Controlled invariant feasibility — A general approach to enforcing strong feasibility in MPC applied to move-blocking

Ravi Gondhalekar, Jun-ichi Imura, Kenji Kashima

Abstract

Strong feasibility of MPC problems is usually enforced by constraining the state at the final prediction step to a controlled invariant set. However, such terminal constraints fail to enforce strong feasibility in a rich class of MPC problems, for example when employing move-blocking. In this paper a generalized, least restrictive approach for enforcing strong feasibility of MPC problems is proposed and applied to move-blocking MPC. The approach hinges on the novel concept of controlled invariant feasibility. Instead of a terminal constraint, the state of an earlier prediction step is constrained to a controlled invariant feasible set. Controlled invariant feasibility is a generalization of controlled invariance. The convergence of well-known approaches for determining maximum controlled invariant sets, and j-step admissible sets, is formally proved. Thus an algorithm for rigorously approximating maximum controlled invariant feasible sets is developed for situations where the exact maximum cannot be determined.

Keywords:
Model predictive control
Constrained control
Strong feasibility
Set invariance
Move-blocking

1. Introduction

Model predictive control (MPC) is attractive for its ability to explicitly accommodate hard constraints on states and control inputs (Bemporad, Morari, Dua, & Pistikopoulos, 2002; Maciejowski, 2002; Mayne, Rawlings, Rao, & Scokaert, 2000). However, strong feasibility of the associated MPC problem must be enforced. Strong feasibility of MPC problems describes the quality that the closed-loop state trajectory from any feasible initial state, due to any feasible control input sequence, never reaches an infeasible state (Kerrigan, 2000). Strong feasibility of finite-horizon MPC problems is usually explicitly enforced by constraining the state at the final prediction step to a controlled invariant (CI) set (Aubin, 1991; Blanchini, 1999; Blanchini & Miani, 2008; Maciejowski, 2002; Mayne et al., 2000). However, this is an indirect method. Terminal constraints cannot enforce strong feasibility in all MPC problems. To reduce the computational complexity of finite-horizon MPC problems it is common practice to parameterize the predicted control move trajectory in some manner (Cagienard, Grieder, Kerrigan & Morari, 2007; Gondhalekar & Imura, 2007; Goodwin, Seron, Middleton, Zhang, Hennessy, Stone, & Menabde, 2006; Maciejowski, 2002; Qin & Badgewell, 2003; Tendel & Johansen, 2002). A large variety of approaches exist. In move-blocking MPC schemes the predicted control move trajectory is parameterized by a control move sequence of fewer steps, and the moves of this sequence applied to sets of multiple prediction steps (Cagienard et al., 2007; Maciejowski, 2002; Tendel & Johansen, 2002). Alternatively, in MPC of continuous-time plants the prediction horizon may be irregularly time-discretized such that predicted sample-periods need not be equal to the actual system step size (Gondhalekar & Imura, 2007; Goodwin et al., 2006). In such cases terminal constraints fail to enforce strong feasibility, so these methods are currently employed without guarantees of constraint satisfaction.

In this paper a generalized approach for enforcing strong feasibility of MPC problems is proposed (Section 3). The MPC problems may employ move-blocking, irregular time-discretization, or other features which render terminal constraints ineffective. To illustrate the new approach this paper focuses on move-blocking MPC schemes. In the proposed method not the terminal state, but the state of an earlier prediction step is constrained to a so-called controlled invariant feasible (CIF) set. In the general case the CIF constraint must be placed on the state of the first prediction step (Section 3.1). In special cases the CIF constraint may be placed on the state of a later prediction step (Section 3.2). The proposed method is least restrictive if the CIF constraint set is the maximum...
CIF (MCIF) set. By considering the uncompensated, full degree of freedom MPC problem as a special member of the class of move-blocking MPC problems it is shown that CI terminal constraints are a special case of CIF constraints. Thus the CIF constraint approach proposed here is a generalization, not specialization, of the well-known terminal constraint method (Mayne et al., 2000). Due to the difficulty in applying the approach to general MPC problems this paper limits itself hereafter to MPC of stabilizable linear time-invariant plants with polytopic constraint sets containing the origin within their interior. The main results of this paper were first reported in Gondhalekar and Imura (2007a), Gondhalekar and Imura (2007b) and Gondhalekar, Imura, and Kashima (2009).

Further in this paper, methods to determine MCIF sets are developed (Section 4). CIF sets are conceptually more general than CI sets, but mathematically equivalent to CI sets for systems with state-dependent control input constraints. The contractive algorithm of Blanchini (1994), Dória and Hennet (1999) and Vidal, Schaffert, Lygeros, and Sastry (2000) for determining maximum controlled invariant (MCI) sets is adapted to determine MCIF sets. Unfortunately this algorithm is undecidable; implementable but not guaranteed to terminate in a finite number of iterations (Vidal et al., 2000). The expansive algorithm of Guttman and Cwikel (1987) and Keerti and Gilbert (1987) for determining CI j-step admissible sets is adapted to determine CIF j-step admissible sets. This paper’s contribution to invariant set theory is to formally prove the convergence of these approaches (Lemma 12), and the subsequent derivation of an MCIF set condition (Theorem 13). As a result a CIF under-approximation of the MCIF set can be computed to arbitrary accuracy, in a finite number of iterations, and with a rigorous error bound obtained at every iteration, even in cases when the exact MCIF set cannot be determined. This issue was not considered in Blanchini (1994), Dória and Hennet (1999), Guttman and Cwikel (1987) and Keerti and Gilbert (1987) and Vidal et al. (2000). Some related ideas are reported in Blanchini, Miani, and Savorgnan (2008) and Raković and Fiacchini (2008).

