2024 European Control Conference (ECC)
June 25-28, 2024. Stockholm, Sweden

On the complexity of computing the Maximal Positive Invariant set

Bogdan Gheorghe[1], Florin Stoican[1] and Ionela Prodan[2]

***Abstract*— Computing the maximal (robust) positive invariant (M(R)PI) set for linear dynamics and a polyhedral constraint set is well-known in the literature but, the effects and limitations of the different methods employed are not sufficiently clear, especially for high dimensional systems.**

In this paper we propose a systematic analysis of the existing techniques as well as the application of new ideas to accelerate the computation of the MPI set. This includes new stop conditions for the set recurrence that spans it. We analyze and compare these variations over a dynamical system whose dimension can be arbitrarily increased to draw conclusions about their relative strengths and weaknesses.

***Index Terms*— maximal positive invariant set; half-space description; minimal representation; model predictive control**

I. Introduction

The Model Predictive Control (MPC) algorithm is widely used in academia and industry because of its capacity to explicitly account for constraints, cost and model nonlinearities [1]. Arguably, the main theoretical difficulty with MPC is to the ensure stability and recursive feasibility [2], [3].

To ensure these properties, modern approaches require a triplet of local controller, terminal cost and terminal set [4]. For the later, the maximal positive invariant (MPI) set is the usual choice (for the the robust case, its counterpart, the maximal robust positive invariant (MRPI) set is used). Computing it entails significant effort: the time required to obtain it, the memory required for storing, and the computational burden added in the constrained optimization problem associated with the MPC algorithm [5]. It is worth mentioning that there are approaches which try to avoid the inherent complexity of the MPI computations. E.g., by working with implicit forms via a sequence of sets [6], by fixing the complexity [7] or, even, by prolonging the prediction horizon sufficiently such as to avoid the need of a terminal set [8], [9]. Needless to say, the difficulties of considering nonlinear dynamics increase exponentially with few existing results in the literature [10].

While the standard recurrence sequence which provides the MPI set in the linear case is both conceptually simple and well-understood [6], [11], it is not always clear which particular variant is superior in what regards computation time and complexity of representation. Specifically, we have identified several common shortcomings:

i) the stop condition in the MPI set recurrence is given as a set inclusion test; checking it, if not done efficiently, can be extremely cumbersome;
ii) often the sets involved are symmetric w.r.t. the origin, fact which is not exploited in the state of the art;
iii) the different implementations give MPI sets which often have redundant descriptions; this makes them difficult to store and to integrate into the MPC scheme.

We tackle these issues through the following elements:

i) we use a variant of Farkas' Lemma to check set inclusion using only their half-space descriptions (as has been done in a different context, e.g., in [12], [13]);
ii) we exploit the sets' symmetry to reduce the computation time via a simplified linear program which uses the cross-polytope's peculiarities [14];
iii) we propose a quasy-minimal representation scheme which bridges the gap between the sufficient and exact stop conditions; we later use the same tools to provide a MPI set in full minimal representation;
iv) we show how the same ideas may be used for the bounded-disturbance case for both the minimal and the maximal robust positive invariant sets.

Fair conclusions cannot really be drawn from small-dimension examples where the computation time and complexity are muddled by platform specific issues (e.g., function overhead, particular solver used). Hence, we analyze all the MPI computation methods discussed over a benchmark whose dimension can be varied arbitrarily [15].

The rest of the paper is organized as follows. Section II introduces basic notions about set theory and the standard MPC problem. Section III details the variants of MPI construction and extensions for the bounded-disturbance case. Section V analyzes the relative performance of the proposed schemes and Section VI draws the conclusions.

Notation. $O_{m\times n}\in\mathbb{R}^{m\times n}$ is the matrix with m rows and n columns whose entries are zero. Whenever $m=n$, we use the shorthand notation O_n. $I_n\in\mathbb{R}^{n\times n}$ is the identity matrix whose diagonal elements are one and zero otherwise. For an arbitrary matrix $G\in\mathbb{R}^{m\times n}$, G_i denotes its i-th column and $G_j^\top$ its j-th row. For two sets, X and Y, their Minkowski sum is defined as $X\oplus Y=\{x+y:\ \forall x\in X,\forall y\in Y\}$ and their Pontryagin difference is defined as $X\ominus Y=\{x\in X: x+y\in X,\forall y\in Y\}$. $|A|$, with $A\in\mathbb{R}^{m\times n}$, represents the element-wise absolute value of A. For a vector $x\in\mathbb{R}^n$, its infinity norm is given as $\|x\|_\infty := \max(|x_1|,|x_2|,\dots,|x_n|)$.

[1]Bogdan Gheorghe, Florin Stoican are with Faculty of Automation Control and Computer Science, National University of Science and Technology Politehnica Bucharest, Romania {bogdan.gheorghe1807, florin.stoican}@upb.ro

[2]Ionela Prodan is with the Univ. Grenoble Alpes, Grenoble INP†, LCIS, F-26000, Valence, France, † Institute of Engineering and Management Univ. Grenoble Alpes. ionela.prodan@lcis.grenoble-inp.fr

* The work of the first author was supported by a grant from the National Program for Research of the National Association of Technical Universities - GNAC ARUT 2023; Project ID: 207, UNSTPB, Romania. The second and third authors' research benefits from the support of the FMJH Program PGMO and from the support to this program from EDF, France.

II. PRELIMINARIES

In this section we introduce several standard notions from set-based control, see [14], [16] for further details.

A. Set prerequisites

A bounded and fully-dimensional polyhedron $X \subset \mathbb{R}^d$, has a dual representation, with both a half-space form [16]

$$X = \{x \in \mathbb{R}^d : F_{X,i}^\top x \leq \theta_{X,i}, i = 1 \dots n_h\}, \tag{1}$$

as an intersection of linear inequalities and a convex sum of its extreme points (i.e., its vertices)

$$X = \{x \in \mathbb{R}^d : x = \sum_{j=1}^{n_v} \alpha_j v_j, \sum_{j=1}^{n_v} \alpha_j = 1, \alpha_j \geq 0\}. \tag{2}$$

The pairs $(F_{X,i}^\top, \theta_{X,i}) \in \mathbb{R}^d \times \mathbb{R}$ denote the inequalities of the *half-space* representation and $v_j \in \mathbb{R}^d$ denote the vertices of the *vertex* representation. When pairing a dynamical system with a set, several notions can be described [17].

