

# Gaussian Processes Model-based Control of Underactuated Balance Robots

Kuo Chen, Jingang Yi, and Dezhen Song

**Abstract**—Ranging from cart-pole systems and autonomous bicycles to bipedal robots, control of these underactuated balance robots aims to achieve both external (actuated) subsystem trajectory tracking and internal (unactuated) subsystem balancing tasks with limited actuation authority. This paper proposes a learning model-based control framework for underactuated balance robots. The key idea to simultaneously achieve tracking and balancing tasks is to design control strategies in slow- and fast-time scales, respectively. In slow-time scale, model predictive control (MPC) is used to generate the desired internal subsystem trajectory that encodes the external subsystem tracking performance and control input. In fast-time scale, the actual internal trajectory is stabilized to the desired internal trajectory by using an inverse dynamics controller. The coupling effects between the external and internal subsystems are captured through the planned internal trajectory profile and the dual structural properties of the robotic systems. The control design is based on Gaussian processes (GPs) regression model that are learned from experiments without need of priori knowledge about the robot dynamics nor successful balance demonstration. The GPs provide estimates of modeling uncertainties of the robotic systems and these uncertainty estimations are incorporated in the MPC design to enhance the control robustness to modeling errors. The learning-based control design is analyzed with guaranteed stability and performance. The proposed design is demonstrated by experiments on a Furuta pendulum and an autonomous bikebot.

**Index Terms**—Gaussian processes, underactuated robots, model predictive control, non-minimum phase systems

## I. INTRODUCTION

Underactuated systems commonly have fewer number of control inputs than the number of degree of freedom (DOF) [1]. Underactuated balance robots, first introduced in [2], is a class of underactuated systems with control task of trajectory tracking for actuated subsystem, while balancing around unstable equilibria for unactuated subsystem. Cart-pole systems [3], Furuta pendulums [4]–[6] and autonomous bicycles [7], [8] are a few examples of underactuated balance robots with the goal to balance the inverted pendulum or bikebot while the base platforms to follow desired trajectories (see Figs. 1(a) and 1(b)). Bipedal walkers (e.g., Fig. 1(c)) are also a type of underactuated balance robots because the

actuated joint angles are commanded to follow the desired trajectories to form certain gaits while the unactuated floating base is kept stable across steps [9]–[11].

Control of underactuated balance robots faces challenges because no analytical casual compensator can achieve exactly trajectory tracking for the non-minimum phase systems [12]. The dynamics of the underactuated balance robotic systems can be naturally partitioned into an actuated (external) subsystem and an unactuated (internal) subsystem. An innovative control design of underactuated balance robots is to take advantages of the interaction between the external and internal subsystems. In [2], by observing the dependency of the balanced equilibria on trajectory tracking performance, a balance equilibrium manifold (BEM) concept is proposed to map and encode the external subsystem trajectory tracking into the desired internal subsystem profiles. A controller is then designed to stabilize the system state onto the BEM in order to achieve both tracking and balancing tasks. Despite of the mathematical elegance and guaranteed stability property, the design in [2] requires accurate dynamics model and control robustness is not ensured to allow the robots to perform well in complex, dynamic environments.

In recent years, using machine learning techniques, data-driven model-based controller design showed promising potentials to capture complex, high-dimensional systems dynamics and achieve superior performance over physical principle model-based controllers. Gaussian processes (GPs) are used as non-parametric machine learning models and have been widely applied to robot modeling and control [13]. When they are applied to capture and model robotic system dynamics, GPs take the current robot states and control actuation and their derivatives as the learning model input and output, respectively. GP models provide differentiable and closed-form mean and covariance distributions and this property is attractive for optimization-based control designs such as model predictive control (MPC) or reinforcement learning [14]–[21]. Compared to other dynamics learning methods, such as artificial neural network or support vector machine, GPs provide predictive covariance that can be used as a quantitative metric of model uncertainty. The covariance has also been used to design robust controllers (e.g., [16]–[19]).

MPC is an optimization-based preview control method. At each control step, the MPC design solves the optimal input sequence that minimizes the objective function. Computational cost is expensive for high-dimensional robotic systems dynamics. In this work, we adopt a singular perturbation method to reduce the dimensionality of the dynamic models of underactuated balance robots such that MPC is applied

The preliminary version of this paper was presented in part at the 2019 IEEE International Conference on Robotics and Automation, May 20-24, 2019, Montreal, Canada. This work was partially supported by the National Science Foundation under awards CMMI-1762556 and CNS-1932370 (J. Yi).

K. Chen and J. Yi are with the Department of Mechanical and Aerospace Engineering, Rutgers University, Piscataway, NJ 08854 USA (e-mail: kc625@scarletmail.rutgers.edu; jgyi@rutgers.edu).

D. Song is with the Department of Computer Science and Engineering, Texas A&M University, College Station, TX 77843-3112, USA (e-mail: dzsong@cse.tamu.edu).

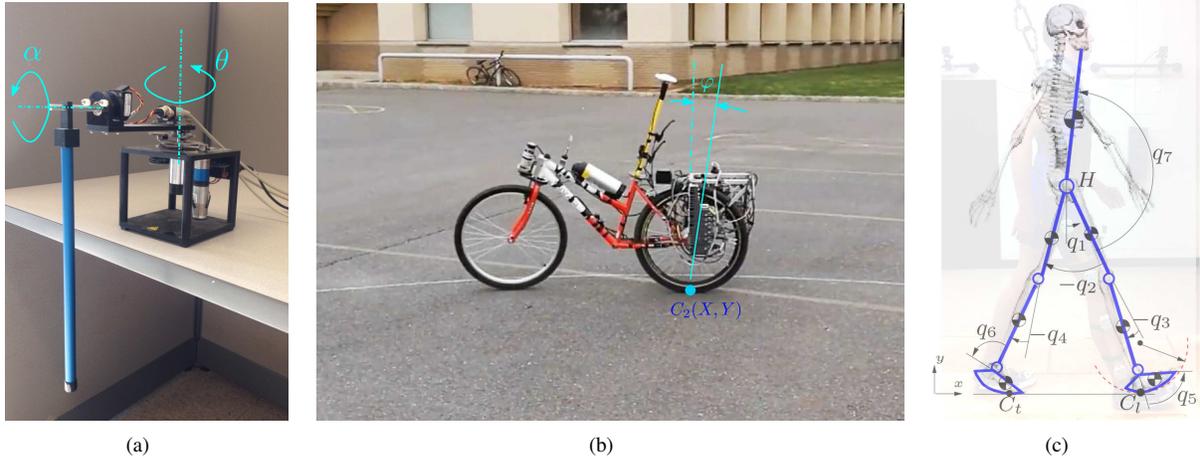


Fig. 1. A few examples of underactuated balance robotic systems. (a) Rotary inverted pendulum. Angular joint  $\theta$  is the actuated DOF and joint angle  $\alpha$  is the unactuated DOF. (b) The bikebot system. The robot has three DOFs (i.e., the rear wheel contact position  $C_2(X, Y)$  and platform roll angle  $\varphi$ ) and only two actuation inputs, that is, steering angle and velocity control. (c) A robotic bipedal walker. The robotic walker has seven DOFs ( $q_1$ - $q_7$ ) and six actuation inputs (i.e., double actuation at hip, knee and ankle joints) during single-stance gait.

to the model effectively and efficiently. By transforming the underactuated balance robot dynamics into an external/internal convertible (EIC) form [2], the internal subsystem is feedback linearizable and the convergence rate of the error dynamics is designed to be much higher than that of the external subsystem dynamics. The internal states are then treated as the control input to the external subsystems. Taking the cart-pole system as an example, through feedback linearization, the pendulum angle is directly controlled with a desired balance angle profile that is treated as an input to the cart position dynamics. We adopt the MPC as an online planner to achieve the desired pendulum balance angle and the cart position tracking simultaneously. Both the external and internal subsystem dynamics are learned from experimental data with GPs models and the MPC trajectory planner takes the model uncertainties into the design to enhance the control robustness. We demonstrate the proposed planning and control design on the Furuta pendulum and the bikebot platforms.

The contribution of this work lies in three aspects. First, the control design is based on learning models without need of obtaining physical dynamics model and therefore, it has attractive for many complex, high-dimensional underactuated balance robotic systems. It is difficult, if not impossible, to obtain dynamic models of many of these robotic systems by physical principles. The proposed learning-based control design takes advantage of the EIC structure of systems dynamics of the underactuated balance robots [2]. Second, the proposed control design is data efficient and effective. Most previous work relies on either the prior knowledge of the physical model or the successful demonstration from human expert or simple linear controller for efficiently training. The proposed approach takes random excitation data for model training, and then achieves successful balancing and tracking tasks. The system only needs to be excited under open-loop system control and the model is learned without any prior knowledge or successful balance demonstration. Finally, our proposed control demonstrates a novel design of explicitly incorporating

the GPs model uncertainty to enhance control robustness. The design is also guaranteed stability and convergence and robustness performance.

The rest of the paper is organized as follows. Section II reviews relevant work. In Section III, we present the control systems design of the underactuated balance robots with physical models. Section IV extends the control design with GP models. We present the control performance analysis in Section V. Experimental results are included in Section VI. Finally, we summarize the concluding remarks and briefly discuss the future research directions.

## II. RELATED WORKS

We mainly review the most relevant work in research areas such as model-based control of underactuated balance robots, learned dynamics models and MPC learning schemes in robotic applications.

[2] presented the EIC models of the underactuated balance robotic dynamics. The EIC form describes the coupling effect of the external and internal subsystem dynamics and the BEM is introduced to capture the dependency of the the internal subsystem equilibria on the external tracking performance. Dynamic inversion technique is used in [2] to compute the BEM and the control system is proven to be asymptotically stable to a neighborhood around the desired trajectories. The work in [3] formulates the EIC form in a multi-time-scale structure based on the singular perturbation theory and output feedback is achieved with extended high-gain observers. The work in [22] extends the BEM approach to learning model-based control. GPs are adopted to identify the system dynamics but the dynamics structure was not successfully captured in spite of small prediction errors. The learned BEM approach demonstrates worse tracking performance than that with the physical model even though the learned model itself generates less prediction errors. The learning model-based BEM in [22] is not accurately estimated due to the flexible structure of GPs and dynamic inversion does not accurately identify the BEM

for the learned models. This observation motivates the work in this paper.

Learning inverse dynamics has been demonstrated in many robot control applications. A review of the model learning and robot control can be found in [23]. The work in [24], [25] adopt an inverse dynamics controller using global and local GPs regression models, respectively. The learned model predicts control inputs based on the robot current states and the desired derivative of robot states. Although GPs provide predictive distribution, only the mean value of the Gaussian distribution is used as the control input. The work in [26] proposes a GP-based inverse dynamics control law and the feedback gain is adapted to the variance of the predictive distribution, that is, using low gains if the learned model is precise and otherwise high gains. The work in [20], [26], [27] give theoretically guaranteed stability or safety regions of GPs-based inverse dynamics control. Besides GPs, polynomial kernel functions are also used to predict the inverse dynamics of robotic systems (e.g., [28]). In [29], deep neural network (DNN) is used to learn inverse dynamics to achieve impromptu trajectory tracking. The work in [30] achieve robotic impromptu trajectory tracking for a cart-pole system and quadrotor system by learning a stable, approximate inverse of a non-minimum phase baseline system. The proposed algorithm first runs a baseline controller, usually a linear controller, to achieve the stabilization task and collect input-state data for DNN training. In training phase, the inverse model of the stabilized baseline system is learned, while in testing phase, given the desired trajectory, the learned DNN model computes a reference trajectory for the baseline system. Under this learning-based inversion controller, the tracking performance is enhanced comparing with the baseline system. The algorithm however requires a baseline controller to stabilize the system for data collection.

Optimization-based controllers such as MPC and reinforcement learning have been applied to underactuated robot system such as cart-pole system, blimps and helicopters. In [14], a learning model captures the difference between the collected acceleration data of the blimp and the prediction from the physical model so that the learning-based design does not have to build the blimp model from scratch. In [31], a helicopter model is learned with maneuvers and trajectories that are successfully demonstrated by human expert. By either adding prior knowledge of the robot model or learning from expert demonstration, the learned models are efficiently trained. The work in [16] do not assume task-specific prior knowledge but take advantage of the probabilistic nature of Gaussian processes to achieve efficient learning. Many GP-based designs take advantage of the predicted Gaussian distribution to achieve robust control performance. For example, in [16], [17], [19]–[21], the objective function is designed to include tracking errors over the prediction horizon with the variance of the predictive distribution. In [18], the predictive variance is used to help reduce the feasible region for the predictive trajectory mean value. Learning-based inverse dynamics control and MPC have been demonstrated in many applications in [32]–[35]. The works in [36]–[38] adopt inverse dynamics controller with the global and local GPs regression models. These inverse

dynamics controllers however cannot be directly applied to underactuated non-minimum phase balance robots due to the unstable internal dynamics. In this paper, we take advantages of the physical model structure of the underactuated balance robotic dynamics and use reduced-dimensional learning models to develop an computationally efficient control system. Moreover, we demonstrate the guaranteed stability and robust control performance with the GP-based design analysis.

### III. BALANCE ROBOTS CONTROL

#### A. Notations

Vectors  $\alpha$  and matrices  $A$  are denoted with bold lower-case and capital characters, respectively. An  $n \times n$  identity matrix is denoted as  $I_n$ . Estimated values of variables are denoted by symbols with hat (e.g.,  $\hat{\alpha}$ ). Natural and real number sets are denoted as  $\mathbb{N}$  and  $\mathbb{R}$ , respectively. Positive real value set and  $n$ -dimensional real valued vector space are denoted as  $\mathbb{R}^+$  and  $\mathbb{R}^n$ , respectively. The smallest and largest eigenvalues of matrix  $A$  are denoted by  $\lambda_{\min}(A)$  and  $\lambda_{\max}(A)$ , respectively. The matrix and vector norms are defined respectively as  $\|A\| = [\lambda_{\max}(A^T A)]^{\frac{1}{2}}$  and  $\|\alpha\| = \sqrt{\alpha^T \alpha}$ . The metric  $\|\alpha\|_P^2 = \alpha^T P \alpha$  is used for positive definition matrix  $P$ .  $\text{tr}(A)$  and  $\det(A)$  denote the trace and determinant of matrix  $A$ , respectively.

The expression  $x \sim \mathcal{N}(\mu, \Sigma)$  represents that  $x$  is a random variable satisfying Gaussian distribution with mean value  $\mu$  and covariance  $\Sigma$ . The expression  $\dot{x} \sim f(x, u)$  represents that  $\dot{x}$  is a random variable satisfying a distribution because either  $(x, u)$  are random variables,  $f$  is a Gaussian process-based random function, or both. The expectation operator is denoted as  $\mathbb{E}$ , variable  $\mathbf{\Pi}$  denotes a probabilistic event and its probability is written as  $\Pr\{\mathbf{\Pi}\}$ . For discrete-time MPC presentation,  $k \in \mathbb{N}$  is used to denote the current time step, and  $k + i$  with  $i \in \mathbb{N}$  is used to denote the  $i$ -step forward time moment. A variable  $\alpha^*$  with a “\*” superscript denotes the optimal value of the design parameter  $\alpha$ .

#### B. Underactuated balance system control

In this section, we present a physical model-based control system design for underactuated balance robots. The presented work will serve as a basic description of the approach that is used for GP-based design in later sections.

An underactuated balance robotic system is described by the following dynamic model

$$D(q)\ddot{q} + H(q, \dot{q}) = B(q)u, \quad (1)$$

where  $q \in \mathbb{R}^{m+n}$  is the generalized coordinate of the system,  $u \in \mathbb{R}^m$  is the control input,  $D(q)$  is the inertia matrix,  $H(q, \dot{q})$  contains the centripetal, Coriolis and gravitational terms and  $B(q)$  is the input mapping matrix [39]. A few examples that share the above dynamic models include cart-pole systems [3], Furuta pendulums [4], bicycles and bikebots [7], [8], and bipedal walkers [9], [10], etc.

Without loss of generality, coordinate  $q = [\theta_1^T \alpha_1^T]^T$  is considered to be decomposed into generalized positions  $\theta_1 \in \mathbb{R}^m$  of the actuated subsystem and  $\alpha_1 \in \mathbb{R}^n$  of the unactuated subsystem. We assume that  $m \geq n$ , that is, the actuated DOF

is not less than the unactuated DOF. We define generalized velocities  $\boldsymbol{\theta}_2 = \dot{\boldsymbol{\theta}}_1$  and  $\boldsymbol{\alpha}_2 = \dot{\boldsymbol{\alpha}}_1$  such that  $\dot{\boldsymbol{q}} = [\boldsymbol{\theta}_2^T \boldsymbol{\alpha}_2^T]^T$ . Equation (1) is then partitioned into actuated and unactuated subsystems as

$$D \begin{bmatrix} \dot{\boldsymbol{\theta}}_2 \\ \dot{\boldsymbol{\alpha}}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{H}_1(\boldsymbol{q}, \dot{\boldsymbol{q}}) \\ \mathbf{H}_2(\boldsymbol{q}, \dot{\boldsymbol{q}}) \end{bmatrix} = \begin{bmatrix} \mathbf{B}_1(\boldsymbol{q}) \\ \mathbf{0}_{n \times m} \end{bmatrix} \boldsymbol{u}, \quad (2)$$

where  $\mathbf{B}_1(\boldsymbol{q}) \in \mathbb{R}^{m \times m}$  is full rank. By inverting the mass matrix  $D(\boldsymbol{q})$  in (2), we obtain

$$\begin{bmatrix} \dot{\boldsymbol{\theta}}_2 \\ \dot{\boldsymbol{\alpha}}_2 \end{bmatrix} = D^{-1} \begin{bmatrix} \mathbf{B}_1(\boldsymbol{q})\boldsymbol{u} - \mathbf{H}_1(\boldsymbol{q}, \dot{\boldsymbol{q}}) \\ -\mathbf{H}_2(\boldsymbol{q}, \dot{\boldsymbol{q}}) \end{bmatrix}. \quad (3)$$

A general state-space representation of (3) is formulated as

$$\begin{cases} \Sigma_e : \dot{\boldsymbol{\theta}}_1 = \boldsymbol{\theta}_2, \dot{\boldsymbol{\theta}}_2 = \mathbf{f}_\theta(\boldsymbol{\theta}, \boldsymbol{\alpha}, \boldsymbol{u}), \\ \Sigma_i : \dot{\boldsymbol{\alpha}}_1 = \boldsymbol{\alpha}_2, \dot{\boldsymbol{\alpha}}_2 = \mathbf{f}_\alpha(\boldsymbol{\theta}, \boldsymbol{\alpha}, \boldsymbol{u}), \end{cases} \quad (4)$$

where  $\boldsymbol{\theta} = [\boldsymbol{\theta}_1^T \boldsymbol{\theta}_2^T]^T$ ,  $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1^T \boldsymbol{\alpha}_2^T]^T$ , and  $\mathbf{f}_\theta(\cdot)$  and  $\mathbf{f}_\alpha(\cdot)$  are nonlinear vector functions that represent state variables and velocity fields for external  $\Sigma_e$  and internal  $\Sigma_i$  subsystems, respectively. The goal of the control system is to force the external subsystem  $\Sigma_e$  to track desired trajectory  $\boldsymbol{\theta}_d = [\boldsymbol{\theta}_{d1}^T \boldsymbol{\theta}_{d2}^T]^T$ ,  $\boldsymbol{\theta}_{d2} = \dot{\boldsymbol{\theta}}_{d1}$ , while the internal subsystem  $\Sigma_i$  to keep balancing around unstable equilibria.

In (4), the external subsystem  $\Sigma_e$  and internal subsystem  $\Sigma_i$  are coupled and considered dual relationship [2]. For example, letting

$$\boldsymbol{v} = \mathbf{f}_\alpha(\boldsymbol{\theta}, \boldsymbol{\alpha}, \boldsymbol{u}), \quad (5)$$

subsystem  $\Sigma_i$  is feedback linearized as  $\dot{\boldsymbol{\alpha}}_2 = \boldsymbol{v}$ . Because of  $\boldsymbol{v} \in \mathbb{R}^n$  and  $\boldsymbol{u} \in \mathbb{R}^m$ , only a subspace of  $\boldsymbol{u}$  is obtained by inverting (5). Letting  $\boldsymbol{u} = [\boldsymbol{u}_d^T \boldsymbol{u}_f^T]^T$ ,  $\boldsymbol{u}_d \in \mathbb{R}^n$  and  $\boldsymbol{u}_f \in \mathbb{R}^{m-n}$ ,  $\boldsymbol{u}_d$  is obtained by an inverse dynamics method

$$\boldsymbol{u}_d = \mathbf{f}_\alpha^{-1}(\boldsymbol{\theta}, \boldsymbol{\alpha}, \boldsymbol{v}, \boldsymbol{u}_f), \quad (6)$$

while  $\boldsymbol{u}_f$  is freely designed. System (4) under (6) becomes

$$\begin{cases} \Sigma_e : \dot{\boldsymbol{\theta}}_1 = \boldsymbol{\theta}_2, \dot{\boldsymbol{\theta}}_2 = \mathbf{f}_\theta(\boldsymbol{\theta}, \boldsymbol{\alpha}, \boldsymbol{u}(\boldsymbol{v}, \boldsymbol{u}_f)), \\ \Sigma_i : \dot{\boldsymbol{\alpha}}_1 = \boldsymbol{\alpha}_2, \dot{\boldsymbol{\alpha}}_2 = \boldsymbol{v}. \end{cases} \quad (7)$$

In (7),  $\Sigma_i$  is directly controlled by  $\boldsymbol{v}$  and not affected by  $\Sigma_e$ , while  $\Sigma_e$  is affected by both inputs  $\boldsymbol{u}_f$  and  $\boldsymbol{v}$ .

Temporarily ignoring the tracking task of  $\boldsymbol{\theta}$  for  $\Sigma_e$ , we design a proportional-differential (PD) controller to force  $\boldsymbol{\alpha}$  to converge to desired trajectory  $\boldsymbol{\alpha}_d = [\boldsymbol{\alpha}_{d1}^T \boldsymbol{\alpha}_{d2}^T]^T$ ,  $\boldsymbol{\alpha}_{d2} = \dot{\boldsymbol{\alpha}}_{d1}$ , namely,

$$\boldsymbol{v}_{pd} = \dot{\boldsymbol{\alpha}}_{d2} - \frac{k_d}{\epsilon} \boldsymbol{e}_{\alpha 2} - \frac{k_p}{\epsilon^2} \boldsymbol{e}_{\alpha 1}, \quad (8)$$

where errors  $\boldsymbol{e}_{\alpha 1} = \boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_{d1}$ ,  $\boldsymbol{e}_{\alpha 2} = \boldsymbol{\alpha}_2 - \boldsymbol{\alpha}_{d2}$ ,  $\boldsymbol{e}_\alpha = [e_{\alpha 1}^T e_{\alpha 2}^T]^T$ ,  $\epsilon > 0$  is a small positive constant called singular perturbation parameter,  $k_p > 0$  and  $k_d > 0$  are constant control gains. To enforce the tracking task for  $\Sigma_e$ , the desired trajectory  $\boldsymbol{\alpha}_d(\boldsymbol{\theta}_d, \boldsymbol{\theta})$  is designed to be dependent on  $(\boldsymbol{\theta}_d, \boldsymbol{\theta})$  such that  $\boldsymbol{\theta} \rightarrow \boldsymbol{\theta}_d$  and BEM is used to capture such dependency. The BEM is defined as

$$\mathcal{E} = \{\boldsymbol{\alpha}_d = \boldsymbol{\alpha}_d^e : \boldsymbol{\alpha}_{d1}^e = \boldsymbol{\alpha}_{d1}(\boldsymbol{\theta}_d, \boldsymbol{\theta}), \boldsymbol{\alpha}_{d2}^e = \mathbf{0}\} \quad (9)$$

and  $\boldsymbol{\alpha}_{d1}^e$  is obtained by inverting an implicit function

$$\mathbf{f}_\theta(\boldsymbol{\theta}, \boldsymbol{\alpha}_d) = \dot{\boldsymbol{\theta}}_{d2} - k_d \boldsymbol{e}_{\theta 2} - k_p \boldsymbol{e}_{\theta 1}, \quad (10)$$

where errors  $\boldsymbol{e}_{\theta 1} = \boldsymbol{\theta}_1 - \boldsymbol{\theta}_{d1}$ ,  $\boldsymbol{e}_{\theta 2} = \boldsymbol{\theta}_2 - \boldsymbol{\theta}_{d2}$ , and  $\boldsymbol{e}_\theta = [e_{\theta 1}^T e_{\theta 2}^T]^T$ . Under assumption of affine error structure, the controller in (8) results in exponential convergence of  $\boldsymbol{\alpha}$  and  $\boldsymbol{\theta}$  to the respective neighborhoods of  $\mathcal{E}$  and  $\boldsymbol{\theta}_d$  simultaneously [2].

