

# Synthesis and verification of robust-adaptive safe controllers

Simin Liu<sup>\*1</sup>

Kai S. Yun<sup>\*2</sup>

John M. Dolan<sup>1</sup>

Changliu Liu<sup>1</sup>

**Abstract**—Safe control with guarantees generally requires the system model to be known. It is far more challenging to handle systems with uncertain parameters. In this paper, we propose a generic algorithm that can synthesize and verify safe controllers for systems with constant, unknown parameters. In particular, we use robust-adaptive control barrier functions (raCBFs) to achieve safety. We develop new theories and techniques using sum-of-squares that enable us to pose synthesis and verification as a series of convex optimization problems. In our experiments, we show that our algorithms are general and scalable, applying them to three different polynomial systems of up to moderate size (7D). Our raCBFs are currently the most effective way to guarantee safety for uncertain systems, achieving 100% safety and up to 55% performance improvement over a robust baseline.

## I. INTRODUCTION

Although much work has been done on safety for known systems, comparatively little work has been done for systems whose dynamical models cannot perfectly be known. These uncertain systems are a challenging domain for safe control on both a theoretical and methodological level. Effective safe controllers must meet two demands: (1) provide mathematical guarantees of safety and (2) be high-performance, interfering minimally with other control objectives (i.e., stabilizing, tracking). Currently, many safe control synthesis methods for uncertain systems either lack guarantees or only obtain safety at a high cost to performance. Secondly, algorithms for controller search should be both generic and tractable, so they can handle many different systems of varying dimensions. This work aims to address all of these requirements.

The technique we use is “sum-of-squares programming” (SOSP) [1, 2]. Any robust-adaptive CBF (raCBF) produced by our SOSP-based algorithm is guaranteed to provide safety. SOSP allows us to pose the search for a valid raCBF, given a partially-known system and a safety requirement, as a constrained optimization problem. This optimization problem can be broken down into a series of convex optimization problems, which can each be solved with an off-the-shelf semidefinite program (SDP) solver.

Typically, uncertain systems are handled with adaptive, robust, or robust-adaptive control [3]. We choose the robust-adaptive paradigm, which combines the merits of robust and adaptive control. In adaptive control, the controller depends on estimates of the unknown system. Typically, the system will be estimated online and the controller *adapts* with each

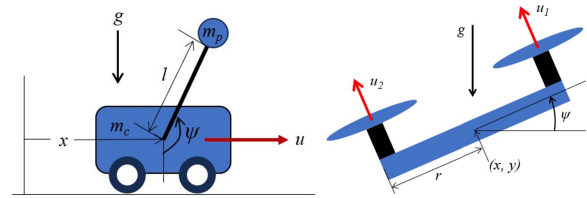


Fig. 1: Two of our three test systems: cartpole with unknown joint friction and quadrotor with unknown drag coefficients.

new estimate. In adaptive fashion, our raCBF is a function of the estimated parameters as well as the state. This raCBF is also paired with a special parameter estimation law. This controller structure allows it to effectively *select* a CBF online with the choice of parameter. The robust-adaptive scheme leverages this to attain *selective conservatism*: more conservative CBFs are only “selected” when necessary, like when the state suddenly moves toward danger.

The raCBF also incorporates elements of robust control. In robust control, the controller only depends on worst-case disturbance bounds. Our raCBF follows robust principles by accounting for maximum estimation error. However, it is able to account for it in a less conservative way compared to robust CBF (rCBF), as it assumes parameter estimation occurs. Ultimately, these differences lead our synthesized raCBF to perform better. That is, it provides safety while minimally hindering control efficiency at tasks like stabilization and tracking.

Our contributions are as follows:

- Propose the first verification and synthesis methods for adaptive-type safe controllers for uncertain systems.
- For verification: derive convex-equivalent conditions for an raCBF to provide safety.
- For synthesis: design a multi-phase algorithm which yields a valid raCBF with a locally optimal invariant set.
- Illustrate our algorithm’s genericness and scalability by applying it to three examples with varying dynamics, dimensions ( $\leq 7D$ ), and safety specifications.
- Illustrate our synthesized raCBF’s safety and superior performance (up to 55% improvement over robust baseline) in simulation.

## II. RELATED WORKS

In this section, we briefly survey existing works on safety for uncertain systems. We primarily focus on CBF works.

**Theory of (r/a)-CBFs:** these works mathematically define novel forms of CBF (like robust [4, 5, 6] and adaptive [7, 8, 9] CBFs). This category has a different focus than synthesis

\* Denotes equal contribution

<sup>1</sup>Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA.

<sup>2</sup>Mechanical Engineering Department, Carnegie Mellon University, Pittsburgh, PA 15213, USA.

and verification (systematically generating CBFs). Hence, works in this category typically just use hand-derivation to find CBFs for toy systems, which is an approach that is difficult, if not impossible, to apply to more complex systems.

**Verification and synthesis of (r/a)-CBFs:** this category, which our work belongs to, deals with developing new theories and algorithms for systematically checking and generating safety functions. While synthesis of ordinary CBFs (i.e., for known systems) has been explored extensively [10, 11, 12], to our knowledge, there are very few works for robust CBF synthesis [13] and none for adaptive synthesis. Our work is one of a handful that synthesize CBFs for uncertain systems, and the first to synthesize an adaptive-type CBF. While our synthesis technique carries guarantees of correctness, there are ad-hoc synthesis works which do not [14, 15]. There are also sample-based methods for verification, which are unable to scale past 5D due to the curse of dimensionality [15]. Compare this to our SOSP-based verification scheme, which has polynomial time complexity and can scale to 7D and possibly beyond.

**Online optimization of (r/a)-CBFs:** this body of work typically assumes a valid CBF already exists. Ordinary CBFs compute safe control inputs using quadratic programs (QPs) solved at each time step [16, 17]. Works in this category investigate how best to pose this online optimization (i.e., possibly as a different type of program, like a second-order cone program) for new forms of CBFs [18, 19, 20]. Note that one advantage of our robust-adaptive formulation is that it can still be applied with a QP, which is simple and fast to solve.

**Ad-hoc safe controllers:** these are works which cannot guarantee safety, because they cannot prove that an admissible safe control input exists for all states and all realizations of the unknown system [21, 22, 23]. By contrast, our approach can ensure that there exists a safe control input for all possible conditions, therefore guaranteeing system safety.

**Non-CBF approaches:** there are bodies of work on using HJB reachability [24] or set-based methods [25, 26] to compute robust backward or forward reachable sets. However, a significant drawback of these methods is they scale poorly, often not beyond 5D.