Definition 1 (robust positive invariance)**.** *A set* Ω *is* robust positive invariant (RPI) *for dynamics* $x_{k+1} = f(x_k, w_k)$ *iff* $x_k \in \Omega \implies x_{k'} \in \Omega,\ \forall k' > k$ *holds for any bounded disturbance* $w_k \in W$*. Equivalently stated in set form, inclusion* $f(\Omega, W) \subseteq \Omega$ *has to hold.*
We call Ω*, a* positive invariant (PI) *set when the dynamics are without disturbances (i.e.,* $x_{k+1} = f(x_k, w_k = 0)$*).* ♦

Among the infinity of PI/RPI sets associated with a dynamic as in Definition 1, we recall the following two cases.

Definition 2. *The* minimum robust positive invariant (mRPI) *is the RPI set that is contained in every closed RPI set of the dynamics. Equivalently, it is the fix point of the set recurrence* $\Omega_0^m = \{0\}$, $\Omega_{k+1}^m = f(\Omega_k^m, W)$*, i.e.,* $\Omega_\infty^m = f(\Omega_\infty^m, W)$. ♦

Definition 3. *When the state is bounded by a set* X*, the largest*[1] *admissible PI (RPI) set contained within it, is called the maximal (robust) positive invariant (M(R)PI) set. Thus,* $\Omega \subseteq f(\Omega, W)$ *and* $\Omega \subseteq X$ *have to hold.* ♦

B. The MPC problem

Consider the linear discrete-time dynamics

$$x_{k+1} = Ax_k + Bu_k, \tag{3}$$

where $x_k \in \mathbb{R}^n$ denotes the state vector, $u_k \in \mathbb{R}^m$, the input vector and matrices $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$ model the dynamics. In the model predictive control (MPC) problem, the control action u_k is obtained by repeatedly solving a constrained optimization [5] of the form:

$$\min_{\bar{u}_0,\dots,\bar{u}_{N-1}} \sum_{i=0}^{N-1} \ell(\bar{x}_i, \bar{u}_i, r_i) + T(\bar{x}_N, r_N) \tag{4a}$$

$$\text{s.t.} \quad \bar{x}_{i+1} = A\bar{x}_i + B\bar{u}_i,\ \forall i = 0, \dots, N-1, \tag{4b}$$

$$\bar{x}_0 = x_k, \tag{4c}$$

$$\bar{x}_N \in \Omega, \tag{4d}$$

$$\bar{u}_i \in \mathcal{U},\ \bar{x}_{i+1} \in \mathcal{X},\ \forall i = 0, \dots, N-1. \tag{4e}$$

[1]In (6), the sets respect a monotonous inclusion. Thus, the ''largest set' is unambiguously the newest, regardless of the measure considered.

State and input constraint sets $\mathcal{X} \subset \mathbb{R}^n$, $\mathcal{U} \subset \mathbb{R}^m$ and terminal set $\Omega \subset \mathbb{R}^n$ are polyhedra, often given in half-space representation (1). The constrained optimization (4):

- predicts a sequence of states $\{\bar{x}_1, \dots, \bar{x}_N\}$, updated through dynamics (4b) by a suitable (resulted from solving (4)) sequence of inputs $\{\bar{u}_0, \dots, \bar{u}_{N-1}\}$;
- such that it minimizes a stage cost $\ell(\bar{x}_i, \bar{u}_i, r_i)$ (which usually penalizes a combination of states x_k, inputs u_k and references r_k) and a terminal cost $T(\bar{x}_N, r_N)$ (penalizing the last predicted state w.r.t. reference r_N);
- while also verifying stage input/state constraints (4e) and a terminal state constraint (4d);
- (4) is integrated in the control scheme, by: i) initializing the prediction with the current plant state value (constraint (4c)), and, ii) applying to the plant dynamics (3) the first element of the input sequence $u_k \hookleftarrow \bar{u}_0$.

While MPC often works well in practice, it still has the risk of failing to find a solution (i.e., (4) is infeasible). The standard approach, [4], is to provide a terminal cost, $T(\cdot, \cdot)$, and a terminal set, Ω, inclusion condition which is guaranteed to hold under the local control law, $\bar{u}_i = K\bar{x}_i$, $\forall i \geq N$, after the N steps of the prediction horizon. Assuming that the terminal set is reachable, i.e., that $\bar{x}_N \in \Omega$, we may enumerate the recursive feasibility conditions [4]:

i) Ω is PI for the local controller: $(A + BK)\Omega \subseteq \Omega$;
ii) state constraints are respected: $\Omega \subseteq X$;
iii) input constraints are respected under the local control law $\bar{u}_i = K\bar{x}_i$: $K\bar{x}_i \in U, \forall x_i \in \Omega$, $\forall i \geq N$;
iv) the stage cost $\ell(\bar{x}_N, \bar{u}_N, r_N)$ and the terminal cost $T(\bar{x}_N, r_N)$ verify a local Lyapunov function condition:

$$\frac{dT(\bar{x}_N, r_N)}{dt} + l(\bar{x}_N, \bar{u}_N, r_N) \leq 0, \forall\, \bar{x}_N \in \Omega.$$

Once the local control law is selected, the key element is the choice of Ω. Condition $\bar{x}_N \in \Omega$ restricts the range of values for the initial state $\bar{x}_0$, it has to stay in the backward reachable set of length N, Ψ_N:

$$\bar{x}_0 \in \Psi_N, \quad \Psi_{i+1} = A^{-1}\Psi_i \oplus A^{-1}B(-U), \quad \Psi_0 = \Omega.$$

Thus, among the infinity of PI/RPI sets, it is reasonable to select the largest admissible one, i.e., the MPI/MRPI set of the closed-loop (with the local controller) dynamics.

