Formal Control Synthesis for Stochastic Neural Network Dynamic Models

Adams, S.J.L.; Lahijanian, Morteza; Laurenti, L.

DOI
10.1109/LCSYS.2022.3178143

Publication date
2022

Document Version
Final published version

Published in
IEEE Control Systems Letters

Formal Control Synthesis for Stochastic Neural Network Dynamic Models

Steven Adams, Morteza Lahijanian, *Member, IEEE*, and Luca Laurenti

***Abstract*—Neural networks (NNs) are emerging as powerful tools to represent the dynamics of control systems with complicated physics or black-box components. Due to complexity of NNs, however, existing methods are unable to synthesize complex behaviors with guarantees for *NN dynamic models* (NNDMs). This letter introduces a control synthesis framework for stochastic NNDMs with performance guarantees. The focus is on specifications expressed in *linear temporal logic interpreted over finite traces* (LTLf), and the approach is based on finite abstraction. Specifically, we leverage recent techniques for convex relaxation of NNs to formally abstract a NNDM into an *interval Markov decision process* (IMDP). Then, a strategy that maximizes the probability of satisfying a given specification is synthesized over the IMDP and mapped back to the underlying NNDM. We show that the process of abstracting NNDMs to IMDPs reduces to a set of convex optimization problems, hence guaranteeing efficiency. We also present an adaptive refinement procedure that makes the framework scalable. On several case studies, we illustrate that our framework is able to provide non-trivial guarantees of correctness for NNDMs with architectures of up to 5 hidden layers and hundreds of neurons per layer.**

***Index Terms*—Formal methods, interval Markov decision processes, neural networks, switched systems, synthesis.**

I. Introduction

AUTONOMOUS systems are becoming increasingly complex, often including black-box components and performing complex tasks in the presence of uncertainty. In this context, because of their data efficiency and representation power, deep neural networks (NNs) can be a transformative technology: NNs have already achieved state-of-the-art performance to model and control dynamical systems in various fields, including reinforcement learning (RL) [1]. However, employing NNs in *safety-critical* applications, such as UAVs, where failures may have catastrophic effects, remains a major challenge due to limitations of existing methods to provide performance guarantees. This letter focuses on this challenge and develops a correct-by-construction synthesis framework for systems with *NN dynamic models* (NNDMs).

To achieve complex behaviors with strong guarantees, formal synthesis for control systems have been well-studied in recent years [2]–[4]. These methods use expressive formal languages such as *linear temporal logic* with infinite (LTL) [5] or *finite* (LTLf) [6] interpretation over traces, to specify complex behaviors. For synthesis, control barrier functions (CBF) [7] and abstraction methods are proposed [3]. CBFs allow the use of continuous dynamics but are typically limited to invariant-set problems. The abstraction methods apply rigorous techniques to represent the dynamics as a finite (Kripke) models. Then, by utilizing model-checking-like algorithms on the abstraction, they synthesize controllers that achieve the specification. A key aspect is that both of these methods generally relies on (simple) analytical models. For modern systems, however, such models are often unavailable due to, e.g., complexity of the physics or black-box components.

For their accurate predictive ability, NNDMs are already employed to model dynamical systems [8] and enhance controller training in RL frameworks [1]. In those works, a NN model of the system is trained in closed-loop with a NN controllers, which can be concatenated in a single NN representing the dynamics of the closed-loop system. These benefits have motivated the recent development of methods for formal analysis of NNDM properties [9]–[11], extending verification algorithms for NNs [12] to support temporal properties. Nevertheless, these methods are limited to simple safety properties and often neglect noise in the dynamics. As a consequence, the state-of-the-art techniques for NNDM are still unable to achieve complex behaviors.

In this letter, we close the gap by introducing a control synthesis framework for stochastic NNDMs to achieve a complex specification with formal guarantees. Our approach is based on finite abstraction, and we use LTLf as the specification language which has the same expressively as LTL, but specifies finite behaviors, making it an appropriate language for stochastic models. In particular, we leverage recent convex relaxation techniques for NNs [12] to build piecewise linear functions that under- and overapproximate the NNDM and construct the abstraction as an interval Markov decision process (IMDP) [13]. Critically, we show that this

Manuscript received March 21, 2022; accepted May 8, 2022. Date of publication May 26, 2022; date of current version June 9, 2022. This work was supported in part by the NSF under Award 2039062. Recommended by Senior Editor C. Seatzu. *(Corresponding author: Steven Adams.)*

Steven Adams and Luca Laurenti are with the Delft Center for Systems and Control, TU Delft, 2628 Delft, The Netherlands (e-mail: s.j.l.adams@tudelft.nl; l.laurenti@tudelft.nl).

Morteza Lahijanian is with the Department of Aerospace Engineering Sciences and Computer Science, University of Colorado Boulder, Boulder, CO 80309 USA (e-mail: morteza.lahijanian@colorado.edu).

This article has supplementary downloadable material available at https://doi.org/10.1109/LCSYS.2022.3178143, provided by the authors.

Digital Object Identifier 10.1109/LCSYS.2022.3178143

discretization-based method only requires solving a set of convex optimization problems, which can be reduced to evaluation of an analytical function on a finite set of points, resulting in efficient abstraction procedure. Then, we use existing tools to synthesize a control strategy that optimizes the probability of satisfying a given specification while guaranteeing robustness against uncertainties due to dynamics approximation and discretization. To ensure scalability, we present an adaptive refinement algorithm that iteratively reduces uncertainty in a targeted manner. Finally, we illustrate the efficacy of our framework in several case studies.

In summary, the contributions of this letter are: (i) a novel framework for formal synthesis for stochastic NNDMs with complex specifications, (ii) an efficient finite abstraction technique for NNDMs, (iii) an adaptive refinement algorithm for uncertainty reduction, and (iv) illustration of the efficacy and scalability of the framework on a set of rich case studies with complex NNDMs, whose architecture include up to five hidden layers and hundreds of neurons per layer.

