A set-based approach for stable MPC with Minkowski cost functions

Jordan Leung[1*] and Abraham P. Vinod[1]

***Abstract*— This paper considers the formulation and implementation of model predictive control (MPC) laws with cost functions defined as Minkowski functions of convex, compact sets. We propose constructive procedures for selecting stabilizing terminal elements based on a λ-contractive set. In addition, we describe a set-based implementation of the exact MPC policy based on constrained zonotopes and a suboptimal MPC policy based on a polytopic set approximation. Our numerical simulations demonstrate that the proposed suboptimal set-based policy achieves performance comparable to the standard trajectory-based optimal control formulation, while requiring less computational effort for low-dimensional systems.**

I. Introduction

Model predictive control (MPC) for linear systems typically solves, over a receding horizon, an optimal control problem (OCP) with convex stage and terminal costs and linear state and input constraints [1]. Stability is often enforced by constraining the terminal state to a polyhedral positive invariant (PI) set [2] for the closed-loop dynamics $x^+ = (A + BK)x$ for some K, while choosing a quadratic terminal cost $x^T Px$ that solves a Lyapunov (in)equality for a given quadratic stage cost $x^T Qx + u^T Ru$ [1]. This results in a stabilizing feedback control policy defined by the solution of a quadratic program (QP).

In this paper, we consider an alternative paradigm where the stage and terminal cost functions are Minkowski functions of convex, compact sets [3]. That is, the cost functions are defined so that their sub-level sets are scalar weightings of some convex, compact sets, and we seek to maintain the states and controls inside of the smallest scaling of these sets. Minkowski cost functions provide a flexible description for several non-quadratic objectives, e.g., $1/\infty$-norm costs and penalization of trajectories near constraint boundaries. They also facilitate different OCP solution strategies, e.g., linear programming and set-based solutions.

Procedures for selecting stabilizing terminal elements for MPC based on Minkowski functions are discussed in [4]–[7]. Specifically, [4] selects a set $\mathcal{P}$ defining the terminal cost by solving a set inclusion problem parameterized by a gain matrix K (i.e., an analogue to the Lyapunov inequality used in linear-quadratic MPC). Then, a valid terminal constraint set is defined by scaling $\mathcal{P}$ so that it is constraint admissible under the feedback $u = Kx$. Similarly, [5] chooses a constraint admissible PI set as the terminal set and a valid terminal cost set $\mathcal{P}$ is synthesized by solving a set inclusion problem. Alternatively, [6] showed that any polyhedral λ-contractive set $\mathcal{C}$ can be chosen as a terminal constraint set by choosing $\mathcal{P} = \frac{1}{\beta}\mathcal{C}$ for a sufficiently large β. Meanwhile, the predictive controller in [7] relies on fixed choices of Minkowski stage and terminal cost to ensure stability.

The first contribution of this paper (Section III) is a systematic procedure that extends the stability result in [6], which only showed existence of a sufficiently large scale factor β. To resolve this gap, we provide a systematic procedure for computing $\beta > 0$ such that $\mathcal{P} = \frac{1}{\beta}\mathcal{C}$ induces a stabilizing terminal cost function for a given polyhedral λ-contractive set $\mathcal{C}$. The procedure uses similar machinery as a recent result for linear-quadratic MPC [8], but applied to MPC with Minkowski cost functions.

The second contribution of this paper (Section IV) is a set-based solution method for implementing the MPC feedback policy. For the case where each set is a polytope, we define a linear program (LP) whose solution is a one-step control action that minimizes the cost-to-go of the OCP for the current state. This is similar to existing works that synthesize one-step LPs using multiparametric programming and (approximate) polyhedral dynamic programming [9]–[12]. However, we demonstrate that the one-step LP proposed herein can be synthesized exactly in scalable fashion using constrained zonotopes [13], [14]. In addition, we demonstrate how suboptimal-but-feasible solutions to this LP can be computed by formulating a facet-based LP based on a polytopic approximation of a constrained zonotope.

Section V shows that the suboptimal MPC policy yields computational benefits over a trajectory-based solution in numerical examples. More importantly, however, the presented examples demonstrate the potential of constrained zonotope based solution strategies for the scalable implementation of MPC based on OCPs whose solutions *require* the use of set-based solutions (e.g., min-max MPC [11], [12]).

II. Preliminaries and Problem Setting

A. Notation, terminology, and operations

Let $\mathbb{N}$ represent the naturals including 0. Let $\mathbb{N}_{[a,b]} = \mathbb{N} \cap [a, b]$. For $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$, let $(x, y) = [x^T\ y^T]^T$. Let $\rho(A)$ denote the spectral radius of a matrix A. A set $\mathcal{S} \subset \mathbb{R}^n$ is polyhedral (a polyhedron) if $\mathcal{S} = \{x \in \mathbb{R}^n \mid Sx \leq b\}$ for some $S \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$. If in addition $\mathcal{S}$ is bounded, then it is polytopic (a polytope). A set $\mathcal{S}$ is a C-set if it convex, compact, and includes the origin in its interior. The set $\mathcal{S}$ is a PC-set when it is a polytopic C-set. For a set $\mathcal{S} \subseteq \mathbb{R}^{n+m}$ containing elements (x, u) with $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$, the projection onto $\mathbb{R}^n$ is $\text{Proj}_x(\mathcal{S}) = \{x \mid \exists u \in \mathbb{R}^m \text{ s.t. } (x, u) \in \mathcal{S}\}$. Given a function $f : \mathbb{R}^n \to \mathbb{R}$, the α sub-level set is $\mathcal{B}[f, \alpha] := \{x \mid f(x) \leq \alpha\}$, and its epigraph is $\text{Epi}(f) = \{(x, c) \mid f(x) \leq c\}$. Two optimization

[1] J. Leung and A. P. Vinod are with Mitsubishi Electric Research Laboratories. Emails: leung@merl.com, abrahampvinod@ieee.org.
[*] Corresponding author.

problems are called *equivalent* if the optimal solution(s) of one problem is readily found from the solution(s) of the other, and vice-versa.

B. Minkowski functions

The *Minkowski function* [3] of a C-set $\mathcal{S}$ is

$$\psi_{\mathcal{S}}(x) := \inf\{\lambda \geq 0 \mid x \in \lambda \mathcal{S}\}. \quad (1)$$

In addition, for a PC-set $\mathcal{S} = \{x \in \mathbb{R}^q \mid Sx \leq 1\}$ with $S \in \mathbb{R}^{p \times q}$, then

$$\psi_{\mathcal{S}}(x) = \max_{i \in \mathbb{N}_{[1,p]}} \{s_i^T x\}, \quad (2)$$

where $s_i \in \mathbb{R}^q$ are the transposed rows of S.

We will use the following properties of $\psi_{\mathcal{S}}$ in this work.

