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

Uncertainty Quantification for Learning-based MPC using Weighted Conformal Prediction

Kong Yao Chee, M. Ani Hsieh, George J. Pappas

Abstract— **Nonlinear model predictive control (MPC) is an established control framework that not only provides a systematic way to handle state and input constraints, but also offers the flexibility to incorporate data-driven models. With the proliferation of machine learning techniques, there is an uptrend in the development of learning-based MPC, with neural networks (NN) being an important cornerstone. Although it has been shown that NNs are expressive enough to model the dynamics of complex systems and produce accurate state predictions, these predictions often do not include uncertainty estimates or have practical finite sample guarantees. In contrast to existing work that either requires the data samples to be exchangeable or relies on properties of the underlying data distribution, we propose an approach that utilizes weighted conformal prediction to alleviate these assumptions and to synthesize provably valid, finite-sample uncertainty estimates for data-driven dynamics models, in a distribution-free manner. These uncertainty estimates are generated online and incorporated into a novel uncertainty-aware learning-based MPC framework. Through a case study with a cartpole system controlled by a state-of-the-art learning-based MPC framework, we demonstrate that our approach not only provides well-calibrated uncertainty estimates, but also enhances the closed-loop performance of the system.**

I. INTRODUCTION

With an increase in data availability, the field of learning-based model predictive control (MPC) is expanding rapidly [1], [2]. A prominent direction within this field is to utilize tools from machine learning to model the dynamics of the system, before applying them within an MPC framework [2]. These learning tools are leveraged to reduce the discrepancy between the true system dynamics and the model, by utilizing data or measurements collected from the system. With more accurate dynamics models, the closed-loop performance of the system, under model-based control frameworks such as nonlinear MPC, has the potential to improve substantially [1].

Related works: The approaches developed to learn dynamics models can be broadly classified into parametric and non-parametric methods. Within the class of non-parametric methods, Gaussian process (GP) regression [3] stands out as a popular choice. For instance, GPs are used in [4] and [5] to model vehicle dynamics, before they are integrated into an MPC framework. Although these models provide uncertainty estimates, a well-known drawback of GPs is their relatively high computational cost, which is detrimental when working with large datasets [3].

On the other hand, a parametric model, such as a neural network (NN), can be used to model the dynamics of nonlinear systems. In [6], a feedforward NN is used to model the dynamics of a reactor, before applying it in an MPC framework. Other NN architectures, *e.g.*, recurrent NNs, have also been utilized to model system dynamics [7], [8], [9]. In these works, since the NNs are used to model the full system dynamics, they tend to be architecturally complex. This incurs significant computational costs when used in conjunction with nonlinear MPC, where a nonlinear optimization problem needs to be solved at every time step. One viable solution is to use NNs to model just the unknown dynamics, instead of the full dynamics. In previous works [10] and [11], a neural ordinary differential equation (NODE) model is used to model the unknown dynamics of the system. Experimental results have shown that this approach improves the performance of the system [10].

While it has been demonstrated that NNs have the ability to model dynamical systems accurately, many of the conventional NN architectures lack the ability to quantify uncertainty about their predictions. Within the machine learning community, there have been attempts to mitigate this, such as incorporating dropout [12], or by constructing Bayesian NNs [13] or deep ensembles [14]. These methods are intuitive, but they often do not come with theoretical guarantees.

A recent approach to quantify uncertainty for NNs is to use tools from the conformal prediction literature [15]. By leveraging statistical properties inferred from the data, conformal prediction not only give estimates of the uncertainty, but also provide probabilistic coverage guarantees for these estimates. This implies that the predictions are guaranteed to lie within a neighborhood of the true values, with high probability. A key assumption in the formulation of conformal prediction is that the data samples are assumed to be *exchangeable*, *i.e.*, the joint probability distribution of the data samples remains the same under any permutation [15]. This is a challenging assumption to make, especially for time-series data. A few methods have been proposed in recent years to alleviate this assumption. In [16], an adaptive conformal inference algorithm is proposed, where the miscoverage rate is updated based on the estimates of the historical miscoverage frequency. In [17], an ensemble of models is used to extract uncertainty predictions and do not require the samples to be exchangeable. In [18], a set of methods that utilize a concept of weighted quantiles is proposed, which alleviates exchangeability in a natural way.

This work was supported by Office of Naval Research (ONR) Award No. N00014-22-1-2157 and DSO National Laboratories, 12 Science Park Drive, Singapore 118225.

The authors are with the University of Pennsylvania, Philadelphia, PA 19104, USA. {ckongyao, m.hsieh, pappasg}@seas.upenn.edu

2023 62nd IEEE Conference on Decision and Control (CDC) | 979-8-3503-0124-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/CDC49753.2023.10383587

In particular, the trade-off required for nonexchangeability is dependent on the amount of distribution shift between the distributions of the training and test data samples.

In the context of learning-based MPC, there are a couple of recent works that incorporate conformal predictions for uncertainty quantification. In [19], conformal prediction is used in conjunction with recurrent NNs to generate prediction regions, which are used as constraints within an MPC scheme. In this work, it is assumed that the data samples are exchangeable. In [20], adaptive conformal inference [16] is introduced to alleviate exchangeability, before integrating the uncertainty estimates as part of the constraints.

In contrast to these works, we utilize the concept of weighted quantiles [18] to counteract the effects of distribution shifts and to ease the exchangeability assumption. Additionally, we present a holistic integration of the uncertainty quantification procedure and the model predictive controller by propagating the uncertainty estimates forward in time, and by incorporating them into the cost function of the MPC scheme. A third distinction between these works and our work is that while our framework is flexible and allows uncertainty quantification of either the full dynamics or just the unknown dynamics, we focus on the latter as it allows the prediction intervals to be small, as observed in our experiments in Section V.

