Reduced Parameterisation MPC for Input-Constrained Unstable Linear Systems Part 1: Implementation

Adrian Medioli*, Maria Seron*, Richard Middleton†

*The ARC Centre for Complex Dynamic Systems & Control, The University of Newcastle, Australia
†The Hamilton Institute, NUI Maynooth, Co Kildare, Ireland

Abstract—This paper introduces a new variant of model predictive control called Reduced Parameterisation Model Predictive Control. The new algorithm uses the structure of the null controllable set of input constrained unstable systems to produce a closed-loop system with a region of attraction that is an arbitrarily close approximation to this set. In addition, this result is achieved with a lower complexity than that of MPC.

I. INTRODUCTION

A common problem faced by the implementation of Model Predictive Control (MPC) is that the complexity of the optimisation is related to the length of the control horizon. However, in general longer control horizons lead to larger regions of attraction for the closed-loop system. Hence a compromise governs the MPC horizon’s selection. In this paper, the first of two, we propose a variation of the MPC formulation that provides the benefits of long horizons without the increased computational complexity.

We specifically address the application of MPC to linear, input constrained1, systems with two2 real unstable modes. It has been shown in for example [2], [3] and [4], that input constrained unstable systems have an associated controllable region that exists irrespective of the control strategy used and commonly referred to as the null controllable region.

When an initial state near the boundary of the null controllable region is selected, the control action required to stabilise the system contains several sequences of equally saturated control moves. From this observation we propose that a single saturated control applied for a longer duration can replace each sequence of equally saturated controls and produce an equivalent control action. This approach to reducing complexity has been investigated by other authors and their techniques have been collectively named move blocking strategies (see [5], and references contained therein, for an overview). Move blocking strategies constrain or block inputs or states to reduce the number of optimisation parameters. For input constrained unstable systems this equates to substituting sequences of control moves with a single control move of a longer duration.

The main difference between the approach presented here and move blocking strategies, is that here the structure of the null controllable set is used to determine the control substitutions. We will refer to the resulting substituted sequence as a reduced parameterisation (RP) sequence.

Adopting this replacement policy for saturated control sequences, it is possible to significantly expand the region of attraction of a system while only increasing the number of MPC decision variables by a relatively small amount. More specifically, for systems having p distinct real unstable poles, [1, Corollary 5.2], has shown that the number of parameters in the RP is at most (2p − 1), thus independent of the length of the MPC horizon.

In this paper we present a new control strategy, based on the use of RP sequences, which we call Reduced Parameterisation MPC (RPMPC). The strategy consists of a recovery mode and a convergence mode. When operating in recovery mode, RP sequences are used, while convergence mode employs a user selected horizon for which a small-sized full parameterisation MPC is used (i.e., a standard MPC algorithm is used). By introducing recovery mode, the length of the overall MPC horizon and hence its region of attraction can be expanded to a large size without the addition of a large number of decision variables.

The paper is organised in several sections describing the systems to which RPMPC is applied, the implementation steps and an example showing the resulting behaviour of the closed-loop system. In a companion paper [6], the convergence properties, stability, worst case and average computational requirements of RPMPC are presented. The work presented here has been extended in [1] to systems with an arbitrary number of real unstable poles. For simplicity of exposition, we have chosen to develop here the case p = 2.

II. SYSTEMS AND CONTROLLABLE SETS

We consider a class of single input, open-loop unstable, linear systems described by the following discrete-time state-space model,

\[ x_{k+1} = Ax_k + Bu_k , \]

where

\[ x_k = \begin{bmatrix} x_{k|u}^T \\ x_{k|s}^T \end{bmatrix}^T , \]

\[ A = \begin{bmatrix} A_u & 0 \\ 0 & A_s \end{bmatrix} ; \quad B = \begin{bmatrix} B_u \\ B_s \end{bmatrix} , \]

\[ u_k \in \mathbb{U} \triangleq [u_{min}, u_{max}] , \]

and \( A_u = \text{diag}(a_{u1}, a_{u2}) \) contains 2 real eigenvalues such that \( a_{u2} > a_{u1} > 1 \), \( A_s \) is a \( q \times q \) matrix whose eigenvalues...
have modulus less than or equal to one, \(B_u \in \mathbb{R}^2, B_b \in \mathbb{R}^n\). The input saturation values satisfy \(u_{\text{min}} < 0\) and \(u_{\text{max}} > 0\), and \((A_u, B_u)\) is controllable.

We will denote the null controllable set for (1)-(3) as \(\mathcal{R}_{\text{max}}\), which by the systems’ structure can also be defined as the Cartesian product of an unstable \(\mathcal{R}_{u,\text{max}}\) and stable \(\mathcal{R}_{b,\text{max}}\) subsets. That is, \(\mathcal{R}_{\text{max}} = \mathcal{R}_{u,\text{max}} \times \mathcal{R}_{b,\text{max}}\). Further, [4, Proposition 1] shows that \(\mathcal{R}_{u,\text{max}}\) is unbounded and therefore the analysis, as presented here, only considers \(\mathcal{R}_{b,\text{max}}\).