Proposition 1. *([3, Proposition 3.12]) For a C-set $\mathcal{S}$: (i) $\psi_{\mathcal{S}}$ is continuous and convex, (ii) $\mathcal{B}[\psi_{\mathcal{S}}, \lambda] = \lambda \mathcal{S}$ for $\lambda \geq 0$, (iii) $0 \leq \psi_{\mathcal{S}}(x) < \infty$ and $\psi_{\mathcal{S}} > 0$ for $x \neq 0$, and (iv) $\psi_{\mathcal{S}}(\lambda x) = \lambda \psi_{\mathcal{S}}(x)$ for $\lambda \geq 0$.*

Proposition 2. *([5, Lemma 1]) Let $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{Z}$ be any three C-sets in $\mathbb{R}^n$, $\mathbb{R}^m$, and $\mathbb{R}^n$, and let $K \in \mathbb{R}^{m \times n}$ be such that $\mathcal{Z} \subseteq \mathcal{X}$ and $K\mathcal{Z} \subseteq \mathcal{Y}$. Then, for all $\alpha > 0$, it holds that: (i) $\psi_{\mathcal{X}}(x) \leq \psi_{\mathcal{Z}}(x)$, (ii) $\psi_{\alpha\mathcal{X}}(x) = \frac{1}{\alpha}\psi_{\mathcal{X}}(x)$, and (iii) $\psi_{\mathcal{Y}}(Kx) \leq \psi_{\mathcal{Z}}(x)$.*

Corollary 1. *Let $\mathcal{Y}$ and $\mathcal{Z}$ be C-sets in $\mathbb{R}^m$ and $\mathbb{R}^n$, let $\mathcal{S} \subseteq \mathbb{R}^n$ be any non-empty set, and let $K \in \mathbb{R}^{m \times n}$ be such that $K(\mathcal{Z} \cap \mathcal{S}) \subseteq \mathcal{Y}$. Then, $\psi_{\mathcal{Y}}(Kx) \leq \psi_{\mathcal{Z}}(x)$ for all $x \in \mathcal{S}$.*

C. Problem Setting

We consider the problem of MPC-based regulation of the origin of a linear time-invariant discrete-time system

$$x_{k+1} = Ax_k + Bu_k, \quad (3)$$

subject to mixed state and input constraints

$$(x_k, u_k) \in \mathcal{Y}, \quad \forall k \geq 0, \quad (4)$$

where $x_k \in \mathbb{R}^n$ is the state, $u_k \in \mathbb{R}^m$ is the control input, and $k \in \mathbb{N}$ is the discrete time index.

Assumption 1. *The pair (A, B) is stabilizable and $\mathcal{Y}$ is a C-set that contains the origin in the interior. In addition, there exist closed sets $\mathcal{X} \subseteq \mathbb{R}^n$ and $\mathcal{U} \subset \mathbb{R}^m$ such that $\mathcal{Y} \subseteq \mathcal{X} \times \mathcal{U}$ and $\mathcal{U}(x) := \{u \in \mathbb{R}^n \mid (x, u) \in \mathcal{Y}\}$ is a C-set for all $x \in \mathcal{X}$.*

We consider a MPC feedback policy for the following optimal control problem (OCP):

$$\begin{aligned}
\mathbb{P}_N(x): \quad & \min_{\{\hat{x}_i, \hat{u}_i\}_{i=0}^{N-1}, \hat{x}_N} V_f(\hat{x}_N) + \sum_{i=0}^{N-1} \ell(\hat{x}_i, \hat{u}_i), && (5a)\\
& \text{s.t.} \quad \hat{x}_0 = x, && (5b)\\
& \hat{x}_{i+1} = A\hat{x}_i + B\hat{u}_i, \quad \forall i \in \mathbb{N}_{[0,N-1]}, && (5c)\\
& (\hat{x}_i, \hat{u}_i) \in \mathcal{Y}, \quad \forall i \in \mathbb{N}_{[0,N-1]}, && (5d)\\
& \hat{x}_N \in \mathcal{X}_f. && (5e)
\end{aligned}$$

where $\ell : \mathbb{R}^{n \times m} \to \mathbb{R}$ is a stage cost, $V_f : \mathbb{R}^n \to \mathbb{R}$ is a terminal cost, and $\mathcal{X}_f \subseteq \mathcal{X}$ is a terminal constraint set. In this paper, we consider the special case where the stage and terminal costs are Minkowski functions, i.e.,

$$\ell(x, u) := \psi_{\mathcal{Q}}(x) + \psi_{\mathcal{R}}(u), \quad V_f(x) := \psi_{\mathcal{P}}(x),$$

where $\mathcal{Q}$, $\mathcal{R}$, $\mathcal{P}$, and $\mathcal{X}_f$ are C-sets of appropriate dimension.

The following standard assumption ensures that the closed-loop system under an MPC feedback policy based on $\mathbb{P}_N(x)$ is exponentially stable (e.g., see [1, Chapter 2]).

Assumption 2. *For all $x \in \mathcal{X}_f$, there exists $u \in \mathbb{R}^m$ such that (i) $(x, u) \in \mathcal{Y}$, (ii) $Ax + Bu \in \mathcal{X}_f$, and (iii) $V_f(Ax + Bu) - V_f(x) \leq -\ell(x, u)$.*

That is, if Assumptions 1-2 hold and $u_0^*(x)$ represents the first control input in an optimal solution to $\mathbb{P}_N(x)$, then (3) in closed-loop with $u_k = u_0^*(x_k)$ is exponentially stable with a region of attraction (RoA) Γ_N, where $\Gamma_N \subseteq \mathcal{X}$ is the set of states for which $\mathbb{P}_N(x)$ is feasible. Furthermore,

$$V_0(Ax + Bu_0^*(x)) - V_0(x) \leq -\ell(x, u_0^*(x)), \quad \forall x \in \Gamma_N, \quad (6)$$

where $V_0(x)$ is the optimal value of $\mathbb{P}_N(x)$.

The first problem we address in this paper concerns how to choose $\mathcal{X}_f$ and $\mathcal{P}$ to satisfy Assumption 2.

Problem 1. *Given (A, B) and a PC-set $\mathcal{Y}$ satisfying Assumption 1, and PC-sets $\mathcal{Q}$ and $\mathcal{R}$, design constructive methods to synthesize PC-sets $\mathcal{P}$ and $\mathcal{X}_f$ satisfying Assumption 2.*

An existing solution [5] to Problem 1 based on PI sets is discussed in Section III-A, while a new approach based on λ-contractive sets is presented in Section III-B.

The second problem addressed in this paper is as follows.

Problem 2. *Given PC-sets $\mathcal{Y}$, $\mathcal{Q}$, $\mathcal{R}$, $\mathcal{P}$, and $\mathcal{X}_f$, define a set-based solution method to solve $\Pi(x)$ for $x \in \Gamma_N$:*

$$\begin{aligned}
\Pi(x): \quad & \min_{\hat{u}} \ \ell(x, \hat{u}) + V_1(Ax + B\hat{u}) && (7a)\\
& \text{s.t.} \ (x, \hat{u}) \in \mathcal{Y}, && (7b)
\end{aligned}$$

where $V_1(x)$ is the optimal value function of $\mathbb{P}_{N-1}(x)$ and we denote the optimal solution to $\Pi(x)$ as $\pi(x)$.

It is well-known that $u_0^*(x) = \pi(x)$. Hence, $\Pi(x)$ can be solved indirectly by solving $\mathbb{P}_N(x)$. However, in Section IV, we propose a direct solution to Problem 2 that aims to transfer as much computation as possible to the offline stage, so that only the solution of a low-dimensional problem similar to (7) is required to compute the online control. The approach is based on a realization that Epi(V_1) can be computed in scalable fashion using constrained zonotopes.

