed to establish the subsequent results.

As in the standard MPC case, the string $u_{[k,k+N_r-1]}$ is varied within the input admissible set to minimize the cost function J for (2). In contrast to the standard MPC which applies the first element in the string of $\{u(k), u(k+1), \ldots, u(k+N_r-1)\}$ to the control signal, we propose the following block MPC scheme:

Definition 2.1 (*Block MPC*). Let $u^*_{[k,k+N_r-1]}$ be the optimal sequence for (2) and N_c be a fixed integer satisfying $1 \leqslant N_c \leqslant N_r$. If

- (a) the first N_c elements, i.e., the sub-string $u^*_{[k,k+N_c-1]}$ of $u^*_{[k,k+N_r-1]}$ are applied at time instants $t=k, k+1, \ldots, k+N_c-1$ and

[1] The dependency of $x(i|k)$ on $\underline{\mathbf{u}}$ is omitted to simplify the notations.

- (b) the optimization for (2) is repeated after N_c steps at $t = k + N_c$ with the new state $x(k + N_c)$,

then we refer to N_c as the control implementation window and the resulting control scheme as the N_c-block MPC algorithm.

The block MPC scheme specified in Definition 2.1 has similar properties to that of the regular MPC algorithm. In fact, the N_c-block MPC can be viewed as a hybrid control scheme where the control update and optimization are carried out over two different sampling intervals: T and N_cT. Like many other multi-rate sampling control systems, it has the advantage of requiring less computing resource. The saving in the computational effort, however, may be achieved at the cost of reduced disturbance rejection capability. Within each control implementation window, the block MPC essentially behaves as an open-loop control system, and as such its inter-sampling behavior is subject to interference from disturbances.

Remark 2.2. The proposed block MPC solves the optimization problem every N_c steps, requiring less computing effort as compared to solving it every step. It also allows more time to perform each optimization task, if the scheme proposed in (Milam, Franz, Hauser, & Murray, 2005) is adopted to precompute the optimization solution using the predicted initial state instead of the measured state. In that case, one can use up to N_cT seconds (T being the sampling interval) instead of T seconds to complete the optimization, thereby making it feasible to solve more complex optimization problems with additional constraints, such as the one proposed later in this paper.

Remark 2.3. The notion of the control implementation window N_c used here is different from the control window defined in Alamir and Bornard (1995). In Alamir and Bornard (1995), the control window N refers to the dimension of the optimization problem, when only the first N elements of the input sequence $u_{[k,k+N_r-1]}$ are allowed to vary in minimizing J.

Remark 2.4. The concept of "block implementation" proposed here is quite different from the notion of "blocking of manipulated variable" discussed in (Ricker, 1985). In the latter, blocking is applied to the optimization problem formulation, where the control variables are grouped into blocks, and within each block the variables are held constant. In our case, blocking refers to implementing a string of optimized control variables within a window instead of a single value at one step.

In the sequel, we explore the additional design flexibility offered by our block MPC scheme to achieve the desired stability properties:

Theorem 2.1. *Let $V(x)$ be a positive definite function of x. If* (A1)–(A4) *are satisfied and the N_c-block MPC scheme as defined in Definition* 2.1 *is designed by solving* (2) *with the following additional constraint*[2]:

$$V(x(k+N_c|k)) \leqslant \gamma V(x(k)) \tag{3}$$

for some $0<\gamma<1$ and for $k=0, N_c, 2N_c, \ldots$, then the closed-loop system with the N_c-block MPC is asymptotically stable.

To prove Theorem 2.1, we first establish the following property for the state trajectory of systems with MPC control:

Lemma 2.2. *If* (A1)–(A4) *are satisfied and the optimal sequence $u^*_{[k,k+N_r-1]}$ of* (2) *is applied to* (1), *then there exist a $r>0$ and a continuous function $g(\cdot)\geqslant 0$ with $g(0)=0$ such that if $\|x(k)\|\leqslant r$, we have*

$$\|x(i|k)\| \leqslant g(\|x(k)\|), \quad \forall i \in \{k+1, \ldots, k+N_r\}. \tag{4}$$

Proof of Lemma 2.2. Let $x^0_{[k,k+N_r]|k}, x_{[k,k+N_r]|k}$ be the state trajectories corresponding to inputs $u^0_{[k,k+N_r-1]} = \{0, \ldots, 0\}$ and $u^*_{[k,k+N_r-1]}$, respectively. Note that

$$x^0(i|k) = f(f(\ldots f(x(k),0)\ldots),0) := f_i(x(k))$$

for $i=k, \ldots, k+N_r$, we have

$$\|x^0(i|k)\| \leqslant \bar{g}(x(k)), i=k, \ldots, k+N_r, \tag{5}$$

where $\bar{g}(x) = max_{k\leqslant i\leqslant k+N_r}\{\|f_i(x)\|\}$. Given that $x=0$ is an equilibrium of the system with $u=0$ and $f(0,0)=0$, we have $f_i(0)=0, i=1, \ldots, N_r$ and therefore $\bar{g}(0)=0$.