Contributions: Our contributions in this work are threefold. First, we utilize conformal prediction [15] with the concept of weighted quantiles [18], collectively referred to as *weighted conformal prediction*, to synthesize provably valid, finite-sample, uncertainty estimates for the predictions of a data-driven dynamics model, in a distribution-free manner. We incorporate these tools into an algorithm, where the uncertainty estimates are generated in an online manner. Second, we propose a novel, uncertainty-aware learning-based MPC framework, where the uncertainty estimates are propagated forward in time using the unscented transformation (UT) [21], with an augmentation of the cost function. To the best of the authors' knowledge, this is the first work that utilizes weighted conformal prediction within a learning-based MPC framework. Third, we verify the efficacy of the proposed framework by considering a case study, where a nonlinear cartpole system is controlled using a learning-based MPC framework, KNODE-MPC. Importantly, we demonstrate through this case study that the uncertainty estimates synthesized by our approach are well calibrated and the proposed framework enhances the closed-loop performance of the system.

Notation: The sets $\mathbb{R}$, $\mathbb{N}$ and $\mathbb{N}_+$ denote the sets of real numbers, non-negative and positive natural numbers. For a vector $x := [x^{(1)} \ \dots \ x^{(n)}]^\top \in \mathbb{R}^n$ and matrix $A \in \mathbb{R}^{n\times n}$, $||x||_A^2$ denotes $x^\top Ax$, while $||x||_1$ and $||x||_2$ denote its 1-norm and 2-norm. The matrix $B := \text{diag}\left(x^{(1)}, \dots, x^{(n)}\right) \in \mathbb{R}^{n\times n}$ is a diagonal matrix with $\{x^{(1)}, \dots, x^{(n)}\}$ as the diagonal components and zero elsewhere. For a number $c \in \mathbb{R}$, $\lceil c \rceil$ denotes the ceiling of c. For a set $\mathcal{D}$, the function $\mathbb{1} : \mathcal{D} \to \{0, 1\}$ denotes the indicator function and $|\mathcal{D}|$ denotes its cardinality.

II. PROBLEM FORMULATION AND LEARNING-BASED MPC

We consider a nonlinear, discrete-time system with partially known dynamics,

$$x^+ = f\left(x,\, u,\, d(x,\, u)\right) \tag{1}$$

where $x,\, x^+ \in \mathbb{R}^n$ are the current and successor states and $u \in \mathbb{R}^m$ is the control input. The function $f : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}^n$ denotes the full dynamics of the system, and the function $d : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^p$ represents the unknown dynamics.

The first step in the application of a learning-based MPC framework is to synthesize a data-driven dynamics model. We consider the utility of a differentiable, parametric model, *e.g.*, a feedforward NN, to parameterize either the full dynamics f or the unknown dynamics d. The parameters of this model, θ, are trained using a backpropagation procedure [22]. This is done by leveraging data collected from the system, denoted by $\mathcal{O} := \{(x_i,\, u_i)\}_{i=1}^T$, where T denotes the number of data points, and through the formulation of a loss function, $\mathcal{L}(\theta)$. After training, we obtain a learned model $\hat{f}(x, u;\, \theta^\star)$ that represents the full dynamics in (1). This model acts as the dynamics model within the learning-based MPC framework.

In this framework, the following finite-horizon constrained optimization problem is solved at each time step $k \in \mathbb{N}$,

$$\begin{aligned}
\underset{U}{\text{minimize}} \quad & \sum_{i=0}^{N-1} ||x_i - x_{r,i}||_Q^2 + ||u_i||_R^2 && \text{(2a)}\\
& + ||x_N - x_{r,N}||_P^2 && \text{(2b)}\\
\text{subject to} \quad & x_{i+1} = \hat{f}(x_i, u_i;\, \theta^\star), \quad \forall\, i \in [0,\, N-1] && \text{(2c)}\\
& x_i \in \mathcal{X}, \quad u_i \in \mathcal{U}, \quad \forall\, i \in [0,\, N-1] && \text{(2d)}\\
& x_N \in \mathcal{X}_f, \quad x_0 = x(k), && \text{(2e)}
\end{aligned}$$

where $N \in \mathbb{N}_+$ is the prediction horizon and $\mathcal{X},\, \mathcal{X}_f \subseteq \mathbb{R}^n, \mathcal{U} \subseteq \mathbb{R}^m$ are sets in which state and control input constraints are defined. The matrices $Q,\, P \in \mathbb{R}^{n\times n}$ and $R \in \mathbb{R}^{m\times m}$ penalize the stage and terminal costs, and the control input cost respectively. The sequence $\{x_{r,0}, \dots, x_{r,N}\}$ denotes reference states, and the vector $x(k) \in \mathbb{R}^n$ is the state measurement obtained at each time step k. Upon solving (2), we obtain a sequence of optimal control inputs, $U^\star(x(k)) := \{u_0^\star, \dots, u_{N-1}^\star\}$. The first vector in this sequence $u_0^\star(x(k)) \in \mathbb{R}^m$ is then applied as the control action. This proceeds in a receding horizon manner, as the prediction horizon shifts forward at each time step.

Problem 1: Although the learned model $\hat{f}(x, u;\, \theta^\star)$ may provide accurate predictions of the system dynamics given sufficient data and after adequate training, there are no formal guarantees that the predicted states will be close to the true states, as depicted by the dashed line in Fig. 1. In the case where the predictions deviate from the true states significantly, the performance of the closed-loop system under the learning-based MPC framework is likely to degrade substantially. Hence, to achieve guarantees for the predictions, we aim to find an uncertainty region such that

$$\mathbb{P}\left\{ f\left(x, u, d(x, u)\right) \in C(x, u, \hat{f}(x, u;\, \theta^\star)) \right\} \geq 1 - \delta,$$

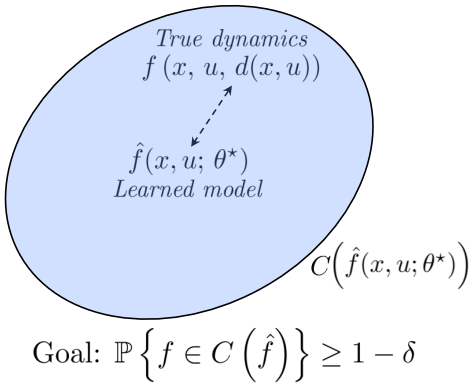

Fig. 1. **Problem setting:** Given that there are no guarantees that $\hat{f}$ is close to f, the goal is to find an uncertainty region $C(\hat{f})$ such that it is guaranteed to contain the true dynamics f with high probability.

where the set C represents the uncertainty region that is guaranteed to contain the true dynamics with a high probability of $1-\delta$. This problem setting is illustrated in Fig. 1.