The intricacies of computing, storing and using the MPI set are discussed in detail in the rest of the paper.

III. MPI CONSTRUCTION

Recall that at the end of the prediction horizon ($k \geq N$) in the MPC problem (4), the control action switches to a fixed feedback law $u_k = Kx_k$ which has to ensure closed-loop stability and admissibility ($x_k \in \mathcal{X}, u_k \in \mathcal{U}$). To simplify the notation for this section, we note $A_\circ := A + BK$ and $\overline{\mathcal{X}} := \mathcal{X} \cap \{x :\ Kx \in \mathcal{U}\}$ to obtain the LTI dynamics

$$x_{k+1} = A_\circ x_k \tag{5}$$

with matrix $A_\circ \in \mathbb{R}^{n\times n}$ and a set $\overline{\mathcal{X}} \subset \mathbb{R}^n$ which bounds the state. Let us recall the standard recurrence for MPI (maximal positive invariant) set construction [17], [18]:

$$\Omega_0 = \overline{\mathcal{X}}, \quad \Omega_{k+1} = A_\circ^{-1}\Omega_k \cap \overline{\mathcal{X}}. \tag{6}$$

Under mild and reasonable assumptions [6, Thm. 3], recurrence (6) is guaranteed to stop by arriving at a fix point $\Omega_{\bar{k}} = \Omega_{\bar{k}+1}$, for some finite index $\bar{k}$.

Since by construction (see also (8)) we have that $\Omega_{k+1} \subseteq \Omega_k$ for all k, it suffices to check that [18]

$$\Omega_k \subseteq \Omega_{k+1}. \tag{7}$$

Equivalently, recurrence (6) may be written[2] as

$$\Omega_{k+1} = \bigcap_{j=0}^{k+1} \mathcal{Y}_j = \bigcap_{j=0}^{k+1} A_\circ^{-j}\overline{\mathcal{X}}. \tag{8}$$

Consequently, the stop condition (7) becomes [6], [17]

$$\Omega_k \subseteq \mathcal{Y}_{k+1} \Leftrightarrow \Omega_k \subseteq A_\circ^{-(k+1)}\overline{\mathcal{X}}. \tag{9}$$

Often, conditions (7), (9) are replaced by the sufficient condition (by way of noting that $\Omega_k \subseteq \overline{\mathcal{X}}$) [6], [18]:

$$\overline{\mathcal{X}} \subseteq A_\circ^{-(k+1)}\overline{\mathcal{X}}. \tag{10}$$

A. Stop condition verification

All set inclusion conditions variants ((7), (9) and (10)) may be verified through multiple methods. To keep the notation simple, let us consider P, Q where P stands for either Ω_k or $\overline{\mathcal{X}}$ and Q denotes Ω_{k+1} or $A_\circ^{-(k+1)}\overline{\mathcal{X}}$.

Considering P, Q polytopic sets defined as in Sec. II, the classic idea is to check whether the vertices of the left operand verify the constraints defining the right operand:

$$F_{Q,i}^\top v_{P,j} \leq \theta_{Q,i}, i = 1 \dots n_P, \forall v_j \in Q. \tag{11}$$

Well-known is also the result based on Farkas's Lemma [16] which involves only the half-space representation of the polytopes.

Proposition 1. *Checking inclusion $P \subseteq Q$ for P, Q given as in (1) is equivalent with the feasibility of the following relations:*

$$\exists H \geq 0,\ HF_Q = F_P,\ H\theta_P \leq \theta_Q. \tag{12}$$

□

This result is extended to the symmetric case next.

Corollary 1. *Assume $P = \{x \in \mathbb{R}^n : |\bar{F}_P x| \leq \bar{\theta}_P\}$ and $Q = \{x \in \mathbb{R}^n : |\bar{F}_Q x| \leq \bar{\theta}_Q\}$. Then $P \subseteq Q$ holds iff*

$$\exists \bar{H},\ \bar{H}\bar{F}_Q = \bar{F}_P,\ |\bar{H}|\bar{\theta}_P \leq \bar{\theta}_Q, \tag{13}$$

with the pairs $(\bar{F}_P, \bar{\theta}_P) \in \mathbb{R}^{\bar{q}_P \times n} \times \mathbb{R}^{\bar{q}_P}$ and $(\bar{F}_Q, \bar{\theta}_Q) \in \mathbb{R}^{\bar{q}_Q \times n} \times \mathbb{R}^{\bar{q}_Q}$. □

Proof: Re-arranging the matrices of the half-space representation of P, Q in the form used by Proposition 1 leads to (13). See [19, Prop. 2] for a similar treatment. ■

[2]Hereinafter we assume that A is invertible for simplicity. Otherwise, applying a singular value decomposition and ancillary manipulations allows to carry the set inclusion tests.

B. Checking inclusion $|\bar{H}|\bar{\theta}_P \leq \bar{\theta}_Q$

Inclusion $|\bar{H}|\theta_P \leq \theta_Q$ with $\bar{H} \in \mathbb{R}^{\bar{q}_Q \times \bar{q}_P}$ and $\bar{\theta}_P \in \mathbb{R}^{\bar{q}_P}_{\geq 0}$, $\bar{\theta}_Q \in \mathbb{R}^{\bar{q}_Q}_{\geq 0}$ has a deceptively simple form but expressing it without the explicit use of the nonlinear and non-smooth module operator is not obvious. As a first step, let us take $\bar{H}_i^\top$, the i-th row of matrix $\bar{H}$, which has to check $|\bar{H}_i^\top|\bar{\theta}_P \leq \bar{\theta}_{Q,i}$, or, equivalently put, has to verify

$$\sum_{j=1}^{\bar{q}_P} |\bar{H}_{ij}|\bar{\theta}_{P,j} \leq \bar{\theta}_{Q,i}. \tag{14}$$