II. Problem Formulation

We consider the following stochastic *neural network dynamic model (NNDM)*:

$$\boldsymbol{x}_{k+1} = f_a^w(\boldsymbol{x}_k) + \boldsymbol{v}_k, \quad \boldsymbol{v}_k \sim \mathcal{N}(0, \mathrm{Cov}_{\boldsymbol{v}}), \tag{1}$$

where $k \in \mathbb{N}$, $\boldsymbol{x}_k, \boldsymbol{v}_k \in \mathbb{R}^n$, and $a \in A = \{a_1, \ldots, a_m\}$ is a finite set of actions. For every a, $f_a^w : \mathbb{R}^n \to \mathbb{R}^n$ is a (trained) feed-forward NN with continuous activation functions, where w denotes the maximum likelihood weights. The noise term $\boldsymbol{v}_k$ is a random variable with stationary non-degenerate Gaussian distribution with zero mean and covariance $\mathrm{Cov}_{\boldsymbol{v}} \in \mathbb{R}^{n\times n}$. Intuitively, $\boldsymbol{x}_k$ is a discrete-time stochastic process whose time evolution is given by iterative predictions of various NNs. We remark that models such as Process (1) are increasingly employed in both robotics and biological systems for both model representation and NN controller training with, e.g., state-of-the-art model-based RL techniques [1], [8], [14]. For instance, Process (1) can represent a NN model in closed loop with different (possibly NN) feedback controllers, and the role of actions a is to switch between different controllers.

Let $\omega_{\boldsymbol{x}}^N = x_0 \xrightarrow{a_0} x_1 \xrightarrow{a_1} \ldots \xrightarrow{a_{N-1}} x_N$ be a finite path of Process (1) of length $N \in \mathbb{N}$ and $\Omega_{\boldsymbol{x}}^{\mathrm{fin}}$ be the set of all finite paths. Paths of infinite length and the set of all paths of infinite lengths are denoted by $\omega_{\boldsymbol{x}}$ and $\Omega_{\boldsymbol{x}}$, respectively, with $\omega_{\boldsymbol{x}}(k)$ denoting the state of $\omega_{\boldsymbol{x}}$ at time k. Given a finite path, a *switching strategy* $\pi_{\boldsymbol{x}} : \Omega_{\boldsymbol{x}}^{\mathrm{fin}} \to A$ chooses the next action of Process (1). The set of all switching strategies is denoted by $\Pi_{\boldsymbol{x}}$. For $a \in A$, $X \subseteq \mathbb{R}^n$, and $x \in \mathbb{R}^n$, we call

$$T^a(X|x) = \int_X \mathcal{N}(\bar{x} \,|\, f_a^w(x), \mathrm{Cov}_{\boldsymbol{v}}) d\bar{x} \tag{2}$$

the *transition kernel* of Process (1) under action a, where $\mathcal{N}(\cdot \,|\, f_a^w(x), \mathrm{Cov}_{\boldsymbol{v}})$ is a normal distribution with mean $f_a^w(x)$ and covariance $\mathrm{Cov}_{\boldsymbol{v}}$. For a strategy $\pi_{\boldsymbol{x}}$, Process (1) defines a probability measure P which is uniquely defined by T^a and by the initial conditions [15] s.t. for every $k > 0$,

$$P[\omega_{\boldsymbol{x}}(k+1) \in X \,|\, \omega_{\boldsymbol{x}}(k) = x, \pi_{\boldsymbol{x}}(\omega_{\boldsymbol{x}}(k)) = a] = T^a(X|x).$$

We are interested in the behavior of Process (1) in compact set $X_{\mathrm{safe}} \subset \mathbb{R}^n$ with respect to the regions of interest in $R = \{\mathfrak{r}_1, \ldots, \mathfrak{r}_{|R|}\}$, where $\mathfrak{r}_i \subseteq X_{\mathrm{safe}}$. To define properties over R, we associate to each region $\mathfrak{r}_i$ the atomic proposition $\mathfrak{p}_i$ such that $\mathfrak{p}_i$ is true iff $x \in \mathfrak{r}_i$. The set of atomic propositions is given by AP, and the labeling function $L{:}X \to 2^{AP}$ returns the set of atomic propositions that are true at each state. Then, we define the observation of path $\omega_{\boldsymbol{x}}^N$ to be $\rho = \rho_0\rho_1 \ldots \rho_N$, where $\rho_i = L(\omega_{\boldsymbol{x}}^N(i))$ for all $i \leq N$.

To express the temporal properties of Process (1), we consider LTLf, which is an expressive language to specify finite behaviors, and hence, appropriate for stochastic systems.

Definition 1: An LTLf formula is built from a set of propositional symbols AP and is closed under the boolean connectives as well as the "until" operator $\mathcal{U}$: $\phi := \top \,|\, \mathfrak{p} \,|\, \neg\phi \,|\, \phi_1 \wedge \phi_2 \,|\, \phi_1 \mathcal{U} \phi_2$, where $\mathfrak{p} \in AP$.

The common temporal operators "eventually" ($\mathcal{F}$) and "globally" ($\mathcal{G}$) are defined as: $\mathcal{F}\phi = \top\mathcal{U}\phi$ and $\mathcal{G}\phi = \neg\mathcal{F}\neg\phi$. The semantics of LTLf can be found in [6]. We say a path $\omega_{\boldsymbol{x}}$ satisfies ϕ, denoted by $\omega_{\boldsymbol{x}} \models \phi$, if a prefix of its observation satisfies ϕ [16].

Problem 1 (Control Synthesis): Given a NNDM as defined in Process (1), a compact set X_{safe}, and an LTLf formula ϕ defined over the regions of interest in X_{safe}, find a switching strategy $\pi_{\boldsymbol{x}}^*$ that maximizes the probability that a path $\omega_{\boldsymbol{x}} \in \Omega_{\boldsymbol{x}}$ of Process (1) satisfies ϕ while remaining in X_{safe}.

