Move Blocking Strategies in Receding Horizon Control

Raphael Cagienard*, Pascal Grieder*, Eric C. Kerrigan† and Manfred Morari*

Abstract— In order to deal with the computational burden of optimal control, it is common practice to reduce the degrees of freedom by fixing the input or its derivatives to be constant over several time-steps. This policy is referred to as "move blocking".

This paper will address two issues. First, a survey of various move blocking strategies is presented and the shortcomings of these blocking policies, such as the lack of stability and constraint satisfaction guarantees, will be illustrated. Second, a novel move blocking scheme, "Moving Window Blocking" (MWB), will be presented. In MWB, the blocking strategy is time-dependent such that the scheme yields stability and feasibility guarantees for the closed-loop system. Finally, the results of a large case-study are presented that illustrate the advantages and drawbacks of the various control strategies discussed in this paper.

Keywords: Move Blocking, Optimal Control, Receding Horizon Control, Model Predictive Control

I. INTRODUCTION

In the literature, numerous variations of discrete-time, finite horizon constrained optimal control have been presented, e.g. [1], [7]. In most implementations, the optimal sequence of inputs is computed at each time-step and subsequently only the first element of the sequence is applied. This policy is referred to as *Receding Horizon Control (RHC)* or *Model Predictive Control (MPC)*. One of the major open issues of MPC is the computational complexity, which becomes prohibitive for large systems or systems with fast dynamics. In the standard problem formulation, the degrees of freedom of an RHC problem correspond to the number of inputs m multiplied with the prediction horizon N. The degrees of freedom are the dominating factor for complexity, regardless of whether the optimization problem is solved on-line, or the equivalent feedback solution is derived off-line [2], [4]. In order to deal with this issue, it is common practice to reduce the degrees of freedom by fixing the input or its derivatives to be constant over several time-steps (e.g. [8], [9]). Although this alleviates the computational restrictions, the resulting control policies do not guarantee stability or constraint satisfaction.

In this paper, we first review various move-blocking strategies and discuss some of their disadvantages. Following this, we introduce a novel blocking strategy that provides guarantees on stability as well as constraint satisfaction. The various schemes are compared in a large case-study that is presented in the final section before concluding.

* Automatic Control Laboratory, Swiss Federal Institute of Technology (ETH), CH-8092 Zurich, Switzerland
† Royal Academy of Engineering Post-doctoral Research Fellow, Department of Engineering, University of Cambridge, UK
Corresponding Author: Email: cagienard@control.ee.ethz.ch

II. STANDARD PROBLEM FORMULATION

In this section we will give a brief overview of the optimal control problem considered in this paper. After presenting the standard finite horizon optimal control problem, a brief discussion on receding horizon control (RHC), stability and feasibility properties of the feedback controller ensues. We restrict ourselves to the following linear, time-invariant, discrete-time system

$$x(k+1) \quad = \quad Ax(k) + Bu(k), \tag{1}$$

with $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$ and (A, B) stabilizable. We will let x or $x(k)$ denote the measured state at time k and x_i denote the predicted state at time $k+i$, given the state $x(k)$ at time k and some finite input sequence $\{u_0, \ldots, u_{i-1}\}$. Assume now that the states and inputs of system (1) are subject to the constraints

$$x(k) \in \mathbb{X} \subseteq \mathbb{R}^n, \quad u(k-1) \in \mathbb{U} \subseteq \mathbb{R}^m, \quad \forall k \geq 1 \tag{2}$$

where $\mathbb{X}$ and $\mathbb{U}$ are compact polyhedral sets containing the origin in their interior. Before going further, we will introduce the following definitions

Definition 1: The set $\mathcal{X}_I(K)$ will denote the maximum positively invariant set for the linear system (1) subject to the stabilizing linear feedback controller K that satisfies (2):

$$\mathcal{X}_I(K) \quad := \quad \{x(0) \in \mathbb{R}^n \mid x(k) \in \mathbb{X}, \; Kx(k) \in \mathbb{U},$$
$$x(k+1) = (A+BK)x(k), \; \forall k \geq 0\}$$

