Safe Learning and Verification of Neural Network Controllers

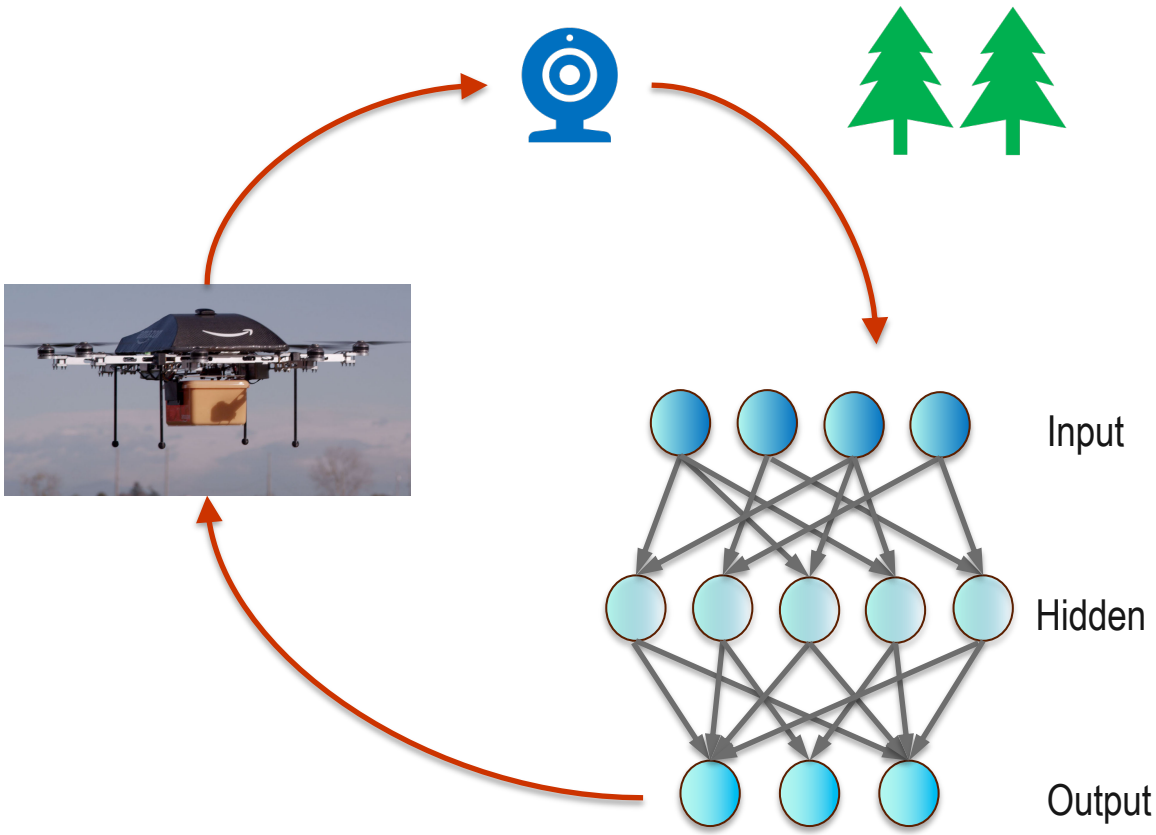
Shiqi Sun, **Yan Zhang**, Xusheng Luo, Panagiotis Vlantis,
Miroslav Pajic and Michael M. Zavlanos

Mechanical Engineering & Materials Science
Duke University

AACE Review Meeting
April 30, 2021

Duke
UNIVERSITY

Cyber-Physical System with NN controller



Why NN controller?

Low onboard computing cost ;

End-to-end implementation …

Safety guarantee?

Duke UNIVERSITY

Stochastic Closed-Loop System

Measurement
$d(x^t)$

$x^{(t+1)} = Ax^t + Bu^t + w$
$w \sim N(\mu, \sigma^2)$

Natural disturbance, uncertain modeling errors, etc.

Input

Hidden

Output

$u^t = ReLU(d(x^t))$

Duke UNIVERSITY

Probabilistic Safety and Problem Formulation

Definition: Let a function $P_k : \mathbb{R}^n \mapsto [0, 1]$ represents the probability of the robot ending up in an unsafe state after k time steps from being initialized at x^t, i.e.:

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

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 call a state x p-safe if $P_k(x) \leq p$, where $p \in [0, 1]$.

Problem Formulation

Given the dynamical system (A, B), the measurement model $d(x)$, the noise model and a trained neural network controller $ReLU(d(x))$, our goal is to verify whether the system is p-safe from a specific state x_0 after k time steps.