To solve Problem 1, our approach is to utilize *weighted conformal prediction* to synthesize provably valid uncertainty estimates for the predictions of the learned model. These uncertainty estimates are constructed in the form of prediction intervals and importantly, the true states are *guaranteed* to lie within these intervals with high probability. This not only enables us to quantify the accuracy and uncertainty about the predictions, but also provides an avenue to enhance uncertainty awareness for learning-based MPC.

The rest of the paper is organized as follows. Conformal prediction, and in particular, weighted conformal prediction are described in Section III. In Section IV, we discuss how the uncertainty estimates can be incorporated into the learning-based MPC framework. In Section V, the proposed framework is analyzed through a case study and a conclusion is drawn in Section VI. A schematic of the overall framework is depicted in Fig. 2.

III. UNCERTAINTY QUANTIFICATION

A. Conformal Prediction

Conformal prediction (CP) is a statistical method that can be applied to a given regression model to produce prediction intervals [15]. While there exist other methods to quantify uncertainty, *e.g.*, quantile regression [23], the estimates obtained from these methods either do not possess any coverage guarantees, *i.e.*, guarantees in which the predictions lie within an interval, or require additional assumptions on the underlying distribution. This motivates the development of conformalized prediction intervals, and in particular, the split conformal prediction (SCP) algorithm [15].

Given a dataset of M samples, $\mathcal{D} := \{(z_i,\, y_i)\}_{i=1}^{M}$, where $\{z_i\}_{i=1}^{M}$ and $\{y_i\}_{i=1}^{M}$ are the features and labels of the dataset, the objective of the SCP algorithm is to generate a conformalized prediction interval $C(z)$ such that for a newly given test data sample $z_{M+1} = z$, its label y_{M+1} lies within the prediction interval, with a pre-specified probability.

The algorithm starts by splitting $\mathcal{D}$ into two disjoint sets, $\mathcal{D}_1$ and $\mathcal{D}_2$, indexed by $\mathcal{I}_1$ and $\mathcal{I}_2$ respectively. Next, a regression model is learned using the first set, $\mathcal{D}_1$. Using the second set $\mathcal{D}_2$ and the learned regression model, a set of mean predictions $\{\hat{y}_i(z_i)\}_{i=1}^{p}$ is computed, together with a set of nonconformity scores $S := \{S_i\}_{i=1}^{p}$ where

$$S_i(z_i) := ||y_i - \hat{y}_i(z_i)||_1, \tag{3}$$

and $p := |\mathcal{I}_2|$ denotes the size of $\mathcal{D}_2$. The nonconformity score accounts for the deviation between the predictions given by the regression model and the true labels in an intuitive way. The larger the nonconformity scores, the less accurate the predictions are. This score acts as a measure of the uncertainty about the predictions. Next, given a target miscoverage rate $\alpha \in (0, 1)$, we compute the $(1-\alpha)^{\text{th}}$ empirical quantile of S, denoted by $Q_{1-\alpha}(S)$. This quantile is also the $\lceil (1-\alpha)(1+p) \rceil$ th-smallest value of S [18]. Equivalently, it can be expressed as

$$Q_{1-\alpha}(S) := Q_{1-\alpha}\left(\sum_{i=1}^{p} \frac{1}{1+p}\delta_{S_i} + \frac{1}{1+p}\delta_{+\infty}\right), \tag{4}$$

where δ_a denotes a probability point mass for a real number a on the extended real line, *i.e.*, $a \in \mathbb{R} \cup \{-\infty, +\infty\}$ [18]. Finally, the following conformalized interval is attained by expanding symmetrically about the mean prediction,

$$\begin{aligned} C(z) &:= [q_l(z),\, q_u(z)] \\ &= [\hat{y}(z) - Q_{1-\alpha}(S),\, \hat{y}(z) + Q_{1-\alpha}(S)]\,. \end{aligned} \tag{5}$$

This interval can be applied to any test data sample $z_{M+1} = z$. Overall, this conformalization procedure uses $\mathcal{D}_2$, also known as the calibration set, to compute the width of the prediction intervals such that coverage guarantees are attained. Formally, these conformalized intervals satisfy the following coverage guarantees, first described in [24].

Theorem 1: (*Exchangeable CP*, [18]) If the data samples $\{(z_i,\, y_i)\}_{i=1}^{p+1}$ are independent and identically distributed (*i.i.d.*), the prediction interval $C\,(z_{p+1})$ produced by the SCP algorithm satisfies

$$\mathbb{P}\,\{y_{p+1} \in C\,(z_{p+1})\} \geq 1-\alpha, \tag{6}$$

where $\alpha \in (0,\, 1)$ is a specified miscoverage rate.

Proof: Refer to Theorem 1 in [18] and Proposition 1 in [24]. ∎

In general, the *i.i.d.* assumption in Theorem 1 can be subsumed under the property of *exchangeability* [15]. This implies that the joint probability distribution of the dataset $\{(z_i,\, y_i)\}_{i=1}^{p+1}$ and consequently the nonconformity scores $\{S_i\}_{i=1}^{p+1}$, has the same distribution as $\{S_{\sigma(i)}\}_{i=1}^{p+1}$, under any permutation σ of the indices $1, \ldots, p+1$. For time series data, requiring the data points to be i.i.d. or exchangeable is a strong assumption. Obtaining coverage guarantees for the prediction intervals for time series data in this nonexchangeable setting is an active research direction with some recent developments reported in [16], [17] and [18]. In this work, we deploy the concept of weighted quantiles [18] to alleviate the exchangeability assumption.

