

# Formal Verification of Stochastic ReLU Neural Network Control System

by

Shiqi Sun

Department of Mechanical Engineering & Materials Science  
Duke University

Date: \_\_\_\_\_

Approved:

\_\_\_\_\_

Michael M. Zavlanos, Advisor

\_\_\_\_\_

Michael M. Zavlanos

\_\_\_\_\_

Miroslav Pajic

\_\_\_\_\_

Leila Bridgeman

\_\_\_\_\_

Dissertation submitted in partial fulfillment of the  
requirements for the degree of Doctor of Philosophy  
in the Department of Mechanical Engineering & Materials Science  
in the Graduate School of  
Duke University

2020

ABSTRACT

Formal Verification of Stochastic ReLU Neural Network  
Control System

by

Shiqi Sun

Department of Mechanical Engineering & Materials Science  
Duke University

Date: \_\_\_\_\_

Approved:

\_\_\_\_\_

Michael M. Zavlanos, Advisor

\_\_\_\_\_

Michael M. Zavlanos

\_\_\_\_\_

Miroslav Pajic

\_\_\_\_\_

Leila Bridgeman

\_\_\_\_\_

An abstract of a dissertation submitted in partial fulfillment of the  
requirements for the degree of Doctor of Philosophy  
in the Department of Mechanical Engineering & Materials Science  
in the Graduate School of  
Duke University

2020

Copyright © 2020 by Shiqi Sun  
All rights reserved

# Abstract

In this work, we address the problem of formal safety verification for stochastic cyber-physical systems (CPS) equipped with ReLU neural network (NN) controllers. Our goal is to find the set of initial states from where, with a predetermined confidence, the system will not reach an unsafe configuration within a specified time horizon. Specifically, we consider discrete-time LTI systems with Gaussian noise, which we abstract by a suitable graph. Then, we formulate a Satisfiability Modulo Convex (SMC) problem to estimate upper bounds on the transition probabilities between nodes in the graph. Using this abstraction, we propose a method to compute tight bounds on the safety probabilities of nodes in this graph, despite possible over-approximations of the transition probabilities between these nodes. Additionally, using the proposed SMC formula, we devise a heuristic method to refine the abstraction of the system in order to further improve the estimated safety bounds. Finally, we corroborate the efficacy of the proposed method with a robot navigation example and present comparative results with commonly employed verification schemes.

# Contents

<b>Abstract</b>	<b>iv</b>
<b>List of Figures</b>	<b>vi</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Problem Formulation</b>	<b>4</b>
<b>3 Transition Graph</b>	<b>7</b>
<b>4 Safety Probability Bounds</b>	<b>11</b>
4.1 Partition Merging . . . . .	13
4.2 Transition Probability Normalization . . . . .	15
4.3 Verification Framework . . . . .	17
<b>5 SNN-SMC based Refinement</b>	<b>18</b>
<b>6 Numerical Experiments</b>	<b>20</b>
<b>7 Conclusion</b>	<b>24</b>
<b>Bibliography</b>	<b>25</b>

## List of Figures

6.1	Partition On Workspace . . . . .	21
6.2	Comparison With Other Algorithm . . . . .	21
6.3	Comparison About Merging and Refinement . . . . .	23
6.4	Comparison On One Region . . . . .	23

# Chapter 1

## Introduction

In recent years, advances in the field of deep learning have furnished a new class of adept and adaptable control schemes for cyber-physical systems which considerably simplify the overall design process. Robot navigation is one such exemplar application where neural network controllers have been successfully employed for steering a variety of robotic platforms in a diversity of situations [1–5]. As these AI-enabled systems open up new possibilities for control, the issues of safety and reliability of neural network controllers, which are still considered understudied in the literature, become more pressing. In addition, as such control schemes are employed to address safety-critical real-world problems, the ability to formally verify the security of the neural networks becomes imperative [6].

To address these challenges, a significant effort has been recently directed to the robustification and verification of deep neural networks. Considering the former direction, Generative Adversarial Network (GAN) methodologies have been successfully employed to train networks with improved tolerance to disturbances [7, 8]. Although these methods may yield noticeably more robust networks, they provide no means of estimating the reliability of the system. On the other hand, verification schemes provide ways to estimate bounds on the output of already trained networks and answer reachability queries related to the corresponding closed-loop dynamics. In [9, 10],

a reachability analysis method for neural networks is proposed that relies on semi-definite programming whereas, in [11], satisfiability modulo theory is employed to provide formal verification guarantees. In [12,13], a hybrid system verification scheme is proposed to answer reachability queries concerning dynamical systems equipped with neural network controllers. Likewise, the Satisfiability Modulo Convex optimization (SMC) approach [14] is adopted in [15] in order to verify the safety of neural networks with ReLU activation functions for robotic platforms equipped with proximity sensors. Stochastic problems are also considered in [16–20]. Particularly, in [19,20] a sampling-based method is proposed to ensure safety of a closed-loop system subject to randomness only in the initial conditions, whereas the methodologies in [17,18] are limited to mixed monotone stochastic systems. Finally, guarantees on the safety of stochastic switched systems equipped with general nonlinear controllers are also derived in [16] using Internal Markov Chain (IMC) and Bound-Parameter Markov Decision Process (BMDP) methods which, however, require very fine partitions of the domain in order to furnish accurate safety probability bounds.

In this paper, we propose a new verification scheme for stochastic dynamical systems equipped with ReLU neural network controllers. Particularly, we consider discrete-time LTI systems with Gaussian noise and partition of the continuous state space into convex sets (e.g., constructed as in [15]). Then, we abstract the system by a graph and formulate a Satisfiability Modulo Convex problem which we solve using existing tools in order to estimate valid upper bounds on the transition probabilities



between pairs of nodes in the graph. Using this transition graph, we also develop an algorithm to estimate tight upper bounds on the probability the system reaches the set of unwanted states after a specified amount of steps, even when the underlying transition probability bounds have been over-estimated. Unlike methodologies such as [19, 20] that yield probably correct estimations of the safety probability bounds, here we provide bounds that are correct by design. Additionally, we use the proposed SMC formula to devise a heuristic method to subdivide the cells in a given abstraction in order to further improve our safety probability estimations. Finally, we provide numerical simulations on a robot navigation problem corroborating the efficacy of our proposed verification method compared to [16], which provides looser bounds on the safety probability for the coarse partitions considered here.

We organize the paper as follows. In chapter 2, we formulate the problem under consideration while in chapter 3 we elaborate on the construction of the graph and the methodology used for computing upper bounds on the transition probabilities. Then, in chapter 4, we present the proposed verification scheme and in chapter 5 we develop the proposed heuristic method to refine the selected state abstraction. Finally, in chapter 6 we conclude this work by presenting comparative results corroborating the efficacy of our scheme.