Duke UNIVERSITY

Literature Review

Reachability Analysis of NN

[A. Lomuscio & L. Maganti, 2017] [W. Ruan, et al., 2018]
[M. Fazlyab, et al., 2019] [M. Fazlyab, et al., 2020]

One-step reachability analysis

Verification on Deterministic System with NN controller

[W, Xiang & T. Johnson 2018] [X. Sun, et al., 2019] [R. Ivanov, et al., 2019]
[J. Ferlez & Y. Shoukry, 2020] [H. Hu, et al., 2020] [M. Zarei, et al., 2020]

Deterministic dynamics

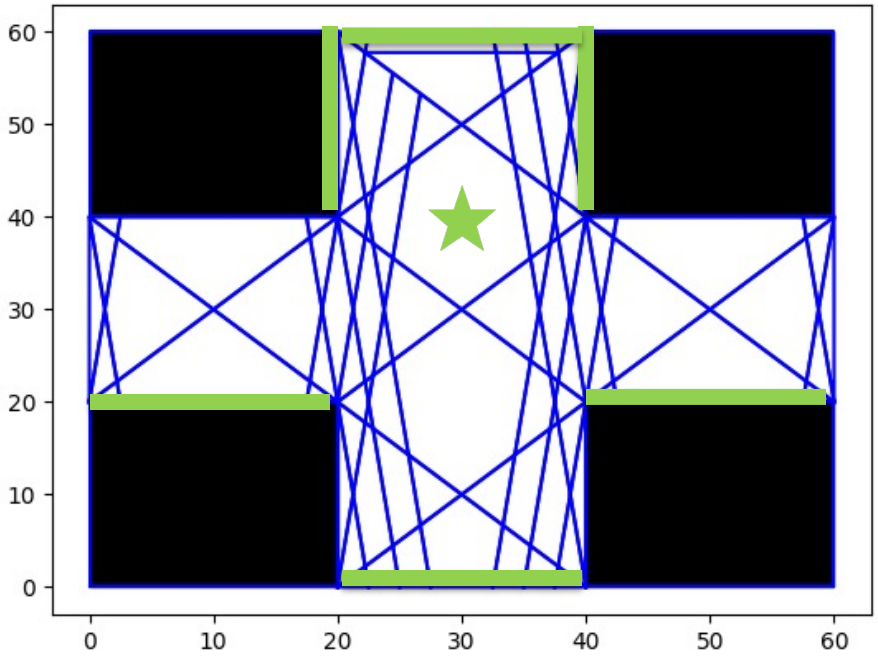
Verification on Stochastic Closed-Loop System

[M. Lahijanian, et al., 2015] [M. Dutreix & S. Coogan, 2018]

Simplified assumption on the system

Duke UNIVERSITY

System Abstraction and Safety over Regions

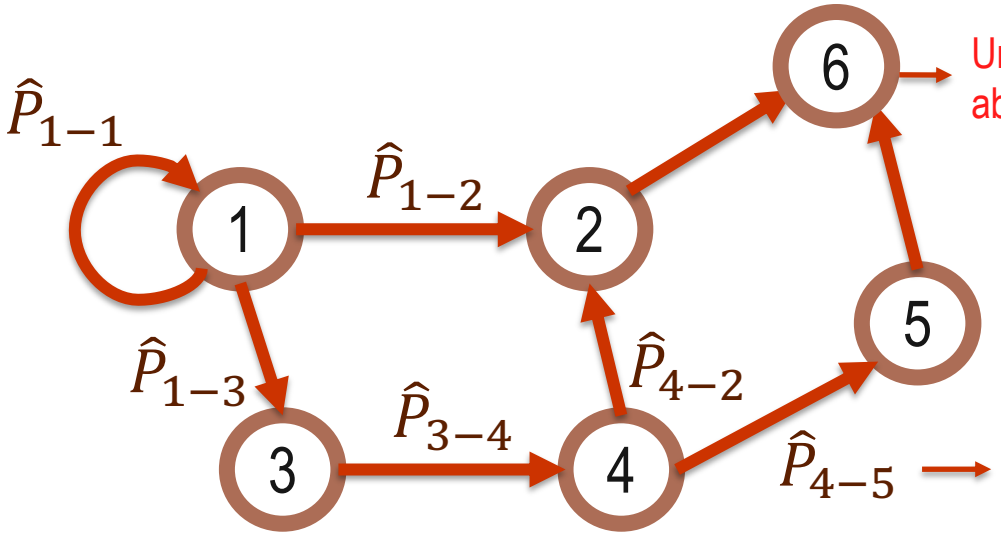


The system is abstracted so that the measurement model $d(x)$ is affine at each specific abstraction [Sun, et al., 2019].

Definition: $\mathcal{S} = \{S_i\}_{i=1}^{p_s}$ is a non-overlapping, polytopic partitioning of the state space. we define $\hat{P}_k : \mathcal{S} \mapsto [0, 1]$ as the extension of the upper bound of P_k over the state abstraction $\mathcal{S}$ as follows:

$$\hat{P}_k(\mathcal{S}_i) = \max_{x \in \mathcal{S}_i}(P_k(x))$$

Dynamic Programming on Transition Graph



Unsafe abstraction

Transition Probability Upper Bound:

$$P(x^{t+1} \in \mathcal{S}_j | x^t \in \mathcal{S}_i) \leq \hat{P}(\mathcal{S}_i, \mathcal{S}_j)$$

Theorem 1: Let S_i be a node of the Transition Graph and let k time step Collison probability $\hat{\mathbf{P}}_\mathbf{k}$ be known. Then, the following holds:

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

How to compute with NN controller?