To solve Problem 1, we abstract Process (1) into a finite Markov model, where the stochastic nature of Process (1) and the error corresponding to the discretization of the space are formally modeled as uncertainties. The abstraction process involves the computation of bounds on the transition probabilities between different regions of the state space. In Section IV-A, we show that by using linear functions that locally under and overapproximate f_a^w, these bounds can be efficiently computed by solving convex optimization problems. For the resulting Markov model, we then synthesize a strategy that maximizes the probability that the paths of the Markov model satisfy ϕ and can be mapped onto Process (1). Finally, in Section V-A, we develop a scheme that iteratively refines the abstraction based on the synthesis results by reducing the conservatism induced by the approximation bounds of f_a^w.

III. Preliminaries

Notation: We denote by $x^{(l)}$ the l-th element of vector $x \in \mathbb{R}^n$. Further, for convex region $X \subset \mathbb{R}^n$, we denote by $X^{[l]} \subset \mathbb{R}$ the interval of values of X in the l-th dimension, i.e., $X^{[l]} = \{x^{(l)} \,|\, x \in X\}$. Given a linear transformation function (matrix) $\mathcal{T} \in \mathbb{R}^{n\times n}$, the image of region $X \subset \mathbb{R}^n$ under $\mathcal{T}$ is defined as $Im(X, \mathcal{T}) = \{\mathcal{T}x \,|\, x \in X\}$. The post image of region X under action a of Process (1) and $\mathcal{T}$ is defined as $Post(X, \mathcal{T}, a) = \{\mathcal{T}f_a^w(x) \,|\, x \in X\}$. For vectors $x, x' \in \mathbb{R}^n$, we denote by $rect(x, x')$ the axis-aligned hyper-rectangle that is defined by the intervals $r_1 \times r_2 \times \cdots \times r_n$, where $r_l = [\min(x^{(l)}, x'^{(l)}), \max(x^{(l)}, x'^{(l)})]$. In addition, for region $X \subset \mathbb{R}^n$, we denote by $rect(X)$ the hyper-rectangular overapproximation of X, i.e., $rect(X) = rect(\check{x}, \hat{x})$, where $\check{x}^{(l)} = \inf(X^{[l]})$ and $\hat{x}^{(l)} = \sup(X^{[l]})$ for every $l \in \{0, \ldots, n\}$.

Lastly, we define a proper linear transformation function as follows.

Definition 2 (Proper Transformation): For $X \subset \mathbb{R}^n$, the transformation matrix $\mathcal{T} \in \mathbb{R}^{n\times n}$ is proper w.r.t. X if $Im(X, \mathcal{T})$ is an axis-aligned hyper-rectangle.

Note that, as $\mathcal{T}$ is a linear transformation, X must necessarily be a convex polytope in order for $\mathcal{T}$ to be proper [17].

Interval Markov Decision Processes: We utilize IMDPs, also called Bounded MDPs [13], to abstract Process (1). IMDPs are a generalized class of MDPs that allows for a range of transition probabilities between states.

Definition 3 (IMDP): An *interval Markov decision process* (IMDP) is a tuple $\mathcal{I} = (Q, A, \check{P}, \hat{P}, AP_{\mathcal{I}}, L_{\mathcal{I}})$, where

- Q is a finite set of states,
- A is a finite set of actions available in each state $q \in Q$.
- $\check{P} : Q \times A \times Q \to [0, 1]$ is a function, where $\check{P}(q, a, q')$ defines the lower bound of the transition probability from state $q \in Q$ to state $q' \in Q$ under action $a \in A$,
- $\hat{P} : Q \times A \times Q \to [0, 1]$ is a function, where $\hat{P}(q, a, q')$ defines the upper bound of the transition probability from state $q \in Q$ to state $q' \in Q$ under action $a \in A$.
- $AP_{\mathcal{I}}$ is a finite set of atomic propositions,
- $L_{\mathcal{I}} : Q \to 2^{AP}$ is a labeling function assigning to each state $q \in Q$ a subset of $AP_{\mathcal{I}}$.

For all $q, q' \in Q$ and $a \in A$, it holds that $\check{P}(q, a, q') \leq \hat{P}(q, a, q')$ and $\sum_{q'\in Q} \check{P}(q, a, q') \leq 1 \leq \sum_{q'\in Q} \hat{P}(q, a, q')$. A path $\omega_{\mathcal{I}}$ of an IMDP is a sequence of states $\omega_{\mathcal{I}} = q_0 \xrightarrow{a_0} q_1 \xrightarrow{a_1} q_2 \xrightarrow{a_2} \dots$ such that $\hat{P}(q_k, a, q_{k+1}) > 0$ for all $k \in \mathbb{N}$. We denote the last state of a finite path $\omega^{\text{fin}}_{\mathcal{I}}$ by $last(\omega^{\text{fin}}_{\mathcal{I}})$ and the set of all finite and infinite paths by $\Omega^{\text{fin}}_{\mathcal{I}}$ and $\Omega_{\mathcal{I}}$, respectively. A strategy of an IMDP $\pi_{\mathcal{I}} : \Omega^{\text{fin}}_{\mathcal{I}} \to A$ maps a finite path $\omega^{\text{fin}}_{\mathcal{I}} \in \Omega^{\text{fin}}_{\mathcal{I}}$ of $\mathcal{I}$ onto an action in A. The set of all strategies is denoted by $\Pi_{\mathcal{I}}$. Let $\mathcal{D}(Q)$ denote the set of discrete probability distributions over Q. Given a strategy $\pi_{\mathcal{I}}$, the IMDP reduces to a set of infinitely many Markov chains defined by the transition probability bounds of the IMDP. An adversary chooses a feasible distribution from this set at each state and reduces the IMDP to a Markov chain. In particular, an adversary is defined as a function $\xi : \Omega^{\text{fin}}_{\mathcal{I}} \times A \to \mathcal{D}(Q)$ that, for each finite path $\omega^{\text{fin}}_{\mathcal{I}} \in \Omega^{\text{fin}}_{\mathcal{I}}$, state $q = last(\omega^{\text{fin}}_{\mathcal{I}})$, and action $a \in A$, assigns a feasible distribution γ^a_q which satisfies $\check{P}(q, a, q') \leq \gamma^a_q(q') \leq \hat{P}(q, a, q')$. The set of all adversaries is denoted by Ξ.