From [7], [8], [4], \(\mathcal{R}_{u,\text{max}}\) is given by

\[
\mathcal{R}_{u,\text{max}} = \left\{ x_0 \in \mathbb{R}^2 : \lim_{k \to \infty} A_u x_k + B_u u_k = 0 \right\}
\]

for \(u_k \in U, k = 0, 1, \ldots \). \(\mathcal{R}_{u,\text{max}}\)

The work in [1] shows that \(\mathcal{R}_{u,\text{max}}\) can be arbitrarily approximated by an \(N\)-step controllable set, denoted by \(\mathcal{R}_N\) and defined as

\[
\mathcal{R}_N = \left\{ x \in \mathbb{R}^2 : x = -\sum_{i=0}^{N-1} A_u^{-i-1} B_u u_i, \right\}
\]

for \(u_i \in U, i = 0, \ldots, N - 1\), with the selection of an arbitrarily large horizon \(N\).

Detailed analysis of \(N\)-step controllable sets presented in [1] shows that they can be represented as a Minkowski sum (denoted by \(\oplus\)) of line segments called generators. More specifically each of the generators used to construct \(\mathcal{R}_N\) are given the term core generators.

**Definition 1 (Core Generator):** The \(i\)th core generator, is the line segment \(g_i = \{ x \in \mathbb{R}^2 : x = g_i u_i \} \) for \(u_i \in U \) for \(i = 0, \ldots, N-1\), where \(U\) is as defined in (3) and the core generator direction \(\tilde{g}_i\) is given by

\[
\tilde{g}_i = -A_u^{-i-1} B_u.
\]

Then, defining the set of core generators as \(G^N = \{ g_0, \ldots, g_{N-1} \}\), we have that \(G^N = \mathcal{R}_N\).

The boundary of \(\mathcal{R}_N\) is the set of extreme points that can be controlled to the origin in \(N\) steps. The edges of this boundary can be described in the following way.

**Definition 2 (Edge Sets):** Let \(\mathcal{R}_N \subset \mathbb{R}^2\) be such that \(\mathcal{R}_N = S^\oplus (G^N)\), where all generators in \(G^N\) are in general position.\(^3\) Then any edge \(\mathcal{E}_i^N\) of \(\mathcal{R}_N\) has the following structure,

\[
\mathcal{E}_i^N = S^\oplus (G_i^{b,N} \oplus z_i^{e,N}),
\]

where \(G_i^{b,N} \subset G^N\) is termed the “edge basis” generator set and \(z_i^{e,N} \in \mathbb{R}^2\) is the “edge offset”. Further we note that \(G_i^{b,N}\) contains a single generator.

**Definition 3 (Core Edge):** An edge whose basis generator is a core generator as per Definition 1 is termed a “core edge”.

As explained the concept of RP is what allows the expansion of a controllable set to be independent of the horizon length. Hence, we introduce some definitions relating to RP sets.

**Definition 4 (RP Sequence):** An RP sequence \(T_{RP}\) is a set of sub-intervals from the interval \(I \triangleq \{0, \ldots, N-1\}\), defined as follows:

\[
T_{RP} \triangleq \{ T_i^{RP} : T_i^{RP} = [s^{RP}(i), s^{RP}(i + 1) - 1], \}
\]

where \(r \leq N\) is the number of parameters in the RP sequence. In addition the elements of \(T_{RP}\) are ordered by the relation \(s^{RP}(i) < s^{RP}(i + 1)\) for \(i = 0, \ldots, r - 2\).

**Definition 5 (RP Generators):** Given an RP index sequence \(T_{RP} = \{ T_0^{RP}, \ldots, T_r^{RP} \}\) as per Definition 4, the associated RP generators \(g_i^{RP}, i = 0, \ldots, r-1\), are the line segments

\[
g_i^{RP} = \{ x \in \mathbb{R}^2 : x = g_i^{RP} u_i \}
\]

for \(u_i \in U\). Therefore, the new set of generators that describe the RP set is given by the following definition.

**Definition 6 (RP N-step Controllable Set):** Given an RP index sequence \(T_{RP}\) as per Definition 4 and the associated RP generators \(g_i^{RP}, i = 0, \ldots, r - 1\), as per Definition 5, the resulting RP generator set \(G_{RP}\) is given by \(G_{RP}(T_{RP}) = \{ g_0^{RP}, \ldots, g_{r-1}^{RP} \}\). Hence, the associated RP set, \(\mathcal{R}_{RP}\), is given by,

\[
\mathcal{R}_{RP} = S^\oplus (G_{RP}(T_{RP})) = S^\oplus (\{ g_0^{RP}, \ldots, g_{r-1}^{RP} \})
\]

**Definition 7 (Composite Generator):** Given a sequential index set \(I \subset T\) such that \(|I| > 1\), the associated composite generator \(G^I\) is the line segment,

\[
g^I = \{ x \in \mathbb{R}^2 : x = g^I u \}
\]

for \(u \in U\). Therefore, \(G^I\) is defined as

\[
G^I = \{ g_0^{RP}, \ldots, g_{r-1}^{RP} \}
\]

**Definition 8 (Composite Edge):** An edge whose basis generator is a composite generator as per Definition 7 is termed a “composite edge”.