# Chapter 2

## Problem Formulation

We consider an autonomous robot moving in a compact, polytopic workspace  $\mathcal{W} \subset \mathbb{R}^p$  occupied by a set of zero or more inner obstacles  $\{\mathcal{O}_i\}_{i=1}^{p_o}$ . Let  $\mathcal{W}_s = \mathcal{W} \setminus \cup_{i=1}^{p_o} \mathcal{O}_i$  be the set of safe robot positions and let  $\mathcal{W}_o$  denote its complement. The robot's dynamics are described by the following linear discrete-time stochastic model

$$x^{t+1} = Ax^t + Bu^t + w^t, \quad (2.1)$$

where  $x^t \in \mathcal{X} \subseteq \mathbb{R}^n$  and  $u^t \in \mathcal{U} \subset \mathbb{R}^m$  denote the robot's state and control input at time  $t$ , respectively, and  $w^t \sim \mathcal{N}(0, \sigma_t)$ ,  $\sigma_t \in \mathbb{R}^n$  is externally induced Gaussian noise, applied at time  $t$ . We assume that the robot is equipped with one or more sensors that allow it to perceive the unoccluded region of the workspace around it. We shall use  $d(x^t)$  to denote the sensor measurements obtained at configuration  $x^t$ , with  $d : \mathbb{R}^n \mapsto \mathbb{R}^q$  being the measurement function. Also, we assume that a pre-trained, feed-forward neural network controller  $f_{\text{NN}} : \mathbb{R}^q \mapsto \mathbb{R}^m$  is provided for steering the robot to a desired configuration, i.e.,  $u^t = f_{\text{NN}}(d(x^t))$ . Particularly, we assume that

the controller consists of  $L$  fully connected layers, i.e.,

$$\begin{aligned}
h^1 &= \max(0, W_\phi^0 d(x^t) + w_\phi^0), \\
h^2 &= \max(0, W_\phi^1 h^1 + w_\phi^1), \\
&\vdots \\
h^L &= \max(0, W_\phi^{L-1} h^{L-1} + w_\phi^{L-1}), \\
u^t &= W_\phi^L h^L + w_\phi^L
\end{aligned} \tag{2.2}$$

where  $W_\phi^l \in \mathbb{R}^{M_i \times M_{i-1}}$ ,  $w_\phi^l \in \mathbb{R}^{M_i}$  are pre-trained weight bias matrices and  $h^i$  denotes the output of the  $i$ -th layer.

Given the stochastic system (2.1) and associated control law (2.2), let  $P_k : \mathbb{R}^n \mapsto [0, 1]$  denote the probability that the robot will reach an unsafe state after  $k$  time steps starting from  $x^t$ , i.e.,

$$P_k(x^t) = P(\mathcal{P}_W(x^{t+k}) \in \mathcal{W}_o \mid x^t), \tag{2.3}$$

where  $\mathcal{P}_W : \mathbb{R}^n \mapsto \mathbb{R}^p$  is a projection operator that returns the robot's current position. In the remainder, we shall say that a state  $x^t$  is  $(p, k)$ -safe if  $P_k(x) \leq p$ , given  $p \in [0, 1]$ . Note that, in practice, computing a precise approximation of  $P_k$  may generally be intractable. Therefore, in this work, we address the problem of computing a correct-by-design tight upper bound on  $P_k$ , which allows us to answer safety queries, albeit more conservatively.

**Problem 1.** *Given a robotic system obeying the closed-loop dynamics (2.1) and (2.2), compute a tight upper bound of the probability function  $P_k(x)$ , for given horizon  $k$  and  $\forall x \in \mathcal{X}$ .*

# Chapter 3

## Transition Graph

In order to address Problem 1, in this section we develop a methodology to construct a discrete abstraction of the system's dynamics and to compute upper bounds on the transition probabilities between different pairs of cells in this abstraction. Then, in chapter 4, we utilize this transition graph to estimate upper bounds on the safety probability  $P_k$ .

We begin by partitioning the state space  $\mathcal{X}$  into a set  $\mathcal{S} = \{\mathcal{S}_i\}_{i=1}^{p_s}$  of  $p_s$  non-overlapping convex polytopes such that  $\mathcal{X} = \cup_{i=1}^{p_s} \mathcal{S}_i$  and  $\mathcal{S}_i \cap \mathcal{S}_j = \emptyset$  for all  $i \neq j$ . Let  $\mathcal{E} \subseteq \mathcal{S} \times \mathcal{S}$  consist of all the pairs  $\mathcal{S}_i, \mathcal{S}_j$  such that there exists at least one  $x^t \in \mathcal{S}_i$  for which  $P(x^{t+1} \in \mathcal{S}_j) > 0$ . Using the partition  $\mathcal{S}$ , we can model the dynamics of the stochastic system in (2.1) by a graph with edge weights that correspond to upper bounds on the transition probabilities between all pairs of abstract states  $\mathcal{S}_i$  and  $\mathcal{S}_j$ . Specifically, we state the following definition.

**Definition 1.** Given the system (2.1), (2.2) and the partition of the space  $\mathcal{S}$ , an Upper Bound Probabilistic Transition Graph is a tuple  $\mathcal{D} = (\mathcal{S}, \mathcal{E}, \hat{P})$  such that  $P(x^{t+1} \in \mathcal{S}_j | x^t \in \mathcal{S}_i) \leq \hat{P}(\mathcal{S}_i, \mathcal{S}_j)$  for all  $(\mathcal{S}_i, \mathcal{S}_j) \in \mathcal{E}$ .

In order to construct the transition graph in Definition 1, we require a function  $\hat{P} : \mathcal{S} \times \mathcal{S} \rightarrow [0, 1]$  that upper bounds the transition probability from state  $\mathcal{S}_i$  to  $\mathcal{S}_j$  from above. To accomplish this, we extend the SMC encoding presented in [15] to

the case of stochastic dynamical systems considered.

Specifically, we consider the evolution  $X^{t+1} = AX^t + Bu^t$ , where  $X^t$  is the expectation of  $x^t$ . Notice that  $x^{t+1}$  is normally distributed, i.e.,  $x^{t+1} \sim \mathcal{N}(X^{t+1}, \delta^{t+1})$ , for all  $X^t \in \mathcal{X}$ . Notice also that any given convex polytope  $\mathcal{S}_i$  can be defined as the intersection of a finite number of hyperplanes  $a_{i,\ell}^T x \leq b_{i,\ell}$ ,  $\ell \in \mathcal{G}(\mathcal{S}_i)$ , with  $\mathcal{G}(\mathcal{S}_i)$  some arbitrary indexing. Following a procedure similar to the one in [21], we can compute the augmented set

$$\bar{\mathcal{S}}_i(q) = \{X \mid \min_{\ell \in \mathcal{G}(\mathcal{S}_i)} (P(V_{i,\ell}(x) \leq 0)) \geq q, x \sim (X, \delta)\}, \quad (3.1)$$

with  $V_{i,\ell}(x) = a_{i,\ell}x - b_{i,\ell}$ ,  $\ell \in \mathcal{G}(\mathcal{S}_i)$ , which is convex and also its complement consists only of states  $X^{t+1}$  such that  $P(x^{t+1} \in \mathcal{S}_i \mid x^{t+1} \sim \mathcal{N}(X^{t+1}, \delta^{t+1})) < q$ . This last fact can be derived by noticing that  $P(x^{t+1} \in \mathcal{S}_i) < \min_{\ell \in \mathcal{G}(\mathcal{S}_i)} (P(a_{i,\ell}^T x^{t+1} \leq b_{i,\ell}))$ . Next, let  $b_i^l$  indicate the activation status of the  $i$ -th node in the  $l$ -th layer of the neural network controller  $f_{\text{NN}}$ , i.e.,  $b_i^l$  is false when  $h_i^l = 0$ . Then, given a probability threshold  $q \in [0, 1]$  and a pair of  $\mathcal{S}_i, \mathcal{S}_j \in \mathcal{S}$ , we can define the following SMC problem, which we refer to as Stochastic Neural Network SMC (SNN-SMC):