IV. IMDP Abstraction

In order to solve Problem 1, we first abstract Process (1) into an IMDP $\mathcal{I} = (Q, A, \check{P}, \hat{P}, AP_{\mathcal{I}}, L_{\mathcal{I}})$. To do that, similarly as in [18], we discretize X_{safe} in such way that the transition kernel in (2) can be computed analytically. Let $\mathcal{T}$ be the Mahalanobis transformation $\mathcal{T} = \Lambda^{-\frac{1}{2}}\mathcal{V}^T$, where $\Lambda = \mathcal{V}^T \text{Cov}_{\boldsymbol{\nu}}\mathcal{V}$ is a diagonal matrix whose entries are the eigenvalues of $\text{Cov}_{\boldsymbol{\nu}}$, and $\mathcal{V}$ is the corresponding orthogonal (eigenvector) matrix. Then, the distribution of $\mathcal{T}\boldsymbol{x}_{k+1}$ given $\boldsymbol{x}_k = x$ under action a becomes $\mathcal{N}(\cdot \mid \mathcal{T}f^w_a(x), I)$, where I is the identity matrix. Consequently, given a region $X \subset \mathbb{R}^n$ for which $\mathcal{T}$ is a proper transformation (i.e., $Im(X, \mathcal{T}) = rect(\check{x}, \hat{x})$), we obtain that $T^a(X|x) = g(\mathcal{T}f^w_a(x))$, where

$$g(z) = \frac{1}{2^n}\prod_{l=1}^{n}\left(\text{erf}\left(\frac{z^{(l)} - \check{x}^{(l)}}{\sqrt{2}}\right) - \text{erf}\left(\frac{z^{(l)} - \hat{x}^{(l)}}{\sqrt{2}}\right)\right), \quad (3)$$

and erf$(\cdot)$ is the error function. Hence, we discretize X_{safe} by using a grid in $Im(X_{\text{safe}}, \mathcal{T})$, and denote by $Q_s = \{q_1, \dots, q_{|Q_s|}\}$ the resulting set of regions. To each cell q_i, we associate a state of the IMDP $\mathcal{I}$. We overload the notation by using q_i for both a region in X_{safe} and a state of $\mathcal{I}$. Then, the set of states of $\mathcal{I}$ is defined as $Q = Q_s \cup \{q_u\}$, where q_u denotes the remainder of the state space, i.e., $\mathbb{R}^n \setminus X_{\text{safe}}$.

We define the set of actions of $\mathcal{I}$ to be the set of actions A of Process (1). To ensure a correct abstraction of Process (1), we assume a discretization of X_{safe} that respects the regions of interest in R, i.e., $\forall r \in R, \exists Q_r \subseteq Q$ such that $\cup_{q\in Q_r} q = r$. Under this assumption, the set of atomic propositions $AP_{\mathcal{I}}$ is equal to AP. We define the labeling function $L_{\mathcal{I}}$ with $L_{\mathcal{I}}(q) = L(x)$ for any choice of $x \in q$.

To compute the transition probability bounds $\check{P}$ and $\hat{P}$ for all $q, q' \in Q_s$ and $a \in A$, we need to derive the following bounds, which are the subject of Section IV-A and IV-B:

$$\check{P}(q, a, q') \leq \min_{x\in q} T^a(q'|x), \;\; \hat{P}(q, a, q') \geq \max_{x\in q} T^a(q'|x) \quad (4)$$

The probability interval for transitioning to the state $q_u \in Q$, i.e., the region outside of X_{safe}, is given by

$$\check{P}(q, a, q_u) \leq 1 - \max_{x\in q} T^a(X_{\text{safe}}|x)$$
$$\hat{P}(q, a, q_u) \geq 1 - \min_{x\in q} T^a(X_{\text{safe}}|x),$$

for all $a \in A$ and $q \in Q_s$. Since we are not interested in the behavior of Process (1) outside X_{safe}, we make q_u absorbing, i.e., $\check{P}(q_u, a, q_u) = \hat{P}(q_u, a, q_u) = 1$ for all $a \in A$.

A. Transition Probability Bounds Computation

We derive an efficient and scalable procedure for the computation of the bounds in (4). Recall that the discretization procedure described above enables to write transition kernel T^a as the product of erf in (3). Hence, the optimization problems in (4) can be performed on (3), i.e., for $q', q \in Q_s$ to bound $T^a(q|x)$, we can optimize $g(z)$ over $z \in Post(q, \mathcal{T}, a)$. However, the exact computation of $Post(q, \mathcal{T}, a)$ is intractable since NN-dynamics are inherently nonconvex. Hence, we instead seek to overapproximate $Post(q, \mathcal{T}, a)$ by recursively finding linear functions on the NN-structure that under- and overapproximate $f^w_a(x)$ for all $x \in q \in Q$, as shown in [12]. We can then use these linear functions to bound $Post(q, \mathcal{T}, a)$ for all $q \in Q_s$ as shown in the following proposition.

Proposition 1: For Process (1) under action a, region $q \subset \mathbb{R}^n$, and proper transformation matrix $\mathcal{T}$ w.r.t. q, let $\{v_1, \dots, v_{(2^n)}\} \in \mathbb{R}^n$ be the vertices of hyper-rectangle $Im(q, \mathcal{T})$, and $\check{f}, \hat{f} : \mathbb{R}^n \to \mathbb{R}^n$ be linear functions that bound $\mathcal{T}f^w_a(\mathcal{T}^{-1}z)$ for all $z \in Im(q, \mathcal{T})$. Define $H = \{rect(\check{f}(v), \hat{f}(v)) \mid v \in \{v_1, \dots, v_{(2^n)}\}\}$. Then, it holds that $Post(q, \mathcal{T}, a) \subseteq conv(H)$, where $conv(H)$ is the convex hull of hyper-rectangles in H.

