IEEE Control Systems Letters paper presented at
2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

A Hybrid Neural Network Approach for Adaptive Scenario-based Model Predictive Control in the LPV Framework

Yajie Bao and Javad Mohammadpour Velni

***Abstract*— This paper presents a hybrid neural network (NN) approach for adaptive scenario-based model predictive control (SMPC) design of nonlinear systems in the linear parameter-varying (LPV) framework. In particular, a deterministic artificial neural network (ANN)-based LPV model is learned from data as the nominal model. Then, a Bayesian NN (BNN) is used to describe the mismatch between the plant and the LPV-ANN model. Adaptive scenarios are generated online based on the BNN model to reduce the conservativeness of scenario generation. Moreover, a probabilistic safety certificate is incorporated into the scenario generation by ensuring that the trajectories of scenarios contain the trajectory of the system and that all the scenarios satisfy the constraints with a high probability. Furthermore, conditions for the recursive feasibility of the SMPC are given. Experiments on the closed-loop simulations of a two-tank system demonstrate that the proposed approach can better model the behaviors of nonlinear systems than sole ANN/BNN models can, and the SMPC based on the hybrid NN (HyNN) model can improve the control performance compared to the SMPC with a fixed scenario tree.**

I. Introduction

Linear parameter-varying (LPV) framework has attracted increasing attention for data-driven modeling and learning-based control of complex systems by virtue of modeling nonlinear and/or time-varying dynamics in the linear structure [1]. While learning-based control approaches have been developed using data-driven models [2], identification and controller design are generally separated in the current LPV literature [3]. In particular, the data-driven models are learned by minimizing the prediction errors and validated on a testing set without considering the control performance. Therefore, the identified models with high prediction accuracy are not necessarily good for control design. Moreover, the problem of describing the mismatch between the plant and the learned models has not been investigated, which hinders robust control design and safety guarantee establishment.

Data-driven modeling of nonlinear systems in the LPV framework has been discussed in [3], [4]. Among deterministic approaches, artificial neural networks (ANNs) have proven advantageous in using large amounts of data to learn nonlinear parametric models for fast online evaluation, compared to kernel-based methods with cubic computational complexity to the data size. Moreover, [4] proposed a Bayesian neural network (BNN)-based approach to quantify uncertainties in data-driven models. While several techniques (e.g., restricting model complexity and transfer learning) have been developed to address such problems as the high computational cost and convergence failure of training BNN, LPV-BNN models still pose challenges to be employed for control design due to the increased complexity compared with deterministic (ANN) models. The authors in [2] reduced the computational burden of online SMPC optimization by quantizing the scenarios generated by BNN but may still need offline evaluations of BNN to further reduce the computational cost. Instead, in this paper, we propose using an LPV-ANN model learned from data as the nominal model and then employing BNN to model the mismatch between the nominal model and the plant to facilitate control design without compromising control performance.

The plant-model mismatch is a common problem for model-based control design and is typically assumed to be bounded. Recently, data-driven approaches have been employed to obtain accurate state- and/or input-dependent descriptions of the possibly time-varying plant-model mismatch such that the conservativeness of the mismatch estimation is reduced and the control performance is improved [5], [6]. In particular, Gaussian process (GP) regression and BNN are two widely used probabilistic approaches to model mismatch. Specifically, BNNs treat the model weights of deterministic neural networks (NNs) as random variables with given prior distributions and provide the estimation of the posterior distributions conditioned on a dataset using variational inference. Compared with GP regression, BNNs can model both epistemic and aleatoric uncertainties with arbitrary distributions, be trained efficiently using Bayes by Backprop [7], and be fast evaluated without using the dataset. Therefore, this paper employs BNNs to model the mismatch.

MPC is commonly used for model-based control of a process while satisfying constraints. For MPC design using LPV model descriptions of nonlinear systems, one challenge lies in the uncertainty of the future evolution of the scheduling variable(s) [8]. One practical solution to address this challenge is to assume fixed scheduling variables in the prediction horizons, which may result in large prediction errors and thus degrade the control performance. Therefore, the uncertainty of the scheduling variables must be considered for MPC design. The authors in [2] proposed to generate scenarios that can represent the joint uncertainty of models and scheduling variables based on pure BNN model descriptions for SMPC design. Additionally, using BNNs to model mismatch for SMPC design has been studied in [9]. The BNN was used to model state- and input-dependent uncertainties, and the statistics (including mean and standard deviation) of the BNN predictions were used to generate

This work was financially supported by the United States National Science Foundation under award CMMI-2302219.

Authors are with the Dept. of Mechanical Engineering, Clemson University, Clemson, SC 29634. E-mail: javadm@clemson.edu.

scenarios. However, [9] assumed a given general nonlinear nominal model and did not establish the recursive feasibility for the proposed SMPC scheme due to the high model complexity of BNN. *This paper uses hybrid and pure data-driven models to take advantage of the LPV-MPC approach and establish recursive feasibility for the proposed LPV-SMPC scheme based on a reasonably accurate characterization of the BNN model.*

The main contribution of this paper lies in presenting a hybrid NN (HyNN) approach in the LPV framework for adaptive SMPC design and providing a set of conditions for recursive feasibility and probabilistic safety guarantees of the proposed approach. Remainder of the paper is organized as follows: Section II describes the problem formulation and the HyNN approach for data-driven modeling. The SMPC design using the HyNN model is explained in Section III. Section IV presents the experimental results, and finally concluding remarks are provided in Section V.

