Du *et al*. *Complex Eng Syst* 2024;4:24
DOI: 10.20517/ces.2024.76

Complex Engineering Systems

Research Article

Open Access

Check for updates

Computation of robust positively invariant set based on direct data-driven approach

Qian Du, Hongli Yang

College of Mathematics and Systems Science, Shandong University of Science and Technology, Qingdao 266590, Shandong, China.

Correspondence to: Prof./Dr. Hongli Yang, College of Mathematics and Systems Science, Shandong University of Science and Technology, Qianwangang Road #579, Huangdao, Qingdao 266590, Shandong, China. E-mail: yhlmath@126.com

How to cite this article: Du Q, Yang H. Computation of robust positively invariant set based on direct data-driven approach. *Complex Eng Syst* 2024;4:24. http://dx.doi.org/10.20517/ces.2024.76

Received: 17 Oct 2024 **First Decision:** 8 Nov 2024 **Revised:** 19 Nov 2024 **Accepted:** 26 Nov 2024 **Published:** 30 Nov 2024

Academic Editor: Duxin Chen **Copy Editor:** Fangling Lan **Production Editor:** Fangling Lan

Abstract

A direct data-driven approach for computing the robust positively invariant sets of a discrete-time system is presented in this study. By transforming the invariance conditions into a set of tractable linear matrix inequality problems and combining them with a semidefinite programming problem, we maximize the volume of invariant sets without violating state constraints. Based on two equivalence conditions of invariance, we investigate two algorithms using the one-step method to maximize the volume of the invariant sets. Ultimately, we opted for Algorithm 1, which is more succinct and effective. To further reduce conservatism, we propose an iterative algorithm based on Algorithm 1. The effectiveness of the proposed algorithm is verified through numerical examples.

Keywords: Direct data-driven, robust positively invariant set, linear discrete-time system, semidefinite programming

1. INTRODUCTION

A set is defined as a robust positively invariant (RPI) set, if every initial state contained within it, the trajectory of the system's state remains confined to that set, regardless of any disturbances or uncertainties in its parameters[1]. In the domain of automatic control, the significance of RPI sets is self-evident. The theories and computational approaches of RPI sets have not only attracted significant interest in the academic community but also demonstrated extensive application values in multiple practical fields such as system stability analysis, controller design, and nonlinear system theory[1−6]. Moreover, RPI sets offer us a powerful tool for evaluating and resolving a series of issues closely related to external unknown disturbances.

For linear discrete systems, several approaches have been proposed to compute the RPI sets[7−10]. These are model-based approaches that assume the system is usable. However, achieving precise and dependable models in practical applications is challenging, and imprecise models can result in a loss of system invariance and breach associated constraints[11,12], potentially leading to unstable system operation or suboptimal controller performance. To address these limitations, direct data-driven approaches have emerged as a viable alternative, eliminating the need for model identification[12]. As we all know, the two important forms of invariant sets are polyhedral and ellipsoidal sets[4]. Ellipsoidal sets may be more conservative than polyhedral sets. Since polyhedral sets offer more flexible and complex representations, they have an advantage over ellipsoidal sets both in theory and practice[1]. Additionally, polyhedral representations naturally capture physical constraints on state and control variables[3]. Due to these factors, this paper emphasizes the study of polyhedral sets.

In this paper, we propose a direct data-driven method[11,13] for computing polyhedral RPI sets. We assume that the RPI set is a 0-symmetric convex polyhedron of predetermined complexity. The method presented in[14] involves a relatively high number of algorithm optimization variables and linearized matrix inequality constraints, resulting in slow running speeds. Our method expands upon and optimizes the approach recently developed by Mejari *et al*[14]. We have derived a new invariance condition and developed an iterative algorithm based on this condition. In addressing the nonlinear problem of the invariance condition, we ingeniously utilize relevant variable transformations and appropriate scaling techniques. Subsequently, a series of LMIs[15] are employed to precisely represent the constraints and invariance requirements of the system. Finally, we resort to the semidefinite programming (SDP) problem for solution, aiming to maximize the volume of the RPI sets. Notably, our method does not require an accurate system model or system identification steps; it merely requires a state trajectory composed of a finite number of data samples as input. Furthermore, the number of optimization variables and LMIs in this iterative algorithm is less than that in[14], thereby reducing running time and operating costs to some extent.

The primary contribution of this paper lies in the introduction of a novel data-driven methodology tailored for computing systems equipped with RPI sets. This approach holds significant relevance, particularly within the realms of control system design and analysis. A standout feature of our method is its reliance on a limited dataset, bypassing the need for comprehensive mathematical model knowledge-a pivotal advantage in numerous practical applications where precise mathematical models are elusive or challenging to derive. Our work commences by establishing two fundamental equivalence criteria pertaining to invariance properties. Building upon these, we devise two single-step algorithms grounded in LMI frameworks for the computation of RPI sets, denominated as Algorithm 1 and Algorithm 2. A comparative assessment between these algorithms is subsequently undertaken.

Experimental evaluations disclose that although both algorithms converge upon identical RPI sets, Algorithm 1 distinguishes itself through reduced optimization variable requirements and exhibits superior computational efficiency relative to Algorithm 2. This distinction carries substantial weight in practical implementations, as augmented computational speed directly contributes to enhanced real-time system performance and cost efficiency.

Capitalizing on the merits of Algorithm 1, we further propose an iterative refinement strategy aimed at incrementally approximating the target volume of RPI sets over successive iterations. This iterative algorithm ingeniously retains the computational efficiency hallmark of Algorithm 1 while simultaneously alleviating resultant conservatism, thereby striking a balance between precision and performance.

The structure of this paper is as follows: Preliminaries and problem formulation are given in Section 2.1. Section 2.2 presents data-based invariances and constraints. Section 2.3 gives the relevant algorithms to maximize the RPI sets. Numerical examples are given in Section 3. Section 4 provides the discussion. Section 5 gives a summary of this article.