Give rise to very loose bounds.

Duke UNIVERSITY

Chance constrained SMC with NN constraints

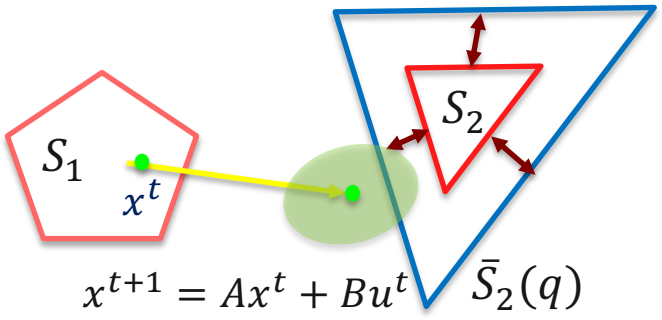
$$\exists \quad x_t, x_{t+1} \in \mathbb{R}^n, u^t \in \mathbb{R}^m, d \in \mathbb{R}^{2N}$$
$$(b^l, h^l, t^l) \in \mathbb{B}^{M_l} \times \mathbb{R}^{M_l} \times \mathbb{R}^{M_l}$$

subject to :

$$\wedge\ x^t \in \mathcal{S}_i$$
$$\wedge\ x^{t+1} \in \bar{\mathcal{S}}_j(q)$$
$$\wedge\ x^{t+1} = Ax^t + Bu^t$$

Feasibility to transit

Chance augmented set



$$\wedge\ \left(t^1 = W^0 d(x^t) + w^0\right) \cap \left(\bigwedge_{l=2}^{L} t^l = W^{l-1} h^{l-1} + w^l\right)$$
$$\wedge\ \left(u^t = W^L h^L + w^L\right)$$
$$\wedge\ \bigwedge_{l=1}^{L}\bigwedge_{i=1}^{M_i} b_i^l \rightarrow \left[\left(h_i^l = t_i^l\right) \wedge \left(t_i^l \geq 0\right)\right]$$
$$\wedge\ \bigwedge_{l=1}^{L}\bigwedge_{i=1}^{M_i} \neg b_i^l \rightarrow \left[\left(h_i^l = 0\right) \wedge \left(t_i^l < 0\right)\right]$$

$$\forall x^{t+1} \notin \bar{\mathcal{S}}_2(q), \hat{P}(\mathcal{S}_i, \mathcal{S}_j) \leq q$$

ReLU Activation Function Constraint

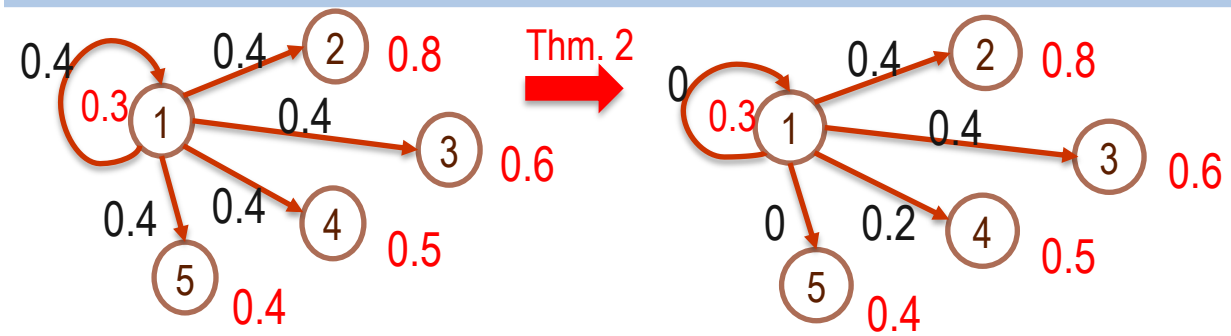
Increase q until we find a value of q making the modified SMC problem become infeasible.