In the following section we will describe the RPMC algorithm, which employs a particular type of RP sequences, and associated RP sets, derived from the generator structure of \(N\)-step controllable sets.

### III. RPMPC Algorithm

As in conventional MPC (see, for example [9]), RPMPC is based on solving a finite horizon optimal control problem at each sampling time. The cost function associated with this problem is

\[
V(X, U, N) \triangleq F(x_N) + \sum_{i=0}^{N-1} L(x_i, u_i)
\]

where \(X = \{ x_0, x_1, \ldots, x_N \}, U = \{ u_0, u_1, \ldots, u_{N-1} \}, N\) is the horizon length, \(F(x_N) = \frac{1}{2} x_N^T P x_N, P = P^T > 0\), represents the terminal cost and \(L(x_i, u_i) = \frac{1}{2} (x_i^T Q x_i + u_i^T R u_i)\), \(Q = Q^T > 0, R = R^T > 0\), for \(i = 0, \ldots, N - 1\), is the per stage cost. For consistency with
the definition of the systems (1)–(3), we partition $Q$ and $P$ into stable and unstable sub-matrices as follows:

$$Q = \begin{bmatrix} Q_u & 0 \\ 0 & Q_s \end{bmatrix}, \quad P = \begin{bmatrix} P_u & 0 \\ 0 & P_s \end{bmatrix}. \tag{13}$$

RPMPC departs from conventional MPC in that, at each iteration $k$ of the RPMPC algorithm, a new horizon $N = N_k$ is dynamically determined as

$$N_k = \text{max} (N_{\text{min}}, N_F), \tag{14}$$

where $N_{\text{min}}$ is the recovery mode horizon whose computation is explained in Section III-A below and $N_F$ is a user defined fixed horizon employed in the convergence mode of RPMPC.

The state will be constrained to the set $\mathcal{X} = \{ R_{N_{\text{max}}} \times R_{s_{\text{max}}} \}$, where $R_{N_{\text{max}}}$ has the form (5) and $R_{s_{\text{max}}}$ is a user-selected maximum horizon to approximate the null controllable set. In addition, equality constraints are introduced in the optimisation problem to reflect the fact that the $N$-move control sequence $U$ is substituted by an RP sequence as per Definition 4.

Thus, the optimisation problem that RPMPC solves at each iteration is,

$$\mathcal{P}_{N_k}(x) : \quad V^{OPT}_{N,k}(x) = \min V(\mathcal{X}, U, N), \tag{15}$$

subject to:

$$N = N_k, \tag{16}$$

$$x_{i+1} = Ax_i + Bu_i \quad \text{for} \quad i = 0, \ldots, N - 1, \tag{17}$$

$$x_0 = x, \tag{18}$$

$$u_j = u_l \quad \forall j, l \in \mathcal{I}_{RP} \quad \text{for} \quad i = 0, \ldots, |\mathcal{I}_{RP}|-1, \tag{19}$$

$$u_i \in U \quad \text{for} \quad i = 0, \ldots, N - 1, \tag{20}$$

$$x_i \in \mathcal{X} \quad \text{for} \quad i = 0, \ldots, N, \tag{21}$$

$$x_N \in \mathcal{X}_f \subset \mathcal{X}, \tag{22}$$

where $U$ is as in (3), $\mathcal{X}_f = \{ x_N \in \mathcal{X} : x_{N|u=0} = 0 \}$ is the terminal set$^4$ and $\mathcal{I}_{RP}^{\mathcal{X}}$ is the RP sequence whose precise structure will be explained below in Definition 9. Each set of indices $\mathcal{I}_{RP} \subset \mathcal{I}_{RP}$ for $i = 0, \ldots, |\mathcal{I}_{RP}|-1$ introduces a sequence of equality constraints to the optimisation problem. Further for the case of $N_k = N_F$ (convergence mode), $\mathcal{I}_{RP} = \emptyset$, or equivalently all control input equality constraints are removed.

The solution to the optimisation problem (15)–(22) results in the optimal control sequence,

$$U^{OPT}_{k} = \{ u_0^{OPT}, u_1^{OPT}, \ldots, u_{N_{k}-1}^{OPT} \}. \tag{23}$$

Then the control applied by RPMPC at time $k$ is the first element of the optimal control sequence $U^{OPT}_{k}$ defined in (23). That is, we set $u_k = u_0^{OPT}$. We summarise the RPMPC algorithm as follows.