The set $\mathcal{X}_I(K)$ is easily computed with the algorithm in [3]. Now consider the finite-time constrained LQR problem

$$V_N^*(x) := \min_{u_0, \ldots, u_{N-1}} \sum_{k=0}^{N-1} \left(u_k' R u_k + x_k' Q x_k \right)$$
$$+ x_N' P x_N, \tag{3a}$$

$$\text{subj. to } x_k \in \mathbb{X}, \quad u_{k-1} \in \mathbb{U}, \quad \forall k \in \{1, \ldots, N\}, \tag{3b}$$

$$x_N \in \mathcal{X}_I(K) \tag{3c}$$

$$x_{k+1} = Ax_k + Bu_k, \quad x_0 = x. \tag{3d}$$

For the sake of simplicity, we will assume that $Q = Q' \succ 0$, $P = P' \succ 0$ and $R = R' \succ 0$ (this can be relaxed to $(Q^{1/2}, A)$ detectable and $Q = Q' \succeq 0$, $P = P' \succeq 0$). The unique optimizer to the above optimal control problem is denoted as $U^*(x) := [u_0^*(x)', \ldots, u_{N-1}^*(x)']'$.

Definition 2: We define the N-step feasible set $\mathcal{X}_N \subseteq \mathbb{R}^n$ as the set of states $x(k)$ for which the optimal control problem (3) is feasible, i.e. for which problem (3)

has a solution satisfying the constraints:

$$\mathcal{X}_N := \{x_0 \in \mathbb{R}^n \mid \exists U = [u_0', \ldots, u_{N-1}']' \in \mathbb{R}^{Nm},$$
$$x_k \in \mathbb{X}, \ u_{k-1} \in \mathbb{U}, \quad \forall k \in \{1, \ldots, N\},$$
$$x_N \in \mathcal{X}_I(K)\}.$$

Because problem (3) depends on the current state x, it can be solved either by solving the quadratic program (3) for a given x, or as shown in [2], [10], by solving problem (3) for all x within a polyhedral set of values, i.e., by considering (3) as a *multi-parametric Quadratic Program* (mp-QP) [2].

Theorem 1: [2] Consider the finite-time constrained regulation problem (3). The set of feasible states $\mathcal{X}_N$ is convex, the optimizer $U^* : \mathcal{X}_N \to \mathbb{R}^{Nm}$ is continuous and piecewise affine (PWA), i.e. $U^*(\cdot)$ has the form:

$$U^*(x) = F_r x + G_r, \quad \text{if} \quad x \in \mathcal{P}_r$$
$$\mathcal{P}_r = \{x \in \mathbb{R}^n \mid H_r x \leq K_r\}, \ r = 1, \ldots, \mathcal{R}$$

and the value function $V_N^* : \mathcal{X}_N \to \mathbb{R}$ is continuous, convex and piecewise quadratic.

In *Receding Horizon Control* (RHC), the optimizer $U^*(x)$ of problem (3) is computed at each time step but only the first input $u_0^*(x)$ of the sequence is applied to the plant, i.e.

$$u(k) = u_0^*(x(k)).$$

This control policy does not guarantee closed loop stability or constraint satisfaction without additional adjustments to the problem formulation in (3). It is standard practice [7] to obtain feasibility (i.e. constraint satisfaction) properties by adding the additional set constraint $x_N \in \mathcal{X}_I(K)$ to the optimal control problem (3). The terminal set constraint guarantees recursive feasibility, i.e. if a feasible input sequence exists at time $k = 0$ this implies that a feasible input sequence will be found for all $k > 0$. If the terminal weight P is chosen such that $F(x) = x'Px$ is a local Lyapunov function for the system subject to the linear feedback law K, then stability is guaranteed as well.

Both stability and feasibility properties can be proven by considering the optimal input sequence at time k, $U^*(x(k))$ and the shifted sequence $\tilde{U}_N(x(k)) = [u_1^*(x(k))', \ldots, u_{N-1}^*(x(k))', (Kx_N^*(x(k)))']'$ at time $k+1$, where $x_N^*(x(k)))$ is the (predicted) solution of the system at time instant $k + N$, given the state $x(k)$ at time k and the input sequence $U^*(x(k))$. The shifted input sequence $\tilde{U}_N(x(k))$ is guaranteed to be feasible at time $k+1$, because $\mathcal{X}_I(K)$ is invariant and constraint-admissible. The value function $V_N^*(\cdot)$ in (3) serves as a Lyapunov function, due to the choice of the terminal weight P. The reader is referred to [7] for further details.

