International Journal of Robust and Nonlinear Control

WILEY

RESEARCH ARTICLE OPEN ACCESS

Control Systems Design With Enlarged Stability Domains for Nonlinear Constrained Systems via Sum-of-Squares Optimization

Bhaskar Biswas | Dmitry Ignatyev | Argyrios Zolotas | Antonios Tsourdos

School of Aerospace, Transport and Manufacturing, Cranfield University, Cranfield, UK

Correspondence: Bhaskar Biswas (bhaskar.biswas@cranfield.ac.uk)

Received: 12 August 2024 | **Revised:** 13 December 2025 | **Accepted:** 26 May 2026

Keywords: constraint polynomial system | nonlinear control system | region of attraction | safe control law | shape function | sum-of-squares optimization | S-procedure

ABSTRACT

Designing safe control laws for nonlinear systems is challenging, especially when ensuring stability under actuator saturation and state constraints. A key aspect is embedding controllers with a Region of Attraction (ROA), which defines initial conditions guaranteeing convergence to a stable equilibrium point (EP). Controllers with enlarged ROAs are desirable for improved stability, performance, and robustness, but their design is a complex and computationally intensive task. This paper proposes a new methodology for designing controllers with expanded ROAs for nonlinear polynomial systems facing actuator saturation and state constraints, utilizing the Sum-of-Squares (SOS) optimization technique. Our approach proposes the use of multiple variable-sized regions defined by multiple positive definite functions known as Shape Functions and the S-procedure within the SOS optimization framework. This approach synthesizes a control law with enlarged ROA, even in the presence of constraints in the system. We developed a numerically efficient algorithm to solve this problem, yielding superior results. Extensive numerical simulations demonstrate the effectiveness of our approach. Additionally, we verified the estimated ROA under the designed control law using Monte Carlo simulations, confirming the robustness and practical applicability of our proposed method.

1 | Introduction

The development of effective control laws for nonlinear systems poses a pivotal and intricate challenge within the realm of control theory. A key challenge in this pursuit is ensuring system stability across varied conditions. An established approach to address this challenge involves embedding controllers with a Region of Attraction (ROA), defining the domain of the initial conditions guaranteed to converge to a stable equilibrium point (EP) under the controller's influence. However, designing controllers with an ROA becomes notably intricate when systems face the constraints of actuator saturation and state limitations. In real-world systems, actuator saturation and state constraints are pervasive challenges impacting performance and stability. Crafting a controller with an ROA adept at handling these real-world challenges becomes imperative. Controllers with an enlarged ROA are particularly desirable, accommodating a broader range of initial conditions, enhancing stability, performance, and overall robustness in nonlinear control systems. Despite its desirability, formulating a controller with an enlarged ROA remains

Abbreviations: CLF, control Lyapunov function; EP, equilibrium point; LD, linearized dynamics; LF, Lyapunov function; LMI, linear matrix inequality; ROA, region of attraction; SF, shape function; SOS, sum-of-squares.

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2026 The Author(s). *International Journal of Robust and Nonlinear Control* published by John Wiley & Sons Ltd.

a challenging and computationally intricate task. Innovative numerical approaches, such as Sum-of-Squares (SOS) optimization-based methods, offer promising avenues for designing safe controllers with an ROA tailored to nonlinear polynomial systems amidst actuator saturation and state constraints.

In the vast expanse of research literature, a plethora of methodologies have emerged to tackle the intricate task of designing controllers for systems plagued by actuator saturation and state constraints. Among these approaches, paper [1] introduces a semi-global stabilization strategy that effectively designs a controller for unstable linear systems capable of achieving exponential stabilization despite the presence of saturating actuators. Using linear matrix inequalities (LMI), [2] and [3] developed a method to expand the ROA while designing the control law for linear systems operating under the influence of saturated actuators. Paper [4] presents a technique centered around the barrier Lyapunov function (LF), which results in the design of a robust controller with an increased ROA. Similarly, paper [5] proposed a method for positive linear systems that leverages convex optimization with LMI constraints to design a controller that estimates ROA. The authors of the papers [6] and [7] devised a technique incorporating anti-windup to assess the ROA for a linear system subjected to actuator saturation. All the methods mentioned in the above papers yield good results, but they are limited to linear systems. To overcome this limitation, several approaches, including the SOS optimization method, have been developed to design control laws with an ROA for constrained nonlinear systems. In this study, we propose a method using the SOS optimization technique to design a control law with an enlarged ROA for the nonlinear polynomial systems subject to actuator saturation and state constraints.

The SOS optimization technique has emerged as a powerful tool in the realm of polynomial optimization. At its core lies the concept of SOS polynomials, which are nonnegative polynomials that can be expressed as a SOS of other polynomials. One notable application of SOS polynomials is in representing the LF, as both SOS polynomials and LFs share the property of being nonnegative, although this representation is not universal across all LFs. The underlying principles and theories relating to SOS polynomials have been thoroughly explored in academic literature, as evidenced by the comprehensive discussions found in [8]. To effectively leverage the SOS optimization technique, it is essential to grasp the conditions that enable the expression of a nonnegative polynomial as a relaxed SOS polynomial. In this regard, comprehensive insights into the relaxation procedure and the necessary conditions for achieving a relaxed SOS polynomial can be found in the insightful discourse presented in [9] and [10]. To assist researchers in understanding and applying the SOS optimization technique, various tutorials have been made available within the literature. Noteworthy references in this regard include [11–16] and [17]. These tutorials serve as valuable resources for individuals seeking to gain a deeper understanding of the SOS optimization method and its practical implementations. To effectively solve problems related to SOS optimization, researchers can leverage the functionality offered by specialized software tools. Notable examples of such tools include SOSOPT [18], SOSTOOLS [19] and BiSOS [20]. These software tools provide a range of features and capabilities specifically designed to address SOS optimization-related challenges.

The authors of papers [21–24], and [25] used the SOS optimization technique to estimate the ROA for nonlinear polynomial systems. However, these papers are not related to the design of control systems. The references [26–29], and [30] designed a feedback control law for the polynomial systems using the SOS optimization method without estimating the related ROA. In [31], the authors created a feedback control law with an ROA using SOS optimization without considering system constraints. The papers referenced as [32–34], and [35] utilized the SOS optimization algorithm to design a control system with an ROA in the presence of saturation in the control input. Additionally, papers [32] and [33] employed a deterministic system, paper [34] considered a nonaffine system, and paper [35] employed a stochastic system. In paper [36], the authors synthesized a control law with ROA for the vehicle lateral control using the SOS optimization method. However, none of the studies from [32–36] considered the limitations imposed by the system's states. Additionally and most importantly, a significant concern is that these authors [31–36] relied on a single positive definite function, known as the shape function (SF), within the SOS optimization method to achieve a larger ROA during control system synthesis. However, when the actual ROA is asymmetric, using a single SF results in a conservative outcome. This occurs because the algorithm increases the size of the SF in each iteration, aiming to find an LF that encompasses it while synthesizing the control law. Eventually, as the SF nears the unstable region in certain directions of the actual asymmetric ROA, the algorithm halts, even though a substantial portion of the ROA remains uncovered in other directions.

In this study, we build upon our previous conference paper [37] to address the aforementioned limitations by proposing a novel approach for designing safe control systems with an expanded ROA, taking into account saturation in control and state constraints, irrespective of the symmetry of the actual ROA for a deterministic polynomial system. Notably, our conference paper did not consider saturation in control and state constraints and presented only one illustrative example without result verification. However, in this journal paper, we introduce an enriched methodology that incorporates both control saturation and state constraints.

To achieve this goal, we propose the utilization of multiple shape functions within the SOS optimization framework, complemented by the introduction of the S-procedure to address system constraints. This approach facilitates the control system's design while estimating enlarged associated ROAs with constraints on both system state and control input, even in scenarios featuring asymmetric or constrained ROAs. Subsequently, we have developed an advanced and comprehensive computational algorithm, improving upon the algorithm presented in our previous conference paper, to solve the problem and enhance results efficiently. To demonstrate the effectiveness of our approach, we present three diverse examples, comprising two 2-D cases and one 3-D case. Furthermore, we rigorously validate the obtained ROA through Monte Carlo simulations for all three examples, thereby ensuring the reliability and robustness of our findings. Through this extended work, we introduce a refined methodology and provide a more comprehensive evaluation, significantly enhancing the applicability and impact of our contributions.

The structure of this paper is organized as follows: Section 2 provides an overview of the background and formulates the problem. Section 3 discusses the proposed concept and its application in designing controllers with an enlarged ROA for constrained systems, along with the related algorithm. The simulation results based on the proposed approach are presented in Section 4. Finally, Section 5 concludes with a summary of our findings and final reflections.

2 | Background and Problem Statement

Let us consider the following polynomial autonomous affine dynamical system with constraints in control input and states,

$$\dot{x} = f(x) + \sum_{i=1}^{m} g_i(x)\text{sat}(u_i),\ f(0) = 0. \tag{1a}$$

The saturated control input is defined in the following way,

$$\text{sat}(u_i) = \begin{cases} \alpha_{\max_i} & \text{if } u_i > \alpha_{\max_i} \\ u_i & \text{if } \alpha_{\min_i} \le u_i \le \alpha_{\max_i} \\ \alpha_{\min_i} & \text{if } u_i < \alpha_{\min_i} \end{cases} \quad [i = 1, \ldots, m] \tag{1b}$$

The safe region of the system is defined by the following state constraints:

$$\{x \ : \ h_i(x) \le \lambda_i\} \quad [i = 1, \ldots, t] \tag{1c}$$

- $x(t) \in \mathbb{R}^n$, $u_i \in \mathbb{R}$ and $u = (u_1, \ldots, u_m) \in \mathbb{R}^m$
- $f(x) : \mathbb{R}^n \to \mathbb{R}^n$ and $g_i(x) : \mathbb{R}^n \to \mathbb{R}^n$ are smooth vector fields
- $\alpha_{\max_i}, \alpha_{\min_i}$ and $\lambda_i \in \mathbb{R}$.
- It should be noted that, if an unsafe region is specified by $\{x: h_i(x) > \lambda_i\}$, the corresponding safe region can be obtained by reversing the inequality sign.

Without loss of generality, we can assume that $x = 0$ is the EP, and the system can be made stable under the proper control input.

Definition 1 (Region of attraction). The ROA refers to the domain that encompasses the EP, including the EP itself. When the initial points of the trajectories remain within this region, the trajectories will eventually converge to the EP as time advances. Consider $\phi(x)$ is the closed-loop solution of the system expressed by Equation (1a). The ROA of origin, $x = 0$ (EP) is, [38],

$$R_A = \{x \in \mathbb{R}^n \ : \ \phi(x) \text{ is } \forall t \ge 0 \text{ and } \lim_{t\to\infty} \phi(x) \to 0\}$$

Lemma 1. *If there exist a continuously scalar function $V(x) : \mathbb{R}^n \to \mathbb{R}$ and a positive scalar $\gamma \in \mathbb{R}^+$ for the system dynamics represented by Equation (1), such that,*

$$V(x) > 0 \forall x \neq 0 \text{ and } V(0) = 0 \tag{2a}$$

$$\Omega_\gamma = \{x \ : \ V(x) \le \gamma\} \text{ is bounded} \tag{2b}$$

$$\Omega_\gamma \subseteq \left\{ x \ : \ \frac{\partial V}{\partial x}\left(f(x) + \sum_{i=1}^{m} g_i(x)\ \text{sat}(u_i) \right) < 0 \right\} \cup \{0\} \tag{2c}$$

Then the origin is asymptotically stable and $\Omega_\gamma \subseteq R_A$ [39].

The lemma mentioned above establishes a strict LF, denoted as $V(x)$. By optimizing the parameter γ, it becomes possible to obtain the maximum estimate of Ω_γ. One method for achieving this optimization is through the application of the SOS optimization technique. Notably, this approach enables the simultaneous design of a controller and LF, $V(x)$ estimation, while also optimizing the value of γ.

2.1 | Sum-of-Squares Optimization

SOS optimization is a powerful technique in polynomial optimization that uses nonnegative SOS polynomials. It enables the simultaneous design of a controller, estimation of LF ($V(x)$), and optimization of γ. The following important definitions, Lemma, and Theorem are crucial for this study [32] and [40].

Definition 2 (Monomial). A Monomial m_α in n variables is a function defined as $m_\alpha(x) = x^\alpha = x_1^{\alpha_1} x_2^{\alpha_2} \ldots x_n^{\alpha_n}$ for $\alpha \in \mathbb{Z}_+^n$. The degree of a monomial is defined, $\deg m_\alpha = \sum_{i=1}^n \alpha_i$.

Definition 3 (Polynomial). A polynomial $f(x)$ in n variables is a finite linear combination of monomials and expressed as, $f(x) = \sum_\alpha c_\alpha m_\alpha = \sum_\alpha c_\alpha x^\alpha$ with $c_\alpha \in \mathbb{R}$. Define $\mathcal{R}_n$ to be the set of all polynomials in n variables. The degree of $f(x)$ is defined as $\deg f(x) = max_\alpha \deg m_\alpha$ with $c_\alpha \neq 0$.

Definition 4 (Sum-of-Squares (SOS) polynomial). A polynomial, $s(x) \in \mathcal{R}_n$ is a SOS polynomial if there exist polynomials $f(x)_i \in \mathcal{R}_n$ such that

$$s(x) = \sum_{i=1}^{t} f_i(x)^2,\ f_i(x) \in \mathcal{R}_n, \quad i = 1, \ldots, t.$$

Again, $s(x)$ is an SOS if there exists $Q \ge 0$ such that $s(x) = Z(x)^T Q Z(x)$, where $Z(x)$ is the vector of monomials. The set of SOS polynomials in n variables is defined as

$$\sum_n = \left\{ s(x) \in \mathcal{R}_n | s(x) = \sum_{i=1}^{t} f_i(x)^2,\ f_i(x) \in \mathcal{R}_n, \quad i = 1, \ldots, t \right\}$$

If $s(x) \in \sum_n \forall x \in \mathbb{R}^n$ then always $s(x) \ge 0$.