### III. PROBLEM FORMULATION

#### A. Safe control for uncertain systems

In this first subsection, we define the problem of safe control for uncertain systems. We also outline system assumptions and our definition of safety. We consider a system that is affine in the unknown parameters and the control input:

$$\dot{x} = f(x) - \Delta(x)\theta + g(x)u \quad (1)$$

$$u \in \mathcal{U}, x \in \mathcal{X}, \theta \in \Theta \quad (2)$$

where  $f, g, \Delta$  are polynomial functions in  $x$  and  $x \in \mathcal{X} \subseteq \mathbb{R}^n, u \in \mathcal{U} \subseteq \mathbb{R}^m, \theta \in \Theta \subset \mathbb{R}^p$ . We also assume the set of admissible inputs and the robust parameter range are both

hyperrectangles (i.e.,  $\underline{u} \leq u \leq \bar{u}$  and  $\underline{\theta} \leq \theta \leq \bar{\theta}$ ). We want to enforce safety of this system in the sense of “set invariance”.

**Definition 1 (Set invariance).** A set  $\mathcal{I}$  is forward invariant if all trajectories starting inside  $\mathcal{I}$  remain inside the set for all time.

The set  $\mathcal{X}_{\text{safe}}$  is provided as part of the problem specification and is represented implicitly using continuous and smooth functions  $b_i(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ :

**Definition 2 (Safety specification).**

$$\mathcal{X}_{\text{safe}} = \{x | b_i(x) \geq 0, \forall i = 1, \dots, t\} \quad (3)$$

#### B. Robust-adaptive CBFs

In this second subsection, we summarize the raCBF formulation and outline the constraints that valid raCBFs must satisfy. We choose the formulation in [27], which is less conservative than [4, 5, 6, 7, 9] and compatible with more nominal controllers than [8]. The raCBF has the form  $\phi(x, \hat{\theta})$  where it depends on not only the state, but also on the estimate of the unknown parameters. The estimate,  $\hat{\theta}$ , is initialized with a guess and then updated online according to:

$$\dot{\hat{\theta}} = \gamma \nu(\rho) \Delta(x)^\top \nabla_x \phi(x, \hat{\theta}), \quad (4)$$

$$\dot{\rho} = \frac{\nu(\rho)}{\nabla \nu(\rho)} \frac{1}{\phi(x, \hat{\theta}) + \eta} [-\sigma \rho + w_{\hat{\theta}}(x)], \quad (5)$$

$$w_{\hat{\theta}} = \begin{cases} 0 & \text{if } \nabla_{\hat{\theta}} \phi(x, \hat{\theta})^\top \dot{\hat{\theta}} \geq 0 \\ -\zeta \nabla_{\hat{\theta}} \phi(x, \hat{\theta})^\top [\gamma \Delta(x)^\top \nabla_x \phi(x, \hat{\theta})] & \text{otherwise} \end{cases} \quad (6)$$

where  $\gamma$  is an admissible adaptation gain,  $\nu(\rho)$  is a scaling function such that  $1 \leq \nu(\rho) \leq \zeta < \infty$  for  $\zeta > 1$  and also  $\nabla \nu(\rho) > 0$ . Also,  $\eta, \sigma, \zeta \in \mathbb{R}_{>0}$  are design parameters. We assume that  $\theta$  belongs to a bounded set  $\Theta$ . For many systems, we can easily provide a guess for  $\Theta$  based on domain knowledge [3].

**Theorem 1.** Define the set  $\mathcal{I}_{\hat{\theta}}^r = \{x \in \mathcal{X}, \hat{\theta} \in \Theta | \phi(x, \hat{\theta}) \geq \frac{1}{2\gamma} \hat{\theta}^\top \hat{\theta}\}$  where  $\hat{\theta}$  is the maximum possible estimation error.  $\hat{\theta} = \bar{\theta} - \theta$ , where  $\theta \in [\underline{\theta} : \bar{\theta}]$ . Then, an raCBF must satisfy the following constraints to ensure invariance of  $\mathcal{I}_{\hat{\theta}}^r$ :

$$\sup_{u \in \mathcal{U}} \left\{ \nabla_x \phi(x, \hat{\theta})(f - \Delta \hat{\theta} + gu) \right\} \geq -\alpha(\phi(x, \hat{\theta}) - \frac{1}{2\gamma} \hat{\theta}^\top \hat{\theta}) \quad (7)$$

$$\phi(x, \hat{\theta}) \geq 0 \implies b_i(x) \geq 0, \forall i = 1, \dots, t. \quad (8)$$

*Proof.* Eq. (7) by Thm. 1 in [27] and Eq. (8) by definition of safety.  $\square$

Eq. (7) says that the raCBF must be a function that defines a robust invariant set under the true dynamics. Specifically, the robust set  $\mathcal{I}_{\hat{\theta}}^r$  is invariant iff there always exists an admissible input to repel the state from the set boundary. This is implied by Eq. (7), which requires  $\sup_{u \in \mathcal{U}} \dot{\phi} \geq 0$  (repulsion) when  $\phi = \frac{1}{2\gamma} \hat{\theta}^\top \hat{\theta}$  (state at the boundary). Now, typically, the invariant set is considered  $\{x | \phi \geq 0\}$ . The robust set is shrunk from this typical set by a margin proportional to the maximum estimation error. Intuitively, this triggers safe control earlier than is typical, which mitigates the impact of estimation error.

Additionally, the raCBF must obey the safety specification. Eq. (8) says that the raCBF may define an invariant set that is

no larger than  $\mathcal{X}_{\text{safe}}$ . Often,  $\mathcal{X}_{\text{safe}}$  itself cannot be an invariant set. There will be some states within  $\mathcal{X}_{\text{safe}}$  that will leave the set for any admissible control input.

Given a valid raCBF, a controller design is immediate. We simply apply the raCBF as an optimization-based safety-filter on top of a nominal control policy. Since any control input satisfying Eq. (7) will keep the system safe, this filter simply projects the nominal control signal to the space of  $u$  satisfying Eq. (7).

**Definition 3** (raCBF quadratic program (raCBF-QP)). *Given nominal controller  $u_{\text{ref}} = \pi(x)$ ,*

$$\min_{u \in \mathcal{U}} \|u_{\text{ref}} - u\|_2^2 \quad (9)$$

$$\nabla_x \phi(x, \hat{\theta})(f - \Delta \hat{\theta} + gu) \geq -\alpha \left( \phi(x, \hat{\theta}) - \frac{1}{2\gamma} \tilde{\theta}^\top \tilde{\theta} \right). \quad (10)$$

Notice that the filtered input is computed using the *estimated* dynamics. This is why we require the raCBF, and the filtered input, by consequence, to be robust to estimation error.