2. Preliminaries

2.1. Notation

The set of reals is denoted by $\mathbb{R}$ ($\mathbb{R}_+$: strictly positive), the set of non-negative integers by $\mathbb{N} = \{0, 1, 2, \ldots\}$. The identity matrix by $I_{n}$. Denote by $I_{n,m} \in \{0, 1\}^{n \times m}$ the identity matrix, by $0_{n,m} \in \{0\}^{n \times m}$ the zero matrix, by $0$ without subscript the zero matrix with dimension deemed obvious by context. The spectral radius of a matrix $A$ is denoted by $\rho(A)$, the element of row-$j$ and column-$k$ of $A$ by $a_{jk}$. The Kronecker product of matrices $A$ and $B$ is denoted by $A \otimes B$. Inequalities $A \leq B$ hold component-wise. The set of non-empty subsets of a set $X$ is denoted by $2^X$. For $X \in 2^\mathbb{R}$, $\partial X$ denotes the closure, $\partial X$ the boundary and $\mathbb{X}$ the interior. For $X, Y \in 2^\mathbb{R}$ the directed Hausdorff distance from $X$ to $Y$ is denoted by $H: 2^\mathbb{R} \times 2^\mathbb{R} \to \mathbb{R}_+$, $H(X, Y) := \sup_{x \in X} \inf_{y \in Y} |x - y|$, where $| \cdot |$ is the Euclidean norm. A sequence of elements $x_i, x \in \mathbb{X}$ is denoted by $[x_i \in \mathbb{X}]_{i \in \mathbb{N}}$. If the elements’ parent set is obvious by context the sequence is denoted simply by $[x_i]_{i \in \mathbb{N}}$. Let $\psi(i,k)$ denote the future value of variable $\psi$ at step $i + k$, as predicted from step $i$. For compact notation $\psi(i,0) := \psi_0$.

2.2. Move-blocking linear-quadratic MPC

Consider discrete-time linear time-invariant system

$$x_{i+1} = Ax_i + Bu_i \quad \text{subject to} \quad E \dot{x}_i + Gu_i \leq W$$

subject to $E \dot{x}_i + Gu_i \leq W$ (1)

with $i \in \mathbb{N}$, state $x \in \mathbb{R}^n$, input $u \in \mathbb{R}^m$, $m \in \mathbb{N}_+$, and $A, B, E, G, W \in \mathbb{R}$ compatible dimensions. Let $\mathbf{Z} := \{z \in \mathbb{R}^{n+m} \mid \mathbf{E} \mathbf{G} z \leq \mathbf{W}\}$, $\mathbf{U}(x) := \{u \in \mathbb{R}^m \mid \mathbf{E} x + Gu \leq \mathbf{W}\}$, $\mathbf{U} : \mathbb{R}^n \to \{2^\mathbb{R}^m, \emptyset\}$, $\mathbf{X} := \{x \in \mathbb{R}^n \mid \mathbf{U}(x) \neq \emptyset\}$.

Assumption 1. The pair $(A, B)$ is stabilizable.

Assumption 2. Constraint set $\mathbf{Z}$ is bounded and contains the origin within its interior ($W > 0$).

MPC achieves closed-loop control action of system (1) by applying at current state $x_i = x_i(0)$ the first control input $u_{i(0)}$ of a predicted open-loop control input trajectory $U_i := [u_{i(0)}, \ldots, u_{i(N)}] \in \mathbb{R}^{N_n}$, $N \in \mathbb{N}_+$. The optimal predicted open-loop control input trajectory is determined by the solution of MPC Problem 3 with quadratic cost matrices $P, Q \in \mathbb{R}^{n \times n}$, $R \in \mathbb{R}^{m \times m}$.

Problem 3. Determine:

$$U_i^*(x_i) := \arg \min_{\hat{U}_i \in \mathbb{R}^{N_n}} \left\{ H^T \hat{U}_i H + \hat{U}_i^T I \hat{X}_i \right\}$$

subject to: $E \hat{x}_i + \hat{G} \hat{U}_i \leq \hat{W}$.

(5)
The MPC control law \( \kappa : \mathbb{R}^n \to [\mathbb{R}^m, \emptyset] \) is given implicitly by \( \kappa(x_i) := FMU^{x_i}(x_i), F := [I_m, 0_{m(N-1)m}] \). The closed-loop state trajectory subsequently evolves according to \( x_{i+1} := Ax_i + Bu(x_i) \). Let \( B = BFM \) and note that the pair \((A, B)\) is stabilizable due to Assumptions 1 and 4. Let \( \bar{U}(x) := \{ \bar{U} \in \mathbb{R}^m| \bar{E}x + \bar{G}\bar{U} \leq \bar{W} \}, \bar{U}: \mathbb{R}^n \to [2^{g^m}, \emptyset] \). Call the set \( \bar{X} := \{ x \in \mathbb{R}^n| \bar{U}(x) \neq \emptyset \} \) the feasible set, and any state \( x \in \bar{X} \) a feasible state.