Using the facts that[3] $\bar{\theta}_P \geq 0$ and the module's product property (i.e, $|ab| = |a| \cdot |b|$), we reformulate (14) into

$$\sum_{j=1}^{\bar{q}_P} \left| \bar{H}_{ij} \cdot \frac{\bar{\theta}_{P,j}}{\bar{\theta}_{Q,i}} \right| \leq 1. \tag{15}$$

The lhs of inequality (15) is the 1-norm of the vector $\tilde{H}_i^\top := \begin{bmatrix} \dots & \bar{H}_{ij} \cdot \bar{\theta}_{P,j}/\bar{\theta}_{Q,i} & \dots \end{bmatrix}^\top$. Hence, all points $\tilde{H}_i \in \mathbb{R}^{\bar{q}_P}$ that verify (15) are those inside the unit ball of the 1-norm (i.e., the cross-polytope of dimension $\bar{q}_P$, denoted hereafter with $\mathrm{CR}_{\bar{q}_P}$). Checking $\tilde{H}_i \in \mathrm{CR}_{\bar{q}_P}$ is clearly linear but with a caveat. Namely, while usually the number of half-spaces is significantly less than the number of vertices for a random polytope, the cross-polytope is the textbook example of the opposite, that is, $\mathrm{CR}_{\bar{q}_P}$ has $2\bar{q}_P$ vertices but $2^{\bar{q}_P}$ half-spaces. Thus, to check $\tilde{H}_i \in \mathrm{CR}_{\bar{q}_P}$ we prefer to verify that $\tilde{H}_i$ can be written as a convex sum of $\mathrm{CR}_{\bar{q}_P}$'s vertices rather than checking that all constraints are verified. This leads to:

$$\tilde{H}_i = \sum_{j=1}^{\bar{q}_P} V_j^{i,-}\alpha_j^{i,-} + V_j^{i,+}\alpha_j^{i,+}, \tag{16a}$$

$$\alpha_j^{i,-}, \alpha_j^{i,+} \geq 0, \quad \sum_{j=1}^{\bar{q}_P} \alpha_j^{i,-} + \alpha_j^{i,+} = 1, \tag{16b}$$

where $V_j^{i,\pm} := \begin{bmatrix} \dots \underbrace{\pm\bar{\theta}_{Q,i}/\bar{\theta}_{P,j}}_{j\text{-th index}} \dots \end{bmatrix}^\top$ denotes the vertices of the cross-polytope $\mathrm{CR}_{\bar{q}_P}$. Repeating (16) for each of the $\bar{q}_Q$ rows of $\bar{H}$ concludes the implementation.

Remark 1. *The symmetric polytope considered in Corollary 1 may be treated as a generic polytope by taking $F_{P/Q} \hookleftarrow \begin{bmatrix} \bar{F}_{P/Q}^\top & -\bar{F}_{P/Q}^\top \end{bmatrix}^\top$ and $\theta_{P/Q} \hookleftarrow \begin{bmatrix} \bar{\theta}_{P/Q}^\top & \bar{\theta}_{P/Q}^\top \end{bmatrix}^\top$ and applying Proposition 1. Note also the link between the number of constraints: $\bar{q}_P = q_P/2$, $\bar{q}_Q = q_Q/2$. Comparing, we observe that, as illustrated in the table below,*

case	*# vars.*	*# eqs.*	*# ineqs.*
eq. (12)	$q_P q_Q$	nq_P	$q_Q + q_P q_Q$
eqs. (13)+ (16)	$3q_P q_Q/4$	$(nq_P + q_Q)/2$	$(q_Q + q_P q_Q)/2$

explicitly exploiting the symmetric case gives a more compact set of relations (fewer constraints and decision variables) which, when embedded into a larger optimization problem, leads to a reduced computational time. ♦

[3]It comes by having P, Q from Cor. 1 as polytopes containing the origin.

C. Quasi-minimal representation of the MPI set

Regardless of the stop condition employed or the method chosen for validating it, the MPI set is the result of recurrence (6), or, equivalently, (8). In half-space representation, the end-result is a set having $\bar{k}\cdot q$ constraints (where q denotes the number of constraints for set $\overline{\mathcal{X}}$ and $\bar{k}$ is the index at which the recurrence stops). This is a non-minimal representation since many of the inequalities may be redundant (they do not change the shape of the set and could be discarded without any loss). This is especially problematic for the sufficient stop condition (10) which often stops at an index $\bar{k}' > \bar{k}$. All blocks of q constraints added in steps $\bar{k}+1 \dots \bar{k}'$ are fully redundant (no constraint therein will change the set's shape) and should be removed from the set's description. Hence, to be fair in comparing the computation times for the MPI construction, we apply in Algorithm 1 a partial redundancy check to arrive at the same complexity of representation.

Algorithm 1: Quasi-minimal representation

Input: MPI set $\Omega = \bigcap\limits_{j=0}^{\bar{k}'} A_\circ^{-j}\overline{\mathcal{X}}$,

$\overline{\mathcal{X}} = \{x \in \mathbb{R}^n : F_x x \leq \theta_x\}$

Output: index $\bar{k}$

1 $k = \bar{k}'$, flag=1;

2 **while** *flag* $\neq 0$ **do**

3 $\quad F = \left[F_x^\top \quad \dots \quad \left(A_\circ^{k-1}F_x\right)^\top\right]^\top$,
$\quad \theta = \left[\theta_x^\top \quad \dots \quad \theta_x^\top\right]^\top$;

4 $\quad$ **for** $j = 1$ **to** q **do**

5 $\quad\quad$ **if** *isNotRedundant*(F, θ, $A_\circ^k F_{x,j}$, $\theta_{x,j}$) **then**

6 $\quad\quad\quad$ flag=0;

7 $\quad\quad$ **end**

8 $\quad$ **end**

9 $\quad k = k - 1$;

10 **end**

11 $\bar{k} = k$;