Duke UNIVERSITY

Transition Probability Normalization

Theorem 1: Let S_i be a node of the Transition Graph and let k time step Collison probability $\widehat{\mathbf{P}}_\mathbf{k}$ be known. Then, the following holds:

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



Theorem 1: $\hat{P}_{k+1}(S_1)$= 1.05

Theorem 2: $\hat{P}_{k+1}(S_1)$= 0.66

Theorem 2: Let S_i be a node of the Transition Graph and let k time step Collison probability $\widehat{\mathrm{P}}_\mathrm{k}$ be known. Assume $\widehat{\mathrm{P}}_\mathrm{k}(\mathrm{S}_a) \geq \widehat{\mathrm{P}}_\mathrm{k}(\mathrm{S}_b)$ when $\mathrm{a} \geq b$. Then, the following holds:

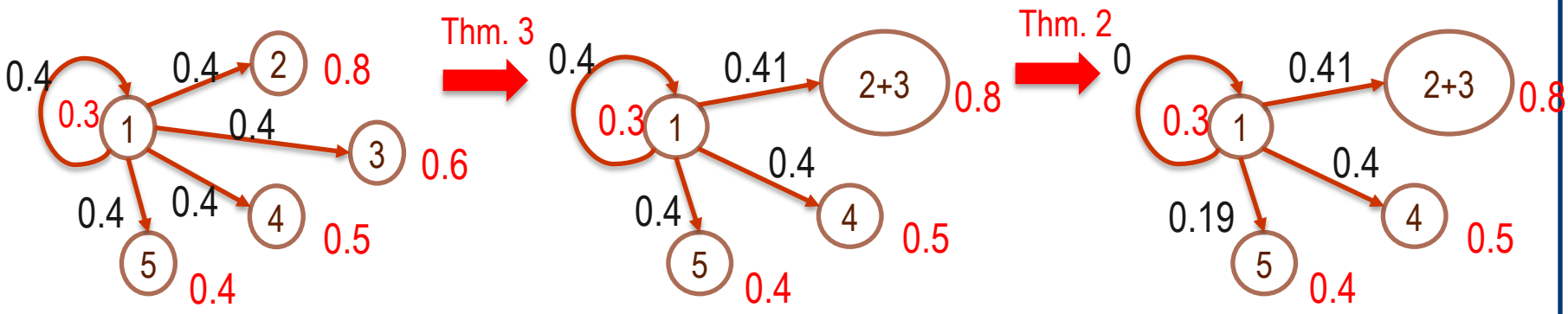
$$\hat{P}_{k+1}(\mathcal{S}_i) = \sum_{j=m+1}^{n} \hat{P}(\mathcal{S}_i, \mathcal{S}_j)\hat{P}_k(\mathcal{S}_j) + \left(1 - \sum_{j=m+1}^{n} \hat{P}(\mathcal{S}_i, \mathcal{S}_j)\right) \hat{P}_k(\mathcal{S}_m)$$

Where m is the first index to make $\sum_{j=m}^{n} \hat{P}(\mathcal{S}_i, \mathcal{S}_j) \geq 1$

Abstraction Merging

Theorem 3: Let S_0, S_1, S_2 be the nodes of transition graph, and $S_1, S_2 \in N_0$. Consider the set $S'_{ij} = S_i \cup S_j$. If $\overline{S_i}(\mathrm{p}) \cap \overline{S_j}(\mathrm{p}) = \emptyset$, then we could merge S_1, S_2 into a new state S'_{ij}.

$$\hat{P}'(\mathcal{S}_o, \mathcal{S}'_{ij}) = \max(\max(\hat{P}(\mathcal{S}_o, \mathcal{S}_i), \hat{P}(\mathcal{S}_o, \mathcal{S}_j)) + p, 2p)$$



