2024 IEEE International Conference on Robotics and Automation (ICRA)
May 13-17, 2024. Yokohama, Japan

Fault Tolerant Neural Control Barrier Functions for Robotic Systems under Sensor Faults and Attacks

Hongchao Zhang[1], Luyao Niu[2], Andrew Clark[1], and Radha Poovendran[2]

2024 IEEE International Conference on Robotics and Automation (ICRA) | 979-8-3503-8457-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICRA57147.2024.10610491

***Abstract*— Safety is a fundamental requirement of many robotic systems. Control barrier function (CBF)-based approaches have been proposed to guarantee the safety of robotic systems. However, the effectiveness of these approaches highly relies on the choice of CBFs. Inspired by the universal approximation power of neural networks, there is a growing trend toward representing CBFs using neural networks, leading to the notion of neural CBFs (NCBFs). Current NCBFs, however, are trained and deployed in benign environments, making them ineffective for scenarios where robotic systems experience sensor faults and attacks. In this paper, we study safety-critical control synthesis for robotic systems under sensor faults and attacks. Our main contribution is the development and synthesis of a new class of CBFs that we term fault tolerant neural control barrier function (FT-NCBF). We derive the necessary and sufficient conditions for FT-NCBFs to guarantee safety, and develop a data-driven method to learn FT-NCBFs by minimizing a loss function constructed using the derived conditions. Using the learned FT-NCBF, we synthesize a control input and formally prove the safety guarantee provided by our approach. We demonstrate our proposed approach using two case studies: obstacle avoidance problem for an autonomous mobile robot and spacecraft rendezvous problem, with code available via `https://github.com/HongchaoZhang-HZ/FTNCBF`.**

I. INTRODUCTION

Robotic systems are increasingly deployed in safety-critical applications such as search and rescue in hazardous environments [1]–[3]. Safety requirements are normally formulated as the positive invariance of given regions in the state space. Violations of safety could lead to catastrophic damage to robots, harm to humans that co-exist in the field of activities, and economic loss [4], [5]. A popular class of methods for safety-critical synthesis is control barrier function (CBF)-based approaches [6]–[9]. A CBF defines a positive invariant set within the safety region such that when the robot reaches the boundary of the set, the control input will steer the robot towards the interior of the set.

The performance and safety guarantees of CBF-based approaches are strongly dependent on the choice of barrier functions. Neural control barrier functions (NCBFs) [10]–[12], which represent CBFs using neural networks, have attracted increasing interest. Compared with the polynomial CBFs found by sum-of-squares (SOS) optimization [13]–[17], NCBFs leverage the universal approximation power of neural networks [18]–[20], and thus allow CBFs to be applied to high-dimensional complex systems [11], [21]–[25], e.g., learning-enabled robotic systems [26] and neural network dynamical models [27], [28].

At present, NCBFs [10]–[12], [29]–[31] are developed for robotic systems designed to operate in fault- and attack-free environments. Sensors mounted on robotic systems, however, have been shown to be vulnerable to a wide range of faults and malicious attacks [32], [33]. As demonstrated in our experiments, the safety guarantees of NCBFs do not hold when robots are operated in such adversarial environments.

In this paper, we study the problem of safety-critical control of robotic systems under sensor faults and attacks. We consider that an adversary can choose an arbitrary fault or attack pattern among finitely many choices, where each pattern corresponds to a distinct subset of compromised sensors. We propose a new class of CBFs called fault-tolerant NCBFs (FT-NCBFs). We present a data-driven method to learn FT-NCBFs. Given the learned FT-NCBFs, we then formulate a quadratic program to compute control inputs. The obtained control inputs guarantee that the robot moves towards the interior of the positive invariant set defined by the FT-NCBF under all attack patterns. Consequently, we can guarantee robot's safety by ensuring that the positive invariant set is contained within the safety region. To summarize, this paper makes the following contributions.

- We propose FT-NCBFs for robotic systems under sensor faults and attacks. We derive the necessary and sufficient conditions for FT-NCBFs to guarantee safety. Based on the derived conditions, we develop a data-driven method to learn FT-NCBFs.
- We develop a fault-tolerant framework which utilizes our proposed FT-NCBFs for safety-critical control synthesis. We prove that the synthesized control inputs guarantee safety under all fault and attack patterns.
- We evaluate our approach using two case studies on the obstacle avoidance problem of a mobile robot and the spacecraft rendezvous problem. We show that our approach guarantees the robot to satisfy the safety constraint regardless of the faults and attacks, whereas the baseline employing the existing NCBFs fails.

The rest of this paper is organized as follows. Section II presents preliminaries and problem formulation. We present our solution in Section III and demonstrate its safety guarantee in Section IV. Section V concludes the paper.

II. PRELIMINARIES AND PROBLEM FORMULATION

This section presents the system and adversary models. We then state the problem studied in this paper. We finally in-

[1]Hongchao Zhang and Andrew Clark are with the Electrical and Systems Engineering Department, McKelvey School of Engineering, Washington University in St. Louis, St. Louis, MO 63130 {hongchao, andrewclark}@wustl.edu

[2]Luyao Niu and Radha Poovendran are with the Network Security Lab, Department of Electrical and Computer Engineering, University of Washington, Seattle, WA 98195-2500 {luyaoniu,rp3}@uw.edu

troduce preliminary background on extended Kalman filters and stochastic control barrier functions.

A. System and Adversary Model

We consider a robotic system with state $x_t \in \mathcal{X} \subseteq \mathbb{R}^n$ and input $u_t \in \mathbb{R}^p$ at time t. The state dynamics and output $y_t \in \mathbb{R}^q$ are described by the stochastic differential equations

$$dx_t = (f(x_t) + g(x_t)u_t)\ dt + \sigma_t\ dW_t \tag{1}$$

$$dy_t = (cx_t + a_t)\ dt + \nu_t\ dV_t \tag{2}$$

where functions $f : \mathbb{R}^n \to \mathbb{R}^n$ and $g : \mathbb{R}^n \to \mathbb{R}^{n\times p}$ are locally Lipschitz, $\sigma_t \in \mathbb{R}^{n\times n}$, W_t is an n-dimensional Brownian motion. In addition, matrix $c \in \mathbb{R}^{q\times n}$, $\nu_t \in \mathbb{R}^{q\times q}$, and V_t is a q-dimensional Brownian motion. Here $a_t \in \mathbb{R}^q$ is an attack signal injected by an adversary, which will be detailed later in this section.

We define a control policy $\mu : \{y_{t'} : t' \in [0, t)\} \to \mathbb{R}^p$ to be a mapping from the sequence of outputs to a control input $u_t \in \mathbb{R}^p$ at each time t. The control policy needs to guarantee the robot to satisfy a safety constraint, which is specified as the positive invariance of a given safety region.