The algorithm tests each block of q constraints going backwards from $\bar{k}'$. As long as the k-th block is fully redundant (the 'flag' variable is never made 0 in step 6), the index k is decremented (step 9) and the block is removed from the description at the next iteration (step 3). The exact stop index $\bar{k}$ is obtained when exiting the 'while' loop (step 11). The 'isNotRedundant($F, \theta, c^\top, d$)' function called at step 5 checks whether the constraint $\{c^\top x \leq d\}$ is redundant with respect to the polyhedral set $\{x \in \mathbb{R}^n : \ Fx \leq \theta\}$. This is done by solving the two linear programs (LPs)

$$d^{\star,\pm} = \arg\min_{x} \quad \pm c^\top x, \quad \text{s.t.} \quad Fx \leq \theta, \tag{17}$$

and checking whether

$$\min\left(d^{\star,-}, d^{\star,+}\right) \leq d \leq \max\left(d^{\star,-}, d^{\star,+}\right)$$

holds. If so, the constraint is not redundant and the flag returns '1'. We may reduce to a single LP per constraint in the case of symmetric polytopes (because $d^{\star,+} = -d^{\star,-}$).

Remark 2. *We call the procedure a 'quasi-minimization' because we stop pruning constraints at block $\bar{k}$. To obtain a fully minimal half-space representation, we may carry the check until all redundant inequalities are eliminated (e.g., change the 'while' stop condition to '$k \geq 1$').* ♦

Remark 3. *Eliminating redundant inequalities needs not to be a post-processing step. It can actually be carried during the set recurrence* (6)*, in which case it functions as another stop condition: whenever the newly added block of q constraints has no irredundant ones, we stop.* ♦

IV. Extensions

The same reasoning may be applied to the robust case by extending dynamics (5):

$$x_{k+1} = A_\circ x_k + \delta_k, \tag{18}$$

with $\delta_k \in \Delta = \{x : \ F_\Delta x \leq \theta_\Delta\} \subset \mathbb{R}^n$.

A. The mRPI case

Let us recall the set sequence of RPI approximations of the *minimal robust positive invariant (mRPI)* set associated with dynamics (18), as given in [20].

Proposition 2 (Thms. 1 and 3 from [20])**.** *If $0 \in int\,\Delta$ and there exists a pair $s \in \mathbb{N}, \alpha \in [0, 1)$ such that*

$$A_\circ^s \Delta \subseteq \alpha\Delta, \tag{19}$$

and a scalar $\epsilon > 0$ such that

$$\epsilon = \arg\min_{\gamma}\left(\frac{\alpha}{1-\alpha}\bigoplus_{\ell=0}^{s} A_\circ^\ell \Delta \subseteq \mathbb{B}_p^n(\gamma)\right), \tag{20}$$

then

$$\Omega(\alpha, s) = \frac{1}{1-\alpha}\bigoplus_{\ell=0}^{s} A_\circ^\ell \Delta, \tag{21}$$

is an RPI approximation to the mRPI set associated with (18)*, and inclusion $\Omega_\infty \subseteq \Omega(\alpha, s) \subseteq \Omega_\infty \oplus \mathbb{B}_p^n(\epsilon)$ holds.* □

The key (computation-wise at least) element in Prop. 2 is finding the smallest scalar α such that (19) holds. Fixing s to a known value and applying Prop. 1 gives

$$\begin{aligned} \alpha^\star(s) = \min_{\alpha} \ & \alpha \\ \text{s.t. } & \exists H \geq 0,\ HF_\Delta A_\circ^{-s} = F_\Delta,\ H\theta_\Delta \leq \alpha\theta_\Delta. \end{aligned} \tag{22}$$

B. The MRPI case

In the presence of disturbances, the set recurrence (6) changes to [11]:

$$\Omega_0 = \overline{\mathcal{X}}, \quad \Omega_{k+1} = \left[A_\circ^{-1}(\Omega_k \ominus \Delta)\right] \cap \overline{\mathcal{X}}, \tag{23}$$

and (8) becomes

$$\Omega_{k+1} = \bigcap_{j=0}^{k+1} \mathcal{Y}_j^\delta = \bigcap_{j=0}^{k+1} A_\circ^{-j}\left(\overline{\mathcal{X}} \ominus \bigoplus_{\ell=0}^{j-1} A_\circ^\ell \Delta\right). \tag{24}$$

Thus (see [11, Lemma 1] for Pontryagin difference properties), the corresponding variant of (9) becomes

$$\Omega_k \subseteq \mathcal{Y}_{k+1}^{\delta} \Leftrightarrow \Omega_k \subseteq A_\circ^{-(k+1)}\left(\overline{\mathcal{X}} \ominus \bigoplus_{\ell=0}^{k} A_\circ^{\ell}\Delta\right). \tag{25}$$

Using (21) we obtain a sufficient condition for (25):

$$\Omega_k \subseteq A_\circ^{-(k+1)}\left(\overline{\mathcal{X}} \ominus \frac{1}{1-\alpha}\bigoplus_{\ell=0}^{s} A_\circ^{\ell}\Delta\right). \tag{26}$$

Remark 4. *By using the RPI approximation* (21) *in* (25) *we decouple the current recurrence step 'k' from the number of iterations 's', used for the mRPI approximation. This allows us to control the complexity of the right-hand term in* (26).

A further approximation, which keeps the complexity of the original (without disturbance) condition (9)*, is to find the smallest* μ *which verifies* $\Omega(\alpha, s) \subseteq \mu\overline{\mathcal{X}}$*. Then, a sufficient condition (exploiting relation* $A_\circ \ominus \lambda A_\circ = (1-\lambda)A_\circ, \forall\lambda \in (0,1)$*) for* (26) *is:*

$$\Omega_k \subseteq (1-\mu)A_\circ^{-(k+1)}\overline{\mathcal{X}}. \tag{27}$$

Either (26) *or* (27) *may easily be put in the framework of Section III, or even adapted for condition* (10). ♦

V. Illustrative example

For illustration purposes we use here the 'CSE' example from the $COMPl_eib$ [15] whose state-space representation is proportional with a parameter that can be changed arbitrarily. This will serve us well in the subsequent analysis of computation time and complexity of representation.

