2005 American Control Conference
June 8-10, 2005. Portland, OR, USA

A Stable Block Model Predictive Control with Variable Implementation Horizon

Jing Sun, Shuhao Chen, Ilya Kolmanovsky

Abstract— **In this paper, we present a stable receding horizon model predictive control for discrete-time nonlinear systems. The standard MPC scheme is modified to incorporate (1) a block implementation scheme where a string of the optimized input 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 constraints enforcement. Stability of the closed-loop system with the proposed algorithm is established. Examples and simulation results 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.**

I. Introduction

Model predictive control (MPC)[1], despite the computational intensity associate with its on-line implementation, has found many successful industrial applications [2]. Its simple and flexible formulation, together with its capability in dealing with constraints and nonlinearities, has been a major advantage. The main challenges that come with the MPC scheme, such as how to ease the computational requirements and how to guarantee stability, have also attracted attentions of many researchers and control practitioners.

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, whose first element 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 function and receding horizon 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 realtime implementation of MPC schemes. However, advances in computing technology and efficiency improvements in optimization algorithms are easing the computational burden and paving the way for new MPC applications. As the cost of computer hardware goes down, the realtime implementation of MPC schemes also becomes more affordable. 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 (see the survey paper [3] and the references therein). 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 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 then switched to a local stabilizing controller [4].

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 the MPC scheme. In particular, we propose a block MPC scheme where a string of the control signal from each optimization run is implemented instead of only the first element. An additional constraint, which guarantees that a Lyapunov-like function will be decreasing over time, is also enforced. By allowing the

Jing Sun and Shuhao Chen are with the Department of Naval Architecture and Marine Engineering, University of Michigan, Ann Arbor, MI 48109, Email: jingsun@umich.edu, shchen@engin.umich.edu.

Ilya V. Kolmanovsky is with Ford Motor Company, 2101 Village Road, Dearborn, MI 48124, Email: ikolmano@ford.com

implementation window to vary in size, the enforceability of the constraints is greatly enhanced. Stability of the closed-loop system with the proposed MPC is established. Numerical simulation examples are also given to illustrate the effectiveness of the proposed scheme.

The paper is organized as follows: In Section 2, we will first introduce the block MPC concept and specify the conditions under which the asymptotic stability can be achieved. The new constraint, added to the block MPC algorithm to guarantee stability, will be analyzed in Section 3 for its enforceability. This leads to the variable block MPC algorithm proposed in Section 4. A numerical example is given in Section 5 to demonstrate the control scheme and to illustrate how the design parameters in the block MPC implementation may impact the performance. Finally, a brief summary is included in Section 6 to conclude the paper.

II. 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 and u the input. Assume that the standard MPC scheme is designed by solving the following optimization problem at each time instant k:

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

where $\mathcal{S}$ and $\mathcal{U}$ are the admissible sets for the state and input respectively, and $L(x,u)$ and $K(x)$ are non-negative 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. The string $\{u(k), u(k+1), \cdots, u(k+N_r-1)\}$ is varied within the input admissible set to minimize the cost function J for (2).

Note that for time invariant systems given by (1), the optimal solution $\{u^*(k), u^*(k+1), \cdots, u^*(k+N_r-1)\}$ for the problem (2) depends only on the starting state $x(k)$. To assure a meaningful MPC formulation, we make the following assumption:

- (A1) Let $\{u^*(k), u^*(k+1), \cdots, u^*(k+N_r-1)\}$ be the solution for (2). If the sequence is applied to (1), then there exist a $r > 0$ and a continuous function $g(\cdot) > 0$ with $g(0) = 0$ such that if $\|x(k)\| \le r$, we have

$$\|x(k+i)\| \le g(\|x(k)\|), \;\forall i \in \{1, \cdots, N_r\}. \tag{3}$$

Note that (A1) does not imply stability[1] 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 non-zero. (A1) is satisfied for most practical systems with the MPC-based control that we encountered. In particular, we have:

Proposition 2.1: For linear systems with a quadratic cost function, (A1) is satisfied if

- (A2) There exists a $\mathcal{B}(x) \subset \mathcal{S}$, with $x = 0$ being an interior point for $\mathcal{B}(s)$, inside which the constraints are inactive for the optimization problem (2).