Notations: The following notations are made in this paper. The set of positive real numbers is denoted by R_+; a diagonal matrix with positive members is denoted by D_+^n; the matrix A's transpose is A^T. R^n denotes n-dimensional Euclidean space, and $R^{n\times m}$ denotes a set of $n\times m$-dimensional matrices. A matrix containing zeros in the relevant dimension is written as $\mathbf{0}$. $\mathbf{1}_m$ represents the vector of ones of dimension m. The identity matrix in m dimensions is written as I_m, and the i-th column of the identity matrix is denoted by the symbol e_i. $*$'s represents the matrix element that is uniquely identifiable by symmetry. For the symmetry matrix X, $X \geq 0$ (> 0) indicates that the matrix X is a semi-positive definite matrix (positive definite matrix). Let $A \in R^{m\times n}$ be a matrix of n vectors $A = [a_1 \ldots a_n]$, we define the vectorization of A as $\vec{A} = [a_1^T \ldots a_n^T]^T \in R^{mn}$. For the finite set $\Theta_v = \left\{\theta^1, \ldots, \theta^t\right\}$ with $\theta^j \in R^n$ for $j = 1, \ldots t$, $conv(\Theta_v) = \left\{\theta \in R^n : \theta = \sum_{j=1}^{t} \alpha_j \theta^j, s.t \sum_{j=1}^{t} \alpha_j = 1, \alpha_j \in [0, 1]\right\}$ denotes the convex-hull of a Θ_v, and $A \otimes B$ denotes the Kronecker product between A and B.

2. METHODS

2.1. Preliminaries and problem formulation

2.1.1 Preliminaries

Lemma 1 ([16]): (Vectorization) For matrices $A \in R^{m\times n}$, $B \in R^{n\times l}$, $C \in R^{l\times k}$ and $D \in R^{m\times k}$, the matrix equation $ABC = D$ is equivalent to

$$\left(C^T \otimes A\right) \vec{B} = \overrightarrow{ABC}, \tag{1a}$$

$$\overrightarrow{ABC} = \left(C^T B^T \otimes I_k\right) \vec{A}. \tag{1b}$$

Lemma 2 ([17]): (Schur complement): Given the matrix $S = \begin{bmatrix} S_1 & S_2 \\ S_3 & S_4 \end{bmatrix}$, where S_1 is a positive definite matrix, and define the Schur complement matrix of S_1 as $M = S_4 - S_3 S_1^{-1} S_2$, then $S \geq 0 (> 0) \Leftrightarrow M \geq 0 (> 0)$.

Definition 1 ([1]): (Polyhedral set) A convex polyhedral set is a set of the form $\mathcal{P}(F, g) = \{x : Fx \leq g\}$. A polyhedral set includes the origin if and only if $g \geq 0$ and includes the origin as an interior point if and only if $g > 0$.

Definition 2 ([1]): (0–Symmetric convex polyhedral set) A 0–symmetric convex polyhedral set can always be represented in the form $\bar{\mathcal{P}}(F, g) = \{x : -g \leq Fx \leq g\}$. If $\bar{\mathcal{P}}(F, g)$ includes 0 as an interior point, up to a normalization, it can be represented as $\bar{\mathcal{P}}(F, \bar{\mathbf{1}}) = \left\{x : -\bar{\mathbf{1}} \leq Fx \leq \bar{\mathbf{1}}\right\}$, where $\bar{\mathbf{1}}^T = [1 \ \ 1 \ldots 1]$.

The properties of 0-symmetric convex polyhedral are as follows:

(i) If $x \in \mathcal{P}$, then $-x \in \mathcal{P}$.

(ii) If $x, y \in \mathcal{P}$, then for any $\lambda \in [0,1]$, $\lambda x + (1 - \lambda) y \in \mathcal{P}$.

2.1.2 Problem formulation

The following linear discrete-time system with no control inputs

$$x(k + 1) = Ax(k) + w(k) \tag{2}$$

is considered in this paper, where $x(k) \in R^n$ and $w(k) \in R^n$ are the system state and the external disturbance at time k, respectively. The matrix A is unknown. A state trajectory of $T+1$ samples $\{x(k)\}_{k=1}^{T+1}$ is generated from the system (2). The generated data is denoted by:

$$X^+ \triangleq [x(2) \ \ x(3) \ldots x(T+1)] \in R^{n\times T}, \tag{3a}$$
$$X \triangleq [x(1) \ \ x(2) \ldots x(T)] \in R^{n\times T}. \tag{3b}$$

The constraints of the system state and disturbances are as follows

$$\mathcal{X} \triangleq \left\{x \in R^n : Fx \leq \mathbf{1}_{n_x}\right\}, \tag{4a}$$
$$\mathcal{W} \triangleq \left\{w \in R^n : \mid Dw \mid \leq \mathbf{1}_{n_w}\right\}, \tag{4b}$$

where $F \in R^{n_x\times n}$ and $D \in R^{n_w\times n}$ are known. Note that the external disturbances of the system are bounded.

Define a set of feasible models as follows:

$$\mathcal{A} \triangleq \left\{A \in R^{n\times n} : x(k+1) - Ax(k) \in \mathcal{W}, k = 1, \ldots, T\right\} \tag{5}$$

By using the matrices defined in (3) and the disturbance set $\mathcal{W}$ in (4), (5) can be expressed as follows:

$$\mathcal{A} \triangleq \left\{A \in R^{n\times n} : -\bar{\mathbf{1}} \leq DX^+ - DAX \leq \bar{\mathbf{1}} \in \mathcal{W}\right\}, \tag{6}$$

where $\bar{\mathbf{1}} \triangleq \left[\mathbf{1}_{n_w} \ \ \mathbf{1}_{n_w} \ \ \ldots \mathbf{1}_{n_w}\right] \in R^{n_w\times T}$. By using Lemma 1, (6) is rewritten as follows,

$$\mathcal{A} \triangleq \left\{A \in R^{n\times n} : -\bar{\mathbf{1}}_{Tn_w} + d \leq Z\vec{A} \leq \bar{\mathbf{1}}_{Tn_w} + d \in \mathcal{W}\right\}, \tag{7}$$

where

$$Z \triangleq \left(X^T \otimes D\right) \in R^{T_{n_w}\times n^2}, \tag{8a}$$
$$d \triangleq \begin{bmatrix} Dx(2) \\ Dx(3) \\ \vdots \\ Dx(T+1) \end{bmatrix} \in R^{T_{n_w}}. \tag{8b}$$

Proposition 1 [12,18]**:** The set $\mathcal{A}$ in (6) is a bounded polyhedron if and only if $rank\ (X) = n$ and D has a full column rank.