A. CSE example from [15]

The 'CSE' dynamics describe a system combining coupled springs, dampers and masses. The mass positions and velocities define the system state and its input are the two forces exerted at the ends of the coupled springs chain [15]. The continuous-time model is given by

$$\dot{x} = \underbrace{\begin{bmatrix} 0 & I \\ -M_c^{-1}K_c & -M_c^{-1}L_c \end{bmatrix}}_{A\in\mathbb{R}^{2\ell\times 2\ell}} x + \underbrace{\begin{bmatrix} 0 \\ M_c^{-1}D_c \end{bmatrix}}_{B\in\mathbb{R}^{2\ell\times 2}} u \tag{28}$$

, where $M_c = \mu I$, $L_c = \delta I$,

$$K_c = k\begin{bmatrix} 1 & -1 & \cdots & 0 & 0 \\ -1 & -2 & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \ddots & -2 & -1 \\ 0 & 0 & \cdots & -1 & 1 \end{bmatrix}, \text{ and } D_c = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & -1 \end{bmatrix}.$$

The numerical values considered are $\mu = 4$, $\delta = 1$, $k = 1$ and the resulted system is discretized with the forward Euler method for a sampling time of 1 sec. The state and input constraints (later used in the MPC) are:

$$\mathcal{X} = \{x \in \mathbb{R}^{2\ell} : \|x\|_\infty \leq 1\}, \mathcal{U} = \{u \in \mathbb{R}^2 : \|u\|_\infty \leq 1\}. \tag{29}$$

B. MPI analysis

In what follows we check multiple pairs of the stop test conditions and validation methods for the MPI set's computation. Specifically, we consider the following scenarios:

stop condition	validation method		
	eq. (11)	eq. (12)	eqs. (13) + (16)
$\Omega_k \subseteq \Omega_{k+1}$	S1)	S2)	S3)
$\Omega_k \subseteq A_\circ^{-(k+1)}\overline{\mathcal{X}}$	S4)	S5)	S6)
$\overline{\mathcal{X}} \subseteq A_\circ^{-(k+1)}\overline{\mathcal{X}}$	S7)	S8)	S9)

to which we add:

S10) the default method of MPT3 [21], arguably the state of the art in Matlab-based MPC computations ();

S11) the half-space redundancy check, as discussed in Remark 3 ().

We construct for each of these scenarios the MPI set associated with dynamics (28) and constraint set

$$\overline{\mathcal{X}} := \mathcal{X} \cap \{x \in \mathbb{R}^{2\ell} : Kx \in \mathcal{U}\}, \tag{30}$$

using the sets (29) and the stabilizing state feedback K obtained by solving the LQR problem for the state and input cost matrices $Q = I_{2\ell}$, $R = I_2$. Each scenario run is stopped after one hour, but we allowed the running of two more identical scenarios with immediate higher sizes. The results are illustrated in Fig. 1 over the interval $\ell \in \{2, \ldots, 30\}$ in a vertical logarithmic scaling. We draw several conclusions:

- small dimensions ($\ell \in \{2, \ldots, 5\}$) do not exhibit clear trends due to, most probably, spending more time with pre-processing steps and function calls;
- regardless of the sets considered, using vertices as in (11), i.e., scenarios S1), S4) and S7), fails quickly ($\ell \approx 7$) due to the exponential increase in vertices with number of constraints and, especially, dimension (McMullen's upper bound theorem [14] gives a worst case of $q^{\lfloor \ell \rfloor}$ vertices);
- the sufficient condition (10), used in scenarios S7-9), is significantly faster than either of the exact ones (with an order of magnitude at $\ell = 15$), i.e., at scenarios S4-6);
- exploiting the problem's symmetry reduces the computation time for the sufficient condition (10), e.g., comparing S8) and S9) we observe that $\frac{t_{S8} - t_{S9}}{t_{S8}} \cdot 100 \in (13, 46)\%$, with an average of 32%;
- counter-intuitively, for test condition (9), the non-symmetric scenario, S5), is more efficient than the symmetric scenario, S6);
- scenario S11) behaves quite efficiently: after $\ell = 6$ it becomes more efficient than any of the other scenarios tested. As the relative performance depends also on the complexity of the initial set $\Omega_0 = \overline{\mathcal{X}}$, we cannot claim that S11) will always win.

Not all MPI sets are created equal. That is, while geometrically all scenarios return the same object, its complexity will differ. The exact stop conditions (7) and (9), employed in