Theorem 1 (S-procedure). *Let us consider that the polynomials $Q_1(x)$ and $Q_2(x)$ are in $\mathcal{R}_n$. and its enclosed regions are defined by $A_{Q_1} = \{x \in \mathbb{R}^n : Q_1(x) \le 0\}$ and $A_{Q_2} = \{x \in \mathbb{R}^n : Q_2(x) \le 0\}$, respectively. The set A_{Q_2} will be a subset of A_{Q_1} ($A_{Q_2} \subseteq A_{Q_1}$) if there exist polynomials $s(x) \in \sum_n$ and its satisfy the below condition [11] and [41],*

$$-Q_1(x) + Q_2(x)s(x) \in \sum_n. \tag{3}$$

Theorem 2 (Union theorem). *Let us consider the given polynomials $V(x)$, $p_1(x), \ldots, p_n(x)$ define sets A_V, $A_1, \ldots, A_n$ such that,*

$$A_V = \{x \in \mathbb{R}^n : V(x) - \gamma \le 0\}$$
$$A_1 = \{x^1 \in \mathbb{R}^n : p_1(x) - \beta_1 \le 0\}$$
$$\vdots$$
$$A_n = \{x^n \in \mathbb{R}^n : p_n(x) - \beta_n \le 0\}.$$

Here, $\{\gamma, \beta_1, ..\beta_n\} \in \mathbb{R}^+$. *Now if there exit polynomials* $s_1 \in \sum_n, .., s_n \in \sum_n$, *such that,*

$$-(V(x)-\gamma)+(p_1(x)-\beta_1)s_1 \in \sum_n \tag{4a}$$

$$\vdots$$

$$-(V(x)-\gamma)+(p_n(x)-\beta_n)s_n \in \sum_n . \tag{4n}$$

Then, $A_1 \cup A_2 \cup \cdots \cup A_n \subseteq A_V$ [40] *and* [42].

2.2 | SOS Optimization Form of Control System Design With ROA Estimation

The goal is to synthesize a state feedback controller $u = K(x)$, where $K \in \mathcal{R}^n$, that achieves an enlarged estimated ROA, denoted as Ω_γ, for the constrained system represented by Equation (1a). Lemma 1 is instrumental in designing a controller with an enhanced estimated ROA, and this is accomplished using SOS optimization. The SOS optimization achieves this by optimizing Ω_γ through finding an optimized LF, V, with a level set defined by γ, which simultaneously determines a feedback control law $u = K(x)$ to satisfy Equation (2c). When Lemma 1 is applied within the SOS optimization framework, the controller design with ROA estimation can be formulated as follows [32],

$$\gamma^* = \max_{s_0, s_k \in \sum_n, V(x), K_i(x) \in \mathcal{R}_n} \gamma$$

$$\text{Subject to:} \quad V - l_1 \in \sum_n \tag{5a}$$

$$-\left[\frac{\partial V}{\partial x}\left(f(x)+\sum_{i=1}^{m} g_i(x)K_i(x)\right)s_k + l_2\right] + (V-\gamma)s_0 \in \sum_n \tag{5b}$$

In this context, l_1, l_2, s_0, and s_k are SOS polynomials. Equation (5) yields a control law with an estimated ROA. However, this designed control law does not account for the saturation constraint represented by Equation (1b). When saturation is incorporated into the control input, Equation (5) is modified as follows [32],

$$\gamma^* = \max_{s_0, s_k, s_{\alpha_{max_1}}, s_{\alpha_{min_1}}, \ldots, s_{\alpha_{max_m}}, s_{\alpha_{min_m}} \in \sum_n, V(x), K_i(x) \in \mathcal{R}_n} \gamma$$

$$\text{Subject to}: \quad V(x) - l_1 \in \sum_n \tag{6a}$$

$$-\left[\frac{\partial V}{\partial x}\left(f(x)+\sum_{i=1}^{m} g_i(x)K_i(x)\right)s_k + l_2\right] + (V(x)-\gamma)s_0 \in \sum_n \tag{6b}$$

$$-\left[K_i(x) - \alpha_{\max_i}\right] + (V(x)-\gamma)s_{\alpha_{\max_i}} \in \sum_n, \quad i = [1, \ldots, m] \tag{6c}$$

$$-\left[\alpha_{\min_i} - K_i(x)\right] + (V(x)-\gamma)s_{\alpha_{\min_i}} \in \sum_n, \quad i = [1, \ldots, m] \tag{6d}$$

Here, $s_{\alpha_{max_i}}$ and $s_{\alpha_{min_i}}$ are SOS polynomials. Equation (6) designs a control system that considers saturation while estimating the ROA. However, the estimated ROA tends to be overly conservative due to the early convergence of the algorithm. To address this, [32] introduced a new function called a shape function. This function, used alongside Equation (6), aims to enhance the ROA estimation during the synthesis of the control system.

Definition 5 (Shape function). The shape function (SF) is a positive definite function that delineates a region enclosed within the boundary defined by the LF ($V(x) \le \gamma$). The SF gradually extends its boundaries, prompting the LF to expand until it reaches the unstable region. The SF controls the size and shape of the area under the LF. The variable-sized region under the SF is defined as follows [32, 40],

$$\epsilon_\beta = \{x \in \mathbb{R}^n : p(x) \le \beta\} \subseteq \Omega_\gamma \Rightarrow \epsilon_\beta = \{x \in \mathbb{R}^n : p(x) \le \beta\} \subseteq \{x \in \mathbb{R}^n : V(x) \le \gamma\} \quad \text{[Using Equation (2b)]} \tag{7}$$

If we introduce the SF (Equation (7)) into Equation (6) then we get the following modified optimization form of Equation (6), [32],

$$\beta^* = \max_{s_0, s_k, s_{\alpha_{max_1}}, s_{\alpha_{min_1}}, \ldots, s_{\alpha_{max_m}}, s_{\alpha_{min_m}}, s_1 \in \sum_n, V(x), K_i(x) \in \mathcal{R}_n, \gamma \in \mathbb{R}^+} \beta$$

$$\text{Subject to}: \quad V(x) - l_1 \in \sum_n, \tag{8a}$$

$$-\left[\frac{\partial V}{\partial x}\left(f(x)+\sum_{i=1}^{m} g_i(x)K_i(x)\right)s_k + l_2\right] + (V(x)-\gamma)s_0 \in \sum_n \tag{8b}$$

$$-\left[K_i(x) - \alpha_{max_i}\right] + (V(x)-\gamma)s_{\alpha_{max_i}} \in \sum_n, \quad i = [1, \ldots, m] \tag{8c}$$

$$-\left[\alpha_{min_i} - K_i(x)\right] + (V(x)-\gamma)s_{\alpha_{min_i}} \in \sum_n, \quad i = [1, \ldots, m] \tag{8d}$$

$$-(V(x)-\gamma)+(p(x)-\beta)s_1 \in \sum_n \tag{8e}$$

In this context, s_1 denotes SOS polynomials. Equation (8) provides a control law that optimizes the size of the region under the SF by maximizing the parameter β, resulting in an optimized LF, $V(x)$ and its associated value γ. The inclusion of the SF does not alter Equation (6) but adds an additional constraint, leading to improved results compared to using Equation (6) alone. However, if the system has a nonsymmetric ROA, Equation (8) may only approximate part of the actual ROA while synthesizing the control law. Additionally, Equation (8) does not address state constraints. Therefore, the next section introduces a new method that simultaneously tackles both state constraints and nonsymmetric ROA issues.

3 | Designing Control Law With Enhanced ROA for the Constraint System

3.1 | Proposed Method for Incorporating State Constraints in Control System Design With ROA Estimation

The aim of this research is to develop a control system that improves stability and robustness by increasing the system's ROA in scenarios involving actuator saturation and state constraints.

The current approach, based on SOS optimization as described in Equation (8), does not address state constraints. Therefore, we propose to use Theorem 1 *(S-procedure)* to incorporate state constraints into Equation (8). To the best of our knowledge, Theorem 1 has not been explicitly used in the existing literature to incorporate state constraints into control system design with an ROA within the SOS optimization framework. Thus, we aim to refine the control system design by incorporating state constraints into Equation (8) with the help of Theorem 1.

Let us consider that the constraints represented by Equation (1c) define the below-desired safe set,

$$H_i = \{x \in \mathbb{R}^n : h_i(x) \leq \lambda_i\}, \quad [i = 1, \ldots, t]. \tag{9}$$

The objective is to ensure that the estimated ROA adheres to the state constraint specified in Equation (9). This means that the region under the estimated ROA should be a subset of the feasible region defined by the state constraints. Using set notation, this requirement can be expressed as follows:

$$\begin{aligned} &\Omega_\gamma = \{x \in \mathbb{R}^n : V(x) \leq \gamma\} \subseteq H_i = \{x \in \mathbb{R}^n : h_i(x) \leq \lambda_i\}, \\ &\qquad\qquad [i = 1, \ldots, t] \\ &\Omega_\gamma = \{x \in \mathbb{R}^n : V(x) - \gamma \leq 0\} \subseteq H_i = \{x \in \mathbb{R}^n : h_i(x) - \lambda_i \leq 0\}, \\ &\qquad\qquad [i = 1, \ldots, t] \end{aligned} \tag{10}$$

To make Equation (10) computationally feasible, we need to convert it from a set equation to an algebraic equation. According to Theorem 1, $\Omega_\gamma \subseteq H_i$, if there exist $s_{sc_i} \in \sum_n$, and $V(x)$ and $h_i(x)$ satisfy the below condition,

$$-(h_i(x) - \lambda_i) + (V(x) - \gamma)s_{sc_i} \in \sum_n \quad [i = 1, \ldots, t]. \tag{11}$$

The equation below resulting from the integration of Equation (8) with Equation (11) furnishes a control law that guarantees an ROA compliant with both actuator saturation and state constraints.

$$\beta^* = \max_{\substack{s_0, s_k, s_{\alpha_{max_1}}, s_{\alpha_{min_1}}, \ldots, s_{\alpha_{max_m}}, s_{\alpha_{min_m}}, s_{sc_1}, \ldots, s_{sc_t}, s_1 \in \sum_n, \\ V(x), K_i(x) \in R_n, \gamma \in \mathbb{R}^+}} \beta$$

$$\text{Subject to}: \quad V(x) - l_1 \in \sum_n \tag{12a}$$

$$-\left[\frac{\partial V}{\partial x}\left(f(x) + \sum_{i=1}^{m} g_i(x)K_i(x)\right)s_k + l_2\right] + (V(x) - \gamma)s_0 \in \sum_n \tag{12b}$$

$$-\left[K_i(x) - \alpha_{max_i}\right] + (V(x) - \gamma)s_{\alpha_{max_i}} \in \sum_n, i = [1, \ldots, m] \tag{12c}$$

$$-\left[\alpha_{min_i} - K_i(x)\right] + (V(x) - \gamma)s_{\alpha_{min_i}} \in \sum_n, i = [1, \ldots, m] \tag{12d}$$

$$-(h_i(x) - \lambda_i) + (V(x) - \gamma)s_{sc_i} \in \sum_n \quad [i = 1, \ldots, t] \tag{12e}$$

$$-(V(x) - \gamma) + (p(x) - \beta)s_1 \in \sum_n \tag{12f}$$

3.2 | Proposed Method for Control System Design With Enhanced ROA Estimation

The single SF-based approach, represented by Equation (8), is a common method for designing control laws while estimating the ROA. However, this methodology's robustness is subject to certain caveats. It is prone to premature convergence or numerical infeasibility, which are particularly prevalent in systems with asymmetric ROAs. Another significant limitation, the single SF's geometric center at the origin poses a significant limitation for ROA estimation, especially for asymmetric ROAs. This limitation will be further elucidated with the aid of Figure 1.

The conceptual interpretation of Equation (8e) is that the region delineated by the SF ($p(x) \leq \beta$) must be contained within the boundaries established by the LF, $V(x)$, and its threshold value γ ($V(x) \leq \gamma$). The solution algorithm of Equation (8) incrementally increases the value of β, such as β_1 and β_2 in Figure 1, during each iteration to encompass additional regions under the SF while synthesizing the control law. This process produces a new LF, like $V(x)_1$ and $V(x)_2$ in Figure 1, which encloses the SF's area. As iterations continue, the SF may approach the unstable region along certain directions within the asymmetric ROA. This is evident in Figure 1, where the SF touches the unstable region along the positive y-axis. Consequently, further adjustments to the parameter β fail to generate new controllers with a corresponding LF that encompasses additional stable regions. This failure occurs because the controller reaches its limiting value in this specific direction, with the LF from the previous iteration nearly intersecting the unstable region.

Although the LF effectively defines the ultimate region in a specific direction (as shown in the positive y direction in the figure), there remains an opportunity to synthesize new controllers capable of encompassing stable regions in other directions, such as the x-axis and negative y-axis in Figure 1, while retaining previously captured regions. However, this potential is not realized within the constraints of the single SF-based approach as represented by Equation (8e).

To overcome these limitations, we propose utilizing the property of Theorem 2 (Union Theorem). Theorem 2 states that under certain conditions, it is possible to obtain a superset defined by a

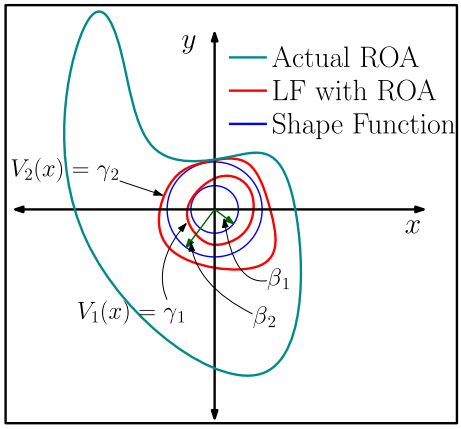


FIGURE 1 | Limitation of single SF based approach.

function V that can encompass a union of regions defined by different functions p_i. Furthermore, the conditional equation represented by Equations (3) of Theorem 2 is similar to Equation (8e). Essentially, Theorem 2 allows the use of multiple SFs to generate an LF that can encircle all the regions under the SFs. Therefore, we propose using multiple SFs instead of a single one in Equation (8) to estimate the ROA while synthesizing the control law. This approach can estimate an enlarged ROA, as explained below using Figure 1.

Consider two SFs along the positive and negative x-axis and two more along the positive and negative y-axis. The solution algorithm will increase the value of all β for every SF, leading to the synthesis of a new controller and an accompanying LF that encompasses additional regions by enclosing all the regions under the SFs. If, after some iterations, the SF along the positive y-axis touches the unstable region, the algorithm will cease operation. However, by this point, the generated LF will have already produced an expanded ROA. This expansion results from the simultaneous increase of all SFs in various directions, thereby broadening the ROA in multiple ways.