#### IV. PRELIMINARIES: SOSP

We use sum-of-squares programming (SOSP) as the optimization framework for verification and synthesis. In the following section, we briefly summarize key definitions and theorems for SOSP. In general, checking polynomial non-negativity is NP-hard. However, checking that some  $p(x) \in \mathcal{R}_n$  (the set of polynomials in  $n$  variables) is an SOS polynomial (Def. 4) is completely tractable [28].

**Definition 4** (Sum-of-squares polynomial). *A polynomial  $p(x)$  is SOS if  $p(x) = \sum_i q_i(x)^2$  for some polynomials  $q_i$ . Clearly, this implies  $p(x) \geq 0$ . We denote the set of SOSP as  $\Sigma$ .*

Hence, a popular kind of optimization problem is the SOSP, which has constraints of this form (enforcing a polynomial to be sum-of-squares).

**Definition 5** (Sum-of-squares program). *An SOSP is a convex optimization problem of the following form for a given cost vector  $c \in \mathbb{R}^{p \times q}$*

$$\min_w c^\top w \quad (11)$$

$$a_0^i(x) + \sum_{\forall j \in 1:p} w_j^i a_j^i(x) \in \Sigma, \quad \forall i = 1, \dots, q \quad (12)$$

where  $a_j^i(x) \in \mathcal{R}_n$ .

The definition above says that an SOSP constrains  $q$  polynomials to be sum-of-squares. These polynomials may have variable coefficients ( $w_j^i$ ) and the objective is linear in these coefficients. Next, we introduce the cornerstone theorem, the Positivstellensatz (P-satz), which allows us to transform all different kinds of constraints (e.g., logical implications of polynomial inequalities) into the standard SOSP constraint form of Eq. (12).

**Theorem 2** (Positivstellensatz [29]). *Given polynomials  $\{f_1, \dots, f_r\}$ ,  $\{g_1, \dots, g_t\}$ , and  $\{h_1, \dots, h_u\}$  in  $\mathcal{R}_n$ , the following are equivalent:*

(a) *The set*

$$\left\{ x \in \mathbb{R}^n \left| \begin{array}{l} f_1(x) \geq 0, \dots, f_r(x) \geq 0 \\ g_1(x) \neq 0, \dots, g_t(x) \neq 0 \\ h_1(x) = 0, \dots, h_u(x) = 0 \end{array} \right. \right\}$$

*is empty.*

(b) *There exist polynomials  $f \in \mathcal{P}(f_1, \dots, f_r)$ ,  $g \in \mathcal{M}(g_1, \dots, g_t)$ ,  $h \in \mathcal{I}(h_1, \dots, h_u)$  such that*

$$f + g^2 + h = 0 \quad (13)$$

where  $\mathcal{M}(g_1, \dots, g_t)$  is the multiplicative monoid generated by the  $g_i$ 's and consists of all their finite products;  $\mathcal{P}(f_1, \dots, f_r)$  is the cone generated by the  $f_i$ 's:

$$\mathcal{P}(f_1, \dots, f_r) = \left\{ s_0 + \sum_{i=1}^l s_i b_i \left| \begin{array}{l} l \in \mathbb{Z}_+, s_i \in \Sigma_n, \\ b_i \in \mathcal{M}(f_1, \dots, f_r) \end{array} \right. \right\}$$

and finally,  $\mathcal{I}(h_1, \dots, h_u)$  is the ideal generated by the  $h_i$ 's:

$$\mathcal{I}(h_1, \dots, h_u) = \left\{ \sum h_k p_k \mid p_k \in \mathcal{R}_n \right\}. \quad (14)$$

The P-satz allows us to convert statements in the form of (a) to the form of (b), which can then straightforwardly be turned into an SOSP-type constraint. However, it is often inconvenient to apply P-satz directly, since it generates complex constraints (e.g., the cone adds  $2^r$  terms, each with a polynomial multiplier). Instead, we can use a simplification of P-satz called the S-procedure:

**Definition 6** (S-procedure [28]). *A sufficient condition for proving  $p(x) \geq 0$  on the semi-algebraic set  $K = \{x \in \mathbb{R}^n \mid b_1(x) \geq 0, \dots, b_m(x) \geq 0\}$  is the existence of polynomials  $r_i(x)$ ,  $i = 0, \dots, m$  that satisfy*

$$(1 + r_0(x))p(x) - \sum_{i=1}^m r_i(x)b_i(x) \in \Sigma \quad (15)$$

$$r_i(x) \in \Sigma, i = 0, \dots, m. \quad (16)$$

Frequently, we simplify this condition further by taking  $r_0(x) = 0$ . Together, the concepts from this section allow us to write the constraints which define valid raCBFs in the form of SOSP constraints, which makes them amenable to verification and synthesis.

#### V. METHODOLOGY

In the following sections, we show how a valid raCBF can be characterized using several SOSP constraints. With this, we can pose raCBF verification as an SOSP; this allows us to efficiently certify whether a given raCBF is valid. Additionally, we propose an algorithm for synthesizing an raCBF, given the partially unknown system and the safety definition. The algorithm is a sequence of SOSPs that are iteratively solved until convergence.

##### A. raCBF verification

In this section, we derive an SOSP that allows us to verify if a given raCBF satisfies Theorem 1. Our goal is to convert Eq. (7), (8) into the form of Eq. (12).

Looking first at Eq. (7), we see that it requires us to show that for all  $x \in \mathcal{X}, \theta \in \Theta$ , there exists some  $u \in \mathcal{U}$  satisfying the constraint. The ‘‘exists’’ quantifier complicates

the conversion, as theorems like P-satz and S-procedure can only support “for all” quantifiers. However, it turns out that for any  $x$ , we can identify which  $u \in \mathcal{U}$  achieves the supremum in Eq. (7) - call it  $u^*(x)$ . Therefore, we only need to check whether Eq. (7) is satisfied for  $u^*(x)$  for all  $x$ .

We derive  $u^*(x)$  by regarding Eq. (7) as a constrained optimization over the input  $u$ . In fact, it is a special kind of optimization problem with an affine objective and a compact, convex constraint set. Hence, the maximizer always occurs at a vertex of the constraint set [30],  $v^j$  where  $v_i^j \in \{u_i, \bar{u}_i\}$ . For each  $x$ , we can further identify which vertex is the maximizer. First, we write Eq. (7) more clearly in control-affine form. Let  $c(x, \hat{\theta}) := \nabla_x \phi(x, \hat{\theta})(f - \Delta \hat{\theta}) + \alpha(\phi(x, \hat{\theta}) - \frac{1}{2\gamma} \hat{\theta}^\top \hat{\theta})$  and  $d(x, \hat{\theta}) := \nabla_x \phi(x, \hat{\theta})g(x)$ , so that Eq. (7) can be rewritten as  $\sup_{u \in \mathcal{U}} c(x, \hat{\theta}) + d(x, \hat{\theta})u$ . For  $i = 1, \dots, m$ , if  $d(x, \theta)_i \geq 0$ , i.e., the  $i^{\text{th}}$  entry of vector  $d(x, \theta)$  is nonnegative, then  $u_i^*(x) = \bar{u}_i$  and vice versa. There are then  $2^m$  groups of  $x$  with distinct sign patterns on  $d(x, \theta)$  that have distinct maximizers  $u^*(x)$ . Hence, we can write  $2^m$  constraints for all the groups of  $x$ .