The proof of the Proposition can be derived by recognizing the fact that the unconstrained optimal solution $\{u^*(k), u^*(k+1), \cdots, u^*(k+N_r-1)\}$ is linear in $x(k)$ for linear systems with a quadratic cost function. Therefore, for sufficiently small $\|x(k)\|$, (A2) guarantees that the solution of (2) is equal to the solution of the corresponding unconstrained problem, thus it is linear in $x(k)$. (A1) then follows immediately.

For linear models, constraints on the states are normally imposed for large signals, when the states go beyond the linear region. Since (A2) in Proposition 2.1 is only needed for a small neighborhood $\mathcal{B}(s)$ around the equilibrium, it is non-restrictive and can almost always be satisfied.

In contrast to the standard MPC which applies the first element in the string of $\{u(k), u(k+1), \cdots, u(k+N_r-1)\}$ to the control signal, we propose the following block MPC scheme:

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

- (a) the first N_c elements, i.e., the sub-string $\{u^*(k), u^*(k+1), \cdots, u^*(k+N_c-1)\}$ of $\{u^*(k), u^*(k+1), \cdots, u^*(k+N_r-1)\}$, are applied to the control signal at time $t = k, k+1, \cdots, k+N_c-1$ and
- (b) the optimization for (2) is repeated after N_c steps at $t = k+N_c$ with the new state $x(k+N_c)$,

we call N_c the control implementation window and the resulting control scheme 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. Without adding new constraints, the block MPC requires less computation effort when $N_c > 1$. In fact, the N_c-block MPC can be viewed as a hybrid control scheme where the control update and optimization are carried out at two different sampling intervals: T and N_cT. Like many other multi-rate sampling control systems, it has the advantage

[1]In this paper, we are interested in the asymptotic stability of the equilibrium $x = 0$. Unless otherwise specified, "stability" in the rest of the paper refers to "asymptotic stability of $x = 0$."

of requiring less computational 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.

It should be pointed out that the notion of the control implementation window N_c used here is different from the control window defined in [5]. In [5], the control window N refers to the dimension of the optimization problem, when only the first N elements of the input sequence $\{u(k), \cdots, u(k+N_r-1)\}$ are allowed to vary in minimizing J.

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

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

$$V(x(k+N_c)) \leq \gamma V(x(k)) \tag{4}$$

is satisfied for some $0 < \gamma < 1$ and for $k \in \{0, N_c, 2N_c, \cdots,\}$, then the closed-loop system with the N_c-block MPC is asymptotically stable. △

For the proof of Theorem 2.1, we choose V as the Lyapunov function. Because of the block implementation, (4) implies that we have:

$$V(k) \leq \gamma^{\frac{k}{N_c}} V(0), \quad \forall k \in \{0, N_c, 2N_c, \cdots\}.$$

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

For the system behavior within the control implementation window, i.e., for $x(k_1+n), n = 1, \cdots, N_c-1, k_1 \in \{0, N_c, 2N_c, \cdots,\}$, assumption (A1) implies that $\|x(k_1+n)\| \leq g(\|x(k_1)\|)$ when $\|x(k_1)\|$ is sufficiently small, which is guaranteed by $x(k_1) \to 0$. Therefore $\|x(k_1)\| \to 0$ and the continuity of $g(\cdot)$ at 0 together imply that $\|x(k_1+n)\| \to 0$ for $k_1 = 0, N_c, \cdots$ and $n = 0, 1, \cdots, N_c-1$. Hence, $x(k) \to 0$ for $k \to \infty$ and the stability follows.

For $N_c = 1$, the result of Theorem 2.1 can be viewed as a special case of the standard MPC with an additional constraint. For $N_c > 1$, even if the constraint (4) is enforced in its optimization of J, the standard MPC cannot guarantee stability because $V(x(k+1)) \leq \gamma V(x(k))$ is not established. With the block MPC algorithm, the constraint (4) is enforced not only in optimization, but also in control implementation and execution.

[2]This condition generalizes the one-step stability enforcing MPC [3].