Therefore, by incorporating multiple SFs into SOS optimization, it becomes possible to design a control law with an enhanced ROA. If we introduce multiple SFs into Equation (12), then it will synthesize a control law with enhanced ROA for the system under actuator saturation and state constraints. Hence, by introducing and incorporating multiple SFs into Equation (12), we can express the modified form of Equation (12) as follows:

$$\text{Subject to}: \quad V(x) - l_1 \in \sum_n \tag{13a}$$

$$-\left[\frac{\partial V}{\partial x}\left(f(x) + \sum_{i=1}^{m} g_i(x)K_i(x)\right)s_k + l_2\right] + (V(x)-\gamma)s_0 \in \sum_n \tag{13b}$$

$$-\left[K_1(x) - \alpha_{max_1}\right] + (V(x)-\gamma)s_{\alpha_{max_1}} \in \sum_n \tag{13c}$$

$$-\left[\alpha_{min_1} - K_1(x)\right] + (V(x)-\gamma)s_{\alpha_{min_1}} \in \sum_n \tag{13d}$$

$$\vdots$$

$$-\left[K_m(x) - \alpha_{max_m}\right] + (V(x)-\gamma)s_{\alpha_{max_m}} \in \sum_n \tag{13e}$$

$$-\left[\alpha_{min_m} - K_m(x)\right] + (V(x)-\gamma)s_{\alpha_{min_m}} \in \sum_n \tag{13f}$$

$$-(h_1(x) - \lambda_1) + (V(x)-\gamma)s_{sc_1} \in \sum_n \tag{13g}$$

$$\vdots$$

$$-(h_t(x) - \lambda_t) + (V(x)-\gamma)s_{sc_t} \in \sum_n \tag{13h}$$

$$-(V(x)-\gamma) + (p_1(x) - \beta_1)s_1 \in \sum_n \tag{13i}$$

$$\vdots$$

$$-(V(x)-\gamma) + (p_n(x) - \beta_n)s_n \in \sum_n . \tag{13j}$$

Explanation 1. Equation (13a) ensures the positivity of V, a key requirement for a LF.

Explanation 2. Equation (13b) indicates that the area bounded by $V \le \gamma$ must consistently reside within the stable region, as dictated by $\dot{V} \le -l_2$.

Explanation 3. Equations (13c–13f) yield a suitable $K_i(x)$ while accommodating saturation, ensuring that the defined level set $\Omega_\gamma = \{x \in \mathbb{R}^n : V(x) \le \gamma\} \subseteq \Omega_{K_i} = \{x \in \mathbb{R}^n : -\alpha_{min_i} \le K_i(x) \le \alpha_{max_i}\}, \quad [i = 1, \ldots, m]$.

Explanation 4. Equations (13g–13h) enforce the confinement of the LF-generated area within specified state constraints, guaranteeing that the level set $\Omega_\gamma = \{x \in \mathbb{R}^n : V(x) \le \gamma\} \subseteq H_i = \{x \in \mathbb{R}^n : h_i(x) \le \lambda_i\}, \quad [i = 1, \ldots, t]$.

Explanation 5. Equations (13i–13j) ensure that the domains enclosed by $p_1(x) \le \beta_1, \ldots, p_n(x) \le \beta_1$ consistently reside within the area bounded by $V(x) \le \gamma$.

The following discussion regarding Equation (13) is crucial for developing an advanced algorithm to solve it.

Assertion 1. *Equation (13) is highly complex because the decision polynomials V, K_i, and all SOS functions s_i do not enter the constraints linearly, making direct solutions infeasible. Consequently, an iterative algorithm known as the $V - s$ algorithm, detailed in [32], is employed to solve Equation (13). However, the original algorithm in [32] is not directly applicable to Equation (13) as it is designed for a single SF centered at the origin without state constraints. In contrast, our proposed method, represented by Equation (13), allows for multiple origin-centered, shifted-centered (non-origin), or a combination of both types of SFs, and includes state constraints. Therefore, we have developed a new algorithm based on the classical $V - s$ iteration algorithm to handle multiple SFs with various centers and state constraints, yielding better results through multiple rounds of iterations.*

The algorithm decomposes the problem (Equation (13)) into three steps: the first is the γ-step, which has two substeps—one synthesizes a control law, and the other maximizes γ; the second is the β-step, which maximizes β; and the final step is the V-step, which finds a new V that satisfies all constraints. To use the algorithm, we need an initial V_i and must define the degrees of V, K_i, and all SOS functions s_i. Additionally, we must define and select the SFs and judiciously calculate their centers' positions. The subsequent remarks and sections address these issues in detail before presenting the algorithm.

Assertion 2. *The γ-step emphasizes the necessity of an initial $V(x)$ to commence the algorithm. This starting $V(x)$ can be obtained through various methods, such as solving the Riccati equation derived from the linearized dynamics (LD) of the system, and it is referred to as the control Lyapunov function (CLF), as noted in [43] and [44]. Additionally, at each iteration, a fresh initial $V(x)$ is required, which is obtained from the preceding iteration since the algorithm furnishes a new $V(x)$ after each cycle. It is indicated that following the V-step, the algorithm formulates an LF for the closed-loop dynamics. Nevertheless, this $V(x)$ serves as a CLF for subsequent iterations when utilized as the initial $V(x)$ for that iteration.*

Assertion 3. *Multiple rounds of iteration consistently yield superior results compared to a single round, as recommended by [45], especially for estimating the ROA. Similarly, while synthesizing the control law, we can apply the same concept to obtain a better*

estimate of the ROA. After running the algorithm with a single or multiple SFs for several iterations, the algorithm eventually stops if a particular SF intersects the unstable area or if it is unable to procure a new LF that satisfies the constraint under it. At this point, we need to initiate another round of iteration to enhance the ROA estimation during the control law design. For this new round of iteration, we can either use new or the same SFs from the previous round, but with different centers, along with an initial $V(x)$. We need to utilize the $V(x)$ generated from the previous round as an initial $V(x)$ for this new round of iteration. Employing multiple rounds of iteration with multiple SFs usually leads to better results. However, it's important to note that despite additional rounds of iteration, the controller with LF may become saturated and fail to capture more area if the algorithm does not obtain new SOS functions, denoted by $s(x)$, in each iteration to satisfy the constraints Equations (13b–13j).

Assertion 4. *Lemma 1 states that the system achieves asymptotic stability if it meets the constraints outlined by Equation (2). We employed Lemma 1 in designing the control system while estimating the ROA. Consequently, our synthesized control system will asymptotically stabilize the system.*

Assertion 5. *Equation (13) is formulated through the principles outlined in Lemma 1, which is centered on identifying an LF conducive to system stability. The primary objective of the iterative solution algorithm is the discovery of this LF. Throughout this iterative process, various free variables are necessitated, including all SOS polynomials denoted as s. Furthermore, the algorithm treats the feedback control law, denoted as $K(x)$, as another free variable. Consequently, while the algorithm aims to discover the LF, it also synthesizes the feedback control law $K(x)$ as a byproduct, thereby ensuring system stability.*

Assertion 6. *The following conditions must be met to obtain a viable solution for Equation (13), [32, 45] and [23],*

$$\deg(V) \geq \deg(l_1) \text{ and } \deg(Vs_0) \geq \deg(l_2) \tag{14a}$$

$$\deg(s_0) \geq \max\left[\deg(fs_k); \deg(g_i K_i s_k)\right] - 1, \quad [i = 1, \ldots, m] \tag{14b}$$

$$\deg(Vs_{\alpha_{max_i}}) \geq \deg(K_i), \quad [i = 1, \ldots, m] \tag{14c}$$

$$\deg(Vs_{\alpha_{min_i}}) \geq \deg(K_i), \quad [i = 1, \ldots, m] \tag{14d}$$

$$\deg(Vs_{sc_i}) \geq \deg(h_i), \quad [i = 1, \ldots, t] \tag{14e}$$

$$\deg(p_i s_i) \geq \deg(V), \quad [i = 1, \ldots, n] \tag{14f}$$

3.3 | Selection of Shape Function

The SF plays a critical role in the design of the control law, influencing both the shape and size of the generated LF and impacting the computational time required for simulations. Different SFs result in different control systems with varied ROA. According to the definition, any positive definite function can serve as an SF. For their simplicity and effectiveness, the following quadratic SFs are commonly employed.

Origin-Centered Shape Function: A shape function is considered origin-centered when it is centered at the origin of the coordinate system.

Shifted Shape Function: A shape function is referred to as shifted when its center is located at a point other than the origin of the coordinate system [45].

A general expression for both types of SFs is given by,

$$p(x) = (x - x^*)' N (x - x^*) \tag{15}$$

In the above general expression, x^* represents the center of the SF. When $x^* = 0$, the function $p(x)$ is an origin-centered SF. Otherwise, it is a shifted SF. The matrix N, known as the shape matrix, dictates the nature of the SF, determining whether it will be circular, elliptical, or another form. The choice of N is crucial as it influences the estimation. Selecting N is problem-specific. In the absence of prior knowledge about N, the identity matrix $N = I$ is often used as a default choice. Alternatively, N can be selected by randomly generating multiple candidates, designing the control systems, and evaluating their respective ROAs to identify the best-performing N. For this study, we have utilized both origin-centered and shifted SFs. In the case of the shifted SF, once N is selected, positioning the center x^* appropriately is essential for obtaining an optimal estimation.

3.3.1 | Center Selection

Selecting the center of the SF is crucial for enhanced ROA estimation and control law design. According to Explanation 4, the center can be anywhere within $V \leq \gamma$. Ignatyev et al. [45] proposed a method for selecting the center to estimate the ROA, which we intend to use in this study. The method determines the center by finding the intersection of a bounded LF ($V = \gamma^*$) and a line with a specific inclination passing through the origin. Although [40] illustrates the method mathematically for the 2-D case, it can be extended to higher dimensions. The relevant equations are,

$$V(x_1, x_2) = \gamma^*, \tag{16a}$$

$$x_2 = \tan\theta x_1. \tag{16b}$$

Here, θ represents the line's inclination, with each SF having a predetermined fixed θ for its shifting center. This θ needs to be established prior to calculating the intersection point. Denote the intersection point of equations (16a) and (16b) as (x^*_{1I}, x^*_{2I}). The distance from the center to this point is $\rho_\alpha = \sqrt{x^{*2}_{1I} + x^{*2}_{2I}}$, defining the shifting center as,

$$x^*_1 = \sigma\rho_\alpha \cos\theta, \tag{17a}$$

$$x^*_2 = \sigma\rho_\alpha \sin\theta. \tag{17b}$$

The value of the shifting factor, σ, lies in the range (0,1). Selecting extremes like 0 or 1 is illogical. For $\sigma = 0$, the SF is centered at the origin, eliminating the need for center calculation. Choosing $\sigma = 1$ places the center on the LF, hindering growth to avoid violating constraints (Equations (13i–13j)). Opting for values near 0 or 1 leads to inefficiency: near 0 prolongs growth iterations, while larger values bring the center close to the boundary, hampering proper expansion. A judicious choice of σ is necessary to achieve a larger ROA with fewer iterations while synthesizing the control law. Further elaboration can be found in [40] and [45].

In the context of 3-D, it's necessary to examine the equations below that describe LF and a 3D straight line.

$$V(x_1, x_2, x_3) = \gamma^*, \tag{18}$$

$$x_1 = r\cos\theta\cos\psi, \tag{19a}$$

$$x_2 = r\cos\theta\sin\psi, \tag{19b}$$

$$x_3 = r\sin\theta. \tag{19c}$$

In this context, θ and ψ represent the predetermined elevation and azimuth angles of the line. The process for finding the shifting center (x_1^*, x_2^*, x_3^*) in the 3-D case is analogous to the 2-D case, but requires selecting the inclination angle of the straight line first. In 2-D, angle selection is relatively straightforward and can be manually adjusted based on the expansion characteristics of the LF. However, in higher dimensions, selecting angles is more complex. Biswas et al. [40] proposes a method for angle selection in the 3-D case, which can be extended to higher dimensions. This method involves using a unit cuboid centered at the system's EP. Lines are drawn from the cuboid's center to its vertices and midpoints on its surfaces. The azimuth and elevation angles for these points are then calculated based on the unit length of the cuboid. A total of fourteen angles are determined: eight from the vertices and six from the midpoints. The angles (in degrees) derived from the unit cuboid for center calculation are outlined below.

$$\begin{bmatrix}\psi\\ \theta\end{bmatrix} = \begin{bmatrix}-45 & -45 & -135 & -135 & 45 & 45 & 135 & 135 & 0 & 90 & 180 & -90 & 0 & 0\\ -35 & 35 & -35 & 35 & -35 & 35 & -35 & 35 & 0 & 0 & 0 & 0 & 90 & -90\end{bmatrix}^\circ. \tag{20}$$

The initial eight angles and the subsequent six angles referenced in Equation (20) pertain to the vertices and midpoints of the cuboid's surfaces. For the scenario in n dimensions, it's essential to utilize an equation tailored to the dimensionality of the space. To accomplish this, we can leverage the n-dimensional straight line with polar coordinates detailed in [46].

As stated in Assertion 2, the first round of iteration requires constructing an initial CLF using the Riccati equation and estimating an initial ROA. According to Explanation 5, the centers of the SFs used in any round of iteration must lie within the related estimated initial ROA; both origin-centered and shifted SFs are admissible. As discussed previously, shifted SFs are typically centered near the ROA boundary [as per Equation (17)]. The size of this initial ROA, obtained from the LD, is system-dependent. If the initial ROA is very small, placing shifted SFs near the boundary may severely limit its allowable growth due to insufficient available space, causing the algorithm to terminate prematurely and preventing further iterations. In such cases, origin-centered SFs are preferable in the 1st round of iteration because they offer sufficient scope for expansion, thereby enabling the algorithm to progress smoothly toward subsequent iteration rounds.

The detailed algorithm underlying our proposed approach is provided in the Appendix A for reference. We now proceed to evaluate its effectiveness through simulation studies.

4 | Simulation Results

The simulation results have been generated using the SOSOPT toolbox, as referenced in [18]. In order to demonstrate the effectiveness of our proposed methodology, we have considered three distinct examples, consisting of two 2-D cases and one 3-D case. Across these examples, the first instance involved no constraints, while the second incorporated control input saturation, and the third integrated both control input saturation and state constraints. It is noted that initializing the simulation requires an initial CLF, obtained by solving the Riccati equation based on the LD. In each subsequent round of iteration, the final LF from the preceding iteration round is utilized as the initial CLF for the next round of iteration, as discussed in Assertion 2. Additionally, it is essential to specify the degree of all SOS polynomials denoted by s_i, along with the LF, $V(x)$, and feedback control system $K(x)$. Any degree for these variables can be selected as per the requirement while satisfying Equation (14). Moreover, the polynomials l_1 and l_2 must be defined. Furthermore, fixing the SFs and inclination angle of the straight line to calculate the center of the corresponding SF at the outset of each iteration round is imperative. For this study, the shape matrix N and the center selection angle in 2D are manual, while Equation (20) guides center selection for the 3D case. We also maintained a consistent value of the shape factor $\sigma = 0.8$ across all three examples.