B. Weighted Conformal Prediction

To mitigate the effects of distribution shifts, weights $\{w_1, \ldots, w_p\} \in [0,\, 1]$ can be introduced into the formulation of the empirical quantiles, as described in [18]. This is motivated by the fact that a higher weight should be given to

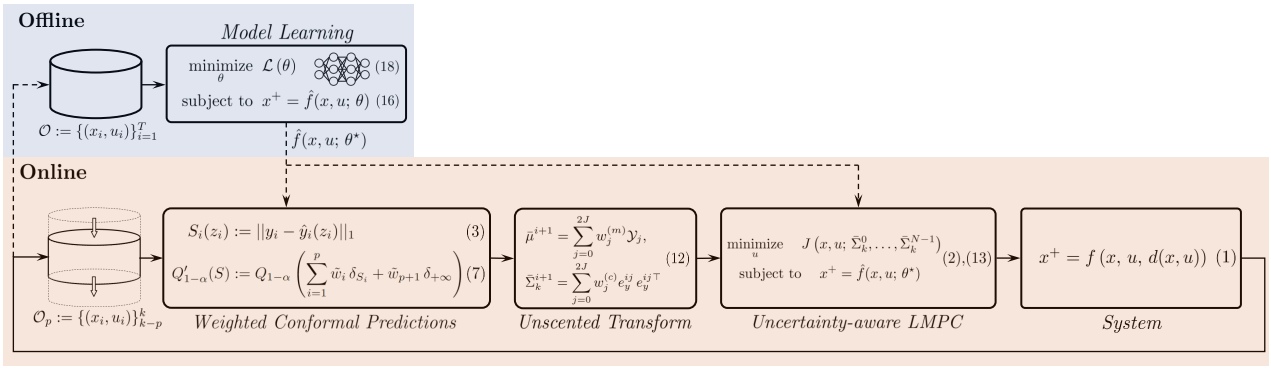

Fig. 2. **Overall schematic of the proposed framework**. *Offline (blue):* The dynamics model $\hat{f}(x,u;\theta)$ is learned via an optimization procedure, using the dataset $\mathcal{O}$ that is collected before deployment. The dashed lines denote signals activated before deployment. *Online (orange):* During deployment, the optimized model $\hat{f}(x,u;\theta^\star)$, together with a moving window of data samples $\mathcal{O}_p$, are utilized to obtain uncertainty estimates via weighted conformal prediction and the unscented transform, before incorporating them into the uncertainty-aware learning-based MPC (LMPC) framework.

a data sample that comes from a distribution that is similar to the distribution of the training dataset. This implies that the empirical quantile in (4) is modified to be

$$Q'_{1-\alpha}(S) := Q_{1-\alpha}\left(\sum_{i=1}^{p} \tilde{w}_i\,\delta_{S_i} + \tilde{w}_{p+1}\,\delta_{+\infty}\right), \quad (7)$$

where

$$\tilde{w}_i := \frac{w_i}{w_1 + \ldots + w_p + 1}, \quad i = 1,\ldots,p, \qquad \tilde{w}_{p+1} := \frac{1}{w_1 + \ldots + w_p + 1} \quad (8)$$

are a set of normalized weights [18]. If the weights are chosen such that $w_i := 1, i = 1,\ldots,p$, the empirical quantile in (4) is recovered. With this modification, we have the following coverage guarantees in the setting when the data samples are nonexchangeable.

Theorem 2: (*Nonexchangeable CP*, [18]) For a given dataset $\mathcal{D}_c := \{(z_i,\, y_i)\}_{i=1}^{p+1}$ and a pre-trained regression model, the nonexchangeable SCP algorithm satisfies

$$\mathbb{P}\{y_{p+1} \in C(z_{p+1})\} \geq 1 - \alpha - \sum_{i=1}^{p} \tilde{w}_i\, d_{TV}\left(S_c,\, S^i\right), \quad (9)$$

where S_c denotes the set of nonconformity scores for $\mathcal{D}_c$, and S^i are the nonconformity scores when the data sample $(z_{p+1},\, y_{p+1})$ is swapped with the i^{th} sample $(z_i,\, y_i)$ in S_c. The metric d_{TV} denotes the total variation between distributions.

Proof: Refer to Theorem 2 in [18]. ∎

With this result, the following corollary shows that the true dynamics $f(x,u,d(x,u))$ is guaranteed to lie within a neighborhood of the learned model $\hat{f}(x,u,\theta^\star)$, providing a solution to *Problem 1*.

Corollary 1: Given a dataset $\{(x_i,u_i)\}_{i=1}^{p+1}$ and the learned model $\hat{f}(x,u,\theta^\star)$, it holds that

$$\mathbb{P}\left\{f(x,u,d(x,u)) \in C\left(x,u;\hat{f}(x,u;\theta^\star)\right)\right\} \geq 1-\delta, \quad (10)$$

where $\delta = \alpha + \sum_{i=1}^{p} \tilde{w}_i\, d_{TV}\left(S_c,\, S^i\right)$.

Proof: We define the state and control inputs (x_i,u_i) and the successor states x_{i+1} to be the features z_i and labels y_i, and let $\hat{f}(x,u,\theta^\star)$ be the regression model for the nonexchangeable CP algorithm. An application of Theorem 2 gives the required result. ∎

While it can be challenging to compute d_{TV} since the distributions are unknown, this set of results offers a number of important insights [18]. When the data points are exchangeable, then $S_c \overset{d}{=} S^i$, in distribution. This implies that the same coverage guarantees in Theorem 1 are attained. Hence, this weighting procedure does not affect coverage, when the data points are indeed exchangeable. While there are no assumptions on the type of distributions, there is a trade-off in the selection of the weights $\{w_i\}_{i=1}^{p}$. If they are chosen to be large, the effects from distribution shifts may not be effectively alleviated. Conversely, if they are small, the quantiles computed with (7) may be overly conservative. In time series data, if the distribution shift occurs with respect to time, the weights can be chosen such that they promote recency, *i.e.*, $w_p \geq \ldots \geq w_1$. One possible approach to select the weights is to apply a geometric decay, *i.e.*, $w_i := \rho^{n-i+1}$, where n is the size of the calibration dataset and $\rho \in (0,1)$, as proposed in [18]. It is important to note that even though the term d_{TV} is difficult to compute, we can use the weights w_i to control or adjust the last term in (10) such that the desired coverage is achieved.

The proposed approach assumes a window of historical data samples, which implies that the CP algorithm should be altered such that it needs to operate in an online manner. To achieve that, we first train the model using an offline dataset $\mathcal{O}$. Next, during deployment, we apply an online version of the CP algorithm described in Section III-A, with the weight modifications in Section III-B, onto a moving window of data samples $\mathcal{O}_p$ to obtain conformalized prediction intervals in an online fashion. Details of the online conformalization procedure are listed in the first part of Algorithm 1.

