Data-Driven Synthesis of Configuration-Constrained Robust Invariant Sets for Linear Parameter-Varying Systems

Manas Mejari, *Member, IEEE*, Sampath Kumar Mulagaleti, Alberto Bemporad, *Fellow, IEEE*

***Abstract*— We present a data-driven method to synthesize *robust control invariant* (RCI) sets for *linear parameter-varying* (LPV) systems subject to unknown but bounded disturbances. A finite-length data set consisting of state, input, and scheduling signal measurements is used to compute an RCI set and invariance-inducing controller, without identifying an LPV model of the system. We parameterize the RCI set as a *configuration-constrained* polytope whose facets have a fixed orientation and variable offset. This allows us to define the vertices of the polytopic set in terms of its offset. By exploiting this property, an RCI set and associated vertex control inputs are computed by solving a single *linear programming* (LP) problem, formulated based on a data-based invariance condition and system constraints. We illustrate the effectiveness of our approach via two numerical examples. The proposed method can generate RCI sets that are of comparable size to those obtained by a model-based method in which exact knowledge of the system matrices is assumed. We show that RCI sets can be synthesized even with a relatively small number of data samples, if the gathered data satisfy certain excitation conditions.**

***Index Terms*— Constrained control, Data driven control, Linear parameter-varying systems, Robust control.**

I. Introduction

A *robust control invariant* (RCI) set is a subset of the state-space in which a system affected by bounded but unknown disturbances can be enforced to evolve *ad infinitum*, by an appropriately designed invariance-inducing controller [4]. Many works have proposed algorithms for computing such RCI sets along with their associated controllers for *linear parameter-varying* (LPV) systems, see, *e.g.*, [7], [14], [15]. These approaches are *model-based*, in that an LPV model of the system is assumed to be known. However, identifying an LPV model poses several challenges [16]. Modelling errors can result in the violation of the invariance property and constraints during closed-loop operations.

To overcome the drawbacks of model-based methods, data-driven approaches have emerged as favorable alternatives. Data-driven *control-oriented* identification algorithms were proposed in [6], [12] which simultaneously compute an RCI set and a controller, while selecting an 'optimal' model from the admissible set. Alternatively, *direct* data-driven approaches were presented in [2], [3], [9], [21], which synthesize RCI sets and controllers directly from open-loop data, without the need of model identification. The algorithm presented in [3], computes a state-feedback controller from open-loop data to induce robust invariance in a *given* polyhedral set, while methods proposed in [2], [9], [21] simultaneously compute invariance-inducing controllers along with RCI sets having zonotopic [2], polytopic [9] or ellipsoidal [21] representations. These contributions, however, are limited to linear time-invariant (LTI) systems. For LPV systems, direct data-driven algorithms have mainly focused on LPV control design, see, *e.g.*, LPV input-output controllers for constrained systems [16], predictive controllers [18], and gain-scheduled controllers [11], [19]. To our knowledge, only a recent work [10] has addressed computation of RCI set for LPV systems in a data-driven setting. This work differs from [10] in terms of description of the RCI sets and computational complexity. We represent the RCI set with a polytope having fixed orientation and varying offset that we optimize in order to maximize the size of the set. As presented in [14], [20], we enforce *configuration constraints* (CC) on this polytope, which enable us to switch between their vertex and hyperplane representations. We exploit this property to parameterize the controller as a vertex control law which is inherently less conservative than a linear feedback control law [8]. A single *linear program* (LP) is formulated and solved to compute the CC-RCI set with associated vertex control law, while the approach in [10] requires to solve a semi-definite programming problem.

II. Notations and Preliminaries

The set of positive reals is denoted by $\mathbb{R}_+$. A set of natural numbers between two integers m and n, $m \leq n$, is denoted by $\mathbb{I}_m^n \triangleq \{m, \ldots, n\}$. Let $A \in \mathbb{R}^{m \times n}$ be a matrix written according to its n column vectors as $A = [\,a_1 \cdots a_n\,]$; we define the vectorization of A as $\vec{A} \triangleq [\,a_1^\top \cdots a_n^\top\,]^\top \in \mathbb{R}^{mn}$, stacking the columns of A. For a finite set $\Theta = \{\theta^1, \ldots, \theta^r\}$, the convex-hull of Θ is given by, $\mathrm{ConvHull}(\Theta) \triangleq \left\{\theta \in \mathbb{R}^n : \theta = \sum_{j=1}^r \alpha_j \theta^j, \text{s.t } \sum_{j=1}^r \alpha_j = 1, \alpha_j \geq 0\right\}$. For matrices A and B, $A \otimes B$ denotes their Kronecker product. The following results will be used in the paper:

M. Mejari is with IDSIA Dalle Molle Institute for Artificial Intelligence, Via la Santa 1, CH-6962 Lugano-Viganello, Switzerland. (email: manas.mejari@supsi.ch)

S. K. Mulagaleti and A. Bemporad are with the IMT School for Advanced Studies Lucca, Italy. (email: {s.mulagaleti, alberto.bemporad}@imtlucca.it)

Lemma 1 (Vectorization): For matrices $A \in \mathbb{R}^{k\times l}$, $B \in \mathbb{R}^{l\times m}$, $C \in \mathbb{R}^{m\times n}$ and $D \in \mathbb{R}^{k\times n}$, the matrix equation $ABC = D$ is equivalent to [1, Ex. 10.18],

$$(C^\top \otimes A)\vec{B} = \overrightarrow{ABC} = \vec{D}, \tag{1}$$

Lemma 2 (Strong duality [17]): Given $a \in \mathbb{R}^n$, $b \in \mathbb{R}$, $M \in \mathbb{R}^{m\times n}$ and $q \in \mathbb{R}^m$, the inequality $a^\top x \leq b$ is satisfied by all x in a nonempty set $\mathcal{X} := \{x : Mx \leq q\}$ if and only if there exists some $\mathbf{\Lambda} \in \mathbb{R}_+^{1\times m}$ satisfying $\mathbf{\Lambda} q \leq b$ and $\mathbf{\Lambda} M = a^\top$.