It is shown in [22] that inverting (10) suffers accuracy issue for a learned model of  $\mathbf{f}_\theta$ . We instead take an MPC approach to solve  $\boldsymbol{\alpha}_d^0$  and obtain  $\mathcal{E}$  under tracking design of  $\boldsymbol{\theta}_d$ . We do not directly apply MPC to (7) to solve  $\boldsymbol{v}$  because in that case the controlled  $\Sigma_i$  might not be stable. We address the challenge of stabilizing the unstable internal subsystem  $\Sigma_i$  and guarantee the stability performance through a singular perturbation design as described in the following subsection.

### C. Model reduction through singular perturbation

We apply controller (8) to (7) and the resulted error dynamics are

$$\begin{cases} \dot{\boldsymbol{\theta}}_1 = \boldsymbol{\theta}_2, \dot{\boldsymbol{\theta}}_2 = \mathbf{f}_\theta(\boldsymbol{\theta}, \boldsymbol{\alpha}_d + \boldsymbol{e}_\alpha, \boldsymbol{u}(\boldsymbol{v}_{pd}, \boldsymbol{u}_f)) \\ \dot{\boldsymbol{e}}_{\alpha 1} = \boldsymbol{e}_{\alpha 2}, \dot{\boldsymbol{e}}_{\alpha 2} = -\frac{k_p}{\epsilon^2} \boldsymbol{e}_{\alpha 1} - \frac{k_d}{\epsilon} \boldsymbol{e}_{\alpha 2}. \end{cases} \quad (11)$$

As  $\epsilon$  goes to zero,  $\boldsymbol{e}_{\alpha 1}$  and  $\boldsymbol{e}_{\alpha 2}$  converges to zero exponentially with a convergence rate of  $-\frac{1}{\epsilon}$ . The  $\boldsymbol{\theta}$  dynamics are considered slow, while  $\boldsymbol{e}_\alpha$  dynamics is referred as a fast one. By singular perturbation theory [40], it can be shown that  $\|\boldsymbol{\theta}(t) - \hat{\boldsymbol{\theta}}(t)\| = O(\epsilon)$  or  $\|\boldsymbol{\theta}(t) - \hat{\boldsymbol{\theta}}(t)\| \leq K\epsilon$  for a constant  $K > 0$ , where  $\hat{\boldsymbol{\theta}}(t) = [\hat{\boldsymbol{\theta}}_1(t)^T \hat{\boldsymbol{\theta}}_2(t)^T]^T$  is the solution of  $\dot{\hat{\boldsymbol{\theta}}}_1 = \hat{\boldsymbol{\theta}}_2$ ,  $\dot{\hat{\boldsymbol{\theta}}}_2 = \mathbf{f}_\theta(\hat{\boldsymbol{\theta}}, \boldsymbol{\alpha}_d, \boldsymbol{u}(\hat{\boldsymbol{\alpha}}_{d2}, \boldsymbol{u}_f))$ .

Since estimating  $\boldsymbol{\theta}$  takes much less computational effort than obtaining  $\boldsymbol{\theta}$  by (11), we formulate the MPC state dynamic model to drive  $\hat{\boldsymbol{\theta}}$  to follow  $\boldsymbol{\theta}_d$ , and similar to (7), the estimated state dynamics are considered as

$$\begin{cases} \dot{\hat{\boldsymbol{\theta}}}_1 = \hat{\boldsymbol{\theta}}_2, \dot{\hat{\boldsymbol{\theta}}}_2 = \mathbf{f}_\theta(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\alpha}}, \boldsymbol{u}(\hat{\boldsymbol{w}}, \boldsymbol{u}_f)), \\ \dot{\hat{\boldsymbol{\alpha}}}_1 = \hat{\boldsymbol{\alpha}}_2, \dot{\hat{\boldsymbol{\alpha}}}_2 = \hat{\boldsymbol{w}} \end{cases} \quad (12)$$

with  $\hat{\boldsymbol{\alpha}}_1 = \boldsymbol{\alpha}_{d1}$ ,  $\hat{\boldsymbol{\alpha}}_2 = \boldsymbol{\alpha}_{d2}$ , and  $\hat{\boldsymbol{w}} = \dot{\boldsymbol{\alpha}}_{d2}$ . We define  $\hat{\boldsymbol{x}} = [\hat{\boldsymbol{\theta}}^T \hat{\boldsymbol{\alpha}}^T]^T$  as the state variable of (12). The design variable of the MPC problem is the input trajectory  $\hat{\boldsymbol{w}}$ ,  $\boldsymbol{u}_f$  and the initial values  $\hat{\boldsymbol{\alpha}}_1(0)$  and  $\hat{\boldsymbol{\alpha}}_2(0)$ . Although the form of (12) is the same as (7),  $\hat{\boldsymbol{\alpha}}(0)$  in (12) is a design variable that needs to be determined, while  $\boldsymbol{\alpha}(0)$  in (7) is measured. We will present the MPC formally in Section IV-B.

## IV. GP-BASED PLANNING AND CONTROL

### A. GP-based inverse dynamics control for trajectory stabilization

Controller (6) and dynamics (7) require precise information about  $\mathbf{f}_\theta$  and  $\mathbf{f}_\alpha^{-1}$ . We consider to use GP models to estimate them. In order to use a zero-mean Gaussian distribution in estimation, we re-write model (7) as

$$\begin{cases} \dot{\boldsymbol{\theta}}_1 = \boldsymbol{\theta}_2, \dot{\boldsymbol{\theta}}_2 = \mathbf{f}_\theta(\boldsymbol{\theta}, \boldsymbol{\alpha}, \boldsymbol{u}_d, \boldsymbol{u}_f) \\ \dot{\boldsymbol{\alpha}}_1 = \boldsymbol{\alpha}_2, \dot{\boldsymbol{\alpha}}_2 + \boldsymbol{\kappa}_\alpha(\boldsymbol{\theta}, \boldsymbol{\alpha}, \dot{\boldsymbol{\alpha}}_2, \boldsymbol{u}_f) = \boldsymbol{u}_d, \end{cases} \quad (13)$$

where  $\mathbf{f}_\theta$  and  $\boldsymbol{\kappa}_\alpha$  are unknown functions that need to be estimated. One benefit of representing the model in (7) into (13) is that the inverse dynamics controller becomes

$\mathbf{u}_d = \mathbf{v} + \boldsymbol{\kappa}_\alpha(\boldsymbol{\theta}, \boldsymbol{\alpha}, \mathbf{v}, \mathbf{u}_f)$  with  $\mathbf{v} = \mathbf{v}_{pd}$  specified in (8) and zero-mean GP for  $\boldsymbol{\kappa}_\alpha$  estimation. Since  $\boldsymbol{\kappa}_\alpha$  is estimated by a zero-mean GP model, when the testing input is far away from the training input,  $\boldsymbol{\kappa}_\alpha$  will be close to zero and the inverse dynamics model degenerates to  $\mathbf{u}_d = \mathbf{v} = \mathbf{v}_{pd}$ . The inverse dynamics controller is stable by choosing high feedback gain in (8). By (13), the learning model is formulated as

$$\begin{cases} \dot{\boldsymbol{\theta}}_1 = \boldsymbol{\theta}_2, \dot{\boldsymbol{\theta}}_2 \sim \mathbf{g}p_\theta(\boldsymbol{\theta}, \boldsymbol{\alpha}, \dot{\boldsymbol{\alpha}}_2, \mathbf{u}_f), \\ \dot{\boldsymbol{\alpha}}_1 = \boldsymbol{\alpha}_2, \mathbf{u}_d - \dot{\boldsymbol{\alpha}}_2 \sim \mathbf{g}p_\alpha(\boldsymbol{\theta}, \boldsymbol{\alpha}, \dot{\boldsymbol{\alpha}}_2, \mathbf{u}_f), \end{cases} \quad (14)$$

where  $\mathbf{g}p_\theta$  and  $\mathbf{g}p_\alpha$  are the GP distributions to estimate  $\mathbf{f}_\theta$  and  $\boldsymbol{\kappa}_\alpha$ , respectively. To train these GP models, the inputs are tuple  $\{\boldsymbol{\theta}, \boldsymbol{\alpha}, \dot{\boldsymbol{\alpha}}_2, \mathbf{u}_f\}$  and the outputs are  $\dot{\boldsymbol{\theta}}_2$  and  $\mathbf{u}_d - \dot{\boldsymbol{\alpha}}_2$ . For each output, an individual GP model is built and the GPs for different outputs are assumed independent.

With (14), the control input  $\mathbf{u}_d$  is obtained as

$$\mathbf{u}_d \sim \mathbf{v} + \mathbf{g}p_\alpha(\boldsymbol{\theta}, \boldsymbol{\alpha}, \mathbf{v}, \mathbf{u}_f), \quad (15)$$

where  $\mathbf{g}p_\alpha(\boldsymbol{\theta}, \boldsymbol{\alpha}, \mathbf{v}, \mathbf{u}_f) \sim \mathcal{N}(\boldsymbol{\mu}_\alpha, \boldsymbol{\Sigma}_\alpha)$  is a predictive Gaussian distribution,  $\boldsymbol{\mu}_\alpha$  and  $\boldsymbol{\Sigma}_\alpha$ <sup>1</sup> are input dependent and computed from (54) in Appendix A. Similar to (8),  $\mathbf{v}$  is designed as an inverse dynamics control for  $\dot{\boldsymbol{\alpha}}_2$  as

$$\mathbf{v} = \hat{\mathbf{w}} - \frac{k_d}{\epsilon} [\boldsymbol{\alpha}_2 - \hat{\boldsymbol{\alpha}}_2(0)] - \frac{k_p}{\epsilon^2} [\boldsymbol{\alpha}_1 - \hat{\boldsymbol{\alpha}}_1(0)] + \mathbf{r}(t), \quad (16)$$

where  $\{\hat{\mathbf{w}}, \hat{\boldsymbol{\alpha}}_1(0), \hat{\boldsymbol{\alpha}}_2(0)\}$  are solutions of the MPC design that will be given in the next section.  $\mathbf{r}(t)$  is an auxiliary control input to be determined later in this section. By (15),  $\mathbf{u}_d \sim \mathcal{N}(\boldsymbol{\mu}_d, \boldsymbol{\Sigma}_d)$  is a Gaussian distribution with  $\boldsymbol{\mu}_d = \mathbf{v} + \boldsymbol{\mu}_\alpha$  and  $\boldsymbol{\Sigma}_d = \boldsymbol{\Sigma}_\alpha$ . The mean value  $\boldsymbol{\mu}_d$  is used as the control input.

Under the inverse dynamics controllers (15) and (16), we now show that the  $\boldsymbol{\alpha}$  subdynamics is stabilized to  $\hat{\boldsymbol{\alpha}}$ . Plugging (15) into  $\boldsymbol{\alpha}$  dynamics (13), the closed-loop  $\Sigma_i$  subsystem dynamics are

$$\begin{cases} \dot{\boldsymbol{\alpha}}_1 = \boldsymbol{\alpha}_2, \\ \dot{\boldsymbol{\alpha}}_2 = \mathbf{v} + \boldsymbol{\mu}_\alpha - \boldsymbol{\kappa}_\alpha(\dot{\boldsymbol{\alpha}}_2) \end{cases} \quad (17)$$

and the error dynamics for  $\Sigma_i$  are

$$\dot{\mathbf{e}}_\alpha = \mathbf{A}\mathbf{e}_\alpha + \mathbf{B}[\mathbf{r}(t) + \boldsymbol{\mu}_\alpha - \boldsymbol{\kappa}_\alpha(\dot{\boldsymbol{\alpha}}_2)], \quad (18)$$

where

$$\mathbf{A} = \begin{bmatrix} 0 & \mathbf{I}_n \\ -\frac{k_p}{\epsilon^2}\mathbf{I}_n & -\frac{k_d}{\epsilon}\mathbf{I}_n \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0 \\ \mathbf{I}_n \end{bmatrix}. \quad (19)$$

Note that  $\mathbf{A}$  is Hurwitz when  $k_p > 0$  and  $k_d > 0$ . To show convergence of  $\mathbf{e}_\alpha$ , it is required that the  $n$ -dimensional disturbance  $\boldsymbol{\mu}_\alpha - \boldsymbol{\kappa}_\alpha(\dot{\boldsymbol{\alpha}}_2)$  is bounded. The disturbance terms  $\boldsymbol{\mu}_\alpha$  and  $\boldsymbol{\kappa}_\alpha(\dot{\boldsymbol{\alpha}}_2)$  have different inputs, that is, the latter has input  $\dot{\boldsymbol{\alpha}}_2$  while the former has  $\mathbf{v}$ . We first analyze the error  $\|\boldsymbol{\mu}_\alpha - \boldsymbol{\kappa}_\alpha(\mathbf{v})\|$ . From Lemma A.3, the modeling error is bounded statistically, namely, for any  $0 < \delta < 1$ ,

$$\text{Pr}\{\|\boldsymbol{\mu}_\alpha - \boldsymbol{\kappa}_\alpha(\mathbf{v})\| \leq \|\boldsymbol{\beta}_\alpha^T \boldsymbol{\Sigma}_\alpha^{\frac{1}{2}}\|\} \geq (1 - \delta)^n, \quad (20)$$

<sup>1</sup>We here drop dependency on  $(\boldsymbol{\theta}, \boldsymbol{\alpha}, \mathbf{v}, \mathbf{u}_f)$  for variables  $\boldsymbol{\mu}_\alpha$  and  $\boldsymbol{\Sigma}_\alpha$  for presentation convenience. For the same reason, in later presentation, we also drop dependency on  $(\boldsymbol{\theta}, \boldsymbol{\alpha}, \mathbf{u}_f)$  for  $\boldsymbol{\kappa}_\alpha$  and only leave the third argument  $\mathbf{v}$  or  $\dot{\boldsymbol{\alpha}}_2$ .

where  $\boldsymbol{\beta}_\alpha$  is  $n$ -dimensional vector with the  $i$ th element  $\beta_{\alpha,i} = \sqrt{2\|\boldsymbol{\kappa}_{\alpha,i}\|_k^2 + 300\gamma_{\alpha,i} \ln^3(\frac{N+1}{\delta})}$ . The following assumption is made in order to achieve deterministic statement on the convergence property.

*Assumption 1:* The modeling error of  $\boldsymbol{\kappa}_\alpha$  is bounded for all testing inputs, i.e.,

$$\|\boldsymbol{\mu}_\alpha - \boldsymbol{\kappa}_\alpha(\mathbf{v})\| \leq \|\boldsymbol{\beta}_\alpha^T \boldsymbol{\Sigma}_\alpha^{\frac{1}{2}}\|. \quad (21)$$

Under Assumption 1, the following lemma gives the bound for the disturbance term in (18), namely,  $\boldsymbol{\mu}_\alpha - \boldsymbol{\kappa}_\alpha(\dot{\boldsymbol{\alpha}}_2)$ .

*Lemma 1:* Under Assumption 1, the  $n$ -dimensional disturbance  $\boldsymbol{\mu}_\alpha - \boldsymbol{\kappa}_\alpha(\dot{\boldsymbol{\alpha}}_2)$  is upper-bounded, namely,

$$\|\boldsymbol{\mu}_\alpha - \boldsymbol{\kappa}_\alpha(\dot{\boldsymbol{\alpha}}_2)\| \leq \rho(\mathbf{e}_\alpha, \boldsymbol{\theta}), \quad (22)$$

where  $\rho(\mathbf{e}_\alpha, \boldsymbol{\theta}) = \lambda_{\min}^{-1}(\mathbf{A}_\kappa) \left( \sum_{i=0}^2 c_i \|\mathbf{e}_\alpha\|^i + \|\boldsymbol{\beta}_\alpha^T \boldsymbol{\Sigma}_\alpha^{\frac{1}{2}}\| \right)$  with  $\mathbf{A}_\kappa = \mathbf{I} + \frac{\partial \boldsymbol{\kappa}_\alpha}{\partial \mathbf{v}}$  and constants  $c_i, i = 0, 1, 2$ , are defined in Appendix C.

The proof of Lemma 1 is included in Appendix C. Since the disturbance term in (18) is upper-bounded, the auxiliary control term  $\mathbf{r}(t)$  is designed according to [39] (Theorem 1 in Chapter 8.4) such that (18) is robustly stable. The following lemma gives the choice of  $\mathbf{r}(t)$  and the convergence property of  $\mathbf{e}_\alpha$ .

*Lemma 2:* Supposing  $k_d^2 > 4k_p > 0$  such that matrix  $\mathbf{A}$  in (19) has real eigenvalues.  $\mathbf{A}$  is diagonalizable with  $\mathbf{A} = \mathbf{M}\boldsymbol{\Lambda}\mathbf{M}^{-1}$ , where  $\boldsymbol{\Lambda}$  is the diagonal matrix and  $\mathbf{M}$  is a non-singular matrix. The auxiliary control  $\mathbf{r}(t)$  is designed as

$$\mathbf{r}(t) = \begin{cases} -\rho(\mathbf{e}_\alpha, \boldsymbol{\theta}) \frac{\mathbf{B}^T \mathbf{P} \mathbf{e}_\alpha}{\|\mathbf{B}^T \mathbf{P} \mathbf{e}_\alpha\|}, & \text{if } \|\mathbf{B}^T \mathbf{P} \mathbf{e}_\alpha\| > \xi \\ -\frac{\rho(\mathbf{e}_\alpha, \boldsymbol{\theta})}{\xi} \mathbf{B}^T \mathbf{P} \mathbf{e}_\alpha, & \text{if } \|\mathbf{B}^T \mathbf{P} \mathbf{e}_\alpha\| \leq \xi \end{cases} \quad (23)$$

with constant  $\xi > 0$  and positive definite matrix  $\mathbf{P}$  is the solution of the Lyapunov equation  $\mathbf{A}^T \mathbf{P} + \mathbf{P} \mathbf{A} = -\mathbf{Q} = \mathbf{M}^{-T} \mathbf{M}^{-1}$ . Under control (23), the values of error  $\|\mathbf{e}_\alpha(t)\|$  satisfy

$$\|\mathbf{e}_\alpha(t)\| \leq d_1 \|\mathbf{e}_\alpha(0)\| e^{\frac{\lambda_1}{4\epsilon} t} + d_2, \quad (24)$$

where  $\lambda_1 = \frac{-k_d + \sqrt{k_d^2 - 4k_p}}{2}$ ,  $d_1 = \sqrt{\frac{\lambda_{\max}(\mathbf{P})}{\lambda_{\min}(\mathbf{P})}}$ ,  $d_2 = \sqrt{\frac{2\epsilon c_3}{\lambda_1 \lambda_{\min}(\mathbf{P})}}$  and constant  $c_3 > 0$  is defined in (60).

The proof of Lemma 2 is given in Appendix D. Since  $\lambda_1 < 0$ , as positive parameter  $\epsilon$  approaches to zero, term  $e^{\frac{\lambda_1}{4\epsilon} t}$  converges to zero rapidly. The GP-based inverse dynamics controller derived above only uses the mean value  $\boldsymbol{\mu}_d$  of the predictive distribution (15). From Lemma 1, covariance  $\boldsymbol{\Sigma}_d = \boldsymbol{\Sigma}_\alpha$  of the predictive distribution determines the disturbance error bound  $\rho(\mathbf{e}_\alpha, \boldsymbol{\theta})$  and from Lemma 2,  $\boldsymbol{\Sigma}_d$  also determines the control performance of  $\mathbf{e}_\alpha$ . We will incorporate  $\boldsymbol{\Sigma}_d$  information into the MPC design to enhance the control performance of  $\mathbf{e}_\alpha$ .

## B. MPC-based planning and control

Applying controllers (15) and (16) to the robot dynamics model (13), the closed-loop dynamics becomes

$$\begin{cases} \dot{\boldsymbol{\theta}}_1 = \boldsymbol{\theta}_2, \dot{\boldsymbol{\theta}}_2 = \mathbf{f}_\theta(\boldsymbol{\theta}, \hat{\boldsymbol{\alpha}} + \mathbf{e}_\alpha, \mathbf{u}_d(\hat{\mathbf{w}} + \dot{\mathbf{e}}_{\alpha_2}, \mathbf{u}_f), \mathbf{u}_f), \\ \dot{\mathbf{e}}_\alpha = \mathbf{A}\mathbf{e}_\alpha + \mathbf{B}[\mathbf{r}(t) + \boldsymbol{\mu}_\alpha - \boldsymbol{\kappa}_\alpha(\dot{\boldsymbol{\alpha}}_2)]. \end{cases} \quad (25)$$

We use  $\alpha = \hat{\alpha} + e_\alpha$  and  $\dot{\alpha}_2 = \hat{w} + \dot{e}_{\alpha_2}$  in argument of  $f_\theta(\cdot)$ . In the previous section, we have shown the convergence property of  $e_\alpha$ . In this section, we discuss how to use MPC to obtain the desired internal subsystem profiles  $\mathbf{u}_f$ ,  $\hat{w}$  and  $\hat{\alpha}(0)$ . A learned GP model  $gp_\theta$  is used to predict unknown function  $f_\theta$  in (25). By singular perturbation theory, assuming  $e_\alpha$  converges to zero rapidly, similar to (12), we obtain the reduced system dynamics as

$$\begin{cases} \dot{\hat{\theta}}_1 = \hat{\theta}_2, \hat{\theta}_2 \sim gp_\theta(\hat{\theta}, \hat{\alpha}, \hat{w}, \mathbf{u}_f), \\ \dot{\hat{\alpha}}_1 = \hat{\alpha}_2, \hat{\alpha}_2 = \hat{w}. \end{cases} \quad (26)$$

For presentation convenience, we use discrete-time representation of the above dynamics for MPC design as follows<sup>2</sup>.

$$\begin{cases} \Delta \hat{\theta}_1(k) = \hat{\theta}_2(k) \Delta t, \Delta \hat{\theta}_2(k) \sim gp_{\hat{\theta}}(k) \Delta t, \\ \Delta \hat{\alpha}_1(k) = \hat{\alpha}_2(k) \Delta t, \Delta \hat{\alpha}_2(k) = \hat{w}(k) \Delta t, \end{cases} \quad (27)$$

where  $\Delta t$  is the sampling period,  $\Delta \hat{\theta}_i(k) = \hat{\theta}_i(k+1) - \hat{\theta}_i(k)$ ,  $\Delta \hat{\alpha}_i(k) = \hat{\alpha}_i(k+1) - \hat{\alpha}_i(k)$ ,  $i = 1, 2$ . We use  $\hat{\theta}(k+i|k)$ ,  $i = 0, \dots, H+1$ , to denote the predicted state variable at the  $(k+i)$ th step given the  $k$ th observation  $\theta(k)$ .  $H$  is the prediction horizon and  $\hat{\theta}(k|k) = \theta(k)$ . We rewrite (27) as

$$\hat{\theta}(k+i+1|k) \sim \mathbf{F} \hat{\theta}(k+i|k) + \mathbf{G} gp_{\hat{\theta}}(k+i), \quad (28)$$

where

$$\mathbf{F} = \begin{bmatrix} \mathbf{I}_m & \Delta t \mathbf{I}_m \\ \mathbf{0}_m & \mathbf{I}_m \end{bmatrix}, \mathbf{G} = \begin{bmatrix} \mathbf{0}_m \\ \Delta t \mathbf{I}_m \end{bmatrix}. \quad (29)$$

$\hat{\theta}(k+i+1|k)$  generally does not satisfy Gaussian distribution even if  $\hat{\theta}(k+i|k)$  is a Gaussian process. To make this prediction manageable, we adopt a linearization method in [16] and the approximation of  $\hat{\theta}(k+i+1|k)$  is a Gaussian distribution with the mean and covariance respectively as

$$\boldsymbol{\mu}_{\hat{\theta}}(k+i+1|k) = \mathbf{F} \boldsymbol{\mu}_{\hat{\theta}}(k+i|k) + \mathbf{G} \boldsymbol{\mu}_{gp_{\hat{\theta}}}(k+i), \quad (30a)$$

$$\boldsymbol{\Sigma}_{\hat{\theta}}(k+i+1|k) = \mathbf{F} \boldsymbol{\Sigma}_{\hat{\theta}}(k+i|k) \mathbf{F}^T + \mathbf{G} \partial \boldsymbol{\Sigma}_{\hat{\theta}}(k+i) \mathbf{G}^T, \quad (30b)$$

where  $\boldsymbol{\mu}_{gp_{\hat{\theta}}}$  and  $\boldsymbol{\Sigma}_{gp_{\hat{\theta}}}$  are the mean and covariance functions of the Gaussian process  $gp_{\hat{\theta}}$ , respectively,  $\partial \boldsymbol{\Sigma}_{\hat{\theta}}(k+i) = \frac{\partial \boldsymbol{\mu}_{gp_{\hat{\theta}}}}{\partial \boldsymbol{\theta}} \boldsymbol{\Sigma}_{\hat{\theta}}(k+i|k) \frac{\partial \boldsymbol{\mu}_{gp_{\hat{\theta}}}}{\partial \boldsymbol{\theta}} + \boldsymbol{\Sigma}_{gp_{\hat{\theta}}}(k+i)$ . Note that  $\boldsymbol{\Sigma}_{\hat{\theta}}(k+i)$  is input dependent and  $\boldsymbol{\mu}_{\hat{\theta}}(k|k) = \hat{\theta}(k) = \theta(k)$ .

