Optimal move blocking strategies for model predictive control

Rohan C. Shekhar¹, Chris Manzie

Department of Mechanical Engineering, The University of Melbourne, Victoria 3010, Australia

ARTICLE INFO

Article history:
Received 18 November 2014
Received in revised form 5 March 2015
Accepted 20 July 2015
Available online 13 August 2015

Keywords:
Move blocking
Model predictive control
Complexity reduction
Parametric optimisation

ABSTRACT

This paper presents a systematic methodology for designing move blocking strategies to reduce the complexity of a model predictive controller for linear systems, with explicit optimisation of the blocking structure using mixed-integer programming. Given a move-blocked predictive controller with a terminal invariant set constraint for stability, combined with an input parameterisation to preserve recursive feasibility, two different optimisation problems are formulated for blocking structure selection. The first problem calculates the maximum achievable reduction in the number of input decision variables and prediction horizon length, subject to the controller's region of attraction containing a specified subset of the state space. Then, for a given fixed horizon length and block count determined by hardware capabilities, the second problem seeks to maximise the volume of an inner approximation to the region of attraction. Numerical examples show that the resulting blocking structures are able to optimally reduce controller complexity and improve region of attraction volume.

© 2015 Elsevier Ltd. All rights reserved.

1. Introduction

Model Predictive Control (MPC) is an automation paradigm that has been successfully applied to industrial control problems for a number of years (Maciejowski, 2002), owing to its intuitive formulation and unique constraint handling capability. At each time step, MPC aims to find a sequence of inputs to optimise a cost function over a finite time horizon, whilst satisfying operating constraints. With an appropriate selection of the cost function and constraints, recursive feasibility of the optimisation problem and subsequent convergence of the system states can be guaranteed (Mayne, Rawlings, Rao, & Scokaert, 2000).

Although advances in computational power have improved tractability of the MPC optimisation problem for an increasingly large number of systems, a long prediction horizon can prevent real-time MPC implementation, limiting its utility for systems with fast dynamics or high sampling rate requirements. Explicit MPC (Bemporad, Morari, Dua, & Pistikopoulos, 2002) seems to offer a solution by computing the MPC control law offline, but this computation becomes intractable for long horizon lengths and large numbers of constraints (Ferreau, Bock, & Diehl, 2008). Given this challenge, a number of alternative methods have been proposed to simplify the online complexity of MPC optimisation directly. Some approaches exploit the structure of the MPC problem to develop fast optimisation algorithms (Ferreau et al., 2008; Nedelcu, Necoara, & Tran-Dinh, 2014; Patrinos & Bemporad, 2014; Wang & Boyd, 2010), whilst others modify the MPC problem itself to reduce complexity at the expense of optimality. One method of achieving the latter is to assume some form of input parameterisation, curtailing the number of degrees of freedom in the online optimisation problem. Various candidate parameterisations have been proposed, including move blocking (Cagienard, Grieder, Kerrigan, & Morari, 2007; Maciejowski, 2002), linear subspaces (Goebel & Allgöwer, 2014; Ong & Wang, 2014) and Laguerre polynomials (Rossiter & Wang, 2008).

Move blocking is a candidate parameterisation that constrains groups of adjacent-in-time predicted inputs to have the same value. These groups are denoted as blocks, from where this parameterisation gets its name. Its advantages over other parameterisation methods include the parameterised input values retaining the same physical meaning as the original inputs, as well as the hardware implementation being straightforward. Whilst move blocking can reduce complexity, providing recursive feasibility guarantees is more challenging, since a “shifted” version of the previously optimal input sequence may no longer be admissible with respect to the blocking structure. In addition, blocking also affects...
the size of the controller’s Region of Attraction (ROA). With a recursively feasible controller, the ROA corresponds to the set of states for which an initial feasible solution to the MPC optimisation problem exists.

Different move blocking approaches have been presented in order to maintain recursive feasibility guarantees and attempt to preserve the controller’s ROA. Ghodhaletkar and Imura (2010) introduce an approach based upon calculating an invariant set that is also blocking admissible. Additionally, the approach recovers the ROA of the original unblocked controller by relaxing constraints at future prediction steps. However, this prevents a terminal constraint from being enforced, so explicit guarantees of closed-loop convergence cannot be given. Cagienard et al. (2007) and Shekhar and Maciejowski (2012b) present approaches that utilise time-varying blocking structures, where the changing structure allows a shifted version of the previously optimal input sequence to be feasible at the following time step. Guarantees of both recursive feasibility and closed-loop convergence can then be provided, with an appropriately designed terminal constraint and cost function. However, the ROA of the controller is necessarily reduced by this constraint, making blocking structure selection an important design consideration.

Blocking structure design can be approached from the perspective of satisfying one of two specifications: either a requirement on minimum ROA size, or a fixed complexity requirement specified in terms of the number of blocks (or block count) and the horizon length. If the ROA is required to contain a given subset of the feasible state space, then the amount by which complexity can be reduced through blocking is restricted. In such cases, reducing the number of blocks and the horizon length by the greatest extent possible whilst ensuring the ROA contains this subset allows for faster computation, reduces the hardware footprint and provides opportunities for performance improvements through parallelisation (Longo, Kerrigan, Ling, & Constantinides, 2011).