Using conditions (A1), (A2) and (A3), we have

$$\begin{aligned} L(x(i|k),0) &\leqslant L(x(i|k), u^*(k+i)) \\ &\leqslant \sum_{j=k}^{k+N_r-1} L(x^0(j|k),0) + K(x^0(k+N_r|k)) \\ &\leqslant \sum_{j=k}^{k+N_r-1} l(\|x^0(j|k)\|) + s(\|x^0(k+N_r|k)\|) \\ &\leqslant N_r l(\bar{g}(\|x(k)\|)) + s(\bar{g}(\|x(k)\|)). \end{aligned} \tag{6}$$

Here the second inequality is due to $u^*_{[k,k+N_r]}$ being the optimal solution of (2) and $x_{[k,k+N_r]|k}$ being the corresponding trajectory, and the last inequality is obtained using (5). The remainder of the proof follows directly from (6) and the assumption (A4). □

Note that Lemma 2.2 does not imply stability[3] for the MPC scheme. It only guarantees that, inside the prediction window, the open loop optimization leads to a state trajectory that is bounded by a function of the initial state. Also note that this property is required only in a local region around the origin. The constant r can be sufficiently small, as long as it is nonzero.

[2] This condition generalizes the one-step stability enforcing MPC (Mayne et al., 2000)

[3] Unless otherwise specified, "stability" in this paper refers to "asymptotic stability of $x=0$."

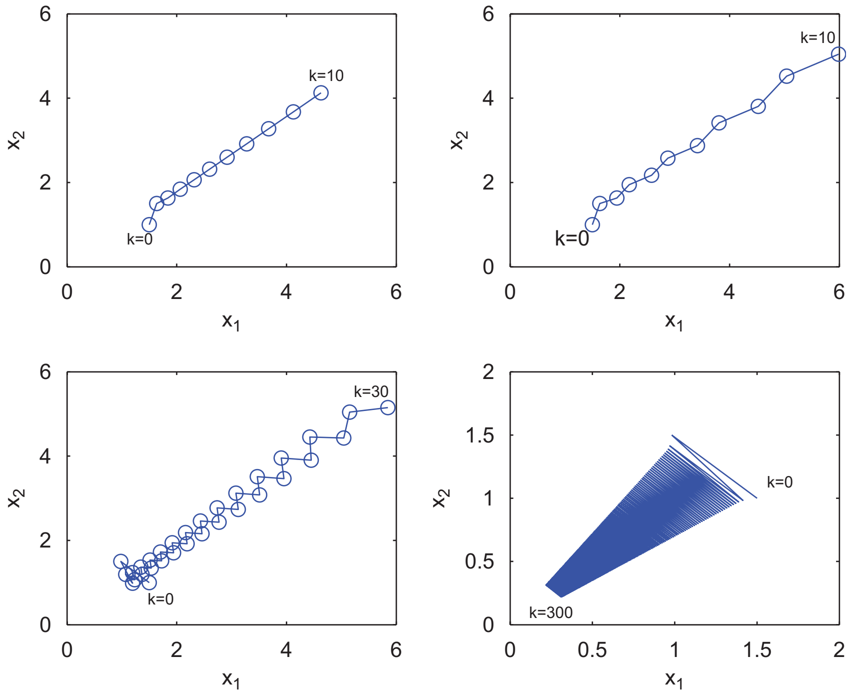


Fig. 1. Standard and block MPC for the plant of Example 2.1: case 1 (upper left): standard MPC; case 2 (upper right): MPC with block implementation, without constraint (9); case 3 (lower left): standard MPC with constraint (9); case 4 (lower right): block MPC with constraint (9).

Proof of Theorem 2.1. Due to the block implementation, (3) implies that we have

$$V(k) \leqslant \gamma^{k/N_c} V(0), \quad \forall k \in \{0, N_c, 2N_c, \ldots, \}. \tag{7}$$

Therefore, the sequence $\{V(k_1)\}, k_1 = 0, N_c, 2N_c, \ldots,$ converges to zero as $k_1 \to \infty$, which implies that $x(k_1) \to 0$ for $k_1 = 0, N_c, 2N_c, \ldots$.

For the system behavior within the control implementation window, i.e., for $x(k_1 + n|k_1), n = 1, \ldots, N_c - 1, k_1 \in \{0, N_c, 2N_c, \ldots, \}$, it follows from Proposition 2.2 that $\|x(k_1 + n|k_1)\| \leqslant g(\|x(k_1)\|)$ when $\|x(k_1)\|$ is sufficiently small. Therefore $\|x(k_1)\| \to 0$ and the continuity of $g(\cdot)$ at 0 together imply that $\|x(k_1 + n|k_1)\| \to 0$ for $k_1 \to 0$ and $n = 0, 1, \ldots, N_c - 1$. Hence, $x(k) \to 0$ for $k \to \infty$. This proves state convergence, while Lyapunov stability immediately follows from the definition of V. □

For $N_c = 1$, the result of Theorem 2.1 addresses a special case of a standard MPC with an additional constraint. For $N_c > 1$, even if the constraint (3) is enforced in its optimization of J, the standard MPC cannot guarantee stability because $V(x(k+1)) \leqslant \gamma V(x(k))$ is not established. With the block MPC algorithm, the constraint (3) is enforced not only in optimization, but also in control implementation and execution.