II. Problem Formulation

Consider a constrained, discrete-time nonlinear system

$$x(k+1) = f(x(k), u(k)), \tag{1a}$$

$$x \in \mathcal{X}, u \in \mathcal{U}, \tag{1b}$$

where x denotes the states, u denotes the control inputs, and $k \in \mathbb{N}$ denotes the time instant. $\mathcal{X} \subseteq \mathbb{R}^{n_x}$ and $\mathcal{U} \subseteq \mathbb{R}^{n_u}$ in (1b) are the constraint sets of the states and inputs, respectively. Additionally, $\mathcal{X}$ is assumed to be convex. Assuming that (1a) is unknown but a sufficient dataset $\mathcal{D} = \{(x(i), u(i), \theta(i)), x(i+1)\}_{i=1}^{N_\mathcal{D}}$ over $\mathcal{X} \times \mathcal{U} \times \Theta$ can be collected from the system, a data-driven model needs to be learned from $\mathcal{D}$ for MPC design. In particular, we learn a HyNN model composed of a deterministic LPV-ANN nominal model to take advantage of the LPV framework and a stochastic BNN-based residual model to take care of the plant-model mismatch for robust MPC design, as discussed in Section I. SMPC has proven to be efficient for employing a BNN-based mismatch model for control design. This paper aims to design SMPC for the system (1) using the HyNN model and provide the recursive feasibility and probabilistic safety guarantees.

The LPV-ANN nominal model is in the form of

$$\hat{x}(k+1) = A(\theta(k))\, x(k) + B(\theta(k))\, u(k) \triangleq \hat{f}(x(k), u(k)), \tag{2}$$

where $A : \mathbb{R}^{n_\theta} \mapsto \mathbb{R}^{n_x \times n_x}$ and $B : \mathbb{R}^{n_\theta} \mapsto \mathbb{R}^{n_x \times n_u}$ are matrix functions represented by ANNs; θ denote the scheduling variables which can be (nonlinear) functions of inputs/states, but are converted into an exogenous signal by confining the values of θ to some suitable set Θ such that the associated set of admissible trajectories (i.e., the set of input and output signals that are compatible with the dynamics) of (2) is a superset of the set of trajectories of the original nonlinear system (1) [8].

Then, a BNN will be used to model the mismatch between the plant and the LPV-ANN model. In particular, we evaluate the mismatch $g(i) := x(i+1) - \hat{x}(i+1)$ where $\hat{x}(i+1)$ is computed by (2) on the dataset $\mathcal{D}$, to obtain the dataset $\mathcal{D}_g = \{(x(i), u(i), \theta(i)), g(i)\}_{i=1}^{N}$ for training the BNN-based mismatch model. The training and evaluation procedures of BNN can be found in [4]. In this paper, we use a multi-layer, fully-connected BNN to model the unknown vector-valued function g. The BNN is trained by minimizing

$$\frac{1}{N_{\text{BNN}}} \sum_{i=1}^{N_{\text{BNN}}} \left[\log q(w^{(i)}; \zeta) - \log p(w^{(i)}) - \log p(\mathcal{D}|w^{(i)})\right] \tag{3}$$

over ζ via stochastic gradient descent where $w^{(i)}$ are the i-th sample generated by Monte Carlo (MC) for approximating the evidence lower bound (ELBO), and N_{BNN} is the MC sample size determined such that (3) is convergent. Using the trained BNN model, the density of $\hat{g}$ at given $(x(k), u(k))$ can be evaluated by drawing samples from the posteriors of weights and calculating the possible $\hat{g}$'s with each set of sampled weights. To provide safety guarantees, we need reliable estimates of g inside the operating region $\mathcal{X} \times \mathcal{U}$, which is similar to [6] and formally described in the following assumption:

Assumption 1. *For a confidence level $\delta_p \in (0, 1]$, there exists a scaling factor β such that with a probability greater than $1 - \delta_p$,*

$$\begin{aligned} &\forall k \in \mathbb{N}, |g_j(k) - \hat{\mu}_{g_j(k)}| \leq \beta_j \hat{\sigma}_{g_j(k)} < |\mathcal{G}_j|, \\ &j = 1, 2, \cdots, n_x, \end{aligned} \tag{4}$$

given $(x(k), u(k)) \in \mathcal{X} \times \mathcal{U}$, where $\hat{\mu}_{g_j(k)}$ and $\hat{\sigma}_{g_j(k)}$ denote the estimated mean and standard deviation of the j-th entry of $g(k)$, respectively, using the learned BNN model with MC methods, and $|\mathcal{G}_j|$ denotes the maximum value of valid g_j.

By Assumption 1, the learned model is sufficiently accurate such that the values of g are contained in the credible intervals of our probabilistic model. If Assumption 1 does not hold for the trained model, then the model accuracy should be improved by adjusting the model architecture and optimization or collecting more data for training until the hypotheses of Assumption 1 are satisfied.