Proof: For $z \in Im(q, \mathcal{T})$, we have that $\mathcal{T}f^w_a(\mathcal{T}^{-1}z) \in rect(\check{f}(z), \hat{f}(z))$. Consequently, $Post(q, \mathcal{T}, a) \subseteq \widetilde{Post}(q, \mathcal{T}, a) =$

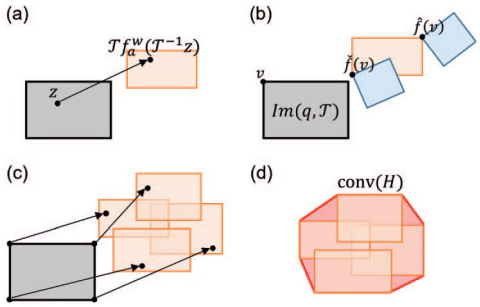

Fig. 1. (a) For each point z in the black rectangle ($Im(q,\mathcal{T})$), $\mathcal{T} f_a^w(\mathcal{T}^{-1}z)$ is contained in the orange rectangle whose vertices are defined by $\check{f}(z)$ and $\hat{f}(z)$. (b) The blue rectangles contain $\check{f}(z)$ and $\hat{f}(z)$ for all $z \in Im(z,\mathcal{T})$. (c) The orange rectangles capture $\mathcal{T} f_a^w(\mathcal{T}^{-1}v)$ for all vertices v of $Im(q,\mathcal{T})$ and (d) fully define the red convex region that captures $\mathcal{T} f_a^w(\mathcal{T}^{-1}z)$ for all $z \in Im(q,\mathcal{T})$.

$\cup_{z\in Im(q,\mathcal{T})} rect(\check{f}(z),\hat{f}(z))$. Note that $Im(q,\mathcal{T})$ is a convex polytope and the construction of $rect(\check{f}(z),\hat{f}(z))$ only involves linear operations. As a consequence, $\widetilde{Post}(q,\mathcal{T},a)$ is fully described by the vertices of $Im(q,\mathcal{T})$. Hence, $\widetilde{Post}(q,\mathcal{T},a) \subseteq conv(H)$. ■

As a result of the above proposition, to construct the post image overapproximation induced by the local linear under- and overapproximations of the NN, we only have to check the vertices of the image as illustrated in Figure 1. Utilizing the analytical reformulation of the transition kernel as in (3) and the post-image overapproximation, we obtain that

$$\min_{x\in q} T^a(q'|x) \geq \min_{z\in conv(H)} g(z) \tag{5}$$

$$\max_{x\in q} T^a(q'|x) \leq \max_{z\in conv(H)} g(z) \tag{6}$$

where H is as defined in Proposition 1. Here, (6) is a log-concave maximization problem, which can be solved with convex optimization algorithms, such as gradient descent [19]. Although (5) is in general non-convex, the following result – a consequence of Corollary 32.3.4 in [17] – guarantees that to compute lower bound (5), i.e., the minimum of a log-concave problem, it suffices to check the vertices of $conv(H)$.

Lemma 1: For $g(z)$ as defined in (3) it holds that

$$\min_{z\in conv(H)} g(z) = \min_{z\in V} g(z) \tag{7}$$

where V is the set of vertices of $conv(H)$.

B. Efficient Computation of Transition Probabilities

Note that, although solving for (5) and (6) reduces to the solution of convex maximization and minimization problems, to build the abstraction, we still need to solve $\mathcal{O}(|Q_s|^2)$ of these problems (one for each pair of states in Q_s). This becomes expensive for large $|Q_s|$, which is often the case for high-dimensional systems. In this section, we propose an alternative approach to reduce this computational burden. In particular, the following theorem shows that if we overapproximate $conv(V)$ by an axis-aligned hyper-rectangle, to find solutions to (4), we only have to check a finite number of points at the boundary of the axis-aligned hyper-rectangle and perform $\mathcal{O}(|Q_s|^2)$ function evaluations, rather than optimizations.

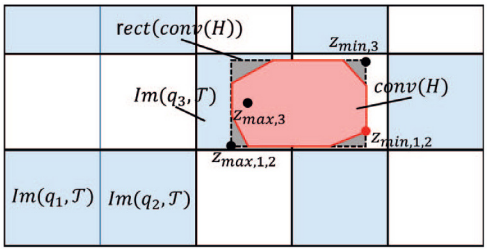

Fig. 2. Regions q_1, q_2 share the same relative position w.r.t. $rect(conv(H))$, i.e., $\forall z \in q_1, q_2$ it holds that $z^{(l)} \leq \inf(rect(conv(H))^{[l]})$ for all $l \in \{0,\ldots,n\}$. Consequently, they share the same z_{min} and z_{max} for problems (8). The white-blue coloring of the regions represents the grouping of regions accordingly. For overlapping region q_3, the minimizing location for (5) is found at a vertex of $conv(H)$.

Theorem 1: For Process (1) under action a, regions $q, q' \subset \mathbb{R}^n$, and proper transformation matrix $\mathcal{T}$ w.r.t. q, q', construct H w.r.t. q per Proposition 1. Further, let vectors $\check{z}, \hat{z}$ define the vertices of $rect(conv(H))$, i.e., $rect(conv(H)) = rect(\check{z},\hat{z})$ such that for every $l \in \{0,\ldots,n\}$ $\check{z}^{(l)} \leq \hat{z}^{(l)}$, and denote by $\bar{v}$ the center of $Im(q',\mathcal{T})$. Then, for $z_{min}, z_{max} \in \mathbb{R}^n$ defined such that for $l \in \{0,\ldots,n\}$

$$z_{min}^{(l)} = \arg\max_{z^{(l)}\in\{\check{z}^{(l)},\hat{z}^{(l)}\}} |z^{(l)} - \bar{v}^{(l)}|,$$

$$z_{max}^{(l)} = \begin{cases} \bar{v}^{(l)} & \text{if } \bar{v}^{(l)} \in [\check{z}^{(l)},\hat{z}^{(l)}] \\ \arg\min_{z^{(l)}\in\{\check{z}^{(l)},\hat{z}^{(l)}\}} |z^{(l)} - \bar{v}^{(l)}| & \text{otherwise} \end{cases}$$

it holds that

$$\min_{x\in q} T^a(q'|x) \geq g(z_{min}), \quad \max_{x\in q} T^a(q'|x) \leq g(z_{max}). \tag{8}$$