We give an example for  $m = 1$ , although the general case is an immediate extension. We have two constraints:

$$\begin{aligned} \forall x \in \mathcal{X}, \theta \in \Theta, \text{ s.t. } \phi(x, \theta) \geq \beta^- : \\ d(x, \theta) \geq 0 \implies c(x, \theta) + d(x, \theta)\bar{u} \geq 0, \quad (17) \\ d(x, \theta) \leq 0 \implies c(x, \theta) + d(x, \theta)\underline{u} \geq 0. \quad (18) \end{aligned}$$

Applying the S-procedure gives us the following SOSP constraints:

$$\begin{aligned} c + d\bar{u} - s_1 d - s_2(\phi - \beta^-) \in \Sigma, \quad s_1, s_2 \in \Sigma, \quad (19) \\ c + d\underline{u} + s_3 d - s_4(\phi - \beta^-) \in \Sigma, \quad s_3, s_4 \in \Sigma. \quad (20) \end{aligned}$$

Likewise, we can also apply the S-procedure to (8) to yield:

$$\phi(x, \theta) - r_i b_i \in \Sigma, \quad \forall i = 1, \dots, t, \quad r_i \in \Sigma. \quad (21)$$

**Theorem 3** (raCBF verification). *For the case where  $m = 1^1$ , a polynomial  $\phi(x, \hat{\theta})$  is a valid raCBF, providing set invariance and finite-time convergence in  $\{x | \phi \geq \beta^-\}$ , if there exists  $s_{1:4}, r_i$  such that the convex constraints in Eq. (7) and (8) are satisfied. This can be determined by seeing if the SOSP consisting of constraints (19), (20), (21) is feasible.*

*Proof.* If there exists such  $s_{1:4}, r_i$ , then by Def. 6, Thm. 1 is satisfied.  $\square$

## B. raCBF synthesis

In this section, we provide an algorithm for synthesizing an raCBF given the partially known system (Eq. 1), the safety definition (Eq. 3), and a feasible initialization for the raCBF. We would like to find  $\phi(x, \hat{\theta})$  satisfying Theorem 1, where  $\phi$  is now a polynomial of fixed degree over  $x, \theta$  with variable coefficients. However, with  $\phi$  containing optimization variables, Eq. (19), (20) are no longer convex, but bilinear. This requires us to apply bilinear alternation on

<sup>1</sup>The case for general  $m$  is an immediate extension, but omitted for notational brevity.

---

## Algorithm 1 Synthesis of raCBFs via a sequence of three SOSPs

---

```

Start with  $\phi^{(0)}, \theta_l^{(0)}, \theta_u^{(0)}, c^{(0)} = \infty, i = 0, \text{ converged} = \text{False}$ .
Denote the objective value as  $c^{(i)}$ .
while not converged do
  Ellipsoid  $\mathcal{E}_\delta^{(i)} \leftarrow \mathbf{P1}(\phi^{(i)}, \theta_l^{(i)}, \theta_u^{(i)})$ 
  Multipliers  $s_i(x, \theta), r_i(x, \theta) \leftarrow \mathbf{P2}(\phi^{(i)}, \theta_l^{(i)}, \theta_u^{(i)}, \mathcal{E}_\delta^{(i)})$ 
   $\phi^{(i+1)}, \theta_l^{(i+1)}, \theta_u^{(i+1)}, c^{(i+1)} \leftarrow \mathbf{P3}(s_i(x, \theta), r_i(x, \theta), \mathcal{E}_\delta^{(i)})$ 
  if  $|c^{(i+1)} - c^{(i)}| \leq \epsilon$  then
    converged  $\leftarrow$  True
  end if
end while

```

---

these constraints, which form the basis of our Alg. 1. We optimize the multipliers  $s_i(x, \theta), r_i(x, \theta)$  in Phase 2 and the raCBF  $\phi(x, \hat{\theta})$  in Phase 3 and continue back and forth.

### Phase 2 (22)

$$\begin{aligned} \min_{s_i, r_i \in \Sigma, t \in \mathbb{R}} t \\ c + d\bar{u} - s_1 d - s_2(\phi - \beta^-) \in \Sigma \quad \text{Ctrl fltr feas. constrs} \\ c + d\underline{u} + s_3 d - s_4(\phi - \beta^-) \in \Sigma \\ \phi - r_i b_i \in \Sigma, \quad \forall i = 1, \dots, t \quad \text{Subset constraints} \\ t - \phi - s_5(\delta - x^\top P x) \in \Sigma \quad \text{Ellipsoid constraint} \\ 1 - \phi(0, \theta) \in \Sigma \quad \text{Anchor constraint} \end{aligned}$$

### Phase 3 (23)

$$\begin{aligned} \min_{\phi, t, \theta_l, \theta_u} t + \theta_l - \theta_u \\ c + d\bar{u} - s_1 d - s_2(\phi - \beta^-) \in \Sigma \quad \text{Ctrl fltr feas. constrs} \\ c + d\underline{u} + s_3 d - s_4(\phi - \beta^-) \in \Sigma \\ \phi - r_i b_i \in \Sigma, \quad \forall i = 1, \dots, t \quad \text{Subset constraints} \\ t - \phi - s_5(\delta - x^\top P x) \in \Sigma \quad \text{Ellipsoid constraint} \\ 1 - \phi(0, \theta) \in \Sigma \quad \text{Anchor constraint} \end{aligned}$$

While bilinear alternation alone would yield a *valid* raCBF, we’d also like the raCBF to capture a large invariant set. A larger invariant set means the system is less restricted, providing the nominal controller more flexibility. Thus, our optimization needs to also maximize the size of the invariant set  $\mathcal{I}_\theta^r$ . To do this, we follow [12]. To increase its size, we perform several steps in each iteration. First, we find the largest ellipsoid  $\mathcal{E}_\delta := \{x | x^\top P x \leq \delta\}$  contained within  $\mathcal{I}_\theta^r$  (solve Phase 1 for the maximum ellipsoid radius  $\delta$ ). Then, we optimize the raCBF so that the margin between  $\mathcal{E}_\delta$  and  $\mathcal{I}_\theta^r$  is as large as possible (Phase 2). This effectively increases the size of  $\mathcal{I}_\theta^r$  over the iterations. We also add an “anchor constraint”, which limits the maximum value of the raCBF. This helps us avoid a trivially optimal raCBF with large values but a small invariant set.