The results in Section III and IV can also be applied independently. That is, the stability results in Section III are independent of how the optimal control input is obtained, while the set-based solution in Section IV can be applied with arbitrary PC-sets $\mathcal{X}_f$ and $\mathcal{P}$.

III. Constructing $\mathcal{X}_f$ and $\mathcal{P}$ for Problem 1

Two solutions to Problem 1 are discussed in this section. The first (Section III-A) is an existing method from [5], where $\mathcal{X}_f$ is selected as a constraint admissible PI set resulting from a static feedback gain, while $\mathcal{P}$ is selected by solving a set inclusion problem. The second (Section III-B) is a novel method where we select $\mathcal{X}_f$ as any λ-contractive PC-set with $\lambda < 1$ and select $\mathcal{P} = \frac{1}{\beta}\mathcal{X}_f$, where $\beta > 0$ is computed through a constructive procedure leveraging the work in [15]. The two approaches are summarized in Algorithms 1 and 2 respectively.

The existing method in Algorithm 1 has the advantage of being computationally tractable for relatively high dimensional systems. In contrast, the novel method in Algorithm 2 results in an MPC policy that induces a much larger RoA than the approach in Algorithm 1, but may be difficult to implement on high dimensional systems. We present both methods as complements to one another.

First, we discuss some preliminaries. We recall the definition of a λ-contractive set [3].

Definition 1. *A set $\mathcal{C} \subseteq \mathcal{X}$ is called λ-contractive for some $\lambda \in [0, 1]$, if, for every $x \in \mathcal{C}$, there exists a u such that*

$$Ax + Bu \in \lambda\mathcal{C}, \quad (x, u) \in \mathcal{Y}, \tag{8}$$

The maximal λ-contractive set is the union of all λ-contractive sets.

Next, we state a well-known result (e.g., see [7, Lemma 2], [3, Exercise 4.7, 13]) that follows from the definition of λ-contractivity for linear systems.

Proposition 3. *Let $\lambda \in [0, 1]$ and let $\mathcal{C} \subseteq \mathcal{X}$ be a λ-contractive C-set. Then, for every $x \in \mathcal{C}$, there exists a control policy $\mu : \mathcal{C} \to \mathbb{R}^m$ such that $(x, \mu(x)) \in \mathcal{Y}$ and*

$$\psi_{\mathcal{C}}(Ax + B\mu(x)) \leq \lambda\psi_{\mathcal{C}}(x). \tag{9}$$

A. Positively Invariant Terminal Set

Algorithm 1 summarizes the procedure for choosing $\mathcal{P}$ and $\mathcal{X}_f$ discussed in [5], while Proposition 4 summarizes the properties discussed therein.

Proposition 4. *([5, Proposition 1]) Let Assumption 1 hold. Then, Algorithm 1 is feasible and its outputs $\mathcal{P}$ and $\mathcal{X}_f$ are PC-sets. Furthermore, for all $x \in \mathcal{X}_f$: (i) $(x, Kx) \in \mathcal{Y}$, (ii) $(A + BK)x \in \mathcal{X}_f$, and (iii) $\psi_{\mathcal{P}}((A + BK)x) - \psi_{\mathcal{P}}(x) \leq -\ell(x, Kx)$. Hence, Assumption 2 is satisfied.*

Algorithm 1 specifies $\mathcal{X}_f$ as a maximal PI terminal set so that Assumption 2.i-ii hold by definition. Then, the choice of $\mathcal{P} = \frac{1}{\alpha}\hat{\mathcal{P}}$ ensures that Assumption 2.iii holds. Algorithm 1 is straightforward and is reminiscent of the procedure used to choose a terminal cost in linear-quadratic MPC. However, the PI set for a particular feedback $u = Kx$ may be significantly smaller than, for example, the maximal $\lambda_{\hat{\mathcal{P}}}$-contractive set. In the following subsection, we demonstrate how to choose $\mathcal{P}$ and $\mathcal{X}_f$ based on a given λ-contractive PC-set.

B. λ-contractive Terminal Set

Consider any λ-contractive PC-set $\mathcal{C}$ with $\lambda < 1$. It is straightforward to see that (9) implies that $\exists\beta > 0$ such that $\mathcal{P} = \frac{1}{\beta}\mathcal{C}$ satisfies Assumption 2.iii with $\mathcal{X}_f = \mathcal{C}$.

Algorithm 1 Positive invariant terminal elements [5]

Input: $A, B, \mathcal{Y}, \mathcal{Q}, \mathcal{R}$

Select a gain K such that $\rho(A + BK) < 1$.

Determine a PC-set $\hat{\mathcal{P}}$ and a $\lambda \in (0, 1)$ satisfying

$$(A + BK)\hat{\mathcal{P}} \subseteq \lambda\hat{\mathcal{P}}, \tag{10}$$

(e.g., using [16, Lemma 5]).

Compute the following constants:

$$\begin{aligned} \lambda_{\hat{\mathcal{P}}} &:= \min_{\eta \geq 0} \left\{ \eta \mid (A + BK)\hat{\mathcal{P}} \subseteq \eta\hat{\mathcal{P}} \right\}, \\ \alpha_{\hat{\mathcal{P}}} &= \alpha_{\mathcal{Q},\hat{\mathcal{P}}} + \alpha_{\mathcal{R},\hat{\mathcal{P}}}, \text{ where} \\ \alpha_{\mathcal{Q},\hat{\mathcal{P}}} &:= \min_{\eta \geq 0} \left\{ \eta \;\middle|\; \hat{\mathcal{P}} \subseteq (1 - \lambda)\eta\mathcal{Q} \right\}, \\ \alpha_{\mathcal{R},\hat{\mathcal{P}}} &:= \min_{\eta \geq 0} \left\{ \eta \;\middle|\; K\hat{\mathcal{P}} \subseteq (1 - \lambda)\eta\mathcal{R} \right\}, \end{aligned} \tag{11}$$

Define $\mathcal{P} = \frac{1}{\alpha}\hat{\mathcal{P}}$ for $\alpha = \alpha_{\hat{\mathcal{P}}}$.

Define $\mathcal{X}_f$ as the maximal PI set (e.g., using [2]) for $x^+ = (A + BK)x$ with constraints $(x, Kx) \in \mathcal{Y}$.

Output: $\mathcal{P}$, $\mathcal{X}_f$

To compute such a β, it is necessary to specify a control policy μ that satisfies Proposition 3 for $\mathcal{C}$. The result in [15] shows that for any such $\mathcal{C}$, there exists a continuous piecewise linear control law $\mu_{\mathcal{C}} : \mathcal{C} \to \mathbb{R}^m$ satisfying Proposition 3 that takes the form

$$\mu_{\mathcal{C}}(x) = \begin{cases} K_1 x & \text{if } x \in \mathcal{C}_1 \\ \vdots & \\ K_s x & \text{if } x \in \mathcal{C}_s \end{cases} \tag{12}$$

where $\mathcal{C}_i$ are simplices formed by n vertices and the origin, $\bigcup_{i=1}^{s} \mathcal{C}_i = \mathcal{C}$, and $\mathcal{C}_i \cap \mathcal{C}_j$ has zero volume for all $i \neq j$. We refer the reader to [3], [8], [15] for a description of how the simplices and gain matrices are computed.