2.3. Strong feasibility of MPC problems

**Definition 6.** An MPC problem is recursively feasible from feasible initial state \( x_0 \) iff the closed-loop state trajectory \( \{ x_i \}_{i=0}^\infty \) remains within the feasible set.

**Definition 7.** An MPC problem is strongly feasible (Kerrigan, 2000) iff from every feasible state the closed-loop state trajectory due to any sequence of feasible predicted open-loop control input trajectories remains within the feasible set.

The set \( \bar{X}_\infty \) of states which MPC Problem 5 is recursively feasible from is given by \( \bar{X}_\infty := \{ x_0 \in \bar{X}| x_{i+1} = Ax_i + Bu(x_i) \in \bar{X} \forall i \in \mathbb{N} \} \).

MPC Problem 5 is strongly feasible if and only if: \( x_{i+1} = Ax_i + Bu_i \in \bar{X} \forall i \in \mathbb{N} \). The notion of recursive feasibility is concerned with a particular instance of an optimal control law \( \kappa(\cdot) \), and whether the closed-loop state trajectory due to this optimal control law remains within the feasible set. In contrast, the notion of strong feasibility is concerned with the constraints of the system only, and is independent of both the prediction cost function defined by matrices \( P, Q \), and \( R \), and also optimality of the predicted control input trajectory \( U_i \).

**Definition 8.** A non-empty set \( \mathcal{C} \subset 2^{\mathbb{R}^m} \) is a CI set (Aubin, 1991; Blanchini, 1999; Blanchini & Miani, 2008) of system (1) iff \( \forall x \in \mathcal{C} \exists u \in U(x) \text{ s.t. } Ax + Bu \in \mathcal{C} \). The MCI set \( \mathcal{C}^* \) is defined as:

\[
\mathcal{C}^* := \left\{ x_0 \in \mathcal{C}| \exists [u_i \in \mathbb{R}^m]_{i=0}^\infty \text{ s.t. } x_{i+1} = Ax_i + Bu_i \in \mathcal{X} \forall i \in \mathbb{N} \right\}.
\]

If \( M = I_n \) then move-blocking MPC Problem 5 is equivalent to unparameterized, full degree of freedom MPC Problem 3. In this special case it is common practice to enforce strong feasibility by a CI terminal constraint. Setting terminal constraint (3) to \( x_N \in \mathcal{C}^* \) is sufficient for generating a least restrictive, strongly feasible MPC problem. In this case, if the state of the previous step was feasible then the shifted control input trajectory from the previous step is guaranteed to be a feasible control input trajectory for the current step, up to but not including the final control move. Existence of a final feasible predicted control move is guaranteed by the CI terminal constraint (Mayne et al., 2000). However, when \( M \neq I_n \) this does not hold due to the extra constraints imposed on the predicted control input trajectory by move-blocking parameterization (4) (Cagnier et al., 2007). The shifted control input trajectory from the previous step is not guaranteed to be a feasible control input trajectory for the current step. Thus CI terminal constraints fail to enforce strong feasibility of move-blocking MPC problems.

A time-dependent move-blocking scheme which guarantees recursive feasibility was proposed in Cagnier et al. (2007). This method is successful in many situations, but is not guaranteed to outperform time-invariant move-blocking, widely employed in practice (Maciejowski, 2002; Qin & Badgwell, 2003; Tendal & Johansen, 2002). Furthermore, the method is in some ways undesirable, as time-dependence precludes the use of common tools of controller design and performance evaluation, e.g. multi-parametric quadratic programming (Bemporad et al., 2002), and the subsequent stability (Ferrari-Trecate, Cuzzola, Mignone, & Morari, 2002); performance (Gonhailekar & Imura, 2008) and invariance (Blanchini, 1999) analysis of time-invariant piecewise affine systems. The purpose of this paper is to provide a means of enforcing strong feasibility for use in situations when time-invariant move-blocking is deemed a successful and desirable control strategy.

3. Controlled invariance feasibility

In order to formulate a strongly feasible MPC problem employing time-invariant move-blocking it is necessary to directly enforce that at each step the state at the next step along the closed-loop state trajectory remains within the feasible set. The notion of controlled invariance feasibility is proposed for this purpose. Controlled invariance itself is agnostic to MPC Problem 5. The key is to bestow knowledge of the constraints of MPC Problem 5 upon the notion of controlled invariance.