### Phase 1 (24)

$$\begin{aligned} \max_{\delta \in \mathbb{R}, s_0 \in \Sigma} r \\ \text{s.t. } \delta - x^\top P x - s_0 \phi \in \Sigma \end{aligned}$$

Another challenge posed by synthesis is the requirement for a feasible initialization. That is, we need to supply a raCBF already satisfying Eq. (7), (8) when using bilinear alternation. Generally, this initialization is something that can be computed straightforwardly, but has a trivially small invariant set. Then, the idea is for synthesis to enlarge the invariant set, while maintaining validity, until it has a reasonable size. We can easily provide such an initialization if the robust range  $\Theta$  is small. In that case, we can use the control Lyapunov function (CLF) corresponding to a linear quadratic regulator (LQR) which stabilizes to an equilibrium point in  $\mathcal{X}_{\text{safe}}$ . This initialization will intrinsically have some robustness to perturbation [31]. However, in the general case, the robust range  $\Theta$  will have moderate size.

To handle this, we consider a small robust range at initialization, so that we may use our LQR-based CLF initialization. Then, we maximize this robust range over the iterations (Phase 3). In our experiments, the ultimate robust range produced by our algorithm is always equal to the desired range. However, we have found, for example, that the algorithm is unable to increase the robust range to double the desired size. This is not a significant issue though, since robust control is generally not recommended when the desired range is large [3].

TABLE I

Results for Feedback Simulations			
	Toy 2D	Cartpole	Quadrotor
Safety Test (% safe trajectories)	100%	100%	100%
Performance Test (% improvement of raCBF over rCBF on task metric)	-	7.014%	54.953%

## VI. RESULTS

In this section, we synthesize raCBFs for three different systems: toy 2D, cartpole, and quadrotor. We show that our algorithm is quite general, producing valid raCBFs for a variety of dynamics and safety specifications. We also compare closed-loop performance in simulation for our robust-adaptive controller and a baseline robust controller [13]. We find that our robust-adaptive safety filter ensures 100% safety *while also* interfering less with the nominal controller. This means it achieves significantly better performance on combined safety-stabilization and safety-tracking tasks. Finally, we also show that our method scales better than the baseline, handling systems of 7D as opposed to 2D.

In particular, we run two sets of experiments with the simulated closed-loop system:

- 1) **Safety test:** ensure the system does not leave the invariant set under an adversarial nominal controller, which tries to drive the system out. Trials vary the initial state and the true parameter(s) value(s).
- 2) **Performance test:** measure stabilizing/tracking performance on combined safety-(stabilizing/tracking) task. Compare with baseline, a robust CBF [13]. Trials vary the true parameter(s) value(s). The cartpole's task

metric is time to reach the goal and the quadrotor's task metric is the average trajectory tracking error.

We use 200 random trials for each test. Results are listed in Table I and analyzed further in the following subsections. All the experimental hyperparameters can be found in the Appendix (Sec. VIII). To solve the SOSPs, we use SOSTOOLS [32] with Mosek [33] on a 3.00GHz Intel(R) Core(TM) i9-9980XE CPU with 16 cores and 126GB of RAM. For all examples, the initial raCBF is derived from an LQR-based CLF that stabilizes the system to the origin. Specifically, we use the solution  $S$  of the algebraic Riccati equation of the system linearized about the origin to define the CLF  $V(x) = x^T S x$  and the raCBF  $\phi^{(0)}(x) = \epsilon - V(x)$ , where  $\epsilon \in \mathbb{R}_{>0}$  is chosen experimentally.

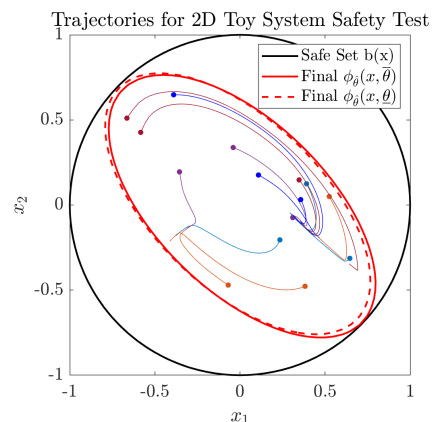


Fig. 2: Random trajectories generated by the safety test. Observe that all stay within the invariant set (in red).

### A. Toy 2D

We use the 2D toy system from [34]:

$$\dot{x} = \begin{bmatrix} x_2 - x_1^3 \\ 0 \end{bmatrix} - \begin{bmatrix} -x_1^2 \\ 0 \end{bmatrix} \theta + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \quad (25)$$

with  $\Theta \in [0.8, 1.5]$  and  $u \in [-2, 2]$ . The safety definition is  $\mathcal{X}_{\text{safe}} = \{x \mid b(x) := 1 - (x_1^2 + x_2^2) \geq 0\}$ .

Our synthesized raCBF attains 100% safety on this system. Observe in Fig. 2 that the trajectories under the safety-filtered adversarial controller never leave the system despite the parametric uncertainty.

### B. Cartpole

For the cartpole system (Fig. 1), we transform the standard trigonometric dynamics [35] to polynomial form by redefining the states as  $[\dot{x}, x, \psi, \sin(\psi), \cos(\psi) + 1]$ . For these new states, we require  $\sin^2(\psi) + \cos^2(\psi) = 1$ , which we incorporate into the SOSP using the S-procedure [36]. The unknown parameter is the friction coefficient at the joint. We let  $\Theta \in [0.28, 0.31]$ , following [37] and  $u \in [-5, 5]$  N. The safety definition confines the pole angle to be small:  $\mathcal{X}_{\text{safe}} := \{\sin^2(\frac{\pi}{3}) - \sin^2(\psi) \geq 0\}$ .