A novel solution to Problem 1 is proposed in Algorithm 2. The following proposition demonstrates that the outputs of Algorithm 2 are well-defined and solve Problem 1.

Algorithm 2 λ-contractive terminal elements

Input: $A, B, \mathcal{Y}, \mathcal{Q}, \mathcal{R}$, λ-contractive PC-set $\mathcal{C}$ with $\lambda < 1$

Compute the matrix-simplex pairs $\{(K_i, \mathcal{C}_i)\}_{i=1}^{s}$ (e.g., using [15]) defining the control law $\mu_{\mathcal{C}}$ in (12).

Compute the following constants:

$$\begin{aligned} \beta_{\mathcal{C}} &= \beta_{\mathcal{Q},\mathcal{C}} + \beta_{\mathcal{R},\mathcal{C}}, \text{ where} \\ \beta_{\mathcal{Q},\mathcal{C}} &:= \min_{\eta \geq 0} \left\{ \eta \;\middle|\; \mathcal{C} \subseteq (1 - \lambda)\eta\mathcal{Q} \right\}, \\ \beta_{\mathcal{R},\mathcal{C}} &:= \min_{\eta \geq 0} \left\{ \eta \;\middle|\; K_i\mathcal{C}_i \subseteq (1 - \lambda)\eta\mathcal{R}, \ \forall i \in \mathbb{N}_{[1,s]} \right\}, \end{aligned} \tag{13}$$

Define $\mathcal{X}_f = \mathcal{C}$ and $\mathcal{P} = \frac{1}{\beta}\mathcal{C}$ with $\beta = \beta_{\mathcal{C}}$.

Output: $\mathcal{P}$, $\mathcal{X}_f$

Proposition 5. *Let Assumption 1 hold. Then, Algorithm 2 is feasible and its outputs $\mathcal{P}$ and $\mathcal{X}_f$ are PC-sets. Furthermore, for all $x \in \mathcal{X}_f$: (i) $(x, \mu_{\mathcal{C}}(x)) \in \mathcal{Y}$, (ii) $Ax + B\mu_{\mathcal{C}}(x) \in \lambda\mathcal{X}_f$,* *and (iii) $\psi_{\mathcal{P}}(Ax+B\mu_{\mathcal{C}}(x))-\psi_{\mathcal{P}}(x) \leq -\ell(x,\mu_{\mathcal{C}}(x))$. Hence, Assumption 2 is satisfied.*

Proof. We begin by proving that each step of Algorithm 2 is feasible. First, existence of a λ-contractive PC-set $\mathcal{C}$ for some $\lambda \in (0,1)$ follows from standard theorems (e.g., [17, Theorem 3.2]). Furthermore, existence of the control policy in (12) follows from [15]. Then, a finite solution to (13) exists since $\mathcal{Q}$, $\mathcal{R}$, and $\mathcal{C}$ are C-sets. Thus, Algorithm 2 is feasible and $\mathcal{X}_f$ and $\mathcal{P}$ are PC-sets by definition.

To prove each of the stated claims, we let $\mathcal{C}$ be any λ-contractive PC-set $\mathcal{C}$ for some $\lambda \in (0,1)$ and let $\mu_{\mathcal{C}}$ be the control policy in (12). Satisfaction of (i) and (ii) follows by Definition 1. Then, all that remains is to prove (iii). Consider the solution of (13) for $\mathcal{C}$. Since $\mathcal{C} \subseteq (1-\lambda)\beta_{\mathcal{Q},\mathcal{C}}\mathcal{Q}$, then

$$(1-\lambda)\beta_{\mathcal{Q},\mathcal{C}}\psi_{\mathcal{C}}(x) \geq \psi_{\mathcal{Q}}(x), \quad \forall x \in \mathbb{R}^n, \tag{14}$$

by Proposition 2.i-ii. Similarly, since $K_i\mathcal{C}_i \subseteq (1-\lambda)\beta_{\mathcal{R},\mathcal{C}}\mathcal{R}$ for all $i \in \mathbb{N}_{[1,s]}$, then $\forall i \in \mathbb{N}_{[1,s]}$:

$$(1-\lambda)\beta_{\mathcal{R},\mathcal{C}}\psi_{\mathcal{C}}(x) \geq \psi_{\mathcal{R}}(K_i x), \quad \forall x \in \mathcal{C}_i, \tag{15}$$

by the combination of Proposition 2.i-ii and Corollary 1, where we have used the fact that $\mathcal{C} \cap \mathcal{C}_i = \mathcal{C}_i$ by definition of $\mathcal{C}_i$ in (12). Then, for any $\beta \geq \beta_{\mathcal{C}}$,

$$\begin{aligned}(1-\lambda)\beta\psi_{\mathcal{C}}(x) &\geq (1-\lambda)[\beta_{\mathcal{Q},\mathcal{C}}+\beta_{\mathcal{R},\mathcal{C}}]\psi_{\mathcal{C}}(x),\\ &\geq \psi_{\mathcal{Q}}(x)+\psi_{\mathcal{R}}(\mu^*(x)), \forall x \in \mathcal{C},\end{aligned} \tag{16}$$

where the first inequality holds trivially and the second holds by (14) and the fact that (15) holds for all $i \in \mathbb{N}_{[1,s]}$. Finally, Proposition 2.ii can be used rewrite (9) as $\psi_{\frac{1}{\beta}\mathcal{C}}(Ax+B\mu_{\mathcal{C}}(x)) - \psi_{\frac{1}{\beta}\mathcal{C}}(x) \leq -(1-\lambda)\beta\psi_{\mathcal{C}}(x)$. Hence, the result in (iii) is obtained from (16) and $\mathcal{P} = \frac{1}{\beta}\mathcal{C}$. □

Both Propositions 4 and 5 hold for any C-sets $\mathcal{Q}$ and $\mathcal{R}$. However, when $\mathcal{Q}$ and $\mathcal{R}$ are PC-sets, then (11) and (13) are solvable as a sequence of LPs.

In general, the maximal λ-contractive set may not be polyhedral even if $\mathcal{Y}$ is a PC-set. However, if a λ-contractive C-set exists for $\lambda \in (0,1)$, then there exists a polyhedral set recursion that: (i) converges to the maximal λ-contractive set if it is a C-set [18], and (ii) can be finitely terminated to produce a λ'-contractive set with $\lambda' \in (\lambda, 1)$ [17]. Any reasonable approximation of a maximal λ-contractive set will be much larger than any constraint admissible PI set under a particular linear feedback law $u = Kx$ (e.g., see [6]–[8]). Hence, the RoA induced by the MPC proposed in this subsection will be much larger than the RoA induced by approaches similar to Section III-A (e.g., [4], [5]).

IV. Set-based Solution to (5)

This section details a solution to Problem 2. First, we describe how $\mathbb{P}_N(x)$ can be reformulated, and then show how constrained zonotopes may be used to address Problem 2.