Algorithm 1 (RPMPC): Given a user selected horizon $N_{max}$ to approximate $R_{max}$ and the user selected horizon $N_F$ for the convergence mode, determine the horizon $N_{min}$ used in (14), (15) and the RP index set $\mathcal{I}_{RP}^{\mathcal{X}}$ used in (19), and solve (15)–(22) at each time $k$ to obtain (23) for the current state $x = x_k \in \mathcal{X}$. Set $u_k = u_0^{OPT}$. \hfill \Box

In the following subsections we describe the algorithms used by RPMPC to obtain the horizon $N_{min}$ and the RP set $\mathcal{I}_{RP}^{\mathcal{X}}$.

A. Determination of $N_{min}$

The recovery horizon $N_{min}$ is defined as

$$N_{min} = \text{min} \{ N \in [N_F, N_{max}] : x_k \in \mathcal{X} \}, \tag{24}$$

with $\mathcal{X}$ of the form (5). The determination of $N_{min}$ is implemented combining a binary search algorithm and a set membership determination procedure explained in Section III-B below. The binary search is implemented as follows.

Algorithm 2 (Search for $N_{min}$): Produce a sequence of test horizons $N_t$ using a binary search over the range $[N_F, N_{max}] \subset \mathbb{N}$ and test for $x_k \in \mathcal{X}$ until $N_{min} = N_t$ such that $x_k \in \mathcal{X}_{N_{min}}$ and $x_k \notin \mathcal{X}_{N_{min}+1}$.

B. Set Membership and Reduced Parameterisation

This subsection describes both a technique for obtaining an RP set $\mathcal{I}_{RP}^{\mathcal{X}}$ used in (19) for the control sequence and the elements required to determine if $x_k \in \mathcal{X}$ (Algorithm 2).

1) Selecting a Reduced Parameterisation: Firstly we present a result from [1] describing the RP sequence with the smallest number of parameters that allows any state on an edge of $\mathcal{R}^N$ to be controlled to the origin in $N$ steps.

Definition 9 (RP Minimal Sequence): Let the edge $e_i^N$ of the $N$-step controllable set $\mathcal{R}^N$ be given by (7). Let the index of the basis generator from the set $G_{i}^{b,N}$ be denoted by $b$, that is $G_{i}^{b,N} = \{ g_b \} \subset \mathcal{G}^N$. Then the minimal RP sequence, is given by the index set

$$\mathcal{I}_{RP}^{\mathcal{X}} = \{ [0, b - 1], b, [b + 1, N - 1] \}, \tag{25}$$

where intervals $[i, j]$ having $i > j$ are removed from $\mathcal{I}_{RP}^{\mathcal{X}}$. In essence the definition constructs an RP sequence set that satisfies the control sequence property established in [1, Corollary 5.1]. This result states that for an initial state on an edge of $\mathcal{R}^N$ a feasible control sequence consists of all control moves $\{ u_0, \ldots, u_{b-1} \}$ having the same saturated value and all control moves $\{ u_{b+1}, \ldots, u_{N-1} \}$ having the complementary saturated value.

Having identified an RP that provides an admissible control sequence for all states of a particular edge, the following theorem shows that this same technique can be used to determine an RP sequence for any state in $\mathcal{R}^N$.

Theorem 1 (RP for Any Initial State): Given a test state $x^t \in \mathcal{R}^N$, let $e_i^N$ be an edge through which a ray emanating from the origin through $x^t$ exits $\mathcal{R}^N$. Consider the RP sequence $\mathcal{T}_{RP} = \mathcal{T}_{k}$ produced by Definition 9 for $e_i^N$, and the associated RP $N$-step controllable set $\mathcal{R}^N_{k}$ as per Definition 6. Then $x^t \in \mathcal{R}^N_{k}$.

Proof: See [1, Theorem 6.1].

\hfill \Box
The implication of Theorem 1 for the RPMPC algorithm, is that a valid RP is given by any RP polytope $\mathcal{R}^{RP}$ that includes the edge of $\mathcal{R}^N$ through which a ray from the origin in the direction of the current state $x_k$ exits. Therefore Definition 9 in combination with Definition 6 readily produce an RP set for any state $x_k \in \mathcal{R}^N$.

2) Membership Determination Algorithm: Using the concepts of generators and RP sequences, we now present an algorithm that allows the membership determination and subsequently the RP to be calculated via the dynamic construction of a series of test polytopes.

The proposed algorithm, uses a test ray $r^t$ emanating from the origin through the state $x^t$. To limit the number of edges that must be tested to find the edge through which $r^t$ exits $\mathcal{R}^N$, we employ RP sets. At each iteration $k$ the aim is to determine the edge of an RP set $\mathcal{R}^{RP}_k$ through which $r^t$ exits and using this information construct a new RP set $\mathcal{R}^{RP}_{k+1}$. Each test polytope $\mathcal{R}^t_{[k]}$ in the search sequence of test polytopes is a subset of $\mathcal{R}^{RP}_k$, designed to contain a portion of the test ray $r^t$, and the entire sequence of test polytopes is constructed so that the final polytope includes the edge through which $r^t$ exits $\mathcal{R}^N$. To achieve this aim, the test polytopes are an RP set constructed from 2 generators. Further, $\mathcal{R}^t_{[0]}$ is constructed around the origin and each $\mathcal{R}^t_{[k]}$ for $k > 0$ is designed to cover the edge of $\mathcal{R}^{[k-1]}_{[k]}$ through which $r^t$ exited.