Example 1. Consider the following system [32],

$$\begin{aligned}\dot{x}_1 &= u\\ \dot{x}_2 &= -x_1 + \frac{1}{6}x_1^3 - u.\end{aligned} \tag{21}$$

The system operates without any constraints on states and control input. To obtain results for this system, we consider Equation (13) exclusively, disregarding Equations (13c–13h). Consequently, Equations (13c–13h) are not required in the iteration algorithm; however, rest of the algorithm remains unchanged. We specify $\deg(V) \to [\min(2), \max(2)]$, $\deg(s_0) \to [\min(2), \max(4)]$, $\deg(s_k) \to [\min(0), \max(2)]$, $\deg(s_{1,\ldots,n}) \to [\min(0), \max(4)]$, $deg(K) \to [\min(1), \max(1)]$, $l_1 = 10^{-6}(x_1^2 + x_2^2)$ and $l_2 = 10^{-6}(x_1^4 + x_2^4)$ for obtaining results. The simulation consists of two rounds of iteration, each with two SFs. The parameters from Table 1 are used for this example. The resulting linear feedback control law and LF derived from this simulation can be found in Appendix B.

The results are depicted in Figure 2. Figure 2a,b illustrate the outcomes of the first and second iteration rounds. Two origin-centered SFs have been utilized in the first round, while two shifted SFs of the same type have been employed in the second round, expanding in both x_1 and x_2 directions. The increase in the related SF areas in each iteration corresponds to the growth of the ROA, as observed in Figure 2a,b. A comparison between the estimated ROA from both rounds in Figure 2c highlights the significantly larger area captured in the second iteration round. To demonstrate the effectiveness of the designed control system, the system is propagated from eight different initial points. The trajectories and associated control input demand are shown in Figure 2d,e. All trajectories reached the origin (Figure 2d), although initially, the control input demand is high due to the absence of constraints on the control input (Figure 2e). However, this demand decreases rapidly, and all trajectories asymptotically reach the EP. Monte Carlo simulation validated the estimated ROA for the closed-loop system, as shown in Figure 2f, where the initial points for stable and unstable trajectories are displayed. While our estimated ROA remains within the stable

TABLE 1 | Simulation Parameters for Example 1.

Round	Initial V_0	Angle ($\theta°$)	Shape matrix (N)
1st (1R)	V_0 = From LD	Not applicable as $x^* = 0$	$N1_{1R} = [0.5\ 0; 0\ 1]$
		Not applicable as $x^* = 0$	$N2_{1R} = [1\ 0; 0\ 0.5]$
2nd (2R)	$V_0 = V_{1R}$	$\theta 1_{2R} = 135$	$N1_{2R} = [0.5\ 0; 0\ 1]$
		$\theta 2_{2R} = 325$	$N2_{2R} = [0.5\ 0; 0\ 1]$

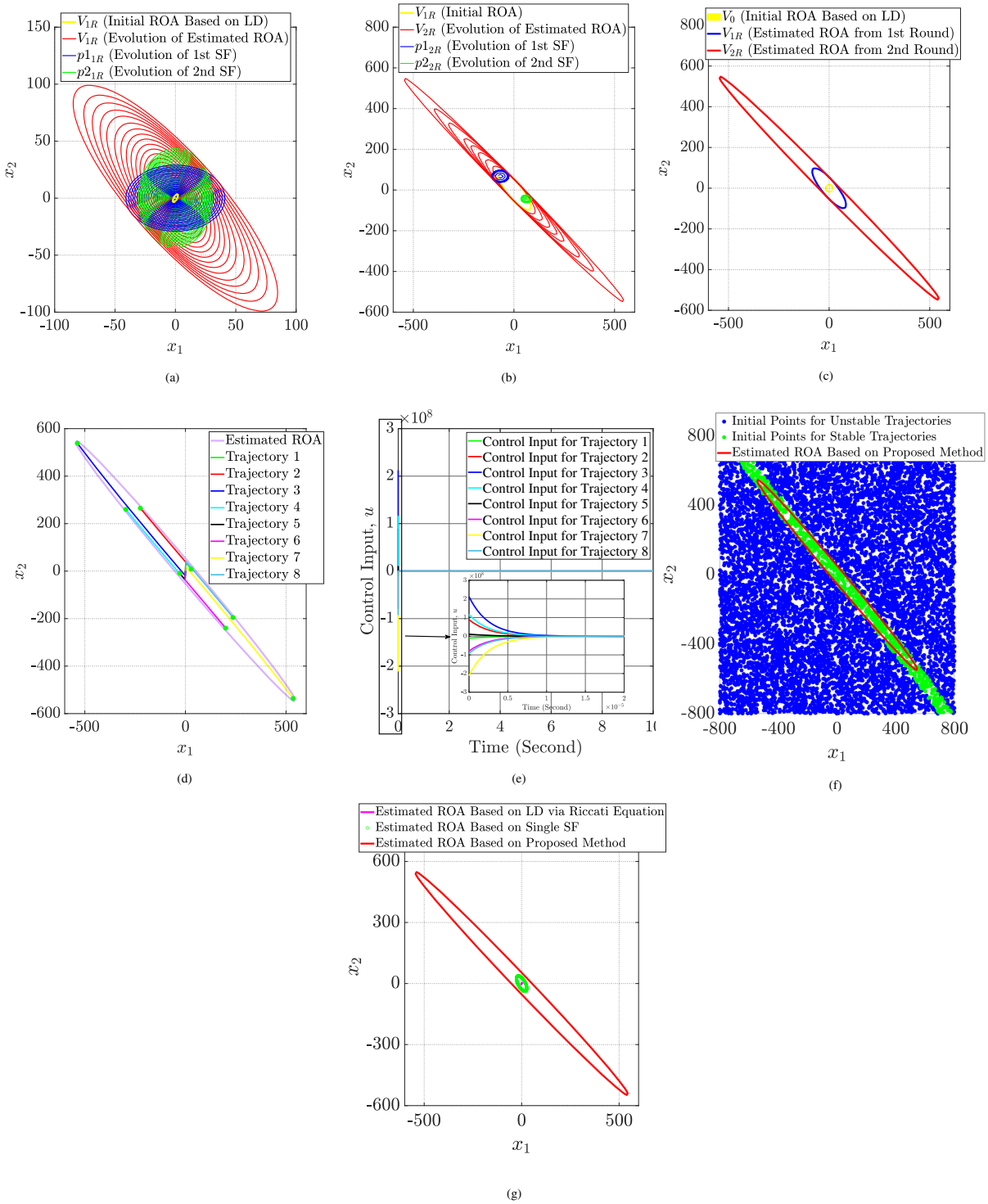


FIGURE 2 | Control System Design with ROA Estimation for Example 1. (a) ROA Estimation from the 1st Round, (b) ROA Estimation from the 2nd Round, (c) Comparison of Estimated ROAs Across all the Rounds, (d) Trajectories of the System Under the Designed Control Law, (e) Control Input Responses for the Related Trajectories, (f) Monte Carlo Simulation with Estimated ROA, (g) Comparison of Estimated ROAs Across Different Method.

region, some stable areas remain uncovered, suggesting further iteration rounds with appropriate SFs and centers could expand coverage. Furthermore, Figure 2g illustrates a comparison of the ROA estimates obtained during control law design using our proposed method against those derived from existing techniques—specifically, methods based on the Riccati equation and a single SF within the SOS framework, revealing the superiority of our proposed method. The simulation required 30 iterations in the first round and 9 iterations in the second round. Although the estimation process has been halted, it could be continued to obtain further stable areas under the estimated ROA.

Example 2. Consider the following nonlinear system [33],

$$\dot{x}_1 = -x_1 + x_1x_2 + x_2\text{sat}(u)$$
$$\dot{x}_2 = x_1 + 2x_2 + x_1^2 + x_1^2x_2 + \text{sat}(u)$$
$$\text{sat}(u) = \begin{cases} 3 & \text{if } u > 3 \\ u & \text{if } -3 \le u \le 3 \\ -3 & \text{if } u < -3 \end{cases} \quad (22)$$

This system experiences actuator saturation. Equations (13g) and (13h) are unnecessary for designing the control system with an ROA for this setup. However, all other equations under Equation (13) remain essential. Similarly, in the algorithm, Equations (13g) to (13h) are not required. We have specified the degrees as follows: $\deg(V) \to [\min(2), \max(4)]$, $\deg(s_0) \to [\min(2), \max(6)]$, $\deg(s_k) \to [\min(0), \max(2)]$, $\deg(s_{\alpha_{\max_i}, \alpha_{\min_i}})_{i=1,\ldots,m} \to [\min(0), \max(2)]$, $\deg(s_{1,\ldots,n}) \to [\min(0), \max(6)]$, and $\deg(K) \to [\min(1), \max(3)]$. Additionally, $l_1 = l_2 = 10^{-6}(x^Tx)$. We have devised a third-degree control system while accounting for saturation. The designed control system and associated LF are provided in Appendix B.

For this example's simulation, we conducted four rounds of iteration. In the initial round, two origin-centered SFs were used, while five shifted SFs were employed in subsequent rounds. Table 2 provides all relevant parameters. Results are shown in Figure 3, with Figure 3a–d displaying the estimated ROA from rounds 1 to 4 during control law synthesis. In the first round, two SFs are chosen, one expanding in the x_1 direction and the other in the x_2 direction, covering stable areas in both directions. In subsequent rounds, different sets of five SFs are selected to expand the ROA while preserving the previous round's ROA. These figures demonstrate the evolving ROA with increasing SF areas in each iteration. A comparison in Figure 3e reveals that each round covers more area than the previous one. To illustrate the effectiveness of the designed control system, we propagate the system from three different initial points, and the related trajectories and control input demands are shown in Figure 3f,g. Notably, control input demand remained within saturation limits, as observed in Figure 3g. However, these three control input demands do not consistently ensure adherence to the saturation limit in all cases. To verify controller values within the estimated ROA, 10^5 points have been selected, and the control input demand is evaluated at every point, confirming adherence to limits ($-3 \le u \le 3$) as shown in Figure 3h. Monte Carlo simulation identified initial points for stable and unstable trajectories, depicted in Figure 3i, indicating that the estimated ROA remains within stable areas and suggesting potential expansion with further iterations. Additionally, a comparison among estimated ROAs based on the Riccati equation, a single SF, and our proposed method is shown in Figure 3j, demonstrating the superiority of our approach. It is worth noting that the estimation process concluded after the fourth round of iteration, but further iterations could enhance ROA estimations while refining the control system. The simulation required 14, 70, 37, and 22 iterations in the first to fourth rounds, respectively.

Example 3. Consider the following nonlinear system,

$$\dot{x}_1 = x_1^2 + x_2$$
$$\dot{x}_2 = x_1 + (1 + x_2^2)x_3$$
$$\dot{x}_3 = \text{sat}(u)$$
$$\text{sat}(u) = \begin{cases} 5, & \text{if } u \ge 5 \\ u, & \text{if } -5 \le u \le 5 \\ -5, & \text{if } u \le -5 \end{cases}$$

Safe region defined by state constraint: $h(x) = x_2 \le 0.5$ (23)

TABLE 2 | Simulation Parameters for Example 2.

Round	Initial V_0	Angle (θ°)	Shape matrix (N)
1st (1R)	V_0 = From LD	Not applicable as $x^* = 0$	$N1_{1R} = [0.5\ 0; 0\ 1]$
		Not applicable as $x^* = 0$	$N2_{1R} = [1\ 0; 0\ 0.5]$
2nd (2R)	$V_0 = V_{1R}$	$\theta 1_{2R} = 45$	$N1_{2R} = [0.5\ 0; 0\ 1]$
		$\theta 2_{2R} = 130$	$N2_{2R} = [0.5\ 0; 0\ 1]$
		$\theta 3_{2R} = 225$	$N3_{2R} = [0.5\ 0; 0\ 1]$
		$\theta 4_{2R} = 190$	$N4_{2R} = [0.5\ 0; 0\ 1]$
		$\theta 5_{2R} = 335$	$N5_{2R} = [0.5\ 0; 0\ 1]$
3rd (3R)	$V_0 = V_{2R}$	$\theta 1_{3R}$ =45	$N1_{3R} = [0.5\ 0; 0\ 1]$
		$\theta 2_{3R} = 130$	$N2_{3R} = [0.5\ 0; 0\ 1]$
		$\theta 3_{3R} = 225$	$N3_{3R} = [0.5\ 0; 0\ 1]$
		$\theta 4_{3R} = 195$	$N4_{3R} = [0.3\ 0; 0\ 5]$
		$\theta 5_{3R} = 333$	$N5_{3R} = [0.5\ 0; 0\ 2]$
4th (4R)	$V_0 = V_{3R}$	Same as 3R except $\theta 4_{4R} = 190$	Same as 3R