Next, we introduce the definition of *safety*. Using $\kappa : \mathcal{X} \times \mathbb{N} \to \mathcal{U}$ to denote the SMPC law, the closed-loop system can be described by

$$\begin{aligned} x(k+1) &= \hat{f}(x(k), \kappa(x(k))) + g\left(x(k), \kappa(x(k))\right) \\ &\triangleq \Phi_\kappa(x(k)). \end{aligned} \tag{5}$$

Additionally, we use $\mathbf{x}(k|x(0))$ to denote the solution $\{x(i)|i = 1 \cdots, k\}$ to (5) given the initial state $x(0)$.

Definition 1. *Given $x(0) \in \mathcal{X}$, the system (1a) is said to be safe under a control law κ if for $\forall k \in \mathbb{N}$,*

$$\Phi_\kappa(x(k)) \in \mathcal{X}, \quad \kappa(x(k)) \in \mathcal{U}. \tag{6}$$

Moreover, the system (1a) is said to be δ-safe under the control law κ if $\forall k \in \mathbb{N}$,

$$Pr\left[\Phi_\kappa(x(k)) \in \mathcal{X}, \kappa(x(k)) \in \mathcal{U}\right] \geq \delta, \tag{7}$$

where $Pr[\cdot]$ denotes the probability of an event.

In general, the hard constraints (6) cannot be enforced without additional assumptions [10], especially when (1a) is unknown. However, δ-safety relaxes the requirements of safety to *safety with a high probability*. Furthermore, an input sequence $\mathbf{u}$ is said to be valid for a system with initial state $x(0)$ if applying the input sequence to the system is safe.

Lemma 1. *Given $x(0)$, a valid control input sequence $\mathbf{u}$, a BNN model that fulfills Assumption 1, and a confidence level $\delta_c \in (0,1]$, there exists an $\bar{N}_{MC}$ such that for $\forall k \in \mathbb{N}$ and $j = 1, \cdots, n_x$,*

$$\Pr\left[\mathbf{x}_j(k|x(0)) \in [\hat{\mathbf{x}}_{j,\min}(k|x(0)), \hat{\mathbf{x}}_{j,\max}(k|x(0))]\right] \geq 1 - \delta_c, \tag{8}$$

where $\hat{\mathbf{x}}_{j,\min}(k|x(0)) = \min_i \hat{\mathbf{x}}_j^{(i)}(k|x(0))$ and $\hat{\mathbf{x}}_{j,\max}(k|x(0)) = \max_i \hat{\mathbf{x}}_j^{(i)}(k|x(0))$ with $\hat{\mathbf{x}}_j^{(i)}(k|x(0)) = \hat{f}(\hat{\mathbf{x}}_j^{(l)}(k-1|x(0)), \mathbf{u}(k-1)) + \hat{g}^{(i)}(\hat{\mathbf{x}}_j^{(l)}(k-1|x(0)), \mathbf{u}(k-1))$; $\hat{g}^{(i)}$ is the prediction of g using the i-th sampled model from the BNN model, $i = 1, \cdots, N_{MC}(k)$, $l = 1, \cdots, N_{MC}(k-1)$; $N_{MC}(k)$ is the number of models drawn from the BNN model using MC methods at the time instant k; $\bar{N}_{MC} = \max_k N_{MC}(k)$.

Proof: Let $k = 0$, $\hat{\mathbf{x}}(0|x(0)) = x(0)$. Since $|g_j(x(0), \mathbf{u}(0)) - \hat{\mu}_{g_j(x(0),\mathbf{u}(0))}| \leq \beta_j \hat{\sigma}_{g_j(x(0),\mathbf{u}(0))}$ by Assumption 1, there exists an $N_{\mathrm{MC}}(0)$ such that

$$\begin{aligned}\mathbf{x}_j(1|x(0)) &= \hat{f}_j(x(0), \mathbf{u}(0)) + g_j(x(0), \mathbf{u}(0)) \\ &\in [\hat{\mathbf{x}}_{j,\min}(1|x(0)), \hat{\mathbf{x}}_{j,\max}(1|x(0))],\end{aligned}$$

holds almost surely, i.e., $\delta_c \to 0$. Then, at time instant $k+1$, $N_{\mathrm{MC}}(1)$ can be found such that $\mathbf{x}_j(2|x(0)) \in [\hat{\mathbf{x}}_{j,\min}(2|x(0)), \hat{\mathbf{x}}_{j,\max}(2|x(0))]$, as

$$\begin{aligned}&\mathbf{x}_j(1|x(0)) \in [\hat{\mathbf{x}}_{j,\min}(1|x(0)), \hat{\mathbf{x}}_{j,\max}(1|x(0))], \\ &|g_j(\hat{\mathbf{x}}_j^{(l)}(1|x(0)), \mathbf{u}(1)) - \hat{\mu}_{g_j(\hat{\mathbf{x}}_j^{(l)}(1|x(0)),\mathbf{u}(1))}| \\ &\leq \beta_j \hat{\sigma}_{g_j(\hat{\mathbf{x}}_j^{(l)}(1|x(0)),\mathbf{u}(1))}, l = 1, \cdots, N_{\mathrm{MC}}(1),\end{aligned}$$

and the support of the weights as random variables in the BNN model is unbounded. Through induction, (8) is obtained using $\bar{N}_{\mathrm{MC}} = \max_k N_{\mathrm{MC}}(k)$.

Lemma 1 guarantees that, with a high probability, the state trajectory of the system is always contained in the multiple trajectories simulated by the BNN model, which is used later to establish safety guarantees.