$$\exists X^t, X^{t+1} \in \mathbb{R}^n, u^t \in \mathbb{R}^m, d \in \mathbb{R}^{2N} \quad (3.2)$$

$$(b^l, h^l, t^l) \in \mathbb{B}^{M_l} \times \mathbb{R}^{M_l} \times \mathbb{R}^{M_l}$$

subject to:

$$X^t \in \mathcal{S}_i \tag{3.3}$$

$$\wedge X^{t+1} \in \bar{\mathcal{S}}_j(q) \tag{3.4}$$

$$\wedge X^{t+1} = AX^t + Bu \tag{3.5}$$

$$\wedge (t^1 = W_\phi^0 d(X^t) + w_\phi^0) \cap \left( \bigwedge_{l=2}^L t^l = W_\phi^{l-1} h^{l-1} + w_\phi^l \right) \tag{3.6}$$

$$\wedge (u^t = W_\phi^L h^L + w_\phi^L) \tag{3.7}$$

$$\wedge \bigwedge_{l=1}^L \bigwedge_{i=1}^{M_i} b_i^l \rightarrow [(h_i^l = t_i^l) \wedge (t_i^l \geq 0)] \tag{3.8}$$

$$\wedge \bigwedge_{l=1}^L \bigwedge_{i=1}^{M_i} \neg b_i^l \rightarrow [(h_i^l = 0) \wedge (t_i^l < 0)]. \tag{3.9}$$

In the above definition, (3.3),(3.4) encode the transition from state  $\mathcal{S}_i$  to  $\mathcal{S}_j$ , (3.5) encodes the system dynamics (2.1), and (3.6)-(3.9) encode the behavior imposed by the neural network controller.

Using this SNN-SMC encoding, we now present our proposed algorithm to compute the upper bounds  $\hat{P}$  in Definition 1 on the underlying transition probabilities. Specifically, observe that for a given pair of cells  $\mathcal{S}_i, \mathcal{S}_j$  if the selected threshold  $q$  is large, then there may not be  $x^t$  such that  $P(x^{t+1} \in \mathcal{S}_j | x^t \in \mathcal{S}_i) \geq q$ , which would render the SNN-SMC problem unsatisfiable. Therefore, any such threshold  $q$  is a valid upper bound on the transition probability from  $\mathcal{S}_i$  to  $\mathcal{S}_j$ , i.e.,  $P(\mathcal{S}_i, \mathcal{S}_j) \leq q$ . Based on this fact, we propose an iterative algorithm outlined in Algorithm 1, that

---

**Algorithm 1** Estimation of  $\hat{P}(\mathcal{S}_i, \mathcal{S}_j)$ 

---

**Input:**  $\mathcal{S}_i, \mathcal{S}_j, dq$ **Output:**  $\hat{P}(\mathcal{S}_i, \mathcal{S}_j)$ 

```
1:  $q = 0$ 
2: while  $q < 1$  do
3:   if SNN-SMC( $\mathcal{S}_i, \mathcal{S}_j, q$ ) not satisfiable then
4:     return  $q$ 
5:    $q = q + dq$ 
6: return 1
```

---

increments  $q$  by a user-specified step size  $dq \in (0, 1)$  until the SNN-SMC problem for  $\mathcal{S}_i, \mathcal{S}_j$  can no longer be satisfied. By executing Algorithm 1 for every pair of cells  $\mathcal{S}_i$  and  $\mathcal{S}_j$  in the partition  $\mathcal{S}$ , we can obtain the desired function  $\hat{P}$ .



# Chapter 4

## Safety Probability Bounds

In this section, we elaborate on how to compute upper bounds on the safety probability  $P_k$  given a valid transition graph  $\mathcal{D}$ . First, we extend the definition of a  $(p, k)$ -safe state  $x$  to a  $(p, k)$ -safe cell. Specifically, we say that a cell  $\mathcal{S}_i$  is  $(p, k)$ -safe if all the states in  $\mathcal{S}_i$  are  $(p, k)$ -safe. Let  $\hat{P}_k : \mathcal{S} \mapsto [0, 1]$  denote any function that bounds the safety probability  $P_k$  from above, i.e.,  $\hat{P}_k(\mathcal{S}_i) \geq \max_{x \in \mathcal{S}_i}(P_k(x))$  for all  $\mathcal{S}_i \in \mathcal{S}$ . Then, it is simple to see that any cell  $\mathcal{S}_i \in \mathcal{S}$  is  $(p, k)$ -safe if  $\hat{P}_k(\mathcal{S}_i) \leq p$ . Let  $\mathcal{N}_{\mathcal{S}_i} = \{\mathcal{S}_j | (\mathcal{S}_i, \mathcal{S}_j) \in \mathcal{E}\}$  denote the set of nodes  $\mathcal{S}_j$  reachable from  $\mathcal{S}_i$ . Next, we discuss how to compute tight safety probability bounds  $\hat{P}_k$  for all cells in the partition  $\mathcal{S}$ . We begin by presenting the following proposition which provides a straightforward method to compute  $\hat{P}_{k+1}$  given  $\hat{P}_k$ .

**Proposition 1.** *Assume that the transition and safety probability bounds  $\hat{P}$  and  $\hat{P}_k$  are known. Then, for any  $\mathcal{S}_i \in \mathcal{S}$  the following holds:*

$$P_{k+1}(x) \leq \sum_{\mathcal{S}_j \in \mathcal{N}_{\mathcal{S}_i}} \hat{P}_k(\mathcal{S}_i) \hat{P}(\mathcal{S}_i, \mathcal{S}_j), \quad \forall x \in \mathcal{S}_i \quad (4.1)$$

*Proof.* Let  $x^t \in \mathcal{S}_i$ . Then, by definition, it holds that

$$\begin{aligned}
P_{k+1}(x^t) &= P(x^{t+k+1} \in \mathcal{W}_o | x^t \in \mathcal{S}_i) \\
&= \sum_{\mathcal{S}_j \in \mathcal{N}_{\mathcal{S}_i}} P(x^{t+k+1} \in \mathcal{W}_o | x^{t+1} \in \mathcal{S}_i) P(x^{t+1} \in \mathcal{S}_i | x^t \in \mathcal{S}_i) \\
&\leq \sum_{\mathcal{S}_j \in \mathcal{N}_{\mathcal{S}_i}} \hat{P}_k(\mathcal{S}_j) \hat{P}(\mathcal{S}_j, \mathcal{S}_i)
\end{aligned}$$