Proof: We consider the max case; the min case follows similarly. By construction $conv(H) \subseteq rect(conv(H))$, hence $\max_{x\in q} T^a(q'|x) \leq \max_{z\in rect(conv(H))} g(z)$. As $rect(conv(H))$ is an axis-aligned hyperrectangle, it holds that $\max_{z\in rect(conv(H))} g(z) = \prod_{l=1}^{n} \max_{z^{(l)}\in[\check{z}^{(l)},\hat{z}^{(l)}]} \int_{\check{v}^{(l)}}^{\hat{v}^{(l)}} \mathcal{N}(\bar{x}^{(l)} \,|\, z^{(l)}, 1) d\bar{x}^{(l)}$, where $[\check{v}^{(l)},\hat{v}^{(l)}]$ is the interval of the $l-$th dimension of $Im(q',\mathcal{T})$. This is a product of n maximization problems that seek for the mean of a Gaussian distribution that maximizes its integral on a set. Each of these is maximized by minimizing the distance of $z^{(l)}$ to the center point $\bar{v}^{(l)}$ of the integration set. Hence, $z_{max}^{(l)}$ is equal to $\bar{v}^{(l)}$ if $\bar{v}^{(l)} \in [\check{z}^{(l)},\hat{z}^{(l)}]$, else to one of the endpoints of $[\check{z}^{(l)},\hat{z}^{(l)}]$. ■

According to the above theorem, given $conv(H)$ there exist a finite number of potential z_{max} and z_{min}. Moreover, given a potential z_{min} or z_{max} we can immediately find all the regions that take optimal value for (8) at this point, based on the positions of the regions in the grid, as illustrated in Figure 2. As a consequence, we can simply check the finite sets of all possible z_{min} and z_{max} once to obtain z_{max} and z_{min} for all regions in the same group, and compute (8) by evaluating function g on z_{min} and z_{max} for each region. Although this dramatic reduction of computation comes at the cost of more conservative bounds compared to solving (5) and (6), the introduced overapproximation can be reduced by a refinement algorithm as proposed in Section V-A.

V. Control Synthesis & Abstraction Refinement

Given Process (1) and an LTLf property ϕ, our objective is to synthesize a strategy that maximizes the probability of satisfying ϕ. The IMDP abstraction $\mathcal{I}$ as constructed above, captures the behavior of Process (1) w.r.t. the regions of interest. Therefore, we can focus on finding a strategy for $\mathcal{I}$ that maximizes ϕ subject to being robust against all the uncertainties (errors) induced by the discretization of space and the NN-dynamics approximation process. This translates to assuming that the adversary's (uncertainty) objective is to minimize the probability of satisfaction. Hence,

$$\pi^*_{\mathcal{I}} = \arg\max_{\pi_{\mathcal{I}} \in \Pi_{\mathcal{I}}} \min_{\xi \in \Xi} \mathbb{P}[\omega_{\mathcal{I}} \models \phi \mid \pi_{\mathcal{I}}, \xi, \omega_{\mathcal{I}}(0) = q]. \tag{9}$$

We note that $\pi^*_{\mathcal{I}}$ can be computed using known algorithms with a computational complexity polynomial in the number of states in the IMDP [3]. To show that $\pi^*_{\mathcal{I}}$ maps to a robust strategy for Process (1), we need to introduce a mapping between the process and the IMDP. Let $M_{\boldsymbol{x}} : \mathbb{R}^n \to Q$ be a function that maps continuous states $x \in \mathbb{R}^n$ to their corresponding discrete regions in Q, i.e., $x \in q \implies M_{\boldsymbol{x}}(x) = q$. In addition, let $M_\omega : \Omega^{\text{fin}}_{\boldsymbol{x}} \to \Omega^{\text{fin}}_{\mathcal{I}}$ be a function that maps finite paths of Process (1) to the finite paths of IMDP $\mathcal{I}$, i.e., for a finite path $\omega^N_{\boldsymbol{x}} = x_0 \xrightarrow{a_0} x_1 \xrightarrow{a_1} \dots \xrightarrow{a_{N-1}} x_N$, $M_\omega(\omega^N_{\boldsymbol{x}}) = M_{\boldsymbol{x}}(x_0) \xrightarrow{a_0} M_{\boldsymbol{x}}(x_1) \xrightarrow{a_1} \dots \xrightarrow{a_{N-1}} M_{\boldsymbol{x}}(x_N)$. Then, we can map $\pi^*_{\mathcal{I}}$ to a switching strategy $\pi_{\boldsymbol{x}}$ through

$$\pi^*_{\boldsymbol{x}}(\omega^N_{\boldsymbol{x}}) = \pi^*_{\mathcal{I}}(M_\omega(\omega^N_{\boldsymbol{x}})). \tag{10}$$

Further, we define the lower and upper bounds of the probability of satisfaction of ϕ under $\pi^*_{\mathcal{I}}$ as

$$\check{p}(q) = \min_{\xi \in \Xi} \mathbb{P}[\omega_{\mathcal{I}} \models \phi \mid \pi^*_{\mathcal{I}}, \xi, \omega_{\mathcal{I}}(0) = q], \tag{11}$$

$$\hat{p}(q) = \max_{\xi \in \Xi} \mathbb{P}[\omega_{\mathcal{I}} \models \phi \mid \pi^*_{\mathcal{I}}, \xi, \omega_{\mathcal{I}}(0) = q], \tag{12}$$

respectively. The following theorem shows that the satisfaction probability bounds also hold for Process (1) under $\pi^*_{\boldsymbol{x}}$.