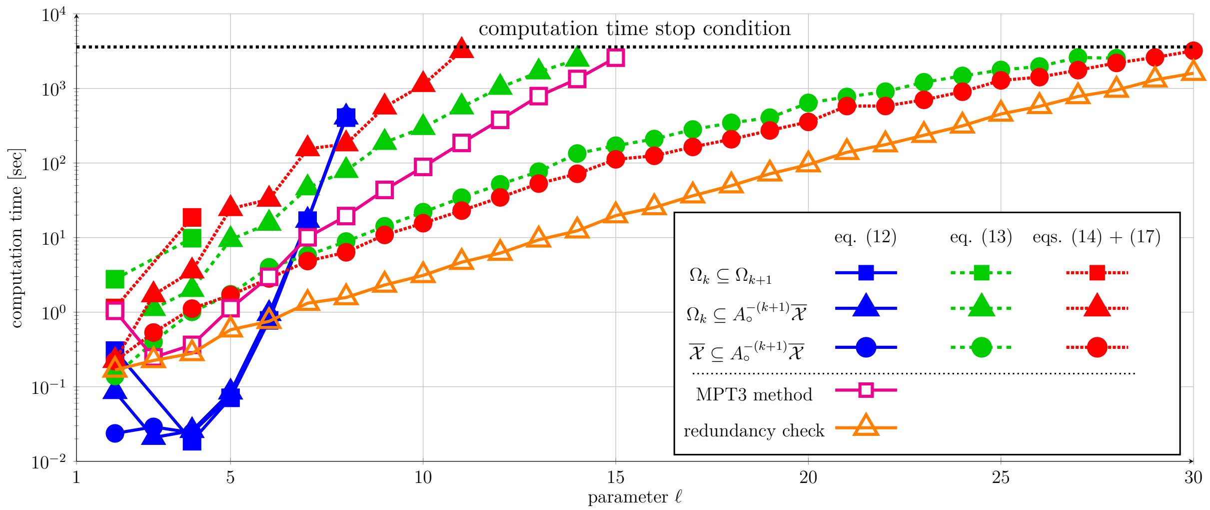
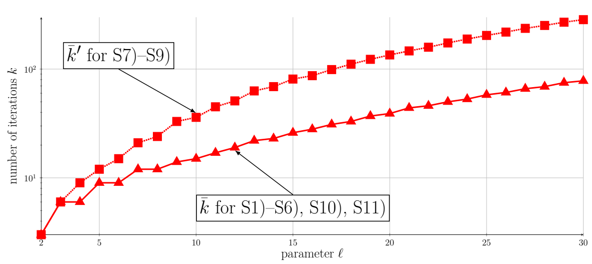
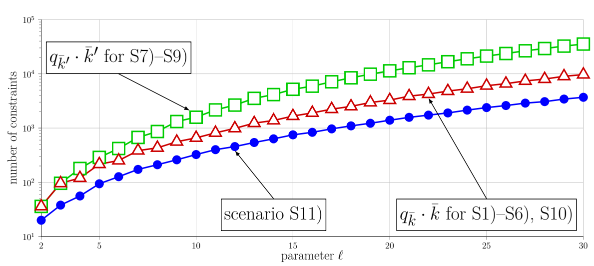


Fig. 1: MPI computation times for scenarios S1) – S11)

S1)–S6), will iterate until $\bar{k}$ and the sufficient stop condition (10), employed in S7)–S9), goes until $\bar{k}' \geq \bar{k}$ in the set recurrence (6). In all these cases no minimal representation is sought, hence the number of constraints[4] will be either $q_\ell \cdot \bar{k}$ or $q_\ell \cdot \bar{k}'$ with deleterious effects in solving the MPC problem in which the MPI set is integrated. On the other hand, Remark 3 employed in S11) stops at $\bar{k}$ and directly provides a minimal (with only irredundant constraints) representation.



(a) number of iterations



(b) number of constraints

Fig. 2: Number of iterations and constraints for the MPI set.

To better grasp the numbers involved we show in Fig. 2 both the number of iterations ($\bar{k}$ and $\bar{k}'$) and the number of constraints after parameter ℓ. Fig. 2a divides the scenarios into those that stop after $\bar{k}$ iterations (via (7), (9), MPT3 or Remark 3) and those that stop after $\bar{k}'$ iterations (via (10)). As seen in Fig. 2b iterating for additional steps greatly increases the number of constraints in the description. Even if one of the exact stop conditions is used, many of the constraints are still redundant (as shown when comparing the set returned by S11) with the one returned by either of S1-S6) or S10)). E.g., at $\ell = 20$ we observe that the exact methods stop at $\bar{k} = 39$ and the sufficient one at $\bar{k}' = 135$. Furthermore, the number of constraints is 1492, 3276, 11340 for, respectively, i) exact and minimal description, ii) exact with redundant constraints and iii) sufficient with redundant constraints.

Computing a fully minimal representation as from S11) may prove too much hassle. We may ask what is the cost of a partial minimization, as the one carried in Algorithm 1 which eliminates the blocks from $\bar{k}+1, \ldots, \bar{k}'$ and thus brings the MPI sets obtained in S7)–S9) into the form returned by S1)–S6). We compare in Fig. 3 the computation time for scenarios S5), S10) and S11) which all stop at iteration $\bar{k}$ and S8') in which we add to the time spent in solving S8) the time spent running Algorithm 1. We observe that, at least in this case, the quasy-minimization (solid blue line, square markers) is better than the methods which stop through an exact condition. The only scenario which is (at least at higher dimensions) better, is S11) where redundancies are eliminated at each iteration of (6). Note that we have neither exploited symmetries nor tweaked the standard LP problems (17) so further improvements should be possible.

C. MPC implementation

We should recall that the justification for constructing the MPI set in the first place was its use in the MPC problem (4). We consider a quadratic cost MPC problem where the penalty matrices for a stage cost are $Q = I_{2\ell}$ and $R = I_2$, the terminal cost penalty matrix P and local control law K are computed accordingly. The prediction horizon is $N_{\text{pred}} = 10$, and the simulation horizon is $N_{\text{sim}} = 100$ steps. The MPC implements a variant of the scheme proposed in [22].

[4]The number of constraints in $\overline{\mathcal{X}}$, defined as in (30), depends on the value of parameter ℓ. Without a redundancy check, and considering box state/input magnitude bounds, we arrive at $q_\ell = 4\ell + 4$.

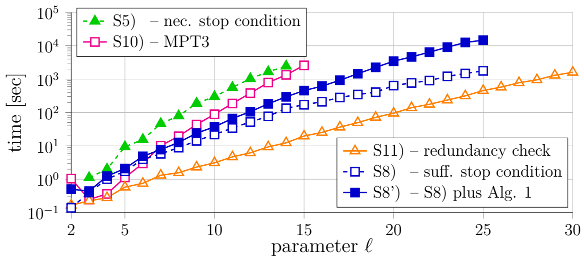


Fig. 3: Computation time for exact condition scenarios versus sufficient scenario plus partial constraint removal.

Thereafter, we consider several cases of implementation for the MPC problem from the viewpoint of the MPI set:

C1) without the MPI set (this serves as a comparison and should not be used in general as it lacks the required recursive feasibility guarantees induced by the MPI set);

C2) with the non-minimal MPI set obtained with one of the exact stop conditions (thus, the stop index is $\bar{k}$);

C3) the same as C2) but for the MPI obtained with the sufficient stop condition (10), i.e., the stop index is $\bar{k}'$;

C4) with the fully-minimal MPI set (as per Remark 2).