□

We remark that a valid choice of  $\hat{P}_0$  is to let  $\hat{P}_0(\mathcal{S}_i) = 1$  if there exists  $x \in \mathcal{S}_i$  such that  $P_W(x) \in \mathcal{W}_o$  and  $\hat{P}_0(\mathcal{S}_i) = 0$  otherwise. Using this choice for  $\hat{P}_0$ , we can recursively compute the safety probability  $\hat{P}_k(\mathcal{S}_i)$  for every cell  $\mathcal{S}_i \in \mathcal{S}$  as

$$\hat{P}_{k+1}(\mathcal{S}_i) = \sum_{\mathcal{S}_j \in \mathcal{N}_{\mathcal{S}_i}} \hat{P}_k(\mathcal{S}_i) \hat{P}(\mathcal{S}_i, \mathcal{S}_j) \tag{4.2}$$

until the desired horizon  $k = T$  is reached. However, note that (4.2) is expected to furnish loose bounds on the safety probabilities when the estimations of the underlying transition probabilities are not tight or the partition is coarse. The reason for the latter case is that Algorithm 1 computes a worst case upper bound on the transition probability from cell  $\mathcal{S}_i$  to cell  $\mathcal{S}_j$  that is close to the transition probability from the worst case state  $X^* = \operatorname{argmax}_{X^t \in \mathcal{S}_i} P(x^{t+1} \in \mathcal{S}_j | X^t)$  with  $x^{t+1} \sim \mathcal{N}(X^{t+1}, \delta^{t+1})$  and  $X^{t+1} = AX^t + Bf_{\text{NN}}(d^t(X^t))$ . Note that if the partition is coarse, it is likely that there are many other states in  $\mathcal{S}_i$  that have much lower transition probabilities to  $\mathcal{S}_j$  but are effectively treated the same as  $X^*$ . If the partition is finer, many of these

states can be grouped in a different cell with lower transition probability to the cell  $\mathcal{S}_j$ . We discuss a way to refine the partition  $\mathcal{S}$  in chapter 5. But first we describe how to improve the safety probability bounds  $\hat{P}_k$  in (4.2) for a given partition  $\mathcal{S}$ .

## 4.1 Partition Merging

In this section, we present a method to merge nodes in  $\mathcal{S}$  in order to improve the transition probability bounds used in (4.2). Specifically, we provide the following result.

**Proposition 2.** *Let  $\mathcal{S}_o \in \mathcal{S}$ ,  $\mathcal{S}_i, \mathcal{S}_j \in \mathcal{N}_{\mathcal{S}_o}$  and  $p \in (0, 1)$ . Consider the set  $\mathcal{S}'_{ij} = \mathcal{S}_i \cup \mathcal{S}_j$ . If  $\bar{\mathcal{S}}_i(p) \cap \bar{\mathcal{S}}_j(p) = \emptyset$ , then  $\mathcal{D}' = (\mathcal{S}', \mathcal{E}', \hat{P}')$  with  $\mathcal{S}' = \mathcal{S} \cup \{\mathcal{S}'_{ij}\}$ ,  $\mathcal{E}' = \mathcal{E} \cup \{(\mathcal{S}_o, \mathcal{S}'_{ij})\} \setminus \{(\mathcal{S}_o, \mathcal{S}_i), (\mathcal{S}_o, \mathcal{S}_j)\}$ ,  $\hat{P}'(\mathcal{S}_o, \mathcal{S}'_{ij}) = \max(\max(\hat{P}(\mathcal{S}_o, \mathcal{S}_i), \hat{P}(\mathcal{S}_o, \mathcal{S}_j)) + p, 2p)$  and  $\hat{P}'(\mathcal{S}_o, \mathcal{S}_i) = \hat{P}'(\mathcal{S}_o, \mathcal{S}_j) = 0$  is a valid transition graph.*

*Proof.* Let  $q_i, q_j \in (0, 1)$ . Recall that the complement of  $\bar{\mathcal{S}}_i(q_i)$  (resp.  $\bar{\mathcal{S}}_j(q_j)$ ) consists only of states  $X \in \mathcal{X}$  such that  $P(x \in \mathcal{S}_i) < q_i$  (resp.  $P(x \in \mathcal{S}_j) < q_j$ ) for  $x \sim \mathcal{N}(X, \delta)$ . Let  $X^t$  be a state in  $\mathcal{S}_o$ . Assuming  $\bar{\mathcal{S}}_i(q_i) \cap \bar{\mathcal{S}}_j(q_j) = \emptyset$ , then  $X^{t+1}$  must lie either in  $\bar{\mathcal{S}}_i(q_i)$ ,  $\bar{\mathcal{S}}_j(q_j)$ , or the complement of their union. As such, the following

inequalities hold:

$$P(x^{t+1} \in \mathcal{S}'_{ij}) < \begin{cases} \hat{P}(\mathcal{S}_o, \mathcal{S}_i) + q_j, & \text{if } X^{t+1} \in \bar{\mathcal{S}}_i(q_i) \\ \hat{P}(\mathcal{S}_o, \mathcal{S}_j) + q_i, & \text{if } X^{t+1} \in \bar{\mathcal{S}}_j(q_j) \\ q_i + q_j, & \text{if } X^{t+1} \in \mathcal{X} \setminus \{\bar{\mathcal{S}}_i, \bar{\mathcal{S}}_j\}. \end{cases} \quad (4.3)$$

Setting  $q_i = q_j = p$  and choosing the worst of these cases concludes the proof.  $\square$

Proposition 2 introduces a new node  $\mathcal{S}'_{ij}$  to the transition graph  $\mathcal{D}$  that is the result of “merging” cells  $\mathcal{S}_i, \mathcal{S}_j \in \mathcal{N}_{\mathcal{S}_o}$  which are far enough from each other so that their augmented sets  $\bar{\mathcal{S}}_i(p)$  and  $\bar{\mathcal{S}}_j(p)$  do not overlap, where  $p \in (0, 1)$  is a user specified probability threshold. Notice that neither  $\mathcal{S}_i$  nor  $\mathcal{S}_j$  are removed from  $\mathcal{D}$ ; what is removed is their edges with  $\mathcal{S}_o$ . Consequently, by repeatedly merging cells in the graph  $\mathcal{D}$  until there are no more cells that can be merged, one can obtain a new graph  $\mathcal{D}'$  which has more nodes than  $\mathcal{D}$  but is not fully connected, i.e.,  $|\mathcal{N}'_{\mathcal{S}_o}| \leq |\mathcal{N}_{\mathcal{S}_o}|$  for all  $\mathcal{S}_o \in \mathcal{S} \subseteq \mathcal{S}'$ . Note also that  $\max_{X^t \in \mathcal{S}_o} P(x^{t+1} \in \mathcal{S}_i \cup \mathcal{S}_j | X^t)$  is bounded from above by  $\max_{X^t \in \mathcal{S}_o} P(x^{t+1} \in \mathcal{S}_i | X^t) + \max_{X^t \in \mathcal{S}_o} P(x^{t+1} \in \mathcal{S}_j | X^t)$  for all cells  $\mathcal{S}_i, \mathcal{S}_j \in \mathcal{N}_{\mathcal{S}_o}$ <sup>1</sup>. In addition, Proposition 2 informs us that  $\hat{P}(\mathcal{S}_o, \mathcal{S}'_{ij}) \ll \hat{P}(\mathcal{S}_o, \mathcal{S}_i) + \hat{P}(\mathcal{S}_o, \mathcal{S}_j)$  for those cells  $\mathcal{S}_i, \mathcal{S}_j$  which are reachable from  $\mathcal{S}_o$  with high transition probabilities but are far away from each other, given small probability threshold  $p$ , because there exists no single state  $X^t$  in  $\mathcal{S}_o$  which is as likely to transition to  $\mathcal{S}_i$  as to  $\mathcal{S}_j$ . As a result,

---

<sup>1</sup>We recall that  $x^{t+1}$  is a random variable sampled from the normal distribution  $\mathcal{N}(X^{t+1}, \delta^{t+1})$  and the probability of  $x^{t+1}$  lying in a set  $\mathcal{S}_o$  is obtained by integrating the corresponding density function over  $\mathcal{S}_o$ .

we can obtain tighter bounds on the safety probabilities  $\hat{P}_k$  since, for the transition graph  $\mathcal{D}'$ , the summation in (4.2) will add fewer terms with associated transition probabilities which are not significantly larger compared to those in the graph  $\mathcal{D}^2$ .