By Lemma A.1, it is straightforward to have  $\|\boldsymbol{\Sigma}_{gp_{\hat{\theta}}}\| \leq \sigma_{\mathbf{f}}^2 \max := \max_{1 \leq j \leq m} (\sigma_{f_j}^2 + \sigma_j^2)$ , where  $j$  is the index of the dimension of  $\mathbf{f}_\theta$ . The following lemma gives a bound of the state covariance  $\boldsymbol{\Sigma}_{\hat{\theta}}(k+i|k)$ .

*Lemma 3:* Assuming that  $\boldsymbol{\mu}_{gp_{\hat{\theta}}}$  has a bounded gradient, with a small  $\Delta t$ , we have

$$\|\boldsymbol{\Sigma}_{\hat{\theta}}(k+i|k)\| \leq i(\Delta t)^2 \|\boldsymbol{\Sigma}_{gp_{\hat{\theta}}}\| \leq i(\Delta t)^2 \sigma_{\mathbf{f}}^2 \max.$$

<sup>2</sup>For notation clarity, we drop all arguments for the GP model and use  $gp_{\hat{\theta}}(k)$  to represent  $gp_\theta(\hat{\theta}(k), \hat{\alpha}(k), \hat{w}(k), \mathbf{u}_f(k))$ .

The proof of Lemma 3 is given in Appendix E. For the reduced system dynamics (26), the objective function of the MPC is first considered as

$$\begin{aligned} \bar{J}_{\hat{\theta}, \hat{W}_H}^k &= \sum_{i=0}^H \left[ \mathbb{E} \|e_{\hat{\theta}}(k+i)\|_{\mathbf{Q}_1}^2 + \|\hat{w}(k+i)\|_R^2 + \|\hat{\alpha}(k)\|_{\mathbf{Q}_2}^2 \right. \\ &\quad \left. + \|\mathbf{u}_f(k+i)\|_R^2 \right] + \mathbb{E} \|e_{\hat{\theta}}(k+H+1)\|_{\mathbf{Q}_3}^2 \\ &= \sum_{i=0}^H l_s(k+i) + l_f(k+H+1) + \|\hat{\alpha}(k)\|_{\mathbf{Q}_2}^2, \end{aligned} \quad (31)$$

where  $e_{\hat{\theta}}(k+i) = \hat{\theta}(k+i|k) - \theta_d(k+i)$ , matrices  $\mathbf{Q}_i$ ,  $i = 1, 2, 3$ , and  $\mathbf{R}$  are positive definite. In (31), the stage cost  $l_s(j)$ ,  $j = k+i$ , is defined as

$$\begin{aligned} l_s(j) &= \mathbb{E} [\|e_{\hat{\theta}}(j)\|_{\mathbf{Q}_1}^2] + \|\hat{w}(j)\|_R^2 + \|\mathbf{u}_f(j)\|_R^2 \\ &= \|\boldsymbol{\mu}_{\hat{\theta}}(j)\|_{\mathbf{Q}_1}^2 + \text{tr}(\mathbf{Q}_1 \boldsymbol{\Sigma}_{\hat{\theta}}(j|k)) + \|\hat{w}(j)\|_R^2 + \|\mathbf{u}_f(j)\|_R^2, \end{aligned} \quad (32)$$

where  $\boldsymbol{\mu}_{\hat{\theta}}(j) := \boldsymbol{\mu}_{\hat{\theta}}(j|k) - \theta_d(j)$ . Similarly, the terminal cost  $l_f(k+H+1)$  is defined as

$$\begin{aligned} l_f(k+H+1) &= \mathbb{E} [\|e_{\hat{\theta}}(k+H+1)\|_{\mathbf{Q}_3}^2] \\ &= \|\boldsymbol{\mu}_{\hat{\theta}}(k+H+1)\|_{\mathbf{Q}_3}^2 + \text{tr}(\mathbf{Q}_3 \boldsymbol{\Sigma}_{\hat{\theta}}(k+H+1|k)). \end{aligned} \quad (33)$$

The  $k$ th-step MPC input variable is

$$\hat{W}(k) = \{\hat{\alpha}(k), \hat{w}(k+i), \mathbf{u}_f(k+i), i = 0, \dots, H\}. \quad (34)$$

We take expectation operator in (31) because  $\theta(k+i)$  is approximated by the probabilistic variable  $\hat{\theta}(k+i|k)$  from (30).

The distribution dynamics (30) is used to predict the future trajectory and this gives computational benefit. The objective function (31) however does not penalize  $\alpha$  convergence. In fact, the convergence of  $e_\alpha$  affects  $\theta$  tracking performance as shown in (25). To include the penalty on the internal subsystem tracking performance, we modify the MPC objective function as

$$J_{\hat{\theta}, \hat{W}_H}^k = \bar{J}_{\hat{\theta}, \hat{W}_H}^k + \nu \|\boldsymbol{\Sigma}_d(k)\|, \quad (35)$$

where  $\boldsymbol{\Sigma}_d(k)$  is the covariance of the predictive distribution (15) at the  $k$ th step and  $\nu > 0$  is a weighting factor. The rationale to include  $\boldsymbol{\Sigma}_d$  in the cost function is to incorporate the inverse dynamics model uncertainty in the MPC design. As shown in Lemmas 1 and 2, the convergence property of  $e_\alpha$  depends on  $\boldsymbol{\Sigma}_d$  values. For a small value of  $\boldsymbol{\Sigma}_d$ , the MPC picks up the desired trajectory that can be stabilized by the inverse dynamics controller with high confidence. The significance of adding term  $\boldsymbol{\Sigma}_d$  into the objective function will be demonstrated in Section VI.

The control input by the MPC design is denoted as

$$\hat{W}^*(k) = \underset{\hat{W}(k)}{\text{argmin}} J_{\hat{\theta}, \hat{W}}^k. \quad (36)$$

The optimization is formulated as an unconstrained MPC and solved with gradient decent method. The control input  $\hat{W}^*(k)$  is used in the inverse dynamics controller (16). Two remarks need to be clarified before the MPC convergence is shown rigorously. First, the approximated model (30) is used instead of the inaccessible model (25) to compute the state prediction. The impact of using this approximation on tracking stability

will be discussed in Section V. Second, although prediction  $\mu_{\hat{\theta}}$  from (30) is an accurate approximation of  $\theta$  for (25), the convergence of  $\mu_{\hat{\theta}}$  to the desired  $\theta_d$  under controller (36) is not straightforward and needs to be further clarified.

The rest of this subsection is devoted to address the second item above. It should be noted that since the prediction model (30a) for  $\mu_{\hat{\theta}}$  is exact, no difference exists between  $\mu_{\hat{\theta}}(k+i|k)$  and  $\mu_{\hat{\theta}}(k+i|k+j)$ ,  $j \leq i$ , in the discussion of the convergence of  $\mu_{\hat{\theta}}$  to  $\theta_d$ . The input given by (36) does not automatically guarantee the convergence of  $\mu_{\hat{\theta}}$  to  $\theta_d$  because of the finite prediction horizon. As shown in [41], the stability is instead ensured with the appropriate choice of the terminal cost  $l_f(k+H+1)$  and the terminal constraint. We here briefly describe the terminal cost design to ensure this convergence.

Suppose that for the desired trajectory  $\theta_d$ , there exists a corresponding inputs  $\{\alpha_d, \mathbf{w}_d, \mathbf{u}_{f,d}\}$  satisfying the mean propagation dynamics (30a), that is,

$$\theta_d(k+i+1) = \mathbf{F}\theta_d(k+i) + \mathbf{G}\mu_{gp\hat{\theta}}(\theta_d, \alpha_d, \mathbf{w}_d, \mathbf{u}_{f,d}). \quad (37)$$

To show the stability of tracking error  $e_{\mu_{\hat{\theta}}} = \mu_{\hat{\theta}} - \theta_d$  under (36), we assess  $e_{\mu_{\hat{\theta}}}$  dynamics by taking the difference between (37) and (30a), namely,

$$\begin{aligned} e_{\mu_{\hat{\theta}}}(k+i+1) &= \mathbf{F}e_{\mu_{\hat{\theta}}}(k+i) + \mathbf{G}[\mu_{gp\hat{\theta}}(\mu_{\hat{\theta}}, \hat{\alpha}, \hat{\mathbf{w}}, \mathbf{u}_f) \\ &\quad - \mu_{gp\hat{\theta}}(\theta_d, \alpha_d, \mathbf{w}_d, \mathbf{u}_{f,d})]. \end{aligned} \quad (38)$$

Defining the input  $\mathbf{u}_e = [\hat{\alpha}^T - \alpha_d^T, \hat{\mathbf{w}}^T - \mathbf{w}_d^T, \mathbf{u}_f^T - \mathbf{u}_{f,d}^T]^T$ , (38) is then linearized around its equilibrium point at the origin and we obtain

$$e_{\mu_{\hat{\theta}}}(k+i+1) = \mathbf{A}_e e_{\mu_{\hat{\theta}}}(k+i) + \mathbf{B}_e \mathbf{u}_e(k+i), \quad (39)$$

$$\mathbf{A}_e = \mathbf{F} + \mathbf{G} \frac{\partial \mu_{gp\hat{\theta}}}{\partial \theta_d}, \text{ and } \mathbf{B}_e = \mathbf{G} \left[ \frac{\partial \mu_{gp\hat{\theta}}}{\partial \alpha_d}, \frac{\partial \mu_{gp\hat{\theta}}}{\partial \mathbf{w}_d}, \frac{\partial \mu_{gp\hat{\theta}}}{\partial \mathbf{u}_{f,d}} \right]^T.$$

By [41], stability of the error dynamics (38) is guaranteed by the solution  $\hat{\mathbf{W}}^{\otimes}(k)$  of the following MPC problem

$$\hat{\mathbf{W}}^{\otimes}(k) = \operatorname{argmin}_{\hat{\mathbf{W}}(k)} J_{\hat{\theta}, \hat{\mathbf{W}}}^{k*}, \quad (40)$$

where  $J_{\hat{\theta}, \hat{\mathbf{W}}}^{k*} = \sum_{i=0}^H l_s^*(k+i) + l_f^*(k+H+1)$ ,

$$\begin{aligned} l_s^*(k+i) &= \|e_{\mu_{\hat{\theta}}}(k+i)\|_{\mathbf{Q}_1^*}^2 + \|e_{\hat{\alpha}}(k+i)\|_{\mathbf{Q}_2^*}^2 \\ &\quad + \|\Delta \hat{\mathbf{w}}(k+i)\|_{\mathbf{R}^*}^2 + \|\Delta \mathbf{u}_d\|_{\mathbf{R}^*}^2, \end{aligned} \quad (41a)$$

$$l_f^*(k+H+1) = \|e_{\mu_{\hat{\theta}}}(k+H+1)\|_{\mathbf{Q}_3^*}^2, \quad (41b)$$

$e_{\hat{\alpha}}(k+i) = \hat{\alpha}(k+i) - \alpha_d(k+i)$ ,  $\Delta \hat{\mathbf{w}}(k+i) = \hat{\mathbf{w}}(k+i) - \mathbf{w}_d(k+i)$ , and  $\Delta \mathbf{u}_d = \mathbf{u}_f(k+i) - \mathbf{u}_{f,d}(k+i)$ . Positive definite matrices  $\mathbf{Q}_i^*$ ,  $i = 1, 2, 3$ , and  $\mathbf{R}^*$  are chosen for design specification. [41] proposed a systematic approach to design the terminal cost matrix  $\mathbf{Q}_3^*$  and the corresponding terminal region  $\Omega_e$ . Within  $\Omega_e$ , a linear state feedback controller  $\mathbf{u}_e = -\mathbf{K}_e e_{\mu_{\hat{\theta}}}$  (with gain  $\mathbf{K}_e$ ) for (39) ensures the stability of the original dynamics (38) with the decreasing terminal cost, that is, if  $e_{\mu_{\hat{\theta}}}(k+H+1) \in \Omega_e$ , controller  $\mathbf{u}_e$  results in  $e_{\mu_{\hat{\theta}}}(k+H+2) \in \Omega_e$  with  $l_f^*(k+H+2) \leq l_f^*(k+H+1) - l_s^*(k+H+1)$ .

Taking the MPC objective function (40) under the optimal input, we have

$$\begin{aligned} J_{\hat{\theta}^*, \hat{\mathbf{W}}^{\otimes}}^{(k+1)*} - J_{\hat{\theta}^*, \hat{\mathbf{W}}^{\otimes}}^{k*} &= -l_s^*(k) + l_f^*(k+H+2) - \\ &\quad l_f^*(k+H+1) + l_s^*(k+H+1) \leq -l_s^*(k). \end{aligned}$$

From the monotonicity of  $J_{\hat{\theta}^*, \hat{\mathbf{W}}^{\otimes}}^{k*}$ , namely,  $\|e_{\mu_{\hat{\theta}}}(k)\|_{\mathbf{Q}_1^*}^2 \leq J_{\hat{\theta}^*, \hat{\mathbf{W}}^{\otimes}}^{k*} \leq \|e_{\mu_{\hat{\theta}}}(k)\|_{\mathbf{Q}_3^*}^2$ , we have

$$J_{\hat{\theta}^*, \hat{\mathbf{W}}^{\otimes}}^{(k+1)*} \leq \left[ 1 - \frac{\lambda_{\min}(\mathbf{Q}_1^*)}{\lambda_{\max}(\mathbf{Q}_3^*)} \right] J_{\hat{\theta}^*, \hat{\mathbf{W}}^{\otimes}}^{k*}.$$

Comparing the MPC problem (36) with (40), we notice two major differences. The first one is that the former one includes the model uncertainty through the covariance term (i.e.,  $\nu \|\Sigma_d(k)\|$ ). The second difference is that the former does not need the desired input trajectories  $\alpha_d$ ,  $\mathbf{w}_d$  and  $\mathbf{u}_{f,d}$ , which are difficult to obtain. The MPC problem (36) only assumes that the desired trajectories exist but no need to be known. This is one of the attractive properties of the proposed control design.

To apply the result of (40) to show the stability of the MPC design in (36), the following lemma is needed.

*Lemma 4:* For terminal cost  $l_f^*(k+H+1)$  and the stage cost  $l_s^*(k+i)$  defined in (41), let matrices  $\mathbf{Q}_1$  and  $\mathbf{R}$  in (32) and  $\mathbf{Q}_3$  in (33) satisfy  $\mathbf{Q}_1 = \mathbf{Q}_1^*$ ,  $\lambda_{\max}(\mathbf{R}) < \lambda_{\min}(\mathbf{R}^*)$  and  $\mathbf{Q}_3 = \mathbf{Q}_3^*$ , then

$$\begin{aligned} l_f(k+H+2) &\leq l_f(k+H+1) + \operatorname{tr}(\mathbf{Q}_3 \Sigma_{\hat{\theta}}(k+H+2)) \\ &\quad - l_s(k+H+1) + \operatorname{tr}(\mathbf{Q}_1 \Sigma_{\hat{\theta}}(k+H+1)) \end{aligned}$$

if the following conditions are satisfied

$$\|\hat{\mathbf{w}}(k+H+1)\| \geq \lambda_R \|\mathbf{w}_d(k+H+1)\|, \quad (42a)$$

$$\|\mathbf{u}_f(k+H+1)\| \geq \lambda_R \|\mathbf{u}_{f,d}(k+H+1)\|, \quad (42b)$$

where  $\lambda_R = \frac{2\lambda_{\min}(\mathbf{R}^*)}{\lambda_{\min}(\mathbf{R}^*) - \lambda_{\max}(\mathbf{R})}$ .

The proof of this lemma is included in Appendix F. With the result in Lemma 4, we obtain the bound of tracking error  $e_{\mu_{\hat{\theta}}}(k+i) := \mu_{\hat{\theta}}(k+i|k) - \theta_d(k+i)$ ,  $i = 0, \dots, H+1$ , through the following lemma.

*Lemma 5:* Using  $J_{\hat{\theta}, \hat{\mathbf{W}}}^k$  under the optimal input (36) as the Lyapunov function candidate, the tracking error satisfies  $\|e_{\mu_{\hat{\theta}}}(k+i)\| \leq a_4(i) \|e_{\theta}(k)\| + a_5(i)$  where  $a_4(i) =$

$$\begin{aligned} d_3^{\frac{i}{2}} \sqrt{\frac{\lambda_{\max}(\mathbf{Q}_3)}{\lambda_{\min}(\mathbf{Q}_1)}}, \quad a_5(i) &= \sqrt{\frac{d_3^i (\alpha_{\max}^2 + \nu \sigma_{\kappa \max}^2) + d_4 \frac{1-d_3^i}{1-d_3}}{\lambda_{\min}(\mathbf{Q}_1)}}, \quad 0 < \\ d_3 &= 1 - \frac{\lambda_{\min}(\mathbf{Q}_1)}{\lambda_{\max}(\mathbf{Q}_3)} < 1, \quad d_4 = m \lambda_m (H+2) (\Delta t)^2 \sigma_f^2 \max + \\ &\quad (1 + \frac{\lambda_{\min}(\mathbf{Q}_1)}{\lambda_{\max}(\mathbf{Q}_3)}) (\nu \sigma_{\kappa \max}^2 + \alpha_{\max}^2), \text{ and } \lambda_m = \lambda_{\max}(\mathbf{Q}_1) + \lambda_{\max}(\mathbf{Q}_3). \end{aligned}$$

The proof of Lemma 5 is included in Appendix G. Because  $0 < d_3 < 1$ ,  $a_4(i)$  converges to zero and  $a_5(i)$  converges to  $\sqrt{\frac{d_4}{(1-d_3)\lambda_{\min}(\mathbf{Q}_1)}}$  exponentially as  $i$  goes to infinity. Since  $J_{\hat{\theta}, \hat{\mathbf{W}}}^k$  is positive definite, Lemma 5 confirms that if we choose  $J_{\hat{\theta}, \hat{\mathbf{W}}}^k$  as the Lyapunov function candidate, the values of  $J_{\hat{\theta}, \hat{\mathbf{W}}}^k$  decrease along the trajectory predicted from model (30) as long as (42) holds. This implies that by solving the MPC problem (36), the mean value variable  $\mu_{\hat{\theta}}$  predicted by (30) is stabilized to track  $\theta_d$  exponentially.

In summary, Fig. 2 illustrates the framework of GP-based control design. In each control step, the trajectory planner solves the MPC problem (36) with model (30) and generates the planned internal subsystem trajectory  $\hat{\mathbf{W}}^*$ . The MPC also incorporates the predictive variance  $\Sigma_d$  from the inverse dynamics model. The inverse dynamics controller takes the

$\hat{\mathbf{W}}^*$  profile and uses (15) and (16) to compute the control input  $\boldsymbol{\mu}_d$ . In the framework, the prediction uncertainty (i.e.,  $\boldsymbol{\Sigma}_d$ ) is used in both the MPC-based trajectory planner and the inverse dynamics stabilization.

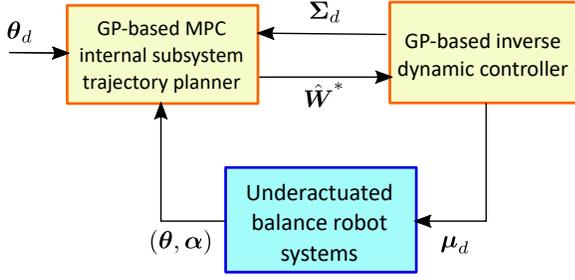


Fig. 2. Schematic flow of the GP-based control framework.

## V. CONTROL PERFORMANCE ANALYSIS

In this section, we show stability and performance analysis of the control design and also discuss the impact of learning-based model errors on controller performance.

Under the learning-based controller (15) and (16), we have the closed-loop dynamics (25). Assuming that both models (13) and (25) are deterministic, we consider a Lyapunov function candidate

$$V(k) = V_\theta(k) + \zeta V_\alpha(k), \quad (43)$$

where constant  $\zeta > 0$ ,  $V_\alpha(k) = \mathbf{e}_\alpha^T(k) \mathbf{P} \mathbf{e}_\alpha(k)$ ,  $\mathbf{P}$  is defined in Lemma 2, and  $V_\theta(k)$  is similar to the MPC cost function in (31) without expectation operator and under the optimal control  $\hat{\mathbf{W}}^*(k)$ , namely,

$$\begin{aligned} V_\theta(k) = \bar{J}_{\theta, \hat{\mathbf{W}}^*}^k &= \sum_{i=0}^H [\| \mathbf{e}_\theta(k+i) \|_{Q_1}^2 + \| \hat{\mathbf{W}}^*(k+i) \|_R^2 \\ &+ \| \mathbf{u}_f^0(k+i) \|_R^2] + \| \hat{\boldsymbol{\alpha}}^0(k) \|_{Q_2}^2 \\ &+ \| \mathbf{e}_\theta(k+H+1) \|_{Q_3}^2. \end{aligned} \quad (44)$$

Here  $\mathbf{e}_\theta(k+i) = \boldsymbol{\theta}(k+i) - \boldsymbol{\theta}_d(k+i)$ . Note that  $\bar{J}_{\theta, \hat{\mathbf{W}}^*}^k$  is a quadratic function of the actual state  $\boldsymbol{\theta}(k+i)$  following the unknown deterministic model (25) under  $\hat{\mathbf{W}}^*$  given by (36). At the  $k$ th step, it is impossible to directly evaluate  $\bar{J}_{\theta, \hat{\mathbf{W}}^*}^k$  because inaccessible future states and the unknown model (25), and instead its value is approximated by  $\bar{J}_{\hat{\boldsymbol{\theta}}^*, \hat{\mathbf{W}}^*}^k$  given by (31).