A. Reformulation of $\mathbb{P}_N(x)$

The problem in (5) can be expressed in an equivalent form by introducing auxiliary variables $\{\gamma_t\}_{t=0}^N$ and $\{\sigma_t\}_{t=0}^{N-1}$:

$$\mathbb{P}_N^\dagger(x): \quad \min_{\{\hat{x}_i,\hat{u}_i,\gamma_i,\sigma_i\}_{i=0}^{N-1},\hat{x}_N,\gamma_N} \gamma_N + \sum_{t=0}^{N-1} \gamma_t + \sigma_t \tag{17a}$$

$$\text{s.t.} \quad (5b)-(5e), \tag{17b}$$

$$\hat{x}_t \in \gamma_t\mathcal{Q}, \ \gamma_t \geq 0, \quad \forall t \in \mathbb{N}_{[0,N-1]}, \tag{17c}$$

$$\hat{u}_t \in \sigma_t\mathcal{R}, \ \sigma \geq 0, \quad \forall t \in \mathbb{N}_{[0,N-1]}, \tag{17d}$$

$$\hat{x}_N \in \gamma_N\mathcal{P}, \ \gamma_N \geq 0, \tag{17e}$$

Proposition 6. *The problem $\mathbb{P}_N^\dagger(x)$ is equivalent to $\mathbb{P}_N(x)$.*

Proof. The objective of (17) ensures that it is equivalent to a problem with (17c)-(17e) replaced with equality constraints: $\psi_{\mathcal{Q}}(\hat{x}_t) = \gamma_t$ and $\psi_{\mathcal{R}}(\hat{u}_t) = \sigma_t$ for all $t \in \mathbb{N}_{[0,N-1]}$ and $\psi_{\mathcal{P}}(\hat{x}_N) = \gamma_N$, which is equivalent to $\mathbb{P}_N(x)$. □

Note that (17) is an LP if $\mathcal{Y}$, $\mathcal{Q}$, $\mathcal{R}$, $\mathcal{P}$, and $\mathcal{X}_f$ are PC-sets.

B. Set-based solution to $\mathbb{P}_N^\dagger(x)$

The goal of the set-based solution is to transfer as much computation as possible to the offline stage, so that the online feedback control only requires the solution to the following single input problem resembling $\Pi(x)$:

$$\Pi^\dagger(x): \quad \min_{\hat{u},\sigma,c} \sigma + c \tag{18a}$$

$$\text{s.t.} \ (x,\hat{u}) \in \mathcal{Y}, \quad \hat{u} \in \sigma\mathcal{R}, \tag{18b}$$

$$(Ax+B\hat{u}, c) \in \mathcal{K}_1, \tag{18c}$$

whose solution we define as $\pi^\dagger(x)$, and

$$\mathcal{K}_1 = \text{Proj}_{(\hat{x}_1,c)}(\mathbb{K}_1), \tag{19}$$

$$\mathbb{K}_1 := \left\{ \begin{array}{c|c} \{(\hat{x}_t,\hat{u}_t)\}_{t=1}^{N-1} & \forall t \in \mathbb{N}_{[1,N-1]} \ \hat{x}_{t+1} = A\hat{x}_t + B\hat{u}_t, (\hat{x}_t,\hat{u}_t) \in \mathcal{Y}, \\ \{(\gamma_t,\sigma_t)\}_{t=1}^{N-1} & \hat{x}_t \in \gamma_t\mathcal{Q}, \hat{u}_t \in \sigma_t\mathcal{R}, (\gamma_t,\sigma_t) \geq 0, \gamma_N \geq 0, \\ \hat{x}_N, \gamma_N, c & \hat{x}_N \in \mathcal{X}_f, \hat{x}_N \in \gamma_N\mathcal{P}, \gamma_N + \sum_{t=1}^{N-1}\gamma_t + \sigma_t = c \end{array} \right\}.$$

Hence, a vector $\mathbf{z} = (\{(\hat{x}_t,\hat{u}_t,\gamma_t,\sigma_t)\}_{t=1}^{N-1}, \hat{x}_N, \gamma_N)$ satisfies $(\mathbf{z},c) \in \mathbb{K}_1$ if and only if $(\mathbf{z},c)$ is a feasible solution to $\mathbb{P}_{N-1}^\dagger(\hat{x}_1)$. Note $\mathbb{K}_1$ and $\mathcal{K}_1$ are polyhedral when $\mathcal{Y}$, $\mathcal{Q}$, $\mathcal{R}$, $\mathcal{P}$, and $\mathcal{X}_f$ are PC-sets. Furthermore, $\mathbb{K}_1$ and $\mathcal{K}_1$ are bounded if $\mathbb{K}_1$ is intersected with any upper bound on c. The following is the primary result of this section.

Proposition 7. *Let $\mathcal{Y}$, $\mathcal{Q}$, $\mathcal{R}$, $\mathcal{P}$, and $\mathcal{X}_f$ be C-sets. Then, $\Pi^\dagger(x)$ is equivalent to $\Pi(x)$ for any $x \in \Gamma_N$. Furthermore, $\mathcal{K}_1 = \text{Epi}(V_1)$.*

Proof. Fix $x \in \Gamma_N$ and let $\gamma = \psi_{\mathcal{Q}}(x)$. It is well-known that $\pi(x) = u_0^*(x)$. Thus, consider an optimal solution to $\mathbb{P}_N^\dagger(x)$, denoted by $\mathbf{z}_0^* = (\{(x_t^*,u_t^*,\gamma_t^*,\sigma_t^*)\}_{t=0}^{N-1}, x_N^*, \gamma_N^*)$, whose optimal cost is $c^*+\sigma_0^*+\gamma$, where $c^* = \gamma_N^* + \sum_{t=1}^{N-1}\gamma_t^*+\sigma_t^*$. By definition, $(x,\hat{u}_0^*) \in \mathcal{Y}$, $\hat{x}_1^* = Ax+B\hat{u}_0^*$, $\psi_{\mathcal{R}}(\hat{u}_0^*) = \sigma_0^*$, and $(\mathbf{z}_1^*, c^*) \in \mathbb{K}_1$, where $\mathbf{z}_1^*$ is defined analogously to $\mathbf{z}_0^*$. Thus, $\pi^* = (u_0^*, \sigma_0^*, c^*)$ must be a feasible solution to $\Pi^\dagger(x)$.

To prove that π^* is an optimizer of $\Pi^\dagger(x)$, assume for the sake of contradiction that there exists a feasible solution $\pi' = (u', \sigma', c')$ to $\Pi^\dagger(x)$ with $\sigma' + c' < \sigma_0^* + c^*$. Then, there exists $\mathbf{z}_1' = (\{(x_t', u_t', \gamma_t', \sigma_t')\}_{t=1}^{N-1}, x_N', \gamma_N')$ with $x_1' = Ax + Bu'$ and $(x, u') \in \mathcal{Y}$ such that $(\mathbf{z}_1', c') \in \mathbb{K}_1$. Thus, the candidate solution $(x, u', \gamma, \sigma', \mathbf{z}_1', c')$ to $\mathbb{P}_N^\dagger(x)$ is feasible and has a cost of $c' + \sigma' + \gamma$. Hence, we have found a feasible solution to $\mathbb{P}_N^\dagger(x)$ with a cost less than $c^* + \sigma_0^* + \gamma$, which is a contradiction. Hence, π^* must be an optimal solution to $\Pi^\dagger(x)$. Thus, $\pi^\dagger(x) = \pi(x) = u_0^*(x)$. Finally, the fact that $\mathcal{K}_1 = \text{Epi}(V_1)$ follows directly from the definition of the epigraph, optimality of V_1, and the definition of $\mathbb{K}_1$ [19]. □

C. Discussion and implementation

Methods to (approximately) solve problems similar to Problem 2 for 1/∞-norm and continuous, convex, piecewise-linear/affine cost functions have long been established in literature – both in deterministic [9], [10] and robust [11], [12], [19] settings. However, these approaches can be computationally prohibitive even for small-scale systems due to their reliance on explicit solutions to multiparametric programming problems. Notably, [12] attempts to alleviate these issues by instead computing a polyhedral *approximation* of a set similar to $\mathcal{K}_1$ through an approximated polyhedral dynamic programming (DP) recursion.

