

An Inversion-Based Learning Approach for Improving Impromptu Trajectory Tracking of Robots with Non-Minimum Phase Dynamics

Siqi Zhou¹, Mohamed K. Helwa^{1,2}, and Angela P. Schoellig¹

Abstract—This paper presents a learning-based approach for impromptu trajectory tracking for non-minimum phase systems, i.e., systems with unstable inverse dynamics. Inversion-based feedforward approaches are commonly used for improving tracking performance; however, these approaches are not directly applicable to non-minimum phase systems due to their inherent instability. In order to resolve the instability issue, existing methods have assumed that the system model is known and used pre-actuation or inverse approximation techniques. In this work, we propose an approach for learning a stable, approximate inverse of a non-minimum phase baseline system directly from its input-output data. Through theoretical discussions, simulations, and experiments on two different platforms, we show the stability of our proposed approach and its effectiveness for high-accuracy, impromptu tracking. Our approach also shows that including more information in the training, as is commonly assumed to be useful, does not lead to better performance but may trigger instability and impact the effectiveness of the overall approach.

Index Terms—Model Learning for Control, Deep Learning in Robotics and Automation

I. INTRODUCTION

HIGH-ACCURACY trajectory tracking is essential for many robotic and automated systems. The concept of using the inverse dynamics to enforce high-accuracy or exact tracking is widely used in the control systems literature [1]. However, for many practical problems ranging from aircraft control [2] to flexible robot arm end-effector tracking [3] and hard disk drive track-following [4], the input-output dynamics are non-minimum phase — i.e., the inverse dynamics are inherently unstable. The non-minimum phase nature poses challenges in classical control design [5] and prohibits the direct application of inversion-based approaches. Moreover, in this work, we consider the task of *impromptu tracking* (i.e., tracking an arbitrary, feasible trajectory with high accuracy in one shot without further changing or tuning the control system) [6], which is even more challenging to achieve.

Manuscript received: September 10, 2017; Revised December 17, 2017; Accepted January 14, 2018.

This paper was recommended for publication by Editor Paolo Rocco upon evaluation of the Associate Editor and Reviewers' comments. This work was supported in part by OCE/SOSCIP TalentEdge Project #27901 and various NSERC research and equipment grants.

¹Siqi Zhou, Mohamed K. Helwa, and Angela P. Schoellig are with the Dynamic Systems Lab (www.dynsyslab.org), Institute for Aerospace Studies, University of Toronto, Canada. Emails: siqi.zhou@robotics.utoronto.ca, mohamed.helwa@robotics.utoronto.ca, schoellig@utias.utoronto.ca

²Mohamed K. Helwa is also with the Electrical Power and Machines Department, Cairo University, Egypt.

Digital Object Identifier (DOI): see top of this page.

Accepted final version. To appear in the *IEEE Robotics and Automation Letters (RA-L)*.

©2018 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

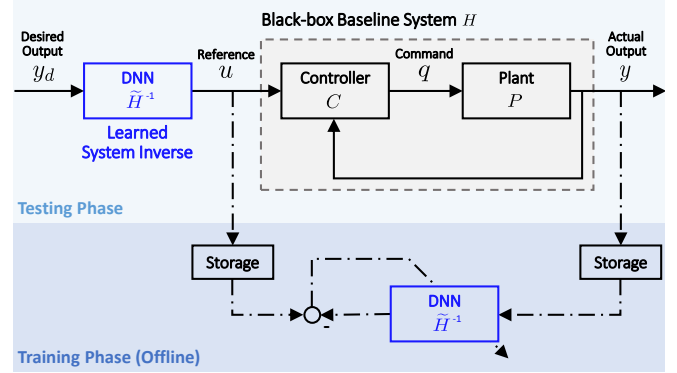


Fig. 1. An illustration of the proposed DNN-enhanced control architecture for output trajectory tracking. A stable baseline control system is treated as a black box and a DNN module is pre-cascaded to the baseline system to adjust reference signals to improve the tracking performance.

In the literature, various model-based inversion approaches have been proposed to resolve the instability issue associated with the system inverse of non-minimum phase systems. These approaches are based on (i) pre-actuation [7] or (ii) inverse approximation [8], [9]. In the pre-actuation approach, first proposed in [7], a bounded input is ensured by pre-loading the system state to a desired initial state designed for the particular desired trajectory. Though exact tracking can be achieved with bounded input signals, the solutions are trajectory-specific and require significant setup time in order to reach the desired initial condition [10]. On the other hand, in the inverse approximation approaches, stability of the inverse is ensured by replacing the unstable components of the inverse dynamics with a stable approximation that is capable of achieving precise tracking (see [8], [9] and the references therein). As compared with the pre-actuation approaches, the approximate inversion approaches are more robust against modeling errors and consequent instability issues. Moreover, since the inversion is system-specific, the approximate inversion approaches can be more easily generalized to impromptu tracking tasks. However, due to the model-based nature of both approaches, the effectiveness depends on sufficiently accurate system models. This limitation motivates the investigation of learning techniques, which leverage data to improve the performance of model-based approaches.

For *minimum phase systems*, different inverse dynamics learning approaches have been studied. In our previous work [6], [11], a deep-neural-network-based (DNN-based) architecture (Fig. 1) was proposed to enhance the tracking performance

of minimum phase black-box systems (i.e., systems whose dynamical models are not available or not sufficiently accurate). With experiments on quadrotors, it was shown that the proposed approach led to an average of 43% tracking error reduction on 30 arbitrary, hand-drawn trajectories, as compared to the baseline controller. In addition to our previous work, the potential of utilizing inverse learning for high-accuracy tracking has been demonstrated using different robotic platforms and learning techniques (e.g., Gaussian Processes (GPs) and Locally Weighted Projection Regression (LWPR)), see for instance [12]–[14]. Nevertheless, the applicability of these inversion-based learning approaches to non-minimum phase systems has not been studied, and systematically extending inverse dynamics learning schemes to non-minimum phase systems is still an open problem.

Previously, for *non-minimum phase systems*, a DNN-based adaptive feedback error learning approach has been proposed to learn an inverse of the open-loop plant for enhancing tracking [15], [16]. In this approach, the DNN training requires the plant or a good model of the plant in place, which may not always be desired in the initial training phase or available in practice. Moreover, similar to the adaptive inverse learning approaches discussed in [11], this approach is more susceptible to instability issues, especially when the DNN is not well-initialized [17].