Theorem 2: Given Process (1), a compact set $X_{\text{safe}} \subset \mathbb{R}^n$, and an LTLf formula ϕ defined over the regions of interest in X_{safe}, let $\mathcal{I}$ be the IMDP abstraction of Process (1) as described in Section IV. Further, let $\pi^*_{\mathcal{I}}$ be computed by (9) with probability bounds $\check{p}$ and $\hat{p}$ as in (11) and (12), respectively. Map $\pi^*_{\mathcal{I}}$ into a switching strategy $\pi^*_{\boldsymbol{x}}$ as in (10). Then for any initial state $x_0 \in X_{\text{safe}}$ it holds that

$$P[\omega_{\boldsymbol{x}} \models \phi \mid \pi^*_{\boldsymbol{x}}, \omega_{\boldsymbol{x}}(0) = x_0] \in [\check{p}(M_{\boldsymbol{x}}(x_0)), \hat{p}(M_{\boldsymbol{x}}(x_0))].$$

The proof of this theorem follows similarly as the proof of [20, Th. 2]. Theorem 2 guarantees that the probability that Process (1) satisfies ϕ is contained in the satisfaction probability bounds $\check{p}$ and $\hat{p}$. The difference between $\check{p}$ and $\hat{p}$ can be viewed as the error induced by space discretization and local approximation of the NN dynamics with linear functions. This error monotonically decreases if the size of the discretization decreases. As a consequence, the synthesized strategy is optimal for an infinitely fine grid.

A. Synthesis Driven Refinement

Here, we present a discretization refinement scheme that aims to efficiently reduce the error induced by the space discretization. In each refinement step, we refine a predefined fixed number of states in Q_s, which we refer to as n_{ref}. To enable the use of Theorem 1 and Proposition 1, our refinement guarantees that all refined regions are axis-aligned hyper-rectangles in the transformed space. Hence, a region is refined by splitting the corresponding hyper-rectangle region $Im(q, \mathcal{T})$ over one dimension. To decide on which states to refine, we define a score function $\theta : Q \to \mathbb{R}^+$ as

$$\theta(q) = (\check{p}(q) - \hat{p}(q)) \sum_{a \in A} \sum_{q' \in Q} \left(\hat{P}(q', a, q) - \check{P}(q', a, q) \right)$$

where $\check{p}$ and $\hat{p}$ are the satisfaction probabilities as defined by (11) and (12). We refine the n_{ref} regions with the highest $\theta(q)$. The score function serves as a measure of uncertainty caused by state $q \in Q_s$ and closely resembles the uncertainty measure proposed in [21] in a verification context.

The rationale behind our choice of which dimension to refine is based on the objective to reduce the conservatism introduced by the NN overapproximation process. In particular, for $q \in Q_s$, we want to find the dimension that minimizes the volume of $conv(H)$, as described in Proposition 1, for both regions created by splitting $Im(q, \mathcal{T})$ over this dimension. To do so, we transform all the edges of $Im(q, \mathcal{T})$ using the bounding functions and measure the expansion of the edges, i.e., the relative difference in distance between the vertices describing an edge before and after the transformation. As $Im(q, \mathcal{T})$ is an axis-aligned hyper-rectangle, we then take the dimension to refine equal to the dimension the largest expanded edge aligns to. The exact procedure to find the dimension to refine can be found in the technical report [22].

VI. Case Studies

We consider 3 different NNDMs learned on non-linear datasets taken from the literature (for details see Appendix I-E in the technical report [22]). All transition probability bounds of the IMDP abstractions were computed using Theorem 1, except for the lower bounds on transitions to regions that overlap with the post image overapproximation of the region from which the transition starts. For those, we used Proposition 1. Empirically, we found this approach to offer good results in balancing between precision and scalability. All experiments were run on an Intel Core i7-10610U CPU at 1.80GHz× 2.30Hz with 16GB of RAM.

1) Efficient Control Synthesis by Iterative Refinement: We consider a 3-D car model from [23], with state space representing position and orientation of the car and seven discrete actions switching between different feedback controllers that steer the car to a given orientation. We are interested in synthesizing a strategy for a static overtaking scenario as shown in Figure 3. Here, the car should globally avoid an obstacle ("O") and eventually reach a desired ("D") region, i.e., $\phi_1 = \mathcal{G}(\neg O) \wedge \mathcal{F}(D)$. To do so, we start with a very coarse abstraction and iteratively refine the discretization, which overall takes approximately 36 minutes. From Figure 3 we observe

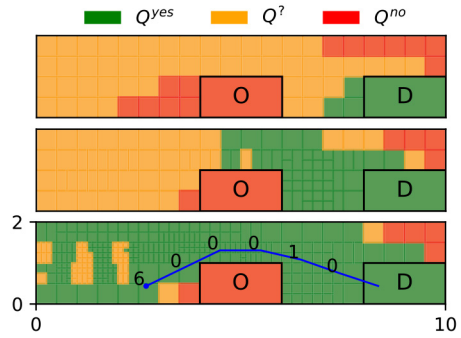

Fig. 3. Region labeling and classification of each initial states $x \in X$ as Q^{yes} if $\check{p}(M_x(x)) \geq 0.95$, Q^{no} if $\hat{p}(M_x(x)) < 0.95$ and $Q^{?}$ otherwise, for the first, an intermediate and the final abstraction of Experiment 1. In blue, a simulated path labeled with the action chosen at each step.

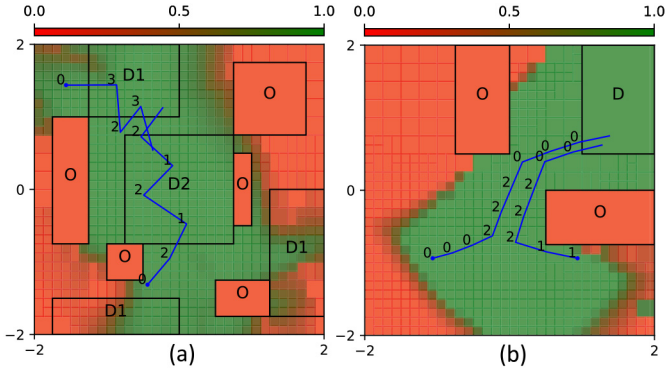

Fig. 4. Region labeling and lower satisfaction probability bounds of the initial states for Experiment 2 (a) and 3 (b).

that the refinement procedure preserves the initial coarseness of the discretization for regions with small uncertainty on the satisfaction probability, whereas the critical regions, such as the corners around the obstacle, are further refined. Hereby, not only the lower bounds improve (the orange regions turn green), but also the upper bounds improve (red regions turn orange or green), and the controller strategies based on non-informative lower bounds are updated (red regions turn green).