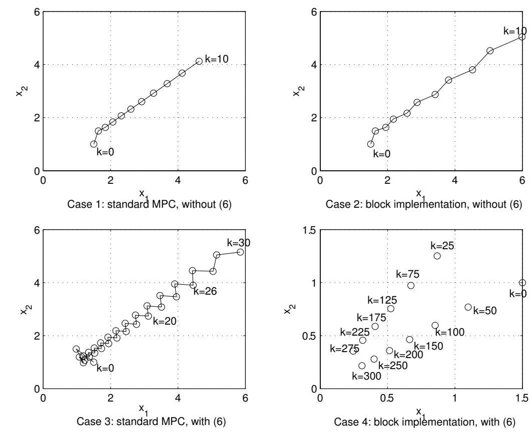

Fig. 1. Standard and block MPC for the plant of Example 2.1: case 1: standard MPC; case 2: MPC with block implementation, without constraint (6); case 3: standard MPC with constraint (6); case 4: block MPC with constraint (6).

Example 2.1. Consider the following second order unstable system

$$x(k+1) = \begin{bmatrix} 1 & 0.25 \\ 1 & 0 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(k), \qquad y(k) = \begin{bmatrix} -\frac{2}{3} & 1 \end{bmatrix} x. \tag{5}$$

With the cost function defined as:

$$J = \sum_{i=0}^{5} [x(k+i)^\top Q x(k+i) + u^2(k+i)]$$

where

$$Q = \begin{bmatrix} 4/9 & -2/3 \\ -2/3 & 1 \end{bmatrix}$$

one can show that the standard MPC yields an unstable response, shown in case 1 in Fig. 1.

Adding a new constraint

$$V(k+2) \leq \gamma V(k) \tag{6}$$

where $V = y^2 + 1/9 x_2^2, \gamma = 0.98$, the 2-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 (6)) does not lead to a stable system for the example considered here (case 2 of Figure 1). On the other hand, adding the constraint (6) to the standard MPC alone(i.e., use 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 the system discussed in Example 2.1, a constraint similar to (6) 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 the proper implementation.

III. Enforceability of the Constraint (4)

With the block MPC scheme, we have shown that when the optimization problem (2) with an additional constraint (4) has feasible solutions for each $k \in \{0, N_c, 2N_c, \cdots\}$, 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 (4) is not enforceable at each sampling instant $\{0, N_c, 2N_c, \cdots,\}$ for arbitrary $x \in \mathcal{S}$. Even if the problem (2) with (4) has feasible solutions for some initial x_0, there is no guarantee that feasible solutions will exist for $x(k)$ when $k > 0$, as the following example shows:

Example 3.1: Consider the simple double integrator system:

$$\begin{cases} x_1(k+1) = x_1(k) + x_2(k) \\ x_2(k+1) = x_2(k) + u(k) \end{cases} \qquad (7)$$

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

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

If $x_1(k)x_2(k) \geq 0$, we always have $V(x(k+1)) \geq V(x(k))$. In other words, for $N_c = 1$, the constraint (4) cannot be satisfied 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 $V(1) \leq \gamma V(0)$ 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 $V(2) \leq \gamma V(1)$.

Similar scenario can be created for $N_c = 2$. For example, if one chooses the initial state as $x(0) = (-4, 3)$, then the constraint $V(2) \leq \gamma V(0)$ ($\gamma = 0.95$) is enforceable. However, $V(4) \leq \gamma V(2)$ cannot be enforced with bounded input $|u| \leq 1$. ∇

The above example shows that the constraint (4) may not be enforceable for all states of interest. Furthermore, even it is enforceable initially, since the property is not invariant for (1), the enforceability may be lost with state transition. The block MPC with a variable implementation window, to be discussed in Section 4, will improve the enforceability of the added constraint, and thus mitigating the problem caused by the lack of feasible solutions 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 $\mathcal{P}_n$ is defined as the set of all the states x_0 for which the constraint $V(x(k+n)) \leq \gamma V(x(k))$, in addition to the constraints given in (2), is enforceable by a proper choice of the string $\{u(k), u(k+1), \cdots, u(k+n-1)\}$.

The n-step (V, γ)-contractible region $\mathcal{P}_n$ depends on the contraction window n and the contraction rate γ, in addition to (1) and the selection of V. For $\gamma_1 \geq \gamma_2$, we have $\mathcal{P}_n(\gamma_1) \supseteq \mathcal{P}_n(\gamma_2)$. However, $n_1 \geq n_2$ does not necessarily imply $\mathcal{P}_{n_1}(\gamma) \supseteq \mathcal{P}_{n_2}(\gamma)$ 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 $V(x) = x_1^2 + x_2^2$, $|u| \leq 1$, and $\gamma = 0.95$. It is obvious that $\mathcal{P}_2$ does not include $\mathcal{P}_1$, neither does $\mathcal{P}_1$ include $\mathcal{P}_2$.