III. Problem Setting

A. Data-generating system and constraints

We consider the following discrete-time LPV data-generating system

$$x_{t+1} = \mathcal{A}(p_t)x_t + \mathcal{B}(p_t)u_t + w_t, \tag{2}$$

where $x_t \in \mathbb{R}^n$, $u_t \in \mathbb{R}^m$, $p_t \in \mathbb{R}^s$, and $w_t \in \mathbb{R}^n$ are the state, control input, scheduling parameter, and (additive) disturbance vectors, at time t, respectively. The matrix functions $\mathcal{A}(p_t)$ and $\mathcal{B}(p_t)$ have a linear dependency on p_t as

$$\mathcal{A}(p_t) = \sum_{j=1}^{s} p_{t,j} A_o^j, \quad \mathcal{B}(p_t) = \sum_{j=1}^{s} p_{t,j} B_o^j, \tag{3}$$

where $p_{t,j}$ denotes the j-th element of $p_t \in \mathbb{R}^s$ and A_o^j, B_o^j, $j \in \mathbb{I}_1^s$ are *unknown* system matrices. Using (3), the LPV system (2) can be written as

$$x_{t+1} = \underbrace{\begin{bmatrix} A_o^1 \cdots & A_o^s & B_o^1 \cdots & B_o^s \end{bmatrix}}_{M_o} \begin{bmatrix} p_t \otimes x_t \\ p_t \otimes u_t \end{bmatrix} + w_t. \tag{4}$$

Assume that a state-input-scheduling trajectory of $T+1$ samples $\{x_t, p_t, u_t\}_{t=1}^{T+1}$ generated from system (2) is available. The generated dataset is represented as follows

$$X^+ \triangleq [x_2 \quad x_3 \quad \cdots \quad x_{T+1}] \in \mathbb{R}^{n\times T}, \tag{5a}$$

$$X_u^p \triangleq \begin{bmatrix} p_1 \otimes x_1 & p_2 \otimes x_2 & \dots & p_T \otimes x_T \\ p_1 \otimes u_1 & p_2 \otimes u_2 & \dots & p_T \otimes u_T \end{bmatrix} \in \mathbb{R}^{(n+m)s\times T}. \tag{5b}$$

Note that the state measurements x_t are generated according to (2), which are affected by disturbance samples w_t for $t \in \mathbb{I}_1^{T+1}$ whose values are *not* known. However, we assume that for all $t \in \mathbb{I}_1^T$,

$$w_t \in \mathcal{W} \triangleq \{w : -h_{n_w} \leq H_w w \leq h_{n_w}\}, \tag{6}$$

i.e., the additive disturbance w_t is unknown but bounded a priori in the 0-symmetric polytope $\mathcal{W}$. Furthermore, we assume that for all $t \in \mathbb{I}_1^T$, the system parameter satisfies $p_t \in \mathcal{P} \triangleq \mathrm{ConvHull}(\{p^j\}, j \in \mathbb{I}_1^{v_p})$, where $\{p^j\}, j \in \mathbb{I}_1^{v_p}$ are v_p given vertices defining the parameter set $\mathcal{P}$. The state and input constraints are given as

$$\mathcal{X} \triangleq \{x : H_x x \leq h_{n_x}\}, \ \mathcal{U} \triangleq \{u : H_u u \leq h_{n_u}\}, \tag{7}$$

where $\mathcal{X}$ and $\mathcal{U}$ are given polytopic sets.

B. Set of feasible models

A set of *feasible models* which are compatible with the measured data X^+, X_u^p and the set $\mathcal{W}$ is given as follows

$$\mathcal{M}_T \triangleq \left\{ M : x_{t+1} - M \begin{bmatrix} p_t \otimes x_t \\ p_t \otimes u_t \end{bmatrix} \in \mathcal{W}, k \in \mathbb{I}_1^T \right\}, \tag{8}$$

where $M = \begin{bmatrix} A^1, \dots A^s, & B^1, \dots B^s \end{bmatrix} \in \mathbb{R}^{n\times(n+m)s}$ are feasible model matrices. Since we assume that the data-generating system in (4) is LPV with known disturbance set $\mathcal{W}$, it follows that $M_o \in \mathcal{M}_T$. Using the definitions in (5) and (6), the feasible model set $\mathcal{M}_T$ is represented as,

$$\mathcal{M}_T \triangleq \left\{ M : -h_w \leq H_w X^+ - H_w M X_u^p \leq h_w \right\}, \tag{9}$$

with $h_w \triangleq \begin{bmatrix} h_{n_w} & h_{n_w} \cdots & h_{n_w} \end{bmatrix} \in \mathbb{R}^{n_w\times T}$. We now rewrite the feasible model set $\mathcal{M}_T$ in (9) using the vectorization Lemma 1 for $\vec{M} \in \mathbb{R}^{n(n+m)s}$ as

$$\mathcal{M}_T \triangleq \left\{ \vec{M} : -\vec{h}_w + h_M \leq H_M \vec{M} \leq \vec{h}_w + h_M \right\}, \tag{10}$$

where we define $H_M \in \mathbb{R}^{Tn_w\times n(n+m)s}$, $h_M \in \mathbb{R}^{Tn_w}$ and $\vec{h}_w \in \mathbb{R}^{Tn_w}$ as

$$H_M \triangleq \left(X_u^{p\top} \otimes H_w \right), h_M \triangleq \begin{bmatrix} H_w x_2 \\ \vdots \\ H_w x_{T+1} \end{bmatrix}, \vec{h}_w \triangleq \begin{bmatrix} h_{n_w} \\ \vdots \\ h_{n_w} \end{bmatrix} \tag{11}$$

Proposition 1 (Bounded feasible model set): The set $\mathcal{M}_T$ is a bounded polyhedron if and only if $\mathrm{rank}\,(X_u^p) = (n+m)s$ and H_w has a full column-rank n [3, Fact 1].
The full row-rank of X_u^p can be checked from the data, which also relates to the *persistency of excitation* condition for LPV systems [19, condition 1].

Remark 1: We have assumed that full-state measurements are available. If they are not, a possible approach is to design an observer and quantify an error bound between the true and the estimated states. Taking into account this uncertainty, developing a combined observer design and RCI set synthesis method will be a subject of future research.

C. Invariance condition

A set $\mathcal{S} \subseteq \mathcal{X}$ is referred to as RCI for LPV system (4), if for any given $p \in \mathcal{P}$, there exists a control input $u \in \mathcal{U}$ such that the following condition is satisfied:

$$x \in \mathcal{S} \ \Rightarrow \ x^+ \in \mathcal{S}, \ \forall w \in \mathcal{W}, \ \forall M \in \mathcal{M}_T, \tag{12}$$

where the time-dependence of the signals is omitted for brevity and x^+ denotes the successor state.