With this integrated learning and conformalization strategy, not only are the states $\hat{x}$ predicted by the model close to the true states, but more importantly, the true states are also guaranteed to lie within the prediction interval $C(\hat{f})$ with a high probability of $1-\delta$.

IV. UNCERTAINTY-AWARE LEARNING MPC

A. Uncertainty Propagation

To allow for a holistic integration between the uncertainty estimates and the learning-based MPC framework, we propagate the uncertainty estimates along the prediction horizon of

the optimization problem in (2), through the learned model $\hat{f}(x, u; \theta^\star)$. This is achieved by utilizing the unscented transformation (UT) [21]. The UT algorithm computes the statistics of a random variable that undergoes a nonlinear transformation. It considers a smaller number of samples, in contrast to the Monte Carlo sampling methods [21]. As a result, it is sample efficient and amenable to practical deployment. With the conformalized prediction intervals obtained from the first step of Algorithm 1, we consider the larger deviation between the quantiles of the prediction interval and the current state measurement to be an estimate of the standard deviation. Specifically, at each time step k, for the ℓ^{th} component of the state x, we define

$$\sigma_k^{(\ell)} := \max\left\{x^{(\ell)}(k) - q_l^{(\ell)}(z_{k-1}),\, q_u^{(\ell)}(z_{k-1}) - x^{(\ell)}(k)\right\}, \quad (11)$$

where $q_l^{(\ell)}(z_{k-1})$ and $q_u^{(\ell)}(z_{k-1})$ denote the ℓ^{th} component in the vectors of lower and upper quantiles, $q_l(z_{k-1})$, $q_u(z_{k-1}) \in \mathbb{R}^n$ respectively. Next, we form the covariance matrix of the states at time step k to be $\Sigma_k := \text{diag}\left(\sigma_k^{(1)^2}, \ldots, \sigma_k^{(n)^2}\right)$. The state measurement $x(k)$ and the covariance matrix Σ_k are then propagated through the learned model to obtain a sequence of covariance matrices. Specifically, for each $i \in \{0, \ldots, N-2\}$ along the horizon in the optimization problem (2), we propagate the state and covariance matrix forward in time by computing the following quantities,

$$\begin{aligned}
&\mathcal{Z}_0 = \bar{\mu}^i,\\
&\mathcal{Z}_j = \bar{\mu}^i + \left(\sqrt{(J+\lambda)\hat{\Sigma}_k^i}\right)_j,\ j = 1, \ldots, J,\\
&\mathcal{Z}_j = \bar{\mu}^i - \left(\sqrt{(J+\lambda)\hat{\Sigma}_k^i}\right)_{j-J},\ j = J+1, \ldots, 2J,\\
&\mathcal{Y}_j = \hat{f}\left(\mathcal{Z}_j, \bar{u}; \theta^\star\right),\ j = 0, \ldots, 2J,\\
&\bar{\mu}^{i+1} = \sum_{j=0}^{2J} w_j^{(m)} \mathcal{Y}_j,\ e_y^{ij} = \mathcal{Y}_j - \bar{\mu}^{i+1},\ \bar{\Sigma}_k^{i+1} = \sum_{j=0}^{2J} w_j^{(c)} e_y^{ij}\, e_y^{ij\top},
\end{aligned} \quad (12)$$

where the initial state and covariance matrix are given as $\bar{\mu}^0 := x(k)$ and $\bar{\Sigma}_k^0 := \Sigma_k$ and the vector $\bar{u}$ is the most recent control input. The matrices $\{\bar{\Sigma}_k^i\}_{i=0}^{N-1}$ denote the sequence of propagated covariance matrices. The weights $\{w_j^{(m)}\}_{j=0}^{2J}$ and $\{w_j^{(c)}\}_{j=0}^{2J}$ are functions of the scaling parameter λ and dimension J. Details on the computation of these weights and the parameter λ are given in [25].

B. Enhancing Uncertainty Awareness

To enhance the uncertainty awareness of learning-based MPC, the cost matrices in (2a) are augmented to account for the sequence of propagated covariance matrices. Given a predefined cost matrix Q, the stage cost matrices at each time step k are modified to be

$$Q_k^i := Q\left(\Delta_k^i\right)^{-1},\ i = 0, \ldots, N-1, \quad (13)$$

where $\Delta_k^i := \text{diag}\left(\frac{|x(k) - \hat{x}(k)|}{2\bar{\sigma}_k^i} + \xi\right)$, where the operations are performed element-wise. The predicted state $\hat{x}(k)$ is computed using the regression model $\hat{f}$ and the vector $\bar{\sigma}_k^i$ is the square root of the diagonal vector of $\bar{\Sigma}_k^i$. The constant ξ shifts the values of the diagonal elements to an appropriate range. This augmentation promotes uncertainty awareness by decreasing the cost weights for states in which the uncertainty estimates are high, *i.e.,* when the true state is far from the predicted state with respect to the width of the uncertainty interval, $2\bar{\sigma}_k^i$. Conversely, the states that are less uncertain are allowed to converge faster.

Algorithm 1: Weighted CP for Learning-based MPC

Input: Incoming data $\{(z_0, y_0), (z_1, y_1), \ldots\}$, learned model $\hat{f}(x, u; \theta^\star)$, window size p, target miscoverage rate α, weights $\{w_i\}_{i=1}^p$, UT parameters $\{\lambda, J\}$, cost matrix Q

Output: Conformalized interval $C(z_k)$, estimated covariances $\{\bar{\Sigma}_k^i\}_{i=0}^{N-1}$, cost matrices $\{Q_k^i\}_{i=0}^{N-1}$