III. MOVE BLOCKING SCHEMES

A. Input Blocking (IB)

Since the computational complexity of solving the optimization problem (3) depends directly on the degrees of freedom (i.e. number of free inputs), it is standard practice

[8], [9] to obtain tractable optimization problem formulations by fixing the inputs to be constant over a certain number of time steps. For example, instead of solving for the optimal $U := [u_0', \ldots, u_{N-1}']' \in \mathbb{R}^{Nm}$, problem (3) can be restated in terms of solving for the optimal vector $\hat{U} := [\hat{u}_0', \ldots, \hat{u}_{M-1}']'$ such that $U = (T \otimes I_m)\hat{U}$, where $\otimes$ denotes the Kronecker product and $T \in \mathbb{R}^{N \times M}$ is a so-called *blocking matrix* with $M < N$. T is assumed to be a matrix of ones and zeros only, with each row containing exactly one non-zero element.

For example, for a SISO system with move blocking $u_1 = u_2 = u_3$ and $N = 4$ we might choose $\hat{U} = [\hat{u}_0' \ \hat{u}_1']'$ and

$$T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}. \tag{4}$$

The $N = 4$ degrees-of-freedom problem has thus been reduced to an $M = 2$ degrees-of-freedom problem.

The control problem is now reformulated as

$$V_N^*(x, T) := \min_{\hat{u}_0, \ldots, \hat{u}_{M-1}} \sum_{k=0}^{N-1} \left(u_k' R u_k + x_k' Q x_k \right)$$
$$+ x_N' P x_N, \tag{5a}$$
$$\text{subj. to} \ x_k \in \mathbb{X}, \ u_{k-1} \in \mathbb{U}, \ \forall k \in \{1, \ldots, N\}, \tag{5b}$$
$$x_N \in \mathcal{X}_I(K) \tag{5c}$$
$$U = [u_0', \ldots, u_{N-1}']' = (T \otimes I_m)[\hat{u}_0', \ldots, \hat{u}_{M-1}']',$$
$$x_{k+1} = Ax_k + Bu_k, \quad x_0 = x. \tag{5d}$$

Though the move-blocking reformulation may make the optimization problem computationally tractable, no stability or feasibility properties can be obtained. This is because a simple shift in the input sequence is no longer possible if the inputs are blocked, i.e. the input sequence obtained at time k may not be used to obtain a feasible solution at time $k + 1$ since the shifted sequence (as in Section II) will not correspond to the initial move blocking strategy imposed by T. Furthermore, there may not exist any input sequence $\hat{U}$ that satisfies the constraints in (3) for $U = (T \otimes I_m)\hat{U}$. Therefore, no guarantee of recursive feasibility can be provided, i.e. obtaining a feasible input sequence at time $k = 0$ does not imply that a feasible input sequence will be found for all future time steps. Furthermore, this move-blocking scheme may have a strong adverse effect on closed-loop performance. These drawbacks are illustrated in Section V.

B. Offset Blocking (OB)

The input blocking scheme described in the previous section can easily lead to very bad closed-loop performance since there is no guarantee that the applied input will be optimal for any feasible state $x \in \mathcal{X}_N$.

To some extent, this issue can be dealt with by 'pre-stabilizing' the system with the unconstrained, infinite-horizon LQR control law and parameterizing the input

sequence in terms of deviations from this control law, i.e. by letting $u_k = K_{\text{LQR}}x_k + c_k$ and formulating the associated optimal control problem in terms of $\mathcal{C} := [c_0', \ldots, c_{N-1}']'$, rather than U.

The infinite-horizon, unconstrained LQR law K_{LQR} is computed off-line and P is chosen to be the solution to the associated algebraic Riccati equation, i.e.

$$K_{\text{LQR}} = -(B^T P B + R)^{-1} B^T P A, \quad (6)$$

where P is the solution to:

$$P = A^T P A - A^T P B (B^T P B + R)^{-1} B^T P A + Q. \quad (7)$$

The optimal control problem is now restated in terms of $\hat{\mathcal{C}} = [\hat{c}_0', \ldots, \hat{c}_{N-1}']'$ and the constraints $\mathcal{C} = (T \otimes I_m)\hat{\mathcal{C}}$ and $u_k = K_{\text{LQR}}x_k + c_k$:

$$V_N^*(x, T) := \min_{\hat{c}_0, \ldots, \hat{c}_{M-1}} \sum_{k=0}^{N-1} \left(u_k' R u_k + x_k' Q x_k \right)$$
$$+ x_N' P x_N, \quad (8a)$$

$$\text{subj. to } x_k \in \mathbb{X}, \quad u_{k-1} \in \mathbb{U}, \quad \forall k \in \{1, \ldots, N\}, \quad (8b)$$

$$x_N \in \mathcal{X}_I(K_{\text{LQR}}), \quad (8c)$$

$$\mathcal{C} = [c_0', \ldots, c_{N-1}']' = (T \otimes I_m)[\hat{c}_0', \ldots, \hat{c}_{M-1}']',$$