Conversely, if a controller is to be designed to utilise the maximum capability of a given hardware platform, then an effective limit is placed on the horizon length and number of blocks for a fixed sampling rate. In this scenario, it is desirable to choose a blocking structure that possesses the largest ROA size for this horizon length and block count, especially for regulation problems. This allows the controller to regulate from a large set of initial states whilst making maximum use of hardware capabilities. However there are currently no systematic methodologies to select blocking structures for either of these design objectives.

This paper addresses this research gap by presenting methods for optimally selecting blocking structures in a computationally tractable manner. It first details how a move-blocked predictive controller is formulated, with terminal constraints used to guarantee convergence and a specific input parameterisation that ensures recursive feasibility, similar to that used by Ong and Wang (2014). This parameterisation allows the blocking structure to remain time invariant, by adding to the input an appropriately shifted scalar multiple of the previously optimal input sequence. Two different optimisation problems are then formulated for selecting blocking structures, which form the primary contribution of this paper:

1. minimising the number of input blocks and the horizon length, whilst ensuring that the ROA contains a desired subset of the state space; and
2. maximising the volume of an ellipsoidal inner approximation to the ROA, for a specified horizon length and number of blocks.

Finally, numerical examples illustrate the approaches on simple two-dimensional systems, which allow ROAs to be easily visualised and their volumes explicitly computed.

1.1. Nomenclature

The sets of integers and real numbers are denoted $\mathbb{Z}$ and $\mathbb{R}$ respectively. $[a, b]$ denotes the set of numbers $\{a, a + 1, \ldots, b\}$, where $a, b \in \mathbb{Z}$, $a \leq b$. The set of all non-negative integers is denoted $\mathbb{Z}_{\geq 0}$. The vertical concatenation $[X_1^T, X_2^T, \ldots, X_p^T]^T$ is written as $[X_1; X_2; \ldots; X_p]$. $I_n$ represents the $n \times n$ identity matrix. $\mathbf{0}$ represents a vector of zeros having length $n$, whereas $\mathbf{1}$ is a block matrix of zeros with dimensions to be inferred from context within a larger block matrix. $\mathbf{1}_n$ represents a vector of ones having length $n$, whereas $\mathbf{1}_{n \times m}$ denotes an $n \times m$ matrix of ones. $\bar{z}_{jk}$ denotes a prediction of signal $z(k)$ made $j$-steps in to the future from the current time $k$. The operator $\otimes$ denotes the Kronecker product. The operator $\oplus$ represents the direct sum. The operator $\text{co} \{ \cdot \}$ denotes the convex hull. The volume of a set is denoted vol$(\cdot)$, $\text{vol}(\cdot)$ denotes a projection onto $\mathbb{R}^p$, $\| \cdot \|_p$ denotes the $p$-norm, whereas $| \cdot |$ denotes the element-wise absolute value. The operator $\lor$ denotes an element-wise logical disjunction.

2. Problem formulation

Consider the discrete-time linear system

\[ x(k + 1) = Ax(k) + Bu(k), \]

where $x(k) \in \mathbb{R}^n$ and $u(k) \in \mathbb{R}^m$. It is assumed that $(A, B)$ is controllable. Defining an output

\[ y(k) := Cx(k) + Du(k) \in \mathbb{R}^p, \]

the system is subject to constraints at each sampling instant of the form

\[ y(k) \in \mathcal{Y} := \{ y \mid Ey \leq f \}, \]

for some $E \in \mathbb{R}^{n \times p}$ and $f \in \mathbb{R}^p$ that define polytope $\mathcal{Y}$. The control objective is to asymptotically steer the state of the system to the origin from the initial state $x(0)$, whilst satisfying the constraints (3) at all times. The output equation (2) is chosen by the control designer to incorporate all state, input and cross-constraints.

A model predictive controller is specified to steer the state of the system (1) to the origin, whilst minimising a given finite-horizon cost function, subject to the constraints (3). At each iteration, the MPC optimisation problem takes the form

\[ J_n^p(x(k)) := \min_{\bar{u}(k)} \{ J_n(x(k), \bar{u}(k), y(k)) \}, \]

subject to

\[ \bar{x}_{0jk} = x(k), \]

\[ \bar{x}_{j+1:k} = A \bar{x}_{j:k} + B \bar{u}_{j:k}, \]

\[ \bar{y}_{j:k} = C \bar{x}_{j:k} + D \bar{u}_{j:k}, \]

\[ \bar{y}_{j:k} \in \mathcal{Y}, \]

\[ \bar{x}_{0jk} \in \mathcal{T}, \]