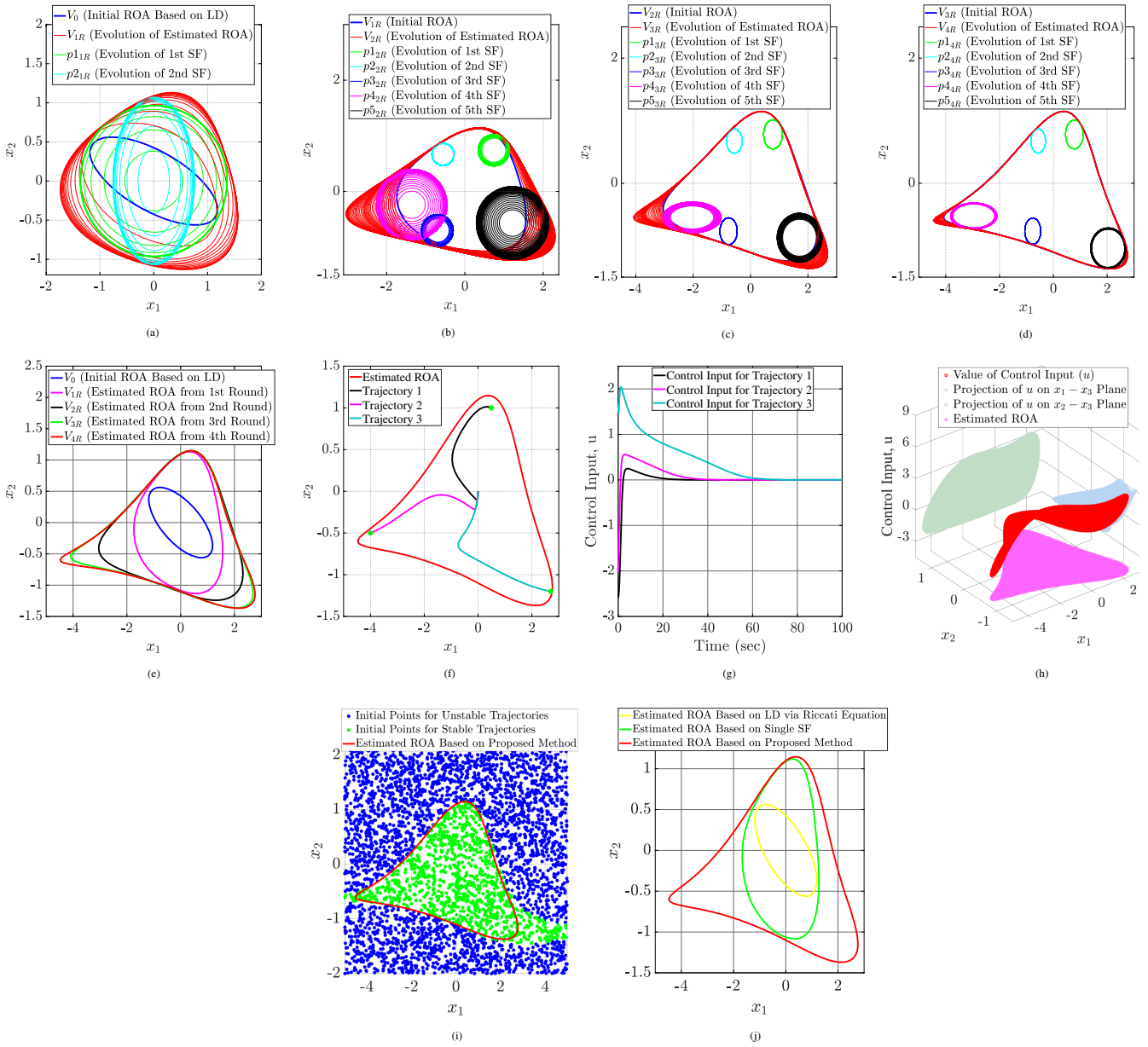


FIGURE 3 | Control System Design with ROA Estimation for Example 2. (a) ROA Estimation from the 1st Round, (b) ROA Estimation from the 2nd Round, (c) ROA Estimation from the 3rd Round, (d) ROA Estimation from the 4th Round, (e) Comparison of Estimated ROAs Across all the Rounds, (f) Trajectories of the System Under the Designed Control Law, (g) Control Input Responses for the Related Trajectories, (h) Value of Control Input within Estimated ROA, (i) Monte Carlo Simulation with Estimated ROA, (j) Comparison of Estimated ROAs Across Different Methods.

This system, with 3 states and constraints on both control input and state, requires consideration of all equations in Equation (13) for simulation. For the simulations, we have determined the degrees of various variables as follows: $\deg(V) \to [\min(2), \max(4)]$, $\deg(s_0) \to [\min(2), \max(6)]$, $\deg(s_k) \to [\min(0), \max(2)]$, $\deg(s_{\alpha_{\max_j}, \alpha_{\min_i}})_{i=1,\ldots,m}) \to [\min(0), \max(2)]$, $\deg(s_{1,\ldots,n}) \to [\min(0), \max(2)]$, $\deg(s_{sc_{1,\ldots,t}}) \to [\min(0), \max(2)]$, and $\deg(K) \to [\min(1), \max(3)]$. Additionally, we have initialized the simulation with $l_1 = 10^{-4}(x_1^2 + x_2^2 + x_3^2)$ and $l_2 = 10^{-4}(x_1^4 + x_2^4 + x_3^4)$.

Furthermore, it is essential to determine the SF and calculate its center, denoted by x^* in Equation (15), prior to commencing the simulation. This process must be repeated before each new iteration round due to the changing nature of the SF. Given that the system comprises three states, a three-dimensional SF is necessary. Moreover, a three-dimensional straight line, as described by Equation (19), is utilized to ascertain the center. The selected ellipsoidal SF is given below,

$$p_0 = \begin{bmatrix} x_1 - x_1^* \\ x_2 - x_2^* \\ x_3 - x_3^* \end{bmatrix} N \begin{bmatrix} x_1 - x_1^* \\ x_2 - x_2^* \\ x_3 - x_3^* \end{bmatrix}; N = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{24}$$

We have conducted three rounds of iteration to obtain the system's results. During the first iteration, a single SF centered at the

origin ($x_1^* = x_2^* = x_3^* = 0$) is utilized, with a shape matrix specified by Equation (25). For the second and third iterations, five and nine SFs are employed, respectively. The specific SFs and corresponding angles required to calculate their centers are provided below.

$$p1_{2R} = p2_{2R} = p3_{2R} = p4_{2R} = p5_{2R} = p \tag{25a}$$

$$\begin{bmatrix} \psi \\ \theta \end{bmatrix}_{2R} = \begin{bmatrix} \psi(7) & \psi(11) & \psi(12) & \psi(13) & \psi(14) \\ \theta(7) & \theta(11) & \theta(12) & \theta(13) & \theta(14) \end{bmatrix} \tag{25b}$$

$$p1_{3R} = p2_{3R} = p3_{3R} = p4_{3R} = p5_{3R} = p6_{3R} = p7_{3R} = p8_{3R} = p9_{3R} = p \tag{25c}$$

$$\begin{bmatrix} \psi \\ \theta \end{bmatrix}_{3R} = \begin{bmatrix} \psi(3) & \psi(4) & \psi(7) & \psi(8) & \psi(9) & \psi(11) & \psi(12) & \psi(13) & \psi(14) \\ \theta(3) & \theta(4) & \theta(7) & \theta(8) & \theta(9) & \theta(11) & \theta(12) & \theta(13) & \theta(14) \end{bmatrix} \tag{25d}$$

In Equations (25b) and (25d), the value $[\psi(i); \theta(i)]$ pertains to the i-th column value of Equation (20). Equation (25b) contains five columns that correspond to the center selection angles of the SFs in Equation (25a), whereas nine columns in Equation (25d) are linked to those in Equation (25c). The control system has been designed as a third-degree system that accounts for the actuator saturation and constraints in the state. The resulting final control system and its related LF are outlined in Appendix B.

The findings are depicted in Figure 4. Specifically, Figure 4a–c illustrate the final ROA along with the associated SFs for the first, second, and third rounds, respectively. In Figure 4d, results from all rounds are juxtaposed for comparison, revealing a consistent increase in the estimated ROA with each new iteration round, surpassing the previous estimations. It is important to note that the estimated ROA remains within the prescribed state constraint ($x_2 \leq 0.5$), as illustrated in Figure 4e.

To demonstrate the effectiveness of the designed control law, we have propagated the system using the control law from ten thousand points within the estimated ROA, displaying the resulting trajectories in Figure 4f. Figure 4g exhibits the control input demand for these trajectories. Notably, no constraints, either actuator saturation or state limits, were considered during propagation. However, all trajectories reach the EP without violating state constraints, as evidenced in Figure 4f. Additionally, the control input demand remains within the prescribed saturation limit ($-5 \leq u \leq 5$), as depicted in Figure 4g.

Monte Carlo simulation is employed to obtain initial points for stable and unstable trajectories under the designed control system, as shown in Figure 4h. Notably, the estimated ROA remains within the stable region, validating the results obtained from the proposed method. Furthermore, in Figure 4i, results based on the Riccati equation, single-SF method, and our proposed method are compared, highlighting the superior performance of our proposed method. However, some stable areas remain uncovered, suggesting that further iterations with appropriate shape functions could refine the estimation. The simulation required 36, 81, and 67 iterations in the first, second, and third rounds, respectively.

Example 4. Consider the following real-world dynamics of the Vought F-8 Crusader aircraft, [47].

$$\begin{aligned} \dot{\alpha} &= -0.878\alpha + q - \alpha^2 q - 0.0896\alpha q - 0.019\theta^2 + 0.473\alpha^2 \\ &\quad + 3.813\alpha^3 - 0.216\delta_e \\ \dot{\theta} &= q \\ \dot{q} &= -4.209\alpha - 0.396q - 0.408\alpha^2 - 2.137\alpha^3 - 20.991\delta_e \end{aligned} \tag{26a}$$

$$\text{sat}(\delta_e) = \begin{cases} 20^\circ, & \text{if } \delta_e \geq 20^\circ \\ \delta_e, & \text{if } -20^\circ \leq \delta_e \leq 20^\circ \\ -20^\circ, & \text{if } \delta_e \leq -20^\circ \end{cases} \tag{26b}$$

Safe Region defined by states constraints:

$$\begin{aligned} -10^\circ \leq h_1 = \alpha \leq 30^\circ \\ -10^\circ \leq h_2 = \theta \leq 30^\circ \\ -50^\circ/\text{s} \leq h_3 = q \leq 50^\circ/\text{s} \end{aligned} \tag{26c}$$

Here, α denotes the angle of attack, θ the pitch angle, q the pitch rate, and δ_e the elevator deflection angle (control input). Equation (26a) describes a nonlinear aircraft model capturing the short-period dynamics of the Vought F-8 Crusader. Equations (26b) and (26c) represent, respectively, the saturation limit on the control input and the state constraints.

The degrees of the various variables used in the simulations are determined as follows: $\deg(V) \to [\min(2), \max(4)]$, $\deg(s_0) \to [\min(2), \max(4)]$, $\deg(s_k) \to [\min(0), \max(2)]$, $\deg(s_{\alpha_{\max_i}, \alpha_{\min_i}})_{i=1,\ldots,m}) \to [\min(0), \max(2)]$, $\deg(s_{1,\ldots,n}) \to [\min(0), \max(2)]$, $\deg(s_{sc_{1,\ldots,t}}) \to [\min(0), \max(2)]$, and $\deg(K) \to [\min(1), \max(1)]$. Additionally, we have initialized the simulation with $l_1 = 10^{-4}(\alpha^2 + \theta^2 + q^2)$ and $l_2 = 10^{-4}(\alpha^4 + \theta^4 + q^4)$.

As in the previous example, before initiating the simulation, it is essential to define the SFs and determine their respective centers. Furthermore, as outlined in the proposed methodology, both the SFs and their centers are updated in each iteration round to enhance the estimation of the ROA during the control law design process. In this example, three types of SFs are employed, defined as follows:

$$P_1 = \begin{bmatrix} \alpha \\ \theta \\ q \end{bmatrix}' \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ \theta \\ q \end{bmatrix} \tag{27a}$$

$$P_2 = \begin{bmatrix} \alpha - \alpha^* \\ \theta - \theta^* \\ q - q^* \end{bmatrix}' \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0.5 \end{bmatrix} \begin{bmatrix} \alpha - \alpha^* \\ \theta - \theta^* \\ q - q^* \end{bmatrix}, \tag{27b}$$

$$P_3 = \begin{bmatrix} \alpha - \alpha^* \\ \theta - \theta^* \\ q - q^* \end{bmatrix}' \begin{bmatrix} 2 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0.5 \end{bmatrix} \begin{bmatrix} \alpha - \alpha^* \\ \theta - \theta^* \\ q - q^* \end{bmatrix}, \tag{27c}$$

$$P_4 = \begin{bmatrix} \alpha - \alpha^* \\ \theta - \theta^* \\ q - q^* \end{bmatrix}' \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0.5 \end{bmatrix} \begin{bmatrix} \alpha - \alpha^* \\ \theta - \theta^* \\ q - q^* \end{bmatrix}. \tag{27d}$$

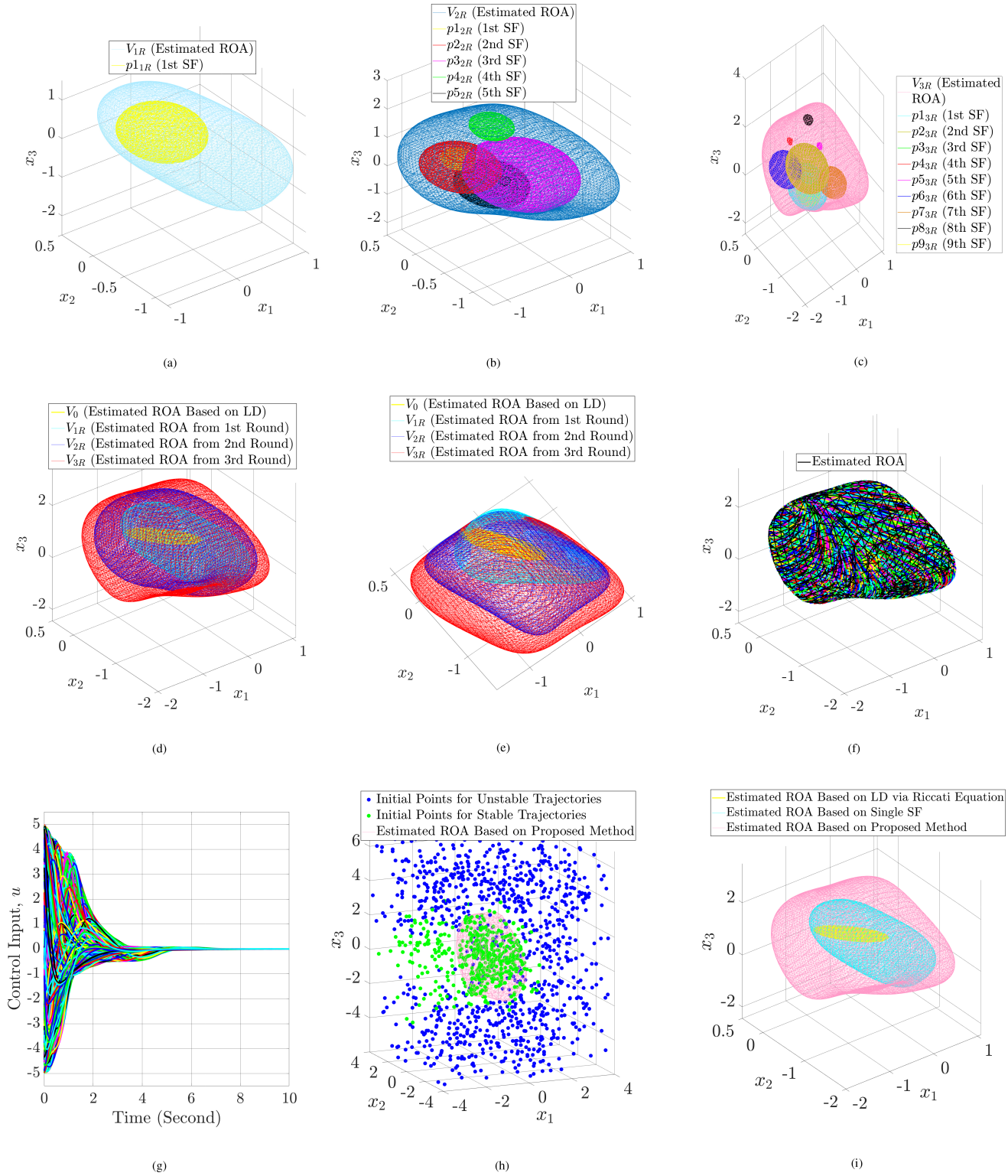


FIGURE 4 | Control System Design with ROA Estimation for Example 3. (a) ROA Estimation from the 1st Round, (b) ROA Estimation from the 2nd Round, (c) ROA Estimation from the 3rd Round, (d) Comparison of Estimated ROAs Across all the Rounds, (e) Projection of the Estimated ROAs on the $x_1 - x_2$ Plane, (f) Trajectories of the System Under the Designed Control Law, (g) Control Input Responses for the Related Trajectories, (h) Monte Carlo Simulation with Estimated ROA, (i) Comparison of Estimated ROAs Across Different Methods.