Remark 1: If the condition is not met, then set $\mathcal{A}$ is unbounded, making it difficult to find a feasible RPI set.

To enhance clarity, denote $x(k+1)$ as x^+, then system (2) can be represented by:

$$x^+ = Ax + w. \tag{9}$$

Consider the following set of 0–symmetric convex polyhedral set with predefined complexity n_p, which is given as follows:

$$\mathcal{P} \triangleq \left\{x \in R^n : -\bar{\mathbf{1}} \leq PW^{-1}x \leq \bar{\mathbf{1}}\right\}, \tag{10}$$

where $P \in R^{n_p\times n}$, $W \in R^{n\times n}$. Assuming that W is reversible, this will be ensured by the invariance conditions in the form of LMIs.

The set $\mathcal{P}$ is the RPI set of system (9), if the following condition is satisfied:

$$x \in \mathcal{P} \Rightarrow x^{+} \in \mathcal{P}, \forall w \in \mathcal{W}, \forall A \in \mathcal{A}. \tag{11}$$

The set $\mathcal{P}$ also must adhere to the state constraints, meaning that $\mathcal{P} \subseteq \mathcal{X}$, which leads to the following:

$$x \in \mathcal{P} \Rightarrow x \in \mathcal{X}. \tag{12}$$

From the above (3)-(4), (9)-(12), the problem addressed in this article is formulated as follows:

Problem: Given the data in (3), state constraints in (4) and matrix P, compute W in (10) such that:

(1) The invariance condition in (11) is satisfied;

(2) The set $\mathcal{P}$ satisfies state constraints in (4);

(3) Maximize the volume of the set $\mathcal{P}$.

2.2. Invariance conditions and constraints based on data

To render the state constraints of the system and invariance conditions more manageable, consider the following coordinate transformation[14]:

$$\theta = W^{-1}x \Leftrightarrow x = W\theta. \tag{13}$$

With the coordinate transformation, (10) can be expressed as:

$$\mathcal{P} \triangleq \{W\theta \in R^{n} : \theta \in \Theta\}, \tag{14}$$

where Θ is a set of 0-symmetric convex polyhedra defined as follows:

$$\Theta \triangleq \left\{\theta \in R^{n} : -\mathbf{1}_{n_p} \leq P\theta \leq \mathbf{1}_{n_p}\right\}. \tag{15}$$

For a fixed P, the vertices of Θ are known, so Θ can be written as a convex hull of these finite vertices[1].

$$\Theta = conv\left(\left\{\theta^{1}, \ldots, \theta^{2\sigma}\right\}\right), \tag{16}$$

where $\theta^{1}, \ldots, \theta^{2\sigma}$ are the vertices of the Θ, and σ is known and determined by matrix P.

Due to the symmetry of the set Θ, there is $\theta^{j+\sigma} = -\theta^{j}$, for $j = 1, \ldots, \sigma$. And by using (13), the state constraints are rewritten in θ state-space as follows

$$\begin{gathered} HW\theta \leq \mathbf{1}_{n_x}, \forall \theta \in \Theta \Leftrightarrow \\ -\mathbf{1}_{n_x} \leq HW\theta^{j} \leq \mathbf{1}_{n_x}, j = 1, \ldots, \sigma. \end{gathered} \tag{17}$$

With (13), system (9) can be expressed as:

$$W\theta^{+} = AW\theta + w, \tag{18}$$

where $A \in \mathcal{A}$, $w \in \mathcal{W}$.

Two equivalent invariance conditions for system (18) in the θ state space are shown below.

Lemma 3 ([14]): For system (18), where the set Θ in (15) is RPI set, then the following two conditions are equivalent:

(i) for all $\theta \in \Theta$, $\forall w \in \mathcal{W}$ and $\forall A \in \mathcal{A}$,

$$\theta^{+} = \left(W^{-1}AW\theta + W^{-1}w\right) \in \Theta \tag{19}$$

(ii) for each vertex θ^{j}, $j = 1, \ldots, 2\sigma$ of the set Θ, $\forall w \in \mathcal{W}$ and $\forall A \in \mathcal{A}$,

$$\left(\theta^{j}\right)^{+} = \left(W^{-1}AW\theta^{j} + W^{-1}w\right) \in \Theta \tag{20}$$

2.3. Maximize the volume of the RPI set

In this section, we propose data-driven sufficient LMI conditions for computing the matrix W, ensuring that the set Θ remains invariant.

First, we will consider condition (19) for RPI of the set Θ. Applying (1b), system (18) can be further written as follows:

$$W\theta^{+} = \underbrace{\left((W\theta)^{T} \otimes I_{n}\right)}_{g(W,\theta)} \vec{A} + w. \tag{21}$$

To achieve fewer conservative LMI conditions, the variables $V_i \in R^{n\times n}$ are introduced and signals $\xi_i = V_i^{-1}W(\theta)^{+}$, for $i = 1, \ldots, n_p$. The dynamics (21) can then be expressed as follows

$$g(W,\theta)\vec{A} + w - V_i\xi_i = \mathbf{0}. \tag{22}$$

From (15), the invariance condition (19) is given as follows: for all $\theta \in \Theta$,

$$1 - (e_i^{T}P\theta^{+})^{2} \geq 0, \forall w \in \mathcal{W}, \forall A \in \mathcal{A}, \tag{23}$$

where e_i is the identity matrix I_{n_p}'s i-th column vector. By using $\xi_i = V_i^{-1}W(\theta)^{+}$, (23) can be written as follows

$$1 - (e_i^{T}PW^{-1}V_i\xi_i)^{2} \geq 0, \forall w \in \mathcal{W}, \forall A \in \mathcal{A}. \tag{24}$$

We multiply (24) by positive scalar variable $\phi_i > 0$ and lower bound it by terms that are known to be non-negative (S-procedure[15]). This gives

$$\begin{aligned}\phi_i(1 - (e_i^{T}PW^{-1}V_i\xi_i)^{2}) \geq\ & \underbrace{2\xi_i^{T}(g(W,\theta)\vec{A} + w - V_i\xi_i)}_{0}\\ & + \underbrace{((\mathbf{1}+d) - Z\vec{A})^{T}\Lambda_i((\mathbf{1}+d) + Z\vec{A})}_{\geq 0}\\ & + \underbrace{(\mathbf{1}+Dw)^{T}\Gamma_i(\mathbf{1}-Dw)}_{\geq 0}\end{aligned} \tag{25}$$

where $\Lambda_i \in D_+^{Tn_w}, \Gamma_i \in D_+^{n_w}$. In this way, we get the sufficient condition of (23), i.e., (25).