3.1. General blocking regimes

**Definition 9.** A non-empty set \( \mathcal{F} \subset 2^{\mathbb{R}^m} \) is a CIF set of MPC Problem 5 iff \( \forall x \in \mathcal{F} \exists U(x) \text{ s.t. } Ax + Bu \in \mathcal{F} \). The MCIF set \( \mathcal{F}^* \) is defined as:

\[
\mathcal{F}^* := \left\{ x_0 \in \mathcal{F}| \exists [U_i \in \mathbb{R}^m]_{i=0}^\infty \text{ s.t. } x_{i+1} = Ax_i + Bu_i \in \mathcal{X} \forall i \in \mathbb{N} \right\}.
\]

Definition 9 explicitly and simultaneously enforces both that each element of a CIF set admits a feasible solution to MPC Problem 5, and also that the solution renders the CIF set positively invariant. Definition 9 of CIF sets is technically equivalent to Definition 8 of CI sets. The conceptual difference is that CI sets only guarantee the existence of a single admissible control move for each element of the set, such that applying the control move renders the set positively invariant. On the other hand, CIF sets guarantee the existence of an admissible control move trajectory, of multiple steps, which must satisfy the constraints of MPC Problem 5 (\( \mathcal{F} \subset \mathcal{F}^* \)). Furthermore, applying only the first control move of the trajectory renders the set positively invariant (\( \mathcal{F}^* \subset \mathcal{C}^* \)).

MCIF set \( \mathcal{F}^* \) contains all elements \( x_0 \) of feasible set \( \bar{X} \) from which there exists an infinite sequence of solutions to MPC Problem 5 such that the resulting closed-loop state trajectory remains within \( \bar{X} \). Thus a state not element of MCIF set \( \mathcal{F}^* \) cannot result in recursive feasibility. Note that MCIF set \( \mathcal{F}^* \) is independent of \( P, Q \), and \( R \). It holds that \( \mathcal{F}^* \supseteq \bar{X}_\infty \) for any choice of \( P, Q \), and \( R \). Using the MCIF set \( \mathcal{F}^* \) the following result is obtained.

**Theorem 10.** MPC Problem 5 is recursively feasible from a feasible state \( x_0 \in \bar{X} \) iff at each state along the closed-loop state trajectory \( \{ x_i \}_{i=0}^\infty \) MPC Problem 5 yields a solution \( U_i^*(x_i) \) s.t. \( x_{i+1} \in \mathcal{F}^* \).

**Proof.** (If) If at feasible state \( x_i \in \bar{X} \) MPC Problem 5 yields a solution \( U_i^*(x_i) \) s.t. \( x_{i+1} \in \mathcal{F}^* \), then \( x_{i+1} = x_{i+1}^* \in \mathcal{F}^* \). Thus \( x_{i+1} \in \bar{X} \). By induction the argument holds recursively. (Only if) Suppose at feasible state \( x_i \in \bar{X} \) MPC Problem 5 yields a solution \( U_i^*(x_i) \) s.t. \( x_{i+1} \notin \mathcal{F}^* \). Then \( x_{i+1} = x_{i+1}^* \notin \mathcal{F}^* \), hence \( \exists U \in \mathbb{R}^m \) s.t. \( x_{i+1} = Ax_i + Bu_i \notin \mathcal{X} \). Consequently \( x_j \notin \bar{X} \) for some \( j > i \).

Motivated by Theorem 10 we propose to solve MPC Problem 5 including the additional CIF constraint \( X_{i+1} \subset \mathcal{F}^* \). MPC Problem 5 with this CIF constraint is the least restrictive, strongly feasible MPC problem satisfying prediction constraints (5). Any CIF set is sufficient for enforcing strong feasibility. However, a non-maximum CIF set results in a restrictive control law. The representation of \( \mathcal{F}^* \) may be very complex, so for implementation it may be desirable to employ a low-complexity under-approximation of \( \mathcal{F}^* \). Analogous statements hold for MCI set \( \mathcal{C}^* \) and terminal constraints.
3.2. A special class of blocking regimes

The preceding discussion on CIF constraints can be specialized for the block-diagonal class of blocking matrix \( M = \text{diag}(l_{i}, M) \), with \( N \in \mathbb{N}_+, N \leq \tilde{N} \) and \( M \in \{0,1\}^{(N-\tilde{N}) \times (N-\tilde{N})} \). This implies that the first \( N \) inputs \( u_{i} \), \( v_{k} \in \mathbb{N}^{\tilde{N}-1} \) of predicted control input trajectory \( U_{i} \) are chosen independently of both each other, as well as of the inputs towards the prediction horizon end. In this special case it is preferable to apply the CIF constraint to prediction state \( x_{i}(\tilde{N}) \) as opposed to \( x_{i}(\tilde{N}+1) \).

Consider the constraints of the latter part of MPC Problem 3. Defining the truncated, move-blocked predicted open-loop control move sequence \( U_{i} := [u_{(i)}^{T}, \ldots, u_{(i-\tilde{N}+1)}^{T}]^{T} \in \mathbb{R}^{(N-\tilde{N})m} \), these constraints can be rewritten in matrix form as \( \hat{E} x_{i}(\tilde{N}) + \hat{G} U_{i} \leq \hat{W} \). The feasible set \( \mathcal{X} \) of the truncated MPC problem is given by \( \mathcal{X} := \{ x \in \mathbb{R}^{m}(\exists u_{i} \in \mathbb{N}^{\tilde{N}-1} \text{ s.t. } \hat{E} x + \hat{G} U \leq \hat{W} \} \). Thus MPC Problem 5 is feasible only if \( \exists(u_{i},h_{i})^{\tilde{N}-1}_{0} \text{ s.t. } x_{i}(\tilde{N}) \in \mathcal{X} \). Consider the MCI set \( \mathcal{X}^{*} \) within the feasible set \( \mathcal{X} \). Equivalently to Theorem 10, MPC Problem 5 is recursively feasible if \( \exists(u_{i},h_{i})^{\tilde{N}-1}_{0} \text{ s.t. } \forall i \in \mathbb{N}, \text{ all constraints are satisfied and } x_{i}(\tilde{N}) \in \mathcal{X}^{*} \). Thus the strongly feasible MPC problem is formulated as MPC Problem 5 with CIF constraint \( x_{i}(\tilde{N}) \in \mathcal{X}^{*} \). Recursive feasibility of the MPC problem over the former part of the prediction horizon is guaranteed. Because the first \( N \) control inputs are independent the shifted truncated control move trajectory from the previous step is always a feasible control move trajectory for the current step, analogously as for full degree of freedom MPC problems (Mayne et al., 2000).