Initialize: $\mathcal{O}_p \leftarrow \{0\}$
for $k = 0, 1, \ldots$ **do**
  // Step 1: Weighted CP
  Append latest data point, $\mathcal{O}_p \leftarrow \mathcal{O}_p \cup \{z_k\}$
  if $k \geq p$ **then**
    for $i = k-p, \ldots, k-1$ **do**
      Compute predictions $\hat{y}(z_i)$ using $\hat{f}(x_i, u_i; \theta^\star)$
      Compute scores $S_i(z_i)$ in (3)
    Compute $Q'_{1-\alpha}(\mathcal{S}_k)$ in (7) using $\{w_i\}_{i=1}^p$ and α, where $\mathcal{S}_k := \{S_i\}_{i=k-p}^{k-1}$
    Remove oldest data point, $\mathcal{O}_p \leftarrow \mathcal{O}_p \setminus \{z_{k-p}\}$
    Compute $C(z_k)$ usings $Q'_{1-\alpha}(\mathcal{S}_k)$ and (5)
  // Step 2: Unscented Transformation
  Compute initial covariance Σ_k with (11)
  Propagate covariances $\{\bar{\Sigma}_k^i\}_{i=0}^{N-1}$ using (12)
  // Step 3: Uncertainty-aware Augmentation
  Compute matrices $\{Q_k^i\}_{i=0}^{N-1}$ with Q and (13)
  Apply $\{Q_k^i\}_{i=0}^{N-1}$ to (2a) and solve (2)

One advantage of this cost augmentation is that it is minimally invasive and does not compromise any existing theoretical guarantees of the learning-based MPC framework. For instance, if the terminal cost matrix P in (2b) and constraint set $\mathcal{X}_f$ in (2e) are constructed such that the conditions described in [26] (see first paragraph of Section 2.2.1 therein) are satisfied, recursive feasibility and stability of the closed-loop system can be established. The proposed cost augmentation in (13) only modifies the stage cost and hence, does not affect recursive feasibility and stability of the closed-loop system.

Furthermore, this online conformalization and uncertainty propagation procedure together with the cost augmentation, as detailed in Algorithm 1, is applicable to a variety of data-driven models and learning-based MPC frameworks. In other words, this framework can be seen as a straightforward, efficient add-on module to existing learning-based MPC frameworks that promotes uncertainty awareness.

V. CASE STUDY WITH KNODE-MPC

We consider a cartpole system with the following dynamics [27],

$$\ddot{\gamma} = \frac{g\sin\gamma - \cos\gamma\left(F + m_p l\dot{\gamma}^2\sin\gamma\right)}{l\left(\frac{4}{3} - \frac{m_p\cos^2\gamma}{m_c+m_p}\right)}, \qquad (14)$$
$$\ddot{x}_c = \frac{F + m_p l\left(\dot{\gamma}^2\sin\gamma - \ddot{\gamma}\cos\gamma\right)}{m_c + m_p},$$

where x_c is the position of the cart and γ is the angle between the pole and the vertical. The cart and the pole have masses m_c and m_p and the pole has a length of $2l$. The gravitational force is denoted by g and F is the force acting on the system. By defining $x := [x_c\ \dot{x}_c\ \gamma\ \dot{\gamma}]^\top \in \mathbb{R}^4$ and $u := F \in \mathbb{R}$, the dynamics in (14) are discretized and act as the known dynamics of the system. For the unknown dynamics, we assume the cart has a true mass of 1.5kg, while the mass of the cart within the known dynamics is 1kg. Additionally, we consider unknown dynamics in the form of additive process noise. The noise distributions are assumed to be Gaussian with zero mean and standard deviations of $\{0.1\text{m}, 0.5\text{m/s}, 1^\text{o}, 20^\text{o}\text{/s}\}$. The full dynamics of the system are numerically simulated using an explicit 5$^{\text{th}}$ order Runge-Kutta method with a sampling time of 0.01s.

We utilize KNODE-MPC [10] as an instantiation of the learning-based MPC framework for the control of the cart-pole. First, we collect data from the system to synthesize a Knowledge-based Neural ODE (KNODE) model. For this model, we assume that the dynamics can be decomposed as

$$x^+ = f(x,\, u,\, d(x,u)) := \tilde{f}(x,\, u) + d(x,\, u), \qquad (15)$$

where the functions $\tilde{f},\, d : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ denote the known and unknown dynamics. Next, given a dataset $\mathcal{O} := \{(x_i,\, u_i)\}_{i=1}^T$ with sampling times $\{t_i\}_{i=1}^T$, we compute one-step predictions of the states,

$$\begin{aligned}\hat{x}_{i+1}(\theta) &:= x_i + \int_{t_i}^{t_{i+1}} \tilde{f}_c(x_i,u_i) + d_c(x_i,u_i;\,\theta)\,dt \\ &:= \hat{f}(x_i,u_i,t_i,t_{i+1};\,\theta),\end{aligned} \qquad (16)$$

where the functions $\tilde{f}_c,\, d_c : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ are the continuous-time representations of the dynamics in (15). The data-driven model $\hat{f} : \mathbb{R}^n\times\mathbb{R}^m\times\mathbb{R}\times\mathbb{R} \to \mathbb{R}^n$ is the KNODE model that represents the full system dynamics $f(x,u)$. For brevity, we suppress the dependency of sampling times for the KNODE model. The unknown dynamics $d_c(x,u)$ are parameterized as a NN with L layers, which is given by

$$\begin{aligned} h_0 &= [x^\top\ u^\top]^\top, \\ h_{l+1} &= \sigma\left(W_l h_l + b_l\right),\ \ l = 0,\ldots,L-1, \\ d_c(x,u;\,\theta) &= W_L h_L + b_L, \end{aligned} \qquad (17)$$

where the matrix $W_l \in \mathbb{R}^{n_{l+1}\times n_l}$ and the vector $b_l \in \mathbb{R}^{n_{l+1}}$ are the weights and biases of the l^{th} layer. The set $\theta := \{(W_0,b_0),\ldots,(W_L,b_L)\}$ to denotes the parameters of the NN. The function $\sigma : \mathbb{R}^{n_l} \to \mathbb{R}^{n_l}$ is an activation function, *e.g.*, the hyperbolic tangent function.

To train the KNODE model, we define a loss function by considering the mean squared errors between the states and the one-step predictions,

$$\mathcal{L}(\theta) := \frac{1}{T-1}\sum_{i=2}^{T} \|\hat{x}_i(\theta) - x_i\|_2^2. \qquad (18)$$

TABLE I
COVERAGE PERCENTAGES, UNDER 3 CP METHODS.