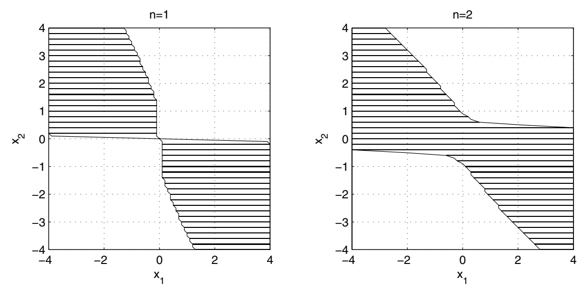

Fig. 2. Contractible regions for the double integrator system, $V(x) = x_1^2 + x_2^2, \gamma = 0.95$.

Remark 3.1: In [6], the λ-contractive region is defined as the set $\mathcal{P}$ that for any $x \in \mathcal{P}$, there exists a control $u(x)$ such that $f(x, u(x)) \in \lambda\mathcal{P}$ 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 [6].

If $x_0 \in \mathcal{P}_n$ does not guarantee $x(k+n) \in \mathcal{P}_n$, the value of Theorem 2.1 is limited. In an attempt to expand $\mathcal{P}_n$ and to make the constraint (4) easier to be satisfied, we introduce:

Definition 3.2: Let $\mathcal{P}_n$ be the n-step contractible region for V as defined in Definition 3.1, where $n = 1, \cdots, N_c$ for some $N_c \leq N_r$, and N_r be the prediction horizon. We define

$$\mathcal{P} \triangleq \bigcup_{n=1}^{N_c} \mathcal{P}_n.$$

If $\mathcal{S} = \mathcal{P}$[3], where $\mathcal{S}$ 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 $\mathcal{S}$. By allowing the implementation window size to vary between 1 and N_c, the resulting region covered by $\mathcal{P}$ 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-contractible property. For example, for $V = x_1^2 + x_2^2$ and $\gamma = 0.95$, the contractible regions for the double integrator (7) and for the system (5) of Example 2.1 are shown in Fig. 2 and Fig. 3 respectively. For $\mathcal{S} = \{x \| x_1| \leq 2, |x_2| \leq 2\}$, V is 2-step contractible for the unstable system (5), but it is not 1-step or 2-step contractible for the double integrator system (7).

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

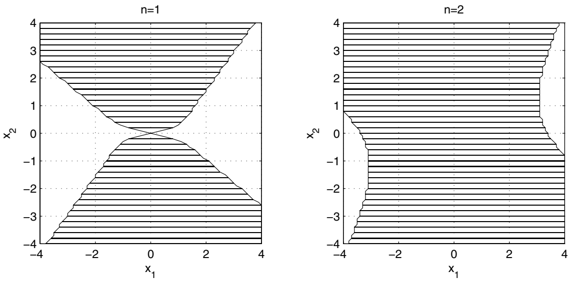

Fig. 3. Contractible regions for the unstable system (5), $V(x) = x_1^2 + x_2^2, \gamma = 0.95$.

Remark 3.2: The condition (4) is a special case of a more general condition

$$V(x(k+N_c)) - V(x(k)) \leq -\alpha(||x(k||), \qquad (8)$$

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 (8) may also not be enforceable at each sampling instant. Enforcing the condition (8) instead of (4) 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 (A2) in Proposition 2.1), one can always find V that is N_c-step contractible for some finite N_c, as long as $\mathcal{S}$ lies inside the stabilizable region with constraints of (2). One can show that any Control Lyapunov Function [7] constructed for the unconstrained system can be used as V. More rigorous proof is omitted due to space limitations.