In Fig. 3, we plot the invariant set along  $(\theta, \dot{\theta})$ . First, observe that the shape of the set is sensible: it is an ellipsoid with primary axis  $\dot{\theta} = -\theta$ , meaning it considers the pole

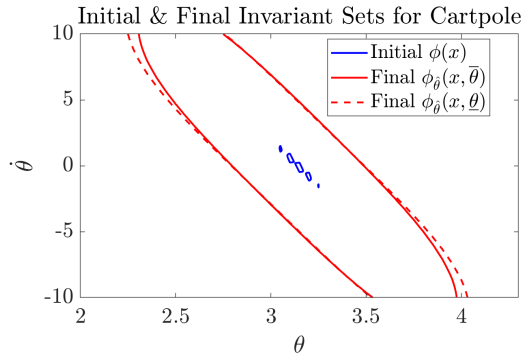


Fig. 3: Note the significant increase in size from the initial to final invariant sets. Also, observe that the final raCBF depends on  $\hat{\theta}$  in an intuitive way.

swinging to the vertical as safe and vice versa. Secondly, note that the synthesized raCBF depends meaningfully on  $\hat{\theta}$ . Comparing the invariant sets for  $\phi(x, \theta)$  (low joint friction) and  $\phi(x, \hat{\theta})$  (high joint friction), we can see that the former set is larger. This makes sense, since for low joint friction, the pole is more responsive and easily controlled. Finally, notice that the initial invariant is almost negligibly small, but the final set is quite large. We conclude that our algorithm is very effective at growing the invariant set.

As for the closed-loop simulations, raCBF attains 100% safety, even with an adversarially unsafe nominal controller. We also synthesize a robust CBF according to [13]. In the performance test, we implement a nominal controller which controls the cart to reach position  $x_{goal}$ . Then, the nominal controller is either layered with an raCBF-based or an rCBF-based safety filter, both of which prevent the pole from tipping.

Overall, the raCBF controller reaches the goal 7.014% faster. One reason for this is that the raCBF has a larger invariant set (Fig. 4), which allows larger pole angles during safe operation (Fig. 5) and hence larger cart acceleration. The difference in invariant set size can be explained as follows: while both raCBF and rCBF are robust to maximum estimation error, the raCBF is able to account for this error in a more moderate way, since it assumes parameter estimation is possible.

The second reason raCBF reaches the goal faster is its use of parameter estimation. The parameter adaptation law renders the raCBF *selectively conservative*, whereas the rCBF is always conservative (always operating under worst-case assumptions). To see this, observe Fig. 6. The law adapts the parameters toward values that increase conservatism when the state becomes increasingly unsafe. This happens at times  $\approx 3, 7, 9$  seconds, when the pole changes direction as the cart oscillates around the goal. This pivot causes the parameter estimate to grow to  $\hat{\theta}$ , which corresponds to the smallest and thus most conservative invariant set ( $\mathcal{I}_{\hat{\theta}}^r$ ) among the family of invariant sets  $\{\mathcal{I}_{\hat{\theta}}^r, \hat{\theta} \in \Theta\}$ . On the other hand, when the state is relatively safe, minimal conservatism is applied. This is the case when the pole is static. At those points, the parameter estimate is  $\hat{\theta} = 0$ , corresponding to the largest invariant set in the family of invariant sets.

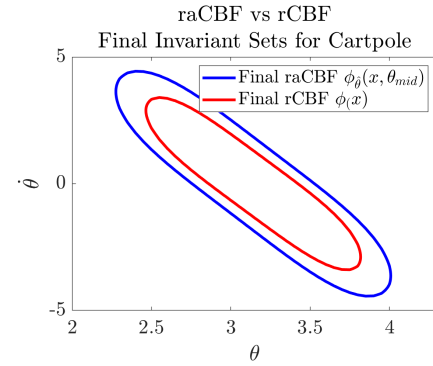


Fig. 4: Notice that raCBF's invariant set is much larger than the baseline's, which accounts for the difference in the performance test.

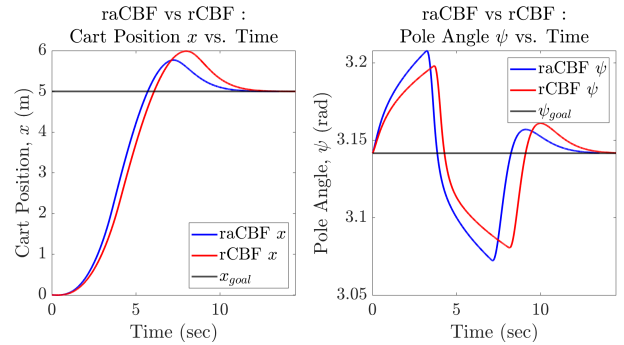


Fig. 5: Detailed analysis of one trial from cartpole's performance test.

### C. Quadrotor

This system is a planar quadrotor (Fig. 1) that we transform from the typical trigonometric dynamics [38] redefining the states as  $[\dot{x}, \dot{y}, \dot{\psi}, x, y, \sin(\psi), \cos(\psi) - 1]$ . We handle the transformation in much the same way as the cartpole (Sec. VI-B). The system has two unknown parameters, which are the drag coefficients in  $x$  and  $y$  that range from  $[0, 0.01]$ , roughly following the physical values from [39]. The propeller thrusts are limited to  $[-4, 4]$  mg. The safety definition enforces the system to avoid walls on the left and right:  $\mathcal{X}_{safe} = \{x \mid x \in [-1, 1]\}$ .

Our synthesis is able to handle a system of this moderate size (7D), producing an raCBF that ensures safety in 100%

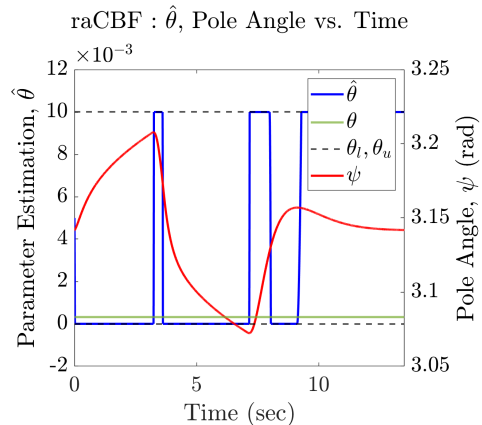


Fig. 6: Analysis of parameter estimation for cartpole's performance test.

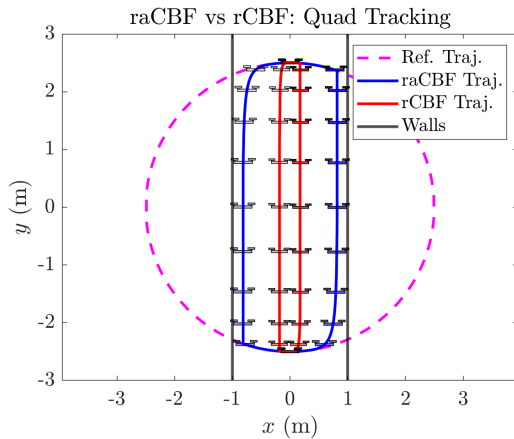


Fig. 7: raCBF enables closer trajectory tracking than rCBF and is therefore more performant.

the randomized trials of the safety test (Table I). The raCBF also has a large invariant set, which is evidenced by how it can track trajectories closely to the wall (Fig. 7). On the other hand, the robust CBF from [13] cannot be applied to a system of this size. However, in order to produce some point of comparison, we transfer the key parts of their formulation into our SOSP algorithm. Essentially, this just modifies the control filter feasibility constraints in phases 2 and 3 of our algorithm. This allows us to produce a result after six hours of optimization, which we use in the following performance test.