We denote the set of all edges of $\mathcal{R}^t_{[k]}$ by $\mathcal{E}^t_{[k]}$ and the $i^{th}$ edge in the set by $\mathcal{E}^t_{i[k]}$ for $i = 1, \ldots, 4$. Using this notation we identify the exit edge by the index $m$, that is, we denote the exit edge of $r^t$ from $\mathcal{R}^t_{[k]}$ as $\mathcal{E}^t_{m[k]}$.

An outline of the membership determination algorithm is given below, and a detailed description of each of its steps is given in the Appendix.

Algorithm 3 (RPMPC Membership Determination):
Given a test state $x^t$ and a test horizon $N^t$ (Algorithm 2), determine the edge of $\mathcal{R}^N$ through which a ray $r^t$ from the origin through $x^t$ exits $\mathcal{R}^N$ and if $x^t \in \mathcal{R}^N$:

1) Set $k = 0$ and construct the initial test polytope $\mathcal{R}^t_{[0]}$.
2) Determine the edge $\mathcal{E}^t_{m[k]}$ of $\mathcal{R}^t_{[k]}$ through which the test ray $r^t$ exits.
3) If the exit edge $\mathcal{E}^t_{m[k]}$ is also an edge of $\mathcal{R}^N$, then test for $x^t \in \mathcal{R}^N$ and terminate the algorithm.
4) If the exit edge $\mathcal{E}^t_{m[k]}$ is not an edge of $\mathcal{R}^N$, then increment the iteration $k = k + 1$ and calculate a new test polytope $\mathcal{R}^t_{[k]}$.
5) Return to step 2.

In summary, RPMPC consists of two operating modes named recovery and convergence. Recovery mode is used when the current state is outside of a small user selected convergence region. In this mode, the structure of $N$-step controllable sets is utilised to implement an efficient ray shooting [10] procedure (Algorithms 2 and 3) that quickly determines the smallest horizon $N_{min}$ whose corresponding controllable set contains the current state. Using this information, it is always possible to determine an admissible control sequence $\mathcal{I}^{RP}_k$ having a reduced number of parameters. The RP is then applied to a standard MPC implementation (15)–(22) via the addition of equality constraints. Once the current state is within the user selected convergence region, standard (fully parameterised) MPC is used.

IV. EXAMPLES

This section illustrates typical control sequences and their corresponding state trajectories to highlight the similarities and differences between RPMPC and MPC. All figures show only the trajectories of the unstable states as the stable states are of little interest for the comparison. For all examples we use an input weighting $R = 1$, and state and terminal state weighting matrices

$$ Q = 10I_n; \quad P = \begin{bmatrix} 0 & 0 \\ 0 & P_s \end{bmatrix}, $$

where $I_n$ is the $n \times n$ identity matrix and $P_s$ is the solution to the discrete-time Lyapunov equation $P_s = A_s^T P_s A_s + Q_s$.

We consider the following $2^{nd}$ order unstable subsystem

$$ x_{k+1} = \begin{bmatrix} 1.105 & 0 \\ 0 & 1.221 \end{bmatrix} x_k + \begin{bmatrix} 1.052 \\ 1.107 \end{bmatrix} u_k. $$

For this example system, two sample initial states are used to compare the behaviour of RPMPC and MPC. In all cases the simulation is terminated once all states are within a small tolerance of the origin. The plots in Fig. 1, show the control inputs applied and the evolution of the unstable state trajectories for two different initial states. We note in the plots for sample 1 (first row of figures) that there is only small differences between RPMPC and MPC in the control inputs and trajectories. The plots for sample 2 (second row...
of figures) have different state trajectories and show clearly that the number of control inputs for RPMPC is 32 and for MPC is 72. This implies that for the chosen initial state, the number of iterations for the unstable states to be within a selected tolerance of the origin is smaller for RPMPC.

\textbf{V. Conclusion}

This paper has presented a novel MPC algorithm for a class of input constrained \( n^{th} \) order systems with two real unstable poles. The new algorithm, called Reduced Parameterisation MPC or RPMPC, uses the structure of controllable sets to determine RP control sequences at each iteration. The algorithm achieves large regions of attraction with a lower computational complexity than that of standard MPC.

Extensive simulation tests have shown that, in general, the control sequences resulting from RPMPC produce trajectories that converge considerably faster than MPC.

In a second paper on RPMPC \cite{6} a detailed analysis of its main properties is presented from which conclusions regarding complexity and stability can be made.

\textbf{Appendix}

Below we present a detailed explanation of the steps listed in Algorithm \ref{alg:3}.