$$u_k = K_{\text{LQR}}x_k + c_k, \quad \forall k \in \{0, \ldots, N-1\}, \quad (8d)$$

$$x_{k+1} = A x_k + B u_k, \quad x_0 = x. \quad (8e)$$

The advantage of this scheme is that the obtained input sequence will be optimal if $x \in \mathcal{X}_I(K_{\text{LQR}})$, since $\hat{\mathcal{C}} = [0', \ldots, 0']'$ will be feasible and optimal, regardless of the blocking strategy. Furthermore, only the offset to the feedback and not the actual input will be blocked, which may lead to greater flexibility in the controller. The computational advantages are maintained since the number of decision variables is less than Nm. Although this scheme may improve on performance, no guarantees on stability and constraint satisfaction can be given for the reasons stated in the previous section. Furthermore, in addition to the state constraints, there is the difficulty of satisfying input constraints, i.e. there may not exist any constant offset such that the input constraints are satisfied for a fixed number of steps. Therefore, parameterized move blocking may not always be preferable to standard move blocking.

C. Delta Blocking (DIB, DOB)

Instead of fixing the input or an offset to a feedback law to be constant for a fixed number of steps, it is possible to impose a constraint on the difference between two consecutive control actions and optimize over the size of this constraint. This corresponds to a linear interpolation of inputs $\hat{u}$ for the case of "delta-input blocking" (DIB). In DIB we set $u_k - u_{k+1}$ to be constant over a predefined number of time steps. The constraint on T also has to be relaxed to allow more than one non-zero element on each row.

Consider (4) for instance, where we have $N = 4$ and $M = 2$. The decision variables for the DIB problem

correspond to u_0, u_3 and the values for the 'blocked' inputs u_1, u_2 follow directly from linear interpolation. Thus the $N = 4$ degrees-of-freedom problem has been reduced to a $M = 2$ degrees-of-freedom problem.

The same kind of linear interpolation can be applied to obtain offset sequences from a small number of offsets $\hat{c}$. This scheme will be referred to as "delta-offset blocking" (DOB). In DOB we set $c_k - c_{k+1}$ to be constant over a predefined number of time steps, but optimize over the size of this constraint. As in the previously presented blocking schemes, both delta-blocking strategies reduce the computational burden but do not provide any guarantees on closed-loop stability or feasibility.

IV. MOVING WINDOW BLOCKING (MWB)

In this section we will describe a novel move-blocking scheme, which provides stability and feasibility guarantees in closed-loop. The scheme implements a time-dependent move blocking strategy where the blocked inputs are shifted at each time step. Specifically, the blocking matrix (denoted $T(k)$) is now time-dependent.

Definition 3: A matrix $T \in \mathbb{R}^{N \times M}$ is an *admissible blocking matrix* if $M < N$, one entry in each row of T is equal to 1 and all other entries in the row are equal to 0, there are no zero columns or zero rows and the elements of the matrix are arranged in an 'upper staircase' form, i.e. if the column in which a 1 occurs in the i'th row is

$$j^*(i) := \{j \mid T_{i,j} = 1\},$$

where $T_{i,j}$ denotes the element in the i'th row and j'th column of T, then $j^*(i+1) \geq j^*(i)$ for all $i \in \{1, \ldots, N-1\}$.

Examples of admissible blocking matrices are:

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}.$$

Examples of non-admissible blocking matrices are:

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}.$$

The sequence of admissible blocking matrices $\{T(0), T(1), \ldots, \}$, is generated by

$$T(k+1) = f(T(k)), \quad (9)$$

where $T(0)$ is an admissible blocking matrix and $f(\cdot)$ corresponds to the following algorithm:

Algorithm 1 (Blocking Matrix Generation $f(T)$):
Given an admissible blocking matrix $T \in \mathbb{R}^{N \times M}$.

1) Let $S := \begin{bmatrix} 0_{(N-1) \times 1} & I_{N-1} \end{bmatrix} T$ be the matrix of the last $N-1$ rows of T, where $S \in \mathbb{R}^{(N-1) \times N}$.
2) Note that the first column of S is, in general, not guaranteed to be non-zero. If the first column of S is

zero, then we shift the columns to the left and append the zero column to the right of the resulting matrix. In other words, let

$$W := \begin{cases} S & \text{if } S_{1,1} = 1 \\ S \begin{bmatrix} 0_{1\times(M-1)} & 0 \\ I_{M-1} & 0_{(M-1)\times 1} \end{bmatrix} & \text{if } S_{1,1} = 0 \end{cases}$$

3) Note that the first $M-1$ columns of W are non-zero and in 'upper staircase' form, but that the last column may be zero. We therefore return the admissible blocking matrix

$$f(T) := \begin{bmatrix} W \\ 0_{1\times(M-1)} & 1 \end{bmatrix},$$

where $f(T) \in \mathbb{R}^{N\times M}$.