Remark 2.5. Note that the function $V(x(k))$ is not required to be monotonically decreasing at each step k. In fact, with the variable implementation window to be introduced later in Section 4, $V(x(k))$ is allowed to even increase over time, as long as one can find a subsequence of $V(x(k))$ which is monotonically decreasing over the time.

Remark 2.6. The condition (3) introduced in Theorem 2.1 is similar to the contractive condition used in (de Oliveira Kothare & Morari, 2000) where it is imposed at the end of the prediction horizon $k + N_r$ to establish the stability of the MPC scheme. In the context of block MPC, however, the time to impose this condition, namely N_c, is a design parameter which will be exploited to enhance the feasibility condition and to achieve good trade-off between computation intensity and disturbance rejection capability.

Example 2.1. Consider the following second order unstable system:

$$x(k+1) = \begin{bmatrix} 1 & 0.25 \\ 1 & 0 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(k), \tag{8}$$

$$y(k) = [-\tfrac{2}{3} \quad 1]\, x.$$

With the cost function defined as

$$J = \sum_{i=k}^{k+5} \left[x(i|k)^\top \begin{bmatrix} \frac{4}{9} & -\frac{2}{3} \\ -\frac{2}{3} & 1 \end{bmatrix} x(i|k) + u^2(i) \right],$$

one can show that the standard MPC (specifically, a nonblock MPC without terminal penalty) yields an unstable response, shown in case 1 in Fig. 1.

Adding a new constraint

$$V(k+2) \leqslant \gamma V(k), \tag{9}$$

where $V=y^2+\frac{1}{9}x_2^2$, $\gamma=0.98$, the two-step block MPC implementation gives a stable response, as shown by case 4 in Fig. 1.

It should be noted that the block implementation alone (i.e., without enforcing the constraint (9)) does not lead to a stable system for the example considered here (case 2 of Fig. 1). On the other hand, adding the constraint (9) to the standard MPC alone (i.e., using only the first control from the optimized sequence) cannot provide the stabilization mechanism either (case 3). It is the combination of the two that leads to the result of case 4 in Fig. 1.

For this example, the stability can also be achieved by imposing a terminal state constraint or by adding a penalty on the terminal state to the standard MPC formulation. As our example demonstrates, the block MPC with the constraint (9) provides an alternative mechanism to guarantee stability.

For the system discussed in Example 2.1, a constraint similar to (9) cannot be enforced in a single step. The main advantage of the block MPC implementation is that it allows longer horizon to enforce a condition that can lead to stability. Furthermore, it assures that the constraint is enforced not only at the optimization stage, but also at the control execution stage by selecting the proper implementation window size.

3. Recursive feasibility

With the block MPC scheme, we have shown that when the optimization problem (2) with an additional constraint (3) has feasible solutions for each $k\in\{0, N_c, 2N_c, \dots,\}$, the stability of the system can be established according to Theorem 2.1. More often than not, given a positive definite function V, the constraint (3) is not enforceable at each sampling instant $\{0, N_c, 2N_c, \dots,\}$ for arbitrary $x\in\mathscr{S}$. The recursive feasibility, namely feasibility preserved under state evolution, is in general not guaranteed even if the initial problem is feasible, as illustrated by the following example:

Example 3.1. Consider a double integrator system:

$$\begin{cases} x_1(k+1)=x_1(k)+x_2(k), \\ x_2(k+1)=x_2(k)+u(k) \end{cases} \tag{10}$$

with $|u|\leqslant 1$. If V is chosen as $V(x)=x_1^2+x_2^2$, we have

$$\begin{aligned} V(x(k+1|k)) &= x_1^2(k+1)+x_2^2(k+1) \\ &\geqslant (x_1(k)+x_2(k))^2 = V(x(k))+2x_1(k)x_2(k). \end{aligned}$$

If $x_1(k)x_2(k)\geqslant 0$, we have $V(x(k+1))\geqslant V(x(k))$, the constraint (3) cannot be satisfied for $N_c=1$ for any state in the first and third quadrants for any $\gamma\in(0,1)$. Now consider an initial state $x(0)=(-2,3)$ for which $V(1)\leqslant\gamma V(0)$ can be satisfied for $\gamma=0.95$ and some u (i.e., $u=-1$). This, however, will lead to $x(1)=(1,2)$ for which no control will exist to satisfy $V(2)\leqslant\gamma V(1)$. Similar scenario can be created for $N_c=2$. For example, starting from the initial state $x(0)=(-4,3)$, one can show that the constraint $V(2)\leqslant\gamma V(0)$ ($\gamma=0.95$) is enforceable with $|u|\leqslant 1$ but $V(4)\leqslant\gamma V(2)$ is not.

The above example shows that the optimization problem (2) with constraint (3) may not be recursively feasible for all states of interest. The block MPC with a variable implementation window, to be discussed in Section 4, will improve the enforceability of the added constraint, and thus assure recursive feasibility for (2). In order to define the block MPC with a variable implementation window, we first introduce the following definition:

Definition 3.1. (n-step (V,γ)-contractible region). Given a positive definite function V, an n-step (V,γ)-contractible region $\mathscr{P}_n$ is defined as the set of all the states x for which the constraint $V(x(n|0))\leqslant\gamma V(x(0))$ with $x(0)=x$, in addition to the constraints given in (2), can be satisfied by a proper choice of the string $u_{[0,n-1]}$.