Through (22), (4), and (7), it can be proved that the right side of (25) is non-negative for all $w \in \mathcal{W}$, for all $A \in \mathcal{A}$. After that, (25) can be rewritten as the following quadratic form:

$$\mathbf{y}^T T_i(W, \phi_i, \Gamma_i, \Lambda_i, V_i)\mathbf{y} \geq 0, \forall \mathbf{y}, \tag{26}$$

where $\mathbf{y} = [1\ \vec{A}^T\ w^T\ -\xi_i^T]^T$ and matrix T_i is symmetrical. The invariance condition holds, if $T_i \geq 0$; i.e.,

$$T_i \triangleq \begin{bmatrix} r_i & -d^T\Lambda_i Z & \mathbf{0} & \mathbf{0} \\ * & Z^T\Lambda_i Z & 0 & g^T(W,\theta) \\ * & * & D^T\Gamma_i D & I_n \\ * & * & * & V_i + V_i^T - V_i^T L_i V_i \end{bmatrix} \geq 0, \tag{27}$$

where $L_i \triangleq \phi_i W^{-T} P^T e_i e_i^T P W^{-1}$, $r_i \triangleq \phi_i - \mathbf{1}^T\Lambda_i\mathbf{1} - \mathbf{1}_{n_w}^T\Gamma_i\mathbf{1}_{n_w} + d^T\Lambda_i d$ and $g(W,\theta)$ is as defined in (21). Take note of the nonlinearity of the block(2,4) in (27) and the nonlinear dependence of the block(4,4) on ϕ_i, V_i and W in (27), which will be resolved by introducing new matrix variables and appropriate scaling.

Theorem 1: Given the matrices X, X^+ and P, if there exists $W \in R^{n\times n}$, $W_1 \in R^n$ and variables $\phi_i \in R_+$, $\Lambda_i \in D_+^{Tn_w}, \Gamma_i \in D_+^{n_w}$, $X_i, V_i \in R^{n\times n}$ that satisfy LMIs for $i = 1, \ldots n_p$,

$$\begin{bmatrix} W + W^T - X_i & r_i P^T e_i \\ r_i e_i^T P & r_i \end{bmatrix} \geq 0, \tag{28}$$

$$\begin{bmatrix} r_i & -d^T\Lambda_i Z & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ * & Z^T\Lambda_i Z & 0 & g^T(W_1) & \mathbf{0} \\ * & * & D^T\Gamma_i D & I_n & \mathbf{0} \\ * & * & * & V_i + V_i^T & V_i^T \\ * & * & * & * & X_i \end{bmatrix} \geq 0, \tag{29}$$

where

$$g(W_1) \triangleq (W_1)^T \otimes I_n \in R^{n\times n^2}, \tag{30a}$$
$$W_1 \triangleq W\theta, \tag{30b}$$
$$r_i \triangleq \phi_i - \mathbf{1}^T\Lambda_i\mathbf{1} - \mathbf{1}_{n_w}^T\Gamma_i\mathbf{1}_{n_w} + d^T\Gamma_i d \in R, \tag{30c}$$

then the set $\mathcal{P}$ in (14) is RPI.

proof:

In order to resolve the nonlinearity in the block (2,4) of (27), let us introduce a new matrix variable $W_1 = W\theta$ such that $g(W_1)$ is linear.

First, let us prove the LMI condition (28) stated in Theorem 1. Introduce the positive-definite matrix variable X_i that satisfies

$$X_i^{-1} - L_i > 0 \Leftrightarrow X_i^{-1} - \phi_i W^{-T} P^T e_i e_i^T P W^{-1} > 0. \tag{31}$$

Applying Lemma 2 to (31), we obtain

$$\begin{bmatrix} X_i^{-1} & \phi_i W^{-T} P^T e_i \\ \phi_i e_i^T P W^{-1} & \phi_i \end{bmatrix} > 0, \tag{32}$$

By applying congruence transformation to (32) using the congruence transformation matrix $diag\{W, I_n\}$ which is full rank real matrix, then we get

$$\begin{bmatrix} W^T X_i^{-1} W & \phi_i P^T e_i \\ \phi_i e_i^T P & \phi_i \end{bmatrix} > 0. \tag{33}$$

To address the nonlinearity in the block (1,1) of (33), we perform the following steps:

$$\begin{aligned} W^T X_i^{-1} W &= (W - X_i)^T X_i^{-1} (W - X_i) \\ &\quad + W + W^T - X_i \\ &\geq W + W^T - X_i, \end{aligned} \tag{34}$$

then the $W^T X_i^{-1} W$ in (33) can be replaced by $W + W^T - X_i$, and we obtain a sufficient LMI condition for (33) as given in (28).

Next, let us prove the LMI condition (29) stated in Theorem 1. From (31), the condition (27) can be rewritten as

$$\begin{bmatrix} r_i & -d^T \Lambda_i Z & \mathbf{0} & \mathbf{0} \\ * & Z^T \Lambda_i Z & 0 & g^T(W_1) \\ * & * & D^T \Gamma_i D & I_n \\ * & * & * & V_i + V_i^T - V_i^T X_i^{-1} V_i \end{bmatrix} \geq 0, \tag{35}$$

then using the Schur complement, we obtain (29).

Considering the condition (20) of the robust invariance of the set Θ, the following Theorem 2 can be obtained.