A. Scenario-based MPC Design

Given the probability distribution of uncertainties, the objective of stochastic MPC at the time instant k is

$$\min \mathbb{E}\left\{\sum_{i=0}^{N-1} \ell(x(i|k), u(i|k)) + V_N(x(N|k))\right\} \tag{9}$$

where $\mathbb{E}$ is the expected value operator over the random vector sequence $\mathbf{g} = \{g(0), \cdots, g(N-1)\}$ with g denoting the plant-model mismatch. It is noted that the uncertainties of g are propagated forward through the prediction model (1a) and thus the closed-form probability density function of $\hat{\mathbf{g}}$ is hard to derive. Moreover, (9) is not directly solvable over generic feedback control law $u(k) = \kappa(x(k))$.

To evaluate (9), SMPC represents the uncertainty of a system using a tree of discrete scenarios. Each particular branch stemming from a node represents a scenario/realization of uncertainty [11]. Then, the scenario-based optimal control problem for an uncertain system at time instant k can be formulated as

$$\min_{x^j, u^j} \sum_{j=1}^{S} p^j \left[\sum_{i=0}^{N-1} \ell\left(x^j(i|k), u^j(i|k)\right) + V_N\left(x^j(N|k)\right)\right] \tag{10a}$$

$$\text{s.t.} \quad x^j(i+1|k) = \hat{f}\left(x^j(i|k), u^j(i|k)\right) + \hat{g}^j(i|k), \tag{10b}$$

$$\left(x^j(i|k), u^j(i|k)\right) \in \mathcal{X} \times \mathcal{U}, \tag{10c}$$

$$x^j(0|k) = x(k), \tag{10d}$$

$$u^j(i|k) = u^l(i|k) \text{ if } x^{p(j)}(i|k) = x^{p(l)}(i|k), \tag{10e}$$

where the superscript j indicates the particular scenario $j \in \{1, \ldots, S\}$; p^j denotes the probability of the j-th scenario; $\ell\left(x^j(i|k), u^j(i|k)\right)$ and $V_N\left(x^j(N|k)\right)$ are the stage cost and terminal cost for the trajectory of the j-th scenario, respectively; N is the prediction horizon; $\hat{g}^j$ denotes the mismatch realization based on the BNN model; and (10e) enforces a *non-anticipativity* constraint, which represents the fact that each control input that branches from the same parent node must be equal ($x^{p(j)}(i)$ is the parent state of $x^j(i+1)$). The non-anticipativity constraint is crucial in order to accurately model the real-time decision problem such that the control inputs do not anticipate the future (i.e., decisions cannot realize the uncertainty). The solution to this optimization problem is used to generate the control law,

$$\kappa(x(k)) = \mathbf{u}^\star(0|k). \tag{11}$$

A potential challenge to the scenario-based optimal control problem is the exponential nature of the scenario tree formulation. To combat this, we utilize a method described in [11], in which a robust horizon $N_r < N$ is defined. The scenario tree stops branching beyond the robust horizon, and the uncertainty realizations are assumed to be constant thereafter. Consequently, considering a fixed number of scenarios s at each node, the total number of scenarios is $S = s^{N_r}$. To further save computational cost, we should accurately approximate (9) with a relatively small number of scenarios.

Given the structure of the scenario tree, it is crucial to generate appropriate scenarios at each stage of the optimization to accurately represent the uncertainty of the system. As such, several methods of scenario generation have been proposed in the literature, including Monte Carlo sampling methods [12], moment matching methods [13], and even machine learning techniques [14]. Despite those efforts, the methods are typically only applied to convex problems and assume full recourse. In this paper, we propose an efficient online scenario generation approach and incorporate a probabilistic safety certificate into the scenario generation. In particular, we generate representative scenarios whose behaviors contain the system behaviors. Then, the system is safe under (11) if (10) where all the scenarios are subject to the constraints is feasible. Moreover, the uncertainties considered are state- and input-dependent, which provides the opportunity to adapt the uncertainty estimation and thus the scenarios at each time instant.

III. Learning-based SMPC Design Using BNNs

In this section, we formulate the SMPC using the HyNN model learned from the data. In particular, the scenario generation method is presented, and the probabilistic safety guarantee is provided.

A. Learning-based Scenario Generation

At each time instant k, we draw $\bar{N}_{\text{MC}}$ samples from normal distributions and calculate weights $w^{(i)}$ by applying the reparameterization trick [7] to the i-th sample. While Lemma 1 claims that the trajectories of the $\bar{N}_{\text{MC}}$ sampled models contain the system trajectory, $\bar{N}_{\text{MC}}$ can be too large for online optimization of the SMPC problems. Specifically, $\bar{N}_{\text{MC}} \geq \frac{2}{\delta}((n_x - 1)\ln(2) - \ln(\beta))$ to guarantee confidence level δ with probability $1 - \beta$ [15]. To reduce the number of scenarios, instead, we estimate $g^{(i)}$ using $w^{(i)}$

$$\hat{\mu}_{g(k)} = \frac{1}{\bar{N}_{\text{MC}}} \sum_{i=1}^{\bar{N}_{\text{MC}}} \hat{g}^{(i)}, \tag{12}$$

$$\hat{\sigma}_{g(k)} = \sqrt{\frac{1}{\bar{N}_{\text{MC}}} \sum_{i=1}^{\bar{N}_{\text{MC}}} (\hat{g}^{(i)} - \hat{\mu}_{g(k)})^\top (\hat{g}^{(i)} - \hat{\mu}_{g(k)})}, \tag{13}$$

and use $\hat{\mu}_{g(k)}$, $\hat{\mu}_{g(k)} + m^j \hat{\sigma}_{g(k)}$, $\hat{\mu}_{g(k)} - m^j \hat{\sigma}_{g(k)}, j = 1, \cdots, \frac{S-1}{2}$ where m^j are the tuning multipliers for S uncertainty realizations. It is noted that a larger S improves the representativeness of the scenario tree but also increases the computational cost of the SMPC. To maintain the original statistical properties, the probabilities of scenarios are calculated using the moment matching method [16]. Additionally, the first four central moments are matched.