The simulation is conducted over three iterative rounds, labeled as 1R, 2R, and 3R. In the first round (1R), a single SF (SF) P_1 is used, centered at the origin ($\alpha^* = \theta^* = q^* = 0$), and defined by Equation (27a). In the second (2R) and third (3R) rounds, five shifted SFs are employed in each, with centers distributed in various directions of the state space to progressively enhance ROA coverage. These centers are computed based on the intersection of the bounded LF and a 3D line parameterized by the azimuth

(ψ) and elevation (θ) angles, as described in Equations (18) and (19), similar to the approach used in the previous example. The inclination angles used and the corresponding SFs configurations for each round are detailed below.

$$p_{1_{2R}} = p_{2_{2R}} = p_{5_{2R}} = p_{6_{2R}} = P_2, \tag{28a}$$

$$p_{3_{2R}} = P_3, \tag{28b}$$

$$p_{4_{2R}} = P_4 \tag{28c}$$

$$\begin{bmatrix} \psi \\ \theta \end{bmatrix}_{2R} = \begin{bmatrix} \psi(5) & \psi(6) & \psi(9) & \psi(10) & \psi(13) & \psi(14) \\ \theta(5) & \theta(6) & \theta(9) & \theta(10) & \theta(13) & \theta(14) \end{bmatrix} \tag{29}$$

$$p_{1_{3R}} = p_{2_{3R}} = p_{5_{3R}} = p_{6_{3R}} = P_2, \tag{30a}$$

$$p_{3_{3R}} = P_3, \tag{30b}$$

$$p_{4_{3R}} = P_4, \tag{30c}$$

$$\begin{bmatrix} \psi \\ \theta \end{bmatrix}_{3R} = \begin{bmatrix} \psi(5) & \psi(6) & \psi(9) & \psi(10) & \psi(13) & \psi(14) \\ \theta(5) & \theta(6) & \theta(9) & \theta(10) & \theta(13) & \theta(14) \end{bmatrix} \tag{31}$$

It is noted that the θ in (29) and (31) refers to the inclination angle of the 3D straight line, whereas the θ in (26a) denotes the pitch angle of the F-8 aircraft. The angles $[\psi(i); \theta(i)]$ in Equations (29) and (31) represent the ith column vectors derived from Equation (20), which are used to position the center of each SF. Specifically, Equation (29) contains five vectors corresponding to the five SFs center in Equation (29), while the five vectors in Equation (31) align with the SFs defined in Equation (30).

To demonstrate the effectiveness of the proposed approach, numerical simulations have been conducted under two scenarios:

Case I: System subjected only to control input saturation, governed by Equations (26a) and (26b).

Case II: System subjected to both control input saturation and state constraints, described by Equations (26a), (26b), and (26c).

For Case I, only a subset of the constraints—excluding Equations (13g) and (13h)—is applied during simulation. In contrast, for Case II, the complete set of equations listed under Equation (13) is utilized to account for both input and state constraints. The system parameters used in both cases remain identical and have been defined earlier in the manuscript. The overall control strategy is formulated using a first-order polynomial control law. The final expressions for the control law and the corresponding LF are provided in Appendix B.

The simulation outcomes are presented in Figures 5 and 6 for Case I and Case II, respectively. Specifically, Figure 5a–c illustrate the final estimated ROAs for Case I across three successive rounds of iteration. Similarly, Figure 6a–c show the ROAs for Case II. The evolution of the estimated ROA across rounds is compared in Figures 5d (Case I) and 6d (Case II), which demonstrate progressive enlargement of the ROA at each round of iteration.

In Case II, the projection in Figure 6d confirms that the estimated ROA not only lies strictly within the defined state constraint boundaries but also covers nearly the entire feasible region. In comparison, the projection in Figure 5d (Case I) reveals a significantly larger estimated ROA, as there are no explicit state constraints restricting the expansion.

To validate the robustness of the designed first-order control law, 5000 randomly sampled initial conditions from within the estimated ROA were propagated under the designed controller. The resulting trajectories, shown in Figures 5e and 6e for Case I and Case II, respectively, confirm that all trajectories converge to the EP while remaining entirely within the estimated ROA. The corresponding control input responses, depicted in Figures 5f and 6f, remain strictly within the prescribed input bounds for both cases.

Monte Carlo simulations were also performed for both scenarios to classify initial conditions into stable and unstable regions under the designed control laws. The results are visualized in Figures 5g and 6g, indicating that the estimated ROAs lie entirely within the empirically determined stable regions.

Finally, Figures 5h and 6h present a comparative analysis of the ROAs obtained using the proposed method against those estimated using existing techniques—specifically, approaches based on the Riccati equation and single SF within the SOS framework. The results clearly demonstrate that the proposed method yields a significantly larger ROA during control law design.

The number of iterations required to complete the three control design rounds was as follows: for Case I, 69, 137, and 49 iterations; and for Case II, 5, 46, and 35 iterations, respectively.

The analyses are performed on Intel(R) Core(TM) i7-4770 CPU@3.40GHz, 16.00 RAM.

5 | Conclusion

This work introduces a novel technique by proposing and employing multiple SFs and the S-procedure within the framework of SOS optimization to design a control system with an enlarged ROA, even under the constraints of actuator saturation and state limitations. An advanced algorithm has been subsequently developed to efficiently solve the problem and enhance results. The effectiveness of this approach has been demonstrated through simulation outcomes across four distinct examples. The designed control law with estimated ROA has been verified through Monte Carlo simulations, and the results establish the superiority of our method.

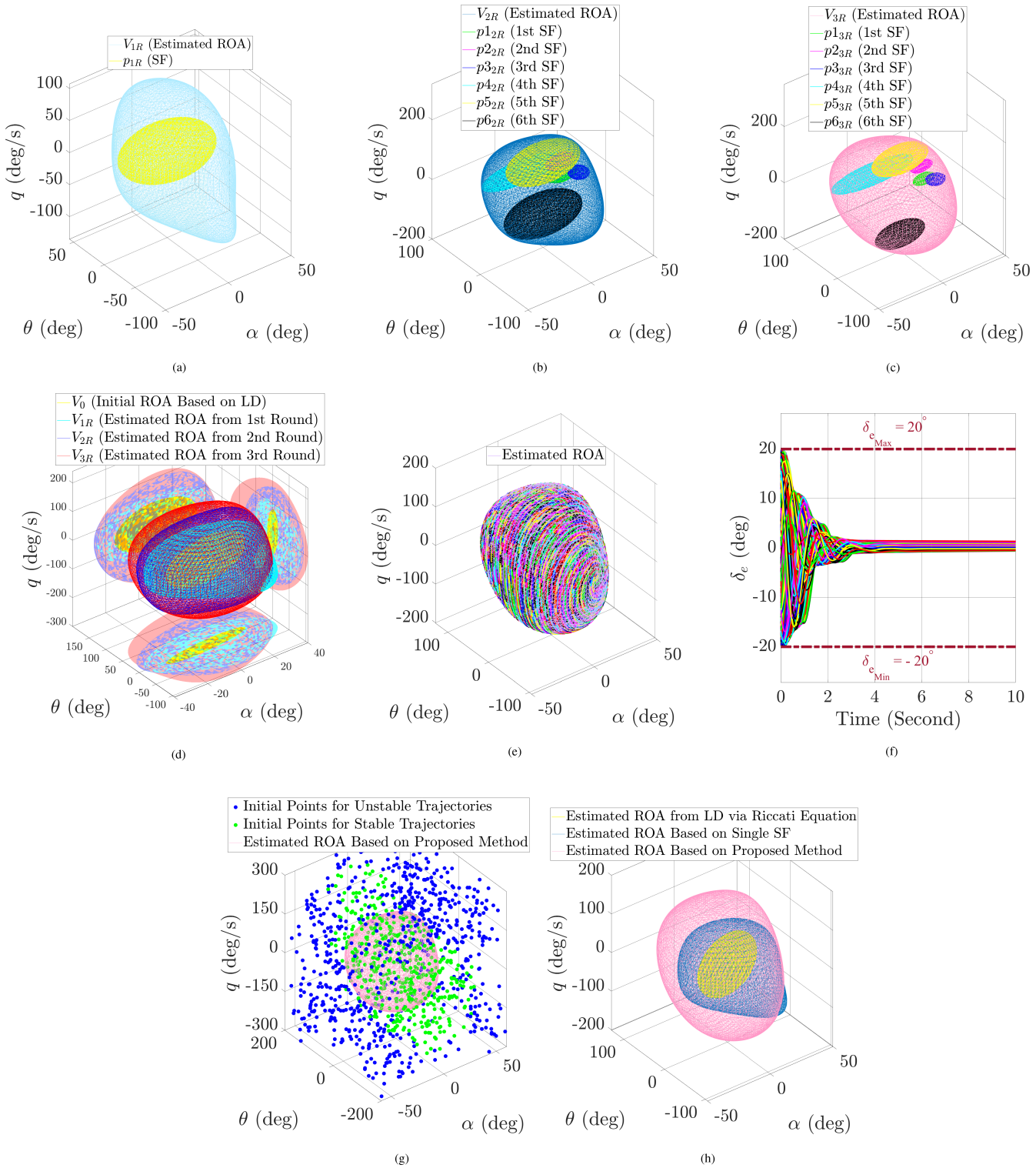


FIGURE 5 | Control System Design with ROA Estimation for Example 4: System with Control Input Saturation Only (Case I). (a) ROA Estimation from the 1st Round, (b) ROA Estimation from the 2nd Round, (c) ROA Estimation from the 3rd Round, (d) Comparison of Estimated ROAs Across all the Rounds, (e) Trajectories of the System Under the Designed Control Law, (f) Control Input Responses for the Related Trajectories, (g) Monte Carlo Simulation with Estimated ROA, (h) Comparison of Estimated ROAs Across Different Methods.

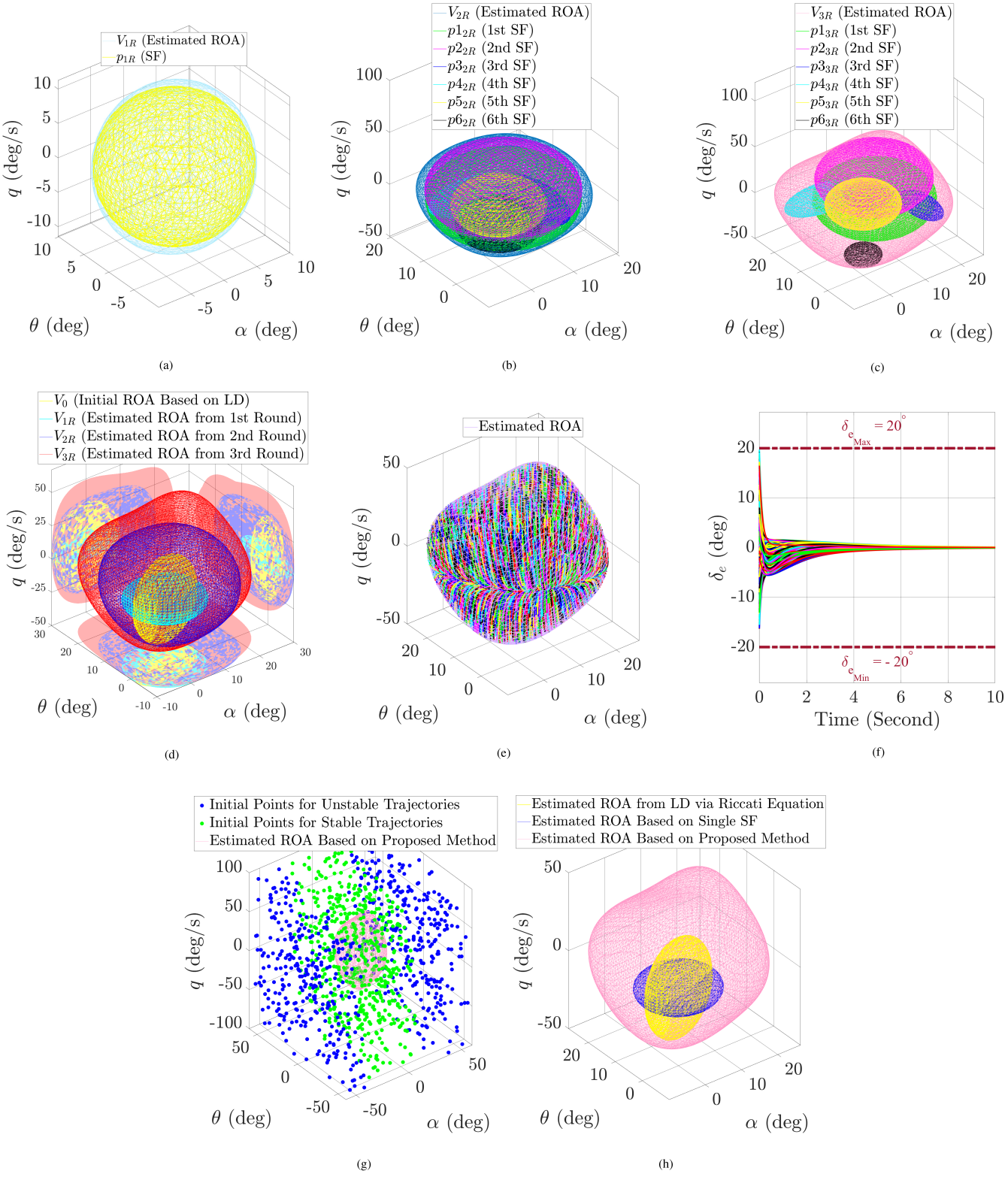


FIGURE 6 | Control System Design with ROA Estimation for Example 4: System with Both Control Input Saturation and State Constraints (Case II). (a) ROA Estimation from the 1st Round, (b) ROA Estimation from the 2nd Round, (c) ROA Estimation from the 3rd Round, (d) Comparison of Estimated ROAs Across all the Rounds, (e) Trajectories of the System Under the Designed Control Law, (f) Control Input Responses for the Related Trajectories, (g) Monte Carlo Simulation with Estimated ROA, (h) Comparison of Estimated ROAs Across Different Methods.