Method	Coverage percentages for each state			
	x_c [%]	$\dot{x}_c$ [%]	γ [%]	$\dot{\gamma}$ [%]
SCP, offline	99.00	98.96	99.14	99.00
Unweighted CP, online	98.96	98.87	99.00	99.02
Weighted CP, online	99.47	99.47	99.51	99.51

We compute the gradient of $\mathcal{L}$ with respect to the parameters θ and update θ in an episodic manner, with a backpropagation procedure. After training, the KNODE model with optimized parameters $\theta^\star$ serves as the dynamics model within an MPC framework. In this case study, the cost matrices in (2a) are set to $Q := \mathbb{I}$ and $R := 0.5$. The prediction horizon N is set to 12 and the control sampling rate is 0.15s. The weighted CP procedure operates at the same rate as the control scheme. The miscoverage rate is $\alpha := 0.01$ and the weights $\{w_i\}_{i=1}^p$ are defined according to a geometric decay, with a scaling factor of 0.995. The size of the moving window is $p := 300$. The UT parameters are $\{\alpha,\beta,\kappa\} := \{0.1, 2, -1\}$. The constant ξ in (13) is set to 0.5. The CasADi library [28] and the solver IPOPT [29] are used to formulate and solve the optimization problem in (2).

A. Results and Discussion

We first verify the coverage guarantees obtained from the online weighted CP procedure by tabulating the empirical coverage percentages in Table I. These are computed by summing the number of time instances in which the true data samples lie within the conformalized prediction intervals across a test dataset. For comparison, we consider three conformalization methods - (i) the SCP procedure described in Section III-A, (ii) an online unweighted CP procedure that utilizes a moving window of historical data samples, and (iii) the online weighted CP scheme that uses the same moving window of data samples, but with weights. The intervals for SCP are computed using a calibration dataset, which is different from the test dataset. While SCP is able to provide a coverage rate close to 99%, the algorithm generates a fixed prediction interval before deployment. Hence, the computed intervals cannot account for distribution shifts in data samples that may happen in subsequent deployments. One instance is observed from the lower coverage percentage for $\dot{x}_c$. The unweighted CP procedure is able to provide uncertainty estimates in an online fashion, but may fall short of the coverage level, as observed in the coverage percentages for some of the states. The online weighted CP is more conservative and provides sufficient and the highest coverage for all states.

Next, we examine the size of the prediction intervals, with respect to the states of the cartpole system in a closed-loop simulation. In this experiment, the cart is required to track a sequence of reference step commands, as shown in Fig. 3. In Fig. 3, the time histories of the states are plotted with the corresponding prediction intervals shown in light blue. It is observed that the intervals have magnitudes that are comparable to those of the states, which makes them practically useful for downstream tasks that require uncertainty quantification.

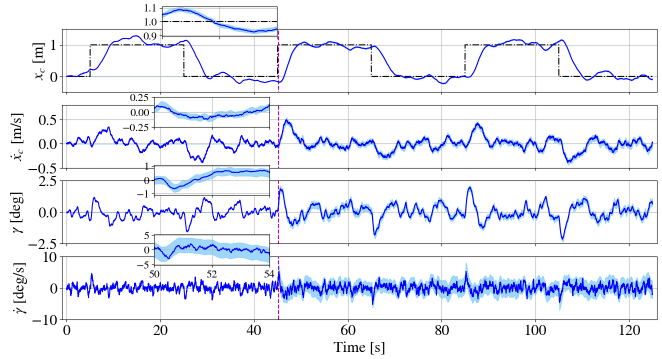

Fig. 3. The time histories of the states of the cartpole system under the proposed framework. The light blue regions depict the intervals computed by the online WCP algorithm, in which the true states are guaranteed to lie within, with a given probability. The black dashed line in the first subplot depicts the reference trajectory. The vertical dashed line indicates the time instance, $t = 45.15s$, when the predictions are activated. The insets are zoomed-in plots from $t = 50 - 54$s.

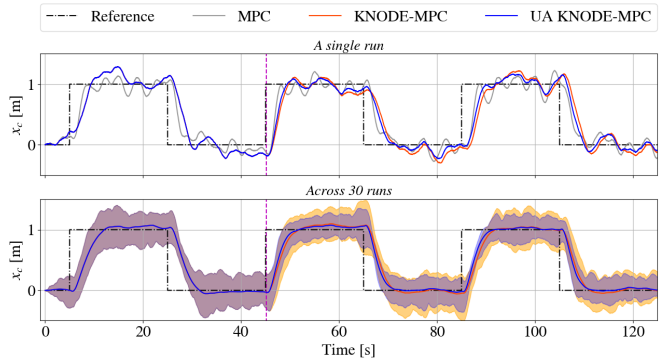

Fig. 4. The time histories of the cart position, under three control frameworks - (i) nominal MPC, (ii) KNODE-MPC: MPC with a KNODE model and without uncertainty awareness, and (iii) Uncertainty-aware (UA) KNODE-MPC. The top panel is for a single run, while the bottom panel depicts the spread across 30 runs. The orange and blue shaded regions in the bottom panel denote the spread of the trajectories across 30 runs for KNODE-MPC and UA KNODE-MPC respectively.

To evaluate the differences in closed-loop performance when the uncertainty awareness augmentation is incorporated into the control scheme, the time histories of the cart position for a single run and across 30 runs are shown in Fig. 4. As observed in the top panel of Fig. 4, due to the presence of the unknown dynamics and inaccuracy of the model, oscillatory responses are observed in the trajectories under a nominal MPC framework. On the other hand, the KNODE-MPC framework mitigates these oscillations and achieves improved tracking performance. With the uncertainty-aware augmentation, it is observed that the closed-loop responses under the uncertainty-aware KNODE-MPC framework are faster and do not induce oscillations. The bottom panel of Fig. 4 depicts the spread of the trajectories of the cart position across 30 runs. The spread is computed using the maximum and minimum values at each time step across 30 runs, while the solid lines depict the mean across the runs at every time step. The uncertainty-aware KNODE-MPC framework achieves a narrower spread, as compared to KNODE-MPC, ascertaining the improvement in responsiveness of the closed-loop system.

Next, we examine the computation times for (i) solving the optimization problem (2), (ii) for computing the quantiles in (7), and (iii) the time required for the unscented transformation, as described in (12). The statistics of the computation times shown in Table II are computed across all time steps of 30 runs, after the uncertainty aware augmentation is activated. It is evident that the main computational limitation lies with the uncertainty propagation step. We leave exploring computationally efficient uncertainty propagation procedures as part of future work.