Alg. 3 step 1) To obtain an initial polytope with a minimal number of test edges we use Definition \ref{def:t1} to construct an RP generator set \( \mathcal{G}^t_{[0]} \equiv \mathcal{G}^{RP}(\mathcal{T}_0) \) corresponding to the index set \( \mathcal{T}_0^{RP} \) as per Definition \ref{def:t4},

\begin{equation}
\mathcal{T}_0^{RP} \equiv \{0, \{1, \ldots, N - 1\}\}.
\end{equation}

Using \( \mathcal{G}^t_{[0]} = \mathcal{G}^{RP}(\mathcal{T}_0^{RP}) \) the initial test polytope \( \mathcal{R}^t_{[0]} \) is given in Minkowski sum representation as \( \mathcal{R}^t_{[0]} \equiv S^{\oplus} \left( \mathcal{G}^t_{[0]} \right) \).

We note that by this definition \( \mathcal{R}^t_{[0]} \) is an RP set, thus \( \mathcal{R}^t_{[0]} = \mathcal{R}^{RP}_t \) and is the Minkowski sum of a core generator (as per Definition \ref{def:t1}) and a composite generator (as per Definition \ref{def:t7}).

Next, we have that each test polytope is constructed to include the previous exit edge. This is achieved by adding a polytope offset vector \( z^t_{[k]} \), which is the offset of the previous exit edge, to the polytope constructed from the test generator set \( \mathcal{G}^t_{[k]} \). This is described in the following definition.

\textbf{Definition 10 (Test Polytope):} The test polytope \( \mathcal{R}^t_{[k]} \) is derived from the sets \( \{\mathcal{G}^t_{[k]}, \mathcal{G}^t_{[0]}, \mathcal{U}^t_{[k]}\} \) as follows:

\begin{equation}
\mathcal{R}^t_{[k]} \equiv S^{\oplus} \left( \mathcal{G}^t_{[k]} \right) \oplus z^t_{[k]},
\end{equation}

where \( \mathcal{G}^t_{[k]} = \left\{ \mathcal{G}^t_{[0]} \right\} \) is a set of 2 test generators with indices in the set \( \mathcal{T}_k \). The test polytope offset \( z^t_{[k]} \) is given by \( z^t_{[k]} = \sum_{j \in \mathcal{E}_k} \mathcal{G}^t_{[k]} \), where \( \mathcal{G}^t_{[k]} = \{\mathcal{G}^t_{[j]}\} \) is the set of test “polytope offset” generator directions, \( \mathcal{T}_k = \mathcal{T}_k^{RP} \cup \mathcal{T}_k \), and \( \mathcal{U}^t_{[k]} \equiv \{u^t_{[j]}\} \) is the test “polytope offset” saturation set, where \( u^t_{[j]} \in \{u_{min}, u_{max}\} \).

The following definition presents the corresponding \( H \)-polytope representation of test polytopes.

\textbf{Definition 11 (Test Polytope as \( H \)-polytope):} The \( H \)-polytope representation of the test polytope \( \mathcal{R}^t_{[k]} \) defined in (29) is given by:

\begin{equation}
\mathcal{R}^t_{[k]} \equiv \left\{ x \in \mathbb{R}^2 : \Gamma^t_{[k]} x^t \leq \gamma^t_{[k]} + \Gamma^t_{[k]} z^t_{[k]} \right\},
\end{equation}

where \( \Gamma^t_{[k]} \in \mathbb{R}^{1 \times 2} \) is a set of support line (i.e., the lines that support the edges) normals and \( \gamma^t_{[k]} \in \mathbb{R}^t \) is the set of support line offsets.

Alg. 3 step 2) To determine the exit edge \( \mathcal{E}^t_{m[k]} \) we begin by formally defining the test ray \( r^t \) in parametric form as,

\begin{equation}
r^t = \alpha x^t \text{ where } \alpha \geq 0.
\end{equation}

Given a test polytope \( \mathcal{R}^t_{[k]} \) and its corresponding set of support lines, each given by an equation of the form

\begin{equation}
\Gamma^t_{[i]} x^t = \gamma^t_{[i]} + \Gamma^t_{[i]} z^t_{[i]},
\end{equation}

for \( i = 1, \ldots, 4 \), the test ray \( r^t \) intersects each line\(^6\) and each intersection yields an intersection parameter. Setting \( x = r^t \) in (32), and by using (31) we have that the intersection parameters are given by,

\begin{equation}
\alpha_{i[k]} = \begin{cases} 
\frac{\gamma^t_{[i]} + \Gamma^t_{[i]} z^t_{[i]}}{\Gamma^t_{[i]} x^t} & \text{if } \Gamma^t_{[i]} x^t \neq 0 \\
\infty & \text{if } \Gamma^t_{[i]} x^t = 0
\end{cases},
\end{equation}

for \( i = 1, \ldots, 4 \). Using (33) in conjunction with the properties of the previous exit edge \( \mathcal{E}^t_{m[k-1]} \) we determine the edge \( \mathcal{E}^t_{m[k]} \) through which \( r^t \) exits \( \mathcal{R}^t_{[k]} \) and its intersection parameter \( \alpha_{m[k]} \).