In this paper, we present a learning-based approach that constructs an approximate inverse of a non-minimum phase, feedback-stabilized system based only on input-output data. In particular, informed by control theory, we select appropriate inputs and outputs of the inverse-learning module, prove stability of the learning-enhanced architecture for both linear and nonlinear systems, and provide theoretical insights on the inverse approximation utilized by the learning module to achieve performance enhancement. The efficacy of the proposed approach for nonlinear systems is verified with experiments on (i) an inverted pendulum on a cart system, and (ii) a modified non-minimum phase quadrotor system. For the quadrotor experiments, the generalizability of the learned inverse is verified by showing impromptu tracking of arbitrary, hand-drawn trajectories. Furthermore, we also show the connection between the proposed learning approach and a common model-based approximate inversion approach for linear systems [8], [9]. The proposed approach shares the same core concept as the model-based approach; yet, without requiring a detailed model, the proposed approach leads to better performance and is applicable to nonlinear systems.

II. PROBLEM FORMULATION

We aim to provide an inversion-based learning approach for enhancing the tracking performance of non-minimum phase systems in impromptu tracking tasks. The proposed approach should satisfy the following objectives:

- O1. Stability — the overall system, including the learning module, is input-to-output stable [18];
- O2. Training — the learning module relies only on the input-output data rather than a system model;
- O3. Performance and Generalizability — with the learning module, the root-mean-square (RMS) tracking error is

reduced for impromptu tracking tasks, compared to the baseline system.

A. Control Architecture

We consider the inversion-based learning architecture shown in Fig. 1, which consists of a baseline system and a pre-cascaded, learned system inverse module enhancing the tracking performance via modifying the reference signal u . In the training phase, the input-output data, u and y , generated from the baseline system is stored and used to construct a training dataset that typically has y and u at selected time steps as the labeled inputs and u at the current time step as the labeled output. When later using the trained module in the testing phase, the desired trajectory y_d is given to the learned inverse model as input (in place of y) to compute a reference u that is sent to the baseline system.

The considered architecture is different from typical inversion-based feedforward architectures where the inverse of the open-loop plant P is used and the output signal from the inverse is directly applied to the plant [12], [16]. By learning the inverse of a stabilized baseline system, the proposed architecture decouples the performance enhancement problem from the plant stabilization problem, which simplifies the design, analysis, and practical implementation [11].

B. System Representations

We first motivate our proposed approach by analyzing linear time-invariant (LTI), single-input-single-output (SISO) systems and then extend our discussion to nonlinear SISO systems. For linear systems, we represent the baseline feedback control system by the transfer function

$$H(z) = \frac{Y(z)}{U(z)} = \frac{N(z)}{D(z)} = \frac{1 + \sum_{i=1}^{n-r} \alpha_i z^i}{\sum_{i=0}^n \beta_i z^i}, \quad (1)$$

where $U(z)$ and $Y(z)$ are the z -transforms of the input and output of the system, $N(z)$ and $D(z)$ are the numerator and denominator polynomials, n is the order of the system, r is the relative degree of the system, and $\alpha_i, \beta_i \in \mathbb{R}$ are scalar constants. For nonlinear systems, we consider the control affine nonlinear system:

$$x(k+1) = f(x(k)) + g(x(k)) u(k), \quad y(k) = h(x(k)), \quad (2)$$

where $k \in \mathbb{Z}_{\geq 0}$ is the discrete time index, $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}$ is the input, $y \in \mathbb{R}$ is the output, and $f(\cdot)$, $g(\cdot)$, $h(\cdot)$ are nonlinear smooth functions (i.e., functions for which all orders of differentiation exist and are continuous).

C. Assumptions

In deriving a solution for our problem, we assume:

- A1. The underlying plant is stabilizable and the baseline system is stable;
- A2. At any time instant k , the current and future values of the desired trajectory are known up to time $k+n$, where n is the order of the baseline system;
- A3. The learned inverse dynamics module are feedforward neural networks (FNNs) with (A3a) finite weights and biases and (A3b) continuous activation functions $\sigma(\cdot)$.

Assumptions A1 through A3 are reasonable in practice. For A1, well-developed control methods, including model-free controllers (e.g., PID controllers), can be used to stabilize a system even in the absence of a dynamical model. For A2, a preview of n time steps of the desired trajectory is typically available, and this assumption does not prevent combinations with on-line trajectory generation and adaptation algorithms. Moreover, for A3, even though we use FNNs in this paper, the proposed approach can be potentially realized with other nonlinear regression techniques (e.g., GPs and LWPR). Assumption A3a can always be satisfied with standard DNN training algorithms, and assumption A3b holds for all common DNN activation functions (e.g., rectified linear units (ReLU), tanh, and sigmoid).

III. NON-MINIMUM PHASE SYSTEM INVERSE LEARNING

For non-minimum phase systems, one approach to resolve the instability issue in inversion-based approaches is to utilize stable inverse approximations. In this section, we adapt this concept to unknown, possibly nonlinear baseline systems using a DNN-based control architecture (Fig. 1).

A. Background on Exact Inverse Learning

Given the control architecture in Fig. 1, in [11], it is shown that for a minimum phase system with a well-defined relative degree¹ r , exact tracking (i.e., $y(k+r) = y_d(k+r)$) can be achieved by training the DNN to model the *exact inverse dynamics* of the baseline system. Following [11], for learning the exact inverse of system (2), the proper selection of inputs \mathcal{I} and outputs \mathcal{O} of the DNN module are $\mathcal{I} = \{x(k), y_d(k+r)\}$ and $\mathcal{O} = \{u(k)\}$. For LTI systems, based on the representation (1), the inputs of the DNN module can be selected as

$$\mathcal{I} = \{y_d(k-n+r : k+r), u(k-n+r : k-1)\}, \quad (3)$$

where consecutive time indices are abbreviated with ‘:’ [11]. In practice, when applying these results to design the DNN module, only basic system properties (i.e., n and r) are needed. A system’s order n can be determined from basic physics laws, and the relative degree r can be determined from simple step-response experiments. Although the *exact inverse learning* approach can be conveniently implemented in practice [6], its effectiveness is restricted to *minimum phase systems* [11].