for all $j \in \{0, N-1\}$, given prediction horizon $N$, input prediction variables $\bar{u}(k) := \left[ \bar{u}_{0jk}; \bar{u}_{1jk}; \ldots; \bar{u}_{N-1jk} \right]$, state prediction variables $\bar{x}_{0jk}, \bar{x}_{1jk}, \ldots, \bar{x}_{Njk}$, output predictions $\bar{y}_{0jk}, \bar{y}_{1jk}, \ldots, \bar{y}_{Njk}$ and cost function $J_n(x(k), \bar{u})$. For stability purposes, a terminal polytopic constraint-admissible control-invariant set

\[ \mathcal{T} := \{ x \mid Gx \leq h \} \]

is defined, for some $G \in \mathbb{R}^{l \times n}$ and $h \in \mathbb{R}^l$. The invariance of this set implies that for all $x \in \mathcal{T}$, there must exist a $\mu$ such that

\[ A x + B \mu \in \mathcal{T}, \]

\[ C x + D \mu \in \mathcal{Y}. \]
Thus, invarient sets can be calculated using iterative methods (Kerrigan, 2000; Rakovic, Kerrigan, Kouramas, & Mayne, 2005).

An optimal solution of (4) subject to (5) will be denoted
$$\tilde{u}_h(x(k)) := \left[ \tilde{u}_{0|k}^1; \tilde{u}_{0|k}^2; \cdots; \tilde{u}_{0|k}^{N-1} \right],$$
which will produce an optimal predicted state sequence $x_{0|k}^1, x_{0|k}^2, \ldots, x_{0|k}^{N-1}$. Applying the control law $u(k) = \tilde{u}_{0|k}$ at each time step, an appropriate choice of cost function $J_R(\cdot, \cdot)$ guarantees asymptotic convergence of the state to the origin from $x(0)$, provided an initial feasible solution exists (Rawlings & Mayne, 2009).

3. Move blocked MPC

Move blocking with recursive feasibility guarantees can be implemented by solving the modified optimisation problem
$$\min_{\mathbf{v}(k), \lambda, \mu} \sum_{j=0}^{N-1} J_R(\tilde{u}(k), x(k)), \quad (7)$$
subject to the constraints (5) and the input parameterisation
$$\tilde{u}(k) = (S \otimes I_m) \mathbf{v}(k) + [\lambda \tilde{u}(k); \mu], \quad (8)$$
where
$$\mathbf{v}(k) := [v_1; v_2; \cdots; v_r] \in \mathbb{R}^m$$
$$\tilde{u}(0) := 0_{(N-1)m}$$
$$\tilde{u}(k+1) := [u_{1|k}^1; u_{2|k}^1; \cdots; u_{N-1|k}^1].$$

In this parameterisation, $S \in \{0, 1\}^{N \times r}$ is a blocking matrix, $\lambda \in \mathbb{R}$ and $\mu \in \mathbb{R}^m$ and $u_{0|k}^1, u_{0|k}^2, \ldots, u_{0|k}^{N-1}$ represents the optimal input sequence to the blocked problem (7), with a corresponding state sequence denoted by $x_{0|k}^1, x_{0|k}^2, \ldots, x_{0|k}^{N-1}$. As described in Shekhar and Maciejowski (2012b), $S$ can be decomposed as
$$S = \bigoplus_{q=1}^r S_q,$$
where
$$\sum_{q=1}^r S_q = r. \quad (10)$$

Thus, $S$ subdivides $\tilde{u}$ into $r$ blocks, with the $q$th column of $S$ constraining a block of $b_q$ contiguous inputs to have the same value. The second term of the parameterisation $\lambda \tilde{u}(k)$ is added for recursive feasibility, in a similar way to Ong and Wang (2014). It is essentially a scalar multiple of the shifted optimal blocked input sequence from the previous time step. The “tail” input $\mu$ extends the shifted input sequence to have length $N$, and is chosen to satisfy (6). Such an input is guaranteed to exist, since $x_{0|k}^r \in \mathcal{T}$. If the terminal invariant set has been designed with respect to some linear controller $u = Kx$, then $\mu = Kx_{0|k}^r$ is guaranteed to be admissible and keep the state within $\mathcal{T}$. The ROA of the parameterised controller is the set of states for which there exists a feasible blocked input sequence with block values $\mathbf{v}$ and structure given by $S$.

This parameterisation reduces the number of input decision variables in (4) from $Nm$ to $(r + 1)m + 1$, given the extra decision variables $\lambda$ and $\mu$. With a terminal linear controller, this further reduces to $m + 1$ variables. The extra decision variables ensure that the shifted input sequence from the previous time step, known to be feasible, can always be chosen by the optimiser.