IV. 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 $\mathcal{P}$, we first define the optimization task $\mathcal{T}_n$, for $1 \leq n \leq N_c$, as:

$$\begin{aligned} \min J &= \min_{\{u(\cdot)\}} \left\{ \sum_{i=k}^{k+N_r-1} L(x(i), u(i)) + K(x(k+N_r)) \right\} \\ & \text{subject to equation (1)} \\ & \text{and constraints } x(i) \in \mathcal{S}, \quad u(i) \in \mathcal{U} \qquad (9) \\ & V(x(k+n)) \leq \gamma V(x(k)) \end{aligned}$$

Then we consider the following algorithms:

Algorithm 1: The minimum cost BMPC algorithm is defined as

1) For $x(k) \in \mathcal{P}$, solve the optimization problem $\mathcal{T}_n$ of (9) for $n = 1, 2, \cdots, N_c$. 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 $\mathcal{T}_n$.
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 algorithms is defined as

1) For $x(k) \in \mathcal{P}$, set $n = 1$.
2) Solve the optimization problem $\mathcal{T}_n$ of (9). If (9) 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 $\mathcal{S} = \mathcal{P}$ and Algorithm 1 or 2 is applied, then $\mathcal{S}$ is an attraction region for the equilibrium $x = 0$ of (1). Namely, for any $x_0 \in \mathcal{S}$, 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 Algorithm 1 and 2.

Proposition 4.1: When $\mathcal{P} \subset \mathcal{S}$, if there exists a subset $\mathcal{S}_0$ satisfying (a) $\mathcal{S}_0 \subseteq \mathcal{P}$, (b) $x = 0$ is an interior point of $\mathcal{S}_0$, (c) there exists a constant c such that the set defined as $\mathcal{R}_v = \{x | V(x) \leq c\}$ satisfies $\mathcal{S}_0 \subseteq \mathcal{R}_v \subseteq \mathcal{P}$, then any trajectory of (1) defined by Algorithm 1 or 2 with $x_0 \in \mathcal{S}_0$ will converge to the origin.

For unbounded $\mathcal{S}$, satisfying $\mathcal{S} = \mathcal{P}$ is very difficult, if not impossible, especially with bounded u. In this case, Proposition 4.1 can be applied to determine the region of attraction $\mathcal{S}_0$.

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

Remark 4.2: The condition associated with $\mathcal{R}_v$ in Proposition 4.1 is needed to guarantee that the enforceability of the constraint (4) will not get lost along the trajectory. Without it, one may encounter the scenarios that, even when (4) is enforced, the trajectories inside $\mathcal{P}$ may leave $\mathcal{P}$ at the end of the implementation window. The requirement for $\mathcal{S}_0 \subseteq \mathcal{R}_v \subseteq \mathcal{P}$, however, might lead to different N_c for different V, as the following example shows.

Example 4.1: Consider the double integrator system of (7), if $V(x) = x_1^2 + x_2^2$ is chosen, we can see from Fig. 2 that the set $\mathcal{S}_0 = \{x | |x_1| \leq 2, |x_2| \leq 2\}$ cannot be covered by $\mathcal{P} = \mathcal{P}_1 \bigcup \mathcal{P}_2$.

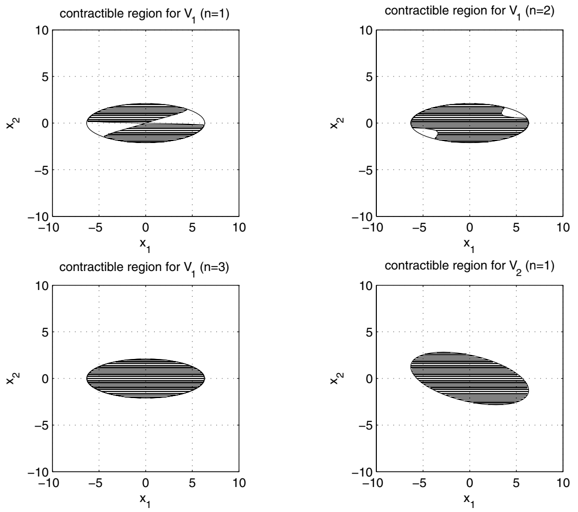

Fig. 4. Contractible regions for the double integrator system, $V_1(x) = x_1^2 + 9x_2^2$, $V_2(x) = (x_1 + x_2)^2 + 4x_2^2, \gamma = 0.95$.

Now we consider two other functions defined as

$$V_1(x) = x_1^2 + 9x_2^2$$

and

$$V_2(x) = (x_1 + x_2)^2 + 4x_2^2.$$

Their contractible regions are shown in Fig. 4.