In the performance test, both controllers attempt to track a trajectory that crosses the walls. The nominal tracking controller is gain-scheduled LQR. As we can see, our raCBF offers an 54.95% improvement over the robust controller. In Fig. 7, we note that while our raCBF allows the quadrotor to approach the wall, the robust CBF, which has a much smaller invariant set in the  $x$  dimension, restricts the quadrotor to around  $x = 0$ . We can conclude that raCBF performs better and currently scales better than rCBF. More work will be required to get rCBF to scale moderately.

## VII. CONCLUSION

In this paper, we proposed the first verification and synthesis methods for adaptive-type safe controllers for uncertain systems. We proved that the constraints characterizing a valid raCBF can be written as an SOS program, and showed we can devise a multi-phase algorithm to synthesize an raCBF. Then, in our experiments, we illustrated that our algorithm is generic and scalable with three examples with varied dynamics, dimensions (up to 7D), and safety specifications. We empirically confirmed our theoretical safety guarantees by showing that 100% of trajectories under the safe controller are safe. Finally, we showed that our raCBF provides up to  $\approx 55\%$  task performance improvement over the baseline. In the future, we are interested in extending synthesis to systems with time-varying unknown parameters.

## ACKNOWLEDGEMENTS

The authors would like to thank Hongkai Dai at Toyota Research Institute and Weiye Zhao at Carnegie Mellon Uni-

versity for their technical guidance and paper revisions. This work was supported by the Qualcomm Innovation Fellowship awarded to Simin Liu and National Science Foundation under Grant No. 2144489.

## REFERENCES

- [1] A. Papachristodoulou and S. Prajna. “A tutorial on sum of squares techniques for systems analysis”. In: *Proceedings of the 2005, American Control Conference, 2005*. IEEE, 2005, pp. 2686–2700.
- [2] Z. Jarvis-Wloszek, R. Feeley, W. Tan, K. Sun, and A. Packard. “Some controls applications of sum of squares programming”. In: *42nd IEEE international conference on decision and control (IEEE Cat. No. 03CH37475)*. Vol. 5. IEEE, 2003, pp. 4676–4681.
- [3] E. Lavretsky and K. A. W. Wise. *Robust and Adaptive Control*. Advanced Textbooks in Control and Signal Processing. Springer London, 2012.
- [4] X. Xu, P. Tabuada, J. W. Grizzle, and A. D. Ames. “Robustness of control barrier functions for safety critical control”. In: *IFAC-PapersOnLine* 48.27 (2015), pp. 54–61.
- [5] M. Jankovic. “Robust control barrier functions for constrained stabilization of nonlinear systems”. In: *Automatica* 96 (2018), pp. 359–367.
- [6] J. Breeden and D. Panagou. “Robust control barrier functions under high relative degree and input constraints for satellite trajectories”. In: *arXiv preprint arXiv:2107.04094* (2021).
- [7] A. J. Taylor and A. D. Ames. “Adaptive safety with control barrier functions”. In: *2020 American Control Conference (ACC)*. IEEE, 2020, pp. 1399–1405.
- [8] M. Maghenem, A. J. Taylor, A. D. Ames, and R. G. Sanfelice. “Adaptive safety using control barrier functions and hybrid adaptation”. In: *2021 American Control Conference (ACC)*. IEEE, 2021, pp. 2418–2423.
- [9] A. Isaly, O. S. Patil, R. G. Sanfelice, and W. E. Dixon. “Adaptive safety with multiple barrier functions using integral concurrent learning”. In: *2021 American Control Conference (ACC)*. IEEE, 2021, pp. 3719–3724.
- [10] W. Zhao, T. He, T. Wei, S. Liu, and C. Liu. “Safety Index Synthesis via Sum-of-Squares Programming”. In: *arXiv preprint arXiv:2209.09134* (2022).
- [11] A. Clark. “A semi-algebraic framework for verification and synthesis of control barrier functions”. In: *arXiv preprint arXiv:2209.00081* (2022).
- [12] H. Dai and F. Permenter. “Convex synthesis and verification of control-Lyapunov and barrier functions with input constraints”. In: *2023 American Control Conference (ACC)*. IEEE, 2023, pp. 4116–4123.
- [13] S. Kang, Y. Chen, H. Yang, and M. Pavone. “Verification and synthesis of robust control barrier functions: multilevel polynomial optimization and semidefinite relaxation”. In: *arXiv preprint arXiv:2303.10081* (2023).
- [14] S. Liu, C. Liu, and J. Dolan. “Safe control under input limits with neural control barrier functions”. In: *Conference on Robot Learning*. PMLR, 2023, pp. 1970–1980.
- [15] T. Wei and C. Liu. “Safe control with neural network dynamic models”. In: *Learning for Dynamics and Control Conference*. PMLR, 2022, pp. 739–750.
- [16] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada. “Control barrier functions: Theory and applications”. In: *2019 18th European control conference (ECC)*. IEEE, 2019, pp. 3420–3431.
- [17] T. Wei and C. Liu. “Safe control algorithms using energy functions: A unified framework, benchmark, and new directions”. In: *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE, 2019, pp. 238–243.