**Theorem 1 (Recursive Feasibility).** Denoting the optimal solution to (7) for $\mu$ as $\mu^*$, a feasible solution at time $k + 1$ is given by
$$\mathbf{v}(k+1) = 0_m, \quad (11)$$
$$\lambda = 1, \quad (12)$$
$$\mu = \mu^*. \quad (13)$$

**Proof.** Choosing $\mathbf{v}$, $\lambda$ and $\mu$ in this manner gives the input sequence
$$\tilde{u}(k+1) = [u_{1|k}^1; u_{2|k}^1; \cdots; u_{N-1|k}^1; \mu^*], \quad (14)$$
which results in the predicted state sequence
$$x_{j|k+1} = x_{j|k}^r, \quad \forall j \in \{0, N-1\}$$
$$x_{N|k+1} = A x_{0|k}^r + B x^r.$$

It is clear that the first $N - 1$ predicted inputs and states from time $k + 1$ satisfy (6), since they are simply shifted from the previously optimal sequence. Additionally, since $x_{0|k}^r \in \mathcal{T}$, it follows that the optimiser can choose $\mu^*$ such that $x_{0|k+1}^r \in \mathcal{T}$, from the control invariance of $\mathcal{T}$ (6).

**Remark 2.** The input sequence $\tilde{u}(k)$ will not necessarily exhibit the blocking structure specified by $S$ if $\lambda \neq 0$. This is not an issue, since blocking is only a computational construct for reducing input complexity. Indeed, the key to recursive feasibility is allowing a “shifted” version of the previously optimal input sequence to be a feasible solution to (7) at the current time step.

**Corollary 3 (Closed-loop Asymptotic Stability).** For any $x(0)$ in the ROA for the blocked controller, asymptotic convergence of the system state is guaranteed, provided that the value function $J_R^*(\cdot)$ is a Lyapunov function.

**Remark 4.** In a similar manner to the unblocked MPC controller, the properties specified in Rawlings and Mayne (2009) can be used to design a cost function such that $J_R^*(\cdot)$ is a Lyapunov function.

It is possible to recover the ROA for the original controller if the unblocked MPC problem is solved only at the first time step and $\tilde{u}$ is initialised to the corresponding optimal input sequence. However, this must be repeated for every initial condition, and may not be computationally tractable for large horizons. If blocking is applied from the initial state onwards, then the ROA will necessarily be reduced in the presence of terminal constraints. The next section describes two different ways in which the blocking structure can be optimised in terms of the ROA.

4. Optimal blocking

Two different methods for optimal selection of blocking structures with respect to the ROA will now be described. In order to do this, the ROA of the controller will be explicitly parameterised, using a mixed-integer representation of the blocking structure. It will then be shown how a blocking structure can be selected to minimise both the number of blocks and the overall prediction horizon length, with the requirement of the ROA containing a given subset of the state space. Subsequently, a different optimisation problem will be formulated to find a blocking structure that maximises the volume of an inner approximation to the ROA, if a desired horizon length and block count are specified.

Note that neither of these optimisation approaches consider cost optimality, since such comparisons are typically based upon open-loop costs, which could significantly differ from the closed-loop cost. Additionally, for the second optimisation approach, it is difficult to fairly compare costs over different sized ROAs. In any case, the cost function of a predictive controller is often just a tuning parameter used to obtain stability guarantees in the closed loop, since the stability requirements stated in Rawlings and Mayne (2009) do not allow the cost to be arbitrarily selected.

4.1. ROA parameterisation

In order to explicitly characterise the closed-loop ROA, define the matrices

\[ \Theta^1_N := \begin{bmatrix} EC; ECA; ECA^2; \ldots; ECA^{N-1} \end{bmatrix} \]
\[ \Theta^2_N := \begin{bmatrix} ED & 0 & 0 & \ldots & 0 \\ ECB & ED & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ ECA^{N-2}B & ECA^{N-3}B & ECA^{N-4}B & \ldots & ED \end{bmatrix} \]
\[ \phi_N := 1_N \otimes f \]
\[ \Gamma^1_N := GA^N \]
\[ \Gamma^2_N := [GA^{N-1}B, GA^{N-2}B, \ldots, GAB, GB]. \]

These matrices arise from condensing the prediction model dynamics and applying the output constraints, as in Shekhar and Maciejowski (2012a). Then, define the set of all admissible blocked input and state pairs corresponding to a blocking matrix \( S \) as
\[
Q_N(S) := \{ (x, v) \in \mathbb{R}^{n+m} \mid \Theta^1_N x + \Theta^2_N S v \leq \phi_N, \Gamma^1_N x + \Gamma^2_N S v \leq h \}. \tag{15} \]

The ROA is then given by the projection
\[
\mathcal{R}_N(S) := \mathbb{P}_u(Q_N(S)). \tag{16} \]

This defines the set of all states for which there exists an admissible sequence of blocked inputs satisfying the control objective. The unblocked ROA is trivially recovered by taking \( S = I_{nm} \).