B. The Proposed Approach: DNN Input Modification

We propose a learning approach that achieves stability (O1) and performance enhancement (O3) through modifying the DNN input selection. We first consider the linear baseline system (1), for which the exact inverse is

$$H^{-1}(z) = \frac{U(z)}{Y_d(z)} = \frac{D(z)}{N(z)} = \frac{\sum_{i=0}^n \beta_i z^i}{1 + \sum_{i=1}^{n-r} \alpha_i z^i}, \quad (4)$$

where $Y_d(z)$ is the z -transform of the desired output $y_d(k)$. For non-minimum phase systems, at least one root of the

denominator $N(z)$ is outside of the unit circle, which is the source of instability that prevents the direct application of the inverse learning scheme in (3). If the input of the DNN module is selected such that the unstable dynamics associated with $N(z)$ cannot be learned, then the instability issues would not arise. By applying the inverse z -transform to (4), it can be shown that

$$u(k) = \sum_{i=0}^n \beta_i y_d(k+i) - \sum_{i=1}^{n-r} \alpha_i u(k+i), \quad (5)$$

or

$$u(k) = F(\underbrace{y_d(k : k+n)}_{\text{from } D(z)}, \underbrace{u(k+1 : k+n-r)}_{\text{from } N(z)}), \quad (6)$$

where $F(\cdot)$ denotes a generic multi-variable function. From (6), it can be seen that the unstable dynamics associated with $N(z)$ are reflected in the dependency of $u(k)$ on the sequence of reference signals $u(k+1 : k+n-r)$.

Proposed Input-Output Selection. Based on (6), we propose the following DNN input-output selection:

$$\mathcal{I} = \{y_d(k : k+n)\} \text{ and } \mathcal{O} = \{u(k)\}, \quad (7)$$

where the sequence of u is removed from the input \mathcal{I} to prevent the DNN module from learning the unstable dynamics associated with $N(z)$.

Note that, while the proposed input-output selection is derived based on linear systems, when applying the proposed approach to nonlinear systems, the DNN module learns an approximate inverse of the nonlinear baseline system rather than a linearized baseline system. This is due to the fact that the DNN module is directly trained with the input-output data generated by the nonlinear baseline system.

C. Stability of the Proposed Approach

The proposed approach was derived from (1) to guarantee stability for the LTI systems. In this subsection, we prove stability for nonlinear systems using assumptions A1 and A3.

Lemma 1. Stability. Consider the inversion-based learning control architecture in Fig. 1 and the nonlinear system (2). Under assumptions A1 and A3, the learning module input-output selection in (7) ensures that the overall control system (from y_d to y) is input-to-output stable.

Proof. From (7), the learning module approximates a mapping from $\mathcal{I} = \{y_d(k : k+n)\}$ to $\mathcal{O} = \{u(k)\}$. For a typical L -layer FNN with $n+1$ inputs and 1 output, by denoting $\zeta_0(k) = [y_d(k) \ y_d(k+1) \ \cdots \ y_d(k+n)]^\top$ as the network input at time k , the output of a neuron i in a hidden layer l , denoted by $\zeta_{l,i}(k)$, can be expressed as $\zeta_{l,i}(k) = \sigma\left(\sum_{j=1}^{N_{l-1}} w_{l,i,j} \zeta_{l-1,j}(k) + b_{l,i}\right)$, where $\sigma(\cdot)$ is the activation function, $l \in \mathbb{N}$, $1 \leq l \leq L-1$, is the layer index, $N_l \in \mathbb{N}$ is the number of neurons in layer l , $\zeta_l \in \mathbb{R}^{N_l}$ is the output of the layer l , $w_l \in \mathbb{R}^{N_l \times N_{l-1}}$ and $b_l \in \mathbb{R}^{N_l}$ are the weights and bias associated with layer l , $\zeta_{l,i}$ and $\zeta_{l-1,j}$ are the i -th element of the vector ζ_l and the j -th element of the vector ζ_{l-1} , $w_{l,i,j}$ is the i -th row and j -th column element of the matrix w_l ,

¹See [11] and the references therein for formal definitions of relative degree. The relative degree of a discrete-time system can be intuitively thought as the inherent time delay of the system. Experimentally, it is the number of time steps between the time at which an input is applied and the system first reacts.

and $b_{l,i}$ is the i -th element of the vector b_l . The output of the network is $\tilde{F}(\zeta_0(k)) = \sum_{j=1}^{N_L-1} w_{L,1j} \zeta_{L-1,j}(k) + b_{L,1}$. By assumptions A3a and A3b, the network parameters w and b are bounded, and σ is continuous; hence, the output of each neuron i in layer l (i.e., $\zeta_{l,i}$) is continuous in ζ_0 . Moreover, since \tilde{F} is a composition of $\zeta_{l,i}$, \tilde{F} is also continuous in ζ_0 . Since every continuous function from a compact space into a metric space is bounded, the network output $u(k)$ is bounded for bounded input $\zeta_0(k)$. Furthermore, by assumption A1, the baseline system is input-to-output stable; thus, for any bounded desired trajectory y_d , the output $u(k)$ of the FNN is bounded, and the overall system from y_d to y is input-to-output stable. \square

Note that the input-to-output stability of the DNN module and the overall DNN-enhanced system rely on the fact that the proposed DNN module is a continuous, static mapping. This stability result holds for both linear and nonlinear systems and is independent of the DNN regression errors.

D. Insights on Performance Enhancement

Given that the stability (O1) is achieved through the input selection of the learning module in (7), in this subsection we address the performance enhancement objective (O3).

Insight 1. Approximate Inverse Learning. For system (1), given a sufficiently high sampling rate, the input selection in (7) enables the FNN to learn an approximate inverse, where the sequence of reference signals in the input of the exact inverse map is approximated by $u(k)$.

In order to clarify the insight above, we first present a toy example. Consider a linear function with input $\xi = [\xi_1 \ \xi_2 \ \dots \ \xi_m]^\top \in \mathbb{R}^m$ and output $v \in \mathbb{R}$: $v = F_1(\xi)$. If a particular input ξ_p is correlated to the output v by the linear function $v = F_2(\xi_p)$ and $\frac{\partial F_1}{\partial \xi_p} \neq \frac{dF_2}{d\xi_p}$, then v can be re-expressed as a linear function of the remaining components of the vector ξ : $v = F_3(\tilde{\xi})$, where $\tilde{\xi} := [\xi_1 \ \dots \ \xi_{p-1} \ \xi_{p+1} \ \dots \ \xi_m]^\top$. This implies that a regression model for the output v can be found with either ξ or $\tilde{\xi}$ as the input. This simple discussion can be generalized to the case when the removal of the dimension ξ_p does not lead to a one-to-many map from $\tilde{\xi}$ to v ; a regression model can be constructed in a lower-dimensional input space to uniquely determine the output v for a given $\tilde{\xi}$. An illustration is shown in Fig. 2. When a component of the input vector is related to the output by the function F_2 , the data points generated by F_1 are restricted to the intersection of the manifolds defined by F_1 and F_2 . When ξ_p is removed from the input of the dataset, the data points are projected onto a lower-dimensional space that is orthogonal to ξ_p .