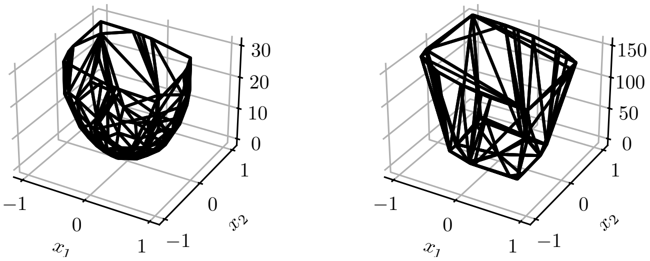
Instead, we propose to compute $\mathcal{K}_1$ directly through (19) using a constrained zonotope representation (CZ-rep) of $\mathbb{K}_1$. This results in an exact CZ-rep of $\mathcal{K}_1$ that is tractable to compute even for high-dimensional systems [14], [19]. The drawback is that auxiliary variables must be introduced to solve $\Pi^\dagger(x)$ if $\mathcal{K}_1$ is in CZ-rep, which will usually result in an LP with a higher number of variables and constraints than the corresponding trajectory optimization problem in (17).

In contrast, if $\mathcal{K}_1$ is represented as a polytope in H-rep, then $\Pi^\dagger(x)$ is an LP with only $m + 2$ optimization variables. However, direction conversion from CZ-rep to H-rep is usually computationally intractable [13]. An alternative is to compute a polytopic inner-approximation $\mathcal{K}_1^{\text{Poly}} \subset \mathcal{K}_1$ (e.g., using ray shooting [20]). Then, replacing $\mathcal{K}_1$ in (18c) with $\mathcal{K}_1^{\text{Poly}}$ results in an $m + 2$ dimensional LP that yields suboptimal solutions to $\Pi^\dagger(x)$, where the suboptimality is dictated by how well $\mathcal{K}_1^{\text{Poly}}$ approximates $\mathcal{K}_1$.

Hence, we propose a *suboptimal* solution to Problem 2 in Algorithm 3. In summary, the practical difference between our approach and existing methods (e.g., [9]–[12]) is that we can scalably compute an exact CZ-rep of $\mathcal{K}_1$, while polytopic approximation is only required to synthesize a more efficient online control policy. In contrast, the method in [12] formulates a set recursion where approximation is performed at every step in the recursion.

Algorithm 3 Set-based MPC policy

1: *Offline*: Form the CZ-rep of $\mathbb{K}_1$ and perform the projection in (19) to obtain the CZ-rep of $\mathcal{K}_1$.
2: Obtain a polytopic approximation $\mathcal{K}_1^{\text{Poly}} \subset \mathcal{K}_1$.
3: *Online*: Obtain a measurement of the current state x_k.
4: Solve for the optimal solution of $\Pi^\dagger(x_k)$ with $\mathcal{K}_1$ in (18c) replaced by $\mathcal{K}_1^{\text{Poly}}$.
5: Apply the control input $u_k = u^*(x_k)$.



(a) PI (b) LC

Fig. 1. Example polytopic approximations of $\mathcal{K}_1$. The plotted sets contain around half as many vertices as $\mathcal{K}_1^{\text{Poly}}$, which `pycvxset` failed to properly render due to the higher number of vertices.

V. Numerical Examples

A. 2D Illustrative Example

We consider the same system as in [6, Example 1]:

$$A = \begin{bmatrix} 1.0250 & 0.0125 \\ 0.0250 & 1.0500 \end{bmatrix}, \quad B = \begin{bmatrix} 0.05 & 0 \\ 0 & 0.05 \end{bmatrix},$$

with $\mathcal{Y} = \mathcal{X} \times \mathcal{U}$, $\mathcal{X} = \mathcal{B}[\|x\|_\infty, 1]$, $\mathcal{U} = \mathcal{B}[\|u\|_\infty, 1]$, $\psi_{\mathcal{Q}}(x) = \|x\|_1$, and $\psi_{\mathcal{R}}(u) = 0.1\|u\|_1$. We design two MPC policies: one based on the positive invariant terminal ingredients computed through Algorithm 1 (i.e., [5]), the other based on the λ-contractive terminal ingredients computed through Algorithm 2 – we refer to these as the PI and LC approaches respectively. Both policies use a horizon of $N = 15$.

In the PI approach, the gain K is the LQR gain resulting from $Q_K = I$ and $R_K = 0.1I$. The maximal PI set is represented using 4 half-spaces. The set $\hat{\mathcal{P}}$ is computed using [16, Lemma 5]. That is, a matrix P is determined by solving the feasibility problem $\bar{A}^T P \bar{A} \leq \bar{\lambda}^2 P$ for $\bar{A} = A + BK$ and $\bar{\lambda} = (1 + \rho(\bar{A}))/2 = 0.926$. Then, $\hat{\mathcal{P}}$ is defined as a polytopic inner-approximation of $\mathcal{B}[\|x\|_P, 1]$ with 10 vertices. This yields $\lambda(\hat{\mathcal{P}}) = 0.856$ and $\alpha = \alpha(\hat{\mathcal{P}}) = 19.7$. Note that $\lambda(\hat{\mathcal{P}}) < 1$ verifies that $\psi_{\hat{\mathcal{P}}}$ satisfies (9) for any $\lambda \in [\lambda(\hat{\mathcal{P}}), 1]$.

In the LC approach, $\mathcal{X}_f$ is a λ-contractive set $\mathcal{C}$ for $\lambda = 0.99$ computed using the standard set recursions (e.g., see [17]). The resulting polytope contains 126 half-spaces. The terminal weight $\beta_{\mathcal{C}} = 220$ is computed by synthesizing the piecewise control law in (12) using the same method as [8].

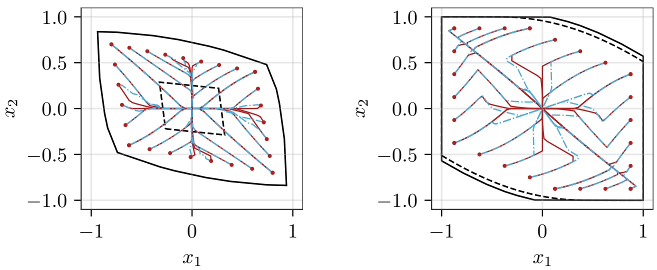
We used `pycvxset` [20] to perform all set-based computations. In the PI case, the CZ-rep of $\mathcal{K}_1$ computed through (19) has latent dimension 357 and 298 equality constraints, whereas in the LC case has latent dimension 469 and 410 equality constraints. For reference, both `pycvxset` and `MPT3` [21] failed to compute $\mathcal{K}_1$ when using polyhedral set operations – even when this process was formulated using lower-dimensional set recursions, similar to [11], [12], [19], due to the underlying vertex-facet enumerations.