Theorem 2 ([14])**:** Given the matrices X, X^+ and P, if there exists $W \in R^{n\times n}$ and variables $\phi_{ij} \in R_+$, $\Lambda_{ij} \in D_+^{Tn_w}$, $\Gamma_{ij} \in D_+^{n_w}$, $X_{ij}, V_{ij} \in R^{n\times n}$ that satisfy LMIs for $i = 1, \ldots n_p$ and $j = 1, \ldots 2\sigma$,

$$\begin{bmatrix} W + W^T - X_{ij} & r_{ij} P^T e_i \\ r_{ij} e_i^T P & r_{ij} \end{bmatrix} \geq 0, \tag{36}$$

$$\begin{bmatrix} r_{ij} & -d^T \Lambda_{ij} Z & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ * & Z^T \Lambda_{ij} Z & 0 & g^T(W, \theta^j) & \mathbf{0} \\ * & * & D^T \Gamma_{ij} D & I_n & \mathbf{0} \\ * & * & * & V_{ij} + V_{ij}^T & V_{ij}^T \\ * & * & * & * & X_{ij} \end{bmatrix} \geq 0, \tag{37}$$

where,

$$g(W, \theta^j) \triangleq \left(W\theta^j\right)^T \otimes I_n \in R^{n\times n^2}, \tag{38a}$$

$$r_{ij} \triangleq \phi_{ij} - \mathbf{1}^T \Lambda_{ij} \mathbf{1} - \mathbf{1}_{n_w}^T \Gamma_{ij} \mathbf{1}_{n_w} + d^T \Gamma_{ij} d \in R, \tag{38b}$$

then the set $\mathcal{P}$ in (14) is RPI.

We aim to identify the largest set $\mathcal{P}$in (14) that satisfies the state constraints outlined in (4a) and the sufficient conditions for invariance specified in (28) and (29) (or alternatively, (36) and (37)). Given a matrix P, it is

Algorithm 1 Computing RPI set.

Input: F,D,P;
Output: Optimal values for $\mathbf{W}$, $\mathbf{X_i}$;
Objective function $\max_{Z_{SDP}} \quad \log\det(W)$
Optimization variables $Z_{SDP} \triangleq (W, W_1, X_i, V_i, \phi_i, \Lambda_i, \Gamma_i)$
Symmetry constraint $W = W'$ (in order for the objective function to be concave)
State constraints (17)
Invariance conditions (28), (29)

Algorithm 2 Computing RPI set.

Input: F,D,P;
Output: Optimal values for $\mathbf{W}$, $\mathbf{X_{ij}}$;
Objective function $\max_{Z_{SDP}} \quad \log\det(W)$
Optimization variables $Z_{SDP} \triangleq \left(W, X_{ij}, V_{ij}, \phi_{ij}, \Lambda_{ij}, \Gamma_{ij}\right)$
Symmetry constraint $W = W'$ (in order for the objective function to be concave)
State constraints (17)
Invariance conditions (36), (37)

known that the volume of P is proportional to its determinant, denoted as $|det(W)|$ [19]. Consequently, we can determine the largest set P by formulating a SDP problem. Algorithms 1 and 2 based on the one-step method are given below.

Therefore, in order to obtain the desired large volume of RPI sets, we need to solve the determinant maximization problem under LMI conditions. However, this problem is easy to solve only if W is symmetric [20]. The symmetry of W would introduce conservatism [21]; thus, we introduce an iterative algorithm based on Algorithm 1. In an iterative algorithm, W does not need to maintain symmetry, and the algorithm also reduces the conservatism caused by the introduction of (34) [19]. At the k-th iteration, let W_k and X_{ik} be the values of the variables W, X_i. At each subsequent iteration, the volume of the RPI set increases, i.e., $|det(W_{k+1})| \geq |det(W_k)|$, if the following LMI condition is imposed,

$$W^T W_k + W_k^T W - W_k^T W_k \geq Z > 0, \tag{39}$$

where $Z = Z^T \in R^{n\times n}$ is the new matrix variable.

From [22], we get

$$\begin{aligned}(W - X_i Z_{ik})^T X_i^{-1} (W - X_i Z_{ik}) &= W^T X_i^{-1} W \\ &-W^T Z_{ik} - Z_{ik}^T W + Z_{ik}^T X_i Z_{ik} \geq 0,\end{aligned} \tag{40}$$

thus we can obtain:

$$W^T X_i^{-1} W \geq W^T Z_{ik} + Z_{ik}^T W - Z_{ik}^T X_i Z_{ik}, \tag{41}$$

where $Z_{ik} \triangleq X_{ik}^{-1} W_k$, and then the linear term to the right of (41) can be used to substitute for the nonlinear term $W^T X_i^{-1} W$ in (29). From (41), the condition (28) can be rewritten as follows,

$$\begin{bmatrix} W^T Z_{ik} + Z_{ik}^T W - Z_{ik}^T X_i Z_{ik} & r_i P^T e_i \\ r_i e_i^T P & \phi_i \end{bmatrix} \geq 0. \tag{42}$$

Therefore, we can obtain the following iterative algorithm based on Algorithm 1:

iterative algorithm Computing RPI set.

Input: F,D,P,W^q, X_i^q;
Output: Optimal values for **W**;
Objective function $\max_{Z_{SDP}} \quad \log \det(Z)$
Optimization variables $Z_{SDP} \triangleq (W, X_i, V_i, \phi_i, \Lambda_i, \Gamma_i, Z)$
State constraints (17)
Invariance conditions (28), (42)

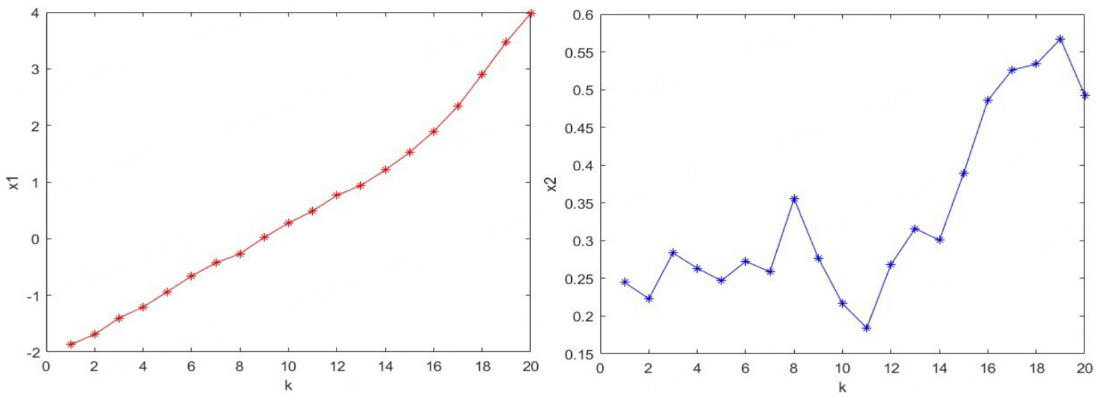

Figure 1. The data samples collected from the liner discrete-time system (43).