Remark 1. *It is noted that the computational cost of the proposed scenario generation approach and moment matching method is high when $\bar{N}_{MC}$ is large. However, the computations can be done offline via a uniform-grid approach. Specifically, we discretize $\mathcal{X} \times \mathcal{U}$ using uniform grids, evaluate the BNN model at the grid points for $\bar{N}_{MC}$ times such that Lemma 1 is fulfilled, and solve the moment matching optimization problem. The grid size is determined such that the estimation of $\hat{\mu}_g$ and $\hat{\sigma}_g$ is stable. Thus, the scenarios and the probability of scenarios at (x, u) can be retrieved online from the offline computation results by finding the results at the grid point that is closest to (x, u).*

To save computational cost, we only update the uncertainty estimation every time instant and fix the scenarios over the prediction horizon. In particular, we use the solution $u^*(1|k-1)$ to (10) at $k-1$ and the state $x(k)$ to estimate uncertainty $\hat{g}(k)$ at k, and $\hat{g}(i|k) = \hat{g}(k), i = 0, \cdots, N-1$ for (10) at k. Using the uncertainties estimated at time instant k is more tractable than considering time-varying uncertainties and adaptive scenarios in the prediction horizon, as the uncertainties are input-dependent and the control input sequence in the prediction horizon are decision variables of the SMPC problem. When the uncertainties do not change significantly in the prediction horizon, fixing the uncertainty estimation over the prediction horizon is reasonable and still less conservative than using the worst-case error bounds.

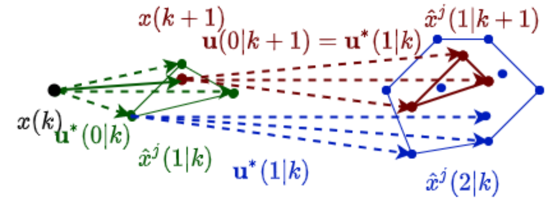


Fig. 1: Recursive feasibility.

B. Recursive Feasibility

Next, we establish the recursive feasibility of the proposed SMPC scheme, which requires a robust controlled invariant terminal set associated with a terminal controller [17], [8].

Assumption 2. *There exists a robust controlled invariant terminal set $\mathcal{X}_f$ for the original nonlinear system* (1) *under the terminal controller $\kappa_f(x) \in \mathcal{U}$ such that for $\forall x \in \mathcal{X}_f$, we have $f(x, \kappa_f(x)) \in \mathcal{X}_f$.*

The terminal set can be under-approximated as the common terminal region $\hat{\mathcal{X}}_f$ [18] of the generated scenarios whose behaviors contain the behaviors of the system, which has been developed in [2].

Assumption 3. *For $\forall k \in \mathbb{N}$, $\forall i = 0, \cdots, N-1$, $\forall l = 1, \cdots, n_x$, we have $|g_l(i|k)| \leq \max\{|\hat{\mu}_{g_l(k)} + m^j \hat{\sigma}_{g_l(k)}|, |\hat{\mu}_{g_l(k)} - m^j \hat{\sigma}_{g_l(k)}| \,|\, j = 1, \cdots, \frac{S-1}{2}\}$.*

Assumption 3 ensures that the uncertainties over the prediction horizon are bounded by the uncertainty estimates using the current state and the optimal control input from the last step. It is noted that Assumption 3 can be fulfilled by tuning m^j. A larger m^j can ensure Assumption 3 holds but increases the conservativeness of the SMPC.

Assumption 4. *Given the set $\mathcal{X}_{i|k}$ of the i-th step predictions of the states at time instant k, applying $\mathbf{u}(i|k)$ results in $\mathcal{X}_{i+1|k}$. Then, for a subset $\mathcal{X}'_{i|k} \subseteq conv(\mathcal{X}_{i|k})$, applying $\mathbf{u}(i|k)$ results in $\mathcal{X}'_{i+1|k} \subseteq conv(\mathcal{X}_{i+1|k})$.*

Lemma 2. *Suppose Assumption 3 & 4 are fulfilled, given $\mathbf{u}^*(i|k) \in \mathcal{U}^N$, then $\mathcal{X}_{i|k+1} \subseteq conv(\mathcal{X}_{i+1|k}) \subseteq \mathcal{X}$ with the control input sequence $\mathbf{u}(i-1|k+1) = \mathbf{u}^*(i|k), i = 1, \cdots, N$.*