Remark 2: An indirect approach would involve first identifying a model $\tilde{M} = [\tilde{A}, \tilde{B}]$ and a set $\tilde{\mathcal{W}}$, and then enforcing the RCI condition $\forall\,(x,p) \in \mathcal{S}\times\mathcal{P}, \ \exists\, u \in \mathcal{U} \ : \ \tilde{M} \begin{bmatrix} p \otimes x \\ p \otimes u \end{bmatrix} \oplus \tilde{\mathcal{W}} \subseteq \mathcal{S}$, where $\tilde{\mathcal{W}} \supseteq \mathcal{W}$ is an *inflated* disturbance set that accounts for finite data [12]. This approach may lead to more conservative RCI sets compared to directly satisfying $\forall\ M \in \mathcal{M}_T$ in (12). Making a thorough comparison of the two approaches in terms of RCI set conservativeness is beyond the scope of this paper.

Let $\{x^i, i \in \mathbb{I}_1^{v_s}\}$ be the v_s vertices of the convex RCI set $\mathcal{S}$. We suppose that a *vertex control input* $\mathbf{u}^i \in \mathcal{U}$ is associated with the i-th vertex x^i of the set $\mathcal{S}$, for $i \in \mathbb{I}_1^{v_s}$, *i.e.*, $\mathbf{u}^i$ is applied to the system when the current state is $x_t = x^i$.

Lemma 3: If the set $\mathcal{S}$ is robustly invariant for system (4), then the following two statements are equivalent:

(i) for all $x \in \mathcal{S}$, for any given $p \in \mathcal{P}$, and $\forall(w, M) \in (\mathcal{W}, \mathcal{M}_T)$,

$$x^+ \triangleq M \begin{bmatrix} p \otimes x \\ p \otimes u \end{bmatrix} + w \in \mathcal{S}; \tag{13}$$

(ii) for each vertex $\{x^i, \mathbf{u}^i, i \in \mathbb{I}_1^{v_s}\}$, for each vertex $\{p^j, j \in \mathbb{I}_1^{v_p}\}$ of the set $\mathcal{P}$, and $\forall(w, M) \in \mathcal{W}, \mathcal{M}$,

$$x^{i,j+} \triangleq M \begin{bmatrix} p^j \otimes x^i \\ p^j \otimes \mathbf{u}^i \end{bmatrix} + w \in \mathcal{S}. \tag{14}$$

Proof: Since for each vertex $x^i, i \in \mathbb{I}_1^{v_s}$ and $p^j, j \in \mathbb{I}_1^{v_p}$, it holds that $x^i \in \mathcal{S}$ and $p^j \in \mathcal{P}$, thus, $(i) \Rightarrow (ii)$. Now, we prove the converse, *i.e.*, $(ii) \Rightarrow (i)$. Any $x \in \mathcal{S}$ can be represented as a convex combination of its vertices: $x = \sum_{i=1}^{v_s} \lambda_i x^i, \quad \sum_{i=1}^{v_s} \lambda_i = 1, \quad \lambda_i \geq 0, \forall\, i \in \mathbb{I}_1^{v_s}$. For this state, we choose the corresponding control input as

$$u = \sum_{i=1}^{v_s} \lambda_i \mathbf{u}^i. \tag{15}$$

Note that, $u \in \mathcal{U}$, as $\mathbf{u}^i \in \mathcal{U}$ and $\mathcal{U}$ is convex. Similarly, any given scheduling parameter $p \in \mathcal{P}$ can be expressed as $p = \sum_{j=1}^{v_p} \alpha_j p^j, \quad \sum_{j=1}^{v_p} \alpha_j = 1, \quad \alpha_j \geq 0, \forall j \in \mathbb{I}_1^{v_p}$. Applying the control input (15) to System (4), for any $w \in \mathcal{W}$, we get,

$$x^+ = M \begin{bmatrix} \left(\sum_{j=1}^{v_p} \alpha_j p^j\right) \otimes \left(\sum_{i=1}^{v_s} \lambda_i x^i\right) \\ \left(\sum_{j=1}^{v_p} \alpha_j p^j\right) \otimes \left(\sum_{i=1}^{v_s} \lambda_i \mathbf{u}^i\right) \end{bmatrix} + w, \tag{16a}$$

$$= \sum_{j=1}^{v_p} \alpha_j \sum_{i=1}^{v_s} \lambda_i \underbrace{\left(M \begin{bmatrix} p^j \otimes x^i \\ p^j \otimes \mathbf{u}^i \end{bmatrix} + w \right)}_{x^{i,j+} \in \mathcal{S}}, \tag{16b}$$

$$= \sum_{j=1}^{v_p} \alpha_j \underbrace{\sum_{i=1}^{v_s} \lambda_i x^{i,j+}}_{x^{j+} \in \mathcal{S}} = \sum_{j=1}^{v_p} \alpha_j x^{j+} \in \mathcal{S}, \tag{16c}$$

where (16b) follows from the distributive property of the Kronecker product. As $\mathcal{S}$ is convex, and from (14) we know that $x^{i,j+} \in \mathcal{S}$, then $x^{j+} \in \mathcal{S}$ in (16c). Similarly, as x^+ in (16c) is obtained as a convex combination of $x^{j+} \in \mathcal{S}$, it follows that $x^+ \in \mathcal{S}$, thus, proving $(ii) \Rightarrow (i)$. ■

We remark that the nonlinearity introduced to ensure robust invariance 'for all' models and 'for all' scheduling parameters is resolved via the vertex enumeration of the scheduling parameter set. Condition (ii) in Lemma 3 allows us to enforce the invariance condition only at a finite set of known vertices, instead of enforcing it for all $p \in \mathcal{P}$.

We now formalize the problem addressed in the paper:

Problem 1: Given data matrices (X^+, X_u^p) defined in (5) and the constraint sets (7), compute an invariant set $\mathcal{S}$ and associated vertex control inputs $\mathbf{u}^i \in \mathcal{U}, \ i \in \mathbb{I}_1^{v_s}$ such that: (i) All elements of the set $\mathcal{S}$ satisfy the state constraints $\mathcal{S} \subseteq \mathcal{X}$; (ii) the invariance condition (14) holds. We also aim at maximizing the size of the RCI set $\mathcal{S}$.