Definition 1 (Safety): A set $\mathcal{D} \subseteq \mathbb{R}^n$ is positive invariant under dynamics (1), (2) and control policy μ if $x_0 \in \mathcal{D}$ and $u_t = \mu(x_t)\ \forall t \geq 0$ imply that $x_t \in \mathcal{D}$ for all $t \geq 0$. If $\mathcal{D}$ is positive invariant, then the system satisfies the safety constraint with respect to $\mathcal{D}$.

We denote the safety region as

$$\mathcal{C} = \{x : h(x) \geq 0\}, \tag{3}$$

where $h : \mathbb{R}^n \to \mathbb{R}$ is locally Lipschitz. We further let the interior and boundary of $\mathcal{C}$ be $\text{int}(\mathcal{C}) = \{x : h(x) > 0\}$ and $\partial\mathcal{C} = \{x : h(x) = 0\}$, respectively. We assume that $x_0 \in \text{int}(\mathcal{C})$, i.e., the system is initially safe.

We consider the presence of an adversary, who can inject an arbitrary attack signal, denoted as $a_t \in \mathbb{R}^q$, to manipulate the output at each time t. The adversary aims to force the robot leaves the safety region $\mathcal{C}$. The attack signal is constrained by $\text{supp}(a_t) \subseteq \mathcal{F}(r)$, where $r \in \{r_1, \ldots, r_m\}$ is the index of possible faults or attacks, m is the total number of possible attack patterns, and $\mathcal{F}(r) \subseteq \{1, \ldots, q\}$ denotes the set of potentially compromised sensors under attack pattern r. Hence, if attack pattern r occurs, then the outputs of any of the sensors in $\mathcal{F}(r_i)$ can be arbitrarily modified. We consider that the set of possible faults or attacks is known, but the exact attack pattern that has occurred is unknown to the controller. In this paper, we make the following assumption.

Assumption 1: The robot in Eq. (1)-(2) and the attack patterns $\mathcal{F}(r_1), \ldots, \mathcal{F}(r_m)$ satisfy the conditions: (i) The system is controllable, and (ii) For each $i, j \in \{1, \ldots, m\}$, the pair $[\frac{\partial \overline{f}}{\partial x}(x, u), \overline{c}_{i,j}]$ is uniformly detectable, where $\overline{c}_{i,j}$ is the corresponding matrix after removing sensors affected by r_i and r_j from matrix c.

We state the problem studied in this paper as follows.

Problem 1: Given a safety region $\mathcal{C}$ defined in Eq. (3) and a parameter $\epsilon \in (0, 1)$, construct a control policy μ such that, for any attack pattern $r \in \{r_1, \ldots, r_m\}$, the probability $Pr(x_t \in \mathcal{C}\ \forall t) \geq (1 - \epsilon)$ when attack pattern r occurs.

B. Preliminaries

The extended Kalman filter (EKF) [34] can be used to estimate the robot's state [35], [36]. We denote the state as $\hat{x}_t$ and let the updated estimate be as follows:

$$d\hat{x}_t = (f(\hat{x}_t) + g(\hat{x}_t)u_t)\ dt + K_t(dy_t - c\hat{x}_t), \tag{4}$$

where $K_t = P_t c^T R_t^{-1}$ is the Kalman gain, $R_t = \nu_t\nu_t^T$, and $\hat{x}_t$ is the state estimate. Positive definite matrix P_t is the solution to $\frac{dP}{dt} = A_tP_t + P_tA_t^T + Q_t - P_tc^TR_t^{-1}cP_t$, where $Q_t = \sigma_t\sigma_t^T$, $A_t = \frac{\partial \bar{f}}{\partial x}(\hat{x}_t, u_t)$, and $\bar{f}(x, u) = f(x) + g(x)u$. We make the following assumption in this paper.

Assumption 2: When $a_t = 0$, the SDEs (1)-(2) satisfy:

1) There exist constants β_1 and β_2 such that $\mathbf{E}(\sigma_t\sigma_t^T) \geq \beta_1 I$ and $\mathbf{E}(\nu_t\nu_t^T) \geq \beta_2 I$ for all t.
2) The pair $[\frac{\partial \bar{f}}{\partial x}(x, u), c]$ is uniformly detectable.
3) Let ϕ be defined by $\overline{f}(x, u) - \overline{f}(\hat{x}, u) = \frac{\partial \overline{f}}{\partial x}(x - \hat{x}) + \phi(x, \hat{x}, u)$. Then there exist real numbers k_ϕ and ϵ_ϕ such that $\|\phi(x, \hat{x}, u)\| \leq k_\phi\|x - \hat{x}\|_2^2$ for all x and $\hat{x}$ satisfying $\|x - \hat{x}\|_2 \leq \epsilon_\phi$.

The accuracy of of EKF is given by the theorem below.

Theorem 1 ([34]): Suppose that Assumption 2 holds. Then there exists $\delta > 0$ such that $\sigma_t\sigma_t^T \leq \delta I$ and $\nu_t\nu_t^T \leq \delta I$. For any $0 < \epsilon < 1$, there exists $\gamma > 0$ such that

$$Pr\left(\sup_{t\geq 0} ||x_t - \hat{x}_t||_2 \leq \gamma\right) \geq 1 - \epsilon.$$

We finally introduce stochastic control barrier functions for robots in the absence of a_t, along with its safety guarantee.

Theorem 2 ([37]): Suppose that $a_t = 0$. For the robot in Eq. (1)-(2) with safety region defined by Eq. (3), define

$$\overline{b}^\gamma = \sup_{x, x^0} \{b(\hat{x}) : ||x - x^0||_2 \leq \gamma \text{ and } b(x^0) = 0\}.$$

Let $\hat{b}^\gamma(\hat{x}) := b(\hat{x}) - \overline{b}^\gamma$ and $\hat{x}_t$ denote the EKF estimate of x_t. Suppose that there exists a constant $\delta > 0$ such that whenever $\hat{b}(\hat{x}_t) < \delta$, u_t is chosen to satisfy

$$\frac{\partial b}{\partial x}(\hat{x}_t)\overline{f}(\hat{x}_t, u_t) - \gamma\|\frac{\partial b}{\partial x}(\hat{x}_t)K_tc\|_2 + \frac{1}{2}\mathbf{tr}\left(\nu_t^TK_t^T\frac{\partial^2 b}{\partial x^2}(\hat{x}_t)K_t\nu_t\right) \geq -\hat{b}^\gamma(\hat{x}_t). \tag{5}$$

Then $Pr(x_t \in \mathcal{D}\ \forall t|\ \|x_t - \hat{x}_t\|_2 \leq \gamma\ \forall t) = 1$, where $\mathcal{D} = \{x \mid b(x) \geq 0\}$.

The function b satisfying inequality (5) is a stochastic control barrier function. It specifies that as the state approaches the boundary, the control input is chosen such that the rate of increase of the barrier function decreases to zero. Hence Theorem 2 implies that if there exists a stochastic control barrier function for a system, then the safety condition is satisfied with probability $(1-\epsilon)$ when an EKF is used as an estimator and the control input is chosen to satisfy Eq. (5).