The n-step (V,γ)-contractible region $\mathscr{P}_n$ depends on the contraction window n and the contraction rate γ, in addition to (1) and the selection of V. For $\gamma_1\geqslant\gamma_2$, we have $\mathscr{P}_n(\gamma_1)\supseteq\mathscr{P}_n(\gamma_2)$. However, $n_1\geqslant n_2$ does not necessarily imply $\mathscr{P}_{n_1}(\gamma)\supseteq\mathscr{P}_{n_2}(\gamma)$ for the same γ. In fact, a larger n does not necessarily lead to a larger (V,γ)-contractible region. For example, Fig. 2 shows the contractible regions for $n=1$ and $n=2$ for the double integrator system with $V(x)=x_1^2+x_2^2$, $|u|\leqslant 1$, and $\gamma=0.95$. It is obvious that $\mathscr{P}_2$ does not include $\mathscr{P}_1$, neither does $\mathscr{P}_1$ include $\mathscr{P}_2$.

Remark 3.1. In (Blanchini, 1994), a λ-contractive region is defined as the set $\mathscr{P}$ that for any $x\in\mathscr{P}$, there exists a control $u(x)$ such that $f(x,u(x))\in\lambda\mathscr{P}$ for some $\lambda\in(0,1]$. It should be noted that even when $n=1$, the (V,γ)-contractible region given by Definition 3.1 is different from the λ-contractive region defined in (Blanchini, 1994).

In an attempt to expand $\mathscr{P}_n$ and to make the optimization problem (2) with (3) recursively feasible, we introduce:

Definition 3.2. Let $\mathscr{P}_n$ be the n-step (V,γ)-contractible region for V as defined in Definition 3.1, where $n=1,\dots,N_c$ for some $N_c\leqslant N_r$, and N_r be the prediction horizon. We define $\mathscr{P}\triangleq\bigcup_{n=1}^{N_c}\mathscr{P}_n$. If[4] $\mathscr{S}=\mathscr{P}$, where $\mathscr{S}$ is the admissible set for x, then we call the function V N_c-step contractible for (1).

Definition 3.2 expands the V-contractible region to cover $\mathscr{S}$. By allowing the implementation window size to vary between 1 and N_c, the resulting region covered by $\mathscr{P}$ is always bigger than the fixed step contractible region.

It should be also pointed out that the contractibility of V is collectively defined by the function V and the system (1). Changing V or the system definition both could change the V-contractibility property. For example, for $V=x_1^2+x_2^2$ and $\gamma=0.95$, the contractible regions for the double integrator (10), and for the system (8) of Example 2.1 are shown in Figs. 2 and 3, respectively. For $\mathscr{S}=\{x;\,|x_1|\leqslant 2,\,|x_2|\leqslant 2\}$, V is two-step contractible for the unstable system (8), but it is not one-step or two-step contractible for the double integrator system (10).

[4] By the definition of $\mathscr{P}_n$, we always have $\mathscr{P}_n\subseteq\mathscr{S}$.

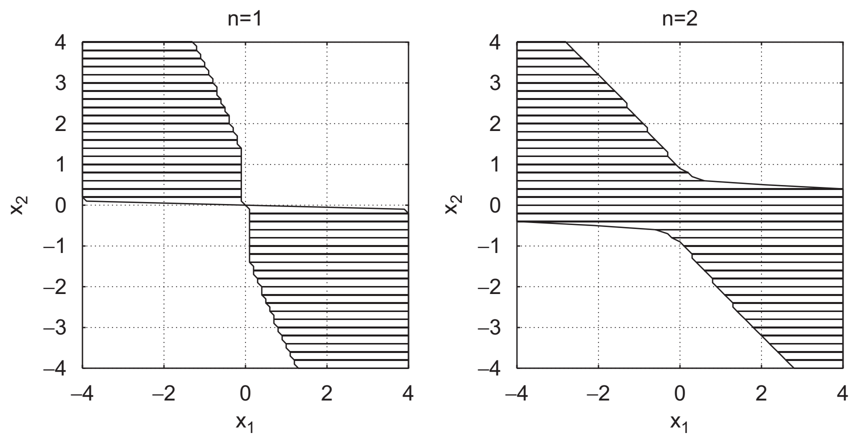


Fig. 2. Contractible regions for the double integrator system, $V(x) = x_1^2 + x_2^2$, $\gamma = 0.95$.