IV. RCI SET PARAMETERIZATION

We parameterize the RCI set $\mathcal{S}$ as the following polytope

$$\mathcal{S} \leftarrow \mathcal{S}(\mathbf{q}) \triangleq \{x : Cx \leq \mathbf{q}\}, \ C \in \mathbb{R}^{n_c \times n}, \tag{17}$$

whose facets have a fixed orientation determined by the user-defined matrix C and offset $\mathbf{q} \in \mathbb{R}^{n_c}$ to be computed. We enforce *configuration constraints* (CC) [20] over $\mathcal{S}(\mathbf{q})$, which enable us to switch between the vertex and hyperplane representation of $\mathcal{S}(\mathbf{q})$ in terms of $\mathbf{q}$. Given a polytope $\mathcal{S}(\mathbf{q})$, having v_s vertices, the configuration constraints over $\mathbf{q}$ are described by the cone

$$\mathbb{S} \triangleq \{\mathbf{q} : \ E\mathbf{q} \leq \mathbf{0}_{n_c v_s}\} \tag{18}$$

with $E \in \mathbb{R}^{n_c v_s \times n_c}$. Let $\{V^i \in \mathbb{R}^{n \times n_c}, i \in \mathbb{I}_1^{v_s}\}$ be the matrices defining the vertex maps of $\mathcal{S}(\mathbf{q})$, *i.e.*, $\mathcal{S}(\mathbf{q}) = \mathrm{ConvHull}\{V^i\mathbf{q}, \ i \in \mathbb{I}_1^{v_s}\}$ for a given $\mathbf{q}$. Then, for a particular construction of $\{V^i, i \in \mathbb{I}_1^{v_s}, E\}$, the configuration constraints (18) dictate that

$$\forall \mathbf{q} \in \mathbb{S} \quad \Rightarrow \quad \mathcal{S}(\mathbf{q}) = \mathrm{ConvHull}\{V^i\mathbf{q}, \ i \in \mathbb{I}_1^{v_s}\}. \tag{19}$$

For a user-specified matrix C parameterizing $\mathcal{S}(\mathbf{q})$ in (17), we assume we are given matrices $\{V^i, i \in \mathbb{I}_1^{v_s}\}$, and E satisfying (19). Such matrices are then used to enforce that the RCI set $\mathcal{S}(\mathbf{q})$ is a CC-polytope. For further details regarding their constructions, we refer the reader to Appendix VIII.

V. COMPUTATION OF RCI SET AND INVARIANCE-INDUCING CONTROLLER

We enforce that the set $\mathcal{S}(\mathbf{q})$ is RCI under vertex control law induced by $\mathbf{u}^i, i \in \mathbb{I}_1^{v_s}$. A particular construction of matrices $\{V^i \in \mathbb{R}^{n \times n_c}, i \in \mathbb{I}_1^{v_s}\}$, and $E \in \mathbb{R}^{n_c v_s \times n_c}$ satisfying (19) is given. We enforce $\mathcal{S}(\mathbf{q})$ is a configuration-constrained polytope through the following constraints

$$E\mathbf{q} \leq \mathbf{0}. \tag{20}$$

A. System constraints

Let us enforce the inclusion $\mathcal{S} \subseteq \mathcal{X}$ and input constraints $\mathbf{u}^i \in \mathcal{U}$. Note that from (19), under the constraint (20), we have the following vertex map of $\mathcal{S}(\mathbf{q})$,

$$\mathcal{S}(\mathbf{q}) = \mathrm{ConvHull}\{V^i\mathbf{q}, \ i \in \mathbb{I}_1^{v_s}\} \tag{21}$$

We now enforce the state and input constraints in (7) in terms of $\mathbf{q}$ and $\mathbf{u}^i$ as follows

$$H_x V^i \mathbf{q} \leq h_{n_x}, \qquad H_u \mathbf{u}^i \leq h_{n_u}, \qquad \forall i \in \mathbb{I}_1^{v_s}. \tag{22}$$

B. Invariance condition

We now enforce the invariance condition $x^{i,j+} \in \mathcal{S}(\mathbf{q})$ in (14) for all $w \in \mathcal{W}$ and for all feasible models in the set $M \in \mathcal{M}_T$. Note that from (21), the vertices of $\mathcal{S}(\mathbf{q})$ are $\{x^i \triangleq V_i\mathbf{q}, \ i \in \mathbb{I}_1^{v_s}\}$, under the constraints in (20). Then, the successor state from $x^{i,j+}$ for parameter p^j, input $\mathbf{u}^i$, and disturbance w is given in terms of $\mathbf{q}$ as follows

$$x^{i,j+} = M \begin{bmatrix} p^j \otimes V^i\mathbf{q} \\ p^j \otimes \mathbf{u}^i \end{bmatrix} + w. \tag{23}$$

Thus, the inclusion in (14) is enforced by the inequality

$$Cx^{i,j+} \le \mathbf{q} - d \quad \forall i \in \mathbb{I}_1^{v_s},\ \forall j \in \mathbb{I}_1^{v_p},\ \forall M \in \mathcal{M}_T, \tag{24}$$

where $d \triangleq \max\{Cw : w \in \mathcal{W}\}$ tightens the set $\mathcal{S}(\mathbf{q})$ by the disturbance set $\mathcal{W}$. Using vectorization in (1), and substituting (23), the inequality (24) can be written as follows[1]

$$C\left(\left(\begin{bmatrix} p^j \otimes V_i\mathbf{q} \\ p^j \otimes \mathbf{u}^i \end{bmatrix}\right)^\top \otimes I_n\right)\vec{M} \le \mathbf{q} - d,$$
$$\forall \vec{M} \in \mathcal{M}_T \triangleq \{\vec{M} : H_M\vec{M} \le h_M\}, \tag{25}$$

where we define $\bar{H}_M = \begin{bmatrix} H_M \\ -H_M \end{bmatrix}$ and $\bar{h}_M = \begin{bmatrix} \vec{h}_w + h_M \\ \vec{h}_w - h_M \end{bmatrix}$ with $H_M, h_M, \vec{h}_w$ defined as in (11). Using strong duality (Lemma 2), the invariance condition (25) holds if and only if there exists some multipliers $\mathbf{\Lambda}^{ij} \in \mathbb{R}_+^{n_c \times 2Tn_w}$ for all $i \in \mathbb{I}_1^{v_s}$, $j \in \mathbb{I}_1^{v_p}$ satisfying

$$\mathbf{\Lambda}^{ij}\bar{h}_M \le \mathbf{q} - d, \tag{26a}$$

$$\mathbf{\Lambda}^{ij}\bar{H}_M = C\left(\left(\begin{bmatrix} p^j \otimes V_i\mathbf{q} \\ p^j \otimes \mathbf{u}^i \end{bmatrix}\right)^\top \otimes I_n\right). \tag{26b}$$