For training, since an arbitrary smooth trajectory can be expressed as a superposition of sinusoidal functions, without loss of generality, we consider in our discussion below a single sinusoidal training trajectory of the form $u(t) = A \sin(\frac{2\pi}{T}t) + b$, where t denotes continuous time. It can be shown using Taylor series expansion of $u(t)$ that at time step k , future references $u(k+p)$ for $p = 1, \dots, n-r$ can be related to the current reference $u(k)$ by

$$u(k+p) = u(k) + \sum_{i=1}^{\infty} \left(\frac{2\pi p \Delta t}{T} \right)^i c_i(k), \quad (8)$$

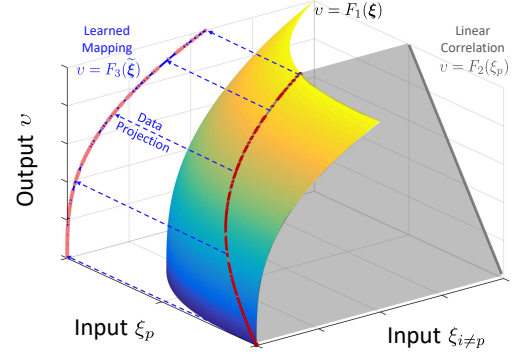


Fig. 2. Illustration of data projection in the approximate inverse learning.

where Δt denotes the sampling time and $|c_i(k)| \leq \frac{A}{i!}$. Given that p is typically a small positive number bounded by $n-r$, if Δt is sufficiently small as compared to the period of the trajectory T , then from (8), at a particular time step k , the future reference $u(k+p)$ and $u(k)$ are approximately correlated by the identity function. Given this approximate correlation and by the result above, though dependent reference components are removed from the FNN input based on the selection in (7), the FNN can still learn a regression model to output a reference u that best matches that in the training dataset. Hence, the FNN acts as an approximate inverse from output y to input u that reduces the error between y_d and y . From (8), the error involved in consecutive reference signal approximations and the inherent regression error in the learned inverse model is smaller for smaller Δt (i.e., higher sampling frequency).

For nonlinear systems, to achieve exact tracking, the learning module should model the output equation of the inverse dynamics, and $u(k)$ should be a nonlinear function of $x(k)$ and $y_d(k+r)$ (see Section III-A); however, for non-minimum phase systems, the internal instability of $x(k)$ can cause numerical issues [11]. One trivial solution is to remove the state $x(k)$ from the DNN input and use $\mathcal{I} = \{y_d(k+r)\}$. Instead, we suggest to use the same proposed input selection as in (7). A rough conjecture for this selection is as follows. Since smooth nonlinear systems can be approximated by piecewise affine/linear systems with arbitrary accuracy [19], one can always represent the considered, smooth nonlinear system as an aggregation of local, n -dimensional, affine/linear models defined on local regions of a cover/partition of the nonlinear system state space. Since all models have order n , by following the derivation in Section III-B for each local model, one obtains the same input selection as in (7) for each local model. Thus, it is reasonable to select the inputs for the DNN as in (7) even for nonlinear systems. The effectiveness of the proposed input selection for nonlinear systems is validated with simulations and experiments in Sections IV and V, respectively.

E. Connection with the ZOS Approach

In this subsection, we show a connection between the proposed approach and a model-based approximate inverse approach for linear systems, the zero-order series (ZOS) approach [8]. In the ZOS approach, the transfer function poly-

mials associated with the unstable zeros are approximated by zero-order Taylor series [8]. In particular, by re-expressing (1) as $H(z) = \frac{N_s(z)N_u(z)}{D(z)}$, the ZOS approximate inverse is

$$\tilde{H}_{\text{ZOS}}^{-1}(z) = \frac{D(z)}{N_u(z)|_{z=1}N_s(z)}, \quad (9)$$

where $N_s(z)$ and $N_u(z)$ denote the numerator polynomials with stable and unstable zeros, respectively.

Insight 2. Connection with ZOS. For linear systems, the approximation of the sequence of reference signals with the current reference $u(k)$ is equivalent to approximating the numerator of the transfer function $N(z)$ in (1) with $N(z)|_{z=1}$. With the input selection in (7), the proposed learning approach achieves stability (O1) and performance enhancement (O3) in a similar manner as the model-based ZOS approach in (9).

The time-domain representation of the exact inverse in (4) is shown in (5). When $u(k+i)$ for $i=1, \dots, n-r$ are approximated by $u(k)$ as in the proposed approach, we obtain $\sum_{i=0}^n \beta_i y(k+i) \approx \left(1 + \sum_{i=1}^{n-r} \alpha_i\right) u(k)$, or $H^{-1}(z) \approx \frac{\sum_{i=0}^n \beta_i z^i}{1 + \sum_{i=1}^{n-r} \alpha_i} = \frac{D(z)}{N(z)|_{z=1}}$ in the z -domain. By comparing the latter expression with the ZOS approximation in (9), it can be seen that they both achieve stability by approximating unstable zero dynamics at $z = 1$, and compensating for the delays introduced by the dynamics associated with the poles ($D(z)$) to improve tracking performance.

Note that the generalizability of the FNN depends on the invariance of the phase and magnitude errors of the transfer function $\frac{Y(z)}{Y_d(z)} = \frac{N(z)}{N(z)|_{z=1}}$ with respect to the frequency of the desired trajectory; it can be shown that the generalizability is better if the zeros (the roots of $N(z)$) are further away from $z = 1$. Moreover, similar to the ZOS approach [9], we expect that the proposed learning approach is more effective for enhancing the tracking performance of desired trajectories with frequencies less than the frequency of the zeros.

IV. SIMULATION RESULTS

We use an inverted pendulum on a cart system (pendulum-cart system) to illustrate the efficacy of the proposed approach for nonlinear non-minimum phase systems.