The polyhedral approximation $\mathcal{K}_1^{\text{Poly}}$ is generated by combining ray-shooting approximations of $\mathcal{K}_1^{c_i} = \mathcal{K}_1 \cap \{(x, c) | \; c \leq c_i\}$ for $i = 1, 2$. The value of c_2 is chosen so that the entire useful volume of $\mathcal{K}_1$ is covered, while c_1 is chosen to refine the approximation in a smaller area

TABLE I

DIMENSIONS AND EXECUTION TIME PERCENTILES OF THE LPS THAT DEFINE EACH CONTROL POLICY IN SECTION V-A.

	(a) PI			(b) LC		
	Traj	Set-CZ	Set-H	Traj	Set-CZ	Set-H
# of LP var	93	361	4	93	473	4
# of LP eq.	32	301	0	32	413	0
# of LP ineq.	285	722	277	398	946	201
Median (ms)	0.82	1.76	0.66	0.90	2.12	0.52
95^{th} perc. (ms)	0.90	1.99	0.73	1.04	2.61	0.59



(a) PI (b) LC

Fig. 2. Closed-loop trajectories resulting from the Traj/Set-CZ policies (red) and the Set-H policy (blue) for the 2D example. The RoA Γ (solid) and the terminal set $\mathcal{X}_f$ (dashed) are plotted in black.

of interest. In the PI case, 150 shooting vectors are used to approximate $\mathcal{K}_1^{30}$ and 206 vectors are used for $\mathcal{K}_1^{15}$. In the LC case, 150 vectors are used for $\mathcal{K}_1^{250}$ and 206 vectors are used for $\mathcal{K}_1^{150}$. The overall approximation $\mathcal{K}_1^{\text{Poly}}$ is obtained by combining each $\mathcal{K}_1^{c_i}$, reducing the V-rep of the combination, and then generating the H-rep. The resulting set $\mathcal{K}_1^{\text{Poly}}$ is represented with 269 half-spaces (147 vertices) and 193 half-spaces (153 vertices) in the PI and LC cases respectively. Offline construction of these sets took $\sim$20 s in both cases. Example approximations of $\mathcal{K}_1$ are shown in Figure 1.

The top half of Table I shows the dimensions of the LPs defining: (i) Traj: the trajectory-based OCP $\mathbb{P}_N^\dagger(x)$, (ii) Set-CZ: the exact set-based problem $\Pi^\dagger(x)$ with $\mathcal{K}_1$ in CZ-rep, and (iii) Set-H: the approximation of $\Pi^\dagger(x)$ based on $\mathcal{K}_1^{\text{Poly}}$. As discussed in Section IV-C, the Set-H problem is much simpler than the Set-CZ problem. The complexity of $\mathcal{K}_1^{\text{Poly}}$ is chosen in this example so that the Set-H problem has fewer inequality constraints than the Traj problem. Hence, it is reasonable to assume that the Set-H problem will facilitate quicker solutions than the Traj problem since it has both fewer variables and constraints. As expected, the Set-CZ policy for the LC case is more complex than the PI case, while the Set-H policies are similar in complexity since the same number of vertices were used in approximation.

The bottom half of Table I reports the execution time[1] percentiles for the simulations shown in Figure 2. The execution times align with the expected results given the problem dimensions. Recall that the trajectories produced by the Traj and the Set-CZ policies are identical (Proposition 7). In contrast, the Set-H policy produces slightly different trajectories due its reliance on $\mathcal{K}_1^{\text{Poly}}$. Nevertheless, the state trajectories are similar and every initial condition is driven to the origin. We reiterate that the Set-H policy depends closely on the method used to form $\mathcal{K}_1^{\text{Poly}}$. Hence, we caution against drawing overly general conclusions from these results.

Finally, we highlight that the LC terminal region $\mathcal{X}_f$ itself is larger than the feasible region of the PI policy. Hence, the difference in RoA size in Figure 2 would be present even if very small horizon lengths were used in the LC case. This reflects the discussion in Section III-B and the motivation behind choosing a near-maximal λ-contractive terminal set.

B. Planar Spacecraft Relative Motion

In the next example, we consider the equations for spacecraft relative motion in a 2D plane. The continuous-time state-space matrices for the coordinates $(x, y, \dot{x}, \dot{y})$ are

$$A_c = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 3n^2 & 0 & 0 & 2n \\ 0 & 0 & -2n & 0 \end{bmatrix}, \quad B_c = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix},$$

where $n = \sqrt{\mu/a^3}$ is the orbital rate, $\mu = 3.986 \times 10^{14}\ \text{m}^3/\text{s}^2$ is the gravitational constant, and $a = 6.878$ km is the orbital radius. The discrete-time matrices (A, B) are obtained using a zero-order hold discretization with $\Delta t = 5$s. The system constraints are $\mathcal{Y} = \mathcal{X} \times \mathcal{U}$ where $\mathcal{X} = \mathcal{X}_{xy} \times [-3, 3] \times [-3, 3]$, $\mathcal{U} = [-0.01, 0.01] \times [-0.01, 0.01]$, and $\mathcal{X}_{xy}$ is a cone with a vertex at $(x, y) = (2, 0)$ and an angle of $\theta = 30°$ intersected with $x \geq -8$ (see Figure 3). The vertex at $(2, 0)$ represents the position of the chief and the boundaries of $\mathcal{X}_{xy}$ impose a line-of-sight constraint and a maximum following distance on the follower spacecraft.

Only a PI policy is considered in this example since computation of a generic λ-contractive set proved to be difficult. The sets $\mathcal{Q}$ and $\mathcal{R}$ are defined so that $\psi_{\mathcal{Q}} = 10|x| + 0.1|y| + |\dot{x}| + 0.1|\dot{y}|$ and $\psi_{\mathcal{R}} = 10^3\|u\|_1$. The gain K is the LQR gain resulting from $Q_K = \text{diag}(10, 0.1, 1, 0.1)$ and $R_K = 10^3 \cdot I$. The set $\hat{\mathcal{P}}$ is determined as in the previous example by using 104 vertices. This yields $\lambda(\hat{\mathcal{P}}) = 0.881$ and $\alpha = \alpha(\hat{\mathcal{P}}) = 746.9$. The MPC horizon is $N = 8$.

The purpose of this example is to demonstrate how the approach scales on a slightly higher dimensional system. The projection operation in (19) is still computed in less than a minute using CZ representations in `pycvxset`. The resulting CZ-rep has latent dimension 885 and 852 equality constraints. However, generating a 5-dimensional polytopic approximation of $\mathcal{K}_1$ is difficult. Similarly to last example, 170 shooting vectors are used to approximate $\mathcal{K}_1^{300}$ and 234 vectors are used for $\mathcal{K}_1^{150}$. This results in a polyhedral approximation $\mathcal{K}_1^{\text{Poly}}$ represented with 3870 half-spaces (158 vertices). Clearly, few vertex points results in a large H-rep due to the complex nature of 5-dimensional geometry. The overall dimensions of the LPs are show in Table II.