For \( k = 0 \) there is no previous exit edge and since \( \mathcal{R}^t_{[0]} \) is equal to an RP set, by definition 0 it is in the interior of \( \mathcal{R}^t_{[0]} \), implying that \( r^t \) begins in the interior of \( \mathcal{R}^t_{[0]} \). The consequence of this and (31) is that the smallest positive intersection parameter \( \alpha_{m[0]} \) reveals the first intersecting line and hence, the index \( m \) of the edge \( \mathcal{E}^t_{m[0]} \) through which the ray exits \( \mathcal{R}^t_{[0]} \) is, \( m[0] = \arg \min_{i \in \{1, \ldots, 4\}} \{\alpha_{i[0]} : \alpha_{i[0]} > 0\} \).

For \( k > 0 \), the test polytopes \( \mathcal{R}^t_{[k]} \) are offset from 0 and therefore may have both an entry and exit edge. Hence, identifying the exit edge depends on the previous iteration’s exit edge intersection parameter \( \alpha_{m[k-1]} \) i.e., \( \alpha_{m[k]} \geq \alpha_{m[k-1]} \).

We summarise the criteria for determining the exit edge’s index \( m[k] \) as follows,

\begin{equation}
m[k] = \begin{cases} 
\arg \min_{i \in \{1, \ldots, 4\}} \{\alpha_{i[k]} : \alpha_{i[k]} > 0\} & \text{if } k = 0, \\
\arg \min_{i \in \{1, \ldots, 4\}} \{\alpha_{i[k]} : \alpha_{i[k]} \geq \alpha_{m[k-1]} \} & \text{if } k > 0,
\end{cases}
\end{equation}

where \( \alpha_{i[k]} \) was defined in (33). Thus, the exit intersection parameter is \( \alpha_{m[k]} \) and the index \( m[k] \) given by (34) is also the index of the \( \mathcal{R}^t_{[k]} \) exit edge. (In the course of the search it is possible that the ray may exit through the intersection of multiple edges. This case results in more than

\(^6\)Unless the line and the test ray are parallel.
one intersection parameter satisfying the criteria for the exit edge. When this case arises we arbitrarily select a core edge if present or a composite edge otherwise.)

Alg. 3 step 3) Once the exit edge is identified, its intersection parameter is also used to determine the membership of the test state $x^t$ with respect to the test polytope $\mathcal{P}_i^{[k]}. That is, if $\alpha_{m[i]} > 1$ then $x^t \in \mathcal{P}_i^{[k]}$ otherwise it is outside.

When $x^t \notin \mathcal{P}_i^{[k]}$ and the exit edge is a core edge as per Definition 3, then by [1, Lemma 5.3] is it also an edge of $\mathcal{R}^N$ and hence the algorithm can terminate indicating that $x^t \notin \mathcal{R}^N$. However, if the exit edge is a composite edge as per Definition 8 then a new test polytope $\mathcal{R}_i^{[k+1]}$ is required.

Alg. 3 step 4) The construction of each $\mathcal{R}_i^{[k+1]}$ relies on manipulating the generator structure of the test polytopes and their edges. The following definition extends Definition 2 to reflect the structure of test polytope edges.

Definition 12 (Test Polytope Edge Generator Sets): For the test polytope $\mathcal{R}_i^{[k]} \subset \mathbb{R}^2$ constructed as per Definition 10, the set of all edges is denoted by $E_i^{[k]}$. Each edge of $E_i^{[k]}$ denoted by $E_i^{[k]}$ for $i = 1, \ldots, 4$, is derived from the sets $(\mathcal{G}_{m[i]}^{[k]}, \mathcal{G}_r^{[k]}, \mathcal{U}_t^{[k]}, \mathcal{G}_t^{[k]}, \mathcal{U}_t^{[k]})$ as follows:

$$E_i^{[k]} = \mathcal{S} \bigoplus (\mathcal{G}_{m[i]}^{[k]} \oplus \mathcal{G}_r^{[k]} \oplus \mathcal{G}_t^{[k]}),$$

where $\mathcal{G}_r^{[k]} = \{ \mathcal{g}_i^{[k]} \}$ is a set of test “edge basis” generators with indices in the set $\mathcal{T}_i^{[k]}$, defined as the $i^{th}$ element of $\mathcal{T}_i^{[k]}$ is $\mathcal{T}_i^{[k]}$; choose $\mathcal{T}_i^{[k]} \subset \mathcal{T}_r^{[k]}$. The test edge offset $z_{i}^{[k]}$ is given by $z_{i}^{[k]} = \sum_{j \in \mathcal{T}_i^{[k]}} \mathcal{g}_j^{[k]} u_{i,j}^{[k]}$.