We assess the decreasing property of the proposed Lyapunov function candidate as

$$\begin{aligned} \Delta V(k) &= \left( \bar{J}_{\theta, \hat{\mathbf{W}}^*}^{k+1} - \bar{J}_{\hat{\boldsymbol{\theta}}^*, \hat{\mathbf{W}}^*}^{k+1} \right) - \left( \bar{J}_{\theta, \hat{\mathbf{W}}^*}^k - \bar{J}_{\hat{\boldsymbol{\theta}}^*, \hat{\mathbf{W}}^*}^k \right) \\ &+ \left( \bar{J}_{\hat{\boldsymbol{\theta}}^*, \hat{\mathbf{W}}^*}^{k+1} - \bar{J}_{\hat{\boldsymbol{\theta}}^*, \hat{\mathbf{W}}^*}^k \right) + \zeta [V_\alpha(k+1) - V_\alpha(k)] \\ &- \nu \left[ \| \boldsymbol{\Sigma}_d(\hat{\mathbf{W}}^*(k+1)) \| - \| \boldsymbol{\Sigma}_d(\hat{\mathbf{W}}^*(k)) \| \right], \end{aligned} \quad (45)$$

where  $\Delta V(k) = V(k+1) - V(k)$  and  $\bar{J}_{\hat{\boldsymbol{\theta}}^*, \hat{\mathbf{W}}^*}^k = \bar{J}_{\theta, \hat{\mathbf{W}}^*}^k + \nu \| \boldsymbol{\Sigma}_d(\hat{\mathbf{W}}^*(k)) \|$  are used in the above expansion. In (45), term  $\bar{J}_{\theta, \hat{\mathbf{W}}^*}^k - \bar{J}_{\hat{\boldsymbol{\theta}}^*, \hat{\mathbf{W}}^*}^k$  quantifies the difference between the approximated cost-to-go and the actual cost-to-go at the  $k$ th step. We use  $\boldsymbol{\theta}(k+i|k)$  for  $i \geq 0$  to denote the predicted value of  $\boldsymbol{\theta}(k+i)$  given the measured  $\boldsymbol{\theta}(k)$  with the initial condition

$\boldsymbol{\theta}(k|k) = \boldsymbol{\theta}(k)$ . Similar to (28), the evolution of  $\boldsymbol{\theta}(k+i|k)$  follows discretized form of (14), namely,

$$\boldsymbol{\theta}(k+i+1|k) \sim \mathbf{F} \boldsymbol{\theta}(k+i|k) + \mathbf{G} g_{p_\theta}(k+i), \quad (46)$$

with the mean value  $\boldsymbol{\mu}_\theta(k+i+1|k)$  and variance  $\boldsymbol{\Sigma}_\theta(k+i+1|k)$  calculations similar to (30). The difference between models (46) and (28) is that the former depends on the actual internal state  $\boldsymbol{\alpha}(k+i)$ , while the latter uses the estimated internal state  $\hat{\boldsymbol{\alpha}}(k+i|k)$ . Model (28) is actually used for  $\boldsymbol{\theta}$  trajectory prediction through the MPC formulation. Figure 3 further illustrates the relationships among the three different  $\boldsymbol{\theta}$ -prediction models (13), (46) and (28).

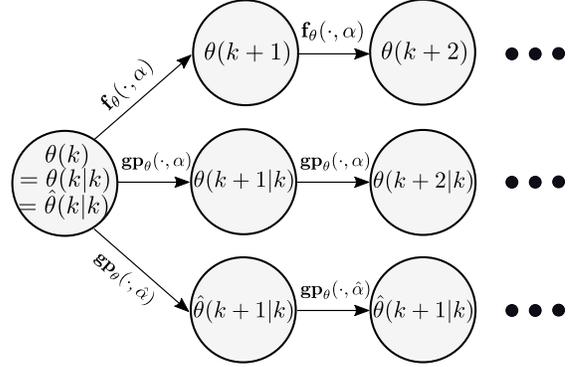


Fig. 3. Flow chart of the state estimation by three predictive models.

We now quantify the difference between  $\hat{\boldsymbol{\theta}}(k+i|k)$  and  $\boldsymbol{\theta}(k+i|k)$ . At  $i=0$ ,  $\hat{\boldsymbol{\theta}}(k|k) = \boldsymbol{\theta}(k|k) = \boldsymbol{\theta}(k)$ . The difference between  $\hat{\boldsymbol{\theta}}(k+i|k)$  and  $\boldsymbol{\theta}(k+i|k)$  comes from the difference between the reduced model (28) and full model (46). We have the following results about their differences.

*Lemma 6:* Assuming the mean value of the predictive distribution  $^3 \boldsymbol{\mu}_{g_{p_\theta}}(\boldsymbol{\mu}_\theta, \boldsymbol{\alpha})$  is Lipschitz in  $\boldsymbol{\mu}_\theta$  and  $\boldsymbol{\alpha}$ , namely,

$$\begin{aligned} \| \boldsymbol{\mu}_{g_{p_\theta}}(\cdot, \boldsymbol{\alpha}) - \boldsymbol{\mu}_{g_{p_\theta}}(\cdot, \hat{\boldsymbol{\alpha}}) \| &\leq L_2 \| \mathbf{e}_\alpha \|, \\ \| \boldsymbol{\mu}_{g_{p_\theta}}(\boldsymbol{\mu}_\theta, \cdot) - \boldsymbol{\mu}_{g_{p_\theta}}(\boldsymbol{\mu}_{\hat{\theta}}, \cdot) \| &\leq L_3 \| \boldsymbol{\mu}_\theta - \boldsymbol{\mu}_{\hat{\theta}} \|, \end{aligned}$$

with constants  $L_2, L_3 > 0$ ,  $\tilde{\boldsymbol{\mu}}_\theta(k+i) := \boldsymbol{\mu}_\theta(k+i|k) - \boldsymbol{\mu}_{\hat{\theta}}(k+i|k)$  satisfies  $\| \tilde{\boldsymbol{\mu}}_\theta(k+i) \| \leq \varrho_\theta(i) \| \mathbf{e}_\alpha(k) \| + \varrho_2(i)$ , where

$$\varrho_\theta(i) = d_1 L_2 \Delta t \left[ \left( \frac{1-a_1^i}{1-a_1} - i \right) \left( 1 - \frac{L_3 \Delta t}{1-a_1} \right) + i \right],$$

$a_1 = e^{\frac{\lambda_1}{4c} \Delta t}$  and  $\varrho_2(i) = d_2 L_2 \Delta t [i + \frac{1}{2} L_3 \Delta t (i-1)]$ .  $d_1, d_2, \lambda_1$  are defined in Lemma 2.

The proof of this lemma is included in Appendix H. We then inspect the difference between  $\boldsymbol{\theta}(k+i|k)$  and  $\boldsymbol{\theta}(k+i)$ . The difference between  $\boldsymbol{\theta}(k+i|k)$  and  $\boldsymbol{\theta}(k+i)$  comes from the difference between the learning model (46) and the unknown actual model (13) as shown in Fig. 3. From Lemma A.3, we obtain the GP learned prediction guaranteed to be closed to the  $m$ -dimensional model  $\mathbf{f}_\theta$  with high probability  $^4$ , namely,

$$\Pr\{ \| \boldsymbol{\mu}_{g_{p_\theta}}(\boldsymbol{\theta}, \boldsymbol{\alpha}) - \mathbf{f}_\theta \| \leq \| \boldsymbol{\beta}_\theta^T \boldsymbol{\Sigma}_{g_{p_\theta}}^{\frac{1}{2}} \| \} \geq (1-\delta)^m, \quad (47)$$

<sup>3</sup>For presentation convenience, we drop the third and four arguments and use notation  $\boldsymbol{\mu}_{g_{p_\theta}}(\boldsymbol{\mu}_\theta, \boldsymbol{\alpha})$  to represent  $\boldsymbol{\mu}_{g_{p_\theta}}(\boldsymbol{\mu}_\theta, \boldsymbol{\alpha}, \hat{\boldsymbol{\alpha}}_2, \mathbf{u}_f)$ .

<sup>4</sup>We here drop all arguments  $(\boldsymbol{\theta}, \boldsymbol{\alpha}, \hat{\boldsymbol{\alpha}}_2, \mathbf{u}_f)$  of functions  $\mathbf{f}_\theta$  and  $\boldsymbol{\Sigma}_{g_{p_\theta}}$  for presentation brevity.

where  $0 < \delta < 1$  and  $\beta_\theta$  is an  $m$ -dimensional vector with its  $j$ th element  $\beta_{\theta,j} = \sqrt{2\|f_{\theta,j}\|_k^2 + 300\gamma_{\theta,j} \ln^3(\frac{N+1}{\delta})}$ .  $\gamma_{\theta,j}$  is the maximum information gain for  $f_{\theta,j}$  ( $j$ th element of  $\mathbf{f}_\theta$ ). To conduct the performance analysis, the following assumption is considered.

*Assumption 2:* The modeling error of  $\mathbf{f}_\theta$  is bounded for all testing inputs, namely,

$$\|\boldsymbol{\mu}_{g_{p\theta}}(\boldsymbol{\theta}, \boldsymbol{\alpha}) - \mathbf{f}_\theta\| \leq \|\beta_\theta^T \Sigma_{g_{p\theta}}^{\frac{1}{2}}\|. \quad (48)$$

Under Assumption 2, the following lemma gives upper-bound of  $\boldsymbol{\theta}_\mu(k+i) := \boldsymbol{\mu}_\theta(k+i|k) - \boldsymbol{\theta}(k+i)$ .

*Lemma 7:* Under Assumption 2, we have  $\|\boldsymbol{\theta}_\mu(k+i)\| \leq \varrho_{\mu_\theta}(i)$ , where  $\varrho_{\mu_\theta}(i) = \Delta t \sum_{j=0}^{i-1} \|\beta_\theta^T \Sigma_{g_{p\theta}}^{\frac{1}{2}}(k+j|k)\|$ . The proof of this lemma is included in Appendix I. Lemmas 6 and 7 give the error bounds on  $\tilde{\boldsymbol{\mu}}_\theta(k+i) = \boldsymbol{\mu}_\theta(k+i|k) - \boldsymbol{\mu}_{\hat{\theta}}(k+i|k)$  and  $\boldsymbol{\theta}_\mu(k+1) = \boldsymbol{\mu}_\theta(k+i|k) - \boldsymbol{\theta}(k+i)$ , respectively. Combining these results, we have the error bound of  $\tilde{\boldsymbol{\theta}}_\mu(k+i) := \boldsymbol{\mu}_{\hat{\theta}}(k+i|k) - \boldsymbol{\theta}(k+i)$  as

$$\begin{aligned} \|\tilde{\boldsymbol{\theta}}_\mu(k+i)\| &\leq \|\tilde{\boldsymbol{\mu}}_\theta(k+i)\| + \|\boldsymbol{\theta}_\mu(k+i)\| \\ &\leq \varrho_{\hat{\theta}}(i)\|\mathbf{e}_\alpha(k)\| + \varrho_2(i) + \varrho_{\mu_\theta}(i). \end{aligned}$$

By defining  $a_2(i) = \varrho_2(i) + \varrho_{\mu_\theta} > 0$ , we obtain that  $\|\tilde{\boldsymbol{\theta}}_\mu(k+i)\| \leq \varrho_{\hat{\theta}}(i)\|\mathbf{e}_\alpha(k)\| + a_2(i)$ . We estimate the difference of  $\bar{J}_{\hat{\theta}^*, \hat{W}^*}^k - \bar{J}_{\theta^*, \hat{W}^*}^k$  in the following lemma with proof given in Appendix J.

*Lemma 8:* Under Assumptions 1 and 2, we obtain

$$|\bar{J}_{\hat{\theta}^*, \hat{W}^*}^k - \bar{J}_{\theta^*, \hat{W}^*}^k| \leq \rho_J(\mathbf{e}_\alpha, \mathbf{e}_\theta),$$

where

$$\begin{aligned} \rho_J(\mathbf{e}_\alpha, \mathbf{e}_\theta) &= \lambda_{\max}(\mathbf{Q}_3) \sum_{i=0}^{H+1} \left\{ \bar{\xi}_1(i)\|\mathbf{e}_\alpha(k)\|^2 + \right. \\ &\quad \bar{\xi}_3(i)\|\mathbf{e}_\alpha(k)\| + \bar{\xi}_2(i)\|\mathbf{e}_\alpha(k)\| \|\mathbf{e}_\theta(k)\| + \\ &\quad \left. \bar{\xi}_4(i)\|\mathbf{e}_\theta(k)\| + \bar{\xi}_5(i) \right\}, \quad (49) \end{aligned}$$

$\bar{\xi}_1(i) = \varrho_{\hat{\theta}}^2(i)$ ,  $\bar{\xi}_2(i) = 2\varrho_{\hat{\theta}}(i)a_4(i)$ ,  $\bar{\xi}_3(i) = 2\varrho_{\hat{\theta}}(i)[a_2(i) + a_5(i)]$ ,  $\bar{\xi}_4(i) = 2a_2(i)a_4(i)$ , and  $\bar{\xi}_5(i) = a_2(i)(a_2(i) + 2a_5(i)) + mi(\Delta t)^2\sigma_{\mathbf{f}}^2 \max \cdot \varrho_{\hat{\theta}}(i)$  is defined in Lemmas 6,  $a_4(i)$  and  $a_5(i)$  are defined in Lemma 5.

The result in Lemma 8 is used for the first two pairs of terms of  $\Delta V(k)$  in (45). Letting  $\mathbf{e}(k) = [\mathbf{e}_\theta^T(k) \ \mathbf{e}_\alpha^T(k)]^T$  denote the error vector, it is straightforward to obtain that the Lyapunov function candidate  $V(k)$  in (43) satisfies  $\underline{\lambda}\|\mathbf{e}(k)\|^2 \leq V(k) \leq \bar{\lambda}\|\mathbf{e}(k)\|^2$ , where  $\underline{\lambda} = \min(\lambda_{\min}(\mathbf{Q}_1), \zeta\lambda_{\min}(\mathbf{Q}))$  and  $\bar{\lambda} = \max(\lambda_{\max}(\mathbf{Q}_1), \zeta\lambda_{\max}(\mathbf{Q}))$ , where matrices  $\mathbf{Q}$  and  $\mathbf{Q}_1$  are defined in Lemma 2 and (31), respectively. We are now ready to give the following main result.

*Theorem 1:* For parameters  $\bar{\xi}_j(i)$ ,  $i = 0, 1, \dots, H+2$ ,  $j = 1, \dots, 5$ , given in Lemma 8, defining  $\xi_j = \bar{\lambda} \left[ \bar{\xi}_j(0) + 2 \sum_{i=1}^{H+1} \bar{\xi}_j(i) + \bar{\xi}_j(H+2) \right]$ ,  $\gamma_1 = \sqrt{\eta}$ ,  $\gamma_2 = \frac{\xi_3}{2\gamma_1}$ ,  $\gamma_3 = \sqrt{\lambda_{\min}(\mathbf{Q}_1)}$ ,  $\gamma_4 = \frac{\xi_4}{\gamma_3}$ , and  $\gamma_5 = \frac{\xi_4^2}{\gamma_3^2} + \frac{\xi_5^2}{4\gamma_1^2} + \xi_5 + \hat{\alpha}_{\max}^2 + \nu\sigma_{\kappa}^2 \max + \zeta c_3 \Delta t + m\lambda_m(H+2)(\Delta t)^2\sigma_{\mathbf{f}}^2 \max$ , where  $\lambda_m = \lambda_{\max}(\mathbf{Q}_1) + \lambda_{\max}(\mathbf{Q}_3)$  and

$$\eta = \frac{1}{4}\zeta\lambda_{\min}(\mathbf{Q})\Delta t - \xi_1 - \frac{\xi_2^2}{2\lambda_{\min}(\mathbf{Q}_1)} - \frac{\lambda_{\min}(\mathbf{Q}_1)}{4} > 0, \quad (50)$$

the following property is then held

$$V(k+1) \leq \gamma_\lambda V(k) + \gamma_5$$

where  $0 < \gamma_\lambda = 1 - \frac{\gamma_2^2}{4\lambda} < 1$ .

The proof of Theorem 1 is given in Appendix K. If  $V(k+1) \leq \gamma_\lambda V(k) + \gamma_5$  holds for  $i$  consecutive steps, we have

$$V(k+i) \leq \gamma_\lambda^i V(k) + \frac{4\gamma_5 \bar{\lambda}(1 - \gamma_\lambda^i)}{\gamma_3^2}.$$

Introducing the static state values  $V_{ss} = \lim_{i \rightarrow \infty} V(k+i)$  and  $\|\mathbf{e}\|_{ss} = \lim_{i \rightarrow \infty} \|\mathbf{e}(k+i)\|$  for any fixed  $k$ , then  $V_{ss} \leq \frac{4\bar{\lambda}}{\gamma_3^2}\gamma_5$  and  $\|\mathbf{e}\|_{ss} \leq \sqrt{\frac{4\bar{\lambda}}{\gamma_3^2\Delta}}\gamma_5$ .

Theorem 1 implies that the error magnitude  $\|\mathbf{e}\|$  decreases exponentially until  $\|\mathbf{e}\|_{ss} \leq \sqrt{\frac{4\bar{\lambda}}{\gamma_3^2\Delta}}\gamma_5$ . Parameter condition (50) can be satisfied by choosing small enough value for singular perturbation parameter  $\epsilon$ . As  $\epsilon$  value is small,  $\lambda_{\min}(\mathbf{Q})$  becomes large according to Lemma 2 and  $\varrho_{\hat{\theta}}$  goes small according to Lemma 6 and henceforth both  $\xi_1$  and  $\xi_2$  values are small. Modeling errors are also important factors for control performance. As the error bound  $\|\beta_\alpha^T \Sigma_\alpha^{\frac{1}{2}}\|$  for  $\kappa_\alpha$  increases, values of  $d_2$  and  $\varrho_2(i)$  increase,  $a_2(i)$  increases,  $\bar{\xi}_3, \bar{\xi}_4, \bar{\xi}_5$  increase,  $\gamma_5$  increases, and finally the bound of  $\|\mathbf{e}\|_{ss}$  increases. As the error bound  $\|\beta_\theta^T \Sigma_{g_{p\theta}}^{\frac{1}{2}}\|$  for  $\mathbf{f}_\theta$  increases, values of  $\varrho_{\mu_\theta}(i)$  and  $a_2(i)$  increase, therefore both  $\gamma_5$  value and the bound of  $\|\mathbf{e}\|_{ss}$  increase. The results in Theorem 1 are obtained under Assumptions 1 and 2. With enough training data for the learning model,  $\delta$  defined in Lemma A.2 can be chosen small so that Assumptions 1 and 2 are satisfied practically.

## VI. EXPERIMENTS

The learning-based control method is implemented and demonstrated independently on two underactuated balance robotic platforms: a rotary inverted pendulum and a bikebot. Figures 1(a) and 1(b) show these two robotic systems and we present the experimental results in this section.

### A. Experimental testbeds

The rotary inverted pendulum shown in Figure 1(a) is a commercial robotic platform provided by Quanser Inc. In this system, the actuated joint is the base angle  $\theta$  that is driven by a motor. The unactuated joint is the pendulum angle  $\alpha$  and its value is defined to be zero when the pendulum is vertically upright. The voltage of the motor, denoted by  $V_m$ , is the control input to the system. The control goal is to balance the pendulum around upright position, while the rotary base tracks a desired trajectory  $\theta_d$ .

The motion of the external subsystem is captured by angular position  $\theta_1 = \theta$  and velocity  $\theta_2 = \dot{\theta}$ , while the motion of the internal subsystem is modeled by position  $\alpha_1 = \alpha$  and velocity  $\alpha_2 = \dot{\alpha}$ . The control input is  $u_d = V_m$ . Defining  $\boldsymbol{\alpha} = [\alpha_1 \ \alpha_2]^T$ , the dynamic model is

$$\begin{cases} \dot{\theta}_1 = \theta_2, \quad \dot{\theta}_2 = f_\theta(\theta_2, \boldsymbol{\alpha}, u_d), \\ \dot{\alpha}_1 = \alpha_2, \quad \dot{\alpha}_2 + \kappa_\alpha(\theta_2, \boldsymbol{\alpha}, \dot{\alpha}_2) = u_d \end{cases} \quad (51)$$

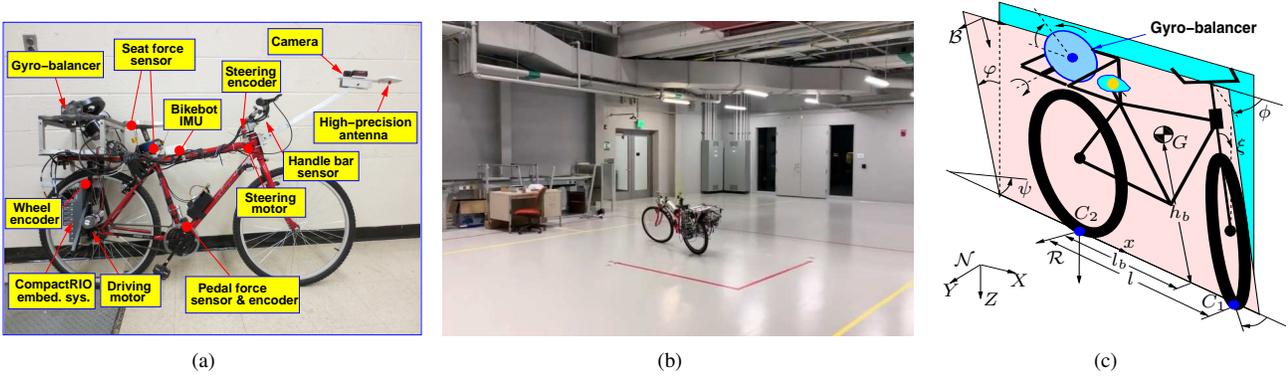


Fig. 4. (a) The autonomous bikebot system with various sensors and actuators developed at Rutgers University. (b) A snapshot of the indoor testing experiment setup. (c) The schematic of the bikebot modeling setup.

with functions  $f_\theta$  and  $\kappa_\alpha$  are given in Appendix L.

The bikebot shown in Fig. 4(a) is a single-tracked vehicle and is equipped with multiple sensors and actuators to study autonomous driving [8] and physical human-robot interactions [42], [43]. Figure 4(b) shows an indoor testing experiment setup that we use in this study. The bikebot position is obtained by a computer vision system with a camera that is mounted on the high ceiling of the lab. The bikebot roll and steering angles are obtained from onboard sensors. The detailed description of the system hardware setup can be found in [8].

Figure 4(c) illustrates the kinematic relationship and configuration of the bikebot system. The bikebot platform consists of a main body structure (with the rear wheel) and a front wheel that is connected with the frame through the steering joint. The rear wheel contact point is denoted as  $C_2$  and its planar coordinate is denoted as  $r_{C_2} = [X \ Y]^T$  in the  $XY$  plane of the inertial frame  $\mathcal{N}$ . The yaw (heading) and roll (with the vertical plane) angles of the bikebot platform are denoted as  $\psi$  and  $\varphi$ , respectively. The steering angle is denoted as  $\phi$  and the rear wheel velocity as  $v_c$ . Due to the nonholonomic constraint of point  $C_2$ , its velocity is obtained as  $v_{C_2} = [\dot{X} \ \dot{Y}]^T = [v_c \cos \psi \ v_c \sin \psi]^T$ . The external subsystem motion of the bikebot is captured by position  $\theta_1 = [X \ Y]^T$  and velocity  $\theta_2 = [\dot{X} \ \dot{Y}]^T$  and the internal subsystem motion is by position  $\alpha_1 = \varphi$  and velocity  $\alpha_2 = \dot{\varphi}$ . The control inputs are  $u = [u_d \ u_f]^T$  with  $u_d = \phi$  and  $u_f = \dot{v}_c$ . The bikebot dynamic model is written as [8]

$$\begin{cases} \dot{\theta}_1 = \theta_2, \quad \dot{\theta}_2 = f_\theta(\theta, \alpha, u), \\ \dot{\alpha}_1 = \alpha_2, \quad \dot{\alpha}_2 + \kappa_\varphi(\theta, \alpha, \dot{\alpha}_2, u_f) = u_d \end{cases} \quad (52)$$

with  $f_\theta$  and  $\kappa_\varphi$  are given in Appendix L. The desired trajectory for the external subsystem is denoted as  $\theta_d = [X_d \ Y_d]^T$ .

## B. Experimental results

1) *Rotary inverted pendulum experiments:* To obtain the learned model of the rotary inverted pendulum, we perturb the system and collect the motion data. An open-loop input is implemented as

$$V_m = \begin{cases} a_1 \sin(\omega_1 t) + a_2 \sin(\omega_2 t), & |\alpha| \leq \frac{\pi}{3}, \\ 0, & |\alpha| > \frac{\pi}{3}, \end{cases} \quad (53)$$

where  $a_1$  and  $a_2$  are chosen to satisfy the input bound  $|V_m| \leq 5$  V,  $\omega_1$  and  $\omega_2$  are designed to excite the system by both low and high frequencies. In experiment, we choose  $a_1 = 3$ ,  $a_2 = 1.5$ ,  $\omega_1 = 8$  rad/s and  $\omega_2 = 40$  rad/s. Under this input, we swing up the pendulum manually by giving an initial velocity when angle  $|\alpha| \geq \frac{\pi}{2}$ . The above open-loop input  $V_m$  cannot stabilize the pendulum to stay around the upright position. For each swing, the pendulum angle  $\alpha$  might stay in the range of  $|\alpha| \leq \frac{\pi}{3}$  for less than one second and then fall. We choose the input in (53) as an example to collect training data and indeed, some other forms of input voltage are also used.