TABLE II

STATISTICS OF COMPUTATION TIMES, IN MILLISECONDS.

	Median	Mean	Std dev.
Optimization time	3.72	3.80	0.31
Quantile computation time	0.70	0.71	0.06
Unscented transform time	40.38	40.62	1.06

VI. CONCLUSION

We present an uncertainty-aware learning-based MPC framework that incorporates an online weighted CP procedure for uncertainty estimation and UT for uncertainty propagation. The framework not only quantifies the uncertainty in the state predictions, but also promotes uncertainty awareness in the closed-loop system. As part of future work, we plan to incorporate the uncertainty estimates into other aspects of the learning-based MPC framework to further enhance uncertainty awareness.

REFERENCES

[1] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger, "Learning-based MPC: Toward safe learning in control," *Annual Review of Control, Robot., and Auton. Syst.*, vol. 3, pp. 269–296, 2020.

[2] A. Mesbah, K. P. Wabersich, A. P. Schoellig, M. N. Zeilinger, S. Lucia, T. A. Badgwell, and J. A. Paulson, "Fusion of machine learning and mpc under uncertainty: What advances are on the horizon?" in *2022 American Control Conf. (ACC)*. IEEE, 2022, pp. 342–357.

[3] C. K. Williams and C. E. Rasmussen, *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2006, vol. 2, no. 3.

[4] J. Kabzan, L. Hewing, A. Liniger, and M. N. Zeilinger, "Learning-based MPC for autonomous racing," *IEEE Robot. and Auto. Letters*, vol. 4, no. 4, pp. 3363–3370, 2019.

[5] G. Torrente, E. Kaufmann, P. Föhn, and D. Scaramuzza, "Data-driven mpc for quadrotors," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3769–3776, 2021.

[6] A. Draeger, S. Engell, and H. Ranke, "Model predictive control using neural networks," *IEEE Control Syst. Magazine*, vol. 15, no. 5, pp. 61–66, 1995.

[7] G. Williams, N. Wagener, B. Goldfain, P. Drews, J. M. Rehg, B. Boots, and E. A. Theodorou, "Information theoretic MPC for model-based reinforcement learning," in *2017 IEEE Int. Conf. on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 1714–1721.

[8] Z. Wu, D. Rincon, Q. Gu, and P. D. Christofides, "Statistical machine learning in model predictive control of nonlinear processes," *Mathematics*, vol. 9, no. 16, p. 1912, 2021.

[9] Y. Chen, Z. Tong, Y. Zheng, H. Samuelson, and L. Norford, "Transfer learning with deep neural networks for model predictive control of hvac and natural ventilation in smart buildings," *Journal of Cleaner Production*, vol. 254, p. 119866, 2020.

[10] K. Y. Chee, T. Z. Jiahao, and M. A. Hsieh, "Knode-mpc: A knowledge-based data-driven predictive control framework for aerial robots," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2819–2826, 2022.

[11] K. Y. Chee, M. A. Hsieh, and N. Matni, "Learning-enhanced nonlinear model predictive control using knowledge-based neural ordinary differential equations and deep ensembles," in *Learning for Dynamics and Control Conference*. PMLR, 2023, pp. 1125–1137.

[12] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *Int. Conf. on machine learning*. PMLR, 2016, pp. 1050–1059.

[13] S. Sun, G. Zhang, J. Shi, and R. Grosse, "Functional variational bayesian neural networks," *arXiv preprint arXiv:1903.05779*, 2019.
[14] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," *Advances in neural Info. process. Syst.*, vol. 30, 2017.
[15] V. Vovk, A. Gammerman, and G. Shafer, *Algorithmic learning in a random world*. Springer, 2005, vol. 29.
[16] I. Gibbs and E. Candes, "Adaptive conformal inference under distribution shift," *Advances in Neural Info. process. Syst.*, vol. 34, pp. 1660–1672, 2021.
[17] C. Xu and Y. Xie, "Conformal prediction interval for dynamic time-series," in *Int. Conf. on Machine Learning*. PMLR, 2021, pp. 11 559–11 569.
[18] R. F. Barber, E. J. Candes, A. Ramdas, and R. J. Tibshirani, "Conformal prediction beyond exchangeability," *The Annals of Statistics*, vol. 51, no. 2, pp. 816–845, 2023.
[19] L. Lindemann, M. Cleaveland, G. Shim, and G. J. Pappas, "Safe planning in dynamic environments using conformal prediction," *IEEE Robotics and Automation Letters*, 2023.
[20] A. Dixit, L. Lindemann, S. X. Wei, M. Cleaveland, G. J. Pappas, and J. W. Burdick, "Adaptive conformal prediction for motion planning among dynamic agents," in *Learning for Dynamics and Control Conference*. PMLR, 2023, pp. 300–314.
[21] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004.
[22] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.
[23] R. Koenker and G. Bassett Jr, "Regression quantiles," *Econometrica: journal of the Econometric Society*, pp. 33–50, 1978.
[24] H. Papadopoulos, K. Proedrou, V. Vovk, and A. Gammerman, "Inductive confidence machines for regression," in *Machine Learning: ECML 2002: 13th European Conference on Machine Learning Helsinki, Finland, August 19–23, 2002 Proceedings 13*. Springer, 2002, pp. 345–356.
[25] E. A. Wan and R. Van Der Merwe, "The unscented kalman filter for nonlinear estimation," in *Proceedings of IEEE 2000 Adaptive Syst. for Signal process., Comm., and Control Symposium*, pp. 153–158.
[26] D. Q. Mayne, "Model predictive control: Recent developments and future promise," *Automatica*, vol. 50, no. 12, pp. 2967–2986, 2014.
[27] A. G. Barto, R. S. Sutton, and C. W. Anderson, "Neuronlike adaptive elements that can solve difficult learning control problems," *IEEE Transactions on Syst., Man, and Cybernetics*, vol. SMC-13, no. 5, pp. 834–846, 1983.
[28] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Math. Program. Comp.*, vol. 11, no. 1, pp. 1–36, 2019.
[29] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical program.*, vol. 106, no. 1, pp. 25–57, 2006.