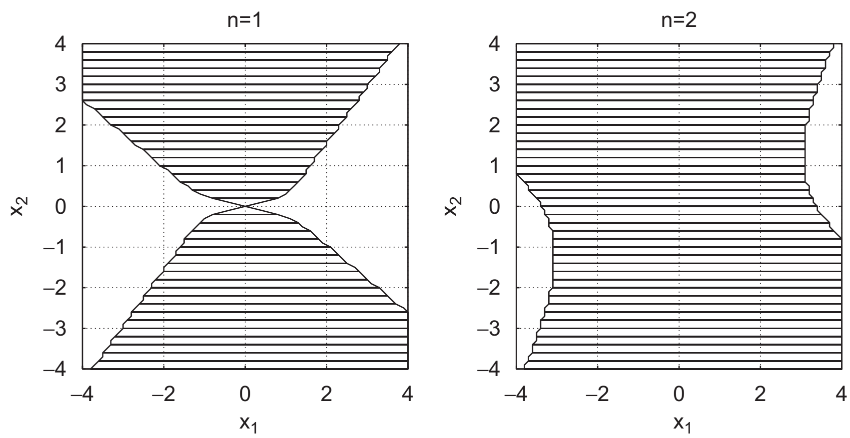


Fig. 3. Contractible regions for the unstable system (8), $V(x) = x_1^2 + x_2^2$, $\gamma = 0.95$.

Remark 3.2. The condition (3) is a special case of a more general condition

$$V(x(k+N_c|k)) - V(x(k)) \leqslant -\alpha(||x(k)||), \quad (11)$$

where α is a continuous function satisfying $\alpha(0) = 0$, $\alpha(x) > 0$ if $x \neq 0$. Since the complete controllability of discrete-time systems depends on the time interval being sufficiently large, the constraint (11) may also not be enforceable at each sampling instant. Enforcing the condition (11) instead of (3) can nevertheless offer additional flexibility due to a choice of $\alpha(\cdot)$.

Remark 3.3. For most "well-behaved" nonlinear systems (such as those satisfy (A1)), one can always find V that is N_c-step contractible for some finite N_c, as long as $\mathscr{S}$ lies inside the stabilizable region with constraints of (2). One can show that any Control Lyapunov Function (Freeman & Kokotovic, 1996) constructed for the unconstrained system can be used as V.

Remark 3.4. For a chosen V and given n, $\mathscr{P}_n$ can be determined as

$$\mathscr{P}_n = \left\{ x \left| \min_{u_{[0,n]} \in \mathscr{U}} V(x(n|0)) \leqslant \gamma V(x(0)),\, x(0) = x \right. \right\}.$$

This calculation can be performed off-line over a set of grid-points for x, yielding a functional approximation of the contractible regions $\mathscr{P}_n$ and $\mathscr{P}$. Depending on V, the optimization problem involved in the definition of $\mathscr{P}_n$ may be nonconvex and difficult to solve numerically even for linear systems.

The following propositions specify the recursive feasibility condition:

Proposition 3.1. *The optimization problem* (2) *with* (3) *is recursively feasible when* $\mathscr{P} = \mathscr{S}$.

Proposition 3.2. *When* $\mathscr{P} \subset \mathscr{S}$, *if there exists a non-empty set* $\mathscr{S}_0$ *such that* $\mathscr{S}_0 \subseteq \mathscr{R}_v \subseteq \mathscr{P}$ *for some* $c > 0$, *where* $\mathscr{R}_v := \{x | V(x) \leqslant c\}$, *then the optimization problem* (2) *with* (3) *is recursively feasible for any* $x \in \mathscr{S}_0$.

These two propositions follow directly from the definition of $\mathscr{P}$.

The following example shows that when $\mathscr{P} \subset \mathscr{S}$, the recursive feasibility condition could lead to different N_c for different V.

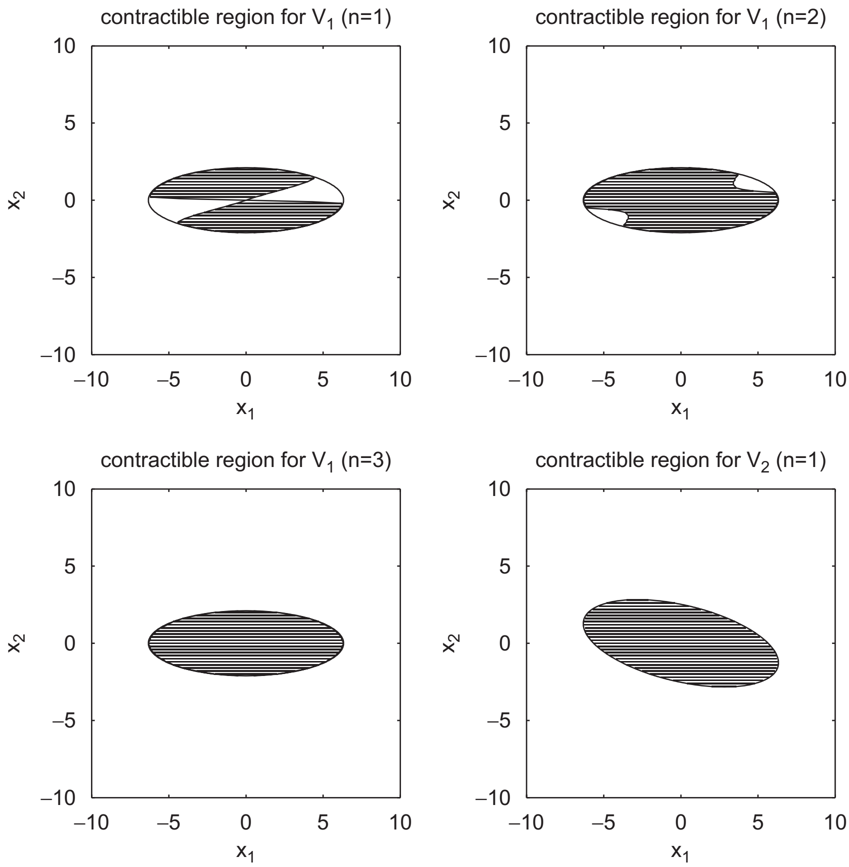


Fig. 4. Contractible regions for the double integrator system, $V_1(x) = x_1^2 + 9x_2^2$, and $V_2(x) = (x_1 + x_2)^2 + 4x_2^2$, $\gamma = 0.95$.