This definition of the ROA is not suitable for convex optimisation, since the blocking matrix enters in a bilinear manner. For use in an optimisation problem, \( S \) can be mapped to the vector
\[
\delta_N := [\delta_0; \delta_1; \ldots; \delta_{N-1}] \in [0, 1]^N
:= \{ 0_{b_{01}-1}; 1; 0_{b_{02}-1}; 1; \ldots; 0_{b_{N-1}-1}; 1 \},
\]

using the decomposition of \( S \) given by (9) and (10). This is a vector with unity denoting the end of each block, and zeros elsewhere. It can be shown that there is a bijective mapping between \( S \) and \( \delta_N \). Then, this representation of the blocking matrix requires the implications
\[
\delta_i = 0 \implies u_{i+1|k} = u_{ijk}, \quad \forall j \in \mathbb{Z}_{0, N-2} \tag{17}
\]
to be enforced on the predicted input sequence. Using a so-called big-M formulation (Bemporad & Morari, 1999), these implications can be represented by the mixed-integer constraints
\[
-M\delta_i \leq u_{i+1|k} - u_{ijk} \leq M\delta_i, \tag{18}
\]
where \( M > 0 \) is some sufficiently large number. It can be seen that when \( \delta_i = 0 \), equality is enforced on inputs \( u_{ijk} \) and \( u_{i+1|k} \). In matrix form, (18) can be written compactly as
\[
|\Omega \otimes I_m|\bar{u}(k) | \leq M\delta_i \otimes 1_m, \tag{19}
\]

defining \( \Omega \in \mathbb{R}^{N \times N} \) as a matrix with \(-1\) along the diagonal, \(1\) along the superdiagonal and zeros elsewhere (i.e. an \( N \)-dimensional Jordan block corresponding to eigenvalue \(-1\)).

This allows the blocked ROA to be represented in mixed-integer form as
\[
\tilde{\mathcal{R}}_N(\delta_N) := \mathbb{P}_u(\tilde{Q}_N(\delta_N)), \tag{20}
\]
where
\[
\tilde{Q}_N(\delta_N) := \{ (x, \bar{u}) \mid \Theta^1_N x + \Theta^2_N \bar{u} \leq \phi_N, \Gamma^1_N x + \Gamma^2_N \bar{u} \leq h, |(\Omega \otimes I_m)\bar{u}| \leq M\delta_i \otimes 1_m \}
\]
for a sufficiently large \( M \).

### 4.2. Block minimisation

If a desired minimum ROA for the move blocked controller is given by the polytope
\[
\mathcal{R}_{\text{min}} := \text{co}\{z_1, z_2, \ldots, z_l\}. \tag{21}
\]

for vertices \( z_l \in \mathbb{R}^n, l \in \mathbb{Z}_{(1, l)} \), an optimisation problem can be formulated to find the smallest \( r \) and corresponding \( N \) and \( S \) such that \( \mathcal{R}_{\text{min}} \subseteq \mathcal{R}_N(S) \). Since the horizon length is now a decision variable, it is necessary to define the new matrices
\[
\Gamma^1_N := [G; GA; GA^2; \ldots; GA^N] \tag{22}
\]
\[
\Gamma^2_N := \begin{bmatrix} 0 & 0 & \ldots & 0 \\ GB & GB & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ GAB & GB & \ldots & GB \end{bmatrix} \tag{23}
\]

\[ \chi_N := 1_{N+1} \otimes h. \tag{24} \]

These matrices allow the different terminal constraints corresponding to each horizon length to be compactly represented. Additionally, the binary vectors
\[
\rho_N := \{ \rho_0, \rho_1, \ldots, \rho_N \} \in [0, 1]^{N+1} \tag{25}
\]
\[
\sigma_N := \{ \sigma_0, \sigma_1, \ldots, \sigma_{N-1} \} \in [0, 1]^N. \tag{26}
\]

are defined for representing the different horizon lengths. For computational tractability, it is necessary to impose the restriction \( N \leq \bar{N} \), for some maximum horizon length \( \bar{N} \). It is assumed that \( N \) and \( \mathcal{R}_{\text{min}} \) are chosen such that
\[
\mathcal{R}_{\text{min}} \subseteq \mathcal{R}_{\bar{N}}(I_{\bar{N}}). \tag{27}
\]

Theorem 5 now describes how \( \rho, \sigma \) and \( \bar{N} \) can be used within a mixed-integer linear program (MILP) to find optimal blocking structures with respect to the number of blocks and the horizon length.

**Theorem 5.** A blocking structure that minimises the number of blocks as the primary objective and the horizon length as the secondary objective, whilst ensuring that \( \mathcal{R}_{\text{min}} \subseteq \mathcal{R}_{\bar{N}}(\delta_N) \), is given by the solution of the MILP
\[
\mathcal{P}(\bar{N}) : \min_{\bar{u}, \bar{\delta}_N, \rho_N} \|\delta_N\|_1 + \frac{1}{\bar{N}} \|\sigma_N\|_1, \tag{28}
\]