A. Simulation Setup

The pendulum-cart system has two degrees of freedom – the cart linear position η and the pendulum angular position θ . By applying Lagrangian's equations, a dynamics model of the pendulum-cart system can be obtained [20]:

$$\begin{aligned} \ddot{\eta} &= \frac{q + mg \sin \theta \cos \theta - m l \dot{\theta}^2 \sin \theta}{M + m \sin^2 \theta} \\ \ddot{\theta} &= \frac{q \cos \theta + (M + m)g \sin \theta - m l \dot{\theta}^2 \sin \theta \cos \theta}{l(M + m \sin^2 \theta)}, \end{aligned} \quad (10)$$

where M and m are the masses of the cart and the pendulum, respectively, l is the effective length of the pendulum relative to the pivot point, and q is the force applied to the cart. By defining the state of the system as $x = [\eta \ \dot{\eta} \ \theta \ \dot{\theta}]^\top$, its input as the force q , and its output as the full state $y = x$, the nonlinear

state-space representation of the pendulum-cart system can be written in the control affine form:

$$\dot{x} = f_1(x_2, x_3, x_4) + g_1(x_3) q, \quad y = x, \quad (11)$$

where $x_2 = \dot{\eta}$, $x_3 = \theta$, and $x_4 = \dot{\theta}$. The control objective is to compute a control input q such that the cart tracks a desired trajectory $\eta_d(t)$ while the pendulum is balanced at the upright position. The desired output is $y_d(t) = [\eta_d(t) \ \dot{\eta}_d(t) \ 0 \ 0]^\top$. Through linearizing the system (11) at $\eta = \eta_d$, $\dot{\eta} = 0$, $\theta = 0$, $\dot{\theta} = 0$, and $q = 0$, the pole placement technique can be used to find a stabilizing controller $q(t) = K_1(u(t) - y(t))$, where u is the reference of the baseline system and for our simulations $K_1 = [-0.8678 \ -1.808 \ 25.46 \ 4.140]$.

A learning module, pre-cascaded to the baseline system as in Fig. 1, is designed based on (7) to enhance the performance of the cart position tracking. Given the desired trajectory η_d (a component of y_d), at a time instance k , the learning module computes an adjusted reference signal η_r (a component of u) to be sent to the baseline system. The $\dot{\eta}_r$ component in u is generated from the η_r trajectory. An FNN with 2 hidden layers of 5 hyperbolic tangent neurons is used for learning the approximate inverse of the baseline system. Assuming that the baseline system succeeds to stabilize the pendulum at the upright position, then from (10), the dynamics associated with η may be approximated by a second-order system; by (7), the input and output of the learning module are selected to be $\mathcal{I} = \{\eta_d(k : k+2)\}$ and $\mathcal{O} = \{\eta_r(k)\}$. The learning module is executed at sampling intervals of 0.015 s. The module is trained on 30 sinusoidal trajectories with different combinations of amplitudes $\{0.5, 1.0, 1.5, 2.0, 2.5, 3.0\}$ m and periods $\{5, 10, 15, 20, 25\}$ s. The training dataset consists of pairs of $(\mathcal{I} = \{\eta_d(k : k+2)\}, \mathcal{O} = \{\eta_r(k)\})$ randomly sampled from the 30 training trajectories with equal proportions. Validation of the FNN model is performed on 30% of the training dataset; additional validation of the learning module is done by running the overall system on untrained trajectories.

B. Results

The tracking performance of the baseline system and the learning-enhanced system are compared in Fig. 3 for test sinusoidal trajectories with frequencies different from those used in training. From Fig. 3, although the baseline system is capable of stabilizing the pendulum-cart system, the tracking error increases with decreasing periods of desired trajectories. In contrast, when the proposed learning module is added to the baseline system, the tracking error is approximately maintained at a smaller constant value over the range of trajectory periods covered by the training dataset, which shows the generalizing capabilities of the learning approach.

Fig. 4 shows the adverse impact when a single past reference is included in the proposed input selection of the learning module, i.e., when $\mathcal{I} = \{\eta_d(k : k+2), u(k-1)\}$. It can be seen that when the additional information is included, the pendulum-cart system quickly becomes unstable. Thus, for non-minimum phase systems, the input selection of the learning module is essential; the inclusion of unnecessary inputs can prevent not only the learning approach but also the baseline system

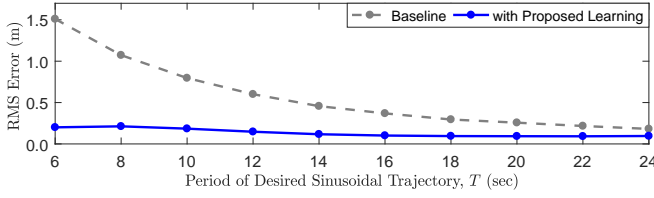


Fig. 3. The RMS tracking error of the baseline and the learning-enhanced system for desired trajectories of the form $\eta_d(t) = \frac{5}{2} \sin(\frac{2\pi}{T}t)$, where the periods T are different from those used for training. The RMS error reduction achieved by the learning module ranges from 47% to 87%. A video for $T = 12$ s can be found at: <http://tiny.cc/fq0mny>.

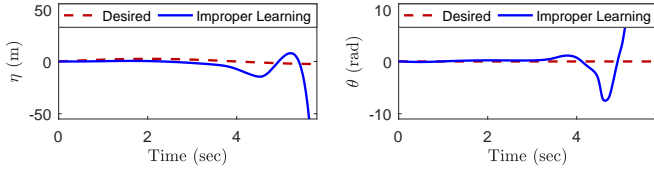


Fig. 4. Illustration of the adverse effect caused by the inclusion of an additional reference component in the input \mathcal{I} of the learning module.

from being functional. From this example, it is interesting to see that, for non-minimum phase systems, *the DNN trained with less inputs leads to a better performance*. In contrast to typical DNN applications (e.g., image classification), for control applications, the training objective (e.g., minimizing regression error) and performance objective (e.g., minimizing tracking error) may not coincide. Consequently, DNN training algorithms may not phase out unnecessary input dimensions to achieve a good performance.

V. EXPERIMENTAL RESULTS

The effectiveness of the proposed approach is further verified using pendulum-cart and quadrotor experiments. Note that, in the experiments, the criterion we use for evaluating tracking performance is the RMS tracking error, which characterizes tracking performance over entire trajectories.

A. Pendulum-Cart Experiments

1) *Experiment Setup*: The setup is similar to that of the simulation (Section IV-A), except that the input force q is replaced by the input voltage v to the cart motor. By using a simple voltage-to-force model $q(t) = -7.74\dot{\eta}(t) + 1.73v(t)$ [21], system (11) can be re-expressed as

$$\dot{x} = f_2(x_2, x_3, x_4) + g_2(x_3)v, \quad y = x, \quad (12)$$

where $\eta(t)$ and $\theta(t)$ are measured, and $x(t)$ is estimated with a full-state observer. A controller $v(t) = K_2(u(t) - y(t))$ with $K_2 = [-105.6 \ -55.04 \ 130.7 \ 23.67]$ is run at 1 kHz.