Theorem 1: $\hat{P}_{k+1}(S_1)$= 1.05

Theorem 2: $\hat{P}_{k+1}(S_1)$= 0.66

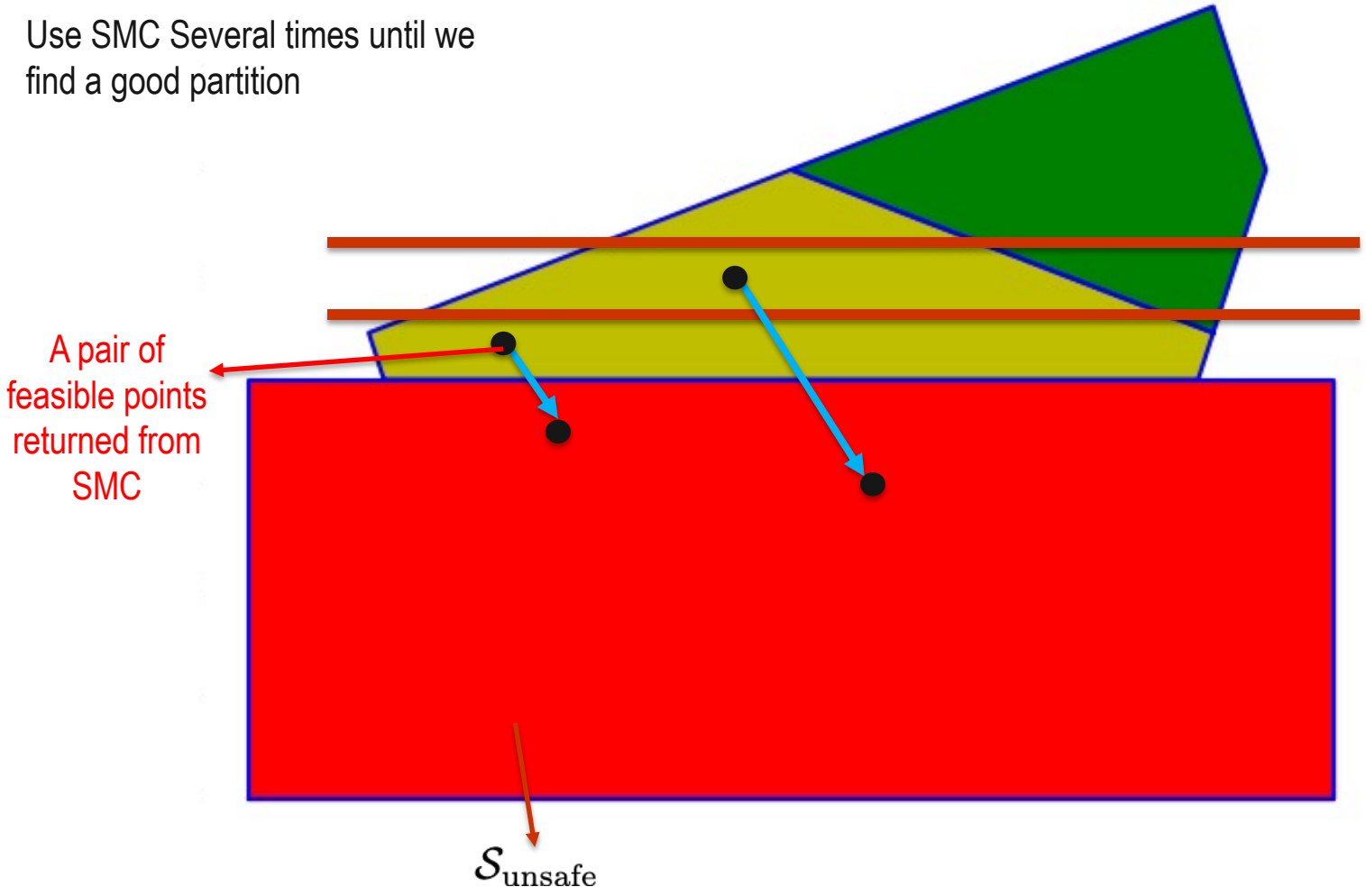
Theorem 3: $\hat{P}_{k+1}(S_1)$= 0.60

Implicit refinement effect without additional computation.