- [18] T. Wei, S. Kang, W. Zhao, and C. Liu. “Persistently feasible robust safe control by safety index synthesis and convex semi-infinite programming”. In: *IEEE Control Systems Letters* 7 (2022), pp. 1213–1218.
- [19] V. Hamdipoor, N. Meskin, and C. G. Cassandras. “Safe Control Synthesis Using Environmentally Robust Control Barrier Functions”. In: *European Journal of Control* (2023), p. 100840.
- [20] K. Long, V. Dhiman, M. Leok, J. Cortés, and N. Atanasov. “Safe control synthesis with uncertain dynamics and constraints”. In: *IEEE Robotics and Automation Letters* 7.3 (2022), pp. 7295–7302.
- [21] W. Xiao, C. Belta, and C. G. Cassandras. “Adaptive control barrier functions”. In: *IEEE Transactions on Automatic Control* 67.5 (2021), pp. 2267–2281.
- [22] H. Parwana and D. Panagou. “Recursive feasibility guided optimal parameter adaptation of differential convex optimization policies for safety-critical systems”. In: *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 6807–6813.
- [23] Y. Lyu, W. Luo, and J. Dolan. “Probabilistic safety-assured adaptive merging control for autonomous vehicles”. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021.
- [24] S. Bansal, M. Chen, S. Herbert, and C. J. Tomlin. “Hamilton-jacobi reachability: A brief overview and recent advances”. In: *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE, 2017, pp. 2242–2253.
- [25] M. Althoff, O. Stursberg, and M. Buss. “Computing reachable sets of hybrid systems using a combination of zonotopes and polytopes”. In: *Nonlinear analysis: hybrid systems* 4.2 (2010), pp. 233–249.
- [26] M. Althoff. “Reachability analysis of nonlinear systems using conservative polynomialization and non-convex sets”. In: *Proceedings of the 16th international conference on Hybrid systems: computation and control*. 2013, pp. 173–182.
- [27] B. T. Lopez and J.-E. Slotine. “Unmatched control barrier functions: Certainty equivalence adaptive safety”. In: *2023 American Control Conference (ACC)*. IEEE, 2023, pp. 3662–3668.
- [28] P. A. Parrilo. *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. California Institute of Technology, 2000.
- [29] G. Stengle. “A nullstellensatz and a positivstellensatz in semialgebraic geometry”. In: *Mathematische Annalen* 207 (1974), pp. 87–97.
- [30] G. B. Dantzig. “Linear programming”. In: *Operations research* 50.1 (2002), pp. 42–47.
- [31] C. Chen. “On the robustness of the linear quadratic regulator via perturbation analysis of the Riccati equation”. PhD thesis. Dublin City University, 2015.
- [32] A. Papachristodoulou, J. Anderson, G. Valmorbida, S. Prajna, P. Seiler, P. A. Parrilo, M. M. Peet, and D. Jagt. *SOSTOOLS: Sum of squares optimization toolbox for MATLAB*. <http://arxiv.org/abs/1310.4716>, 2021. URL: <https://github.com/oxfordcontrol/SOSTOOLS>.
- [33] M. ApS. *The MOSEK optimization toolbox for MATLAB manual. Version 9.0*. 2019. URL: <http://docs.mosek.com/9.0/toolbox/index.html>.
- [34] S. Prajna, P. A. Parrilo, and A. Rantzer. “Nonlinear control synthesis by convex optimization”. In: *IEEE Transactions on Automatic Control* 49.2 (2004), pp. 310–314. DOI: 10.1109/TAC.2003.823000.
- [35] C. C. Chung and J. Hauser. “Nonlinear control of a swinging pendulum”. In: *Automatica* 31.6 (1995), pp. 851–862. ISSN: 0005-1098.
- [36] M. Posa, M. Tobenkin, and R. Tedrake. “Stability analysis and control of rigid-body systems with impacts and friction”. In: *IEEE Transactions on Automatic Control* 61.6 (2016), pp. 1423–1437. DOI: 10.1109/TAC.2015.2459151.
- [37] A. G. Barto, R. S. Sutton, and C. W. Anderson. “Neuronlike adaptive elements that can solve difficult learning control problems”. In: *IEEE Transactions on Systems, Man, and Cybernetics* SMC-13.5 (1983), pp. 834–846. DOI: 10.1109/TSMC.1983.6313077.
- [38] T. Luukkonen. “Modelling and control of quadcopter”. PhD thesis. School of Science at Aalto University, 2011.
- [39] P. Ventura Diaz and S. Yoon. “High-fidelity computational aerodynamics of multi-rotor unmanned aerial vehicles”. In: *2018 AIAA Aerospace Sciences Meeting*. 2018, p. 1266.

## VIII. APPENDIX

TABLE II

Computation Lengths for Synthesis			
	2D Toy	Cartpole	Quadrotor
Total Time (sec)	23	655	68,934
Total Iterations	11	159	337
Avg. P1 Time (sec)	0.1	0.12	0.87
Avg. P2 Time (sec)	1	2.8	74
Avg. P3 Time (sec)	0.8	1.2	130

TABLE III

Synthesis & Estimation Hyperparameters			
	2D Toy	Cartpole	Quadrotor
Deg. $\phi(x, \hat{\theta})$	3	3	3
$\beta^-$	-1.0	-0.5	-0.1
$\alpha$	0.1	0.01	0.01
$\gamma$	10.0	10.0	10.0
$\nu(\rho)$	$\text{atan}(\rho) + 1$	$\text{atan}(\rho) + 1$	$\text{atan}(\rho) + 1$
$\eta$	100	100	1000
$\sigma$	1	1	1
$\xi$	1	1	1
$\rho$ range	[0,10]	[0,10]	[0,10]

### A. System Parameters

- Cartpole:  $m_c = 1\text{kg}$ ,  $m_p = 0.1\text{kg}$ ,  $l = 0.5\text{m}$
- Quadrotor:  $m = 0.486\text{kg}$ ,  $r = 0.25\text{m}$ ,  $I = 0.00383\text{kg}\cdot\text{m}^2$

### B. Final raCBF

$$\mathbf{2D\ toy: } \phi(x, \hat{\theta}) = (0.3666)\hat{\theta}^2 + (0.07823)\hat{\theta}x_1 + (0.0007759)\hat{\theta}x_2 - (1.673)x_1^2 - (2.005)x_1x_2 - (1.726)x_2^2 - (0.8380)\hat{\theta} - (0.09071)x_1 - (0.0007912)x_2 + 1.098$$

$$\mathbf{Cartpole: } \phi(x, \hat{\theta}) = (23.24)\hat{\theta}^2 + (0.0002120)\hat{\theta}z_1 - (22.73)\hat{\theta}z_2 - (0.04065)x_3^2 + (1.136)x_3z_1 - (10.06)z_1^2 - (8.015)z_2^2 - (12.98)\hat{\theta} + (10.27)z_2 + 2.785$$

$$\mathbf{Quadrotor: } \phi(x, \hat{\theta}) = (-2.663)\hat{\theta}_1^2 - (0.06829)\hat{\theta}_1\hat{\theta}_2 + (0.05703)\hat{\theta}_1z_2 + (0.3199)\hat{\theta}_2^2 - (0.08092)\hat{\theta}_2z_2 - (0.7239)x_1^2 + (0.03167)x_1x_3 - (1.138)x_1x_4 + (3.074)x_1z_1 - (0.01136)x_3^2 + (0.02167)x_3x_4 - (0.1209)x_3z_1 - (1.462)x_4^2 + (2.118)x_4z_1 - (5.706)z_1^2 - (5.308)z_2^2 + (0.06482)z_2 + 0.9881$$