Remark 2: Algorithm 2 has $10\sigma n_p + 1$ optimization variables, the invariance conditions (36), (37) consist of $2\sigma n_p$ linear matrix inequalities respectively. The iterative algorithm has $5n_p + 2$ optimization variables; the invariance conditions (28), (42) consist of n_p linear matrix inequalities respectively. This suggests that both the iterative algorithm has fewer optimization variables and fewer LMIs for invariant conditions than Algorithm 2. In addition, Algorithm 1 is more conservative than iterative algorithm (see Example 2).

3. RESULTS

The algorithms in the experiments are implemented in Matlab by using CVX[23] and solver SeDuMi. And the MPT toolbox is used to manage polytopes[24]. The following liner discrete-time system is considered:

$$x(k+1) = \underbrace{\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}}_{A} x(k) + w(k). \tag{43}$$

The matrice A is unknown, but it is only used to gather data. $T = 20$ samples of data are collected by system (43), as shown in Figure 1. Assume the disturbance w in system (43) ranges between -0.1 and 0.1, with the state constraints being $(x_1, x_2) \in [-2, 2] \times [-2, 2]$.

Example 1: By taking different matrices P, the complexity of the RPI set is denoted as $n_p = 2, 3, 4$ respectively, and the corresponding matrices are computed as follows

Table 1. $|det(W)|$ **(Algorithm 1)** vs. **complexity of the matrix** P

complexity	$n_p=2$	$n_p=3$	$n_p=4$
$\lvert det(W)\rvert$	3.5	384.0	8001.2

$$P_2=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}, P_3=\begin{bmatrix}10 & 10\\ 10 & 0\\ 1 & 11\end{bmatrix}, P_4=\begin{bmatrix}-18 & -55\\ 18 & 55\\ 55 & -18\\ 55 & 18\end{bmatrix}.$$

The matrice W is computed by running Algorithms 1 and 2 as follows:

$$W_{21}=\begin{bmatrix}3.5 & 0\\ 0 & 1\end{bmatrix}(\text{Algorithm 1, 2.039 s}), W_{22}=\begin{bmatrix}3.5 & 0\\ 0 & 1\end{bmatrix}(\text{Algorithm 2, 2.770 s}),$$

$$W_{31}=\begin{bmatrix}35 & 1\\ 1 & 11\end{bmatrix}(\text{Algorithm 1, 2.418 s}), W_{32}=\begin{bmatrix}35 & 1\\ 1 & 11\end{bmatrix}(\text{Algorithm 2, 4.552 s}),$$

$$W_{41}=\begin{bmatrix}151.4106 & 18.1135\\ 18.1135 & 55.0114\end{bmatrix}(\text{Algorithm 1, 2.410 s}), W_{42}=\begin{bmatrix}151.4106 & 18.1135\\ 18.1135 & 55.0114\end{bmatrix}(\text{Algorithm 2, 4.725 s}),$$

where W_{21} and W_{22} are obtained from P_2; W_{31} and W_{32} are obtained by taking P_3; W_{41} and W_{42} are obtained by taking P_4.

It is discovered from experiment results that while Algorithm 2 yields the same results, Algorithm 1 runs faster and requires fewer optimization variables (see Remark 2) than Algorithm 2.

Example 2: The RPI sets with complexities $n_p = 2, 3, 4$ (i.e., P_2, P_3, P_4 are selected respectively) which are obtained by Algorithm 1 are as follows

$$W_2=\begin{bmatrix}3.5 & 0\\ 0 & 1\end{bmatrix}, W_3=\begin{bmatrix}35 & 1\\ 1 & 11\end{bmatrix}, W_4=\begin{bmatrix}151.4106 & 18.1135\\ 18.1135 & 55.0114\end{bmatrix}.$$

The corresponding RPI sets obtained by Algorithm 1 are shown in Figure 2.

After five times iterations of the iterative algorithm, the RPI sets $\mathcal{P}$ with complexities $n_p = 2, 3, 4$ are obtained as

$$W_2=\begin{bmatrix}3.5 & 0\\ 0 & 1\end{bmatrix}, W_3=\begin{bmatrix}35 & 1\\ 1 & 11\end{bmatrix}, W_4=\begin{bmatrix}192.5402 & -63.3969\\ 18.1136 & 55.0115\end{bmatrix}.$$

The corresponding RPI sets obtained by iterative algorithm are shown in Figure 3.

We find that the volume of RPI sets obtained by the iterative algorithm is larger than that obtained by Algorithm 1 (see Table 1 and Table 2), which indicates that Algorithm 1 is more conservative than the iterative algorithm. For the given matrix P, the volume of $\mathcal{P}$ is proportional to the determinant $|det(W)|$[19]. And as n_p rises, the absolute value of the determinant of W grows; thus, the volume of the RPI set increases (see Table 1 and Figure 4), so that an invariant set with a higher volume can be obtained by adding n_p as a parameter.