As an example, the following sequence of admissible blocking matrices were generated by the function $f(\cdot)$:

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

In order to obtain stability and feasibility guarantees, we propose to parameterize the input as in Section III-B, i.e. P is the solution to the ARE (7) and $u_k = K_{\text{LQR}}x_k + c_k$. We also propose to replace the terminal constraint (8c) with a set of constraints that is dependent on the current blocking matrix $T = T(k)$. Specifically, the following problem is solved at each time instant k, given the current state $x = x(k)$ and current blocking matrix $T = T(k)$:

$$V_N^*(x, T) := \min_{\hat{c}_0,\dots,\hat{c}_{M-1}} \sum_{k=0}^{N-1} \left(u_k'Ru_k + x_k'Qx_k \right)$$
$$+ x_N'Px_N, \qquad (10a)$$
$$\text{subj. to } x_k \in \mathbb{X}, \quad u_{k-1} \in \mathbb{U}, \; \forall k \in \{1,\dots,N\}, \quad (10b)$$
$$x_\ell \in \mathcal{X}_I(K_{\text{LQR}}), \; \ell = p(T), \qquad (10c)$$
$$[c_0',\dots,c_{\ell-1}']' = (([I_\ell, 0_{\ell\times(N-\ell)}]T) \otimes I_m)[\hat{c}_0',\dots,\hat{c}_{M-1}']', \qquad (10d)$$
$$c_k = 0, \; \forall k \in \{\ell,\dots,N-1\}, \qquad (10e)$$
$$u_k = K_{\text{LQR}}x_k + c_k, \quad \forall k \in \{0,\dots,N-1\}, $$
$$x_{k+1} = Ax_k + Bu_k, \quad x_0 = x. \qquad (10f)$$

where $p(T)$ in (10c) is the row of the first non-zero element in the last column of T, i.e.

$$p(T) := \min_{i\in\{1,\dots,N\}} \{i \,|\, T_{i,M} = 1\}. \qquad (11)$$

The key part to guaranteeing feasibility and stability is the choice of constraints (10c)–(10e). Clearly, if $p(T) = N$, then (10e) can be removed and (10d) becomes

$$[c_0',\dots,c_{N-1}']' = (T \otimes I_m)[\hat{c}_0',\dots,\hat{c}_{M-1}']'.$$

If we denote the optimizer to the above problem as

$$\hat{\mathcal{C}}^*(x,T) := [\hat{c}_0^*(x,T)',\dots,\hat{c}_{M-1}^*(x,T)']'$$

and

$$\mathcal{C}^*(x,T) := [c_0^*(x,T)',\dots,c_{N-1}^*(x,T)']'$$
$$:= (T \otimes I_m)\hat{\mathcal{C}}^*(x,T)$$

then the RHC law is given by

$$u(k) = K_{\text{LQR}}x(k) + c_0^*(x(k),T(k))$$

and the closed-loop system is

$$x(k+1) = (A + BK_{\text{LQR}})x(k) + Bc_0^*(x(k),T(k)) \quad (12)$$

with

$$T(k+1) = f(T(k)).$$

Theorem 2: If the MWB scheme is applied as above and $T(0)$ is an admissible blocking matrix, then the origin of the closed-loop system (12) is an asymptotically stable equilibrium with a region of attraction equal to the set of initial states $x(0)$ for which a solution to the optimization problem in (10) exists.

Proof: The proof is based on using the standard method [7] of considering, at time $k+1$, a shifted version of the optimal perturbation sequence found at time k:

$$\tilde{\mathcal{C}}^*(x(k),T(k)) = [c_1^*(x(k),T(k))',\dots,c_{N-1}^*(x(k),T(k))',0]'.$$

If, at time $k+1$, there exists a $\hat{\mathcal{C}}$ such that the constraints in (10) are satisfied with $\mathcal{C} = \tilde{\mathcal{C}}^*(x(k),T(k))$, then it is easy to show that the value function at time $k+1$ is less than the value function at time k for all $x(k) \neq 0$, i.e. $V_N^*(x(k+1),T(k+1)) < V_N^*(x(k),T(k))$. This allows one to use $V_N^*(\cdot)$ as a Lyapunov function for the close-loop system.