C. Maximizing the size of the RCI set

We characterize the size of the RCI set $\mathcal{S} \subseteq \mathcal{X}$ as

$$\mathrm{d}_{\mathcal{X}}(\mathcal{S}) := \min_{\epsilon}\{\|\epsilon\|_1 \quad \text{s.t.} \quad \mathcal{X} \subseteq \mathcal{S} \oplus \mathcal{D}(\epsilon)\}, \tag{27}$$

where $\mathcal{D}(\epsilon) \triangleq \{x : Dx \le \epsilon\}$ is a polytope having user-specified normal vectors $\{D_i^\top, i \in \mathbb{I}_1^{m_d}\}$. Thus, we want to compute a desirably large RCI set $\mathcal{S}$ by minimizing the 'distance' $\mathrm{d}_{\mathcal{X}}(\mathcal{S})$ in (27). Let $\{\mathrm{y}^l, l \in \mathbb{I}_1^{v_x}\}$ be the known vertices of the state-constraint set $\mathcal{X}$, *i.e.*, $\mathcal{X} = \mathrm{ConvHull}\{\mathrm{y}^l, l \in \mathbb{I}_1^{v_x}\}$. For each vertex y^l of $\mathcal{X}$, let $\mathbf{z}^l \in \mathcal{D}(\epsilon)$ and $\mathbf{s}^l \in \mathcal{S}$ for $l \in \mathbb{I}_1^{v_x}$ be the corresponding points in the sets $\mathcal{D}$ and $\mathcal{S}$. The inclusion $\mathcal{X} \subseteq \mathcal{S} \oplus \mathcal{D}(\epsilon)$ in (27) is equivalent to [13],

$$\forall l \in \mathbb{I}_1^{v_x},\ \exists\{\mathbf{z}^l, \mathbf{s}^l\} : \mathrm{y}^l = \mathbf{z}^l + \mathbf{s}^l,\ D\mathbf{z}^l \le \epsilon,\ C\mathbf{s}^l \le \mathbf{q} \tag{28}$$

We now consider the following LP problem which aims at computing the RCI set parameter $\mathbf{q}$ and invariance inducing vertex control inputs $\{\mathbf{u}^i, i \in \mathbb{I}_1^{v_s}\}$ for the LPV system (2). Our goal is to maximize the size of the RCI set $\mathcal{S}(\mathbf{q})$ (or equivalently, to minimize (27)), while satisfying the system constraints, the invariance condition, and the configuration constraints, for all $i \in \mathbb{I}_1^{v_s}$, $j \in \mathbb{I}_1^{v_p}$ and $l \in \mathbb{I}_1^{v_x}$:

$$\begin{aligned} \min_{\{\mathbf{q}, \mathbf{u}^i, \mathbf{\Lambda}^{ij}, \mathbf{z}^l, \mathbf{s}^l, \epsilon\}} \quad & \|\epsilon\|_1 \\ \text{subject to:} \quad & (20) \quad \text{(configuration constraints)}, \\ & (22) \quad \text{(state-input constraints)}, \\ & (26) \quad \text{(invariance condition)}, \\ & (28) \quad \text{(set-size constraints)}. \end{aligned} \tag{29}$$

The LP in (29) consists of $n_c v_s$ linear inequalities for expressing the configuration constraints, $(n_x + n_u)v_s$ linear inequalities for system constraints, $v_s v_p n_c$ number of linear inequality and $v_s v_p n_c n(n+m)s$ number of linear equality constraints for invariance, and $v_x(m_d + n_c + n)$ linear equality-inequality constraints for maximizing the size of the RCI set. The number of optimization variables is $(n_c + v_s(m + 2v_p n_c T n_w) + 2nv_x + m_d)$. The method can be computationally expensive for high dimensional systems as the computational complexity is impacted by the chosen representational complexity of the RCI set n_c, v_s, and the system dimension n.

[1] We used a halfspace representation of $\mathcal{M}_T$ for its computational advantages over the vertex representation. The main drawback of using a vertex representation is its computational complexity, as $\mathcal{M}_T \subset \mathbb{R}^{n(n+m)s}$, which can induce a very large number of vertices.

D. Invariance-inducing controller

The vertex control inputs $\{\mathbf{u}^i, i \in \mathbb{I}_1^{v_s}\} \in \mathcal{U}$ obtained by solving the LP (29) correspond to the vertices $\{x^i, i \in \mathbb{I}_1^{v_s}\}$ of the RCI set $\mathcal{S}$. Then, for any $x_t \in \mathcal{S}$, an admissible control input u_t can be obtained as follows,

$$u_t = \sum_{i=1}^{v_s} \lambda_t^{i,\star}\mathbf{u}^i, \tag{30}$$

where $\lambda_t^{i,\star}, i \in \mathbb{I}_1^{v_s}$ are computed by solving the following LP:

$$\begin{aligned} \{\lambda_t^{i,\star}\} = \quad & \arg\min_{\{\lambda_t^i\}} \textstyle\sum_{i=1}^{v_s} \lambda_t^i \\ \text{subject to:} \quad & \textstyle\sum_{i=1}^{v_s} \lambda_t^i x^i = x_t, \quad 0 \le \lambda_t^i \le 1. \end{aligned} \tag{31}$$

VI. Numerical examples

We demonstrate the effectiveness of the proposed approach via two numerical examples. All computations are carried out on an i7 1.9-GHz Intel core processor with 32 GB of RAM running MATLAB R2022a.