We validate this result in chapter 6.

## 4.2 Transition Probability Normalization

Given the transition graph  $\mathcal{D}$  constructed in Section 4.1, next we present an alternative way to recursively compute tight bounds  $\hat{P}_k$  on the safety probabilities when the underlying transition probability bounds are over-approximated. Specifically, we provide the following result to truncate the sum in (4.2) while ensuring that the new estimation remains a valid upper bound of  $P_k$ .

**Proposition 3.** *Let  $\mathcal{S}_o \in \mathcal{S}$  and  $\hat{\kappa}_o : \mathbb{N} \mapsto \mathbb{N}$  such that  $\hat{P}_k(\mathcal{S}_{\hat{\kappa}_o(i)}) \leq \hat{P}_k(\mathcal{S}_{\hat{\kappa}_o(j)})$ , for all  $i \leq j$  with  $\mathcal{S}_{\hat{\kappa}_o(i)}, \mathcal{S}_{\hat{\kappa}_o(j)} \in \mathcal{N}_{\mathcal{S}_o}$ . Also, let  $n = |\mathcal{N}_{\mathcal{S}_o}|$  and  $\hat{m}$  such that  $\sum_{i=\hat{m}+1}^n \hat{P}(\mathcal{S}_o, \mathcal{S}_{\hat{\kappa}_o(i)}) \leq 1$ . Then, the following inequality holds for all  $x \in \mathcal{S}_o$ :*

$$P_{k+1}(x) \leq \sum_{i=\hat{m}+1}^n \hat{P}(\mathcal{S}_o, \mathcal{S}_{\hat{\kappa}_o(i)}) \hat{P}_k(\mathcal{S}_{\hat{\kappa}_o(i)}) + \left(1 - \sum_{i=\hat{m}+1}^n \hat{P}(\mathcal{S}_o, \mathcal{S}_{\hat{\kappa}_o(i)})\right) \hat{P}_k(\mathcal{S}_{\hat{\kappa}_o(\hat{m})}), \quad (4.4)$$

---

<sup>2</sup>Note that  $\hat{P}(\mathcal{S}_o, \mathcal{S}'_{ij}) < \hat{P}(\mathcal{S}_o, \mathcal{S}_i) + \hat{P}(\mathcal{S}_o, \mathcal{S}_j)$  but  $\hat{P}(\mathcal{S}_o, \mathcal{S}'_{ij}) > \max(\hat{P}(\mathcal{S}_o, \mathcal{S}_i), \hat{P}(\mathcal{S}_o, \mathcal{S}_j))$  for all  $\mathcal{S}_i, \mathcal{S}_j \in \mathcal{N}_{\mathcal{S}_o}$  satisfying the conditions of Proposition 2.

*Proof.* We begin by noticing that

$$P_{k+1}(x^t) \leq \sum_{\mathcal{S}_i \in \mathcal{N}_{\mathcal{S}_o}} P(\mathcal{S}_o, \mathcal{S}_i) \cdot P_k(\mathcal{S}_i), \quad \forall x^t \in \mathcal{S}_o, \quad (4.5)$$

where  $P(\mathcal{S}_o, \mathcal{S}_i) \triangleq P(x^{t+1} \in \mathcal{S}_i | x^t \in \mathcal{S}_o)$  and  $P_k$  denote the lowest upper bounds on the safety probabilities, i.e.,  $P_k(\cdot) \leq \hat{P}_k(\cdot)$ . We recall that  $\sum_{i=1}^{\hat{m}} P(\mathcal{S}_o, \mathcal{S}_{\hat{\kappa}_o(i)}) = 1 - \sum_{i=\hat{m}+1}^n P(\mathcal{S}_o, \mathcal{S}_{\hat{\kappa}_o(i)})$ . By separating the r.h.s. of (4.5) into two sums, one consisting of the  $n - \hat{m}$  terms corresponding to the cells with the highest estimated safety probability bounds  $\hat{P}_k$  and one consisting of the rest, we get

$$\begin{aligned} P_{k+1}(x^t) &\leq \sum_{i=\hat{m}+1}^n P(\mathcal{S}_o, \mathcal{S}_{\hat{\kappa}_o(i)}) \cdot P_k(\mathcal{S}_{\hat{\kappa}_o(i)}) \\ &\quad + P_k(\mathcal{S}_{\hat{\kappa}_o(m)}) \cdot \left( 1 - \sum_{i=\hat{m}+1}^n P(\mathcal{S}_o, \mathcal{S}_{\hat{\kappa}_o(i)}) \right), \end{aligned} \quad (4.6)$$

where  $m = \operatorname{argmax}_{i \in \{1, 2, \dots, m\}} (P_k(\mathcal{S}_{\hat{\kappa}_o(i)}))$ . In general,  $m \leq \hat{m}$  but one can readily see that  $P_k(\mathcal{S}_{\hat{\kappa}_o(m)}) \leq \hat{P}_k(\mathcal{S}_{\hat{\kappa}_o(m)}) \leq \hat{P}_k(\mathcal{S}_{\hat{\kappa}_o(\hat{m})})$ . Therefore, one can verify that the r.h.s. of the above inequality is bounded from above by

$$\begin{aligned} &\sum_{i=\hat{m}+1}^n P(\mathcal{S}_o, \mathcal{S}_{\hat{\kappa}_o(i)}) \cdot \hat{P}_k(\mathcal{S}_{\hat{\kappa}_o(i)}) \\ &\quad + \hat{P}_k(\mathcal{S}_{\hat{\kappa}_o(\hat{m})}) \cdot \left( 1 - \sum_{i=\hat{m}+1}^n P(\mathcal{S}_o, \mathcal{S}_{\hat{\kappa}_o(i)}) \right). \end{aligned} \quad (4.7)$$

Subtracting (4.7) from the r.h.s. of (4.4) and recalling that  $\hat{P}(\mathcal{S}_o, \cdot) > P(\mathcal{S}_o, \cdot)$  and  $\hat{P}_k(\mathcal{S}_{\hat{\kappa}_o(i)}) \geq \hat{P}_k(\mathcal{S}_{\hat{\kappa}_o(\hat{m})})$  by construction of  $\hat{\kappa}_o$ , concludes the proof.  $\square$