The resulting closed-loop trajectories are shown in Figure 3. Unlike the previous example, there are significant

[1]The LPs were solved in Python 3.12.4 on a 2023 MacBook Pro (M3 Max) with 64 GB RAM using the ECOS solver [22] in `cvxpy` with default settings and no warm-starting. The reported percentiles are based on the cumulative data obtained from every simulation.

TABLE II

DIMENSIONS OF THE LPS IN SECTION V-B.

	Traj	Set-CZ	Set-H
# of var	69	889	4
# of eq.	36	857	0
# of ineq.	857	1778	3878

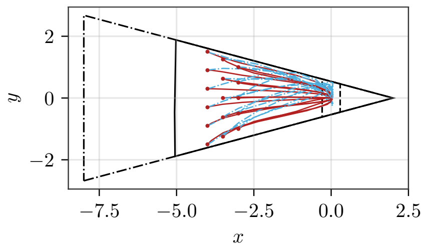


Fig. 3. Closed-loop (x, y) trajectories for the spacecraft relative motion example. Slices at $\dot{x} = \dot{y} = 0$ of the RoA Γ (solid), the terminal set $\mathcal{X}_f$ (dashed), and the constraint set $\mathcal{X}_{xy}$ (dash-dotted) are plotted in black.

differences between the Traj/Set-CZ trajectories and the Set-H trajectories due to poor approximation of $\mathcal{K}_1$. Nevertheless, the Set-H policy remains robust enough to retain convergence even when the set $\mathcal{K}_1$ is poorly approximated due to the curse of dimensionality. However, there is not a clear computational advantage to the Set-H policy in this example due to a higher number of constraints.

We conclude by reiterating that the exact Set-CZ policy is computationally tractable in this example and the corresponding online LP does not suffer greatly from the curse of dimensionality. That is, the latent dimension and number of equality constraints in the CZ-rep of $\mathcal{K}_1$ in this example is $\sim 2.5\times$ the size of the previous example, which is a reasonable increase to incur when doubling the dimension of the state-space. Of course, there is no clear reason to implement the Set-CZ policy over the Traj policy in this example. However, we note that the use of CZ-based set operations provides promise as a computationally scalable way to implement MPC based on OCPs whose solutions *require* the use of set-based solutions (e.g., min-max MPC [11], [12]). In addition, developing better methods for inner-approximating CZs with polytopes would directly improve the practical applicability of the Set-H approach. Future work will explore these directions.

VI. CONCLUSIONS

This paper addressed the problem of stabilization and implementation of MPC with Minkowski cost functions. A new constructive method for selecting a terminal Minkowski cost function based on a given λ-contractive set was derived. Thereafter, a set-based implementation strategy for the feedback policy was developed and methods for its exact (via constrained zonotopes) and suboptimal (via polytopic approximation) implementation were discussed. A simple numerical example showed good performance of a suboptimal set-based policy whose online solution is obtained from a linear program with significantly fewer optimization variables than the corresponding trajectory-based optimal control problem. Future extensions will focus on the use constrained zonotopes to implement MPC based on optimal control problems whose solutions require the use of set-based techniques (e.g., min-max MPC [11], [12]).

REFERENCES

[1] J. Rawlings, D. Q. Mayne, and M. M. Diehl, *Model Predictive Control: Theory, Computation, and Design*. Madison, WI: Nob Hill Publishing, 2nd ed., 2017.

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

[3] F. Blanchini and S. Miani, *Set-Theoretic Methods in Control*. Systems & Control: Foundations & Applications, Birkhäuser Cham, second ed., 2015.

[4] C. Danielson, "Terminal-cost design for model predictive control with linear stage-costs: A set-theoretic method," *Optimal Control Applications and Methods*, vol. 42, no. 4, pp. 943–964, 2021.

[5] S. V. Raković and M. Lazar, "Minkowski terminal cost functions for MPC," *Automatica*, vol. 48, no. 10, pp. 2721–2725, 2012.

[6] S. Grammatico and G. Pannocchia, "Achieving a large domain of attraction with short-horizon linear MPC via polyhedral Lyapunov functions," in *2013 European Control Conference (ECC)*, pp. 1059–1064, 2013.

[7] M. Schulze Darup and M. Mönnigmann, "A stabilizing control scheme for linear systems on controlled invariant sets," *Systems & Control Letters*, vol. 79, pp. 8–14, 2015.

[8] M. Johansson and H. Taghavian, "Stable MPC with maximal terminal sets and quadratic terminal costs," *arXiv:2406.02760*, 2024.

[9] A. Bemporad, F. Borrelli, and M. Morari, "Model predictive control based on linear programming - the explicit solution," *IEEE Transactions on Automatic Control*, vol. 47, no. 12, pp. 1974–1985, 2002.

[10] B. Lincoln and A. Rantzer, "Suboptimal dynamic programming with error bounds," in *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, vol. 2, pp. 2354–2359, 2002.

[11] M. Diehl and J. Bjornberg, "Robust dynamic programming for min-max model predictive control of constrained uncertain systems," *IEEE Transactions on Automatic Control*, vol. 49, no. 12, pp. 2253–2257, 2004.

[12] J. Björnberg and M. Diehl, "Approximate robust dynamic programming and robustly stable MPC," *Automatica*, vol. 42, no. 5, pp. 777–782, 2006.

[13] J. K. Scott, D. M. Raimondo, G. R. Marseglia, and R. D. Braatz, "Constrained zonotopes: A new tool for set-based estimation and fault detection," *Automatica*, vol. 69, pp. 126–136, 2016.

[14] A. P. Vinod, A. Weiss, and S. Di Cairano, "Projection-free computation of robust controllable sets with constrained zonotopes," *Automatica*, vol. 175, p. 112211, 2025.

[15] P.-O. Gutman and M. Cwikel, "Admissible sets and feedback control for discrete-time linear dynamical systems with bounded controls and states," *IEEE Transactions on Automatic Control*, vol. 31, no. 4, pp. 373–376, 1986.

[16] S. V. Rakovic, "Minkowski algebra and Banach contraction principle in set invariance for linear discrete time systems," in *2007 46th IEEE Conference on Decision and Control*, pp. 2169–2174, 2007.

[17] F. Blanchini, "Ultimate boundedness control for uncertain discrete-time systems via set-induced Lyapunov functions," *IEEE Transactions on Automatic Control*, vol. 39, no. 2, pp. 428–433, 1994.

[18] M. Schulze Darup and M. Cannon, "On the computation of λ-contractive sets for linear constrained systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 3, pp. 1498–1504, 2017.

[19] A. P. Vinod, A. G. Kamath, A. Weiss, and S. Di Cairano, "Set-based lossless convexification for a class of robust nonlinear optimal control problems," in *2025 IEEE 64th Conference on Decision and Control (CDC)*, pp. 1769–1776, 2025.

[20] A. P. Vinod, "pycvxset: A python package for convex set manipulation," 2024.

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

[22] A. Domahidi, E. Chu, and S. Boyd, "ECOS: An SOCP solver for embedded systems," in *European Control Conference (ECC)*, pp. 3071–3076, 2013.