Proof: Lemma 2 states that the sets of the states over the prediction horizon at $k+1$ are contained in the convex hull of the corresponding sets at k. At time instant k, since $\mathbf{u}^*(i|k)$ is the solution to (10), $\mathcal{X}_{i+1|k} \in \mathcal{X}$ and thus $\text{conv}(\mathcal{X}_{i+1|k}) \in \mathcal{X}$. Using the scenario generation approach in Section III-A, $x(k+1) \in \text{conv}(\mathcal{X}_{1|k})$ after applying $\mathbf{u}^*(0|k)$ to the system at k. Furthermore, for $\forall x(1|k) \in \mathcal{X}_{1|k}$, applying $\mathbf{u}^*(1|k)$ results in $x(2|k) \in \mathcal{X}_{2|k}$. Hence, $\mathcal{X}_{1|k+1} \in \text{conv}(\mathcal{X}_{2|k})$ with $\text{u}(0|k+1) = \mathbf{u}^*(1|k)$ by Assumption 4. Similarly, we have $\mathcal{X}_{i|k+1} \subseteq \text{conv}(\mathcal{X}_{i+1|k}) \subseteq \mathcal{X}, i = 1, \cdots, N-1$.

Theorem 1 (Recursive feasibility)**.** *Consider system* (1) *under the control law* (11) *by solving* (10) *fulfills Assumptions 1-4. If optimization problem* (10) *is feasible for $x(0)$, then it is feasible for all time instants $k \in \mathbb{N}$, i.e., it is recursively feasible.*

Proof: The proof is done by constructing a candidate solution for each k, which is illustrated in Fig. 1. Let $\{u^*(i|k)\}_{i=0}^{N-1}$ be the minimizer to (10) at k. Applying control input (11) results in the state $x(k+1) \in [\min_j x^j(1|k), \max_j x^j(1|k)] \subseteq \mathcal{X}$. Then, we consider the candidate solution $\{u^*(1|k), \cdots, u^*(N-1|k), \kappa_f\left(x^1(N|k), \cdots, x^S(N|k)\right)\}$ which satisfies the input constraints and results in $x^j(i|k+1), i = 0, \cdots, N, j = 1, \cdots, S$. Using Lemma 2, we have $x^j(i|k+1) \in \mathcal{X}$ when $i < N$ and $x^j(N|k+1) \in \mathcal{X}_f$, which proves recursive feasibility.

C. Probabilistic Safety Guarantee

Using the scenario generation approach in Section III-A, the safety certificate can be formalized into our main result.

Theorem 2. *Let Assumptions 1-4 hold. Then, the system under the scenario-based MPC law is δ-safe if* (10) *is feasible for* $x(0)$.

Proof: By Lemma 1, (8) holds using $\bar{N}_{\mathrm{MC}}$ samples. Consequently, at time instant k, there exist m^j's for the scenario generation using $\hat{\mu}_{g(k)}$ and $\hat{\sigma}_{g(k)}$ estimated from the $\bar{N}_{\mathrm{MC}}$ samples such that the predictions $\hat{x}(k+1)$ by the generated scenarios contain the real $x(k+1)$ of the system under Assumption 1. Furthermore, (10) is recursively feasible by Theorem 1, and thus (7) holds for all k, which proves the system is δ-safe by Definition 1.

IV. Closed-loop Simulations and Validation

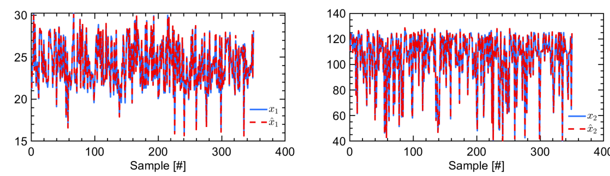
In this section, we validate the proposed HyNN-based control design approach using simulations of a cascaded two-tank system [19]. The system is described by

$$\rho S_1 \dot{h}_1 = -\rho A_1 \sqrt{2gh_1} + u, \tag{14a}$$

$$\rho S_2 \dot{h}_2 = \rho A_1 \sqrt{2gh_1} - \rho A_2 \sqrt{2gh_2}, \tag{14b}$$

where $\rho = 0.001\ \mathrm{kg} \cdot \mathrm{cm}^{-3}$ is the liquid density; $S_1 = 2500\ \mathrm{cm}^2$, $S_2 = 1600\ \mathrm{cm}^2$, $A_1 = 9\ \mathrm{cm}^2$, and $A_2 = 4\ \mathrm{cm}^2$ are the cross-sectional areas of the upper tank, the lower tank, the pipe through which the liquid flows into the lower tank, and the pipe through which the liquid flows out, respectively; h_1 and h_2 denote the liquid levels of the upper and lower tanks, respectively; u denotes the flow of liquid pumped into the upper tank. The control objective is to regulate the levels h_1 and h_2 at given set points. u is available as a control input and subject to the constraint $\mathbb{U} = \{u | 0\ \mathrm{kg.s}^{-1} \leq u \leq 4\ \mathrm{kg.s}^{-1}\}$. The liquid levels satisfy the bounds $\mathcal{X} = \{x = [h_1, h_2]^{\mathrm{T}} | 1\ \mathrm{cm} \leq h_1 \leq 35\ \mathrm{cm}, 10\ \mathrm{cm} \leq h_2 \leq 200\ \mathrm{cm}\}$. The system model (14) is assumed to be unknown for control design and only used for simulations. In the simulations, the goal is to reach a reference value $h_2^* = 115$ cm of the lower tank. Moreover, the translated state and input variables $\tilde{x} = x - [22.72, 115]^{\mathrm{T}}$ and $\tilde{u} = u - 1.90$ are introduced to convert the problem into a stabilization problem.