Control input  $V_m$  and motion data are recorded when  $|\alpha| \leq \frac{\pi}{3}$ . The joint angles  $\theta$  and  $\alpha$  are measured with encoders. Their velocities and accelerations are obtained by numerically differentiation of the filtered joint angles. The open-loop controller and data collection are implemented at a frequency of 100 Hz. Figure 5(a) shows an example of collected  $\theta$  and  $\alpha$  angles under the open-loop input (53). It is clear that the pendulum does not achieve balance under (53). Multiple trials of manual swing are applied to the pendulum to collect enough data for  $|\alpha| \leq \frac{\pi}{3}$ . The controller is implemented through Matlab Real-Time Workshop. The MPC is implemented with a period of 0.02 s, that is,  $\Delta t = 0.02$  s, and preview horizon is  $H = 27$ . In implementation, the weight matrices in (31) are chosen as  $Q_1 = Q_3 = \text{diag}\{1000, 100\}$ ,  $Q_2 = \text{diag}\{100, 100\}$ ,  $R = 10I_2$  and  $\nu = 1$ .

A set of 800 points are collected and used as the training data. In testing and validation experiments, the desired external trajectory was designed as  $\theta_d = 0.6 \sin(t) + 0.4 \sin(4t)$  rad. We chose this smooth curve as a representative profile to demonstrate the performance. Figure 6(a) shows the tracking results of the external subsystem base angle  $\theta$  and Fig. 6(b) for the internal subsystem roll angle  $\alpha$ . For comparison purpose, the physical model-based EIC control performance [2] is implemented and included in the figure. The EIC-based control is used as the benchmark and other physical model-based control designs (e.g., sliding mode control [5], orbital stabilization [4], [6], etc.) produce similar performance. The parameter values of the physical model are obtained from the vendor's manual and also validated by experimental tests. Figures 6(c) and 6(d) compare the tracking errors  $e_\theta$  and  $e_\alpha$  under these two controllers. Figures 7(a) and 7(b) further shows the error mean

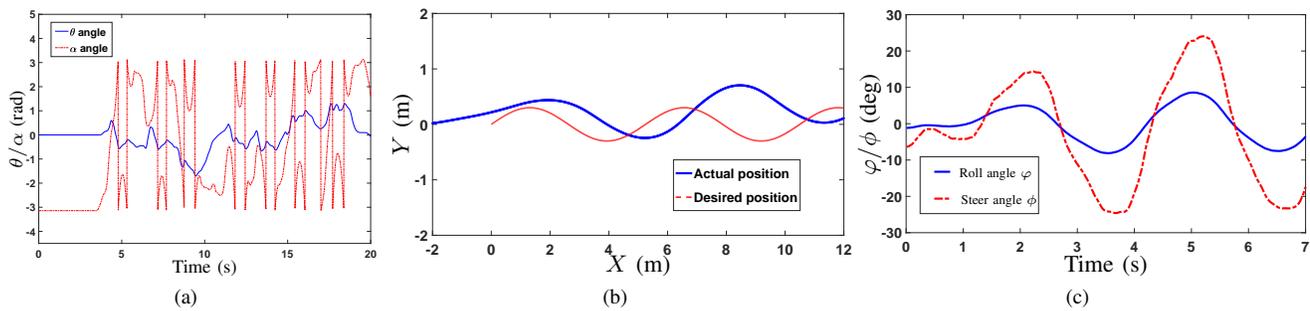


Fig. 5. Example profiles of the collected training data for (a) rotary pendulum experiments (under open-loop control) and for the bikebot experiments (under baseline EIC controller): (b) Bikebot position and (c) bikebot roll and steer angle.

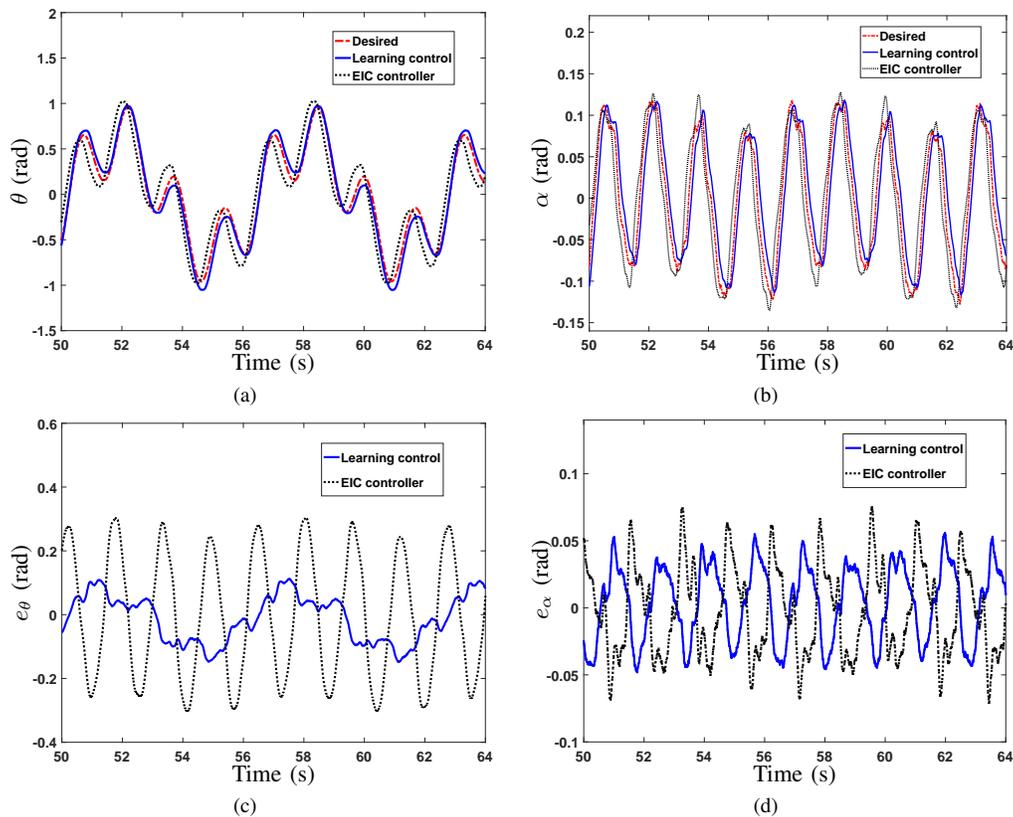


Fig. 6. Tracking errors for one experimental run under the learning-based and the EIC-based controllers for the rotary inverted pendulum. (a) External angle  $\theta$  tracking profiles. (b) Internal angle  $\alpha$  tracking profiles. (c) External angle tracking errors  $e_\theta$ . (d) Internal angle tracking errors  $e_\alpha$ .

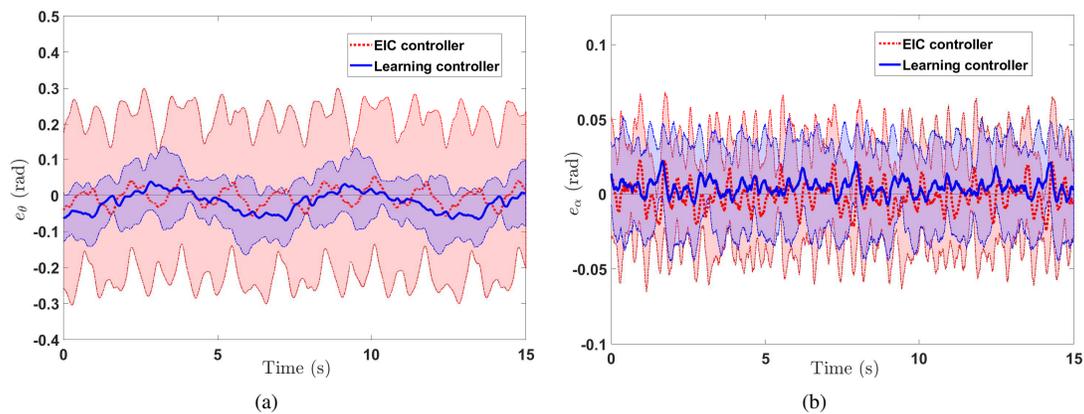


Fig. 7. Tracking errors  $e_\theta$  and  $e_\alpha$  by multiple experimental runs under the learning-based and EIC-based controllers for the rotary inverted pendulum. Mean error and standard deviation profiles for (a)  $e_\theta$  and for (b)  $e_\alpha$ .

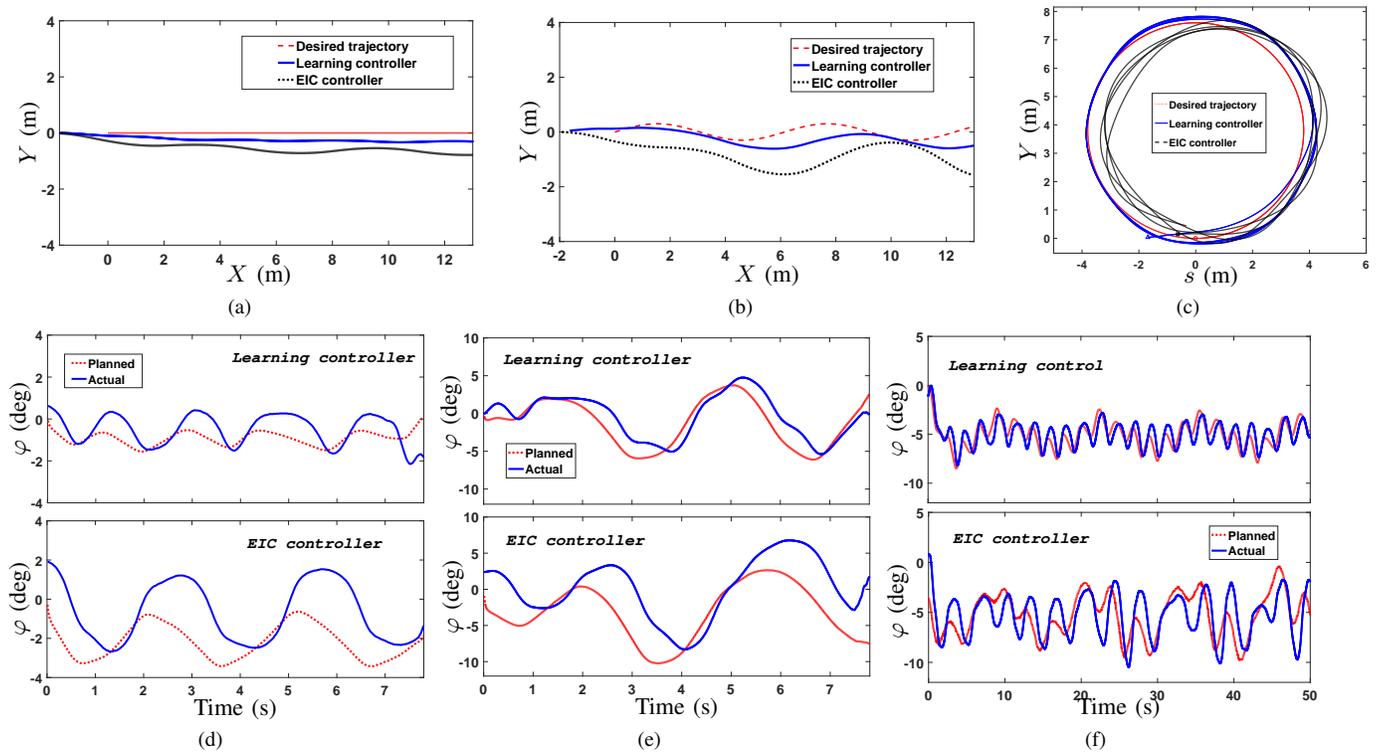


Fig. 8. Performance comparison of the bikebot tracking under the learning-based control and the EIC-based control designs for one experimental run. (a)-(c) for  $X$ - $Y$  position tracking profiles and (d)-(f) for roll angle profiles for straight-line, sinusoidal and circular trajectories, respectively.

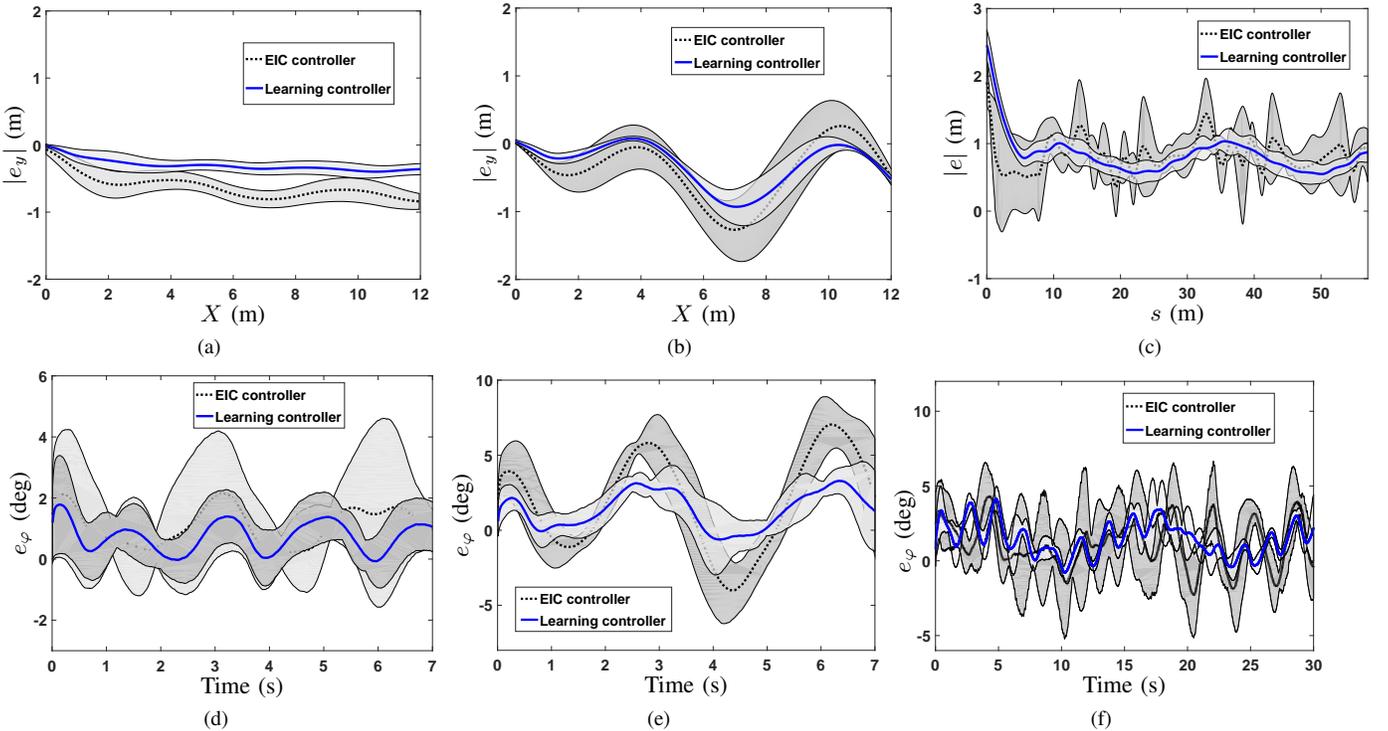


Fig. 9. The bikebot position and roll angle tracking error profiles with multiple experimental runs under the learning-based and the EIC-based controllers. (a)-(c) for  $Y$  position tracking error profiles and (d)-(f) for roll angle error profiles for straight-line, sinusoidal and circular trajectories, respectively.

and standard deviation profiles over multiple experimental runs. Table I lists the comparison of the root mean square (RMS) errors and their deviations under these two controllers. It is clear from these results that the learning-based control

design effectively captures the underactuated balance robotic dynamics and both the external tracking and internal balancing tasks are satisfactory. The performance under the learning-based design outperforms that with the physical model-based

controller with more than 50% reduction in mean values of errors and variances.

TABLE I

ROOT MEAN SQUARE (RMS) ERRORS AND THEIR STANDARD DEVIATIONS OF THE BASE ANGLE (DEG) AND ROLL ANGLE (DEG) COMPARISON UNDER TWO CONTROLLERS FOR THE ROTARY PENDULUM.

	EIC control		Learning control	
	$\theta$	$\alpha$	$\theta$	$\alpha$
RMSE	$19.5 \pm 12.0$	$3.3 \pm 2.0$	$7.5 \pm 4.8$	$1.6 \pm 0.2$

2) *Bikebot experiments*: The bikebot system has high DOFs and sophisticated sensing and actuation components. Because the falling experiments would severely damage the hardware platform, for training data collection, the bikebot is controlled to track sinusoidal-shape trajectories under the EIC-based baseline controller. Different sinusoidal-shape trajectories are designed as  $X_d = v_d t$ ,  $Y_d = A_y \sin\left(\frac{2\pi}{T_y} t\right)$ , where  $v_d = 2$  m/s is the  $x$ -direction desired velocity,  $A_y$  is the magnitude around the  $y$ -direction and  $T_y = 3.5$  s. The training data are collected by 7 different experiment trails and each of them lasts 7 s. In these experiments,  $A_y$  values are chosen from 0.2 m to 0.5 m and the use of these different trajectories aims to perturb the bikebot dynamics. Figures 5(b) and 5(c) show one trial of bikebot training data under the EIC-based controller.

Using the trained model, we conduct the learning model-based control experiments to track various trajectories such as straight-lines, sinusoidal-shape (0.8 m peak-to-peak amplitude), and circular (around 3.8 m radius) trajectories. For comparison purpose, we also conduct experiments and include the results under the physical model-based EIC controller. Figure 8 shows the comparison results under the learning-based and EIC controllers for one experimental run. It is clear that the trajectory tracking results under the learning-based control outperform these under the physical model-based EIC controller (Fig. 8(a)-8(c)). Similarly, the results shown in Fig. 8(d)-8(f) also demonstrate that the roll angles under the learning-based control oscillate less significantly than those under the EIC controller. The learning-based controller also demonstrates quicker reaction in circular tracking than the EIC controller.

Figure 9 further shows the planar bikebot tracking errors and roll angle errors under the learning-based and EIC controllers. In the figure, we plot the trajectory and roll angle tracking errors and their deviations by using five experimental trials. Figures 9(a)-9(c) show the error and deviation profiles for straight-line, sinusoidal and circular trajectories, respectively. The root mean square errors (RMSE) in the  $Y$ -direction are listed in Table II for both the learning-based and EIC-based controllers. It is clearly seen from these results that the learning-based control outperforms the EIC control. Figures 9(d)-9(f) show the roll angle tracking errors for three types of trajectories in multiple runs. The roll angle error magnitudes and variances under the learning-based controller are much smaller than these under the EIC controller and therefore, the learning controller results in agile and smooth tracking behaviors. In Table II, we also list the RMSEs for the roll angles during these runs and these calculations confirm

small variations under the learning control as shown in the figures.

TABLE II

ROOT MEAN SQUARE ERRORS (RMSEs) AND THEIR STANDARD DEVIATIONS OF THE TRACKING POSITION (M) AND ROLL ANGLE (DEG) COMPARISON UNDER TWO CONTROLLERS FOR THE BIKEBOT.

Trajectories	EIC control		Learning control	
	Posit.	Roll	Posit.	Roll
Straight-line	$0.6 \pm 0.2$	$2.3 \pm 1.2$	$0.3 \pm 0.1$	$0.9 \pm 0.5$
Sinusoidal	$0.9 \pm 0.5$	$4.1 \pm 2.3$	$0.4 \pm 0.3$	$2.0 \pm 1.1$
Circular	$0.9 \pm 0.3$	$2.5 \pm 1.5$	$0.7 \pm 0.2$	$1.7 \pm 1.1$

To understand the influence of training data on control performance, we first vary the sizes of the training data sets from 200 to 800 points to obtain different learned models in pendulum platform experiments. These models are used to track the same trajectory  $\theta_d(t)$  as those in the experiments. Figure 10(a) shows the error distribution contours under different sizes of training data sets for the learning control and the EIC control. For each learned model, the plot includes the tracking errors of a 90-sec motion duration. The results clearly imply that with only 200 training data, the controller barely achieves the balancing and tracking tasks with large errors. With the increased training data points, the magnitudes of both the balancing and tracking errors decrease. With a set of 800 training data points, the learned model-based controller achieves superior performance than that under the analytical model-based controller. Theorem 1 reveals that the error trajectory finally falls into a bounded regions and the plots in Fig. 10(a) demonstrate this error analysis.

The trade-off between the tracking and balancing performance is tuned by the choice of  $\nu$  value in the MPC objective function (35). Experiments are conducted to demonstrate the performance with the same learned model under different values of  $\nu$  using the rotary inverted pendulum. The learned model is obtained by using 200 training data points. We intentionally chose a slightly inaccurate learned model and the value of  $\|\Sigma_d\|$  in (35) is relatively large. Figure 10(b) shows the contours of the tracking and balancing errors with different  $\nu$  values. These contours are plotted as the smallest convex cover of the corresponding error data points. When  $\nu = 0$ , the system shows large error distributions due to the poor inverse dynamics model. With  $\nu = 10$ , the system achieves a good trade-off between balancing and tracking tasks. But with a further increased  $\nu$  value (i.e.,  $\nu = 40, 60$ ), the tracking performance becomes similar or slightly worse than those with  $\nu = 10$ , and when  $\nu > 80$  the controller even fails to balance the pendulum. The average variances of the inverse dynamics model for 60-second trials are 0.255, 0.174, 0.108 and 0.108 for  $\nu = 0, 10, 40, 60$ , respectively. The results clearly show that with increased  $\nu$  values, the magnitude of  $\Sigma_d$  decreases. This confirms that the integration of  $\|\Sigma_d\|$  in the objective function helps improve the control performance.

### C. Discussions

The accuracy of the learned models depends on the quality of the training data. We briefly conduct data quality analysis

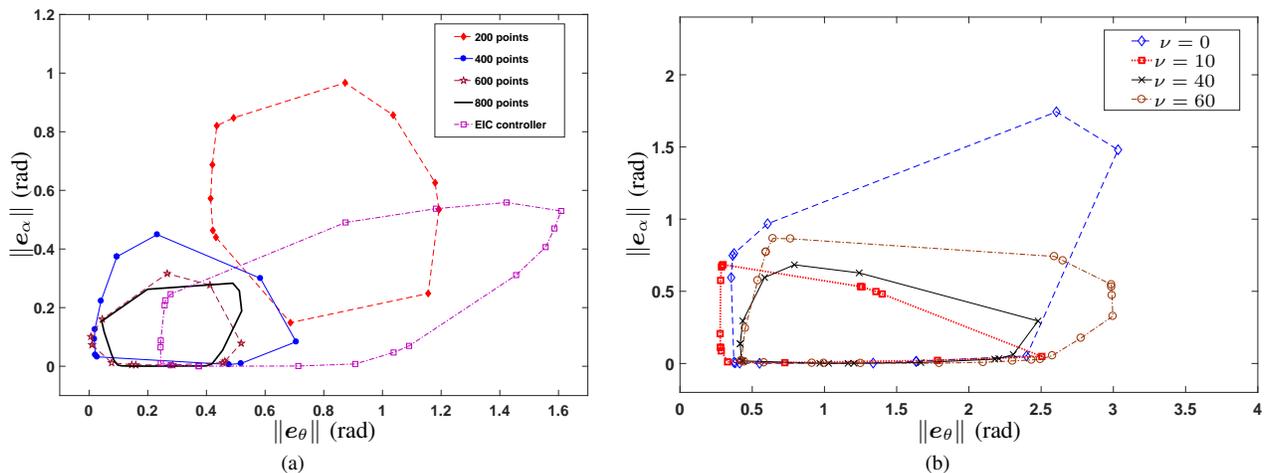


Fig. 10. (a) Comparison results of the internal balance error  $\|e_\alpha\|$  and external tracking error  $\|e_\theta\|$  under the learning-based control by various training data points and the EIC-based control for the rotary inverted pendulum. (b) Comparison of the balance and tracking errors  $\|e_\theta\|$  and  $\|e_\alpha\|$  under different values of the weight factor  $\nu$ .