Applying the CIF constraint on state \( x_{i}(\tilde{N}) \) is preferable over \( x_{i}(\tilde{N}+1) \) for two reasons. First, the sensitivity of the size of the feasible set w.r.t. the size of the CIF constraint set is reduced. This is because when applying a CIF constraint on state \( x_{i}(\tilde{N}+1) \) the feasible set is given by the 1-step admissible set to the CIF constraint set. On the other hand, when applying the CIF constraint to state \( x_{i}(\tilde{N}) \) the feasible set is given by the \( N \)-step admissible set to the CIF constraint set. This is irrelevant when the employed CIF constraint set is the MCF set. However, this point is important in case it is not possible to determine the exact MCF set, or when it is desirable to use a low-complexity CIF under-approximation of the MCF set. Second, computing the MCF set is generally less computationally demanding, as the feasibility and controlled invariance properties are decoupled.

Note that in general the maximum controlled invariant set \( \mathcal{X}^{*} \) within the feasible set \( \mathcal{X} \) is not simply maximum controlled invariant set \( \mathcal{X}^{*} \). This is because not all elements \( x \in \mathcal{X}^{*} \) are guaranteed to admit a feasible solution \( U \) satisfying \( \hat{E} x + \hat{G} U \leq \hat{W} \). If \( M = I_{k} \), i.e. \( N = \tilde{N} = N \), then \( \mathcal{X} = \mathcal{X} \) and \( \mathcal{X}^{*} = \mathcal{X}^{*} \). Thus in the full degree of freedom case when \( M = I_{k} \) the MCF constraint is equivalent to the well-known MCI terminal constraint (Mayne et al., 2000). In this sense the approach proposed in this paper is a generalization, not specialization, of current CI terminal constraint methods.

4. Computing MCI/F sets

Methods for computing MCI/F set \( \mathcal{X}^{*}/\mathcal{F}^{*} \) are discussed in this section. Existing methods to determine MCI and CI \( j \)-step admissible sets are extended to accommodate state-dependent input constraints in Section 4.1. A proof of convergence of these methods, an MCI set condition and a method for rigorously approximating MCI sets in cases when the exact MCI set cannot be determined are derived in Section 4.2. The results are directly applicable for systems with state-independent input constraints.

For system (1) under Assumptions 1, 2 and 4 the MCI/F sets \( \mathcal{X}^{*}/\mathcal{F}^{*} \) contain the origin within their interiors and are compact and convex. Analogous results are reported in Blanchini (1994) and Dória and Hennet (1999). A proof is omitted for brevity. For a compactness proof the reader is referred to the proof of Proposition 3.1 of Blanchini (1994). Other proofs are straightforward. These properties are not guaranteed in more general settings.

4.1. Extension of existing methods

For a set \( \Omega \subset \mathbb{N}^{m} \) define the Reach: \( 2^{\mathbb{N}^{m}} \rightarrow 2^{\mathbb{N}^{m}} \) and Pre: \( 2^{\mathbb{N}^{m}} \rightarrow 2^{\mathbb{N}^{m}} \) operators: Reach(\( \Omega \)) := \( \{ x \in \mathbb{R}^{m} \mid u \in \mathbb{N}(x) \text{ s.t. } Ax + Bu \in \Omega \} \), Pre(\( \Omega \)) := Reach(\( \Omega \)) \cap \( \Omega \). Lemma 11 (1,2) extends analogous CI set results of Theorem 2.1 in Dória and Hennet (1999), and Proposition 2 in Vidal et al. (2000), to the case of state-dependent inputs, respectively. A proof of Lemma 11 is straightforward but omitted for brevity.

Lemma 11. A set \( \Omega \subset \mathbb{N}^{m} \) is a CI set iff: (1) \( \Omega \subset \text{Reach}(\Omega) \cup \Omega \) (2) \( \Omega = \text{Pre}(\Omega) \).