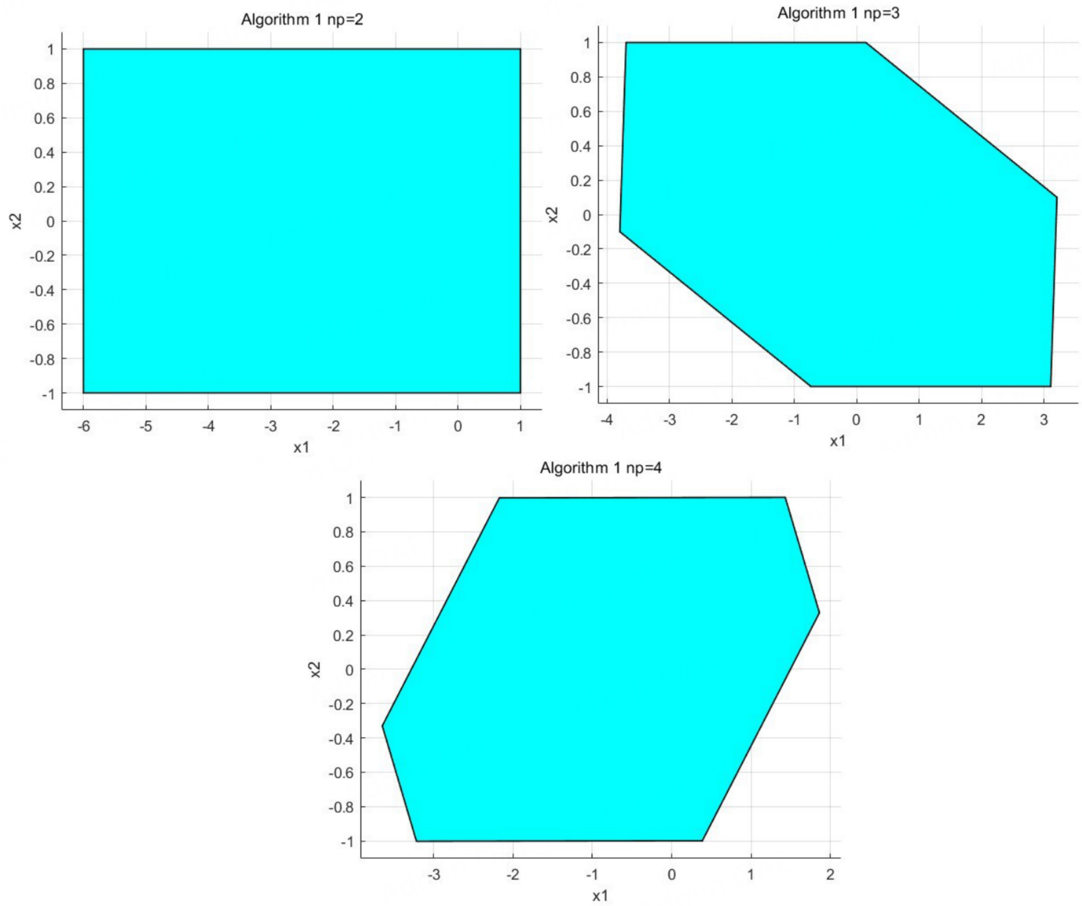

Figure 2. Maximum volume RPI sets $\mathcal{P}$ with different complexities: n_p = 2 (left), n_p = 3 (right), n_p = 4 (bottom).

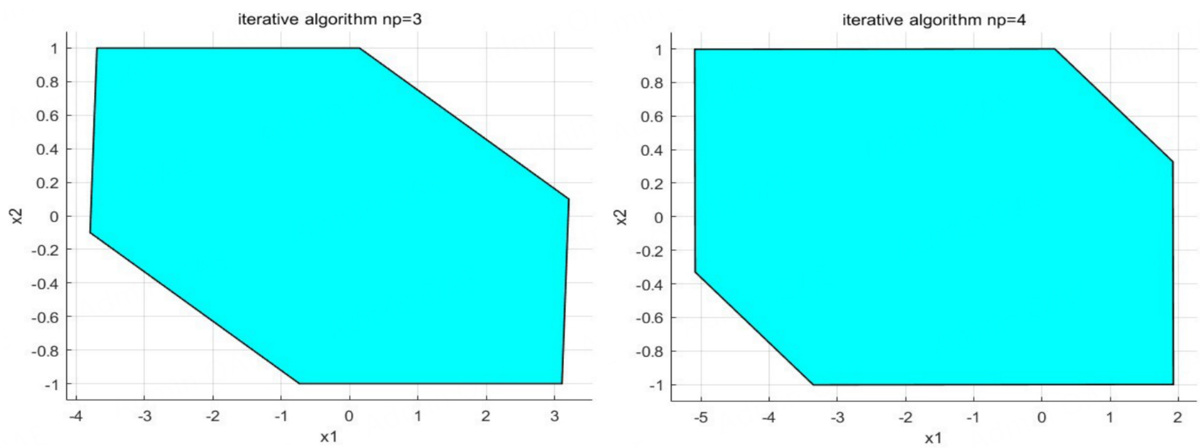

Figure 3. Maximum volume RPI sets $\mathcal{P}$ with different complexities: n_p = 3 (left), n_p = 4 (right).

Table 2. $|det(W)|$ **(iterative algorithm)** vs. **complexity of the matrix** P

complexity	$n_p = 2$	$n_p = 3$	$n_p = 4$
$\|det(W)\|$	3.5	385.0	11740.3

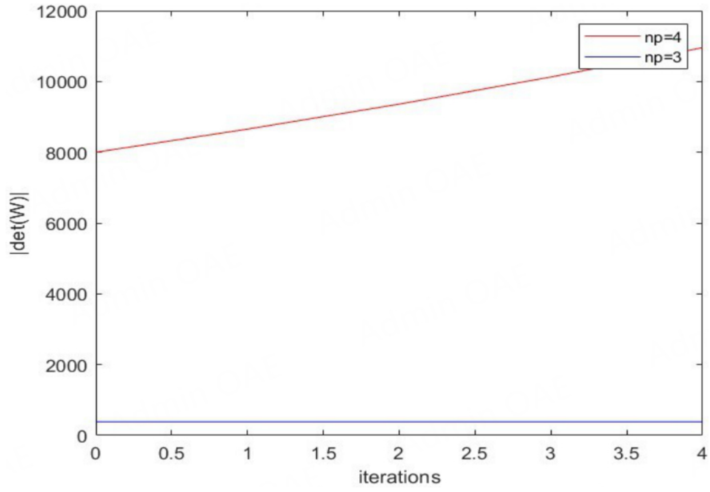

Figure 4. $|det(W)|$ $vs.$ iterations of iterative algorithm.