Acknowledgments

The first author acknowledges the Government of India, Ministry of Social Justice and Empowerment for supporting the studies under the National Overseas Scholarship for Ph.D. with scholarship number 11015/33/2019 SCD-V NOS.

Funding

This work was supported by the National Overseas Scholarship, Ministry of Social Justice and Empowerment, India, Grant/Award Number: 11015/33/2019 SCD-V NOS.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

The MATLAB code supporting the findings of this study is available at: https://github.com/Bhaskar-Biswas/Control-Systems-Design-with-Enlarged-Stability-Domains-via-SOS-Optimization/releases/tag/v1.0

References

1. T. Hu, Z. Lin, and L. Qiu, "Stabilization of Exponentially Unstable Linear Systems With Saturating Actuators," *IEEE Transactions on Automatic Control* 46, no. 6 (2001): 973–979, https://doi.org/10.1109/9.928610.

2. T. Hu and Z. Lin, "On Enlarging the Basin of Attraction for Linear Systems Under Saturated Linear Feedback," in *Proceedings of the 2000 American Control Conference*, vol. 3 (IEEE, 2000), 1766–1770, https://doi.org/10.1109/ACC.2000.879504.

3. G. Garcia, S. Tarbouriech, and J. M. G. Silva, "Dynamic Output Controller Design for Linear Systems With Actuator and Sensor Saturation," in *Proceedings of the 2007 American Control Conference* (IEEE, 2007), 5834–5839, https://doi.org/10.1109/ACC.2007.4282517.

4. M. Mera and I. Salgado, "Robust Control of Linear Systems Under Input Saturation Using Barrier Lyapunov Functions," *International Journal of Dynamics and Control* 6, no. 3 (2018): 1231–1238, https://doi.org/10.1007/s40435-016-0294-2.

5. J. Zhang, Y. Wu, Y. Wang, and Z. Gao, "Stability Analysis and Controller Design of Positive Linear Systems Subject to Actuator Saturation," *Proceedings of the 32nd Chinese Control Conference* (2013): 33–38.

6. J. Da Silva and S. Tarbouriech, "Anti-Windup Design With Guaranteed Regions of Stability: An LMI-Based Approach," in *Proceedings of the 42nd IEEE International Conference on Decision and Control*, vol. 5 (IEEE, 2003), 4451–4456, https://doi.org/10.1109/CDC.2003.1272226.

7. Y. Y. Cao, Z. Lin, and D. Ward, "An Antiwindup Approach to Enlarging Domain of Attraction for Linear Systems Subject to Actuator Saturation," *IEEE Transactions on Automatic Control* 47, no. 1 (2002): 140–145, https://doi.org/10.1109/9.981734.

8. V. Powers and T. Wormann, "An Algorithm for Sums of Squares of Real Polynomials," *Journal of Pure and Applied Algebra* 127, no. 1 (1998): 99–104, https://doi.org/10.1016/S0022-4049(97)83827-3.

9. P. A. Parrilo, "Structured Semidefinite Programs and Semi-Algebraic Geometry Methods in Robustness and Optimization. PhD Thesis. California Institute of Technology, California," (2000), https://resolver.caltech.edu/CaltechETD:etd-05062004-055516.

10. P. A. Parrilo, "Semidefinite Programming Relaxations for Semi-Algebraic Problems," *Mathematical Programming* 96, no. 2 (2003): 293–320, https://doi.org/10.1007/s10107-003-0387-5.

11. G. Balas, A. Packard, P. Seiler, and U. Topcu, *Robustness Analysis of Nonlinear Systems. NASA Workshop* (NASA Langley Research Center, 2009), http://users.cms.caltech.edu/~utopcu/saveMaterial/LangleyWorkshop.html.

12. A. Jasour, *Risk Aware and Robust Nonlinear Planning. Course Notes for MIT 16.s498* (Massachusetts Institute of Technology, 2019), https://ocw.mit.edu/courses/16-s498-risk-aware-and-robust-nonlinear-planning-fall-2019/pages/lecture-notes/.

13. M. M. Peet, *LMI Methods in Optimal and Robust Control. Lecture* (Arizona State University, 2020), https://control.asu.edu/MAE598\LY1\textbackslash_frame.htm.

14. A. Papachristodoulou and S. Prajna, "A Tutorial on Sum of Squares Techniques for Systems Analysis," in *Proceedings of the 2005, American Control Conference*, vol. 4 (IEEE, 2005), 2, https://doi.org/10.1109/ACC.2005.1470374.

15. A. A. Ahmadi, *Sum of Squares (SOS) Techniques: An Introduction. lecture* (Princeton University, 2016), https://www.princeton.edu/~aaa/Public/Teaching/ORF523/ORF523\LY1\textbackslash_Lec15.pdf.

16. S. Lall, *Sums of Squares* (Stanford University, 2011), https://web.stanford.edu/class/ee364b/lectures/sos\LY1\textbackslash_slides.pdf.

17. D. Kunisky, "Lecture Notes on Sum-of-Squares Optimization," (2022), http://www.kunisky.com/static/teaching/2022spring-sos/sos-notes.pdf.

18. P. Seiler, "SOSOPT: A Toolbox for Polynomial Optimization," (2013), https://doi.org/10.48550/arXiv.1308.1889.

19. A. Papachristodoulou, J. Anderson, G. Valmorbida, et al., "SOSTOOLS Version 4.00 Sum of Squares Optimization Toolbox for MATLAB," (2013), https://doi.org/10.48550/arXiv.1310.4716.

20. T. Cunis and B. Legat, "Sequential Sum-Of-Squares Programming for Analysis of Nonlinear Systems," *American Control Conference* (2023): 756–772, https://doi.org/10.23919/ACC55779.2023.10156153.

21. U. Topcu and P. Seiler, "Local Stability Analysis Using Simulations and Sum-Of-Squares Programming," *Automatica* 44, no. 10 (2008): 2669–2675, https://doi.org/10.1016/j.automatica.2008.03.010.

22. A. Chakraborty, P. Seilera, and G. Balas, "Nonlinear Region of Attraction Analysis for Flight Control Verification and Validation," *Control Engineering Practice* 19, no. 4 (2011): 335–345, https://doi.org/10.1016/j.conengprac.2010.12.001.

23. L. Khodadadi, B. Samadi, and H. Khaloozadeh, "Estimation of Region of Attraction for Polynomial Nonlinear Systems: A Numerical Method," *ISA Transactions* 53, no. 1 (2014): 25–32, https://doi.org/10.1016/j.isatra.2013.08.005.

24. A. Iannelli, A. Marcos, and M. Lowenberg, "Robust Estimations of the Region of Attraction Using Invariant Sets," *Journal of the Franklin Institute* 356, no. 8 (2019): 4622–4647, https://doi.org/10.1016/j.jfranklin.2019.02.013.

25. J. Anderson and A. Papachristodoulou, "Robust Nonlinear Stability and Performance Analysis of an F/A-18 Aircraft Model Using Sum of Squares Programming," *International Journal of Robust and Nonlinear Control* 23, no. 10 (2013): 1099–1114, https://doi.org/10.1002/rnc.2928.

26. S. P. Khesal and I. Mohammadzaman, "Nonlinear Robust Roll Autopilot Design Using Sum-Of-Squares Optimization," *Journal of Dynamic Systems, Measurement, and Control* 140, no. 11 (2018): 111005, https://doi.org/10.1115/1.4040218.

27. E. August and A. Papachristodoulou, "Feedback Control Design Using Sum of Squares Optimization," *European Journal of Control* 68 (2022): 100683, https://doi.org/10.1016/j.ejcon.2022.100683.

28. H. J. Ma and G. H. Yang, "Fault-Tolerant Control Synthesis for a Class of Nonlinear Systems: Sum of Squares Optimization Approach," *International Journal of Robust and Nonlinear Control* 19, no. 5 (2009): 591–610, https://doi.org/10.1002/rnc.1346.

29. B. Samadi and L. Rodrigues, "A Sum of Squares Approach to Backstepping Controller Synthesis for Piecewise Affine and Polynomial Systems," *International Journal of Robust and Nonlinear Control* 24, no. 16 (2014): 2365–2387, https://doi.org/10.1002/rnc.2994.

30. J. J. Bramburger, S. Steven Dahdah, and J. R. Forbes, "Synthesizing Control Laws From Data Using Sum-Of-Squares Optimization," (2023), https://doi.org/10.48550/arXiv.2307.01089.

31. F. Meng, D. Wang, P. Yang, G. Xie, and F. Guo, "Application of Sum-Of-Squares Method in Estimation of Region of Attraction for Nonlinear Polynomial Systems," *IEEE Access* 8 (2020): 14234–14243, https://doi.org/10.1109/ACCESS.2020.2966566.

32. Z. J. Wloszek, R. Feeley, T. Weehong, K. Sun, and A. Packard, "Some Controls Applications of Sum of Squares Programming," in *42nd IEEE International Conference on Decision and Control*, vol. 5 (IEEE, 2003), 4676–4681, https://doi.org/10.1109/CDC.2003.1272309.

33. G. Valmorbida, S. Tarbouriech, and G. Garcia, "Design of Polynomial Control Laws for Polynomial Systems Subject to Actuator Saturation," *IEEE Transactions on Automatic Control* 58, no. 7 (2013): 1758–1770, https://doi.org/10.1109/TAC.2013.2248256.

34. A. M. Ribeiro, A. R. Fioravanti, A. Moutinho, and E. C. Paiva, "Control Design Based on Sum of Squares Programming for Non-Affine in Input Systems," in *Proceedings of the 2020 IEEE 6th International Conference on Control Science and Systems Engineering* (IEEE, 2020), 130–135, https://doi.org/10.1109/ICCSSE50399.2020.9171951.

35. E. Ahbe, P. Listov, A. Iannelli, and R. S. Smith, "Feedback Control Design Maximizing the Region of Attraction of Stochastic Systems Using Polynomial Chaos Expansion," *IFAC-PapersOnLine* 53, no. 2 (2020): 7197–7203, https://doi.org/10.1016/j.ifacol.2020.12.550.

36. A. M. Ribeiro, A. R. Fioravanti, A. Moutinho, and D. E. C. Paiva, "Nonlinear State-Feedback Design for Vehicle Lateral Control Using Sum-Of-Squares Programming," *Vehicle System Dynamics* 60, no. 743 (2022): 769, https://doi.org/10.1080/00423114.2020.1844905.

37. B. Biswas, D. Ignatyev, A. Zolotas, and A. Tsourdos, "Design of Controller With Enlarged Region of Attraction Using Union Theorem in Sum-Of-Squares Optimization for the F-8 Aircraft," *AIAA SCITECH 2024 Forum* (2024), https://doi.org/10.2514/6.2024-1018.

38. K. H. Khalil, *Nonlinear Systems*, 3rd ed. (Prentice Hall, 2002).

39. M. Vidyasagar, *Nonlinear Systems Analysis*, 2nd ed. (SIAM, 2002).

40. B. Biswas, D. Ignatyev, A. Zolotas, and A. Tsourdos, "Region of Attraction Estimation Using Union Theorem in Sum-Of-Squares Optimization," (2023), https://doi.org/10.48550/arXiv.2305.11655.

41. I. Pólik and T. Terlaky, "A Survey of the S-Lemma," *SIAM Review* 49, no. 3 (2007): 371–418, https://doi.org/10.1137/S003614450444614X.

42. B. Biswas, I. Ignatyev, A. Zolotas, and A. Tsourdos, "Region of Attraction Estimation Using Multiple Shape Functions in Sum-Of-Squares Optimization for Aircraft Flight Control System," *Aerospace Europe Conference 2023 10th EUCASS 9th CEAS* (2023), https://doi.org/10.13009/EUCASS2023-853.

43. X. S. Cai and Z. Z. Han, "Universal Construction of Control Lyapunov Functions for Linear Systems," *Latin American Applied Research* 36, no. 1 (2006): 15–22.

44. B. Biswas, I. Ignatyev, A. Zolotas, and A. Tsourdos, "Construction of Control Lyapunov Function With Region of Attraction Using Union Theorem in Sum-Of-Squares Optimization," *2023 31st Mediterranean Conference on Control and Automation (MED)* (2023): 771–776, https://doi.org/10.1109/MED59994.2023.10185818.

45. D. Li, D. Ignatyev, A. Tsourdos, and Z. Wang, "Estimation of Non-Symmetric and Unbounded Region of Attraction Using Shifted Shape Function and R-Composition," *ISA Transactions* 136 (2023): 308–322, https://doi.org/10.1016/j.isatra.2022.11.015.

46. T. M. Nguyen, "N-Dimensional Quasipolar Coordinates - Theory and Application. Master's Thesis. University of Nevada. Las Vegas, USA," (2014), https://doi.org/10.34917/5836144.

47. D. Zhao and J. Wang, "Stability Analysis and Design for Polynomial Nonlinear Systems Using SOS With Application to Aircraft Flight Control," *IFAC Proceedings Volumes* 41, no. 2 (2008): 8684–8689, https://doi.org/10.3182/20080706-5-KR-1001.01468.

48. J. F. Sturm, "SeDuMi Version 1.3," (2001), https://sedumi.ie.lehigh.edu/?page_id=58.

Appendix A

Algorithm

In this section, we present the algorithm, which consists of three main steps: the γ-step (with two substeps), the β-step, and the υ-step. Prior to detailing the algorithm, we provide an explanation of the γ-step's substeps. This stage involves synthesizing a control law by optimizing γ^* and determining the related decision variables for a given LF, $V(x)$, and a fixed parameter l_2. However, direct optimization is challenging due to the interdependence of $\boldsymbol{K}_i(x)$ and s_k in Equation (13b). Therefore, this step is divided into two substeps: in each substep, either s_k or $\boldsymbol{K}_i(x)$ is fixed while optimizing γ with respect to the other. The developed Algorithm 1 is detailed below.

ALGORITHM 1 | Iterative Algorithm for Control Law Design With Improved ROA in Constrained Systems.

Initialize:
Find initial CLF V_0 of the linearized system by solving the Riccati Equation.
Define l_1, l_2, degree of V, K_i, and all $s_i \in \sum_n$.
Set iteration numbers: M_{SF}, M, and M_γ.
Set tolerance numbers: δ_{SF}, δ_γ and δ_β.

First Round of Iteration:
Select the number of SF, shape matrix N_i, and center selection angle θ_i.