Consider the sequence \( \{ \Phi_{j} \subset \mathbb{R}^{m}_{\text{in}} \}_{j=0}^{\infty} \) defined by: \( \Phi_{0} := \emptyset, \Phi_{j+1} := \text{Reach}(\Phi_{j}) j \in \mathbb{N} \). The following statements hold: \( \forall i j, i \in \mathbb{N}, \Phi_{j} \cap \mathcal{X}^{*} \):\( \subset \mathcal{X}^{*} \) \( \subset \mathcal{X}^{*} \) \( \subset \mathcal{X}^{*} \) \( \subset \mathcal{X}^{*} \) \( \subset \mathcal{X}^{*} \). If follows that \( \mathcal{X}^{*} = \bigcap_{j=0}^{\infty} \Phi_{j} \). This is the method for determining MCI sets described in Blanchini (1994), Blanchini and Miani (2008), Dória and Hennet (1999) and Vidal et al. (2000). The problem of computing the MCI set \( \mathcal{X}^{*} \) via the sequence \( \{ \Phi_{j} \}_{j=0}^{\infty} \) is undecidable. Unless the sequence arrives at a fixed-point (i.e. \( \Phi_{j} = \Phi_{j+1} \)) the sets \( \Phi_{j} \) are not CI sets. The algorithm of Blanchini (1994), Blanchini and Miani (2008) and Dória and Hennet (1999) for determining so-called \( \lambda \)-contractive sets determines a CI under-approximation of the MCI set in a finite number of iterations. Thus the problem of under-approximating \( \mathcal{X}^{*} \) is decidable, but the number of iterations is unknown and no error bound on the level of approximation is obtained.

Assume the knowledge of a CI set \( \mathcal{C} \subset \mathcal{C}^{*} \) satisfying \( 0 \subset \mathcal{C} \) and consider the sequence \( \{ \Psi_{j} \subset \mathbb{R}^{m}_{\text{in}} \}_{j=0}^{\infty} \) defined by: \( \Psi_{0} := \mathcal{C}, \Psi_{j+1} := \text{Reach}(\Psi_{j}) j \in \mathbb{N} \). The following statements hold: \( \forall i j, i \in \mathbb{N}, \mathcal{C} \subset \Psi_{j} \subset \mathcal{C}^{*} \). By construction every \( \Psi_{j} \) is a CI set. This is the method for determining CI \( j \)-step admissible sets described in Gutman and Cwiokel and Keerthi and Gilbert (1987). In Gutman and Cwiokel and Keerthi and Gilbert (1987) only controllable modes are considered, and the initial set \( \mathcal{C} \) is the origin. By Assumption 1 the system here is only stabilizable. Thus an initial CI set containing the origin within its interior is chosen.
\( x_i^{[1]} = A x_i^{[1]} + \sum_{j=0}^{i-1} A^{i-1-j} B u_j^{[1]} \in C^* \),

\( E x_i^{[1]} + G u_i^{[1]} \leq W \).

Consider some \( K \in \mathbb{R}^{n \times n} \) s.t. \( \rho(A + BK) < 1 \). Such a \( K \) exists because \( (A, B) \) is stabilizable by Assumption 1. Consider applying the stabilizing state feedback control law \( u_i = K x_i \). This control law is allowed to ignore the constraints. The following statements hold \( \forall i \in \mathbb{N}^+ \):

\[
\begin{align*}
x_i^{[2]} &= A x_i^{[2]} + \sum_{j=0}^{i-1} A^{i-1-j} B u_j^{[2]} = (A + BK) x_i^{[1]}.

\end{align*}
\]

Because \( \rho(A + BK) < 1 \) then \( \lim_{i \to \infty} x_i^{[2]} = 0 \). Furthermore, \( \forall \epsilon > 0 \) \( \exists \delta > 0 \) s.t. \( \sum_{j=0}^{i} A^{i-j} B [\lambda (\mu_{j} u_j^{[1]} + (1-\mu_j) u_j^{[2]})] \leq \epsilon \).

Consider some \( \lambda \in (0, 1) \) and \( x_0 = \lambda^0 \mu^{[1]} + (1-\lambda) u_0 \). Then it is possible to choose some \( \mu \in (0, 1) \) s.t. \( \lambda \mu + (1-\mu) L \leq 1 \). It is desirable to choose \( \mu \) as small as possible. Consider the control move sequence \( u_{[1]} = \lambda [\mu x_{[1]} + (1-\mu) u_1] \in \mathbb{N} \) and corresponding closed-loop state trajectory:

\[
\begin{align*}
x_i &= A x_i + \sum_{j=0}^{i-1} A^{i-1-j} B u_j^{[1]} \in C^* \forall i \in \mathbb{N}.

\end{align*}
\]

Thus the closed-loop state trajectory satisfies \( x_i \in C^* \) \( \forall i \in \mathbb{N} \). Further consider the constraints. The following statements hold \( \forall i \in \mathbb{N} \):

\[
\begin{align*}
E x_i + G u_i^{[1]} &= E \left[ \lambda (\mu x_i^{[1]} + (1-\mu) x_i) \right] \\
&\geq \lambda \left[ \mu E x_i^{[1]} + (1-\mu) E u_i^{[1]} \right] \\
&\geq \lambda \left[ \mu W + (1-\mu) L W \right] \\
&\leq \lambda [\mu W + (1-\mu) L W] W < W.

\end{align*}
\]

Thus the control input trajectory \( u_{[1]} \) is feasible at each state \( x_i \). The following holds:

\[
\begin{align*}
\lim_{i \to \infty} x_i &= \lambda \mu x_i^{[1]} \in \lambda \mu C^* \subseteq C^*.