We compare the proposed learning approach with the baseline system and the model-based ZOS approach. In the experiments, the learning module is run at 70 Hz [6]; the design and training procedure for the inverse-learning module are similar to that of the simulations (see Section IV). The training dataset is constructed from 18 sinusoidal trajectories with combinations of amplitudes $\{0.04, 0.06, 0.08\}$ m and periods $\{5, 6, 7, 8, 9, 10\}$ s. The ZOS approach is implemented based on the linearized state-space model of system (12). From the linearized system, a discrete-time transfer function

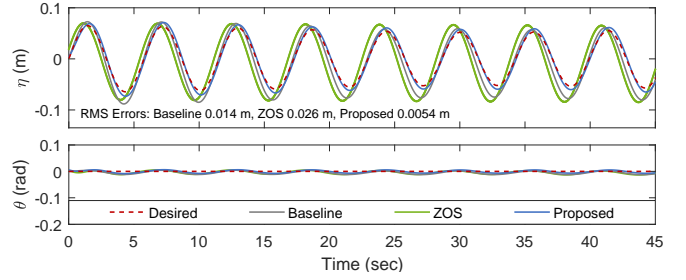


Fig. 5. The cart position η and the pendulum angle θ of the baseline, the ZOS, and the proposed learning-based systems on a test trajectory $\eta_d(t) = \frac{117}{2000} \sin(\frac{2\pi}{5}t) + \frac{13}{2000} \sin(\frac{4\pi}{11}t)$.

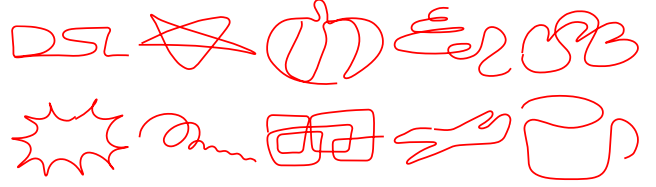


Fig. 6. Illustrations of 10 hand-drawn test trajectories used for evaluating the tracking performance of the quadrotor controllers.

from the reference η_r to the output η can be determined. By applying (9), the ZOS approximate inverse is obtained: $\tilde{H}_{\text{ZOS}}^{-1}(z) = \frac{z^4 - 3.5217z^3 + 4.6504z^2 - 2.7290z + 0.6005}{0.00137z^2 - 0.0001066z - 0.001066}$. For the experimental comparison, the ZOS approximate inverse $\tilde{H}_{\text{ZOS}}^{-1}(z)$ replaces the learning module in Fig. 1.

2) *Results*: Fig. 5 shows the comparison of the tracking performance of the baseline, the ZOS, and the proposed learning-based systems on a test trajectory $\eta_d(t) = \frac{117}{2000} \sin(\frac{2\pi}{5}t) + \frac{13}{2000} \sin(\frac{4\pi}{11}t)$, which was not included in the training phase. The stability objective is achieved by all three systems, and the pendulum position is kept approximately at the upright position. From the cart position $\eta(t)$ plot, the proposed DNN (blue) effectively compensates for the phase and magnitude errors in the baseline system response (gray). For this test trajectory, the learning module reduces the RMS tracking error by 60%.

In contrast, by comparing the $\eta(t)$ of the ZOS approach (green) with the baseline response (gray), the addition of the approximate inversion led to worse tracking performance. Though the linearized state-space model is sufficiently accurate for deriving a baseline controller that stabilizes the pendulum-cart system, the application of the model-based system inversion approach requires a much more detailed and accurate system model. Thus, in comparison with the ZOS approach, *the proposed DNN-based learning approach (blue) is capable of achieving a better performance without relying on a detailed dynamic model of the baseline system*.

B. Quadrotor Experiments

The efficacy of the proposed approach on higher degree-of-freedom systems is demonstrated using quadrotor vehicles. In this set of experiments, the objective is to enhance a baseline controller of a quadrotor for tracking arbitrary, hand-drawn trajectories (Fig. 6) in one shot [6], [11].

1) *Experiment Setup*: The state vector of the quadrotor system consists of the positions $\mathbf{p} = (x, y, z)$, velocities

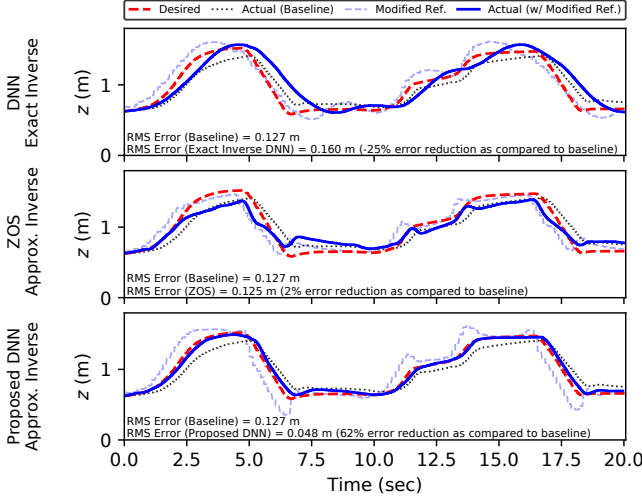


Fig. 7. Comparison of the DNN exact inverse approach [6], [11], the ZOS approximate inverse approach, and the proposed DNN approximate inverse approach for enhancing the tracking performance of the modified non-minimum phase quadrotor system. The desired z -position trajectory is from the first hand-drawn test trajectory shown in Fig. 6 (left, top).

$\mathbf{v} = (\dot{x}, \dot{y}, \dot{z})$, roll-pitch-yaw Euler angles $\boldsymbol{\theta} = (\phi, \theta, \psi)$, and rotational velocities $\boldsymbol{\omega} = (p, q, r)$. The control objective is to control the position of the quadrotor to track a desired trajectory $\mathbf{p}_d(t)$. The baseline tracking controller is a standard nonlinear controller composed of a nonlinear transformation and PD control [6] running at 70 Hz. For the purpose of studying non-minimum phase systems, non-minimum phase zeros at 1.2 are introduced to the baseline system by modifying the baseline z position and velocity references (z_r and \dot{z}_r). Note that, in this paper, we purposely introduce a non-minimum phase zero to the baseline system for evaluating our proposed approach; in practice, this non-minimum phase nature can occur in apparent minimum phase robotic systems when the sampling rate is high [22].

In the experiments, we examine three inversion-based approaches that adapt the reference signals of the baseline controller \mathbf{p}_r and \mathbf{v}_r to reduce the tracking error between the desired position \mathbf{p}_d and the actual position \mathbf{p} :

- (M1) DNN exact inverse learning: the learning-based approach effective for minimum phase systems [6], [11];
- (M2) ZOS approximate inverse: a model-based approach for non-minimum phase systems;
- (M3) DNN approximate inverse learning: the proposed learning-based approach with input-output selection based on (7).