A. Example 1: LPV Double integrator

We consider the following LPV double integrator data-generating system [7],

$$x_{t+1} = \begin{bmatrix} 1+\delta_t & 1+\delta_t \\ 0 & 1+\delta_t \end{bmatrix} x_t + \begin{bmatrix} 0 \\ 1+\delta_t \end{bmatrix} u_t + w_t, \tag{32}$$

where $|\delta_t| \le 0.25$, with constraints $\mathcal{X} \triangleq \{x : |x| <= [5\ 5]^\top\}$, $\mathcal{U} \triangleq \{u : |u| \le 1\}$, and $\mathcal{W} \triangleq \{w : |w| \le [0.25\ 0]^\top\}$. This system can be brought to the LPV form (2) with

$$A^1 = \begin{bmatrix} 1.25 & 1.25 \\ 0 & 1.25 \end{bmatrix}, A^2 = \begin{bmatrix} 0.75 & 0.75 \\ 0 & 0.75 \end{bmatrix}, \begin{matrix} B^1 = \begin{bmatrix} 0 & 1.25 \end{bmatrix}^\top, \\ B^2 = \begin{bmatrix} 0 & 0.75 \end{bmatrix}^\top, \end{matrix} \tag{33}$$

using $p_{t,1} = 2(0.25+\delta_t)$, $p_{t,2} = 2(0.25-\delta_t)$. This corresponds to the simplex scheduling-parameter set $\mathcal{P} = \{p \in \mathbb{R}^2 : p \in [0,1], p_1 + p_2 = 1\} = \mathrm{ConvHull}([\begin{smallmatrix}1\\0\end{smallmatrix}], [\begin{smallmatrix}0\\1\end{smallmatrix}])$. The system matrices in (33) are *unknown* and only used to gather the data. A single state-input-scheduling trajectory of $T = 100$ samples is gathered by exciting system (32) with inputs uniformly distributed in $[-1, 1]$. The data satisfies the rank conditions given in Proposition 1, *i.e*, $\mathrm{rank}(X_u^p) = (n+m)s = 6$. We choose matrix C defining an RCI set with representational complexity given by $n_c = 50$, *i.e.*, $C \in \mathbb{R}^{50\times 2}$, such that $\mathcal{S}(\mathbf{1}_{50})$ is an entirely simple polytope. Each row of C is chosen as follows [20, Remark 3]: $C^i = \left[\cos\left(\frac{2\pi(i-1)}{n_c}\right),\ \sin\left(\frac{2\pi(i-1)}{n_c}\right)\right], i \in \mathbb{I}_1^{n_c}$. Based on the selected C, we build $\{V^i, i \in \mathbb{I}_1^{50}\}$, and E satisfying the configuration constraints in (19). We set $D = C$ defining the distance in (27).

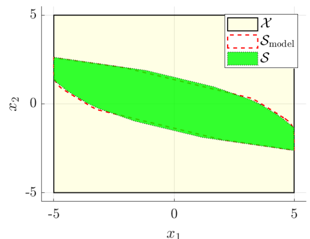

Fig. 1: Example 1: model-based CC-RCI set $\mathcal{S}_{\text{model}}$ (dashed-red), data-based CC-RCI set $\mathcal{S}$ (green).

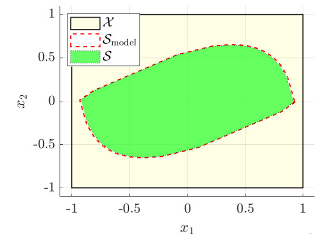

Fig. 3: Example 2: model-based CC-RCI $\mathcal{S}_{\text{model}}$ (dashed-red), proposed data-driven CC-RCI set $\mathcal{S}$ (green).

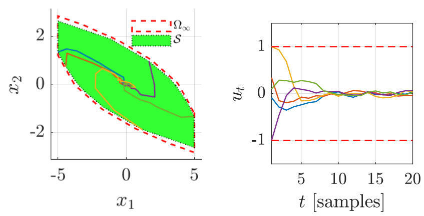

Fig. 2: Example 1: Left panel: CC-RCI set $\mathcal{S}$ (green) with closed-loop state trajectories and MRCI Ω_∞ (dashed-red); Right panel: corresponding control input trajectories and input constraints (dashed red).

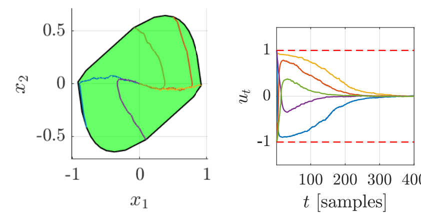

Fig. 4: Example 2: Left panel: CC-RCI set $\mathcal{S}$ with closed-loop state trajectories; Right panel: Corresponding control trajectories and input constraints (dashed red).

The RCI set $\mathcal{S}(\mathbf{q})$ obtained by solving the LP problem (29) is shown in Fig. 1. The total construction and solution time is 40.6 s. We compare the proposed approach to a model-based method, where we compute a CC-RCI set $\mathcal{S}_{\text{model}}$ using the knowledge of the true system matrices. In particular, we fix the model matrix M in (14) to the true system matrices $M = [A^1, A^2, B^1, B^2]$ given in (33), and compute $\mathcal{S}_{\text{model}}$ solving an LP minimizing $\mathrm{d}_{\mathcal{X}}(\mathcal{S}_{\text{model}})$. In the model-based case, invariance constraints (24) are directly computed for a given fixed M. The volume of the RCI set $\mathcal{S}$ obtained with the proposed data-driven proposed algorithm is 25.43, while that provided by the model-based method is 24.56, which shows that the proposed data-based approach generates RCI sets that are of comparable size to those of model-based method. Fig. 2 depicts closed-loop state trajectories starting from some of the vertices of the RCI set, and corresponding control input trajectories. The maximal RCI (MRCI) set Ω_∞ computed using a model-based geometric approach [5, Algorithm 10.5] is also plotted. The state trajectories are obtained by simulating the true system (32) in closed-loop with the invariance inducing controller u_t in (30) computed by solving the LP (31) at each time instance. Note that for each closed-loop simulation, a different realization of the scheduling signal p taking values in the given interval $[0, 1]$ is generated. Moreover, during each closed-loop simulation, different realizations of the disturbance signal $w_t \in \mathcal{W}$ are acting on the system. The result shows that the approach guarantees robust invariance w.r.t. all possible scheduling signals taking values in a given set as well as in the presence of a bounded but unknown disturbance, while respecting the state-constraints. The corresponding input trajectories shown in Fig. 2 (right panel) show that the input constraints are also satisfied. Lastly, we analyse the effect of the number T of data samples on the size of the RCI set. The volume of the RCI set and the LP objective $\mathrm{d}_{\mathcal{X}}(\mathcal{S}(\mathbf{q}))$ for varying T are reported in Table I. As T increases, the

T	30	50	100	Model-based	Ω_∞
volume	22.23	24.47	25.43	24.56	28.19
$\mathrm{d}_{\mathcal{X}}(\mathcal{S}(\mathbf{q}))$	168.31	166.15	164.68	162.11	-

TABLE I: Example 1: Size of the RCI set *vs* samples T.

feasible model set $\mathcal{M}_T$ shrinks progressively, $\mathcal{M}_{T+1} \subseteq \mathcal{M}_T$, thus constraint $\forall M \in \mathcal{M}_T$ is less restrictive, resulting in an increased size of the RCI set.