Problem size is not a perfect indicator of computation time since solver heuristics greatly influence it (e.g., the tolerance values considered, internal structure of the problem, specific algorithm employed). We depict in Fig. 4 the total run time for solving the MPC problem in each of the aforementioned cases, with parameter $\ell \in \{2, \dots, 15\}$.

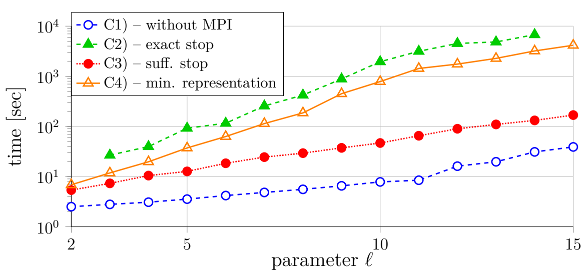


Fig. 4: Computation time for the MPC problem.

For C2) we used the MPI set from S5), for C3) the one from S8), and for C4) the one from S11). As expected, C1) is clearly the fastest. Conclusions become more muddled for the three MPI-using cases. Still, we observe that C4) is better than C2) but, surprisingly, worse than C3). The later in fact we expected to be the worst performing among the considered cases. These results require further investigation.

All the results have been obtained with the help of MPT3 [21] and CasADi [23].

VI. Conclusions

This paper considered exact and sufficient stop inclusion conditions for the set recurrence leading to the maximal positive invariant set associated with linear dynamics and a polyhedral constraint set. We analyzed the computation time and representation complexity for various methods of validating the inclusion test (with and without exploiting the sets' symmetry). Finally, we analyzed the effect of minimal and non-minimal representations and the case of dynamics affected by bounded disturbances.

References

[1] S. Sun, A. Romero, P. Foehn, E. Kaufmann, and D. Scaramuzza, "A Comparative Study of Nonlinear MPC and Differential-Flatness-Based Control for Quadrotor Agile Flight," *IEEE Transactions on Robotics*, vol. 38, no. 6, p. 3357–3373, Dec. 2022.

[2] S. V. Raković and S. Zhang, "Model predictive control with implicit terminal ingredients," *Automatica*, vol. 151, p. 110942, 2023.

[3] N. T. Nguyen and I. Prodan, "Stabilizing a multicopter using an NMPC design with a relaxed terminal region," *IFAC-PapersOnLine*, vol. 54, no. 6, pp. 126–132, 2021.

[4] H. Chen and F. Allgower, "A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability," *Automatica*, vol. 34, no. 10, pp. 1205–1217, 1998.

[5] S. V. Raković and W. S. Levine, *Handbook of model predictive control*. Springer, 2018.

[6] S. V. Raković and S. Zhang, "The implicit maximal positively invariant set," *IEEE Tran. on Automatic Control*, 2022.

[7] M. Lazar and M. Tetteroo, "Computation of terminal costs and sets for discrete–time nonlinear MPC," *IFAC-PapersOnLine*, vol. 51, no. 20, pp. 141–146, 2018.

[8] L. Grüne, J. Pannek, L. Grüne, and J. Pannek, "Stability and suboptimality without stabilizing constraints," *Nonlinear Model Predictive Control: Theory and Algorithms*, pp. 113–163, 2011.

[9] F. Rußwurm, W. Esterhuizen, K. Worthmann, and S. Streif, "On MPC without terminal conditions for dynamic non-holonomic robots," *IFAC-PapersOnLine*, vol. 54, no. 6, pp. 133–138, 2021.

[10] W. Esterhuizen, T. Aschenbruck, and S. Streif, "On maximal robust positively invariant sets in constrained nonlinear systems," *Automatica*, vol. 119, p. 109044, 2020.

[11] S. Rakovic and M. Fiacchini, "Invariant Approximations of the Maximal Invariant Set or "Encircling the Square"," in *Proc. 17th IFAC World Congress IFAC*, vol. 8, 2008, pp. 6377–6382.

[12] G. Bitsoris, "Positively invariant polyhedral sets of discrete-time linear systems," *International Journal of Control*, vol. 47, no. 6, pp. 1713–1726, 1988.

[13] S. Sadraddini and R. Tedrake, "Linear encodings for polytope containment problems," in *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE, 2019, p. 4367–4372.

[14] K. Fukuda, "Polyhedral computation," 2020, publisher: Department of Mathematics, Institute of Theoretical Computer Science ETH Zurich.

[15] F. Leibfritz, "COMPleib: COnstrained Matrix–optimization Problem library – a collection of test examples for nonlinear semidefinite programs, control system design and related problems," 2006.

[16] G. M. Ziegler, *Lectures on polytopes*. Springer Science & Business Media, 2012, vol. 152.

[17] F. Blanchini and S. Miani, *Set-theoretic methods in control*. Springer, 2008, vol. 78.

[18] E. Gilbert and K. Tan, "Linear systems with state and control constraints: the theory and application of maximal output admissible sets," *IEEE Tran. on Automatic Control*, vol. 36, no. 9, pp. 1008–1020, 1991.

[19] B. Gheorghe, F. Stoican, and I. Prodan, "A variable terminal set NMPC construction: application to multicopter stabilisation," in *2023 European Control Conference (ECC)*. IEEE, 2023, pp. 1–6.

[20] S. V. Rakovic, E. C. Kerrigan, K. I. Kouramas, and D. Q. Mayne, "Invariant approximations of the minimal robust positively invariant set," *IEEE Tran. on Automatic Control*, vol. 50, no. 3, pp. 406–410, 2005.

[21] M. Herceg, M. Kvasnica, C. Jones, and M. Morari, "Multi-Parametric Toolbox 3.0," in *Proc. of the European Control Conference*, Zürich, Switzerland, July 17–19 2013, pp. 502–510.

[22] D. Simon, J. Löfberg, and T. Glad, "Reference tracking mpc using dynamic terminal set transformation," *IEEE Tran. on Automatic Control*, vol. 59, no. 10, pp. 2790–2795, 2014.

[23] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi: a software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.