---

**Algorithm 2** Verification Framework

---

**Input:**  $\mathcal{D}, \hat{P}_0, p$ **Output:**  $\hat{P}_T$ 

- 1: **for**  $i$  in  $1, 2, \dots, |\mathcal{S}|$  **do**
  - 2:    $\mathcal{D} \leftarrow \text{MergeCells}(\mathcal{D}, \mathcal{S}_i, \mathcal{N}_{\mathcal{S}_i}, p)$
  - 3: **for**  $i$  in  $1, 2, \dots, T$  **do**
  - 4:   **for**  $j$  in  $1, 2, \dots, |\mathcal{S}|$  **do**
  - 5:      $\hat{P}_i(\mathcal{S}_i) \leftarrow \text{SafetyProbabilityEstimation}(\mathcal{S}_i, \mathcal{N}_{\mathcal{S}_i}, P_{i-1})$
- 

Proposition 3 provides an alternative formula to (4.2) to recursively estimate the safety probability bounds  $\hat{P}_k$  that uses the  $\hat{m}$  cells with the largest estimated safety probabilities  $\hat{P}_k$ . Therefore, when the transition probabilities to cells with average or small safety probabilities  $\hat{P}_k$  have been over-approximated, (4.4) allow us to purge them and obtain tighter bounds on  $P_{k+1}$  since, in practice,  $\hat{m}$  is less than  $|\mathcal{N}_{\mathcal{S}_o}|$ .

### 4.3 Verification Framework

Given a graph  $\mathcal{D}$ , upper bounds on the transition and safety probabilities and a user specified horizon  $T$ , we now present an algorithm to compute  $\hat{P}_k$ . Specifically, let  $p \in (0, 1)$  denote a desired probability threshold. Then, for each cell  $\mathcal{S}_o \in \mathcal{S}$  we use Proposition 2 to merge every pair  $\mathcal{S}_i, \mathcal{S}_j$  in  $\mathcal{N}_{\mathcal{S}_i}$  whose augmented sets  $\bar{\mathcal{S}}_i(p)$  and  $\bar{\mathcal{S}}_j(p)$  are not intersecting. Finally, using the upper bounds in Proposition 3, we recursively compute  $\hat{P}_k$  until the desired horizon  $T$  has been reached. The algorithm is illustrated in Algorithm 2.

# Chapter 5

## SNN-SMC based Refinement

In this section, we present a method to refine the cells of a given partition  $\mathcal{S}$  in order to obtain tighter upper bounds on the safety probabilities  $\hat{P}_k$  compared to those obtained for a coarser initial partition. We begin by presenting the following proposition which provides bounds on the transition probabilities of the cells  $\mathcal{S}_i^1, \mathcal{S}_i^2$  obtained by cutting a cell  $\mathcal{S}_i \in \mathcal{S}$  into two disjoint cells separated by a hyperplane  $H$ .

**Proposition 4.** *Given the transition graph  $\mathcal{D}$  and safety probabilities  $\hat{P}_k$ , let  $H$  be a hyperplane splitting  $\mathcal{S}_i \in \mathcal{S}$  into  $\mathcal{S}_i^1$  and  $\mathcal{S}_i^2$ . For all  $\mathcal{S}_j \in \mathcal{N}_{\mathcal{S}_i}$ , if  $\hat{P}(\mathcal{S}_i^1, \mathcal{S}_j) = \hat{P}(\mathcal{S}_i, \mathcal{S}_j)$ , then  $\hat{P}(\mathcal{S}_i^2, \mathcal{S}_j) \leq \hat{P}(\mathcal{S}_i, \mathcal{S}_j)$ . Also, the transition graph  $\mathcal{D}' = (\mathcal{S}', \mathcal{E}', \hat{P}')$  is valid, where  $\mathcal{S}'$  and  $\mathcal{E}'$  are obtained by replacing the cell  $\mathcal{S}_i$  with  $\mathcal{S}_i^1$  and  $\mathcal{S}_i^2$ . Moreover,  $\hat{P}_{k+1}(\mathcal{S}_i^l) \leq \hat{P}_{k+1}(\mathcal{S}_i)$  for all  $l \in \{1, 2\}$ .*

In words, given  $\mathcal{S}_j \in \mathcal{N}_{\mathcal{S}_i}$ , the transition probability bound  $\hat{P}(\mathcal{S}_i^\ell, \mathcal{S}_j)$  computed using Algorithm 1 is the same as  $\hat{P}(\mathcal{S}_i, \mathcal{S}_j)$  for at least one  $\ell \in \{1, 2\}$ . The reason is that satisfiability of the SNN-SMC problem in Algorithm 1 implies that there exists at least one  $X^t \in \mathcal{S}_i = \mathcal{S}_i^1 \cup \mathcal{S}_i^2$  that marginally satisfies the inequality  $P(x^{t+1} \in \mathcal{S}_j | X^t) \geq \hat{P}(\mathcal{S}_i, \mathcal{S}_j) - dq$ . Let  $\mathcal{X}_{ij}^*$  denote the set of all these  $X^t$ . Assuming that  $\mathcal{X}_{ij}^*$  lies strictly in the interior of  $\mathcal{S}_i$ , it can be placed in one of the subcells (e.g.,  $\mathcal{S}_i^1$ ) through a proper selection of the hyperplane  $H$  so that the transition probability of



the other subcell (e.g.,  $\hat{P}(\mathcal{S}_i^2, \mathcal{S}_j)$ ) becomes strictly less than  $\hat{P}(\mathcal{S}_i, \mathcal{S}_j)$ . To find this hyperplane, we consider a state  $X^t \in \mathcal{X}_{ij}^*$  that marginally satisfies the SNN-SMC problem for  $\mathcal{S}_i, \mathcal{S}_j$  and define the hyperplane  $H$  that is perpendicular to the line connecting  $X^t, X^{t+1}$  and contains  $X^t$ , where  $X^{t+1} = AX^t + Bf_{\text{NN}}(d(X^t))$ . Assuming that all  $X^t \in \mathcal{X}_{ij}^*$  lie on the same half-space defined by  $H$ , translating this hyperplane away from  $\mathcal{X}_{ij}^*$  will decrease the transition probability bound from one of the new cells  $\mathcal{S}_i^1$  or  $\mathcal{S}_i^2$  to  $\mathcal{S}_j$ . Thus, given a pair of cells  $\mathcal{S}_i \in \mathcal{S}$  and  $\mathcal{S}_j \in \mathcal{N}_{\mathcal{S}_i}$ , we can refine the partition  $S$  by translating the hyperplane  $H$  so that either  $\hat{P}(\mathcal{S}_i^1, \mathcal{S}_j)$  or  $\hat{P}(\mathcal{S}_i^2, \mathcal{S}_j)$  is minimized.