The key point to recognize is that, because $\mathcal{X}_I(K_{\text{LQR}})$ is invariant under the control law $u = K_{\text{LQR}}x$, the (optimal) predicted terminal state $x_N^*(x(k),T(k))$ at time k is in $\mathcal{X}_I(K_{\text{LQR}})$, regardless of the value of $p(T(k))$. This implies that there always exists a $\hat{\mathcal{C}}$ such that $\mathcal{C} = \tilde{\mathcal{C}}^*(x(k),T(k))$ is feasible at time $k+1$. We distinguish between two distinct cases, where the proofs of feasibility differ in subtle, but important ways:

(i) $p(T(k+1)) = N$ implies that $c_{N-1} = \hat{c}_{M-1}$ can be chosen independently of $c_{N-2} = \hat{c}_{M-2}$. It also implies that the $(2,1)$ element of $T(k)$ is zero, i.e. the first column of S (the matrix in the definition of $f(T(k))$) is zero. This implies that the first row *and* column of $T(k)$ was removed when computing $T(k+1)$ and hence a *shifted* version of $\hat{\mathcal{C}}^*(x(k),T(k))$ is feasible at time $k+1$. In other words, it is easy to verify that

$$\hat{\mathcal{C}} = [\hat{c}_1^*(x(k),T(k))',\dots,\hat{c}_{M-1}^*(x(k),T(k))',0]'$$

results in $\mathcal{C} = \tilde{\mathcal{C}}^*(x(k),T(k))$ at time $k+1$.

(ii) If $1 \leq p(T(k+1)) \leq N-1$, then we cannot use the same arguments as for $p(T(k+1)) = N$; the shifted version of $\hat{\mathcal{C}}^*(x(k),T(k))$ cannot be guaranteed to be feasible. However, note that $p(T(k+1)) = p(T(k))-1$.

It is now easy to verify from (10d)–(10e) that a *non-shifted* version of $\hat{\mathcal{C}}^*(x(k), T(k))$, i.e.

$$\hat{\mathcal{C}} = [\hat{c}_0^*(x(k), T(k))', \ldots, \hat{c}_{M-1}^*(x(k), T(k))']'$$

results in $\mathcal{C} = \tilde{\mathcal{C}}^*(x(k), T(k))$ at time $k+1$. Note that this corresponds to cropping the number of free offsets (see (10e)) at time $k+1$. ∎

Remark 1: Note that for all states $x \in \mathcal{X}_I(K_{\text{LQR}})$ the input $u_k = K_{\text{LQR}} x_k$ will be feasible, i.e. $\hat{\mathcal{C}} = 0$ will be a feasible and optimal solution.

Remark 2: Note that Theorem 2 does not hold if, as in Section III-A, $\hat{U}$ is chosen as the optimizer. For example, there may not be a constant sequence of inputs which keep the state within $\mathcal{X}_I(K)$ such that recursive feasibility can be proven.

V. NUMERICAL EXAMPLES

In this section, we will compare the various move blocking schemes of Section III as well as the moving window blocking strategy in Section IV with the optimal full degree of freedom (FDOF) controller.

A. Problem Formulation

Specifically, 10 random open-loop stable systems and 10 random open-loop unstable systems were generated. All systems have $n = 2$ states and $m = 1$ inputs. Subsequently, we computed the mp-QP [2] solution to (3) for $N = 10$ (FDOF) as well as the mp-QP solutions to the input blocking (IB), offset-blocking (OB), delta-input blocking (DIB), delta-offset blocking (DOB) and moving window blocking (MWB) strategies for $N = 10$, $M = 2$.

We will use the notation $[i \ j]$ to denote that the first i elements and the last j elements of the optimizer are fixed to be constant ($u_k = u_{k+1}, \forall k \in \{0, \ldots, i-1\}$ and $\forall k \in \{i, \ldots, j-1\}$) in IB, OB and MWB blocking schemes. Although we considered all possible combinations $[i \ j]$, only a small subset will be shown here due to space constraints. For delta blocking strategies u_0, u_9 and c_0, c_9, are the decision variables for DIB and DOB, respectively, denoted by $[0 \ 9]$.