2) Control Synthesis for Complex Specifications: To show that our framework can handle complex specifications, we use four nonlinear 2-D datasets generated by the nonlinear system considered in [24], and perform control synthesis given the same labeling of the domain and complex LTLf specification as in [24], i.e., $\phi_2 = \mathcal{G}(\neg O) \wedge \mathcal{F}(D1) \wedge \mathcal{F}(D2)$. The iterative abstraction and control synthesis procedure takes approximately 65 seconds, and the final abstraction consists of $1,500$ states. Figure 4a shows that, although we assume a noisier dataset, we are able to compute informative satisfying probability bounds that resemble the result in [24].

3) Scalability (High-Dimensional and Complex NN-Structures): Last, we test the scalability of our framework on a 5-D system with NNs of 5 hidden layers with 100 neurons per layer. Here, we consider the reach-avoid specification ϕ_1 with the labeling of the space as shown in Figure 4b. We again start with a coarse abstraction and iteratively improve the abstraction, which overall takes approximately 2 hours, from which 1.5 hours for generating the IMDP abstractions. The final abstraction consists of approximately $15,000$ states. Figure 4b shows that we are able to guarantee for a large part of the domain that the initial states almost always (green regions) or almost never (red regions) satisfy the specification using complex controller strategies as indicated by the simulated paths and corresponding actions.

VII. Conclusion

We introduced a formal control synthesis framework for a stochastic NNDMs with LTLf specifications. We showed that in practice the abstraction can be constructed very efficiently and developed an iterative refinement scheme to efficiently minimize the number of states of this discretization-based method. By experiments on various datasets, we showed that our framework enables efficient control synthesis of provably correct strategies for complex NNDM of several input dimensions on nontrivial control tasks. In the future, we plan to extend our framework to NNDMs driven by Recurrent Neural Networks and NNDMs with a continuous action space.

References

[1] A. Nagabandi, G. Kahn, R. S. Fearing, and S. Levine, "Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning," in *Proc. ICRA*, 2018, pp. 7559–7566.
[2] P. Tabuada, *Verification and Control of Hybrid Systems: A Symbolic Approach*. New York, NY, USA: Springer, 2009.
[3] M. Lahijanian, S. B. Andersson, and C. Belta, "Formal verification and synthesis for discrete-time stochastic systems," *IEEE Trans. Autom. Control*, vol. 60, no. 8, pp. 2031–2045, Aug. 2015.
[4] L. Doyen, G. Frehse, G. J. Pappas, and A. Platzer, *Verification of Hybrid Systems*. Cham, Switzerland: Springer, 2018.
[5] C. Baier and J. Katoen, *Principles of Model Checking*. Cambridge, MA, USA: MIT Press, 2008.
[6] G. De Giacomo and M. Y. Vardi, "Linear temporal logic and linear dynamic logic on finite traces," in *Proc. IJCAI*, 2013, pp. 854–860.
[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] 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," *J. Comput. Phys.*, vol. 378, pp. 686–707, Feb. 2019.
[9] T. Wei and C. Liu, "Safe control with neural network dynamic models," 2021, *arXiv:2110.01110*.
[10] M. Wicker, L. Laurenti, A. Patane, N. Paoletti, A. Abate, and M. Kwiatkowska, "Certification of iterative predictions in Bayesian neural networks," in *Proc. UAI*, 2021, pp. 1713–1723.
[11] M. Fazlyab, M. Morari, and G. J. Pappas, "An introduction to neural network analysis via semidefinite programming," in *Proc. CDC*, 2021, pp. 6341–6350.
[12] K. Xu *et al.*, "Automatic perturbation analysis for scalable certified robustness and beyond," 2020, *arXiv:2002.12920*.
[13] R. Givan, S. Leach, and T. Dean, "Bounded-parameter Markov decision processes," *Artif. Intell.*, vol. 122, nos. 1–2, pp. 71–109, 2000.
[14] H. Zhao, X. Zeng, T. Chen, Z. Liu, and J. Woodcock, "Learning safe neural network controllers with barrier certificates," *Formal Aspects Computing*, vol. 33, pp. 437–455, Apr. 2021.
[15] D. P. Bertsekas and S. E. Shreve, *Stochastic Optimal Control: The Discrete-Time Case*. Belmont, MA, USA: Athena Sci., 1996.
[16] A. M. Wells, M. Lahijanian, L. E. Kavraki, and M. Y. Vardi, "LTLf synthesis on probabilistic systems," in *Proc. EPTCS*, 2020, pp. 1–16.
[17] R. T. Rockafellar, *Convex Analysis*. Princeton, NJ, USA: Princeton Univ. Press, 2015.
[18] N. Cauchi, L. Laurenti, M. Lahijanian, A. Abate, M. Kwiatkowska, and L. Cardelli, "Efficiency through uncertainty: Scalable formal synthesis for stochastic hybrid systems," in *Proc. HSCC*, 2019, pp. 240–251.
[19] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[20] J. Jackson, L. Laurenti, E. Frew, and M. Lahijanian, "Formal verification of unknown dynamical systems via Gaussian process regression," 2021, *arXiv:2201.00655*.
[21] M. Dutreix and S. Coogan, "Efficient verification for stochastic mixed monotone systems," in *Proc. ICCPS*, 2018, pp. 150–161.
[22] S. Adams, M. Lahijanian, and L. Laurenti, "Formal control synthesis for stochastic neural network dynamic models," 2022, *arXiv:2203.05903*.
[23] R. Rajamani, *Vehicle Dynamics and Control*. New York, NY, USA: Springer, 2011.
[24] J. Jackson, L. Laurenti, E. Frew, and M. Lahijanian, "Strategy synthesis for partially-known switched stochastic systems," in *Proc. HSCC*, 2021, pp. 1–11.