where $\mathcal{g}_j^{[k]} = \{ \mathcal{g}_i^{[k]} \} \subset \mathcal{T}_i^{[k]}$ is the set of test “edge offset” generator directions, $\mathcal{T}_i^{[k]} \subset \mathcal{T}_i^{[k]} \subset \mathcal{T}_i^{[k]}$, and $\mathcal{U}_t^{[k]} \triangleq \{ u_{i,j}^{[k]} \}$ is the test “edge offset” saturation set, where $u_{i,j}^{[k]} \in \{ \min, \max \}$ and the entire sequence is determined by the optimising sequence of the edge’s support line as per [1, Theorem 5.2]. Finally, the term $z_{i}^{[k]}$ in (35) is the test “polytope offset”, derived from the sets $(\mathcal{G}_{m[i]}^{[k]}, \mathcal{U}_t^{[k]})$ as per Definition 10.

To construct a new test polytope around $E_i^{[m]}$, we employ a technique called an edge expansion.

Algorithm 4 (Edge Expansion): For a composite edge $E_i^{[m]}$, derived from the sets $(\mathcal{G}_{m[i]}^{[k]}, \mathcal{G}_r^{[k]}, \mathcal{U}_t^{[k]})$, $\mathcal{G}_t^{[k]}, \mathcal{U}_t^{[k]}$) as per Definition 12, of the test polytope $\mathcal{R}_i^{[k]}$, derived from the sets $(\mathcal{G}_{m[i]}^{[k]}, \mathcal{G}_r^{[k]}, \mathcal{U}_t^{[k]})$ as per Definition 10, the edge expansion proceeds as follows,

1. Initialise the following new sets $G_{m+i}^{[k+1]} \triangleq G_{m+i}^{[k]}$,
   $G_{r+i}^{[k+1]} \triangleq G_{r+i}^{[k]} \oplus G_{r+i}^{[k]}$ and $G_{t+i}^{[k+1]} \triangleq G_{t+i}^{[k]} \cup G_{t+i}^{[k]}$.
2. Select the composite generator $g_R^{[k]} \in G_{r+i}^{[k+1]}$.
3. Re-index the elements of $G_{i}^{[k+1]} \oplus G_{r+i}^{[k+1]}$ and $U_t^{[k+1]}$ with respect to the composite generator index $e_i$.
4. Partition $g_R^{[k]}$ and replace it in $G_{i}^{[k+1]}$ with its resulting partitioned generator pair.
5. Derive a new test polytope offset $z_{i}^{[k+1]}$ from the sets $(\mathcal{G}_{m+i}^{[k+1]}, \mathcal{U}_t^{[k+1]})$ as per Definition 10.

6) Construct $\mathcal{R}_i^{[k+1]}$ using $G_{i}^{[k+1]}$ and $z_{i}^{[k+1]}$.

The partitioning of a composite generator described in step 4 is explained by the following definition.

Definition 13 (Composite Generator Partitioning): Given a composite generator $g_R^{[k]}$ defined as

$$g_R^{[k]} = \{ x \in \mathbb{R}^2 : x = g_R^{[k]}(u_{i,j}^{[k]}) \} \subset \mathbb{R}^2$$

where $g_R^{[k]} = \sum_{i=\min}^{\max} \mathcal{g}_i^{[k]}$, and $g_i^{[k]}$ is a core generator direction as in Definition 1 for $i = \min, \max, \max$.

Then partitioning $g_R^{[k]}$ results in the two new generators $g_{+1}^{[k+1]} = \{ x = g_{+1}^{[k+1]}(u_{i,j}^{[k+1]}) \}$ where $u_{i,j}^{[k+1]} \in \mathbb{R}^2$, and $g_{+1}^{[k+1]} = \{ x = g_{+1}^{[k+1]}(u_{i,j}^{[k+1]}) \}$ where $u_{i,j}^{[k+1]} \in \mathbb{R}^2$. (the selection of the notation is the result of re-indexing) and

where $R = \{ e(c[i]) \} = \{ e(c[k]) \} / 2 + s(c[k])$ is the integer mid-point of the interval of generator indices.

After partitioning $g_R^{[k]} \in G_{m+i}$ using Definition 13, the resulting two new generators are used to construct the new test polytope generator set.

$$G_{ni}^{[k]} \triangleq \{ G_{m+i}^{[k]} \cup G_{+1}^{[k]} \} \cup \{ G_{+1}^{[k]} \cup G_{+1}^{[k+1]} \}.$$ (37)

Next using the test polytope offset generator set $G_{e_i}^{[k]}$, its saturated values set $U_t^{[k]}$ and Definition 10, we construct in step 5 the new test polytope offset vector $z^{[k]}$. It is of note that due to the initialisation, $z^{[k]}$ is equivalently constructed using the previous iteration’s offset vector $z^{[k]}$.

$$z_{ni}^{[k]} = z_{ni}^{[k]} / z_{ni}^{[k+1]} = z_{ni}^{[k]} / z_{ni}^{[k+1]}.$$ (38)

REFERENCES