III. Solution Approach to Safety-Critical Control and Synthesis of Fault Tolerant NCBF

In this section, we first present an overview of our solution to safety-critical control synthesis for the robot in Eq. (1)-(2) such that safety can be guaranteed under sensor faults and attacks. The key to our approach is the development of a new class of control barrier functions named *fault tolerant neural control barrier functions (FT-NCBFs)*.

A. Overview of Proposed Solution

This subsection presents our proposed solution approach to safety-critical control synthesis. Since the attack pattern is unknown, we maintain a set of m EKFs, where each EKF uses measurements from $\{1,\dots,q\} \setminus \mathcal{F}(r_i)$ for each $i \in \{1,\dots,m\}$. We denote the state estimates and Kalman gain obtained using $\{1,\dots,q\} \setminus \mathcal{F}(r_i)$ as $\hat{x}_{t,i}$ and $K_{t,i}$, respectively. If there exists a function b_θ parameterized by θ such that $\mathcal{D}_\theta = \{\hat{x} | b_\theta(\hat{x}) \geq 0\} \subseteq \mathcal{C}$, then Theorem 2 indicates that any control input u within the feasible region

$$\Omega_i = \{u : \frac{\partial b_\theta}{\partial x} f(\hat{x}_{t,i}) + \frac{\partial b_\theta}{\partial x} g(\hat{x}_{t,i}) u - \gamma_i \| \frac{\partial b_\theta}{\partial x}(\hat{x}) K_{t,i} c_i \|_2 + \frac{1}{2}\mathbf{tr}\left(\nu_i^T K_{t,i}^T \frac{\partial^2 b_\theta}{\partial x^2}(\hat{x}_{t,i}) K_{t,i} \nu_i\right) + \hat{b}_\theta^{\gamma_i}(\hat{x}_{t,i}) \geq 0\},$$

guarantees safety under attack pattern r_i, where c_i is obtained by removing rows corresponding to $\mathcal{F}(r_i)$ from matrix c, $\hat{b}_\theta^{\gamma_i}(\hat{x}) = b_\theta(\hat{x}) - \overline{b}_\theta^{\gamma_i}(\hat{x})$, and

$$\overline{b}_\theta^{\gamma_i} = \sup_{\hat{x},\hat{x}^0} \left\{ b_\theta(\hat{x}) : ||\hat{x} - \hat{x}^0||_2 \leq \gamma_i \text{ and } b_\theta(\hat{x}^0) = 0 \right\}.$$

If there exists a control input $u \in \cap_{i=1}^m \Omega_i \neq \emptyset$, such a control input can guarantee the safety under any attack pattern r_i.

We note that the existence of a control input u satisfying the constraints specified by $\Omega_1,\dots,\Omega_m$ simultaneously may not be guaranteed because sensor faults and attacks can significantly bias the state estimates. Thus we develop a mechanism to identify constraints conflicting with each other, and resolve such conflicts. Our idea is to additionally maintain $\binom{m}{2}$ EKFs, where each EKF computes state estimates using sensors from $\{1,\dots,q\} \setminus (\mathcal{F}(r_i) \cup \mathcal{F}(r_j))$ for all $i \neq j$. We use a variable Z_t to keep track of the attack patterns that will not raise conflicts. The variable Z_t is initialized as $\{1,\dots,m\}$. If $\cap_{i \in Z_t} \Omega_i = \emptyset$, we compare state estimates $\hat{x}_{t,i}$ with $\hat{x}_{t,j}$ for all $i,j \in Z_t$ and $i \neq j$. If $\|\hat{x}_{t,i} - \hat{x}_{t,j}\|_2 \geq \alpha_{ij}$ for some chosen parameter $\alpha_{ij} > 0$, then Z_t is updated as

$$Z_t = \begin{cases} Z_t \setminus \{i\}, & \text{if } \|\hat{x}_{t,i} - \hat{x}_{t,i,j}\|_2 \geq \alpha_{ij}/2 \\ Z_t \setminus \{j\}, & \text{if } \|\hat{x}_{t,j} - \hat{x}_{t,i,j}\|_2 \geq \alpha_{ij}/2 \end{cases}.$$

After updating Z_t, if $\cap_{i \in Z_t} \Omega_i \neq \emptyset$, then control input u_t can be chosen as

$$\min_{u_t \in \cap_{i \in Z_t} \Omega_i} \; u_t^T u_t. \tag{6}$$

Otherwise, we will remove indices i corresponding to attack pattern r_i causing largest residue $y_{t,i} - c_i \hat{x}_{t,i}$ until $\cap_{i \in Z_t} \Omega_i \neq \emptyset$. Here $y_{t,i}$ is the output from sensors in $\{1,\dots,q\} \setminus \mathcal{F}(r_i)$.

The positive invariance of set $\mathcal{D}_\theta$ using the procedure described above is established in the following theorem.

Theorem 3 ([37]): Suppose $\gamma_1,\dots,\gamma_m$, and α_{ij} for $i < j$ are chosen such that the following conditions are satisfied:

1) Define $\Lambda_i(\hat{x}_{t,i}) = \frac{\partial b_\theta}{\partial x} g(\hat{x}_{t,i})$. There exists $\delta > 0$ such that for any $X_t' \subseteq X_t(\delta) := \{i \mid \hat{b}_\theta^{\gamma_i}(\hat{x}_{t,i}) < \delta\}$ satisfying $||\hat{x}_{t,i} - \hat{x}_{t,j}||_2 \leq \alpha_{ij}$ for all $i,j \in X_t'$, there exists u such that
$$\Lambda_i(\hat{x}_{t,i}) u > -\frac{\partial b_\theta}{\partial x} f(\hat{x}_{t,i}) + \gamma_i \| \frac{\partial b_\theta}{\partial x}(\hat{x}_{t,i}) K_{t,i} c_i \|_2 - \frac{1}{2}\mathbf{tr}\left(\nu_i^T K_{t,i}^T \frac{\partial^2 b_\theta}{\partial x^2}(\hat{x}) K_{t,i} \nu_i\right) - \hat{b}_\theta^{\gamma_i}(\hat{x}_{t,i}) \tag{7}$$
for all $i \in X_t'$.
2) For each i, when $r = r_i$,
$$Pr(\|\hat{x}_{t,i} - \hat{x}_{t,i,j}\|_2 \leq \alpha_{ij}/2 \; \forall j, \|\hat{x}_{t,i} - x_t\|_2 \leq \gamma_i \; \forall t) \geq 1 - \epsilon. \tag{8}$$

Then $Pr(x_t \in \mathcal{D}_\theta \; \forall t) \geq 1 - \epsilon$ for any $r \in \{r_1,\dots,r_m\}$.

Based on Theorem 3, we note that the key to our solution approach is to find the function b_θ. We name the function b_θ as *fault tolerant neural control barrier function (FT-NCBF)*, whose definition is given as below.