Example 3.2. Consider the double integrator system of (10), if $V(x) = x_1^2 + x_2^2$ is chosen, we can see from Fig. 2 that the set $\mathscr{S}_0 = \{x; |x_1| \leqslant 2, |x_2| \leqslant 2\}$ cannot be covered by $\mathscr{P} = \mathscr{P}_1 \bigcup \mathscr{P}_2$.

Now we consider two other functions defined as

$$V_1(x) = x_1^2 + 9x_2^2, \quad V_2(x) = (x_1 + x_2)^2 + 4x_2^2.$$

Their contractible regions are shown in Fig. 4.

Note that for V_1, even though we have $\mathscr{S}_0 \subseteq \mathscr{P}_1 \bigcup \mathscr{P}_2$, no constant c can be found such that $\mathscr{R}_v = \{x | V_1(x) \leqslant c\}$ is covered by $\mathscr{P}_1 \bigcup \mathscr{P}_2$ and covers $\mathscr{S}_0$. Therefore, we have to increase the control implementation window to $N_c = 3$ when $\mathscr{R}_{v1} = \{x | V_1(x) \leqslant 40\}$ satisfies the conditions in the Proposition 3.2, thus making $\mathscr{S}_0$ a region over which recursive feasibility condition will be satisfied.

For V_2, on the other hand, Fig. 4 shows that the recursive feasibility condition can be satisfied for $\mathscr{S}_0$ with $\mathscr{R}_{v2} = \{x | V_2(x) \leqslant 32\}$ for $N_c = 1$.

4. Stable block-MPC with a variable implementation window

We now exploit the properties of the contractible V and the variable control implementation window to design a stable MPC. Let N_c be the maximum control implementation window that defines $\mathscr{P}$, we first define the optimization task $\mathscr{T}_n$, for $1 \leqslant n \leqslant N_c$, as

$$\min J = \min_{\{u(\cdot)\}} \left\{ \sum_{i=k}^{k+N_r-1} L(x(i|k), u(i)) + K(x(k+N_r|k)) \right\}$$

subject to Eq.(1)

and constraints $x(i) \in \mathscr{S}, \quad u(i) \in \mathscr{U}$

$$V(x(k+n|k)) \leqslant \gamma V(x(k)). \tag{12}$$

Then we consider the following algorithms:

Algorithm 1. The minimum cost BMPC algorithm is defined as

1. For $x(k) \in \mathscr{P}$, solve the optimization problem $\mathscr{T}_n$ of (12) for $n = 1, 2, \ldots, N_c$. If no feasible solution exists, set $J_n^* = \infty$.
2. Find n^* that corresponds to $\min_n J_n^*$, where J_n^* is the minimum cost achieved for $\mathscr{T}_n$.
3. Apply n^*-step block MPC as defined in Definition 2.1.
4. Set $k = k + n^*$ and repeat the process after n^* steps.

Algorithm 2. The minimum window BMPC algorithm is defined as

(1) For $x(k) \in \mathscr{P}$, set $n = 1$.
(2) Solve the optimization problem $\mathscr{T}_n$ of (12). If (12) has a feasible solution, set $n^* = n$ and go to step 3. Otherwise, set $n = n + 1$ and repeat step 2.
(3) Apply the n^*-step block MPC as defined in Definition 2.1.
(4) Set $k = k + n^*$ and repeat the process after n^* steps.

For the two algorithm defined above, the following theorems establish their stability properties:

Theorem 4.1. *If the recursive feasibility condition given by Proposition* 3.1 (*or Proposition* 3.2) *is satisfied and either Algorithm* 1 or 2 *is applied, then* $\mathscr{S}$ (*or* $\mathscr{S}_0$) *is an attraction region for the equilibrium* $x = 0$ *of* (1). *Namely, for any* $x_0 \in \mathscr{S}$ (*or* $x_0 \in \mathscr{S}_0$), *the trajectory defined by the minimum-cost or the minimum window BMPC will converge to the equilibrium* $x = 0$.

The proof of Theorem 4.1 follows immediately from Theorem 2.1 and the definitions of Algorithms 1 and 2.

Remark 4.1. If the sets $\mathscr{P}_n$ are pre-calculated, then Algorithms 1 and 2 can be substantially simplified. For the minimum cost BMPC, the optimization task $\mathscr{T}_n$ does not need to be carried out for all n, but only for those whose corresponding $\mathscr{P}_n$ includes $x(k)$. Similarly, the minimum window n^* in Algorithm 2 can be determined without solving the optimization problem.