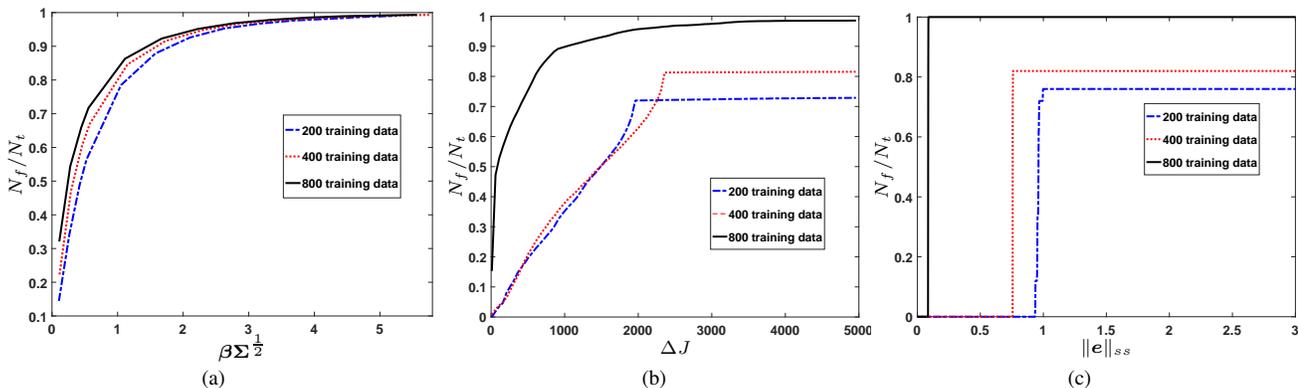


Fig. 11. (a) An approximated probability (computed as  $N_f(\beta\Sigma^{\frac{1}{2}})/N_t$ ) of the learned model prediction accuracy from  $\kappa_\alpha$  as function of training data variance. (b) An approximated probability (computed as  $N_f(\Delta J)/N_t$ ) as function of the estimated MPC cost function error. (c) An approximated probability (computed as  $N_f(\|e\|_{ss})/N_t$ ) as function of steady-state error.

for control performance. We take and validate the performance analysis using rotary inverted pendulum control simulation. First, we evaluate the learned model prediction accuracy that is given in Lemma A.2 about estimate bounds. The training data is collected from the simulation of the ground truth dynamics with additive white noise. The learned model is built on the training data without any knowledge of the true model. The prediction accuracy could be quantified by the mean square errors between the outputs from the learned model and the true model. The prediction accuracy is quantified by the probability frequency of the output differences that fall in the error bound  $\beta\Sigma^{\frac{1}{2}}$  as shown in Lemma A.2. To conduct such simulation experiment, the learned model is tested on  $N_t = 10,000$  independently randomly sampled testing data. The testing data is sampled from a Gaussian distribution whose variance is larger than that of the training data. For each value of the error bound  $\beta\Sigma^{\frac{1}{2}}$ , we count frequency  $N_f(\beta\Sigma^{\frac{1}{2}})$ , i.e., the times that the difference between the outputs from the learned model and the true model falls in that error bound, and compute frequency ratio  $\frac{N_f(\beta\Sigma^{\frac{1}{2}})}{N_t}$  as an approximation of the probability measure, that is,  $\text{Prob} \approx \frac{N_f(\beta\Sigma^{\frac{1}{2}})}{N_t}$ . Figure 11(a) shows the frequency

ratio (namely, probability) versus the experimental cumulative distribution of the error bound  $\beta\Sigma^{\frac{1}{2}}$ . As the bound  $\beta\Sigma^{\frac{1}{2}}$  becomes larger, the frequency ratio  $\frac{N_f}{N_t}$  converges to one. As shown in Fig. 11(a), the converging speed of  $\frac{N_f(\beta\Sigma^{\frac{1}{2}})}{N_t}$  to 1 becomes faster when the number of training data increases.

In MPC computation, the prediction error between the learned model and the true model is cumulated over the prediction horizon. The difference between the predicted trajectory  $\hat{\theta}(t)$  from the learned model  $gp_\theta$  and the actual trajectory  $\theta(t)$  from  $f_\theta$  can be quantified by their cost function difference under the same input trajectory, that is,  $\Delta J^k = \bar{J}_{\hat{\theta}, \hat{W}^*}^k - \bar{J}_{\theta, \hat{W}^*}^k$ . In the simulation experiment, MPC is applied on  $N_t$  values independently sampled from the Gaussian distribution. Similar to the above case, ratio  $\frac{N_f(\Delta J)}{N_t}$  denotes the frequency (i.e., probability) that the cost function error is smaller than  $\Delta J$ . Figure 11(b) shows the experimental cumulative distribution of  $\Delta J$ .

Finally, the proposed controller is tested on  $N_t$  trials whose initial conditions are independently sampled from the Gaussian distribution. The steady-state error  $\|e\|_{ss}$  for each trial is collected to quantify the control performance. For each value

of  $\|e\|_{ss}$ , the frequency of trials that end up with a steady-state error smaller than that value is counted as  $\frac{N_f(\|e\|_{ss})}{N_t}$ . Figure 11(c) shows the chosen initial condition frequency as the distribution of steady-state error magnitude  $\|e\|_{ss}$ . For example, the model trained from a data set of 400 points can drive the steady-state error below 0.76 for about 82% of the initial conditions. However, the system diverges for the other 18% initial conditions due to the inaccuracy of the learned model prediction. The simulation results demonstrate the probabilistic behavior of the proposed learning-based controller. In practical sense, a learned model trained with 800 training data could stabilize the system for most of the situations.

## VII. CONCLUSION AND FUTURE WORK

This paper proposed a learning model-based control framework for underactuated balance robots. One characteristic of underactuated balance robots is that the equilibra of the internal subsystem depends on and varies according to the external subsystem trajectory tracking. The control design consisted an integrated trajectory tracking of the external subsystem and stabilization of the internal subsystem. The trajectory tracking of the external subsystem was designed through an MPC approach, while an inverse dynamics controller was used to simultaneously stabilize the planned internal subsystem trajectory. The GPs models were used to estimate the system dynamics and provide predictive distribution of model uncertainties. The control design explicitly incorporated prediction variances with tracking and stabilization performance through online optimization. The learned GP models were obtained without need of prior knowledge about the robotic systems dynamics nor successful balance demonstration. Moreover, the stability and closed-loop control performance were guaranteed through comprehensive closed-loop control systems analysis. We demonstrated the control systems design independently using two underactuated balance robotic platforms: a rotary inverted pendulum and a bikebot.

We are currently working on testing the bikebot system on various terrain conditions to explore the performance under complex, dynamic environments. Real-time machine learning techniques are currently designed and developed on dedicated hardware to improve the control performance. Finally, quantitative analysis of training data quality is also among the future research directions.

## ACKNOWLEDGMENT

The authors would like to thank Dr. Pengcheng Wang for valuable discussions on theoretical analysis and his help on bikebot experiments. The authors are also grateful to Yongbin Gong of Rutgers University for his implementation help of the vision-based localization system for bikebot experiments.

## APPENDIX

### A. Gaussian Process

A Gaussian process (GP) is a collection of random variables, any finite number of which have a joint Gaussian distribution. A real value process  $f(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$  is determined by its mean value function  $\mu(\mathbf{x})$  and covariance function  $k(\mathbf{x}, \mathbf{x}')$

as  $\mu(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})]$  and  $k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - \mu(\mathbf{x}))(f(\mathbf{x}') - \mu(\mathbf{x}'))]$ . Suppose the training data set contains  $N$  input output data pairs  $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^N$ . The observed output  $y_i$  is a noisy observation of the underlying function value with zero mean Gaussian noise  $\varepsilon$ , i.e.,  $y_i = f(\mathbf{x}_i) + \varepsilon$  with  $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ . The observation vector is denoted as  $\mathbf{y} = [y_1 \cdots y_N]^T$  and the input design matrix is denoted as  $\mathbf{X} = [\mathbf{x}_1^T \cdots \mathbf{x}_N^T]^T$ . At a testing point  $\mathbf{x}^* \in \mathbb{R}^n$ , the function value  $f^*$  is predicted by the observed training data  $\mathcal{D}$ . The joint distribution of  $\mathbf{y}$  and the testing output  $f^*$  is

$$\begin{bmatrix} \mathbf{y} \\ f^* \end{bmatrix} \sim \mathcal{N} \left( \mathbf{0}, \begin{bmatrix} \mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I}_N & \mathbf{k}(\mathbf{X}, \mathbf{x}^*) \\ \mathbf{k}(\mathbf{X}, \mathbf{x}^*)^T & k(\mathbf{x}^*, \mathbf{x}^*) \end{bmatrix} \right),$$

where  $\mathbf{K}(\mathbf{X}, \mathbf{X})$  is an  $N \times N$  kernel matrix whose element is  $\mathbf{K}_{i,j}(\mathbf{X}, \mathbf{X}) = k(\mathbf{x}^i, \mathbf{x}^j)$ .  $\mathbf{k}(\mathbf{X}, \mathbf{x}^*)$  is an  $N \times 1$  column vector whose element is  $\mathbf{k}_i(\mathbf{X}, \mathbf{x}^*) = k(\mathbf{x}^i, \mathbf{x}^*)$ .

The probabilistic prediction of  $f^*$  is given by the conditional distribution

$$f^* | \mathbf{x}^*, \mathcal{D} \sim \mathcal{N}(\mu(\mathbf{x}^*), \Sigma(\mathbf{x}^*))$$

where  $\mathcal{N}(\cdot, \cdot)$  represents a normal distribution,  $\mu(\mathbf{x}^*)$  and  $\Sigma(\mathbf{x}^*)$  are the posterior mean and covariance functions as

$$\begin{aligned} \mu(\mathbf{x}^*) &= \mathbf{k}(\mathbf{X}, \mathbf{x}^*)^T [\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I}_N]^{-1} \mathbf{y}, \\ \Sigma(\mathbf{x}^*) &= k(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{k}(\mathbf{X}, \mathbf{x}^*)^T [\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I}_N]^{-1} \mathbf{k}(\mathbf{X}, \mathbf{x}^*). \end{aligned} \quad (54)$$

GPs can also be applied to learn  $n$ -dimensional vector-valued function  $\mathbf{f}(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}^n$ . In such cases, GPs are adopted to learn each function  $f_i(\mathbf{x}), i = 1, \dots, n$ , as  $f_i^* | \mathbf{x}^*, \mathcal{D} \sim \mathcal{N}(\mu_i(\mathbf{x}^*), \Sigma_i(\mathbf{x}^*))$  independently. The predictive distribution is written as

$$\mathbf{f}^* | \mathbf{x}^*, \mathcal{D} \sim \mathcal{N}(\boldsymbol{\mu}(\mathbf{x}^*), \boldsymbol{\Sigma}(\mathbf{x}^*)), \quad (55)$$

where  $\boldsymbol{\mu}(\mathbf{x}^*) = [\mu_1(\mathbf{x}^*) \cdots \mu_n(\mathbf{x}^*)]^T$  and  $\boldsymbol{\Sigma}(\mathbf{x}^*) = \text{diag}\{\Sigma_1(\mathbf{x}^*), \dots, \Sigma_n(\mathbf{x}^*)\}$ .

### B. GP-based estimation error bounds

The Gaussian process is determined by the covariance function (also called kernel function), which corresponds to a set of basis feature function in regression problem. The choice of the covariance function form depends on the input data and the commonly used covariance function is the squared exponential (SE) function as

$$k(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 \exp \left( -\frac{1}{2} \Delta \mathbf{x}_{ij}^T \mathbf{W} \Delta \mathbf{x}_{ij} \right) + \sigma^2 \delta_{ij}, \quad (56)$$

where  $\Delta \mathbf{x}_{ij} = \mathbf{x}_i - \mathbf{x}_j$ ,  $\mathbf{W}$  is a positive definite weighting matrix,  $\sigma_f^2$  and  $\sigma^2$  are hyper-parameters,  $\delta_{ij} = 1$  if  $i = j$ ; otherwise  $\delta_{ij} = 0$ . The values of the above SE covariance function only depend on the distance between two points  $\|\Delta \mathbf{x}_{ij}\|$ . The following result gives the upper bound of covariance of the SE kernel.

*Lemma A.1:* For any testing point  $\mathbf{x}^* \in \mathbb{R}^n$ , the posterior covariance  $\Sigma(\mathbf{x}^*)$  of the SE kernel is bounded by  $\Sigma(\mathbf{x}^*) \leq \sigma_f^2 + \sigma^2$ , where  $\sigma_f$  and  $\sigma$  are the hyper-parameters in (56). For  $n$ -dimensional function  $\mathbf{f}$  in (55),  $\|\Sigma(\mathbf{x}^*)\| \leq$

$\max_{1 \leq i \leq n} (\sigma_{f_i}^2 + \sigma_i^2)$ , where  $\sigma_{f_i}$  and  $\sigma_i$  are the hyper-parameters for corresponding  $f_i$ .

*Proof:* From (54) and (56), noting the positive definiteness of  $\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I}_N$ , we have  $\Sigma(\mathbf{x}^*) \leq k(\mathbf{x}^*, \mathbf{x}^*) \leq \sigma_f^2 + \sigma^2$ . Since  $\Sigma(\mathbf{x}^*) = \text{diag}\{\Sigma_1(\mathbf{x}^*), \dots, \Sigma_n(\mathbf{x}^*)\}$ , by the definition of matrix norm  $\|\Sigma(\mathbf{x}^*)\| = \lambda_{\max}(\Sigma(\mathbf{x}^*)) \leq \max_{1 \leq i \leq n} (\sigma_{f_i}^2 + \sigma_i^2)$ . This proves the lemma.  $\blacksquare$

For a testing point  $\mathbf{x}$ , the predictive distribution conditioned on observations  $\mathcal{D}$  is  $\mathcal{N}(\mu(\mathbf{x}), \Sigma(\mathbf{x}))$ . The following lemma gives the learning error bound.

*Lemma A.2* ([44, Theorem 6]): Let  $\delta \in (0, 1)$ , then

$$\Pr\{\|\mu(\mathbf{x}) - f(\mathbf{x})\| \leq \beta \Sigma^{\frac{1}{2}}(\mathbf{x})\} \geq 1 - \delta$$

with  $\beta = \sqrt{2\|f\|_k^2 + 300\gamma \ln^3(\frac{N+1}{\delta})}$ ,  $\gamma \in \mathbb{R}$  is the maximum information gain defined as  $\gamma = \max_{\mathbf{X}} I_g(\mathbf{y}; f)$ .

The information gain in the above lemma is defined as  $I_g(\mathbf{y}; f) = H(\mathbf{y}) - H(\mathbf{y}|f)$ , where  $H(\cdot)$  is the entropy function. In GP context, the prior distribution  $\mathbf{y} \sim \mathcal{N}(\mathbf{0}, \mathbf{K} + \sigma^2 \mathbf{I}_N)$  and the conditional distribution  $\mathbf{y}|f \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_N)$ , the entropies are  $H(\mathbf{y}) = \frac{1}{2} \log\{\det[2\pi e(\mathbf{K} + \sigma^2 \mathbf{I}_N)]\}$  and  $H(\mathbf{y}|f) = \frac{1}{2} \log(2\pi e \sigma^2)$ . Therefore, the information gain is  $I_g(\mathbf{y}; f) = \frac{1}{2} \log \det(\mathbf{I}_N + \sigma^{-2} \mathbf{K})$ . According to [44], the maximum information gain  $\gamma$  for the SE kernel is in the order of  $O((\ln(N))^{n+1})$ . For  $n$ -dimensional vector function  $\mathbf{f}(\mathbf{x})$ , if every dimension is independent of each other, the results in Lemma A.2 are extended to the following lemma.

*Lemma A.3* ([26, Lemma 1]): Let  $\delta \in (0, 1)$ , then

$$\Pr\{\|\boldsymbol{\mu}(\mathbf{x}) - \mathbf{f}(\mathbf{x})\| \leq \|\boldsymbol{\beta}^T \Sigma^{\frac{1}{2}}(\mathbf{x})\|\} \geq (1 - \delta)^n,$$

where  $\boldsymbol{\mu}(\cdot)$  and  $\Sigma(\cdot)$  are defined in (55), vector  $\boldsymbol{\beta} \in \mathbb{R}^n$  and its  $i$ th element  $\beta_i = \sqrt{2\|f_i\|_k^2 + 300\gamma_i \ln^3(\frac{N+1}{\delta})}$ , and  $\gamma_i$  is the maximum information gain for  $f_i$ .

### C. Proof of Lemma 1

We calculate  $\boldsymbol{\mu}_\alpha - \boldsymbol{\kappa}_\alpha(\dot{\boldsymbol{\alpha}}_2)$  by Taylor expansion as

$$\begin{aligned} \boldsymbol{\mu}_\alpha - \boldsymbol{\kappa}_\alpha(\dot{\boldsymbol{\alpha}}_2) &= \boldsymbol{\mu}_\alpha - \boldsymbol{\kappa}_\alpha(\mathbf{v}) + \boldsymbol{\kappa}_\alpha(\mathbf{v}) - \boldsymbol{\kappa}_\alpha(\dot{\boldsymbol{\alpha}}_2) \\ &= \boldsymbol{\mu}_\alpha - \boldsymbol{\kappa}_\alpha(\mathbf{v}) - \frac{\partial \boldsymbol{\kappa}_\alpha}{\partial \mathbf{v}}(\dot{\boldsymbol{\alpha}}_2 - \mathbf{v}) + O(\|\dot{\boldsymbol{\alpha}}_2 - \mathbf{v}\|^2) \\ &= \boldsymbol{\mu}_\alpha - \boldsymbol{\kappa}_\alpha(\mathbf{v}) - \frac{\partial \boldsymbol{\kappa}_\alpha}{\partial \mathbf{v}}[\boldsymbol{\mu}_\alpha - \boldsymbol{\kappa}_\alpha(\dot{\boldsymbol{\alpha}}_2)] + O(\|\dot{\boldsymbol{\alpha}}_2 - \mathbf{v}\|^2). \end{aligned}$$

The third equality results from (17). Note that  $O(\|\dot{\boldsymbol{\alpha}}_2 - \mathbf{v}\|^2) \leq c_2\|e_\alpha\|^2 + c_1\|e_\alpha\| + c_0$ , with constants  $c_i > 0$ ,  $i = 0, 1, 2$ .

Note that  $\mathbf{A}_\kappa = \mathbf{I} + \frac{\partial \boldsymbol{\kappa}_\alpha}{\partial \mathbf{v}}$  is the linearization of the left-hand side of the second equation (i.e.,  $\boldsymbol{\alpha}_2$  subdynamics) of (13) with respect to  $\dot{\boldsymbol{\alpha}}_2$ . It serves like the inertia matrix of  $\boldsymbol{\alpha}_2$ -dynamics and therefore,  $\mathbf{A}_\kappa$  is non-singular and also positive definite. From the above equation and  $\dot{\boldsymbol{\alpha}}_2$  in (13), we have

$$\boldsymbol{\mu}_\alpha - \boldsymbol{\kappa}_\alpha(\dot{\boldsymbol{\alpha}}_2) = \mathbf{A}_\kappa^{-1}[O(\|\dot{\boldsymbol{\alpha}}_2 - \mathbf{v}\|^2) + \boldsymbol{\mu}_\alpha - \boldsymbol{\kappa}_\alpha(\mathbf{v})].$$

Taking the norm and applying the results in Lemma A.3 to  $\boldsymbol{\mu}_\alpha - \boldsymbol{\kappa}_\alpha(\mathbf{v})$ , we have

$$\Pr\left\{\boldsymbol{\Pi} = \left\{\|\boldsymbol{\mu}_\alpha - \boldsymbol{\kappa}_\alpha(\dot{\boldsymbol{\alpha}}_2)\| \leq \lambda_{\min}^{-1}(\mathbf{A}_\kappa) \left(\sum_{i=0}^2 c_i \|e_\alpha\|^i + \|\boldsymbol{\beta}_\alpha^T \Sigma_\alpha^{\frac{1}{2}}\|\right)\right\}\right\} \geq (1 - \delta)^n,$$

where  $\lambda_{\min}(\mathbf{A}_\kappa) > 0$  is the smallest eigenvalue of  $\mathbf{A}_\kappa$ . Defining

$$\rho(e_\alpha, \boldsymbol{\theta}) = \lambda_{\min}^{-1}(\mathbf{A}_\kappa) \left(\sum_{i=0}^2 c_i \|e_\alpha\|^i + \|\boldsymbol{\beta}_\alpha^T \Sigma_\alpha^{\frac{1}{2}}\|\right), \quad (57)$$

we prove the lemma.

### D. Proof of Lemma 2

Defining  $\lambda_1 = -\frac{1}{2}(k_d - \sqrt{k_d^2 - 4k_p}) < 0$  and  $\lambda_2 = -\frac{1}{2}(k_d + \sqrt{k_d^2 - 4k_p}) < 0$ , we first show that  $\mathbf{A}$  is diagonalizable with  $n$ -eigenvalue as  $\frac{\lambda_1}{\epsilon}$  and the other  $n$ -eigenvalue as  $\frac{\lambda_2}{\epsilon}$ . To see that, introducing nonsingular matrix  $\mathbf{M}$  and diagonal matrix  $\boldsymbol{\Lambda}$  as

$$\mathbf{M} = \begin{bmatrix} \epsilon \mathbf{I}_n & \epsilon \mathbf{I}_n \\ \lambda_1 \mathbf{I}_n & \lambda_2 \mathbf{I}_n \end{bmatrix}, \quad \boldsymbol{\Lambda} = \begin{bmatrix} \frac{\lambda_1}{\epsilon} \mathbf{I}_n & \mathbf{0} \\ \mathbf{0} & \frac{\lambda_2}{\epsilon} \mathbf{I}_n \end{bmatrix}, \quad (58)$$

it is straightforward to verify that  $\mathbf{A} = \mathbf{M} \boldsymbol{\Lambda} \mathbf{M}^{-1}$ . To assess the convergence property of  $e_\alpha$ , we introduce  $e_\alpha = \mathbf{M} e_{\alpha'}$  and error dynamics (18) becomes

$$\dot{e}_{\alpha'} = \boldsymbol{\Lambda} e_{\alpha'} + \mathbf{M}^{-1} \mathbf{B}[\mathbf{r}(t) + \boldsymbol{\mu}_\alpha - \boldsymbol{\kappa}_\alpha(\dot{\boldsymbol{\alpha}}_2)]. \quad (59)$$

Since  $\boldsymbol{\Lambda}$  is Hurwitz, there exists a positive definite matrix  $\mathbf{P}_\alpha$  such that  $\boldsymbol{\Lambda}^T \mathbf{P}_\alpha + \mathbf{P}_\alpha \boldsymbol{\Lambda} = -\mathbf{I}_{2n}$ . We choose the Lyapunov function candidate  $V_{\alpha'} = e_{\alpha'}^T \mathbf{P}_\alpha e_{\alpha'}$  for (59) and then

$$\dot{V}_{\alpha'} = -e_{\alpha'}^T e_{\alpha'} + 2(\mathbf{B}^T \mathbf{M}^{-T} \mathbf{P}_\alpha e_{\alpha'})^T [\mathbf{r} + \boldsymbol{\mu}_\alpha - \boldsymbol{\kappa}_\alpha(\dot{\boldsymbol{\alpha}}_2)].$$

Auxiliary control  $\mathbf{r}(t)$  is designed as  $\mathbf{r}(t) = -\rho(e_\alpha, \boldsymbol{\theta}) \frac{\mathbf{B}^T \mathbf{M}^{-T} \mathbf{P}_\alpha e_{\alpha'}}{\|\mathbf{B}^T \mathbf{M}^{-T} \mathbf{P}_\alpha e_{\alpha'}\|}$  if  $\|\mathbf{B}^T \mathbf{M}^{-T} \mathbf{P}_\alpha e_{\alpha'}\| > \xi$ ;  $\mathbf{r}(t) = -\frac{\rho(e_\alpha, \boldsymbol{\theta})}{\xi} \mathbf{B}^T \mathbf{M}^{-T} \mathbf{P}_\alpha e_{\alpha'}$  if  $\|\mathbf{B}^T \mathbf{M}^{-T} \mathbf{P}_\alpha e_{\alpha'}\| \leq \xi$  for constant  $\xi > 0$  and  $\rho(e_\alpha, \boldsymbol{\theta})$  is defined by (57).