The inverse blocks receive the desired position \mathbf{p}_d and desired velocity \mathbf{v}_d as inputs, and compute the adjusted position reference \mathbf{p}_r and velocity reference \mathbf{v}_r for the baseline system. For comparison purposes, the DNN training and architecture are similar to [6], [11]. In particular, the DNNs are fully-connected feedforward networks with 4 hidden layers of 128 ReLUs. During the training phase, the baseline system is used to track a 400-second, 3-dimensional sinusoidal trajectory, and the input-output data of the baseline system is collected at 7 Hz. The training dataset of the DNN consists of $(\mathcal{I}, \mathcal{O})$ pairs randomly sampled from the input-output data of the baseline

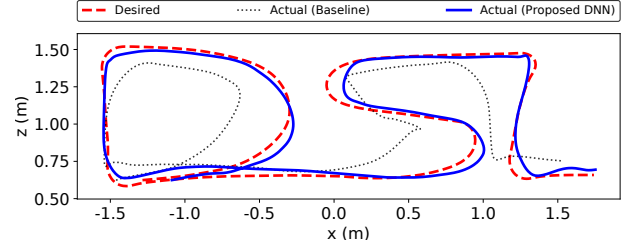


Fig. 8. Example of the performance enhancement achieved by the proposed DNN approximate inverse approach for the modified non-minimum phase quadrotor system. Here, the proposed DNN leads to a 67% error reduction.

system. Overall, 90% of the dataset is used for training, and the remainder of the dataset is used for validation. For evaluating the effectiveness and generalizability of the inversion-based approaches, test trajectories generated from arbitrary hand drawings are utilized (Fig. 6).

2) *Results*: We first examine the three inversion-based approaches for enhancing the tracking performance of the modified non-minimum phase quadrotor baseline system, where non-minimum phase zeros are introduced in the dynamics associated with the z -direction. The implementation of (M1) follows from that in [11]; the inputs and outputs of the DNN are selected to be $\mathcal{I} = \{x_d(k+5) - x(k), y_d(k+5) - y(k), z_d(k+3) - z(k), \dot{x}_d(k+4) - \dot{x}(k), \dot{y}_d(k+4) - \dot{y}(k), \dot{z}_d(k+2) - \dot{z}(k), \boldsymbol{\theta}(k), \boldsymbol{\omega}(k)\}$ and $\mathcal{O} = \{x_r(k) - x(k), y_r(k) - y(k), z_r(k) - z(k), \dot{x}_r(k) - \dot{x}(k), \dot{y}_r(k) - \dot{y}(k), \dot{z}_r(k) - \dot{z}(k)\}$. The implementation of (M2) is based on the approximation of the dynamics of the baseline system with decoupled second-order linear systems; by applying Eqn. (9), the ZOS approximate inverse is found to be $H_{\text{ZOS}}^{-1}(z) = \frac{z^3 - 1.713z^2 + 0.7493z}{0.2692z - 0.2331}$, and is applied to adjust the position and velocity references z_r and \dot{z}_r . In the implementation of (M3), we need to estimate the system order n . We assume that the quadrotor has decoupled double-integrator dynamics in the x , y , and z directions. By further accounting for the experimentally determined time delays in each direction and applying (7), the inputs and outputs of the DNN module are selected to be $\mathcal{I} = \{x_d(k+1 : k+7) - x_d(k), y_d(k+1 : k+7) - y_d(k), z_d(k+1 : k+5) - z_d(k), \dot{x}_d(k+1 : k+6) - \dot{x}_d(k), \dot{y}_d(k+1 : k+6) - \dot{y}_d(k), \dot{z}_d(k+1 : k+4) - \dot{z}_d(k)\}$ and $\mathcal{O} = \{x_r(k) - x_d(k), y_r(k) - y_d(k), z_r(k) - z_d(k), \dot{x}_r(k) - \dot{x}_d(k), \dot{y}_r(k) - \dot{y}_d(k), \dot{z}_r(k) - \dot{z}_d(k)\}$. Following previous work [6], [11], in the implementations of (M1) and (M3), we utilized a difference learning scheme (i.e., training with relative positions and velocities) to improve training efficiency.

Fig. 7 shows a comparison of the three inversion-based approaches for a test trajectory in the z -direction, $z_d(t)$, from the first hand drawing shown in Fig. 6. From the top panel, as expected, due to the inherent instability of the inverse, the approach (M1) does not lead to an improved tracking performance. Instead, it introduces undesired oscillations in the system response and leads to worse performance as compared with the baseline controller. We next consider (M2) shown in the middle panel. From the computed reference z_r (light blue dotted line), it can be seen that the model-based system approximate inversion tends to compensate for the delay in the system response; however, with the linearized model, the approximate inverse H_{ZOS}^{-1} cannot effectively reduce

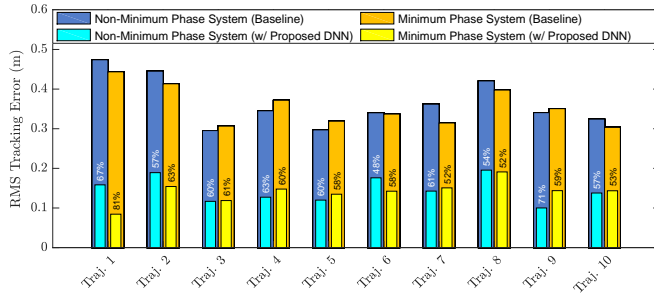


Fig. 9. RMS tracking errors on 10 hand-drawn test trajectories (shown in Fig. 6) for the modified non-minimum phase quadrotor system and the original minimum phase quadrotor system. The percentage above each bar indicates the error reduction achieved by the addition of the proposed DNN module. On average, the DNN modules lead to approximately 60% error reductions for both the non-minimum phase and minimum phase systems.

the magnitude error of the system response. In contrast, for the proposed approach (M3), shown in the bottom panel, the reference computed by the DNN module efficaciously compensates for the tracking errors of the baseline response. With (M3), the RMS tracking error in the z -direction is reduced by approximately 62%, while the percentage reductions for (M1) and (M2) are approximately -25% and 2%, respectively.