5. Example and discussion

We consider an example of a double integrator (10) with the control constraint $|u(k)| \leqslant 1$. The problem, patterned after the ship application in (Chen & Sun, 2005), is to control the system from the initial state $x(0)$ into the target set $S_f = \{x | x_1^2 + x_2^2 \leqslant 0.01\}$ so that the *total control effort*, denoted by $\sum u^2$, is minimized. The cost function (2) has a general form

$$J = \sum_{i=k}^{k+N_r-1} u(i)^2 + K(x(k + N_r|k)).$$

We let $V(x) = x_1^2 + 9x_2^2$, $\gamma = 0.95$, $N_r = 5$, and $K(x) = \beta \cdot V(x)$, where $\beta \geqslant 0$. The required theoretical assumptions are satisfied in this case, see Remark 2.1.

Numerical simulations were performed for (i) the block MPC with fixed implementation window N_c (3-BMPC, $N_c = 3$ for this study) and $\beta = 0$; (ii) the minimum cost BMPC (mc-BMPC) and $\beta = 0$; and (iii) the standard MPC (s-MPC, which essentially corresponds to $N_c = 1$ and $\gamma = +\infty$) and $\beta = 0.001$. In the s-MPC case, $\beta = 0.001$ was picked from a finite set of candidate values to minimize $\sum u^2$ for the initial conditions we considered. According to (Chen & Sun, 2005), the time to reach S_f (denoted by t_f) can delineate a more preferable solution in case two solutions provide approximately equal total control effort.

Table 1
Comparison of different MPC schemes for the double integrator example

	$x_0 = (5, 1)$		$x_0 = (6, 0)$		$x_0 = (-7, 0)$	
	$\sum u^2 dt$	t_f	$\sum u^2 dt$	t_f	$\sum u^2 dt$	t_f
mc-BMPC	0.6872	48	0.0008	119	0.001	119
3-BMPC	1.1526	60	0.0050	69	0.0108	92
s-MPC	0.1231	252	0.0107	132	0.0145	133

The values of $\sum u^2$ and t_f are summarized in Table 1 for a set of three initial conditions, which were selected arbitrarily.

As these results indicate, mc-BMPC can lower the total control effort as compared to 3-BMPC, and it can also provide a lower total control effort as compared to s-MPC for some initial conditions. The computational effort is less for 3-BMPC than for s-MPC since the optimization is performed less often; this advantage is eroded for mc-BMPC since multiple optimization problems must be solved within each time period.[5]

Design parameters for the BMPC schemes include: γ, the contraction rate; N_r, the prediction horizon; and the function V. Their choices will affect system performance as well as the attraction region of the equilibrium. In general, the effects of the prediction horizon, N_r, on the performance of the BMPC are similar to that of the standard MPC, which are discussed in (Mayne et al., 2000). On the other hand, in our specific example it can be shown that BMPC solution satisfies $u(k + N_c) = \cdots = u(k + N_r - 1)$ which can be exploited to simplify the computations. In general, reducing γ will make it harder to enforce constraint (3) and thus take more control effort. The effect of different choices of V on the response, however, is more complicated. It primarily affects the contractible region $\mathscr{P}_n$ and thus the control implementation window. For the double integrator, along with $V_1(x) = x_1^2 + 9x_2^2$, we also examined $V_2(x) = (x_1 + x_2)^2 + 4x_2^2$. The contractibility properties for these two functions are illustrated in Fig. 4. The results of the mc-BMPC algorithm with these two different V's were compared for the same initial condition $x_0 = (2, 2)$. The function V_2 leads to a trajectory with $\sum u^2 = 1.3501$ and $t_f = 51$, both are slightly larger than $\sum u^2 = 1.3017$ and $t_f = 49$ which are resulted from using the function $V_1(x)$.

While more detailed comparisons with other MPC approaches which can be used for this problem are beyond the scope of this brief paper, we note that block MPC can be used as an "add-on" (i.e., not a competing scheme), providing an extra degree of freedom in the control design.

6. Conclusion

In this paper, we proposed a novel MPC scheme which uses block implementation to assure stability. With the design

[5] Our implementation of the computations in this example (using MATLAB on a regular 2 GHz PC) was not optimized; hence we provide a directional assessment only, rather than reporting exact computing times.

flexibility offered by the block implementation with variable window size, we are able to enforce a constraint that leads to the decrease of a Lyapunov-like function over the time interval, therefore guaranteeing stability. This new design feature can also be exploited to improve performance, such as shown in the example in Section 5, or to save the on-line computational effort as the optimization is performed every $N_c T$ second for the block MPC instead of every T second for the standard MPC.

References

Alamir, M. (2006). A low dimensional contractive NMPC scheme for nonlinear systems stabilization: Theoretical framework and numerical investigation on relatively fast systems. In *Assessment and future directions in nonlinear model predictive control*. Lecture notes in control and information sciences. Berlin, Springer.

Alamir, M., & Bornard, G. (1995). Stability of a truncated infinite constrained receding horizon scheme, the general discrete nonlinear case. *Automatica*, 31 (9), 1353-1356.

Bemporad, A., & Morari, M. (1999). Control of systems integrating logic, dynamics and constraints. *Automatica*, *35*, 407–427.

Blanchini, F. (1994). Ultimate boundness control for uncertain discrete-time systems via set-induced Lyapunov functions. *IEEE Transactions on Automatic Control*, *39*(2), 428–433.