MWB is initialized such that $T(0)$ is equal to T of IB and OB. Note that the scheme in Section IV may thus force the initial terminal set constraint to be too restrictive, i.e. $p(T(0))$ may be smaller than necessary. In order to avoid this issue, the MWB controller is initialized with a 'warm start' as described in the following. For example, starting with $T(0) = [1 \ 9]$ the constraint (10c) would set $l = 2$, i.e. our optimization problem would be solved with a prediction horizon of $N = 2$. Instead, we here solve the problem at the first time step $k = 0$ for the full horizon $N = 10$, $l = 10$ with blocking matrix $T = [1 \ 9]$. At the next time step $k = 1$, the MWB scheme is applied with $T(1) = [9 \ 1]$. Feasibility and stability from $k = 0$ to $k = 1$ can be trivially guaranteed by considering a shifted version of the input sequence $k = 0$

at time step $k = 1$. A similar 'warm start' was applied for the other blocking matrices $T(0)$.

The control objective (e.g. (3a)) was set to $Q = I$, $R = 1$ and the terminal weight P as the solution to the ARE (7). The partition of the full degree of freedom controller with $x_N \in \mathcal{X}_I(K_{\text{LQR}})$ is invariant by design. Therefore constraint satisfaction is guaranteed for all time and all states $x(k) \in \mathcal{X}_N$. Since this does not hold for IB, OB, DIB and DOB we computed the invariant subset of the move blocking partitions with the method in [5]. In a second step, the invariant set was gridded to obtain at least 50 feasible initial states and the corresponding closed-loop performance of the various controllers was computed and compared.

B. Results

Table I compares the closed-loop performance and Table II the volume of the controllable state space of the move blocking schemes versus the FDOF optimal controller. The results present the ratio to the FDOF case in percentages. The abbreviations *min*, *avg* and *max* denote the minimum, average and maximum values, respectively. The abbreviation (*n.c.*) indicates the number of systems that were not controllable with the proposed move blocking scheme.

For stable systems offset blocking strategies OB, DOB, and MWB yield slightly better performance than input blocking strategies IB and DIB (Table I). But the controllable state space obtained with input blocking strategies is significantly larger than that obtained with offset blocking strategies (Table II). This is mostly because, in general, no constant offset $\hat{c}$ exists, which satisfies the input constraints for those initial states that are far away from the origin.

For unstable systems input blocking strategies are inferior versus offset blocking strategies in terms of performance. Closed-loop performance can be up to 50 times worse than that obtained with an optimal controller (FDOF). In some cases it is not possible to control the system with input blocking strategies at all.

Note the difference in the controllable systems between the three IB schemes $[1 \ 9]$, $[2 \ 8]$ and $[3 \ 7]$. The results suggest that an uneven number of blocked moves is more likely to yield the desired closed-loop behavior for unstable systems. This was reaffirmed in a detailed study of additional systems where it was observed that this behavior occurs for systems with unstable negative poles. For these types of systems, an uneven number of blocked moves generally leads to more aggressive controller action on the first input, which in turn increases the invariant set. The controller is more aggressive on the first step because the states on 'one end' of the unstable oscillation are weighed more heavily in the cost function. MWB and OB give similar results in terms of performance and volume of the controllable state space, but it cannot be stressed enough that only the moving window strategy guarantees closed-loop stability and recursive feasibility. Furthermore, the results show that for certain systems a moving-window

	stable systems		unstable systems		
	avg (%)	max (%)	avg (%)	max (%)	n.c.
IB T[1 9]	0.8	5.9	510.2	1982.4	-
IB T[2 8]	3.7	11.1	832.2	4691.3	3
IB T[3 7]	2.0	12.1	220.4	1121.5	2
DIB [0 9]	2.2	14.8	91.9	319.3	1
OB T[1 9]	0.7	3.2	2.3	11.9	-
OB T[2 8]	0.3	2.0	1.0	5.1	-
OB T[3 7]	0.7	5.7	1.8	6.3	-
DOB [0 9]	1.1	8.9	4.2	13.3	-
MWB T[1 9]	1.2	7.2	1.9	5.1	-
MWB T[2 8]	0.4	2.0	1.7	4.8	-
MWB T[3 7]	0.7	5.7	1.7	5.7	-

TABLE I

COMPARISON OF MOVE BLOCKING SCHEMES VERSUS THE OPTIMAL CONTROLLER: CLOSED-LOOP PERFORMANCE DECAY (E.G. 0% EQUALS OPT. CTRL.). $n.c.$ DENOTES THE NUMBER OF SYSTEMS THAT WERE NOT CONTROLLABLE WITH THE RESPECTIVE MOVE BLOCKING SCHEME.