Duke UNIVERSITY

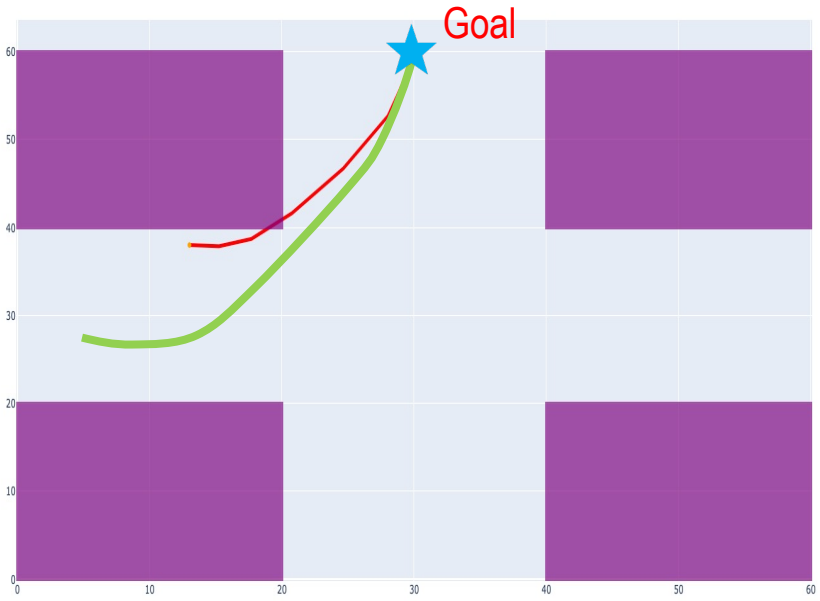
Chance Constrained SMC Based Refinement

Use SMC Several times until we find a good partition

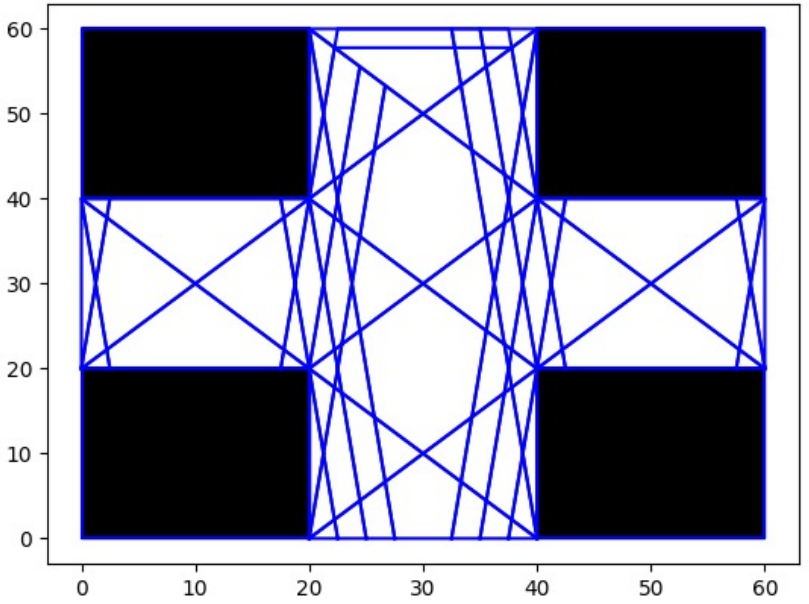


Duke UNIVERSITY

Case Study: Navigation with NN controller



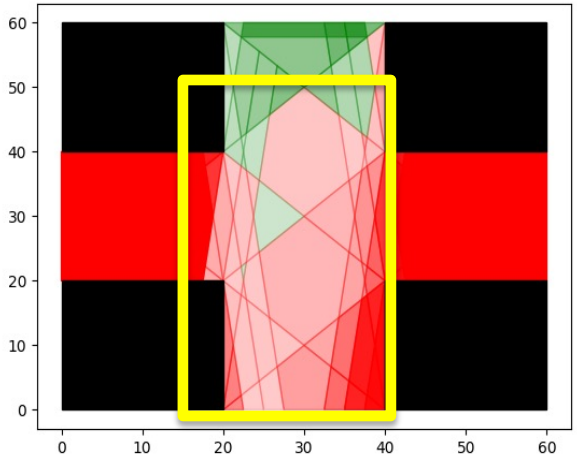
NN Generated Path in the 2D Map



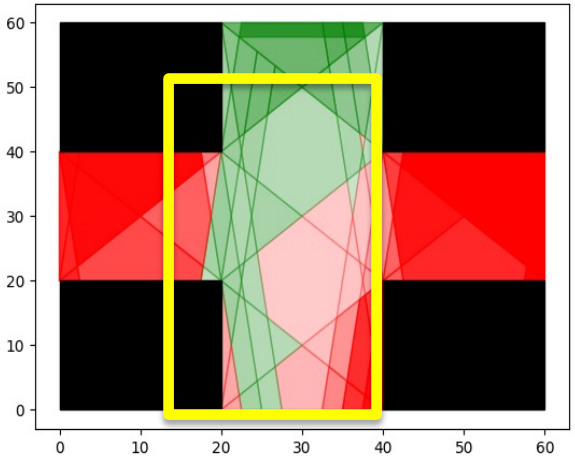
System Abstraction in the 2D Map

Duke
UNIVERSITY

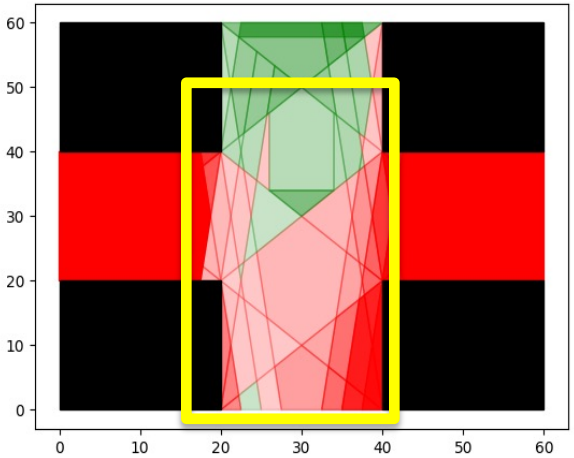
Verification Results with Merging and Refinement



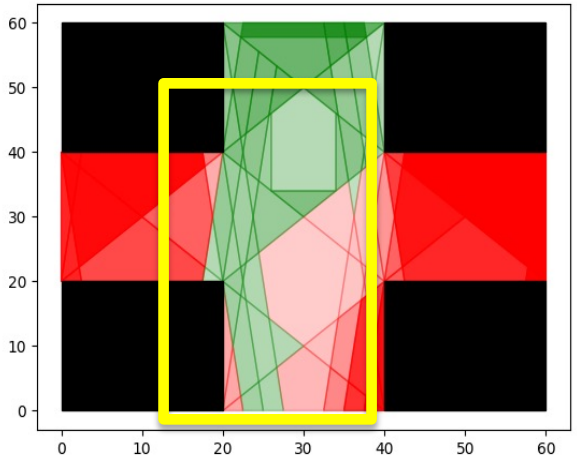
(a) Without merging and refinement, k = 6