B. Example 2: Van der Pol oscillator embedded as LPV

Consider the Euler-discretized LPV representation of the Van der Pol oscillator [14] in the form (2) with

$$\left[\, A^1 \mid A^2 \,\right] = \left[\begin{array}{cc|cc} 1 & T_s & 1 & T_s \\ -T_s & 1 & -T_s & 2 \end{array}\right], B^{1,2} = \begin{bmatrix} 0 \\ T_s \end{bmatrix}, \quad (34)$$

where $T_s = 0.1$ is the sampling time. The scheduling parameters are chosen as $p_{t,1} = 1 - \mu T_s(1 - x_{t,1}^2)$ with $\mu = 2$ and $p_{t,2} = 1 - p_{t,1}$. The system constraints are $\mathcal{X} \triangleq \{x : \|x\|_\infty \leq 1\}$, $\mathcal{U} \triangleq \{u : |u| \leq 1\}$ and $\mathcal{W} \triangleq \{w : |w| \leq [10^{-3} \;\; 10^{-3}]^\top\}$. The scheduling parameter set is $\mathcal{P} \triangleq \{p : p_1 \in [1 - \mu T_s, 1], p_2 \in [0, 1], p_1 + p_2 = 1\} = \mathrm{ConvHull}\left(\left[\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}\right], \left[\begin{smallmatrix} 1-\mu T_s \\ \mu T_s \end{smallmatrix}\right]\right)$. The system matrices $\{A^1, A^2, B\}$ are *unknown* and only used to gather the data. A single state-input-scheduling trajectory of $T = 100$ samples is gathered by exciting system (34) with inputs uniformly distributed in $[-1, 1]$. The data satisfy the rank conditions given in Proposition 1, *i.e*, $\mathrm{rank}(X_u^p) = (n + m)s = 6$. The matrix C parameterzing RCI set is selected with $n_c = 30$. Each row of $C \in \mathbb{R}^{30\times 2}$ is set according to [20, Remark 3], such that $\mathcal{S}(\mathbf{1}_{30})$ is an entirely simple polytope. Based on the chosen C, we build the matrices $\{V^i, i \in \mathbb{I}_1^{30}\}$, and E satisfying the configuration constraints in (19). We set $D = C$ defining the distance in (27). The RCI set $\mathcal{S}(\mathbf{q})$ obtained by solving the LP problem (29) is shown in Fig. 3. The total construction and solution time is 21.5 s. For comparision, we also compute the CC-RCI set $\mathcal{S}_{\text{model}}$ with the model-based approach using the knowledge of the true system matrices given in (34). The volume of the RCI set $\mathcal{S}$ with the proposed data-driven algorithm is 1.59, while that of $\mathcal{S}_{\text{model}}$ is 1.62, which shows that the proposed method is able to generate RCI sets that are of similar size to those of the model-based method. Fig. 4 shows closed-loop state trajectories starting from the vertices of the RCI set for different realizations of the scheduling and disturbance signals during closed-loop simulation and corresponding invariance-inducing control inputs, which satisfy the input constraints. The volume of the RCI set and the LP objective $\mathrm{d}_{\mathcal{X}}(\mathcal{S}(\mathbf{q}))$ for varying T are reported in Table II. As T increases, the feasible model set $\mathcal{M}_T$ becomes smaller, resulting in an increased size of the RCI set.

T	20	50	100	$\mathcal{S}_{\text{model}}$
volume	1.50	1.56	1.59	1.62
$\mathrm{d}_{\mathcal{X}}(\mathcal{S}(\mathbf{q}))$	19.04	18.81	18..67	18.54

TABLE II: Example 2: Size of the RCI set *vs* T.

VII. CONCLUSIONS

The paper proposed a data-driven approach to compute a polytopic CC-RCI set and a corresponding vertex control laws for LPV systems. A data-based invariance condition was proposed which utilizes a single state-input-scheduling trajectory without requiring to identify an LPV model of the system. The CC-RCI sets are computed by solving a single LP problem. The effectiveness of the proposed algorithm was shown via two numerical examples to generate RCI sets from a 'small' number of collected data samples.

REFERENCES

[1] K. M. Abadir and J. R. Magnus. *Matrix Algebra*. Econometric Exercises. Cambridge University Press, 2005.

[2] M. Attar and W. Lucia. Data-driven robust backward reachable sets for set-theoretic model predictive control. *IEEE Control Systems Letters*, 7:2305–2310, 2023.

[3] A. Bisoffi, C. De Persis, and P. Tesi. Controller design for robust invariance from noisy data. *IEEE Transactions on Automatic Control*, 68(1):636–643, 2023.

[4] F. Blanchini and S. Miani. *Set-Theoretic Methods in Control*. Birkhäuser, Boston, MA, 2015.

[5] F. Borrelli, A. Bemporad, and M. Morari. *Predictive Control for Linear and Hybrid Systems*. Cambridge University Press, 2017.

[6] Y. Chen and N. Ozay. Data-driven computation of robust control invariant sets with concurrent model selection. *IEEE Transactions on Control Systems Technology*, 30(2):495–506, 2022.

[7] A. Gupta, M. Mejari, P. Falcone, and D. Piga. Computation of parameter dependent robust invariant sets for LPV models with guaranteed performance. *Automatica*, 151:110920, 2023.

[8] 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*, 31(4):373–376, 1986.

[9] M. Mejari and A. Gupta. Direct data-driven computation of polytopic robust control invariant sets and state-feedback controllers. *arXiv:2303.18154*, to appear, Conf. on Decision and Control, 2023.

[10] M. Mejari, A. Gupta, and D. Piga. Data-driven computation of robust invariant sets and gain-scheduled controllers for linear parameter-varying systems. *IEEE Control Systems Letters*, 7:3355–3360, 2023.

[11] J. Miller and M. Sznaier. Data-driven gain scheduling control of linear parameter-varying systems using quadratic matrix inequalities. *IEEE Control Systems Letters*, 7:835–840, 2023.

[12] S. K. Mulagaleti, A. Bemporad, and M. Zanon. Data-driven synthesis of robust invariant sets and controllers. *IEEE Control Systems Letters*, 6:1676–1681, 2022.