# Chapter 6

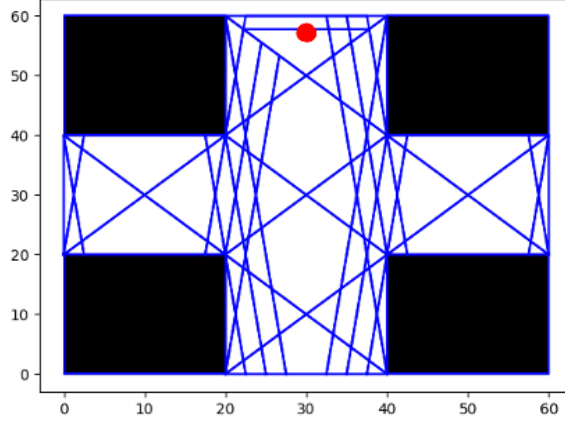
## Numerical Experiments

In this section, we present simulation results to validate the proposed bounds on the probability that a point-sized robot collides with the boundary of the non-convex planar workspace  $\mathcal{W}$ . Particularly, we consider a scenario similar to the one in [15] and assume that the robot’s dynamics can be modeled as a single-integrator, i.e.:

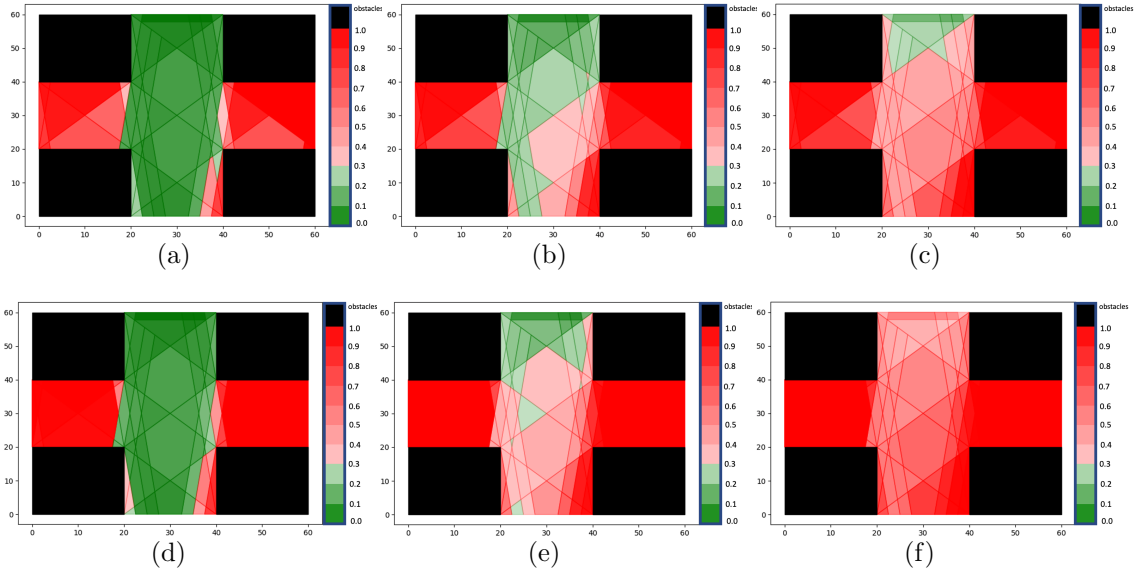
$$x_{t+1} = f_{NN}(d(x_t)) + w^t, \quad (6.1)$$

where  $x^t \in \mathbb{R}^2$  denotes the robot’s position at time step  $t$  and  $w^t \sim \mathcal{N}(0, 3)$ . Additionally, we assume that the robot is equipped with a LiDAR scanner that emits a set of  $q$  lasers evenly distributed in a  $2\pi$  rad fan, i.e.,  $d(x^t) = [(d_0(x^t))^T, \dots, (d_q(x^t))^T]^T$ , where  $d_i(x^t) = [r_i(x^t) \cos \theta_i, r_i(x^t) \sin \theta_i]^T$  and  $r_i(x^t)$  denotes the distance measured between the robot and the closest obstacle in the direction  $[\cos(\theta_i), \sin(\theta_i)]$ , for all  $i \in \{1, 2, \dots, q\}$ . To drive the robotic system to a predetermined goal position using only the feedback  $d(x^t)$ , we employed a ReLU neural network controller  $f_{NN}$  consisting of three hidden layers and a total of 32 neurons. Lastly, we used the partitioning method proposed in [15] to partition the domain  $\mathcal{X}$ , which in this scenario coincides with the workspace  $\mathcal{W}$ . This partition is depicted in Figure 6.1.

In Figure 6.2 we provide a comparison between the bound  $\hat{P}_T$  computed over the

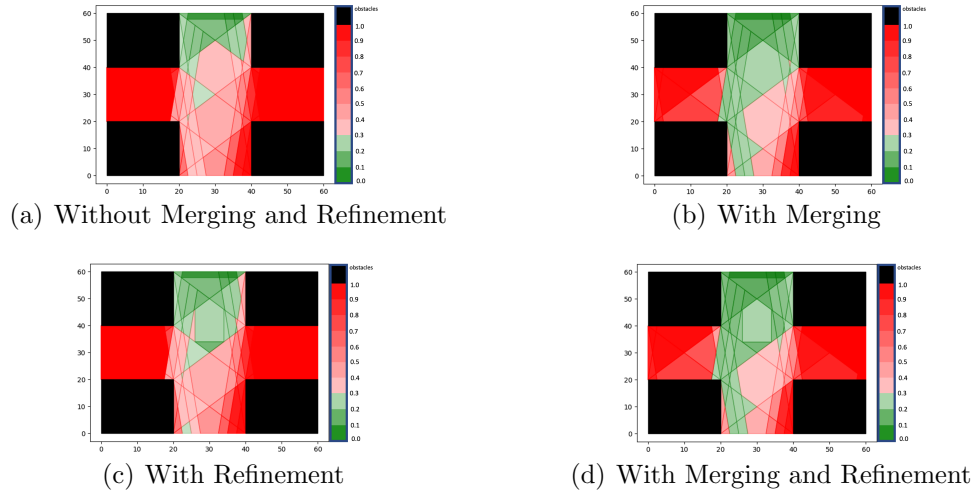


**Figure 6.1:** Compact workspace  $\mathcal{W}$  and partition  $\mathcal{S}$ . Areas shaded in black correspond to obstacles whereas blue lines indicate the boundaries of the convex cells. The red dot marks the goal position of the robot.

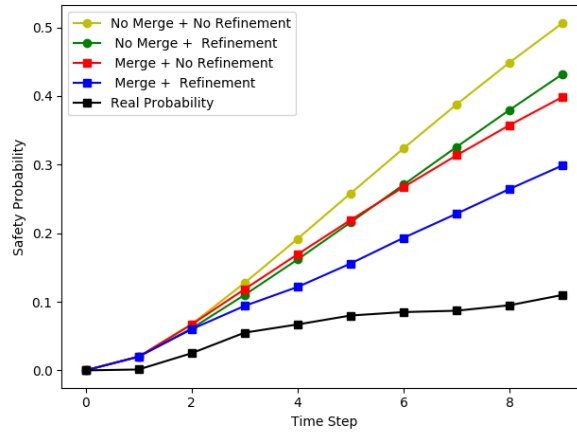


**Figure 6.2:** Safety probability bound  $\hat{P}_T$  for various horizons computed using Algorithm 2 ((a), (b) and (c)) and the algorithm proposed in [16] ((d), (e), (f)). Left column corresponds to  $T = 3$ , middle column to  $T = 6$  and right column to  $T = 9$ .