(b) With merging, k = 6



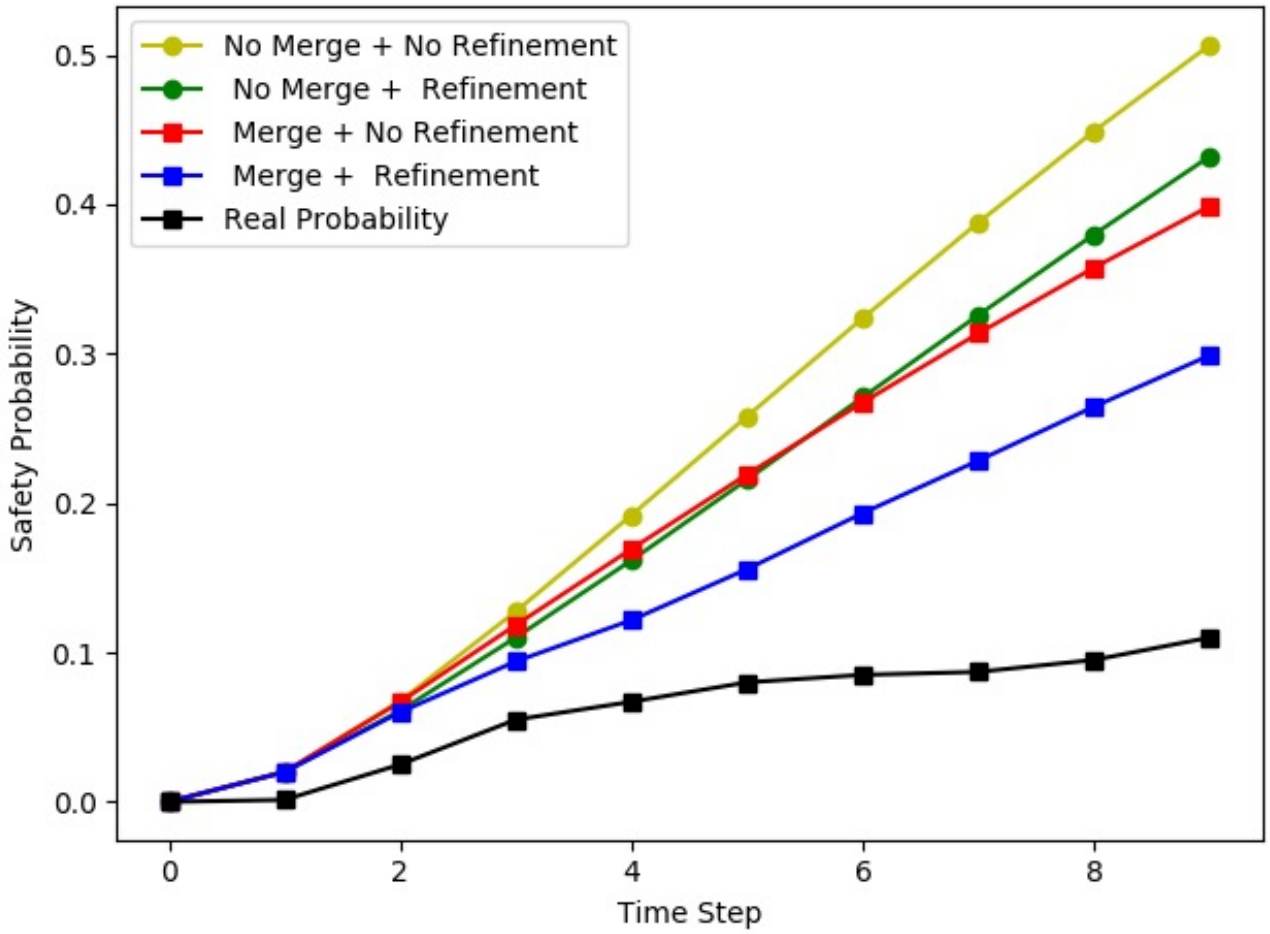
(c) With refinement, k = 6



(d) With refinement and merging, k = 6

Duke UNIVERSITY

Correctness of the Safety Guarantee



Bound on and actual probability of unsafe events at a given abstraction.

Duke UNIVERSITY

Summary

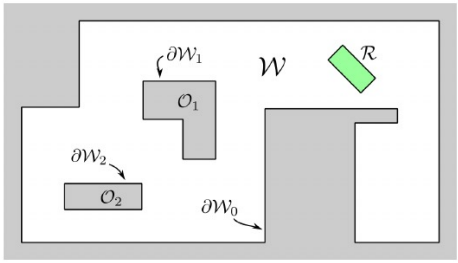
We modify the SMC formulation to compute the upper bound on the transition probability in the probabilistic transition graph with ReLU controller;

We propose to add an abstraction merging step in the verification framework, which requires less computation and can possibly achieve tighter bound on the probability of unsafe events than just doing refinement on the original abstraction.