Definition 2: A function b_θ parameterized by θ is a fault tolerant neural control barrier function for the robot in Eq. (1)-(2) if it there exists a control input u satisfying Eq. (7) under the conditions in Theorem 3.

Solving Problem 1 hinges on the task of synthesizing an FT-NCBF for the robot in (1)-(2), which will be our focus in the remainder of this section. Specifically, we first investigate how to synthesize NCBFs when there exists no adversary (Section III-B). We then use the NCBFs as a building block, and present how to synthesize FT-NCBFs. We construct a loss function to learn FT-NCBFs in Section III-C. We establish the safety guarantee of our approach in Section III-D.

B. Synthesis of NCBF

In this subsection, we describe how to synthesize NCBFs. We first present the necessary and sufficient conditions for stochastic control barrier functions, among which NCBFs constitute a special class represented by neural networks.

Proposition 1: Suppose Assumption 2 holds. The function $b(\hat{x})$ is a stochastic control barrier function if and only if there is no $\hat{x} \in \mathcal{D}^\gamma := \{\hat{x} \mid \hat{b}(\hat{x}) \geq 0\}$, satisfying $\frac{\partial b}{\partial x} g(\hat{x}) = 0$ and $\xi^\gamma(\hat{x}) < 0$, where

$$\xi^\gamma(\hat{x}) = \frac{\partial b}{\partial x} f(\hat{x}) + \frac{1}{2}\mathbf{tr}\left(\nu^T K_t^T \frac{\partial^2 b}{\partial x^2}(\hat{x}) K_t \nu\right) - \gamma || \frac{\partial b}{\partial x}(\hat{x}) K_t c ||_2 + \hat{b}^\gamma(\hat{x}). \tag{9}$$

The proposition is based on [38, Proposition 2]. We omit the proof due to space constraint. We note that the class of NCBFs is a special subset of stochastic control barrier functions. We denote the NCBF as $b_\theta(\hat{x})$, where θ is the parameter of the neural network representing the function.

In the following, we introduce the concept of *valid* NCBFs, and present how to synthesize them. A valid NCBF needs to satisfy the following two properties.

Definition 3 (Correct NCBFs): Given a safety region $\mathcal{C}$, the NCBF b_θ is correct if and only if $\mathcal{D}_\theta \subseteq \mathcal{C}$.

The correctness property requires the NCBF b_θ to induce a set $\mathcal{D}_\theta \subseteq \mathcal{C}$. If $\mathcal{D}_\theta$ is positive invariant, then $\mathcal{C}$ is also positive invariant, ensuring the robot to be safe with respect to $\mathcal{C}$. We next give the second property of valid NCBFs.

Definition 4 (Feasible NCBF): The NCBF b_θ parameterized by θ is feasible if and only if $\forall \hat{x} \in \mathcal{D}_\theta^\gamma := \{\hat{x} | \hat{b}_\theta^\gamma(\hat{x}) \geq 0\}$, there exists u such that $\xi_\theta^\gamma(\hat{x}) + \frac{\partial b_\theta}{\partial x} g(\hat{x}) u \geq 0$, where

$$\xi_\theta^\gamma(\hat{x}) = \frac{\partial b_\theta}{\partial x} f(\hat{x}) + \frac{1}{2}\mathbf{tr}\left(\nu^T K_t^T \frac{\partial^2 b_\theta}{\partial x^2}(\hat{x}) K_t \nu\right) - \gamma \| \frac{\partial b_\theta}{\partial x}(\hat{x}) K_t c \|_2 + \hat{b}_\theta^\gamma(\hat{x}). \quad (10)$$

The feasibility property in Definition 4 ensures that a control input u can always be found to satisfy the inequality (5), and hence can guarantee safety.

We note that there may exist infinitely many valid NCBFs. In this work, we focus on synthesizing valid NCBFs that encompass the largest possible safety region. To this end, we define an operator $Vol(\mathcal{D}_\theta)$ to represent the volume of the set $\mathcal{D}_\theta$, and synthesize a valid NCBF such that $Vol(\mathcal{D}_\theta)$ is maximized. The optimization program is given as follows

$$\max_\theta \quad Vol(\mathcal{D}_\theta) \quad (11)$$

$$s.t. \quad \xi_\theta^\gamma(\hat{x}) \geq 0 \quad \forall \hat{x} \in \partial \mathcal{D}_\theta^\gamma \quad (12)$$

$$b_\theta(\hat{x}) \leq h(\hat{x}) \quad \forall \hat{x} \in \mathcal{X} \backslash \mathcal{D}_\theta \quad (13)$$

where $\partial \mathcal{D}_\theta^\gamma$ represents the boundary of set $\mathcal{D}_\theta^\gamma$. Here constraints (12) and (13) require parameter θ to define feasible and correct NCBFs, respectively. Solving the constrained optimization problem is challenging. In this work, we convert the constrained optimization to an unconstrained one by constructing a loss function which penalizes violations of the constraints. We then minimize the loss function over a training dataset to learn parameters θ and thus NCBF b_θ.

We denote the training dataset as $\mathcal{T} := \{\hat{x}_1, \ldots, \hat{x}_N \mid \hat{x}_i \in \mathcal{X}, \forall i = 1, \ldots, N\}$, where N is the number of samples. The dataset $\mathcal{T}$ is generated by simulating estimates with fixed point sampling as in [11]. We first uniformly discretize the state space into cells with length vector L. Next, we uniformly sample the center of discretized cell as fixed points x_f. Then we simulate the estimates by introducing a perturbation $\rho[j]$ sampled uniformly from interval $[x_f[j] - 0.5L[j], x_f[j] + 0.5L[j]]$. Finally, we have the sampling data $\hat{x}_i = x_f + \rho \in \mathcal{T} \subseteq \mathcal{X}$.

We then formulate the following unconstrained optimization problem to search for θ

$$\min_\theta \quad -Vol(\mathcal{D}_\theta) + \lambda_f \mathcal{L}_f(\mathcal{T}) + \lambda_c \mathcal{L}_c(\mathcal{T}) \quad (14)$$

where $\mathcal{L}_f(\mathcal{T})$ is the loss penalizing the violations of constraint (12), $\mathcal{L}_c(\mathcal{T})$ penalizes the violations of constraint (13), and λ_f and λ_c are non-negative coefficients. The objective function (11) is approximated by the following quantity

$$Vol(\mathcal{D}_\theta) = \sum_{\hat{x} \in \mathcal{T}} -ReLU(h(\hat{x})) ReLU(-b_\theta(\hat{x})). \quad (15)$$