\end{align*}
\]

Thus the number of iterations \( \tilde{j} \) of the MCIF sets is tabulated in Table 1. Two computation times are stated; \( T \) is the time to compute sequence \( \{\Phi_i\}_{i=0}^{\infty} \), i.e., the cumulative time to compute Pre(\cdot); \( T_{all} \) is the
Algorithm 14

\( \dot{\theta}_1 = \frac{1}{J_1} \left[ (\theta_2 - \theta_1)k + (\dot{\theta}_2 - \dot{\theta}_1)v \right] \)
\( \dot{\theta}_2 = \frac{1}{J_2} \left[ (\theta_1 - \theta_2)k + (\dot{\theta}_1 - \dot{\theta}_2)v + u \right] . \)  

For \( M_1 \), the MCIF set \( \mathbb{R}^2 \) is equal to the MCI set \( \mathbb{C}^* \) for all \( N \), which is equal to the MCIF set for \( M_2 \) and \( N = 1 \). These results are thus useful for comparing the changes in computational complexity due to the problem size alone, as the output set is identical for all \( N \).

The number of iterations decreases inversely with increasing \( N \) until settling at 1, as expected. Interestingly, computing the same MCI(F) set in one iterations using \( N = 10 \) is about twice as fast as computing it in ten iterations using \( N = 1 \). Thus MCIF sets may offer a path for the rapid determination of usual MCI sets. For \( M_2 \), large and erratic variations in iteration number and computation times can be seen. However, the volume of the MCIF set decreases monotonically with increasing \( N \). This is because for a particular \( N \), states close to the boundary of the MCIF set become infeasible as \( N \) is increased while the number of degrees of freedom \( \tilde{N} \) is held constant.

### 5.2. Undecidable example

The purpose of this example is to demonstrate the undecidability property and application of Algorithm 14 to approximate MCIF sets. Consider system (1) with \( A \) := diag(1.5, 0.9) and \( B \) := diag(0.5, 0.9), where the sub-system defined by \( (A, \varphi) \) is the triple integrator \( C(s) = s^{-1} \) with sample-period \( h = 0.5 \) s \( n = 4, m = 2 \). Further compute state constraints \( |x|_{\infty} \leq 2 \) and control input constraints \( |u|_{\infty} \leq 1 \). The problem of computing the MCIF set for any pair \((N, N)\) is undecidable. Consider only the mode of dimension 1. The sets \( \{x \in \mathbb{R}^4 | x_1 = -1 \} \) are stationary under minimum (maximum) control input \( \{u \in \mathbb{R}^2 | u_1 = -1 \} \) \( \{u \in \mathbb{R}^2 | u_1 = 1 \} \). The Pre(·)/Reach(·) operators thus cannot remove/add elements of such sets from/to the parameter sets passed to them. Thus a fixed-point is never reached when computing sequences \( \{\Phi_j\}_{j=0}^{\infty} \) and \( \{\Psi_j\}_{j=0}^{\infty} \).

A CIF under-approximation \( \tilde{\Phi}^* \) of the MCIF set was computed for \( \epsilon = 1 \), \( N \in \mathbb{N}_1 \) and blocking matrices \( M_1 = I_{\tilde{N}}, M_2 = [1, \ldots, 1]^T \in (1)^N \). For each \( N \) the number of iterations, computation time and error bound \( E \) at the final iteration are tabulated in Table 2. The algorithm was initialized using \( \Psi_0 = \{x_0 \in \mathbb{R}^2 | x_1 = -1 \} \) \( \{x_0 \in \mathbb{R}^2 | x_1 = 1 \} \). \( \tilde{\Phi}^* \) is thus computed using Algorithm 14 for different pairs \((N, M)\), but does not necessarily result in the lowest number of iterations and computation time. For \( M_2 \) and large \( N \) the required iterations and computation time become very high. This is due to the very small size of \( \Psi_0 \) when using the above method of automatic initialization.