1) Data-driven Modeling: We apply a random input signal drawn from uniform distribution $U[0, 4]$ to collect observations for model identification. The sampling time is 0.9 seconds. Furthermore, 1000 samples are collected and split into training and testing sets with a ratio of 65%/35%. Since we assume (14) is unknown, we cannot choose the scheduling variables and transform (14) into an exact LPV embedding as [19], and thus we cannot use the approach in [19] for control design. Instead, we simply use the states as the scheduling variables to learn the LPV-ANN nominal model (2), and then treat the scheduling variables as free variables in the prediction horizon of SMPC. In particular, we use a five-layer fully-connected ANN to represent $A(\cdot)$. All the hidden layers of the ANN have 32 hidden units. Moreover, we use one dense layer with 2 hidden units to represent $B(\cdot)$, and the dense layer does not use bias terms. Furthermore, we use a five-layer BNN to model the mismatch between the plant and the nominal model. All the hidden layers of the BNN are dense layers with 16 hidden units while the output layer of the BNN is a DenseVariational layer [4]. Additionally, all the hidden layers in the experiments use exponential linear unit (ELU) activation functions while the output layers do not use any activation function. Adam optimizer is used with a learning rate set to 0.001 and other hyper-parameters as default. Fig. 2 shows the validation of the nominal model, and Fig. 3 shows that the mismatches are contained in the bounds of the BNN predictions, which indicates Assumption 1 was fulfilled.



(a) $\mathrm{BFR}_{x_1} = 92.82\%$. (b) $\mathrm{BFR}_{x_2} = 97.77\%$.

Fig. 2: Validation of the nominal LPV-ANN model.

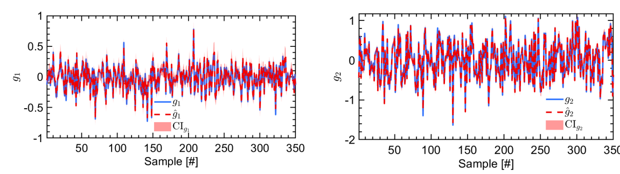
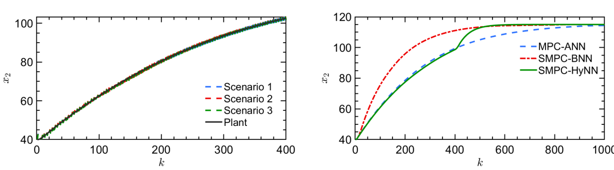


Fig. 3: Validation of the BNN-based mismatch model.

2) Validation of the Proposed SMPC Scheme: To demonstrate the efficiency of the proposed approach, we examine the performance of variants of LPV-MPC. In the first case, we examine the MPC only using the LPV-ANN nominal model. In the second case, we examine the performance of SMPC using an LPV-BNN model [2] of the original system. The LPV-BNN model was composed of the same number of hidden layers and units as the LPV-ANN nominal model and was directly learned from $\mathcal{D}$. In the final case, we consider the proposed SMPC approach using the HyNN model. In our comparison, we used the prediction horizon of $N = 4$ and the robust horizon of $N_r = 1$ for the SMPC. The stage cost was $\ell = \sum_{i=1}^{N-1} x(i|k)^\top x(i|k) + 10\Delta u^2(i|k)$ where $\Delta u(i|k) = u(i|k) - u(i-1|k)$. The terminal set and terminal controller were designed using the approach in [2]. The conservativeness of the LPV-BNN model affects the volume of the terminal set and thus the SMPC performance.

For the SMPC using HyNN, we sampled $\bar{N}_{\text{MC}} = 10$ models to estimate the mean $\hat{\mu}_g$ and standard deviation $\hat{\sigma}_g$ of the mismatch g. Increasing the number of sampled models can improve control performance but also increase the computational cost. Subsequently, at each node of the scenario tree, we used $\hat{\mu}_g$, $\hat{\mu}_g + 0.6\hat{\sigma}_g$, and $\hat{\mu}_g - 0.6\hat{\sigma}_g$ as three discrete scenarios of the plant-model mismatch. Furthermore, we set $|g_1| \leq 0.8$ and $|g_2| \leq 3.0$ based on $\max_i |g_j^{(i)}|, j = 1, 2$ in the dataset $\mathcal{D}_g$. When the predictions of the scenarios are out of the bounds of g due to the limited generalization of the BNN model, we use the bounds instead of the predictions and uniform distribution as the probability of scenarios to avoid too conservative uncertainty estimation. In our simulation, the bounds were only used for 0.4% of the time instants, which demonstrates the usefulness of the BNN-based mismatch model. The system remained in the terminal set under the terminal control law after entering the terminal set at time instant $k = 400$, which indicates Assumption 2 was fulfilled. Moreover, the trajectory of the plant was bounded by the trajectories of the scenarios which demonstrates that Assumptions 3 & 4 were fulfilled. Increasing m^j can increase the probability of safety but reduce the feasible domain of the optimization problem (10) and thus decrease the control performance. Furthermore, the SMPC was indeed feasible throughout the simulations. Fig. 4(b) shows the designed MPC can bring the liquid level h_2 of the lower tank to the reference value while satisfying the system constraints. The proposed SMPC approach with HyNN (green line) achieved better control performance than the other LPV-MPC approaches. While the performance of SMPC-BNN is close to that of SMPC-HyNN, SMPC-BNN used a more complex BNN and was more computationally intensive than SMPC-HyNN.