Eq. (15) penalizes the samples $\hat{x}$ in the safety region but not in $\mathcal{D}_\theta$, i.e., $h(\hat{x}) > 0$ and $b_\theta(\hat{x}) < 0$. The penalty of violating the feasibility property in Eq. (12) is defined as

$$\mathcal{L}_f(\mathcal{T}) = \sum_{\hat{x} \in \mathcal{T}} -\Delta(\hat{x}) ReLU(-\xi_\theta^\gamma(\hat{x}) - \frac{\partial b_\theta}{\partial x} g(\hat{x}) u + \hat{b}_\theta^\gamma(\hat{x})),$$

where $\Delta(\hat{x})$ is an indicator function such that $\Delta(\hat{x}) := 1$ if $b_\theta(\hat{x}) = \overline{b}_\theta^\gamma$ and $\Delta(\hat{x}) := 0$ otherwise. The function Δ allows us to find and penalize sample points $\hat{x}$ satisfying $\hat{b}_\theta^\gamma(\hat{x}) = 0$ and $\xi_\theta^\gamma(\hat{x}) + \frac{\partial b_\theta}{\partial x} g(\hat{x}) u < 0$. For each sample $\hat{x} \in \mathcal{T}$, the control input u in $\mathcal{L}_f$ is computed as follows

$$\begin{aligned} \min_u \quad & u^T u \\ s.t. \quad & \xi_\theta^\gamma(\hat{x}) + \frac{\partial b_\theta}{\partial x} g(\hat{x}) u \geq 0 \end{aligned} \quad (16)$$

The loss function to penalize the violations of the correctness property in Eq. (13) is constructed as

$$\mathcal{L}_c(\mathcal{T}) = \sum_{\hat{x} \in \mathcal{T}} ReLU(-h(\hat{x})) ReLU(b_\theta(\hat{x})) \quad (17)$$

Eq. (17) penalizes $\hat{x}$ outside the safety region but being regarded safe, i.e., $h(\hat{x}) < 0$ and $b_\theta(\hat{x}) > 0$. When $\mathcal{L}_c(\mathcal{T})$ and $\mathcal{L}_f(\mathcal{T})$ converge to 0, constraints (12)-(13) are satisfied.

C. Synthesis of FT-NCBF

In Section III-B, we presented the training of NCBFs when there exists no adversary. In this subsection, we generalize the construction of the loss function in Eq. (14), and present how to train a valid FT-NCBF for robotic systems in Eq. (1)-(2) under unknown attack patterns. With a slight abuse of notations, we use b_θ to denote the FT-NCBF in the remainder of this paper. We define $\hat{b}_\theta^{\gamma_i}(\hat{x}) = b_\theta(\hat{x}) - \overline{b}_\theta^{\gamma_i}(\hat{x})$, where

$$\overline{b}_\theta^{\gamma_i} = \sup_{\hat{x}, \hat{x}^0} \left\{ b_\theta(\hat{x}) : ||\hat{x} - \hat{x}^0||_2 \leq \gamma_i \quad \text{and } b_\theta(\hat{x}^0) = 0 \right\}.$$

The following proposition gives the necessary and sufficient conditions for a function b_θ to be an FT-NCBF.

Proposition 2: Suppose Assumption 2 holds. The function $b_\theta(\hat{x})$ is an FT-NCBF if and only if there is no $\hat{x}_{t,i} \in \mathcal{D}_\theta^{\gamma_i}$, satisfying $\frac{\partial b_\theta}{\partial x} g(\hat{x}_{t,i}) = 0$, $\xi_\theta^{\gamma_i}(\hat{x}_{t,i}) < 0$ for all $i \in \{1, \ldots, m\}$ where

$$\xi_\theta^{\gamma_i}(\hat{x}_{t,i}) = \frac{\partial b_\theta}{\partial x} f(\hat{x}_{t,i}) + \frac{1}{2}\mathbf{tr}\left(\nu_i^T K_{t,i}^T \frac{\partial^2 b_\theta}{\partial x^2}(\hat{x}) K_{t,i} \nu_i\right) - \gamma_i \| \frac{\partial b_\theta}{\partial x}(\hat{x}_{t,i}) K_{t,i} c_i \|_2 + \hat{b}_\theta^{\gamma_i}(\hat{x}_{t,i}). \quad (18)$$

The proposition can be proved using the similar idea to Proposition 1. We omit the proof due to space constraint.

We construct the loss function below to learn FT-NCBFs

$$\min_\theta \quad -Vol(\mathcal{D}_\theta) + \lambda_f \sum_{i \in \{1, \ldots, m\}} \mathcal{L}_f^i(\mathcal{T}) + \lambda_c \mathcal{L}_c(\mathcal{T}), \quad (19)$$

where $\mathcal{L}_f(\mathcal{T}) = \sum_{i \in \{1, \ldots, m\}} \mathcal{L}_f^i(\mathcal{T})$ is the penalty of violating the feasibility property,

$$\mathcal{L}_f^i(\mathcal{T}) = \sum_{\hat{x} \in \mathcal{T}} -\overline{\Delta}_i(\hat{x}) ReLU(-\xi_\theta^{\gamma_i}(\hat{x}) - \frac{\partial b_\theta}{\partial x} g(\hat{x}) u + \hat{b}_\theta^{\gamma_i}(\hat{x})),$$

and $\overline{\Delta}(\hat{x})$ is an indicator function such that $\overline{\Delta}(\hat{x}) := 1$ if $b_\theta(\hat{x}) \leq \max_{i \in Z_t} \{\overline{b}_\theta^{\gamma_i}\}$ and $\overline{\Delta}(\hat{x}) := 0$ otherwise. The control input u used to compute $\mathcal{L}_f^i(\mathcal{T})$ for each sample $\hat{x}$ is calculated as follows.

$$\begin{aligned} \min_{u} \quad & u^T u \\ s.t. \quad & \xi_\theta^{\gamma_i}(\hat{x}) + \frac{\partial b_\theta}{\partial x} g(\hat{x}) u \geq 0 \quad \forall i \in \{1, \ldots, m\} \end{aligned} \tag{20}$$

If $\mathcal{L}_c(\mathcal{T})$ and $\mathcal{L}_f(\mathcal{T})$ converge to 0, b_θ is a valid FT-NCBF.

D. Safety Guarantee of Proposed Approach

In this subsection, we establish the safety guarantee of our approach for the robot in Eq. (1)-(2). First, we note that Theorem 3 establishes the positive invariance of set $\mathcal{D}_\theta$. However, the theorem depends on the existence of u_t. The following proposition provides the sufficient condition of the existence of u_t for all $\hat{x} \in \mathcal{D}_\theta^{\gamma_i},\ \forall i \in Z_t$.

Proposition 3: Suppose that the interval length L used to sample the training dataset $\mathcal{T}$ satisfies $L \leq s$ and $s \to 0$. If an FT-NCBF b_θ satisfies $\mathcal{L}_f(\mathcal{T}) + \mathcal{L}_c(\mathcal{T}) = 0$, then there always exists u_t such that $\frac{\partial b_\theta}{\partial x} g(\hat{x}) u_t + \xi_\theta^{\gamma_i}(\hat{x}) \geq 0\ \forall \hat{x} \in \mathcal{D}_\theta^{\gamma_i},\ \forall i \in Z_t$.