given domain using Algorithm 2 and the method proposed in [16], for various horizons  $T$ . We observe that our method provides tighter bounds compared to [16] for every horizon, even for the coarse partition considered in this scenario. Additionally, in Figure 6.3, we show the bounds on the safety probability  $P_6$  computed over this domain using Equation 4.2 and Algorithm 2 on the original and refined transition graphs, respectively. The refined graph was obtained using the heuristic subdivision scheme presented in chapter 5 to the largest cell located near the center of the workspace. It can be seen in Figure 6.3 that the safety probability bounds estimated using merging and/or refinement are noticeably tighter than the ones without. Finally, we certify the correctness of the proposed safety probability bounds by comparing them to the true safety probability of for the cell that is adjacent to the one subject to the refinement. To estimate the true safety probability, we simulate a sufficiently large number of robot trajectories starting from states within that cell. In Figure 6.4 we present the safety probability bounds returned by our method for that given cell for different horizons and compare these bounds to the estimated true safety probability. We observe that all bounds returned by our method correctly upper bound the true probability, while the bounds obtained by using both merging and refinement are the tightest. We also remark that the gap between the estimated bounds and the true safety probability becomes larger as the horizon increases.



**Figure 6.3:** Safety probability bound  $\hat{P}_6$  computed using Equation 4.2 (left column) and Algorithm 2 for horizon  $T = 7$  on the original and refined graphs, respectively.



**Figure 6.4:** Bound on and actual safety probability  $P_k$  of a given cell  $\mathcal{S}_i$  for different horizons.

# Chapter 7

## Conclusion

In this work, we addressed the problem of formal safety verification of stochastic cyber-physical systems (CPS) equipped with a ReLU neural network (NN) controllers. Particularly, we presented a method to compute sets of initial states which ensure that the system trajectories are safe within a specified horizon. To do this, we designed a suitable discrete abstraction of the system and formulated an SMC problem to estimate upper bounds on the transition probabilities between cells in this discrete abstraction. Additionally, we proposed a method to obtain tighter bounds on the corresponding safety probability as well as a heuristic for refining the abstraction in a way that may further improve the results. Finally, we presented simulation results verifying the efficacy of our method compared to existing methodologies proposed in the literature.

## Bibliography

- [1] Amini, A., Rosman, G., Karaman, S., & Rus, D. (2019, May). Variational end-to-end navigation and localization. In 2019 International Conference on Robotics and Automation (ICRA) (pp. 8958-8964). IEEE.
- [2] Shamsfakhr, F., & Bigam, B. S. (2017). A neural network approach to navigation of a mobile robot and obstacle avoidance in dynamic and unknown environments. *Turkish journal of electrical engineering & computer sciences*, 25(3), 1629-1642.
- [3] Nägeli, T., Alonso-Mora, J., Domahidi, A., Rus, D., & Hilliges, O. (2017). Real-time motion planning for aerial videography with dynamic obstacle avoidance and viewpoint optimization. *IEEE Robotics and Automation Letters*, 2(3), 1696-1703.
- [4] Chen, Y. F., Everett, M., Liu, M., & How, J. P. (2017, September). Socially aware motion planning with deep reinforcement learning. In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 1343-1350). IEEE.
- [5] Schmuck, V., & Meredith, D. (2019). Training networks separately on static and dynamic obstacles improves collision avoidance during indoor robot navigation. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning* European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (pp. 655-660). Ciaco-i6doc. com.
- [6] Xiang, W., Musau, P., Wild, A. A., Lopez, D. M., Hamilton, N., Yang, X., ... & Johnson, T. T. (2018). Verification for machine learning, autonomy, and neural networks survey. *arXiv preprint arXiv:1810.01989*.
- [7] Charikar, M., Steinhardt, J., & Valiant, G. (2017, June). Learning from untrusted data. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing* (pp. 47-60).

- [8] Li, B., Chen, C., Wang, W., & Carin, L. (2019). Certified Adversarial Robustness with Additive Noise. In *Advances in Neural Information Processing Systems* (pp. 9459-9469).
- [9] Fazlyab, M., Morari, M., & Pappas, G. J. (2019). Probabilistic Verification and Reachability Analysis of Neural Networks via Semidefinite Programming. arXiv preprint arXiv:1910.04249.
- [10] Fazlyab, M., Morari, M., & Pappas, G. J. (2019). Safety verification and robustness analysis of neural networks via quadratic constraints and semidefinite programming. arXiv preprint arXiv:1903.01287.
- [11] Katz, G., Barrett, C., Dill, D. L., Julian, K., & Kochenderfer, M. J. (2017, July). Reluplex: An efficient SMT solver for verifying deep neural networks. In *International Conference on Computer Aided Verification* (pp. 97-117). Springer, Cham.
- [12] Ivanov, R., Weimer, J., Alur, R., Pappas, G. J., & Lee, I. (2019, April). Verisig: verifying safety properties of hybrid systems with neural network controllers. In *Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control* (pp. 169-178).
- [13] Ivanov, R., Carpenter, T. J., Weimer, J., Alur, R., Pappas, G. J., & Lee, I. (2019). Case Study: Verifying the Safety of an Autonomous Racing Car with a Neural Network Controller. arXiv preprint arXiv:1910.11309. controllers
- [14] Shoukry, Y., Nuzzo, P., Sangiovanni-Vincentelli, A. L., Seshia, S. A., Pappas, G. J., & Tabuada, P. (2017, April). SMC: Satisfiability modulo convex optimization. In *Proceedings of the 20th International Conference on Hybrid Systems: Computation and Control* (pp. 19-28).
- [15] Sun X, Khedr H, Shoukry Y. Formal verification of neural network controlled autonomous systems, *Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control*. 2019: 147-156.
- [16] Lahijanian, M., Andersson, S. B., & Belta, C. (2015). Formal verification and synthesis for discrete-time stochastic systems. *IEEE Transactions on Automatic Control*, 60(8), 2031-2045.



- [17] Dutreix, M., & Coogan, S. (2018, April). Efficient verification for stochastic mixed monotone systems. In 2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS) (pp. 150-161). IEEE.
  
- [18] Dutreix M, Coogan S. Specification-guided verification and abstraction refinement of mixed monotone stochastic systems[J]. IEEE Transactions on Automatic Control, 2020.
  
- [19] Zarei, M., Wang, Y., & Pajic, M. (2020, April). Statistical verification of learning-based cyber-physical systems. In Proceedings of the 23rd International Conference on Hybrid Systems: Computation and Control (pp. 1-7).
  
- [20] Wang, Y., Zarei, M., Bonakdarpour, B., & Pajic, M. (2019). Statistical verification of hyperproperties for cyber-physical systems. ACM Transactions on Embedded Computing Systems (TECS), 18(5s), 1-23.
  
- [21] Blackmore L, Ono M, Williams B C. Chance-constrained optimal path planning with obstacles[J]. IEEE Transactions on Robotics, 2011, 27(6): 1080-1094.
  
- [22] Sun, S., Zhang, Y., Luo, X., Vlantis, P., Pajic, M., & Zavlanos M M. (2021, April). Formal Verification of Stochastic Systems with ReLU Neural Network Controllers (pp. 270-276). IEEE. (in preparation)