(a) Trajectories of the scenarios and plant. (b) Performance comparison of variants of LPV-MPC.

Fig. 4: Validation of the proposed SMPC scheme.

V. Concluding Remarks

In this paper, a hybrid NN approach was proposed for adaptive SMPC of nonlinear systems in the LPV framework with recursive feasibility and probabilistic safety guarantees. In particular, an LPV-ANN model was first learned from data as the nominal model, and then a BNN was used to model the mismatch between the original system and the nominal model. The BNN-based mismatch model was later used to generate scenarios online for SMPC. To ensure safety, the behaviors of the generated scenarios contained the system behavior with high probability, and the constraints were enforced for all the scenarios. Moreover, a robust controlled invariant set was employed to establish recursive feasibility. The closed-loop simulations on a two-tank model demonstrated that the proposed approach could improve model accuracy and control performance compared with SMPC only using ANNs and BNNs.

References

[1] Y. Bao and J. Mohammadpour Velni, "An overview of data-driven modeling and learning-based control design methods for nonlinear systems in LPV framework," in *Proc. of the 5th IFAC Workshop on Linear Parameter Varying Systems*, 2022.

[2] Y. Bao, H. S. Abbas, and J. Mohammadpour Velni, "A learning- and scenario-based mpc design for nonlinear systems in LPV framework with safety and stability guarantees," *International Journal of Control*, 2023.

[3] P. B. Cox, "Towards efficient identification of linear parameter-varying state-space models," Ph.D. dissertation, Ph. D. dissertation, Eindhoven University of Technology, 2018.

[4] Y. Bao, J. Mohammadpour Velni, and M. Shahbakhti, "Epistemic uncertainty quantification in state-space LPV model identification using bayesian neural networks," *IEEE Control Systems Letters*, vol. 5, no. 2, pp. 719–724, 2021.

[5] A. D. Bonzanini, J. A. Paulson, G. Makrygiorgos, and A. Mesbah, "Fast approximate learning-based multistage nonlinear model predictive control using gaussian processes and deep neural networks," *Computers & Chemical Engineering*, vol. 145, p. 107174, 2021.

[6] Y. Bao, K. J. Chan, A. Mesbah, and J. Mohammadpour Velni, "Learning-based adaptive-scenario-tree model predictive control with improved probabilistic safety using robust bayesian neural networks," *International Journal of Robust and Nonlinear Control*, vol. n/a, no. n/a. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rnc.6560

[7] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural network," in *International Conference on Machine Learning*. PMLR, 2015, pp. 1613–1622.

[8] J. Hanema, "Anticipative model predictive control for linear parameter-varying systems," *[Doctoral dissertation]. Technische Universiteit Eindhoven*, 2018.

[9] Y. Bao, K. J. Chan, A. Mesbah, and J. Mohammadpour Velni, "Learning-based adaptive-scenario-tree model predictive control with probabilistic safety guarantees using bayesian neural networks," in *2022 American Control Conference (ACC)*. IEEE, 2022, pp. 3260–3265.

[10] T. Koller, F. Berkenkamp, M. Turchetta, and A. Krause, "Learning-based model predictive control for safe exploration," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 6059–6066.

[11] S. Lucia, T. Finkler, and S. Engell, "Multi-stage nonlinear model predictive control applied to a semi-batch polymerization reactor under uncertainty," *Journal of Process Control*, vol. 23, no. 9, pp. 1306–1319, 2013.

[12] A. Shapiro, "Monte carlo sampling methods," *Handbooks in operations research and management science*, vol. 10, pp. 353–425, 2003.

[13] K. Høyland, M. Kaut, and S. W. Wallace, "A heuristic for moment-matching scenario generation," *Computational optimization and applications*, vol. 24, no. 2, pp. 169–185, 2003.

[14] B. Defourny, "Machine learning solution methods for multistage stochastic programming," *PhD diss., University of Liege. https://www. lehigh. edu/defourny/PhDthesis_B_Defourny. pdf*, 2010.

[15] G. Calafiore and M. C. Campi, "Uncertain convex programs: randomized solutions and confidence levels," *Mathematical Programming*, vol. 102, no. 1, pp. 25–46, 2005.

[16] K. Høyland and S. W. Wallace, "Generating scenario trees for multistage decision problems," *Management science*, vol. 47, no. 2, pp. 295–307, 2001.

[17] G. C. Calafiore and L. Fagiano, "Stochastic model predictive control of LPV systems via scenario optimization," *Automatica*, vol. 49, no. 6, pp. 1861–1866, 2013.

[18] M. Maiworm, T. Bäthge, and R. Findeisen, "Scenario-based model predictive control: Recursive feasibility and stability," *IFAC-PapersOnLine*, vol. 48, no. 8, pp. 50–56, 2015.

[19] J. Hanema, R. Tóth, and M. Lazar, "Stabilizing non-linear model predictive control using linear parameter-varying embeddings and tubes," *IET Control Theory & Applications*, vol. 15, no. 10, pp. 1404–1421, 2021.