Proof: By the constructions of $\mathcal{L}_f^i$ and $\mathcal{L}_c$, these losses are non-negative. Thus if $\mathcal{L}_f(\mathcal{T}) + \mathcal{L}_c(\mathcal{T}) = 0$, we have $\mathcal{L}_f^i(\mathcal{T}) = \mathcal{L}_f(\mathcal{T}) = \mathcal{L}_c = 0$. According to the definitions of $\mathcal{L}_f$ and $\mathcal{L}_f^i$ as well as the conditions that $L \leq s$ and $s \to 0$, we then have that there must exist some control input u that solves the optimization program in Eq. (20) for all $\hat{x} \in \mathcal{D}_\theta^{\gamma_i}$ when $\mathcal{L}_f^i(\mathcal{T}) + \mathcal{L}_f(\mathcal{T}) = 0$. Otherwise losses $\mathcal{L}_f^i$ and $\mathcal{L}_f$ will be positive. ■

We finally present the safety guarantee of our approach.

Theorem 4: Suppose that the interval length L used to sample the training dataset $\mathcal{T}$ satisfies $L \leq s$ and $s \to 0$. Let b_θ be an FT-NCBF satisfying $\mathcal{L}_f(\mathcal{T}) + \mathcal{L}_c(\mathcal{T}) = 0$. Suppose $\gamma_1, \ldots, \gamma_m$, and α_{ij} for $i < j$ are chosen such that the conditions in Theorem 3 hold. Then $Pr(x_t \in \mathcal{C}\ \forall t) \geq 1 - \epsilon$ for any attack pattern $r \in \{r_1, \ldots, r_m\}$.

Proof: The theorem follows from Theorem 3, Proposition 3, and the correctness property that $\mathcal{D}_\theta \subseteq \mathcal{C}$. ■

IV. Experiments

In this section, we evaluate our proposed approach using two case studies, namely the obstacle avoidance of an autonomous mobile robot [39] and the spacecraft rendezvous problem [40]. Both case studies are conducted on a laptop with an AMD Ryzen 5800H CPU and 32GB RAM. The hyper-parameters in both studies can be found in our code.

A. Obstacle Avoidance Problem of Mobile Robot

We consider an autonomous mobile robot navigating on a road following the dynamics [41] given below

$$\dot{x} = f(x) + g(x)u,$$

where $x := [x_1, x_2, \psi]^T \in \mathcal{X} \subseteq \mathbb{R}^3$ is the state consisting of the location (x_1, x_2) of the robot and its orientation ψ, u is the input that controls the robot's orientation, $f(x) = [\sin\psi, \cos\psi, 0]^T$, and $g(x) = [0, 0, 1]^T$.

The mobile robot is required to stay in the road while avoid pedestrians sharing the field of activities. We set the location of the pedestrian as $(0, 0)$. Then the safety region is formulated as $\mathcal{C} = \{\mathbf{x} \in \mathcal{X} : x_1^2 + x_2^2 \geq 0.04,\ \text{and}\ x_2 \geq -0.3\}$, where $\mathcal{X} = [-2, 2]^3$. We consider that one IMU and two GNSS sensors are mounted on the mobile robot. These sensors jointly yield the output model $y = [x_1, x_1, x_2, x_2, \psi]^T + \nu$, where the measurement noise $\nu \sim \mathcal{N}(0, \sigma I_5) \in \mathbb{R}^5$, $\sigma = 0.001$, and I_5 is the five-dimensional identity matrix. There exists an adversary who can spoof the readings from one GNSS sensor, leading to two possible attack patterns, $\{r_1, r_2\}$. The compromised sensors associated with attack patterns r_1 and r_2 are the second or fourth dimension of y, denoted as $y[2]$ and $y[4]$, respectively.

We compare our approach with a baseline which adopts the method from [11] and learns an NCBF ignoring the presence of sensor faults and attacks. The baseline computes the control input by solving $\min_{u \in \bar{\Omega}} u^T u$, where $\bar{\Omega}$ is the feasible region specified by the learned NCBF.

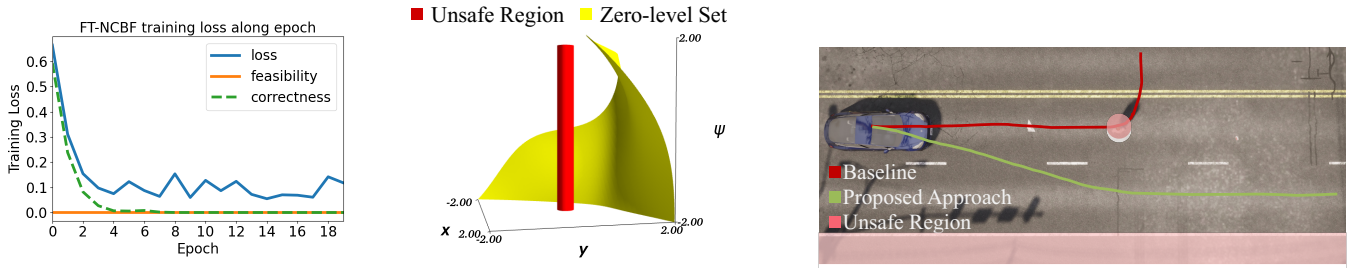
When applying our approach, we first sample the training dataset $\mathcal{T}$ with $L = 0.125$, making $|\mathcal{T}| = 32^3$. Given the training dataset $\mathcal{T}$, we learn an FT-NCBF using Eq. (19) with $\gamma_1 = 0.002$ and $\gamma_2 = 0.0015$. The training process took about 604 seconds. The values of loss function, $L_f(\mathcal{T})$, and $\mathcal{L}_c(\mathcal{T})$ at each epoch during training are presented in Fig. 1a. We observe that the loss function decreases towards zero during the training process. In particular, $L_f(\mathcal{T}) + \mathcal{L}_c(\mathcal{T}) \to 0$ as we train more epochs. By the construction of the loss function, it indicates that our approach finds a valid FT-NCBF. The positive invariant set $\mathcal{D}_\theta$ induced by the learned FT-NCBF is shown in Fig. 1b. We observe that the zero-level set $\partial\mathcal{D}_\theta$ in yellow color stays close to the boundary of the safety region, while it does not overlap with the unsafe region in red color. We implement the control policy calculated using our approach and simulate the trajectory of the mobile robot using CARLA [42]. In Fig. 1c, we observe that our proposed approach with parameter $\alpha_{12} = 0.1$ avoids any contact with the pedestrian while remain in the road (green color curve) and thus is safe, whereas the baseline approach (red color curve) crashes with the pedestrian and hence fails. A video clip of our simulation is available as the supplement.