subject to
\[
\Theta^1_N Z + \Theta^2_N \bar{U} \leq (\phi_N + M(1_{N} - \sigma_N) \otimes 1_{\bar{N}}) \otimes 1_{l} \tag{29a}
\]
\[
\Gamma^1_N Z + \Gamma^2_N \bar{U} \leq (\chi_N + M(1_{N+1} - \rho_N) \otimes 1_{\bar{N}}) \otimes 1_{l} \tag{29b}
\]
\[
|(\Omega \otimes I_m)\bar{U}| \leq M\delta_N \otimes 1_{m \times 1}, \tag{29c}
\]
\[
\sigma_i = 1 - \sum_{j=0}^{i} \rho_j, \quad \forall i \in \mathbb{Z}_{[0, N-1]} \tag{29d}
\]
\[
\|\rho_N\|_1 = 1, \tag{29e}
\]

where
\[
Z := [z_1, z_2, \ldots, z_l]. \tag{30}
\]
\[ \bar{U} \in \mathbb{R}^{m \times \bar{N}} \] and \( M \) is a suitably large number. A feasible solution to this optimisation problem always exists if (27) is satisfied.
For the objective function (28), the vector $\rho_j$ is used for encapsulating the horizon length within the optimisation problem. The different horizon lengths are represented by the implication, 
\[ \rho_j = 1 \implies N = j. \] (31)

The constraint (29e) ensures that $\rho_j$ can only represent a single horizon length. From (29d), the horizon length is then related to $\sigma_N$ by
\[ \sigma_j = 1, \quad \forall j < N \leq \bar{N}, \] (32)
so the term $\|\sigma_N\|_1$ can be used to measure the horizon length in (28). Given that the number of blocks, indicated by $\|\delta_N\|_1$, is the primary optimisation objective, the horizon length is divided by its maximum value $\bar{N}$. This ensures that a reduction in the number of blocks is always preferable to any reduction in the horizon length. The optimal blocking structures are therefore those that minimise the horizon length, out of all structures that minimise the number of blocks.

The set membership condition $\mathcal{R}_{\min} \subseteq \mathcal{R}(\delta_N)$ can be written in the form
\[ \exists \tilde{u}(x) : \Theta_1^j x + \Theta_2^j \tilde{u}(x) \leq \Phi_j, \quad \Gamma_1^j x + \tilde{c}_N \tilde{u}(x) \leq h, \] (33)
for all $x \in \mathcal{R}_{\min}$. Considering the output constraints (33), the binary vector $\sigma_N$ can be used to write the equivalent condition
\[ \Theta_1^j \tilde{u}(x) + \Theta_2^j \tilde{u}(x) \leq \Phi_j + M(1 - \sigma_N) \otimes 1_s, \] (35)
which restricts the constraints (5d) for $j > N$ (i.e., beyond the end of the horizon length indicated by $\rho_j = 1$). Since the constraints and $\mathcal{R}_{\min}$ are convex, the universal quantifier over all $x \in \mathcal{R}_{\min}$ can be replaced by the vertex-based constraints
\[ \Theta_1^l \tilde{u}_l + \Theta_2^j \tilde{u}_l \leq \Phi_j + M(1 - \sigma_N) \otimes 1_s, \] (36)
for all $l \in \mathcal{L}(1, l)$, where $\tilde{u}_l$ is an input sequence corresponding to the $l$th vertex of $\mathcal{R}_{\min}$. Defining
\[ \bar{U} := \{ \tilde{u}_1, \tilde{u}_2, \ldots, \tilde{u}_l \}, \] (37)
substituting (30) and using the Kronecker product to duplicate the right-hand side of (36) for each vertex gives the constraint (29a).

The terminal constraint (34) can be approached in a similar manner. The matrices (22)–(24) are used to capture all possible terminal constraints for any $N \leq \bar{N}$, giving the constraints
\[ \Gamma_1^j x + \Gamma_2^j \tilde{u}(x) \leq x_N + M(1_{\bar{N}+1} - \rho_N) \otimes 1_s. \] (38)

A big-M formulation ensures that only the constraint corresponding to horizon length indicated by $\rho_N$ is enforced. As with the output constraints, the vertices of $\mathcal{R}_{\min}$ are then used to formulate (29b). The remaining constraint (29c) enforces the blocking as in (19), over all rows of $U$. Since an unblocked horizon is a feasible solution to $\mathcal{S}(N)$, it is evident that a solution will always exist if (27) is satisfied.

**Remark 6.** The constraints (29a)–(29b) can be rewritten in terms of a halfspace description of $\mathcal{R}_{\min}$ by extending the dualisation procedure of Goulart, Kerrigan, and Maciejowski (2006). This may result in a simpler optimisation problem for high state dimensions. In either case, the optimisation is performed offline.

The reduction in online computation time provided by solving (28) is highly dependent on the structure of the system and constraints, as well as the solution method employed for optimisation. For example, a sparse constraint structure with more decision variables may solve faster than a dense structure with fewer variables, depending on the solver. In Theorem 5, the objective has been designed for a condensed dual active-set solution algorithm, where the number of decision variables ($O(m)$) is more critical than the number of constraints ($O(N)$).