According to Lemma 2, with the above design and choosing  $\xi = \frac{\lambda_{\min}(\mathbf{A}_\kappa)}{c_2 \|\mathbf{M}\|^2}$ , we obtain

$$\begin{aligned} \dot{V}_{\alpha'} &\leq -\|e_{\alpha'}\|^2 + \frac{\xi \rho(e_\alpha, \boldsymbol{\theta})}{2} = -\frac{1}{2} \|e_{\alpha'}\|^2 + \\ &\quad \frac{c_1}{2c_2 \|\mathbf{M}\|} \|e_{\alpha'}\| + \frac{c_0}{2c_2 \|\mathbf{M}\|^2} + \frac{\|\boldsymbol{\beta}_\alpha^T \Sigma_\alpha^{1/2}\|}{2c_2 \|\mathbf{M}\|^2} \\ &= -\frac{1}{4} \|e_{\alpha'}\|^2 - \frac{1}{4} \left( \|e_{\alpha'}\| - \frac{c_1}{c_2 \|\mathbf{M}\|} \right)^2 + c_3 \\ &\leq -\frac{1}{4} \|e_{\alpha'}\|^2 + c_3, \end{aligned} \quad (60)$$

where  $c_3 = \frac{1}{4} \frac{c_1^2}{c_2^2 \|\mathbf{M}\|^2} + \frac{1}{2} \frac{c_0}{c_2 \|\mathbf{M}\|^2} + \frac{1}{2} \frac{\|\boldsymbol{\beta}_\alpha^T \Sigma_\alpha^{1/2}\|}{2c_2 \|\mathbf{M}\|^2} > 0$ . Since  $\mathbf{P}_\alpha$  is the solution of Lyapunov equation with  $\boldsymbol{\Lambda}$  in (58), it has  $n$ -eigenvalue at  $-\frac{\epsilon}{2\lambda_1} > 0$  and the other  $n$ -eigenvalue at  $-\frac{\epsilon}{2\lambda_2} > 0$ . Thus, we obtain  $V_{\alpha'} \leq -\frac{\epsilon}{2\lambda_1} \|e_{\alpha'}\|^2$ . Using this result, from (60), we obtain  $\dot{V}_{\alpha'} \leq \frac{\lambda_1}{2\epsilon} V_{\alpha'} + c_3$  and therefore,

$$V_{\alpha'}(t) \leq V_{\alpha'}(0) e^{\frac{\lambda_1}{2\epsilon} t} - \frac{2\epsilon}{\lambda_1} c_3. \quad (61)$$

Considering  $V_\alpha(t) = e_\alpha(t)^T \mathbf{P} e_\alpha(t)$  and positive definiteness of  $\mathbf{P} = \mathbf{M}^{-T} \mathbf{P}_\alpha \mathbf{M}^{-1}$ , it is straightforward to check that  $V_\alpha(t) = e_{\alpha'}(t)^T \mathbf{M}^T \mathbf{P} \mathbf{M} e_{\alpha'}(t) = e_{\alpha'}(t)^T \mathbf{P}_\alpha e_{\alpha'}(t) =$

$V_{\alpha'}(t)$ . Defining  $\mathbf{Q} = \mathbf{M}^{-T}\mathbf{M}^{-1}$ ,  $\mathbf{P}$  is the solution of Lyapunov equation  $\mathbf{A}^T\mathbf{P} + \mathbf{P}\mathbf{A} = -\mathbf{Q}$ . Using (60), we obtain

$$\dot{V}_{\alpha'}(t) \leq -\frac{1}{4}e_{\alpha}^T\mathbf{Q}e_{\alpha} + c_3. \quad (62)$$

Using  $\mathbf{P}$  and  $e_{\alpha}$ , we write the control input  $\mathbf{r}(t)$  as in (23). Noting that  $\lambda_{\min}(\mathbf{P})\|e_{\alpha}\|^2 \leq V(t)_{\alpha} \leq \lambda_{\max}(\mathbf{P})\|e_{\alpha}\|^2$ , from (61), we have

$$\begin{aligned} \|e_{\alpha}(t)\| &\leq \sqrt{\frac{\lambda_{\max}(\mathbf{P})}{\lambda_{\min}(\mathbf{P})}}\|e_{\alpha}(0)\|e^{\frac{\lambda_1}{4\epsilon}t} + \sqrt{-\frac{2\epsilon c_3}{\lambda_1\lambda_{\min}(\mathbf{P})}} \\ &= d_1\|e_{\alpha}(0)\|e^{\frac{\lambda_1}{4\epsilon}t} + d_2 \end{aligned}$$

with  $d_1$  and  $d_2$  are given in the lemma.

### E. Proof of Lemma 3

Taking norm on both sides of (30) and applying the upper-bound of the gradient  $\|\frac{\partial \mu_{gp\theta}}{\partial \theta}\| \leq L_1$ , we obtain

$$\begin{aligned} \|\Sigma_{\hat{\theta}}(k+i+1|k)\| &\leq (\|\mathbf{F}\|^2 + \|\mathbf{G}\|^2 L_1^2)\|\Sigma_{\hat{\theta}}(k+i|k)\| \\ &\quad + \|\mathbf{G}\|^2\|\Sigma_{gp\theta}\| \\ &\leq (\|\mathbf{F}\|^2 + \|\mathbf{G}\|^2 L_1^2)\|\Sigma_{\hat{\theta}}(k+i|k)\| \\ &\quad + \|\mathbf{G}\|^2\sigma_{\mathbf{f}}^2. \end{aligned}$$

Applying the above process iteratively with  $\Sigma_{\hat{\theta}}(k|k) = \mathbf{0}$ , we have

$$\|\Sigma_{\hat{\theta}}(k+i|k)\| \leq \frac{1 - (\|\mathbf{F}\|^2 + \|\mathbf{G}\|^2 L_1^2)^i}{1 - (\|\mathbf{F}\|^2 + \|\mathbf{G}\|^2 L_1^2)}\|\mathbf{G}\|^2\sigma_{\mathbf{f}}^2.$$

From (29), the norms of  $\mathbf{F}$  and  $\mathbf{G}$  are calculated as

$$\|\mathbf{F}\| = \sqrt{1 + \frac{\Delta t}{2}(\Delta t + \sqrt{(\Delta t)^2 + 4})}, \quad \|\mathbf{G}\| = \Delta t.$$

For  $\Delta t \ll 1$ , taking approximation  $\|\mathbf{F}\| \approx 1$  and fact that  $(1+x)^n \approx 1+nx$  for  $|x| \ll 1$ , we obtain the upper-bound as shown in the lemma.

### F. Proof of Lemma 4

It is straightforward to obtain

$$\begin{aligned} l_f(k+H+2) &= l_f^*(k+H+2) + \text{tr}(\mathbf{Q}_3\Sigma_{\hat{\theta}}(k+H+2)) \\ &\leq l_f^*(k+H+1) - l_s^*(k+H+1) \\ &\quad + \text{tr}(\mathbf{Q}_3\Sigma_{\hat{\theta}}(k+H+2)) \\ &\leq l_f(k+H+1) - l_s^*(k+H+1) \\ &\quad + \text{tr}(\mathbf{Q}_3\Sigma_{\hat{\theta}}(k+H+2)). \end{aligned}$$

Under conditions (42), we obtain  $l_s(k+H+1) \leq l_s^*(k+H+1) + \text{tr}(\mathbf{Q}_3\Sigma_{\hat{\theta}}(k+H+1))$  and combining with the above inequality, the proof is completed.

### G. Proof of Lemma 5

We first show the decreasing property of  $J_{\hat{\theta}^*, \hat{W}^*}^k$ . Inspired by the approach in [45], we take the technique to construct a following intermediary policy  $\hat{W}^e(k+1)$  extended from  $\hat{W}^*(k)$  as

$$\begin{aligned} \hat{W}^e(k+1) &= \{\hat{\alpha}^e(k+1), \hat{w}^e(k+i+1), \mathbf{u}_f^e(k+i+1), \\ &\quad i = 0, \dots, H\}, \end{aligned}$$

where  $\hat{\alpha}_1^e(k+1) = \hat{\alpha}_1^*(k) + \hat{\alpha}_2^*(k)\Delta t$ ,  $\hat{\alpha}_2^e(k+1) = \hat{\alpha}_2^*(k) + \hat{W}^*(k)\Delta t$ , and  $\hat{w}^e(k+i) = \hat{W}^*(k+i)$ ,  $\mathbf{u}_f^e(k+i) = \mathbf{u}_f^*(k+i)$  for  $i = 1, \dots, H$ , and  $\hat{w}^e(k+H+1)$  and  $\mathbf{u}_f^e(k+H+1)$  satisfy (42). The choice of the above design guarantees that inputs  $\{\hat{\alpha}^e(k+i), \hat{w}^e(k+i), \mathbf{u}_f^e(k+i)\}$  of  $\hat{W}_H^e(k+1)$  are the same as  $\{\hat{\alpha}^*(k+i), \hat{W}^*(k+i), \mathbf{u}_f^*(k+i)\}$  of  $\hat{W}^*(k)$  for  $i = 1, \dots, H$ . Consequently, the predicted states  $\mu_{\hat{\theta}}^e(k+i)$ ,  $\Sigma_{\hat{\theta}}^e(k+i)$  by (30) under  $\hat{W}^e(k+1)$  are the same as these under control  $\hat{W}^*(k)$  at these steps. Let  $l_s^e(k+i)$  ( $l_f^e(k+i)$ ) and  $l_s^*(k+i)$  ( $l_f^*(k+i)$ ) denote the stage and terminal costs under controls  $\hat{W}^e(k+1)$  and  $\hat{W}^*(k)$ , respectively. It is then straightforward to obtain that  $l_s^e(k+i) = l_s^*(k+i)$  for  $i = 1, \dots, H$  and  $l_f^e(k+H+1) = l_f^*(k+H+1)$ . Therefore,

$$\begin{aligned} J_{\hat{\theta}^e, \hat{W}^e}^{k+1} - J_{\hat{\theta}^*, \hat{W}^*}^k &= l_s^e(k+H+1) + l_f^e(k+H+2) \\ &\quad - l_s^*(k) - l_f^*(k+H+1) + \nu\Delta\Sigma_{dk}^{e*} + \Delta\hat{\alpha}_{\mathbf{Q}_2k}^*, \end{aligned}$$

where  $\Delta\Sigma_{dk}^{e*} = \|\Sigma_d^{\hat{W}^e}(k+1)\| - \|\Sigma_d^{\hat{W}^*}(k)\|$  and  $\Delta\hat{\alpha}_{\mathbf{Q}_2k}^* = \|\hat{\alpha}^e(k+1)\|_{\mathbf{Q}_2}^2 - \|\hat{\alpha}^*(k)\|_{\mathbf{Q}_2}^2$ . Noting that  $\hat{w}^e(k+H+1)$  and  $\mathbf{u}_f^e(k+H+1)$  satisfy (42), by Lemma 4, we have

$$\begin{aligned} J_{\hat{\theta}^e, \hat{W}^e}^{k+1} - J_{\hat{\theta}^*, \hat{W}^*}^k &\leq -l_s^*(k) + \nu\Delta\Sigma_{dk}^{e*} + \Delta\hat{\alpha}_{\mathbf{Q}_2k}^* + \\ &\quad \text{tr}(\mathbf{Q}_1\Sigma_{\hat{\theta}}(k+H+1)) + \text{tr}(\mathbf{Q}_3\Sigma_{\hat{\theta}}(k+H+2)). \end{aligned}$$

Because of  $J_{\hat{\theta}^*, \hat{W}^*}^{k+1} \leq J_{\hat{\theta}^e, \hat{W}^e}^{k+1}$ , from the above result, we have

$$\begin{aligned} J_{\hat{\theta}^*, \hat{W}^*}^{k+1} - J_{\hat{\theta}^*, \hat{W}^*}^k &\leq -\lambda_{\min}(\mathbf{Q}_1)\|e_{\theta}(k)\|^2 + \nu\Delta\Sigma_{dk}^{e*} \\ &\quad + \Delta\hat{\alpha}_{\mathbf{Q}_2k}^* + \text{tr}(\mathbf{Q}_1\Sigma_{\hat{\theta}}(k+H+1)) \\ &\quad + \text{tr}(\mathbf{Q}_3\Sigma_{\hat{\theta}}(k+H+2)). \quad (63) \end{aligned}$$

From Lemma 3,  $\Sigma_{\hat{\theta}}(k+H+1) \leq (H+1)(\Delta t)^2\sigma_{\mathbf{f}}^2$ . From Lemma A.1,  $\Delta\Sigma_{dk}^{e*} \leq \|\Sigma_d^{\hat{W}^e}(k+1)\| \leq \max_{1 \leq i \leq n}(\sigma_{\alpha_i}^2 + \sigma_i^2) := \sigma_{\kappa \max}^2$ . Letting  $\|\alpha(k+1)\|_{\mathbf{Q}_2}^2 \leq \alpha_{\max}^2$  as the constant upper-bound, we have  $\Delta\hat{\alpha}_{\mathbf{Q}_2k}^* \leq \|\hat{\alpha}^e(k+1)\|_{\mathbf{Q}_2}^2 \leq \alpha_{\max}^2$  and thus,

$$\begin{aligned} J_{\hat{\theta}^*, \hat{W}^*}^{k+1} - J_{\hat{\theta}^*, \hat{W}^*}^k &\leq -\lambda_{\min}(\mathbf{Q}_1)\|e_{\theta}(k)\|^2 + \nu\sigma_{\kappa \max}^2 \\ &\quad + \alpha_{\max}^2 + m[\lambda_{\max}(\mathbf{Q}_1) + \lambda_{\max}(\mathbf{Q}_3)] \\ &\quad (H+2)(\Delta t)^2\sigma_{\mathbf{f}}^2. \quad (64) \end{aligned}$$

Furthermore, from the definition of  $J_{\hat{\theta}^*, \hat{W}^*}^k$ , we have  $J_{\hat{\theta}^*, \hat{W}^*}^{k+1} \geq \lambda_{\min}(\mathbf{Q}_1)\|e_{\mu_{\hat{\theta}}}(k+1)\|^2$ . By the monotonicity of the value function (Lemma 2.15 in [45]), we have

$$\begin{aligned} J_{\hat{\theta}^*, \hat{W}^*}^k &\leq l_f(k) + \|\hat{\alpha}^*(k)\|_{\mathbf{Q}_2}^2 + \nu\|\Sigma_d^{\hat{W}^*}(k)\| \\ &\leq \lambda_{\max}(\mathbf{Q}_3)\|e_{\theta}(k)\|^2 + \alpha_{\max}^2 + \nu\sigma_{\kappa \max}^2. \end{aligned}$$

Substituting the above inequalities into (64) to cancel  $\|e(k)\|^2$ , we obtain  $J_{\hat{\theta}^*, \hat{W}^*}^{k+1} \leq d_3 J_{\hat{\theta}^*, \hat{W}^*}^k + d_4$  with  $d_3 = 1 - \frac{\lambda_{\min}(\mathbf{Q}_1)}{\lambda_{\max}(\mathbf{Q}_3)} < 1$  and  $d_4 = \left[1 + \frac{\lambda_{\min}(\mathbf{Q}_1)}{\lambda_{\max}(\mathbf{Q}_3)}\right](\nu\sigma_{\kappa \max}^2 + \alpha_{\max}^2) + m\lambda_m(H+2)\Delta^2 t\sigma_{\mathbf{f}}^2$  where  $\lambda_m = \lambda_{\max}(\mathbf{Q}_1) + \lambda_{\max}(\mathbf{Q}_3)$ . Therefore,

$$J_{\hat{\theta}^*, \hat{W}^*}^{k+i} \leq d_3^i J_{\hat{\theta}^*, \hat{W}^*}^k + d_4 \frac{1 - d_3^i}{1 - d_3}, \quad (65)$$

and consequently,  $\|\mathbf{e}_{\mu_{\hat{\theta}}}(k+i)\| \leq a_4(i)\|\mathbf{e}_{\theta}(k)\| + a_5(i)$  where  $a_4(i) = d_3^{\frac{1}{2}}\sqrt{\frac{\lambda_{\max}(\mathbf{Q}_3)}{\lambda_{\min}(\mathbf{Q}_1)}}$  and  $a_5(i) = \sqrt{\frac{d_3^{\frac{1}{2}}(\alpha_{\max}^2 + \nu\sigma_{\kappa}^2) + d_4\frac{1-d_3^i}{1-d_3}}{\lambda_{\min}(\mathbf{Q}_1)}}$ . This proves the lemma.

#### H. Proof of Lemma 6

Plugging the iterative relation (30) for  $\mu_{\hat{\theta}}(k+i|k)$  and counterpart for  $\mu_{\theta}(k+i|k)$  into  $\tilde{\mu}_{\theta}(k+i)$ , the difference is then

$$\begin{aligned} \tilde{\mu}_{\theta}(k+i) &= \|\mathbf{F}\|\|\tilde{\mu}_{\theta}(k+i-1)\| + \|\mathbf{G}\| \\ &\quad \|\mu_{gp_{\theta}}(\mu_{\theta}(k+i-1), \alpha(k+i-1)) - \\ &\quad \mu_{gp_{\theta}}(\mu_{\hat{\theta}}(k+i-1|k), \hat{\alpha}(k+i-1|k))\| \\ &\leq \|\mathbf{F}\|\|\tilde{\mu}_{\theta}(k+i-1)\| + L_2\|e_{\alpha}(k+i-1)\| \\ &\quad + L_3\|\mathbf{G}\|\|\tilde{\mu}_{\theta}(k+i-1)\| \\ &= (\|\mathbf{F}\| + L_3\|\mathbf{G}\|\|\tilde{\mu}_{\theta}(k+i-1)\| + \\ &\quad L_2\|\mathbf{G}\|\|e_{\alpha}(k+i-1)\|. \end{aligned}$$

In the above derivations, we use the Lipschitz assumptions. For small sampling period  $\Delta t \ll 1$ ,  $\|\mathbf{F}\| \approx 1$  and  $\|\mathbf{G}\| = \Delta t$ , when  $i=1$ , with the fact that  $\mu_{\theta}(k|k) = \mu_{\hat{\theta}}(k|k)$ , we have  $\tilde{\mu}_{\theta}(k+1) \leq L_2\Delta t\|e_{\alpha}(k)\|$ . For  $i \geq 2$ , applying the above process iteratively, we obtain

$$\begin{aligned} \tilde{\mu}_{\theta}(k+i) &\leq \sum_{j=0}^{i-1} (\|\mathbf{F}\| + L_3\|\mathbf{G}\|)^{i-j-1} L_2\|\mathbf{G}\| \\ &\quad \left[ d_1 e^{\frac{\lambda_1}{4\epsilon}j\Delta t} \|e_{\alpha}(k)\| + d_2 \right], \end{aligned}$$

where the result in Lemma 2 is used to obtain  $\|e_{\alpha}(k+j)\| \leq d_1 e^{\frac{\lambda_1}{4\epsilon}j\Delta t} \|e_{\alpha}(k)\| + d_2$ . Using approximation  $(1 + L_3\Delta t)^{i-j-1} \approx 1 + (i-j-1)L_3\Delta t$  for small  $L_3\Delta t \ll 1$ , we obtain the upper-bound as shown in the lemma.

#### I. Proof of Lemma 7

Substituting the iterative model similar to (30) for both  $\mu_{\theta}(k+i|k)$  and  $\theta(k+i)$ , the error calculation is reduced to

$$\begin{aligned} \theta_{\mu}(k+i) &= \mathbf{F}\theta_{\mu}(k+i-1) + \mathbf{G}[\mu_{gp_{\theta}}(k+i-1|k) - \\ &\quad \mathbf{f}_{\theta}(k+i-1)]. \end{aligned}$$

Taking the norm on both sides of the above equation and using approximation  $\|\mathbf{F}\| \approx 1$ ,  $\|\mathbf{G}\| = \Delta t$ , and assumption (48), we obtain the iterative relationship of the error bound as  $\|\theta_{\mu}(k+i)\| \leq \|\theta_{\mu}(k+i-1)\| + \Delta t\|\beta_{\theta}^T \Sigma_{gp_{\theta}}^{\frac{1}{2}}(k+i-1|k)\|$ . With the initial condition  $\mu_{\theta}(k|k) = \theta(k)$ , we then obtain that  $\|\theta_{\mu}(k+i)\| \leq \Delta t \sum_{j=0}^{i-1} \|\beta_{\theta}^T \Sigma_{gp_{\theta}}^{\frac{1}{2}}(k+j|k)\|$ .

#### J. Proof of Lemma 8

To assess  $\bar{J}_{\hat{\theta}^*, \hat{W}^*}^k - \bar{J}_{\theta, \hat{W}^*}^k$ , we use (31)-(44) and obtain

$$\begin{aligned} &\bar{J}_{\hat{\theta}^*, \hat{W}^*}^k - \bar{J}_{\theta, \hat{W}^*}^k \\ &= \sum_{i=0}^H \{ \mathbb{E}[\|\mathbf{e}_{\hat{\theta}}(k+i|k)\|_{\mathbf{Q}_1}^2] - \|\mathbf{e}_{\theta}(k+i)\|_{\mathbf{Q}_1}^2 \} + \\ &\quad \mathbb{E}[\|\mathbf{e}_{\hat{\theta}}(k+H+1)\|_{\mathbf{Q}_3}^2] - \|\mathbf{e}_{\theta}(k+H+1)\|_{\mathbf{Q}_3}^2 \\ &= \sum_{i=0}^H \{ \|\mathbf{e}_{\mu_{\hat{\theta}}}(k+i)\|_{\mathbf{Q}_1}^2 + \text{tr}(\mathbf{Q}_1 \Sigma_{\hat{\theta}}(k+i)) - \\ &\quad \|\mathbf{e}_{\theta}(k+i)\|_{\mathbf{Q}_1}^2 \} + \|\mathbf{e}_{\mu_{\hat{\theta}}}(k+H+1)\|_{\mathbf{Q}_1}^2 + \\ &\quad \text{tr}(\mathbf{Q}_3 \Sigma_{\hat{\theta}}(k+H+1)) - \|\mathbf{e}_{\theta}(k+H+1)\|_{\mathbf{Q}_3}^2 \\ &= \sum_{i=0}^H \{ -\|\tilde{\theta}_{\mu}(k+i)\|_{\mathbf{Q}_1}^2 + \text{tr}(\mathbf{Q}_1 \Sigma_{\hat{\theta}}(k+i)) + \\ &\quad 2\tilde{\theta}_{\mu}^T(k+i)\mathbf{Q}_1\mathbf{e}_{\mu_{\hat{\theta}}}(k+i) \} - \|\tilde{\theta}_{\mu}(k+H+1)\|_{\mathbf{Q}_3}^2 + \\ &\quad \text{tr}(\mathbf{Q}_3 \Sigma_{\hat{\theta}}(k+H+1)) + \\ &\quad 2\tilde{\theta}_{\mu}^T(k+H+1)\mathbf{Q}_3\mathbf{e}_{\mu_{\hat{\theta}}}(k+H+1). \end{aligned} \quad (66)$$

The above last equality comes from the observation:

$$\begin{aligned} &\|\mathbf{e}_{\mu_{\hat{\theta}}}\|_{\mathbf{Q}_1}^2 - \|\mathbf{e}_{\theta}\|_{\mathbf{Q}_1}^2 = \|\mathbf{e}_{\mu_{\hat{\theta}}}\|_{\mathbf{Q}_1}^2 - \|\mathbf{e}_{\mu_{\hat{\theta}}} - \tilde{\theta}_{\mu}\|_{\mathbf{Q}_1}^2 \\ &= \|\mathbf{e}_{\mu_{\hat{\theta}}}\|_{\mathbf{Q}_1}^2 - \|\mathbf{e}_{\mu_{\hat{\theta}}}\|_{\mathbf{Q}_1}^2 - \|\tilde{\theta}_{\mu}\|_{\mathbf{Q}_1}^2 + 2\tilde{\theta}_{\mu}^T\mathbf{Q}_1\mathbf{e}_{\mu_{\hat{\theta}}} \\ &= -\|\tilde{\theta}_{\mu}\|_{\mathbf{Q}_1}^2 + 2\tilde{\theta}_{\mu}^T\mathbf{Q}_1\mathbf{e}_{\mu_{\hat{\theta}}} \end{aligned}$$

with  $\mathbf{e}_{\mu_{\hat{\theta}}} = \mu_{\hat{\theta}} - \theta_d$ ,  $\tilde{\theta}_{\mu} = \mu_{\hat{\theta}} - \theta$  and  $\mathbf{e}_{\theta} = \theta - \theta_d$ . The rationale to use the above formulation is to put a bound on  $\bar{J}_{\hat{\theta}^*, \hat{W}^*}^k - J_{\theta, \hat{W}^*}^k$  by terms  $\|\tilde{\theta}_{\mu}\|$  and  $\|\mathbf{e}_{\mu_{\hat{\theta}}}\|$ .