B. Spacecraft Rendezvous Problem

In this section, we demonstrate the proposed approach using the spacecraft rendezvous between a chaser and a target satellite. We follow the setting in [40], and represent the dynamics of the satellites using the linearized Clohessy–Wiltshire–Hill equations as follows

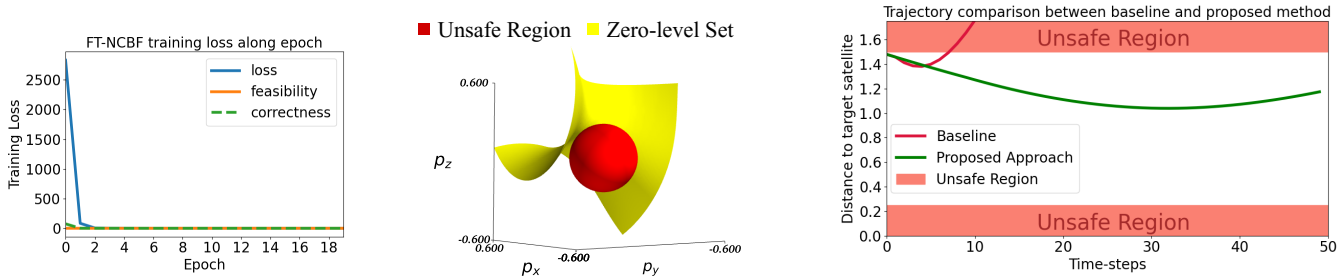
$$\dot{x} = \begin{bmatrix} & I_3 & & & \mathbf{0}_3 & \\ 3n^2 & 0 & 0 & 0 & 2n & 0 \\ 0 & 0 & 0 & -2n & 0 & 0 \\ 0 & 0 & -n^2 & 0 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} \mathbf{0}_3 \\ I_3 \end{bmatrix} u$$

where $x = [p_x, p_y, p_z, v_x, v_y, v_z]^T$ is the state of the chaser satellite, $u = [u_x, u_y, u_z]^T$ is the control input representing the chaser's acceleration, and $n = 0.056$ represents the mean-motion of the target satellite.



(a) (b) (c)

Fig. 1: This figure presents the experimental results on obstacle avoidance of an autonomous mobile robot. Fig. 1a presents the values of loss function, $\mathcal{L}_f(\mathcal{T})$, and $\mathcal{L}_c(\mathcal{T})$. The loss function decreases towards zero during the training process. Fig. 1b shows the zero-level set of $\mathcal{D}_\theta$ corresponding to the FT-NCBF b_θ. The set $\mathcal{D}_\theta$ does not overlap with the unsafe region in red color. Fig. 1c presents the trajectory of the mobile robot when using control policies obtained by our approach and the baseline approach. We observe that our approach guarantees safety whereas the baseline crashes with the pedestrian.



(a) (b) (c)

Fig. 2: This figure presents the experimental results on spacecraft rendezvous problem. In Fig. 2a, we demonstrate that the value of loss function in Eq. (14) quickly converges to zero during training. Fig. 2b presents the zero-level set of $\mathcal{D}_\theta$, which never overlaps with the unsafe region in red color. Fig. 2c simulates the trajectories of the chaser satellite using our approach and the baseline. We observe that our approach allows the chaser satellite to maintain a proper distance to the target satellite (green curve), whereas the baseline fails (red curve).

We define the state space and safety region as $\mathcal{X} = [-2, 2]^6$ and $\mathcal{C} = \{x : r \in [0.25, 1.5], r = \sqrt{p_x^2 + p_y^2 + p_z^2}\}$, respectively. The chaser satellite is required to maintain a safe distance from the target satellite as a safety constraint. The chaser satellite is equipped with a set of sensors to obtain the output $y = [p_x, p_x, p_y, p_y, p_y, v_x, v_y, v_z]^T + \nu$, where $\nu \sim \mathcal{N}(0, \Sigma)$ and $\Sigma = 10^{-5} \times Diag([100, 100, 100, 1, 1, 1, 1, 1])$. We consider two fault patterns $\{r_1, r_2\}$, where r_1 and r_2 are associated with compromised measurements from $y[2]$ and $y[4]$, respectively, raised by a perturbation $a \sim \mathcal{N}(-1, 0.1)$.

We evaluate our approach by comparing with the same baseline approach in Section IV-A. We sample from state space $\mathcal{X}$ using $L = 1$ and obtain a training dataset with $|\mathcal{T}| = 4096$. The training of FT-NCBF took about 1411 seconds with the loss $\mathcal{L}_f(\mathcal{T})$, and $\mathcal{L}_c(\mathcal{T})$ shown in Fig. 2a. We observe that the loss $\mathcal{L}_f(\mathcal{T})$ and $\mathcal{L}_c(\mathcal{T})$ quickly converge to 0, and thus the learned FT-NCBF is valid. We visualize the FT-NCBF b_θ in Fig. 2b. We synthesize a control policy using b_θ in Eq. (6). We observe in Fig. 2c that the chaser satellite never leaves the safety region using the control policy obtained by our approach, whereas the baseline fails to maintain a proper distance from the target satellite, leading to failures in the docking operation.

V. Conclusion

In this paper, we focused on the problem of ensuring safety constraints for stochastic robotic systems under sensor faults and attacks. To tackle the problem, we proposed FT-NCBFs and studied the synthesis of FT-NCBFs by first deriving the necessary and sufficient conditions for FT-NCBFs to guarantee safety. We then developed a data-driven method to learn FT-NCBFs by minimizing a loss function which penalizes the violations of our derived conditions. We investigated the safety-critical control synthesis using the learned FT-NCBFs and established the safety guarantee. Specifically, we maintained a bank of EKFs to estimate system states, and developed a mechanism to resolve conflicting estimates raised by sensor faults and attacks. We demonstrated our approach using the obstacle avoidance of a mobile robot and spacecraft rendezvous. Future work will investigate practical limitations, including on-board computational complexity and data-driven dynamical models.

Acknowledgment

This research was supported by the AFOSR (grants FA9550-22-1-0054 and FA9550-23-1-0208), and NSF (grant CNS-1941670).

REFERENCES

[1] V. N. Fernandez-Ayala, X. Tan, and D. V. Dimarogonas, "Distributed barrier function-enabled human-in-the-loop control for multi-robot systems," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7706–7712, IEEE, 2023.

[2] C. Peng, O. Donca, G. Castillo, and A. Hereid, "Safe bipedal path planning via control barrier functions for polynomial shape obstacles estimated using logistic regression," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3649–3655, IEEE, 2023.

[3] Z. Jian, Z. Yan, X. Lei, Z. Lu, B. Lan, X. Wang, and B. Liang, "Dynamic control barrier function-based model predictive control to safety-critical obstacle-avoidance of mobile robot," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3679–3685, IEEE, 2023.

[4] J. C. Knight, "Safety critical systems: Challenges and directions," in *24th International Conference on Software Engineering*, pp. 547–550, 2002.

[5] W. Schwarting, J. Alonso-Mora, and D. Rus, "Planning and decision-making for autonomous vehicles," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, pp. 187–210, 2018.