### 4.3 Volume optimisation

Consider now the problem of choosing the blocking strategy that maximises the volume of the ROA, assuming that the horizon length $N$ and number of blocks $r$ are known design parameters derived from the maximum capabilities of the controller hardware. There is no closed-form expression for the exact volume of (16) as a function of $S$. In addition, even numerically computing this volume for a given $S$ is difficult, since it requires polytope projection, which is a computationally challenging operation for long prediction horizons and high state dimensions. Indeed, Fourier–Motzkin elimination (Dantzig & Eaves, 1973), one of the most common half-space projection algorithms, has a worst-case double-exponential complexity, given the increasing number of constraints generated at each iteration.

To avoid this issue, consider an ellipsoid in $\mathbb{R}^n$ lifted onto a blocking-admissible translated $n$-dimensional subspace of $\mathbb{R}^{n+N}$. Such an ellipsoid can be expressed as an affine transformation of an $n$-dimensional 2-norm ball, which can be written as
\[ \mathcal{E} := \{ z \in \mathbb{R}^{n+N} \mid z = [P; Q] \xi + [c; d], \forall z \in \mathbb{R}^n : \| \xi \|_2 \leq 1 \}, \] (39)
where $P \in \mathbb{R}^{n \times n}$, $P \succ 0$ is the shape matrix and $c \in \mathbb{R}^n$ is the centre. The coefficients
\[ Q := [Q_1; Q_2; \ldots; Q_n], \quad d := [d_1; d_2; \ldots; d_n], \quad Q_i \in \mathbb{R}^{m \times n}, \quad d_i \in \mathbb{R}^{m \times n}, \quad \forall j \in \mathbb{Z}[1, n] \]
determine the translated subspace into which the ellipsoid is lifted.

Let the set of all such lifted ellipsoids be denoted $\mathcal{E}$. If $\mathcal{E}^*(\delta_N)$ denotes the largest volume ellipsoid in $\mathcal{E}$ satisfying
\[ \mathcal{E}^*(\delta_N) \subseteq Q_N(\delta_N), \] then it is evident that
\[ \mathcal{P}_n(\mathcal{E}^*(\delta_N)) \supseteq \mathcal{P}_n(Q_N(\delta_N)) = \mathcal{R}(\delta_N), \]
from (20). Thus, $\mathcal{P}_n(\mathcal{E}^*(\delta_N))$ is an inner approximation to the ROA that can be used for estimating the volume. Theorem 7 describes how this approximation can be employed for selecting blocking structures.

**Theorem 7.** A blocking structure to subdivide $N$ inputs into $r$ blocks, whilst maximising an inner approximation to the ROA volume, is given by the solution of the mixed-integer problem
\[ \mathcal{S}^*(N, r) : V^*(N, r) = \max \log \det P, \quad P \succeq 0, \forall \xi, \forall d, \forall \delta_N \] subject to
\[ \phi_i^\top [c; d] + \| [P; Q] \phi_i \|_2 \leq \psi_i, \quad \forall \xi \in \mathbb{Z}[1, N+n], \] (a)
\[ (\Omega \otimes I_m) \delta_N \leq M \delta_N \otimes 1_{m \times m}, \] (b)
\[ (\Omega \otimes I_n) d \leq M \delta_N \otimes 1_{m}, \] (c)
\[ \| \delta_N \|_1 = r, \] (d)
for some suitable large $M$, where $\phi_i^\top$ and $\psi_i$ denote the $i$th rows of the matrices.
\[ \phi := [\phi_1^\top, \phi_2^\top, \ldots, \phi_n^\top]^\top \quad \text{and} \quad \psi := [\phi_N]^\top \] respectively.
Proof. An inner ellipsoid of the form (39) contained within \( \mathcal{R}_N \) that has maximum projected volume can be found by extending the standard Löwner–John problem (Zhang & Gao, 2003). This leads to the maximisation of the objective function \( \log \det P \) \cite{40}, which is proportional to the projected volume, subject to (41a). Since the matrices \( Q_k \) and \( c_k \) denote the ellipsoid coefficients traversing the input subspace, the extra constraints (41b)–(41c) are used to enforce the blocking structure on these matrices as specified by \( \delta_k \), in a similar way to (19). Finally, given that there must be precisely \( r \) blocks, it follows that the elements of \( \delta_r \) must sum to \( r \) \cite{41d}.

Remark 8. It can be shown that a feasible solution to this problem always exists for any nonempty convex constraint set \( Y \) that contains the origin, based upon the controllability assumption on the system (Section 2).

As discussed in Boyd, El Ghaoui, Feron, and Balakrishnan \cite{94}, with convex Linear Matrix Inequality (LMI) constraints, the maximisation of \( \log \det P \) may be cast as a semidefinite program (SDP). Incorporating the blocking structure \( \delta_k \) into the problem leads to a Mixed-Integer Semidefinite Program (MISDP). The solution can be found using an upper-level branch-and-bound method in addition to a lower-level SDP solution algorithm. For small problems, it may be possible to simply calculate the volume approximation over all possible blocking structures for a given \( N \) and \( r \), choosing the structure giving the largest approximate ROA volume. There are however \( (r-1) \) possible ways of blocking an \( N \)-step horizon into \( r \) blocks (Shekhar & Maciejowski, 2012b), which grows rapidly with \( N \).