	stable systems		unstable systems		
	avg (%)	max (%)	avg (%)	max (%)	n.c.
IB T[1 9]	89.4	46.7	46.3	4.2	-
IB T[2 8]	85.8	59.6	44.9	2.7	3
IB T[3 7]	87.7	63.6	50.7	19.0	2
DIB [0 9]	89.1	62.7	40.6	0.6	1
OB T[1 9]	59.3	21.9	56.5	13.5	-
OB T[2 8]	29.1	16.9	44.2	9.8	-
OB T[3 7]	25.9	12.9	43.7	10.4	-
DOB [0 9]	30.9	20.4	50.2	11.8	-
MWB T[1 9]	59.3	22.0	56.5	13.5	-
MWB T[2 8]	30.4	20.5	45.9	10.0	-
MWB T[3 7]	26.0	12.9	43.8	10.4	-
$\mathcal{X}_I(K_{\mathrm{LQR}})$	12.5	2.6	24.2	4.2	-

TABLE II

COMPARISON OF MOVE BLOCKING SCHEMES AND MAXIMUM INVARIANT SET VERSUS THE OPTIMAL CONTROLLER: VOLUME OF THE CONTROLLABLE STATE SPACE. $n.c.$ DENOTES THE NUMBER OF SYSTEMS THAT WERE NOT CONTROLLABLE WITH THE RESPECTIVE MOVE BLOCKING SCHEME.

strategy may yield excellent performance for a large set of states, together with stability and feasibility guarantees.

The examples presented in this section are only a small subset of all the cases examined. Different cost objectives, system sizes and blocking combinations $[i\ j]$ were considered. The heuristic insights gained in the case study are presented in the following section.

All results were created with the MPT Toolbox [6].

VI. CONCLUSION

This paper pointed out the theoretical and practical shortcomings of various move-blocking strategies. In addition, a novel move blocking scheme, termed 'Moving Window Blocking' (MWB), was introduced. The MWB strategy provides stability and feasibility guarantees. An extensive case study was also presented which shows the advantages and drawbacks of the different strategies.

From the large number of examined systems we were able to deduce the following heuristic insights:

- For stable systems, the IB strategy tends to give better results with respect to performance *and* size of the controllable state space than any of the other strategies which were considered.
- For unstable systems, the offset blocking strategies (e.g. OB, DOB and MWB) give better performance results than input blocking schemes (e.g. IB, DIB) and are always able to stabilize a subset of the controllable state space.
- If the controllable state space is large in volume, then strategies using offset blocking (e.g. OB, MWB, DOB) are generally not able to obtain feasible control sequences for states which are far from the origin. For stable systems the controllable state-space is therefore significantly smaller than that obtained with input blocking schemes.
- Systems with negative unstable poles are more difficult to control with IB when choosing an even number of blocked moves for the first blocking sequence in the prediction horizon.
- MWB is the only blocking scheme that provides guarantees on stability and constraint satisfaction.

REFERENCES

[1] A. Bemporad and M. Morari. Robust model predictive control: A survey. In *Robustness in Identification and Control*, number 245 in LNCS, pages 207–226. 1999.

[2] A. Bemporad, M. Morari, V. Dua, and E.N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3–20, January 2002.

[3] E. G. Gilbert and K. T. Tan. Linear systems with state and control constraints: the theory and applications of maximal output admissible sets. *IEEE Trans. Automatic Control*, 36(9):1008–1020, 1991.

[4] P. Grieder, F. Borrelli, F.D. Torrisi, and M. Morari. Computation of the constrained infinite time linear quadratic regulator. *Automatica*, 40(4):701–708, April 2004.

[5] P. Grieder, M. Lüthi, P. Parillo, and M. Morari. Stability & feasibility of receding horizon control. In *European Control Conference*, Cambridge, UK, September 2003.

[6] M. Kvasnica, P. Grieder, M. Baotić, and M. Morari. Multi Parametric Toolbox (MPT). In *Hybrid Systems: Computation and Control*, Lecture Notes in Computer Science. Springer Verlag, 2003. `http://control.ee.ethz.ch/˜mpt`.

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

[8] S.J. Qin and T.A. Badgwell. An overview of industrial model predictive control technology. In *Chemical Process Control - V*, volume 93, no. 316, pages 232–256. AIChe Symposium Series - American Institute of Chemical Engineers, 1997.

[9] P. Tøndel and T.A. Johansen. Complexity reduction in explicit model predictive control. In *IFAC World Congress*, Spain, Barcelona, 2002, 2002.

[10] P. Tøndel, T.A. Johansen, and A. Bemporad. An algorithm for multi-parametric quadratic programming and explicit MPC solutions. *Automatica*, 39(3):489–497, March 2003.