Fig. 8 shows the tracking performance of the proposed approach (M3) on the hand-drawn test trajectory corresponding to that shown in Fig. 7. On this hand-drawn test trajectory, the proposed approach reduces the 3-dimensional RMS tracking error by 67%. The generalizability of the proposed approach is tested on 10 hand-drawn trajectories (Fig. 6), which are not seen during the training phase. Fig. 9 shows a summary of the 3-dimensional RMS errors of the non-minimum phase baseline quadrotor tracking system (dark blue bars) and the system enhanced by the proposed DNN approximate inverse learning (light blue bars). On average, 60% error reduction is achieved by the proposed DNN module. In addition, the dark and light yellow bars in Fig. 9 show that the proposed DNN also effectively enhances the performance of the original minimum phase quadrotor system studied in [6], [11].

Note that, with the proposed approach, it is expected that the performance enhancement of the DNN module is better for input trajectory frequencies closer to those seen in the training phase; in practice, the DNN inverse module should be trained on a dataset that sufficiently covers the operational space.

VI. CONCLUSIONS AND FUTURE WORK

Many robotic systems can exhibit non-minimum phase behaviours; in this paper, we present a learning-based approach to enhance the impromptu tracking performance of non-minimum phase systems. In our approach, a learning module approximates the inverse of a stabilized baseline system, and the stability of the learning module is ensured through appropriate input selection. As demonstrated with experiments on a pendulum-cart and quadrotor system, the proposed approach, requiring only input-output data of the baseline system, leads to significantly better performance as compared to the ZOS approximate inverse, one of the typical model-based approaches in the literature. A promising direction for future research is to incorporate probabilistic modeling approaches to provide

uncertainty and performance enhancement estimates for the learned inverse module.

REFERENCES

- [1] G. M. Clayton, S. Tien, K. K. Leang, Q. Zou, and S. Devasia, "A review of feedforward control approaches in nanopositioning for high-speed SPM," *Journal of Dynamic Systems, Measurement, and Control*, vol. 131(6), pp. (061 101) 1–19, 2009.
- [2] S. A. Al-Hiddabi and N. H. McClamroch, "Tracking and maneuver regulation control for nonlinear nonminimum phase systems: Application to flight control," *IEEE Trans. on Control Systems Technology*, vol. 10(6), pp. 780–792, 2002.
- [3] A. De Luca, P. Lucibello *et al.*, "Inversion techniques for trajectory control of flexible robot arms," *Journal of Field Robotics*, vol. 6(4), pp. 325–344, 1989.
- [4] J. Levin, N. O. Perez-Arancibia, P. A. Ioannou, and T. Tsao, "A neural-networks-based adaptive disturbance rejection method and its application to the control of hard disk drives," *IEEE Trans. on Magnetics*, vol. 45(5), pp. 2140–2150, 2009.
- [5] J. B. Hoagg and D. S. Bernstein, "Nonminimum-phase zeros – much to do about nothing – classical control revisited Part II," *IEEE Control Systems*, vol. 27(3), pp. 45–57, 2007.
- [6] Q. Li, J. Qian, Z. Zhu, X. Bao, M. K. Helwa, and A. P. Schoellig, "Deep neural networks for improved, impromptu trajectory tracking of quadrotors," in *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2017, pp. 5183–5189.
- [7] S. Devasia, D. Chen, and B. Paden, "Nonlinear inversion-based output tracking," *IEEE Trans. on Automatic Control*, vol. 41(7), pp. 930–942, 1996.
- [8] B. P. Rigney, L. Y. Pao, and D. A. Lawrence, "Nonminimum phase dynamic inversion for settle time applications," *IEEE Trans. on Control Systems Technology*, vol. 17(5), pp. 989–1005, 2009.
- [9] J. J. E. Slotine and W. Li, *Applied nonlinear control*. Prentice Hall, Englewood Cliffs, NJ, 1991.
- [10] Y. Zhang, Q. Zhu, and R. Xiong, "Pre-action and stable inversion based precise tracking for non-minimum phase system," in *Proc. of the IEEE Conf. on Decision and Control (CDC)*, 2016, pp. 5682–5687.
- [11] 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. of the IEEE Conf. on Decision and Control (CDC)*, 2017, pp. 5201–5207.
- [12] A. S. Polydoros, L. Nalpantidis, and V. Krüger, "Real-time deep learning of robotic manipulator inverse dynamics," in *Proc. of the IEEE Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2015, pp. 3442–3448.
- [13] D. Nguyen-Tuong, J. Peters, M. Seeger, and B. Schölkopf, "Learning inverse dynamics: a comparison," in *Proc. of the European Symposium on Artificial Neural Networks (ESANN)*, no. EPFL-CONF-175477, 2008.
- [14] C. Williams, S. Klanke, S. Vijayakumar, and K. M. Chai, "Multi-task Gaussian process learning of robot inverse dynamics," in *Advances in Neural Information Processing Systems*, 2009, pp. 265–272.
- [15] S. Jung and S. S. Kim, "Control experiment of a wheel-driven mobile inverted pendulum using neural network," *IEEE Trans. on Control Systems Technology*, vol. 16(2), pp. 297–303, 2008.
- [16] A. de Almeida Neto, W. R. Neto, L. C. S. Góes, and C. L. Nascimento, "Feedback-error-learning for controlling a flexible link," in *Proc. of the IEEE Brazilian Symposium on Neural Networks*, 2000, pp. 273–278.
- [17] F.-C. Chen and H. K. Khalil, "Adaptive control of a class of nonlinear discrete-time systems using neural networks," *IEEE Trans. on Automatic Control*, vol. 40(5), pp. 791–801, 1995.
- [18] E. D. Sontag and Y. Wang, "Notions of input to output stability," *Systems & Control Letters*, vol. 38(4), pp. 235–248, 1999.
- [19] M. K. Helwa and P. E. Caines, "Epsilon controllability of nonlinear systems on polytopes," in *Proc. of the IEEE Conf. on Decision and Control (CDC)*, 2015, pp. 252–257.
- [20] A. M. Bloch, N. E. Leonard, and J. E. Marsden, "Controlled Lagrangians and the stabilization of mechanical systems I: The first matching theorem," *IEEE Trans. on Automatic Control*, vol. 45(12), pp. 2253–2270, 2000.
- [21] Quanser Consulting Inc., "IP02 self-erecting inverted pendulum user's guide," 1996, Available at: http://www.mecatronica.eesc.usp.br/wiki/upload/1/11/Manual_SelfErecting.pdf.
- [22] J. A. Butterworth, L. Y. Pao, and D. Y. Abramovitch, "The effect of nonminimum-phase zero locations on the performance of feedforward model-inverse control techniques in discrete-time systems," in *Proc. of the American Control Conference*, 2008, pp. 2696–2702.