Theorem 9 now shows that optimising with fewer blocks can only decrease the volume of the inner ellipsoidal approximation. The result demonstrates that a better ROA volume cannot be obtained by reducing the number of blocks specified in (41d).

Theorem 9 (Cost Monotonicity for Blocking Relaxation). \( V^*(N, \tilde{r}) \leq V^*(N, r), \) for all \( \tilde{r} < r \).

Proof. Let a feasible solution to \( \mathcal{P}(N, r) \) be given by the tuple \((P, c, Q, d, \delta_N)\). Then, a corresponding feasible solution to \( \mathcal{P}(N, \tilde{r}) \) is given by the tuple \((P, c, Q, d, \delta_N \cup \gamma)\), where \( \delta_N \cup \gamma = \delta_N \cup \gamma, \) for any \( \gamma \in \{0, 1\}^r \) such that \( \|\delta_N\| = r \). This is evident from the fact that a more restrictive blocking structure with \( \tilde{r} \) blocks can always be relaxed (Shekhar & Maciejowski, 2012b) to a less restrictive structure with \( r > \tilde{r} \) blocks, whilst still admitting \( Q \) and \( d \) as part of the feasible solution. In other words, reducing the number of blocks always introduces more restrictive equality constraints on (41b)–(41c). Hence, every feasible inner ellipsoidal approximation corresponding to \( \mathcal{P}(N, \tilde{r}) \) is also a feasible solution to \( \mathcal{P}(N, r) \). By optimality, this means that \( V^*(N, \tilde{r}) \), which is monotonically related to the volume of the inner approximation, is a lower bound on \( V^*(N, r) \).

Remark 10. For a given system, the computational complexity for both Problems \( \mathcal{P}(\cdot) \) and \( \mathcal{P}(\cdot, \cdot) \) depend on the maximum horizon length, since this determines the number of integer variables required. However, the complexity has no impact on online controller evaluation, since both problems are solved once offline to determine optimal blocking structures for given control problems.

5.2. ROA volume optimisation for fixed complexity

For the second example, consider regulating the state of the unstable system

\[
\begin{bmatrix}
  1 & 1 \\
  -0.1 & 1 \\
\end{bmatrix} x(k) + \begin{bmatrix}
  0.5 \\
  1 \\
\end{bmatrix} u(k),
\]

subject to the state and input constraints \( |x(k)| \leq \begin{bmatrix} 8 & 5 \end{bmatrix} \) and \( |u(k)| \leq 2.5 \). The terminal constraint \( \tilde{r} \) is again chosen to be the maximum control invariant set that satisfies \( \|x\|_\infty \leq 1 \), for all \( x \in \tilde{r} \). The system in this example has a larger margin of instability than the previously considered system, since the effect of blocking structure on ROA volume is more evident for such systems.

It is desired to find the blocking structure that subdivides \( N = 7 \) inputs into \( r = 4 \) blocks, using the volume approximation method presented in Section 4.3. This small horizon length and block count
allows explicit volume computation over all blocking structures, to verify the efficacy of the optimisation technique. Problem (7, 4) (Theorem 7) is then solved using MOSEK as the inner SDP solver together with YALMIP’s built-in branch-and-bound solver. The problem solves in 0.78 s, giving the blocking matrix

\[ S^* = 1 \oplus [1:1] \oplus [1:1] \oplus [1:1]. \] (46)

Implementing this blocking structure gives an ROA volume of 74.75, which is also the maximum achievable volume over all blocking structures for this \( N \) and \( r \). In contrast, a “classical” blocking structure \( S = I_3 \oplus 1_4 \), gives a volume of 62.28. This blocking structure corresponds to standard industry practice of having a shorter control horizon compared to the longer prediction horizon (Maciejowski, 2002).

Fig. 3 shows the nominal ROA, together with the ROAs for optimal and classical blocking. Table 1 shows the volume over all possible blocking structures for this horizon length and block count. It can be seen that whilst the volume of the optimal ROA is larger, it is not a superset of the classical blocking ROA. However, it is possible to modify the optimisation problem to give preference to ROA volume in certain directions if this is needed. In particular, the constraints (41a) can be modified by a weighting matrix to scale the volume measure in the desired coordinate directions.

6. Conclusion

This paper has shown how move blocking strategies for predictive control can be optimised to improve computational tractability when a desired ROA is specified, or increase ROA volume for fixed computational complexity. It is shown how a certain input parameterisation can be used to guarantee recursive feasibility, allowing time-invariant blocking structures to be considered. A mixed-integer approach has been shown to allow direct optimisation of the blocking structure, for two different design requirements.

Future work will consider including implementation-specific parameters like sparsity in the design of complexity-reducing blocking structures. It will also consider different blocking structures for different input channels, as well as the augmentation of robustness modifications to the controller. Finally, it will be investigated whether closed-loop cost can be explicitly considered in the optimisation problems, for specific cost functions.