[13] S. K. Mulagaleti, A. Bemporad, and M. Zanon. Computation of safe disturbance sets using implicit RPI sets. *IEEE Transactions on Automatic Control*, pages 1–16, 2023.

[14] S. K. Mulagaleti, M. Mejari, and A. Bemporad. Parameter dependent robust control invariant sets for LPV systems with bounded parameter variation rate. *arXiv:2309.02384*, 2023.

[15] H. Nguyen, S. Olaru, P. Gutman, and M. Hovd. Constrained control of uncertain, time-varying linear discrete-time systems subject to bounded disturbances. *IEEE Transactions on Automatic Control*, 60(3):831–836, 2015.

[16] D. Piga, S. Formentin, and A. Bemporad. Direct data-driven control of constrained systems. *IEEE Transactions on Control Systems Technology*, 26(4):1422–1429, 2018.

[17] S. Sadraddini and R. Tedrake. Linear encodings for polytope containment problems. In *58th IEEE Conf. on Decision and Control (CDC)*, pages 4367–4372, 2019.

[18] C. Verhoek, H. S. Abbas, R. Tóth, and S. Haesaert. Data-driven predictive control for linear parameter-varying systems. In *4th IFAC Workshop on Linear Parameter Varying Systems*, pages 101–108, 2021.

[19] C. Verhoek, R. Toth, and H. S. Abbas. Direct data-driven state-feedback control of linear parameter-varying systems. *arXiv:2211.17182*, 2023.

[20] M. E. Villanueva, M. A. Müller, and B. Houska. Configuration-Constrained Tube MPC. *arXiv:2208.12554v1*, 2022.

[21] B. Zhong, M. Zamani, and M. Caccamo. Synthesizing safety controllers for uncertain linear systems: A direct data-driven approach. In *Proc. of the Conference on Control Technology and Applications (CCTA)*, pages 1278–1284, Trieste, Italy, 2022.

VIII. APPENDIX: CONFIGURATION-CONSTRAINED POLYTOPES

We summarize the main results from [20]. Let $\mathcal{S}(\mathbf{q}) \triangleq \{x \in \mathbb{R}^n : Cx \leq \mathbf{q}\}$, $\mathbf{q} \in \mathbb{R}^{n_c}$. We assume that $\mathbf{q}$ is such that $\mathcal{S}(\mathbf{q}) \neq \emptyset$. Let $\mathcal{I} \triangleq \{i_1, \cdots, i_{|\mathcal{I}|}\} \subseteq \mathbb{I}_1^{n_c}$ be the index set based on which we define matrices $C_{\mathcal{I}} \triangleq [C_{i_1}^\top \cdots C_{i_{|\mathcal{I}|}}]^\top \in \mathbb{R}^{|\mathcal{I}|\times n}$ and $\mathbf{q}_{\mathcal{I}} \triangleq [\mathbf{q}_{i_1} \cdots \mathbf{q}_{i_{|\mathcal{I}|}}]^\top \in \mathbb{R}^{|\mathcal{I}|}$ by collecting the rows of matrix C and elements of vector $\mathbf{q}$ corresponding to the indices in set $\mathcal{I}$. The face of $\mathcal{S}(\mathbf{q})$ associated with the set I is defined as $\mathcal{F}_{\mathcal{I}}(\mathbf{q}) \triangleq \{x \in \mathcal{S}(\mathbf{q}) : \; C_{\mathcal{I}}x \geq \mathbf{q}_{\mathcal{I}}\}$.

Definition 1: A polytope $\mathcal{S}(\mathbf{q})$ is entirely simple if for all index sets $\mathcal{I}$ such that the corresponding face is nonempty, *i.e.*, $\mathcal{F}_{\mathcal{I}}(\mathbf{q}) \neq \emptyset$, the condition $\mathrm{rank}(C_{\mathcal{I}}) = |\mathcal{I}|$ holds. □

For some *given* vector $\sigma \in \mathbb{R}^{n_c}$, suppose that $\mathcal{S}(\sigma)$ is an entirely simple polytope. Then, the set of all n-dimensional index sets with corresponding faces being nonempty is defined as $\mathcal{V} \triangleq \{\mathcal{I} : |\mathcal{I}| = n, \; \mathcal{F}_{\mathcal{I}}(\sigma) \neq \emptyset\}$. Let $|\mathcal{V}| = v_s$, *i.e.*, $\mathcal{V} = \{\mathcal{V}_1, \cdots, \mathcal{V}_{v_s}\}$ with $|\mathcal{V}_k| = n$ for each $k \in \mathbb{I}_1^{v_s}$. Then, according to the definition of entirely simple polytopes, $\mathrm{rank}(C_{\mathcal{V}_k}) = n$, such that $C_{\mathcal{V}_k}$ is invertible. Let $\mathbf{I}_{\mathcal{V}_k}^{n_c} \in \mathbb{R}^{n\times n_c}$ be the matrix constructed using rows of identity matrix I_{n_c} corresponding to indices in $\mathcal{V}_k$. Then, defining the matrices $V^k := C_{\mathcal{V}_k}^{-1}\mathbf{I}_{\mathcal{V}_k}^{n_c} \in \mathbb{R}^{n\times n_c}$, we note that $\{V^1\sigma, \cdots, V^{v_s}\sigma\} \in \mathcal{S}(\sigma)$ are the v_s vertices of $\mathcal{S}(\sigma)$. Using matrices $\{V^k, k \in \mathbb{I}_1^{v_s}\}$, define the cone $\mathbb{S} \triangleq \{\mathbf{q} : E\mathbf{q} \leq \mathbf{0}\}$, with $E \triangleq \begin{bmatrix} CV^1 - I_{n_c} \\ \vdots \\ CV^{v_s} - I_{n_c} \end{bmatrix}$ which was described in (18). The following result is the basis for the relationship in (19).

Proposition 2: [20, Theorem 2] Suppose that $\mathcal{S}(\sigma)$ is an entirely simple polytope, based on which the vertex mapping matrices $\{V^k, k \in \mathbb{I}_1^{v_s}\}$ and a matrix E defining the cone $\mathbb{S}$ are constructed as discussed above. Then, $\mathcal{S}(\mathbf{q}) = \mathrm{ConvHull}\{V^k\mathbf{q}, \; k \in \mathbb{I}_1^{v_s}\}$ for all $\mathbf{q} \in \mathbb{S} \triangleq \{\mathbf{q} : E\mathbf{q} \leq \mathbf{0}\}$.