### Table 1

<table>
<thead>
<tr>
<th>( j )</th>
<th>( M_1 )</th>
<th>( T )</th>
<th>( E )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.0</td>
<td>0.77</td>
<td>6.39</td>
</tr>
<tr>
<td>1</td>
<td>0.93</td>
<td>6.43</td>
<td>9.74</td>
</tr>
<tr>
<td>2</td>
<td>1.17</td>
<td>6.86</td>
<td>8.31</td>
</tr>
<tr>
<td>3</td>
<td>1.48</td>
<td>5.40</td>
<td>16.8</td>
</tr>
<tr>
<td>4</td>
<td>1.76</td>
<td>6.17</td>
<td>9.43</td>
</tr>
<tr>
<td>5</td>
<td>2.33</td>
<td>6.23</td>
<td>6.21</td>
</tr>
<tr>
<td>6</td>
<td>2.56</td>
<td>5.92</td>
<td>1.27</td>
</tr>
<tr>
<td>7</td>
<td>2.38</td>
<td>4.10</td>
<td>1.14</td>
</tr>
<tr>
<td>8</td>
<td>1.98</td>
<td>3.77</td>
<td>0.87</td>
</tr>
<tr>
<td>9</td>
<td>2.61</td>
<td>4.31</td>
<td>32.3</td>
</tr>
<tr>
<td>10</td>
<td>3.41</td>
<td>5.17</td>
<td>173</td>
</tr>
<tr>
<td>11</td>
<td>4.30</td>
<td>6.00</td>
<td>178</td>
</tr>
<tr>
<td>12</td>
<td>5.37</td>
<td>7.13</td>
<td>602</td>
</tr>
<tr>
<td>13</td>
<td>6.55</td>
<td>8.31</td>
<td>428</td>
</tr>
<tr>
<td>14</td>
<td>7.87</td>
<td>9.66</td>
<td>428</td>
</tr>
<tr>
<td>15</td>
<td>9.26</td>
<td>11.0</td>
<td>326</td>
</tr>
<tr>
<td>16</td>
<td>10.8</td>
<td>12.5</td>
<td>198</td>
</tr>
<tr>
<td>17</td>
<td>12.6</td>
<td>13.6</td>
<td>115</td>
</tr>
<tr>
<td>18</td>
<td>14.6</td>
<td>16.4</td>
<td>579</td>
</tr>
<tr>
<td>19</td>
<td>16.6</td>
<td>18.4</td>
<td>415</td>
</tr>
<tr>
<td>20</td>
<td>18.9</td>
<td>20.5</td>
<td>359</td>
</tr>
<tr>
<td>21</td>
<td>21.4</td>
<td>23.2</td>
<td>39 ( 340 )</td>
</tr>
<tr>
<td>22</td>
<td>24.3</td>
<td>25.0</td>
<td>37.5</td>
</tr>
<tr>
<td>23</td>
<td>27.3</td>
<td>29.1</td>
<td>39.3</td>
</tr>
<tr>
<td>24</td>
<td>30.7</td>
<td>32.5</td>
<td>39.2</td>
</tr>
<tr>
<td>25</td>
<td>34.5</td>
<td>36.2</td>
<td>68.3</td>
</tr>
<tr>
<td>26</td>
<td>38.6</td>
<td>40.5</td>
<td>52.3</td>
</tr>
<tr>
<td>27</td>
<td>42.7</td>
<td>44.5</td>
<td>56.5</td>
</tr>
<tr>
<td>28</td>
<td>48.0</td>
<td>49.7</td>
<td>75.0</td>
</tr>
</tbody>
</table>

### Table 2

<table>
<thead>
<tr>
<th>( j )</th>
<th>( M_1 )</th>
<th>( T )</th>
<th>( E )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### 6. Conclusion

Enforcing strong feasibility via CIF constraints is so far the only method capable of enforcing strong feasibility in time-invariant MPC problems employing move-blocking or irregular prediction horizon time-discretization. The well-known CI terminal constraint method for enforcing strong feasibility in usual, full degree of freedom MPC problems is a special case of the CIF constraint approach proposed here. Thus the proposed method is a generalization, not specialization, of current terminal constraint methods. The explicit use of MCIF sets is deemed indispensable in the design of least restrictive, low-complexity MPC controllers which rigorously enforce constraint satisfaction. The success of the proposed
method for linear systems is motivation to extend this method to more general systems and constraints.

Acknowledgements

Ravi Gondhalekar gratefully acknowledges funding received through the Program of Promotion of Environmental Improvement to Enhance Young Researchers’ Independence under the Special Coordination Funds for Promoting Science and Technology, Japan Ministry of Education, Culture, Sports, Science and Technology.

References


Ravi Gondhalekar was born in Boston, USA, in 1979. He received the B.A. and M.Eng. degrees in engineering in 2002 from the University of Cambridge, UK, and the Ph.D. degree in informatics in 2008 from the Tokyo Institute of Technology, Japan. He is currently an Assistant Professor at Osaka University, Japan. His research interests include model predictive control, constrained control, set invariance and hybrid systems. Ravi has been employed at the Massachusetts Institute of Technology, the University of Cambridge, Princeton University, Pi Technology, the Rutherford Appleton Laboratory and the United Kingdom Atomic Energy Authority.

Jun-ichi Imura was born in Gifu, Japan, in 1964. He received the M.S. degree in applied systems science, and the Ph.D. degree in mechanical engineering from Kyoto University, Japan, in 1990 and 1995, respectively. He served as a Research Associate at the Department of Mechanical Engineering, Kyoto University from 1992 to 1996, and as an Associate Professor at the Division of Machine Design Engineering, Faculty of Engineering, Hiroshima University from 1996 to 2001. From May 1998 to April 1999, he was a visiting researcher at the Faculty of Mathematical Sciences, University of Twente, The Netherlands. Since 2001, he has been with the Department of Mechanical and Environmental Informatics, Graduate School of Information Science and Engineering, Tokyo Institute of Technology, where he is currently a Professor. His research interests include control of nonlinear systems and analysis and control of hybrid systems. Ravi has been an Associate Editor of Automatica, and an Associate Editor of SICE Journal of Control, Measurement, and System Integration. He is a member of IEEE, SICE, ISCIE, IEICE, and the Robotics Society of Japan.

Kenji Kashima was born in 1977 in Oita, Japan. He received his Bachelor’s degree in engineering and his Master’s and Doctoral degrees in informatics from Kyoto University, Japan, in 2000, 2002 and 2005, respectively. He is currently an Assistant Professor at the Graduate School of Information Science and Engineering, Tokyo Institute of Technology. His research interests include general theory for infinite-dimensional systems and stochastic systems, and their applications focusing on emerging areas.