Since  $\lambda_{\max}(\mathbf{Q}_1) < \lambda_{\max}(\mathbf{Q}_3)$ , from (66), we have

$$\begin{aligned} &|\bar{J}_{\hat{\theta}^*, \hat{W}^*}^k - \bar{J}_{\theta, \hat{W}^*}^k| \leq \lambda_{\max}(\mathbf{Q}_3) \sum_{i=0}^{H+1} \{ \|\tilde{\theta}_{\mu}(k+i)\|^2 + \\ &\quad \text{tr}(\Sigma_{\hat{\theta}}(k+i)) + 2\|\mathbf{e}_{\mu_{\hat{\theta}}}(k+i)\| \|\tilde{\theta}_{\mu}(k+i)\| \}. \end{aligned} \quad (67)$$

From Lemma 5,  $\|\mathbf{e}_{\mu_{\hat{\theta}}}(k+i)\| \leq a_4(i)\|\mathbf{e}_{\theta}(k)\| + a_5(i)$ . From Lemma 3,  $\|\Sigma_{\hat{\theta}}(k+i)\| \leq i(\Delta t)^2\sigma_{\mathbf{f}}^2$ . Noting that matrix  $\Sigma_{\hat{\theta}}(k+i)$  is diagonal, we obtain  $\text{tr}(\Sigma_{\hat{\theta}}(k+i)) \leq m\|\Sigma_{\hat{\theta}}(k+i)\| \leq im(\Delta t)^2\sigma_{\mathbf{f}}^2$ . Adding the above upper bounds for each term in (67), we obtain that  $|\bar{J}_{\hat{\theta}^*, \hat{W}^*}^k - \bar{J}_{\theta, \hat{W}^*}^k| \leq \rho_J(e_{\alpha}, e_{\theta})$ , where  $\rho_J(e_{\alpha}, e_{\theta})$  is given by (49).

#### K. Proof of Theorem 1

First, it is straightforward to obtain the lower-bound of  $V(k)$  by the fact that  $V(k) \geq \lambda_{\min}(\mathbf{Q}_1)\|\mathbf{e}_{\theta}(k)\|^2 + \zeta\lambda_{\min}(\mathbf{Q})\|e_{\alpha}(k)\|^2 \geq \underline{\lambda}\|e(k)\|^2$  and similarly for upper-bound  $V(k) \leq \bar{\lambda}\|e(k)\|^2$ . Therefore, we have

$$\underline{\lambda}\|e(k)\|^2 \leq V(k) \leq \bar{\lambda}\|e(k)\|^2. \quad (68)$$

From (45), we obtain

$$\begin{aligned} \Delta V(k) &\leq |\bar{J}_{\hat{\theta}^*, \hat{W}^*}^{k+1} - \bar{J}_{\theta, \hat{W}^*}^{k+1}| + |\bar{J}_{\hat{\theta}^*, \hat{W}^*}^k - \bar{J}_{\theta, \hat{W}^*}^k| + \\ &\quad \zeta[V_{\alpha}(k+1) - V_{\alpha}(k)] + \nu[\|\Sigma_d(\hat{W}^*(k))\| - \\ &\quad \|\Sigma_d(\hat{W}^*(k+1))\|] + (J_{\hat{\theta}^*, \hat{W}^*}^{k+1} - J_{\theta, \hat{W}^*}^{k+1}). \end{aligned} \quad (69)$$

We apply the results in Lemma 8 to the first two terms in the above equation. For the third difference term, from Lemma 2, we consider the discrete-time form of (62)

$$\begin{aligned} V_\alpha(k+1) - V_\alpha(k) &\leq -\frac{1}{4}\Delta t e_\alpha^T(k) \mathbf{Q} e_\alpha(k) + c_3 \Delta t \\ &\leq -\frac{1}{4}\Delta t \lambda_{\min}(\mathbf{Q}) \|e_\alpha(k)\|^2 + c_3 \Delta t. \end{aligned}$$

For the last two difference terms in (69), by (63) we have

$$\begin{aligned} &J_{\hat{\theta}^*, \hat{W}^*}^{k+1} - J_{\hat{\theta}^*, \hat{W}^*}^k + \nu [\|\Sigma_d(\hat{W}^*(k))\| - \\ &\|\Sigma_d(\hat{W}^*(k+1))\|] \\ &\leq -\lambda_{\min}(\mathbf{Q}_1) \|e_\theta(k)\|^2 + \Delta \hat{\alpha}_{Q_2}^* k + \\ &\nu [\|\Sigma_d(\hat{W}^e(k+1))\| - \|\Sigma_d(\hat{W}^*(k+1))\|] + \\ &\text{tr}(\mathbf{Q}_1 \Sigma_{\hat{\theta}}(k+H+1)) + \text{tr}(\mathbf{Q}_3 \Sigma_{\hat{\theta}}(k+H+2)) \\ &\leq -\lambda_{\min}(\mathbf{Q}_1) \|e_\theta(k)\|^2 + \hat{\alpha}_{\max}^2 + \nu \sigma_{\kappa \max}^2 + \\ &m \lambda_m (H+2) (\Delta t)^2 \sigma_f^2. \end{aligned}$$

In the above last inequality, we use the facts that  $\|\hat{\alpha}^*(k+1)\|_{Q_2}^2 \leq \hat{\alpha}_{\max}^2$  and  $\|\Sigma_d(\hat{W}^e(k+1))\| \leq \sigma_{\kappa \max}^2$ .

Substituting the above derivations into (69), we obtain

$$\begin{aligned} \Delta V(k) &\leq \xi_1 \|e_\alpha(k)\|^2 + \xi_2 \|e_\alpha(k)\| \|e_\theta(k)\| + \xi_3 \|e_\alpha(k)\| \\ &+ \xi_4 \|e_\theta(k)\| + \xi_5 - \frac{\zeta}{4} \Delta t \lambda_{\min}(\mathbf{Q}) \|e_\alpha(k)\|^2 + \\ &\zeta c_3 \Delta t - \lambda_{\min}(\mathbf{Q}_1) \|e_\theta(k)\|^2 + \hat{\alpha}_{\max}^2 + \nu \sigma_{\kappa \max}^2 \\ &+ m \lambda_m (H+2) (\Delta t)^2 \sigma_f^2 \\ &= -\frac{1}{2} \left( \gamma_3 \|e_\theta(k)\| - \frac{\xi_2}{\gamma_3} \|e_\alpha(k)\| \right)^2 - \frac{\gamma_3^2}{4} \|e(k)\|^2 \\ &- (\gamma_1 \|e_\alpha(k)\| - \gamma_2)^2 - \left( \frac{\gamma_3}{2} \|e_\theta(k)\| - \gamma_4 \right)^2 + \gamma_5 \end{aligned}$$

if (50) is held. Considering the above result and (68), we have  $V(k+1) \leq \gamma V(k) + \gamma_\kappa$  and this proves the theorem.

The unconstrained MPC given in (36) is solved at each step by a gradient decent method. The gradient of the objective function with respect to the design variables is obtained by a back propagation approach. From (35) and (31), it is straightforward to obtain

$$\begin{aligned} J_{\hat{\theta}, \hat{W}}^k &= \sum_{i=0}^H l_s(k+i) + \|\hat{\alpha}(k)\|_{Q_2}^2 + l_f(k+H+1) \\ &+ \nu \|\Sigma_d(k)\| \end{aligned}$$

The partial derivatives of  $J_{\hat{\theta}, \hat{W}}^k$  with respect to  $\hat{\theta}_{k+i} = \hat{\theta}(k+i|k)$ ,  $i = 0, \dots, H+1$ , is obtained as

$$\frac{\partial J_{\hat{\theta}, \hat{W}}^k}{\partial \hat{\theta}_{k+i}} = \begin{cases} \frac{\partial l_s(k+i)}{\partial \hat{\theta}_{k+i}}, & i = 0, \dots, H \\ \frac{\partial l_f(k+H+1)}{\partial \hat{\theta}_{k+i}}, & i = H+1. \end{cases}$$

Noting that the current state  $\hat{\theta}(k+i|k)$  affects the future states  $\hat{\theta}(k+i+1|k)$ , the gradient calculation has the following backward iterative relationship

$$\frac{dJ_{\hat{\theta}, \hat{W}}^k}{d\hat{\theta}_{k+i}} = \frac{\partial J_{\hat{\theta}, \hat{W}}^k}{\partial \hat{\theta}_{k+i}} + \frac{\partial J_{\hat{\theta}, \hat{W}}^k}{\partial \hat{\theta}_{k+i+1}} \frac{\partial \hat{\theta}_{k+i+1}}{\partial \hat{\theta}_{k+i}} \quad (70)$$

with the terminal condition  $\frac{dJ_{\hat{\theta}, \hat{W}}^k}{d\hat{\theta}_{k+H+1}} = \frac{\partial J_{\hat{\theta}, \hat{W}}^k}{\partial \hat{\theta}_{k+H+1}}$ . The similar computation is applied to obtain the partial derivatives of  $J_{\hat{\theta}, \hat{W}}^k$

with respect to  $\hat{w}(k+i)$  and we omit here. We conduct the computation from the terminal condition at  $i = K+H+1$  and then back propagate for  $i = 1, \dots, H$  in (70).

#### L. Rotary inverted pendulum

The rotary inverted pendulum dynamics model is obtained from Lagrangian mechanics and written in the form of (1) with  $\mathbf{q} = [\theta_1 \ \alpha_1]^T$ ,  $\dot{\mathbf{q}} = [\dot{\theta}_2 \ \dot{\alpha}_2]^T$ ,  $\mathbf{u} = V_m$  and

$$\mathbf{D} = \begin{bmatrix} D_1 & D_3 \\ D_3 & D_2 \end{bmatrix}, \mathbf{H} = \begin{bmatrix} H_1 \\ H_2 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

where

$$\begin{aligned} D_1 &= m_p l_r^2 + \frac{1}{4} m_p l_p^2 \sin^2 \alpha_1 + J_r, \\ D_3 &= -\frac{1}{2} m_p l_p l_r \cos \alpha_1, D_2 = J_p + \frac{1}{4} m_p l_p^2, \\ H_1 &= \frac{1}{2} m_p l_p^2 \theta_2 \alpha_2 \sin \alpha_1 \cos \alpha_1 + \frac{1}{2} m_p l_p l_r \alpha_2^2 \sin \alpha_1 \\ &+ d_r \theta_2 + k_g^2 k_t k_m / R_m \theta_2, \\ H_2 &= -\frac{1}{4} m_p l_p^2 \cos \alpha_1 \sin \alpha_1 \theta_2^2 + d_p \alpha_2 - \frac{1}{2} m_p l_p g \sin \alpha_1, \end{aligned}$$

$l_r$  and  $J_r$  denote the base arm's length and moment of inertia, respectively, and  $l_p$ ,  $m_p$ , and  $J_p$  denotes the pendulum link's length, mass, and moment of inertia, respectively. Parameters  $d_r$  and  $d_p$  are the viscous damping coefficients of the base arm and pendulum joints, respectively.  $k_g$ ,  $k_t$ ,  $k_m$ , and  $R_m$  are DC motor's electromechanical parameters [46]. It is straightforward to write the above dynamics in the form of (13) with

$$f_\theta(\theta_2, \boldsymbol{\alpha}, u_d) = \frac{D_2 V_m - H_1 D_2 + D_3 H_2}{D_1 D_2 - D_3^2}$$

and  $\kappa_\alpha(\theta_2, \boldsymbol{\alpha}, \dot{\alpha}_2) = H_1 - \frac{1}{D_3} [(D_3 + D_1 D_2 - D_3^2) \dot{\alpha}_2 + H_2 D_1]$ .

#### M. Autonomous bikebot

The physical model of bikebot is obtained from Lagrangian mechanics and the bicycle kinematics relationship. As shown in Fig. 4(c), considering a kinematic model and a nonholonomic constraint at rear contact point  $C_2$ , the motion equation of  $C_2$  is written as

$$\ddot{X} = \dot{v}_c \cos \psi - v_c \sin \psi \dot{\psi}, \quad (71a)$$

$$\ddot{Y} = \dot{v}_c \sin \psi + v_c \cos \psi \dot{\psi}. \quad (71b)$$

The yaw rate is calculated from the geometric relationship between the steering and rear frames as [47]

$$\dot{\psi} = \frac{v_c \cos \xi}{l \cos \varphi} \tan \phi \quad (72)$$

and the equation of roll motion is obtained as

$$J_t \ddot{\varphi} = -m_b h_b \frac{v_c^2}{l} \cos \xi \tan \phi + m_b h_b g \sin \varphi, \quad (73)$$

where  $m_b$  is the total mass of the bikebot,  $l$  is the wheel base (i.e., distance between wheel contact points  $C_1$  and  $C_2$ ),  $\xi$  is the steering casting angle,  $h_b$  is the height of mass center,  $J_t = m_b h_b^2 + J_b$  is the mass moment of inertia of the bikebot along the  $x$  axis of frame  $\mathcal{R}$  ( $J_b$  is the mass moment of inertia along the  $x_b$  axis of body frame  $\mathcal{B}$ ); see Fig. 4(c). From

bikebot model (71), the steering angle input  $\phi$  affects both the balancing and the position tracking tasks.

Plugging (72) into (71) and combining with (73), we obtain the dynamic model in (52) with functions

$$\mathbf{f}_\theta(\boldsymbol{\theta}, \boldsymbol{\alpha}, \mathbf{u}) = \begin{bmatrix} u_f \cos \psi - \frac{v_c^2 \sin \psi \cos \xi}{l \cos \varphi} \tan u_d \\ u_f \sin \psi + \frac{v_c^2 \cos \psi \cos \xi}{l \cos \varphi} \tan u_d \end{bmatrix}$$

and  $\kappa_\alpha(\boldsymbol{\theta}, \boldsymbol{\alpha}, \dot{\alpha}_2, u_f) = \arctan\left(\frac{m_b h_b g l \sin \varphi - J_p l \ddot{\varphi}}{m_b h_b v_c^2 \cos \xi}\right) - \ddot{\varphi}$ . Note that in the above equations, yaw angle is calculated as  $\psi = \text{atan2}(\dot{Y}, \dot{X})$  from state variable  $\boldsymbol{\theta}_2$ .

## REFERENCES

- [1] A. Choukchou-Braham, B. Cherki, M. Djemaï, and K. Busawon, *Analysis and Control of Underactuated Mechanical Systems*. New York, NY: Springer, 2014.
- [2] N. Getz, "Dynamic inversion of nonlinear maps with applications to nonlinear control and robotics," Ph.D. dissertation, Dept. Electr. Eng. and Comp. Sci., Univ. Calif., Berkeley, CA, 1995.
- [3] J. Lee, R. Mukherjee, and H. K. Khalil, "Output feedback stabilization of inverted pendulum on a cart in the presence of uncertainties," *Automatica*, vol. 54, pp. 146–157, 2015.
- [4] A. S. Shiriaev, L. B. Freidovich, A. Robertsson, R. Johansson, and A. Sandberg, "Virtual-holonomic-constraints-based design of stable oscillations of Furuta pendulum: Theory and experiments," *IEEE Trans. Robotics*, vol. 23, no. 4, pp. 827–832, 2007.
- [5] M.-S. Park and D. Chwa, "Orbital stabilization of inverted-pendulum systems via coupled sliding-mode control method," *IEEE Trans. Ind. Electron.*, vol. 56, no. 9, pp. 3556–3570, 2009.
- [6] L. B. Freidovich, A. S. Shiriaev, F. Gordillo, F. Gómez-Estern, and J. Aracil, "Partial-energy-shaping control for orbital stabilization of high-frequency oscillations of the Furuta pendulum," *IEEE Trans. Contr. Syst. Technol.*, vol. 17, no. 4, pp. 853–858, 2009.
- [7] J. Yi, D. Song, A. Levandowski, and S. Jayasuriya, "Trajectory tracking and balance stabilization control of autonomous motorcycles," in *Proc. IEEE Int. Conf. Robot. Autom.*, Orlando, FL, 2006, pp. 2583–2589.
- [8] P. Wang, J. Yi, T. Liu, and Y. Zhang, "Trajectory tracking and balance control of an autonomous bikebot," in *Proc. IEEE Int. Conf. Robot. Autom.*, Singapore, 2017, pp. 2414–2419.
- [9] E. R. Westervelt, J. W. Grizzle, C. Chevallereau, J. H. Choi, and B. Morris, *Feedback Control of Dynamic Bipedal Robot Locomotion*. Boca Raton, FL: CRC Press, 2007.
- [10] K. Chen, M. Trkov, and J. Yi, "Hybrid zero dynamics of human walking with foot slip," in *Proc. Amer. Control Conf.*, Seattle, WA, 2017, pp. 2124–2129.
- [11] M. Trkov, K. Chen, and J. Yi, "Bipedal model and extended hybrid zero dynamics of human walking with foot slips," *ASME J. Computat. Nonlinear Dyn.*, vol. 14, no. 10, 2019, article 101002.
- [12] J. Grizzle, M. Di Benedetto, and F. Lamnabhi-Lagarigue, "Necessary conditions for asymptotic tracking in nonlinear systems," *IEEE Trans. Automat. Contr.*, vol. 39, no. 9, pp. 1782–1794, 1994.
- [13] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA: MIT Press, 2006.
- [14] J. Ko, D. J. Klein, D. Fox, and D. Haehnel, "Gaussian processes and reinforcement learning for identification and control of an autonomous blimp," in *Proc. IEEE Int. Conf. Robot. Autom.*, Roma, Italy, 2007, pp. 742–747.
- [15] J. Kocijan, R. Murray-Smith, C. E. Rasmussen, and A. Girard, "Gaussian process model based predictive control," in *Proc. Amer. Control Conf.*, Boston, MA, 2004, pp. 2214–2219.
- [16] M. P. Deisenroth, D. Fox, and C. E. Rasmussen, "Gaussian processes for data-efficient learning in robotics and control," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 37, no. 2, pp. 408–423, 2015.
- [17] G. Cao, E. M.-K. Lai, and F. Alam, "Gaussian process model predictive control of an unmanned quadrotor," *J. Intell. Robot. Syst.*, vol. 88, no. 1, pp. 147–162, 2017.
- [18] C. J. Ostafew, A. P. Schoellig, and T. D. Barfoot, "Robust constrained learning-based NMPC enabling reliable mobile robot path tracking," *Int. J. Robot. Res.*, vol. 35, no. 13, pp. 1547–1563, 2016.
- [19] R. Murray-Smith, D. Sbarbaro, C. E. Rasmussen, and A. Girard, "Adaptive, cautious, predictive control with gaussian process priors," *IFAC Proc. Vol.*, vol. 36, no. 16, pp. 1155–1160, 2003.
- [20] T. Koller, F. Berkenkamp, M. Turchetta, and A. Krause, "Learning-based model predictive control for safe exploration," in *Proc. IEEE Conf. Decision Control*, Miami Beach, FL, 2018, pp. 6059–6066.
- [21] C. D. McKinnon and A. P. Schoellig, "Learning probabilistic models for safe predictive control in unknown environments," in *Proc. Europ. Control Conf.*, Napoli, Italy, 2019, pp. 2472–2479.
- [22] K. Chen, J. Yi, and T. Liu, "Learning-based modeling and control of underactuated balance robotic systems," in *Proc. IEEE Conf. Automat. Sci. Eng.*, Xi'an, China, 2017, pp. 1118–1123.
- [23] D. Nguyen-Tuong and J. Peters, "Model learning for robot control: A survey," *Cogn. Process.*, vol. 12, pp. 319–340, 2011.
- [24] D. Nguyen-Tuong, J. Peters, M. Seeger, and B. Schölkopf, "Learning inverse dynamics: a comparison," in *Europ. Symp. Artificial Neural Networks*, 2008.
- [25] D. Nguyen-Tuong, M. Seeger, and J. Peters, "Model learning with local Gaussian process regression," *Auton. Robots*, vol. 23, no. 15, pp. 2015–2034, 2009.
- [26] T. Beckers, J. Umlauf, D. Kulic, and S. Hirche, "Stable gaussian process based tracking control of lagrangian systems," in *Proc. IEEE Conf. Decision Control*, Melbourne, Australia, 2017, pp. 5180–5185.
- [27] M. K. Helwa, A. Heins, and A. P. Schoellig, "Provably robust learning-based approach for high-accuracy tracking control of lagrangian systems," 2018, arXiv preprint arXiv:1804.01031.
- [28] A. D. Libera and R. Carli, "A data-efficient geometrically inspired polynomial kernel for robot inverse dynamic," *IEEE Robot. Automat. Lett.*, vol. 5, no. 1, pp. 24–31, 2020.
- [29] S. Zhou, M. K. Helwa, and A. P. Schoellig, "Design of deep neural networks as add-on blocks for improving impromptu trajectory tracking," in *Proc. IEEE Conf. Decision Control*, Melbourne, Australia, 2017, pp. 5201–5207.
- [30] —, "An inversion-based learning approach for improving impromptu trajectory tracking of robots with non-minimum phase dynamics," *IEEE Robot. Automat. Lett.*, vol. 3, no. 3, pp. 1663–1670, 2018.
- [31] P. Abbeel, A. Coates, and A. Y. Ng, "Autonomous helicopter aerobatics through apprenticeship learning," *Int. J. Robot. Res.*, vol. 29, no. 13, pp. 1608–1639, 2010.
- [32] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1238–1274, 2013.
- [33] I. Lenz, R. Knepper, and A. Saxena, "DeepMPC: Learning deep latent features for model predictive control," in *Proc. Robotics: Sci. Syst.*, Rome, Italy, 2015.
- [34] A. Tamar, G. Thomas, T. Zhang, S. Levine, and P. Abbeel, "Learning from the hindsight plan - Episodic MPC improvement," in *Proc. IEEE Int. Conf. Robot. Autom.*, Singapore, 2017, pp. 336–343.
- [35] 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 *Proc. IEEE Int. Conf. Robot. Autom.*, Singapore, 2017, pp. 1714–1721.
- [36] S. Schaal and C. G. Atkeson, "Learning control in robotics," *IEEE Robot. Automat. Mag.*, vol. 17, no. 2, pp. 20–29, 2010.
- [37] G. Chowdhary, H. A. Kingravi, J. P. How, and P. A. Vela, "Bayesian nonparametric adaptive control using Gaussian processes," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 26, no. 3, pp. 537–550, 2015.
- [38] J. Umlauf and L. P. S. Hirche, "An uncertainly-based control Lyapunov approach for control-affine systems modeled by Gaussian process," *IEEE Control Syst. Lett.*, vol. 2, no. 3, pp. 483–488, 2018.
- [39] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*. New York, NY: John Wiley & Sons, Inc., 2006.
- [40] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Upper Saddle River, NJ: Prentice Hall, 2002.
- [41] H. Chen and F. Allgower, "A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability," *Automatica*, vol. 34, pp. 1205–1217, 1998.
- [42] K. Chen, Y. Zhang, J. Yi, and T. Liu, "An integrated physical-learning model of physical human-robot interactions with application to pose estimation in bikebot riding," *Int. J. Robot. Res.*, vol. 35, no. 12, pp. 1459–1476, 2016.
- [43] Y. Zhang, K. Chen, and J. Yi, "Rider trunk and bicycle pose estimation with fusion of force/inertial sensors," *IEEE Trans. Biomed. Eng.*, vol. 60, no. 9, pp. 2541–2551, 2013.
- [44] N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger, "Information-theoretic regret bounds for gaussian process optimization in the bandit setting," *IEEE Trans. Inform. Theory*, vol. 58, no. 5, pp. 3250–3265, 2012.
- [45] J. B. Rawlings and D. Q. Mayne, *Model Predictive Control: Theory and Design*. Madison, WI: Nob Hill Publishing, LLC, 2009.

- [46] J. Apkarian, P. Karam, and M. Levis, *Instructor Workbook: Inverted Pendulum Experiment for Matlab/Simulink Users*, Quanser Inc., Markham, Ontario, Canada, 2011.
- [47] P. Wang, J. Yi, and T. Liu, "Stability and control of a rider-bicycle system: Analysis and experiments," *IEEE Trans. Automat. Sci. Eng.*, vol. 17, no. 1, pp. 348–360, 2020.