Note that for V_1, even though we have $\mathcal{S}_0 \subseteq \mathcal{P}_1 \bigcup \mathcal{P}_2$, no constant c can be found such that $\mathcal{R}_v = \{x|V_1(x) \leq c\}$ is covered by $\mathcal{P}_1 \bigcup \mathcal{P}_2$ and covers $\mathcal{S}_0$. Therefore, we have to increase the control implementation window to $N_c = 3$ when $\mathcal{R}_{v1} = \{x|V_1(x) \leq 40\}$ satisfies the conditions in the Proposition 4.1, thus making $\mathcal{S}_0$ an attractive region with V_1 used in the BMPC algorithm.

For V_2, on the other hand, Fig. 4 shows that the conditions on $\mathcal{R}_v$ can be satisfied with $\mathcal{R}_{v2} = \{x|V_2(x) \leq 32\}$ for $N_c = 1$. ∇

V. Example and Discussion

As an example, we consider the minimum energy control problem for the double integrator system. The control objective is to regulate the system from the initial state $x(0)$ into the target set $S_f = \{x|x_1^2 + x_2^2 \leq 0.01\}$ with a minimum control effort. For comparison, numerical simulations are performed for three different MPC schemes: the minimum cost BMPC (mc-BMPC), the block MPC with fixed implementation window N_c (3-BMPC, $N_c = 3$ for this study), and the standard MPC (s-MPC) with a constraint $V(x(k+3)) \leq \gamma V(x(k))$.

With $V(x) = x_1^2 + 9x_2^2$, $\gamma = 0.95$, $N_r = 5$, the simulation results, which include the total control cost and time to reach the target, are summarized in Table 1 for three different MPC schemes. For all three initial conditions which are selected randomly, the mc-BMPC gives the best performance in terms of the control effort.

Table 1: Comparison of Different MPC Schemes

	$x_0 = (5,1)$		$x_0 = (6,0)$		$x_0 = (2,2)$	
	$\int u^2 dt$	t_f	$\int u^2 dt$	t_f	$\int u^2 dt$	t_f
mc-BMPC	0.6872	48	0.0008	119	1.3017	49
3-BMPC	1.1526	60	0.0050	69	1.4158	119
s-MPC	0.9586	86	0.0041	94	1.6798	122

t_f is the time it takes to reach the target set.

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. The effects of the prediction horizon on the performance of the BMPC are similar to that of the standard MPC, which are discussed in [3]. In general, reducing γ will make it harder to enforce constraint (4) 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 $\mathcal{P}_n$ and thus the control implementation window. For the double integrator system, two different functions $V_1(x) = x_1^2 + 9x_2^2$ and $V_2(x) = (x_1 + x_2)^2 + 4x_2^2$ are used and their contractibility properties are shown in Fig. 4. The results of the same algorithm with different V are compared for the same initial condition $x_0 = (2,2)$. The function V_2 leads to a trajectory with $\int_0^{t_f} u^2 = 1.3501$ and $t_f = 51$, both are slightly larger than those with $V_1(x)$ shown in Table 1.

VI. Conclusion

In this paper, we propose a new MPC scheme which uses block implementation to assure stability. With the design flexibility offered by the block implementation with variable window size, we are able to enforce a constraint that lead 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_cT instead of every T interval for the block MPC.

References

[1] J. M. Maciejowski, *Predictive Control with Constraints*, Pearson Education, 2002.

[2] S. J. Qin and T. A. Badgwell, "An overview of industrial model predictive control technology," In J. C. Kantor, C. E. Garcia and B. Carnahan (Eds.) *Fifth International Conference on Chemical Process Control*, CACHE, AIChE, pp. 232-256, 1997.

[3] D. Q. Mayne, J. B. Rawlings, C. V. Rao, P. O. M. Scokaert, "Constrained model predictive control: stability and optimality," *Automatica*, Vol. 36, pp789-814, 2000.

[4] Michalska and Mayne, "Robust receding horizon control of constrained nonlinear systems," IEEE Transactions on Automatic Control, Vol. 38, 1623-1632, 1993.

[5] Alamir, Bornard, "Stability of a truncated infinite constrained receding horizon scheme, the general discrete nonlinear case," Automatica, 31(9), 1353-1356, 1995

[6] F. Blanchini, "Ultimate Boundness Control for Uncertain Discrete-Time Systems via Set-Induced Lyapunov Functions," IEEE Transactions on Automatic Control, vol. 39, no. 2, pp.428-433, Feb. 1994.

[7] R. Freeman and P. V. Kokotovic, Robust nonlinear control design state space and Lyapunov techniques, Birkhäuser, 1996.