[6] 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)*, pp. 3420–3431, IEEE, 2019.

[7] X. Xu, P. Tabuada, J. W. Grizzle, and A. D. Ames, "Robustness of control barrier functions for safety critical control," *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 54–61, 2015.

[8] W. Xiao and C. Belta, "High-order control barrier functions," *IEEE Transactions on Automatic Control*, vol. 67, no. 7, pp. 3655–3662, 2022.

[9] X. Tan, W. S. Cortez, and D. V. Dimarogonas, "High-order barrier functions: Robustness, safety, and performance-critical control," *IEEE Transactions on Automatic Control*, vol. 67, no. 6, pp. 3021–3028, 2021.

[10] C. Dawson, Z. Qin, S. Gao, and C. Fan, "Safe nonlinear control using robust neural Lyapunov-barrier functions," in *Conference on Robot Learning*, pp. 1724–1735, PMLR, 2022.

[11] C. Dawson, S. Gao, and C. Fan, "Safe control with learned certificates: A survey of neural Lyapunov, barrier, and contraction methods for robotics and control," *IEEE Transactions on Robotics*, 2023.

[12] S. Liu, C. Liu, and J. Dolan, "Safe control under input limits with neural control barrier functions," in *Conference on Robot Learning*, pp. 1970–1980, PMLR, 2023.

[13] A. Clark, "Verification and synthesis of control barrier functions," in *2021 60th IEEE Conference on Decision and Control (CDC)*, pp. 6105–6112, IEEE, 2021.

[14] A. Clark, "A semi-algebraic framework for verification and synthesis of control barrier functions," *arXiv preprint arXiv:2209.00081*, 2022.

[15] S. Kang, Y. Chen, H. Yang, and M. Pavone, "Verification and synthesis of robust control barrier functions: Multilevel polynomial optimization and semidefinite relaxation," *arXiv preprint arXiv:2303.10081*, 2023.

[16] H. Dai and F. Permenter, "Convex synthesis and verification of control-Lyapunov and barrier functions with input constraints," *arXiv preprint arXiv:2210.00629*, 2022.

[17] P. Jagtap, S. Soudjani, and M. Zamani, "Formal synthesis of stochastic systems via control barrier certificates," *IEEE Transactions on Automatic Control*, vol. 66, no. 7, pp. 3097–3110, 2020.

[18] P. Tabuada and B. Gharesifard, "Universal approximation power of deep residual neural networks through the lens of control," *IEEE Transactions on Automatic Control*, 2022.

[19] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.

[20] B. C. Csáji *et al.*, "Approximation with artificial neural networks," *Faculty of Sciences, Etvs Lornd University, Hungary*, vol. 24, no. 48, p. 7, 2001.

[21] M. Srinivasan, A. Dabholkar, S. Coogan, and P. A. Vela, "Synthesis of control barrier functions using a supervised machine learning approach," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7139–7145, IEEE, 2020.

[22] W. Xiao, R. Hasani, X. Li, and D. Rus, "Barriernet: A safety-guaranteed layer for neural networks," *arXiv preprint arXiv:2111.11277*, 2021.

[23] C. Dawson, B. Lowenkamp, D. Goff, and C. Fan, "Learning safe, generalizable perception-based hybrid control with certificates," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1904–1911, 2022.

[24] A. Robey, H. Hu, L. Lindemann, H. Zhang, D. V. Dimarogonas, S. Tu, and N. Matni, "Learning control barrier functions from expert demonstrations," in *2020 59th IEEE Conference on Decision and Control (CDC)*, pp. 3717–3724, IEEE, 2020.

[25] L. Lindemann, H. Hu, A. Robey, H. Zhang, D. Dimarogonas, S. Tu, and N. Matni, "Learning hybrid control barrier functions from data," in *Conference on Robot Learning*, pp. 1351–1370, PMLR, 2021.

[26] A. Tampuu, T. Matiisen, M. Semikin, D. Fishman, and N. Muhammad, "A survey of end-to-end driving: Architectures and training methods," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 4, pp. 1364–1384, 2020.

[27] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019.

[28] J. C. B. Gamboa, "Deep learning for time-series analysis," *arXiv preprint arXiv:1701.01887*, 2017.

[29] F. B. Mathiesen, S. C. Calvert, and L. Laurenti, "Safety certification for stochastic systems via neural barrier functions," *IEEE Control Systems Letters*, vol. 7, pp. 973–978, 2022.

[30] R. Mazouz, K. Muvvala, A. Ratheesh Babu, L. Laurenti, and M. Lahijanian, "Safety guarantees for neural network dynamic systems via stochastic barrier functions," *Advances in Neural Information Processing Systems*, vol. 35, pp. 9672–9686, 2022.

[31] H. Zhao, X. Zeng, T. Chen, and Z. Liu, "Synthesizing barrier certificates using neural networks," in *Proceedings of the 23rd international conference on hybrid systems: computation and control*, pp. 1–11, 2020.

[32] Y. Mo and B. Sinopoli, "False data injection attacks in control systems," in *Preprints of the 1st workshop on Secure Control Systems*, vol. 1, 2010.

[33] Y. Guan and X. Ge, "Distributed attack detection and secure estimation of networked cyber-physical systems against false data injection attacks and jamming attacks," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 4, no. 1, pp. 48–59, 2017.

[34] K. Reif, S. Gunther, E. Yaz, and R. Unbehauen, "Stochastic stability of the continuous-time extended Kalman filter," *IEE Proceedings-Control Theory and Applications*, vol. 147, no. 1, pp. 45–52, 2000.

[35] P. K. Panigrahi and S. K. Bisoy, "Localization strategies for autonomous mobile robots: A review," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 8, pp. 6019–6039, 2022.

[36] C. Urrea and R. Agramonte, "Kalman filter: historical overview and review of its use in robotics 60 years after its creation," *Journal of Sensors*, vol. 2021, pp. 1–21, 2021.

[37] A. Clark, Z. Li, and H. Zhang, "Control barrier functions for safe CPS under sensor faults and attacks," in *59th IEEE Conference on Decision and Control (CDC)*, pp. 796–803, IEEE, 2020.

[38] H. Zhang, Z. Li, and A. Clark, "Safe control for nonlinear systems under faults and attacks via control barrier functions," *arXiv preprint arXiv:2207.05146*, 2022.

[39] A. J. Barry, A. Majumdar, and R. Tedrake, "Safety verification of reactive controllers for UAV flight in cluttered environments using barrier certificates," in *2012 IEEE International Conference on Robotics and Automation*, pp. 484–490, IEEE, 2012.

[40] C. Jewison and R. S. Erwin, "A spacecraft benchmark problem for hybrid control and estimation," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, pp. 3300–3305, IEEE, 2016.

[41] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *American Journal of Mathematics*, vol. 79, no. 3, pp. 497–516, 1957.

[42] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Conference on robot learning*, pp. 1–16, PMLR, 2017.