4. DISCUSSION

A direct data-driven method is proposed to calculate the robust positive invariant (RPI) sets for discrete-time systems. To further reduce conservatism, an iterative algorithm based on Algorithm 1 is introduced. This approach does not require prior knowledge of the model or system identification. Future research could extend this methodology to nonlinear systems. Additionally, while the RPI sets discussed here are symmetric, the exploration of asymmetric RPI sets is another potential area for investigation.

5. CONCLUSIONS

This paper proposes a direct data-driven method to calculate RPI sets for discrete-time systems by deriving a set of invariance conditions expressed as linear matrix inequalities (LMIs). Subsequently, we maximize the volume of the invariant set using a SDP problem. We have developed two one-step algorithms based on LMIs to compute the RPI sets; experimental results indicate that Algorithm 1 requires fewer optimization variables compared to Algorithm 2 and demonstrates superior computational efficiency. Additionally, we have introduced an iterative approach based on Algorithm 1 to further reduce conservatism. Numerical examples have verified the effectiveness of the proposed algorithm.

DECLARATIONS

Acknowledgments
We would like to express our great appreciation to the editors and reviewers.

Authors' contributions
Conceptualization, methodology, resources, supervision: Yang H
Software, validation, writing - original draft preparation: Du Q
Formal analysis, investigation, writing - review and editing, visualization: Yang H, Du Q
All authors have read and agreed to the published version of the manuscript.

Availability of data and materials
The data cannot be shared publicly as the partner (company) does not permit public disclosure. It is available from the corresponding author upon reasonable request.

Financial support and sponsorship
None.

Conflicts of interest

Both authors declared that there are no conflicts of interest.

Ethical approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Copyright

REFERENCES

1. Blanchini F, Miani S. Set-theoretic methods in control, 3th ed. Springer, 2008. Available from: https://link.springer.com/book/10.1007/978-3-319-17933-9 [Last accessed on 28 Nov 2024].
2. Castelan EB, Tarbouriech S. Positively invariant polyhedral sets for discrete-time singular systems with additive perturbations. In Proceedings of the 35th IEEE Conference on Decision and Control; 13 December 1996; Kobe, Japan. DOI
3. Blanchini F. Set invariance in control. *Automatica* 1999;35:1747-67. DOI
4. Kolmanovsky I, Gilbert EG. Theory and computation of disturbance invariant sets for discrete-time linear systems. *Math Probl Eng* 1998;4:317-67. DOI
5. Raković SV, Kerrigan EC, Mayne DQ, Kouramas KI. Optimized robust control invariance for linear discrete-time systems: theoretical foundations. *Math Probl Eng* 2007;43:831-41. DOI
6. Wang P, Feng X, Li W. Design of robust positively invariant set for nonholonomic vehicle. In Proceedings of the 2016 Chinese Control and Decision Conference (CCDC); 28-30 May 2016; Yinchuan, China. DOI
7. Esterhuizen W, Aschenbruck T, Streif S. On maximal robust positively invariant sets in constrained nonlinear systems. *Automatica* 2020;119:109044. DOI
8. Rakovic SV, Kerrigan EC, Kouramas K, Mayne DQ. Invariant approximations of the minimal robust positively invariant set. *IEEE Trans Automat Contr* 2005;50:406-10. DOI
9. Trodden P. A one-step approach to computing a polytopic robust positively invariant set. *IEEE Trans Automat Contr* 2016;61:4100-5. DOI
10. Xue B, Zhan N. Robust invariant sets computation for discrete-time perturbed nonlinear systems. *IEEE Trans Automat Contr* 2022;67:1053-60. DOI
11. Hou ZS, Wang Z. From model-based control to data-driven control: survey, classification and perspective. *Inf Sci* 2013;235:3-35 DOI
12. Zhong B, Zamani M, Caccamo M. Synthesizing safety controllers for uncertain linear systems: a direct data-driven approach. In Proceedings of the 35th IEEE Conference on Decision and Control; 23-25 August 2020; Trieste, Italy. DOI
13. Bisoffi A, De Persis C, Tesi P. Data-based guarantees of set invariance properties. *IFAC-PapersOnLine* 2020;53:3953-8 DOI
14. Mejari M, Gupta A. Direct data-driven computation of polytopic robust control invariant sets and state-feedback controllers. In Proceedings of the 62nd IEEE Conf. on Decision and Control (CDC); 13-15 December 2023; Singapore, Singapore. DOI
15. Scherer CW. A full block S-procedure with applications. In Proceedings of the 36th IEEE Conf. on Decision and Control; 12 December 1997; San Diego, CA, USA. DOI
16. Abadir KM, Magnus JR. Matrix algebra, 3th ed. Cambridge University Press; 2005. DOI
17. Cottle RW. Manifestations of the Schur complement. *Linear Algebra Appl* 1974;8:189-211. DOI
18. Willems JC, Markovsky I, Rapisarda P, De Moor BLM. A note on persistency of excitation. *Syst Control Lett* 2005;54:325-9. DOI
19. Gupta A, Köroğlu H, Falcone P. Computation of robust control invariant sets with predefined complexity for uncertain systems. *Intl J Robust Nonlinear* 2021;31:1674-88. DOI
20. Vandenberghe L, Boyd S, Wu SP. Determinant maximization with linear matrix inequality constraints. *SIAM J Matrix Anal Appl* 1998;19:499-533. DOI
21. Gupta A, Köroğlu H, Falcone P. Restricted-complexity characterization of control-invariant domains with application to lateral vehicle dynamics control. In Proceedings of the 2017 IEEE 56th Annual Conference on Decision and Control (CDC); 12-15 December 2017; Melbourne, VIC, Australia. DOI
22. Gupta A, Köroğlu H, Falcone P. Computation of low-complexity control-invariant sets for systems with uncertain parameter dependence. *Automatica* 2019;101:330-7. DOI
23. CVX: matlab software for disciplined convex programming; 2020. Available from: http://cvxr.com/cvx [Last accessed on 28 Nov 2024].
24. Herceg M, Kvasnica M, Jones CN, Morari M. Multi-parametric toolbox 3.0. In Proceedings of the 2013 European Control Conference (ECC); 17-19 July 2013; Zurich, Switzerland. DOI