Center (x^*) Selection of SF, $p_i = (x - x_i^*)N_i(x - x_i^*),\ i = 1, \dots, N_{SF}$**:**
if p_i shifted shape function **then**
  for $j_{SF} = 1$ to M_{SF} **do**
    Set $V \leftarrow V_0$

Step 1_{SF}**:**
Set $s_k = 1$.
Solve $\gamma_{1_{SF}}^* = \max_{s_0, s_{\alpha_{max_1}}, s_{\alpha_{min_1}}, \dots, s_{\alpha_{max_m}}, s_{\alpha_{min_m}}, s_{sc_1}, \dots, s_{sc_t} \in \sum_n,\ K_i(x) \in R_n} \gamma$ subject to [Eq. (13b) - Eq. (13h)] to find $K_i(x)$, s_0, $s_{\alpha_{max_1}}, s_{\alpha_{min_1}}, \dots, s_{\alpha_{max_m}}, s_{\alpha_{min_m}}, s_{sc_1}, \dots, s_{sc_t}$, and $\gamma_{1_{SF}}^*$.

Step 2_{SF}**:**
Using $K_i(x)$ from Step 1_{SF}, solve $\gamma_{2_{SF}}^* = \max_{s_0, s_k, s_{\alpha_{max_1}}, s_{\alpha_{min_1}}, \dots, s_{\alpha_{max_m}}, s_{\alpha_{min_m}}, s_{sc_1}, \dots, s_{sc_t} \in \sum_n} \gamma$ subject to [Eq. (13b) - Eq. (13h)] to find updated values of s_0, s_k, $s_{\alpha_{max_1}}, s_{\alpha_{min_1}}, \dots, s_{\alpha_{max_m}}, s_{\alpha_{min_m}}, s_{sc_1}, \dots, s_{sc_t}$, and $\gamma_{2_{SF}}^*$.

Update $V \leftarrow V_0/\gamma_{2_{SF}}^*$ and $s_k \leftarrow s_k$.
if $\gamma_{2_{SF}}^* - \gamma_{1_{SF}}^* \le \delta_{SF}$ **then**
  break
end if
end for
Output: $\gamma_{SF}^* = \gamma_{2_{SF}}^*$
Find intersection point x_I^* of $V_0 = \gamma_{SF}^*$ and $Y = AX$ (Equation of Straight Line).
Calculate x^*
else
$x^* = 0$
end if

---

for $j = 1$ to M **do**
Set $V \leftarrow V_0$

γ**-step:**
for $j_\gamma = 1$ to M_γ **do**

Substep 1:
Set $s_k = 1$ and using fixed V, solve $\gamma_1^* = \max_{s_0, s_{\alpha_{max_1}}, s_{\alpha_{min_1}}, \dots, s_{\alpha_{max_m}}, s_{\alpha_{min_m}}, s_{sc_1}, \dots, s_{sc_t} \in \sum_n, K_i(x) \in R_n} \gamma$ subject to [Eq. (13b) - Eq. (13h)] to find $K_i(x)$, s_0, $s_{\alpha_{max_1}}, s_{\alpha_{min_1}}, \dots, s_{\alpha_{max_m}}, s_{\alpha_{min_m}}, s_{sc_1}, \dots, s_{sc_t}$, and γ_1^*.

Substep 2:
Using $K_i(x)$ from Substep 1, solve $\gamma_2^* = \max_{s_0, s_k, s_{\alpha_{max_1}}, s_{\alpha_{min_1}}, \dots, s_{\alpha_{max_m}}, s_{\alpha_{min_m}}, s_{sc_1}, \dots, s_{sc_t} \in \sum_n} \gamma$ subject to [Eq. (13b) - Eq. (13h)] to find updated values of s_0, s_k, $s_{\alpha_{max_1}}, s_{\alpha_{min_1}}, \dots, s_{\alpha_{max_m}}, s_{\alpha_{min_m}}, s_{sc_1}, \dots, s_{sc_t}$, and γ_2^*.

Update $V \leftarrow V_0/\gamma_2^*$ and $s_k \leftarrow s_k$.
if $\gamma_2^* - \gamma_1^* \le \delta_\gamma$ **then**
  break
end if
end for
Output: $\gamma^* = \gamma_2^*$ and $K_i(x)$

β-step:
Using p_i, fixed V, and γ^* from γ-step, solve $\{\beta_1^*,\dots,\beta_n^*\} = \max_{s_1,\dots,s_n\in\sum_n} \{\beta_1,\dots,\beta_n\}$ subject to [Eq. (13i) - Eq. (13j)] to find $\beta_1^*,\dots,\beta_n^*$ and corresponding $s_1,\dots,s_n$.

V-step:
With γ^*, $K_i(x)$, s_0, s_k, $s_{\alpha_{max_1}}, s_{\alpha_{min_1}}, \dots, s_{\alpha_{max_m}}, s_{\alpha_{min_m}}, s_{sc_1}, \dots, s_{sc_t}$, $\beta_1^*,\dots,\beta_n^*$ from γ-step and β-step, determine new $V(x)$ subject to [Eq. (13a) - Eq. (13j)].

Update $V \leftarrow V/\gamma^*$
if V is infeasible or $\beta_i(j) - \beta_i(j-1) \le \delta_\beta$ **then**
  break
end if
Output: $V_{1R} = V$, $\gamma_{1R}^* = \gamma^*$, and $K_{i_{1R}}(x) = K_i(x)$
end for

Second Round of Iteration:
Select number of SF, shape matrix N_i, and center selection angle θ_i.
Replace V_0 with V_{1R} and repeat steps from 5 to 40.
Output: $V_{2R} = V$, $\gamma_{2R}^* = \gamma^*$, and $K_{i_{2R}}(x) = K_i(x)$

Third Round of Iteration:
Repeat same as the Second Round.

Repeat the rounds of iteration till the algorithm can synthesize the feedback control law $K_i(x)$, V, and related γ^*.

Output: Feedback Control Law $K_{i_{nR}}(x)$ and Estimated ROA represented by $V_{nR} = \gamma_{nR}^*$.

The developed algorithm is implemented using the SOSOPT toolbox [18] to obtain simulation results. The SOSOPT toolbox translates the relevant steps into semidefinite programming and utilizes the SeDuMi [48] toolbox for problem-solving. Further details are provided in [18].

Appendix B

Designed Control Law and Associated ROA

The final control law design, along with the related ROA defined by the Lyapunov function V and the parameter γ, are provided for all the examples below.

Example 1.

$$u(x) = -377386.6197x_1 + 17695.9174x_2$$

$$V(x) = 0.00069322x_1^2 + 0.0013739x_1x_2 + 0.00068724x_2^2$$

$$\gamma = 1.9836.$$

Example 2.

$$\begin{aligned} u(x) &= 0.012415x_1^3 - 0.5589x_1^2x_2 + 0.43228x_1x_2^2 \\ &\quad - 0.32737x_2^3 - 0.61339x_1^2 + 0.74733x_1x_2 \\ &\quad - 0.11123x_2^2 - 0.48251x_1 - 2.204x_2 \\ V(x) &= 0.0027654x_1^4 + 0.01059x_1^3x_2 + 0.096222x_1^2x_2^2 \\ &\quad - 0.064285x_1x_2^3 + 0.11903x_2^4 + 0.037543x_1^3 \\ &\quad + 0.28817x_1^2x_2 - 0.34975x_1x_2^2 \\ &\quad + 0.00049996x_2^3 + 0.23333x_1^2 + 0.0072942x_1x_2 + 0.68308x_2^2 \\ \gamma &= 1.004. \end{aligned}$$

Example 3.

$$\begin{aligned} u(x) &= -2.015272903074579x_1^3 - 3.053120399180933x_1^2x_2 \\ &\quad + 0.3853638304019745x_1^2x_3 + 0.01005903561886043x_1x_2^2 \\ &\quad + 1.52795130757319x_1x_2x_3 + 0.3377437325899847x_1x_3^2 \\ &\quad - 2.291290382798506x_2^3 + 0.6639661741751245x_2^2x_3 \\ &\quad - 0.06271481330214004x_2x_3^2 + 0.1142286768531654x_3^3 \\ &\quad - 4.481489362409095x_1^2 - 3.033063543418997x_1x_2. \\ &\quad + 0.2345902988136049x_1x_3 - 5.82589132718947x_2^2 \\ &\quad + 0.7924733779914171x_2x_3 - 0.2694986632315007x_3^2 \\ &\quad - 4.260810233597976x_1 - 6.115679519707461x_2 \\ &\quad - 2.401729193400217x_3, \\ V(x) &= 1.174819243802376x_1^4 + 1.793885001825741x_1^3x_2 \\ &\quad + 0.6106542570169028x_1^3x_3 + 1.044410967242089x_1^2x_2^2 \\ &\quad + 0.4468283081884593x1^2x_2x_3 \\ &\quad + 0.0968634061739823x_1^2x_3^2 \\ &\quad + 0.6092026306693128x_1x_2^3 + 0.1206397643853388x_1x_2^2x_3 \\ &\quad + 0.1545027401089984x_1x_2x_3^2 \\ &\quad - 0.001643781832404293x_1x_3^3 \\ &\quad + 1.480325221310148x_2^4 + 0.3219555529753837x_2^3x_3 \\ &\quad + 0.4050556487513388x_2^2x_3^2 \\ &\quad + 0.07112121534637852x_2x_3^3 \\ &\quad + 0.006797300925833009x_3^4 + 2.41504756144577x_1^3 \\ &\quad + 3.061701992791699x_1^2x_2 \end{aligned}$$

$$
\begin{aligned}
&+ 1.001723466000406x_1^2x_3 \\
&+ 1.739190464677991x_1x_2^2 + 0.4895982744733679x_1x_2x_3 \\
&+ 1.739190464677991x_1x_2^2 + 0.4895982744733679x_1x_2x_3 \\
&+ 0.165587337733398x_1x_3^2 + 3.5779738850015x_2^3 \\
&+ 0.449094252745854x_2^2x_3 + 0.3549196604599001x_2x_3^2 \\
&+ 0.02443530992910684x_3^3 + 1.506565973689276x_1^2 \\
&+ 1.411397416156399x_1x_2 + 0.5994564568474934x_1x_3 \\
&+ 2.57010203351349x_2^2 + 0.4982858500459816x_2x_3 \\
&+ 0.2202595048937496x_3^2
\end{aligned}
$$

$\gamma = 1.0004.$

Example 4. *Case I*

$$
\begin{aligned}
\delta_e &= 0.4884931355165316\alpha + 0.043922705576599 66\theta \\
&\quad + 0.05930674079274591q \\
V &= 1.510741367641813\alpha^4 - 0.242411958956612\alpha^3\theta \\
&\quad + 0.4232415818886222\alpha^3q + 0.5586059568059923\alpha^2\theta^2 \\
&\quad - 0.09458355491546673\alpha^2\theta q + 0.04243745636389808\alpha^2q^2 \\
&\quad - 0.174648586416764\alpha\theta^3 + 0.07929906300704438\alpha\theta^2q \\
&\quad + 0.009326326442751649\alpha\theta q^2 + 0.004028294000145 37\alpha q^3 \\
&\quad + 0.048226679419330 16\theta^4 + 0.005296267723818913\theta^3q \\
&\quad + 0.030573086899308 14\theta^2q^2 + 0.003885521576047 62\theta q^3 \\
&\quad + 0.001254050884119508q^4 + 0.001253339295513801\alpha^3 \\
&\quad - 0.5188025120144631\alpha^2\theta + 0.1750567973350869\alpha^2q \\
&\quad + 0.1802911814046348\alpha\theta^2 + 0.007306706367104545\alpha\theta q \\
&\quad - 0.03022052243424863\alpha q^2 - 0.046402239983774 66\theta^3 \\
&\quad + 0.018438342987325 59\theta^2q - 0.001017308468522 1 35\theta q^2 \\
&\quad + 0.002194112546619497q^3 + 2.038099690115847\alpha^2 \\
&\quad + 0.1188941800656749\alpha\theta + 0.05889498921225145\alpha q \\
&\quad + 0.073090712411432 56\theta^2 + 0.019369859139587 06\theta q \\
&\quad + 0.08280965022372345q^2
\end{aligned}
$$

$\gamma = 1.0014.$

Example 5. *Case II*

$$
\begin{aligned}
\delta_e &= -0.07918187266858052\alpha + 0.19483127442675 55\theta \\
&\quad + 0.3955190112841407q \\
V &= 112.658893354019\alpha^4 + 28.81876887438133\alpha^3\theta \\
&\quad + 11.75505391588004\alpha^3q + 81.35861053798675\alpha^2\theta^2 \\
&\quad + 2.962060954611452\alpha^2\theta q + 9.932817538018412\alpha^2q^2 \\
&\quad + 15.78805851835598\alpha\theta^3 + 15.46056251683486\alpha\theta^2q \\
&\quad - 12.17494917221524\alpha\theta q^2 + 0.7605426858611394\alpha q^3 \\
&\quad + 103.52974857987 53\theta^4 + 8.239909590406 43\theta^3q \\
&\quad + 18.01545111351598\theta^2q^2 + 2.189522043383072\theta q^3 \\
&\quad + 1.677340629174642q^4 - 82.55454054435796\alpha^3 \\
&\quad - 51.69911317486084\alpha^2\theta - 6.394651419863822\alpha^2q \\
&\quad - 30.67604229222711\alpha\theta^2 - 8.599517868965277\alpha\theta q \\
&\quad - 2.045491375434521\alpha q^2 - 80.985446138452 46\theta^3 \\
&\quad - 9.342712580812167\theta^2q - 5.579623980333669\theta q^2 \\
&\quad - 0.8249437765478003q^3 + 20.113335922087\alpha^2 \\
&\quad + 10.03330240953354\alpha\theta + 1.333097042896634\alpha q \\
&\quad + 19.491414672295 29\theta^2 + 2.757465874652423\theta q \\
&\quad + 0.8234032492500502q^2
\end{aligned}
$$

$\gamma = 1.0004.$

In the control law δ_e and the Lyapunov function V formulated above for Example 4, the variables are expressed in radians. Multiplying each variable by $\pi/180$ allows the conversion of δ_e and V into degree units.



https://dspace.org/

Control systems design with enlarged stability domains for nonlinear constrained systems via sumofsquares optimization

Biswas, Bhaskar; Ignatyev, Dmitry; Zolotas, Argyrios; Tsourdos, Antonios

2026-12-31
Wiley
https://doi.org/10.1002/rnc.70606
Biswas B, Ignatyev D, Zolotas A, Tsourdos A. (2026) Control systems design with enlarged stability domains for nonlinear constrained systems via sumofsquares optimization. International Journal of Robust and Nonlinear Control, Available online 23 June 2026, Article number rnc.70606

Downloaded from a DSpace repository. https://dspace.org/