Duke
UNIVERSITY

Ongoing Work: Safe Deep NN Control Learning

Arbitrary Robot Shape



Complex Workspace

$$z_{k+1} = f\left(z_k, u_k\right),$$

$$u = \phi(z) = \phi_{N_\phi}\left(\phi_{N_\phi - 1}(\ldots \phi_1(z)\ldots)\right),$$

$$\phi_i(x) = h_i(w_i \cdot x + b_i), \quad \forall i \in \mathfrak{I}_{N_\phi},$$

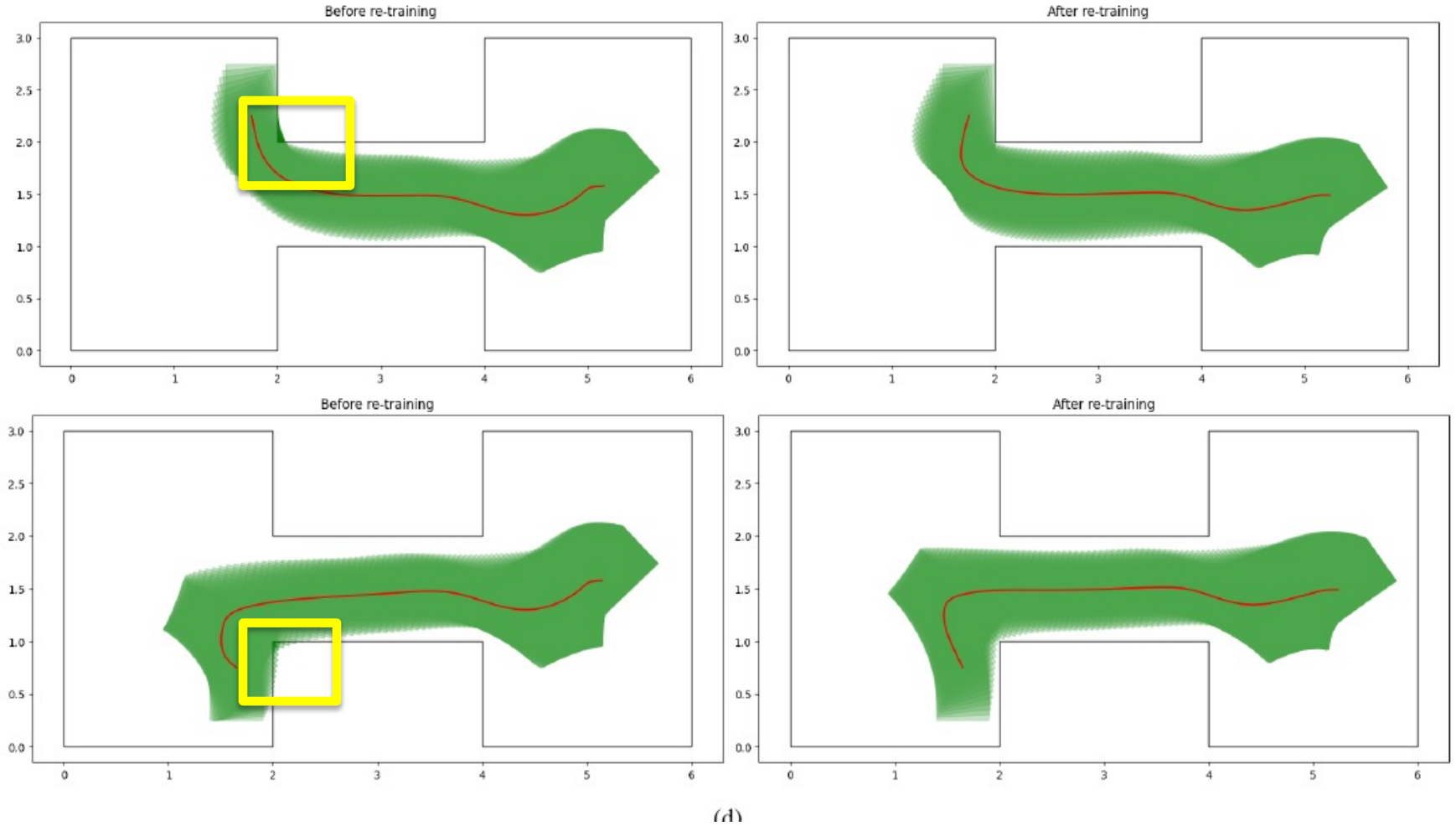
Problem: Given static workspace $\mathcal{W} \subset \mathbb{R}^2$, polygonal robot $\mathcal{R} \subset \mathbb{R}^2$, and dataset $\mathcal{D}$ of points $(z_i, u_i) \in \mathcal{Z} \times \mathcal{U}$:

- Compute tight subset $\underline{\mathcal{Z}}_s$ of set of safe states $\mathcal{Z}_s$.
- Re-train NN controller $\phi(z)$ that fits $\mathcal{D}$ and renders $\underline{\mathcal{Z}}_s$ invariant.

Use reachability analysis on closed-loop system to encapsulate safety constraints.

Duke UNIVERSITY

Preliminary Results



(d)

Duke UNIVERSITY

Thank You for Your Attention!

Duke
UNIVERSITY