Chen, H., & Allgower, F. (1998). A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, *34*(10), 1205–1217.

Chen, S. H., & Sun, J. (2005). Reconfiguration control with energy and time constraints: a model predictive control approach. *Proceedings of the 2005 ASNE symposium on reconfiguration and survivability* FL: Turtle Beach.

de Oliveira Kothare, S. L., & Morari, M. (2000). Contractive model predictive control for constrained nonlinear systems. *IEEE Transactions on Automatic Control*, *45*(6), 1053–1071.

Freeman, R., Kokotovic, P.V., (1996). *Robust nonlinear control design state space and Lyapunov techniques*. Birkhäuser.

Lee, Y. I., Kouvaritakis, B., & Cannon, M. (2002). Constrained receding horizon control for nonlinear systems. *Automatica*, *38*, 2093–2102.

Maciejowski, J. M., (2002). *Predictive control with constraints*. Pearson Education.

Mayne, D. Q., Rawlings, J. B., Rao, C. V., & Scokaert, P. O. M. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, *36*, 789–814.

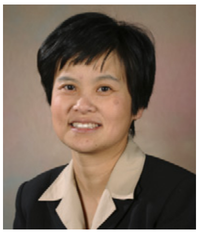
Michalska, H., & Mayne, D. Q. (1993). Robust receding horizon control of constrained nonlinear systems. *IEEE Transactions on Automatic Control*, *38*, 1623–1632.

Milam, M. B., Franz, B., Hauser, J. E., & Murray, R. (2005). Receding horizon control of a vectored thrust flight experiment. *IEE proceedings control theory applications* 152 (3).

Qin, S.J. & Badgwell, T.A., (1997). An overview of industrial model predictive control technology. In: Kantor, J.C., Garcia, C.E., Carnahan, B. (Eds.), *Fifth international on conference chemical process control*, CACHE, AIChE (pp. 232–256).

Ricker, N. L. (1985). Use of quadratic programming for constrained internal model control. *Industrial Engineering Chemical Process Design and Development*, *24*, 925–936.

Sun, J., Chen, S.H., & Kolmanovsky, I.V., (2005). A stable block model predictive control with variable implementation horizon. *Proceedings of the 2005 American control conference*, Portland, Oregon.



Jing Sun received her Ph. D. degree from the University of Southern California in 1989, and her B. S. and M. S. degrees from the University of Science and Technology of China in 1982 and 1984, respectively. From 1989–1993, she was an Assistant Professor in Electrical and Computer Engineering Department, Wayne State University. She joined the Ford Research Laboratory in 1993 where she worked in the Powertrain Control Systems Department. After spending almost 10 years in industry, she came back to academia and joined the faculty of the College of Engineering at the University of Michigan in 2003 as an Associate Professor. Her research interests include system and control theory and its applications to marine and automotive propulsion systems.
She holds over 30 US patents and has co-authored a textbook on Robust Adaptive Control. She is an IEEE Fellow and one of the three recipients of the 2003 IEEE Control System Technology Award. She is also a subject editor for International Journal of Adaptive Control and Signal Processing.

Ilya V. Kolmanovsky studied as an undergraduate at the Moscow Aviation Institute, Russia. He received the M.S. degree and the Ph.D. in aerospace engineering, in 1993 and 1995, respectively, and the M.A. degree in mathematics, in 1995, all from the University of Michigan, Ann Arbor. He is currently a Technical Leader in Powertrain Control at the Ford Research & Advanced Engineering of Ford Motor Company, Dearborn, MI. In addition to expertise in the automotive engine and powertrain control, his research interests include potential of advanced control techniques as an enabling technology for advanced automotive systems, and several areas of control theory, which include constrained control, optimization-based and model-predictive control, and control of nonlinear mechanical, nonholonomic, and underactuated systems.
Dr. Kolmanovsky has served as an Associate Editor of IEEE CSS Conference Editorial Board (1997–1999), IEEE Transactions on Control Systems Technology (2002–2004), IEEE Transactions on Automatic on (2005 till present), and he was the Chair of IEEE Technical Committee on Automotive Control (2005–2006). He is the recipient of 2002 Donald P. Eckman Award of American Automatic Control Council for contributions to nonlinear control and for pioneering work in automotive engine control of powertrain systems, and of 2002 IEEE Transactions on Control Systems Technology Outstanding Paper Award.

Reza Ghaemi was born in Tehran, Iran in 1976. He received the B.S. and M.S. degrees from the University of Tehran, Tehran, Iran, in 1998 and 2001, respectively. From 2001 to 2004, he conducted research on control and monitoring of power electronic systems at the Power Research Institute, Tehran, Iran. He is presently a graduate research assistant in the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, USA. His research interests include optimal control theory and model predictive control.

Shuhao Chen received his B.S. degree in automatic control from the Tsinghua University, Beijing, China, his M.S. degree in industrial automation from Xi'an Jiaotong University, Xi'an, China, and his Ph.D. degree in electrical engineering from the University of Virginia, Charlottesville, VA, in 1993, 1998, and 2004, respectively. His research interests include adaptive control, fault-tolerant control, and power system control.