Provably Robust Learning-Based Approach for High-Accuracy Tracking Control of Lagrangian Systems

Mohamed K. Helwa, Adam Heins, and Angela P. Schoellig

***Abstract*— Inverse dynamics control and feedforward linearization techniques are typically used to convert the complex nonlinear dynamics of robots to a set of decoupled double integrators, and then a standard, outer-loop controller can be used to calculate the commanded acceleration for the linearized system. However, these methods typically depend on having a very accurate system model, which is not available in practice. While this challenge has been addressed in the literature using different learning approaches, most of these approaches do not provide safety guarantees in terms of stability of the learning-based control system. In this paper, we provide a novel, learning-based control approach based on Gaussian processes (GPs) that ensures both stability of the closed-loop system and high-accuracy tracking. In our proposed approach, we use GPs to approximate the error between the commanded acceleration and the actual acceleration of the system, and then use the predicted mean and variance of the GP to calculate an upper bound on the uncertainty of the linearization. This uncertainty bound is then used in a robust, outer-loop controller to ensure stability of the overall system. Moreover, we show that using our proposed strategy, the tracking error converges to a ball with a radius that can be made arbitrarily small through appropriate control design. Furthermore, we verify the effectiveness of the proposed approach via simulations on *(i)* a 2 degree-of-freedom (DOF) planar manipulator, and *(ii)* a 6 DOF industrial manipulator using the Gazebo simulator.**

I. Introduction

High-accuracy tracking is an essential requirement in advanced manufacturing, self-driving cars, medical robots, and autonomous flying vehicles, among other application areas. To achieve high-accuracy tracking for these complex, typically high-dimensional, nonlinear robotic systems, a typical approach is to use inverse dynamics control [1] or feedforward linearization techniques [2] to convert the complex nonlinear dynamics into a set of decoupled double integrators, and then a standard, linear, outer-loop controller, e.g., a proportional-derivative (PD) controller, can be used to make the decoupled linear system track the desired trajectory [1]. However, these linearization techniques depend on having accurate system models, which are difficult to obtain from the first principles of physics.

To address this problem, for many decades robust control techniques have been used to design the outer-loop controllers to account for the uncertainties in the linearization [3]. However, the selection of the uncertainty bounds in the robust controller design is challenging. On the one hand, selecting high bounds typically results in a conservative behavior, and hence, a large tracking error. On the other hand, relatively small uncertainty bounds may not represent the true upper bounds of the uncertainties, and consequently, stability of the overall system is not ensured. Alternatively, several approaches have been proposed for learning the inverse system dynamics from collected data for the practical cases where the system models are not available or not sufficiently accurate; see [4]–[7]. Combining a-priori model knowledge with learning data has also been studied in [4], [8]. However, these learning approaches typically neglect the learning regression errors in the analysis, and they do not provide a proof of stability of the overall, learning-based control system, which is crucial for safety-critical applications such as medical robots. The limitations of the robust control and the learning-based techniques show the urgent need for novel, robust, learning-based control approaches that ensure both stability of the overall control system and high-accuracy tracking. This sets the stage for the research carried out in this paper.

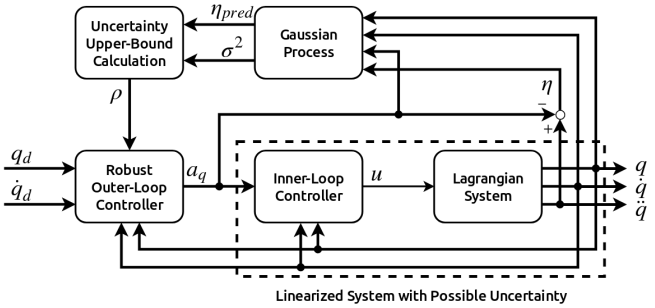

Fig. 1. Block Diagram of our proposed control strategy; GP regression models learn the uncertainty in the linearization, and then using the mean and variance of the GP, one can calculate an upper bound on the uncertainty to be used in a robust, outer-loop controller. The symbol q is the actual position vector, q_d is the desired position vector, a_q is the commanded acceleration vector, u is the force/torque input vector, and η is the uncertainty vector.

In this paper, we provide a novel, robust, learning-based control technique that achieves both closed-loop stability and high-accuracy tracking. In particular, we use Gaussian processes (GPs) to approximate the error between the commanded acceleration to the linearized system and the actual acceleration of the robotic system, and then use the predicted mean and variance of the GP to calculate an upper bound on the uncertainty of the linearization. This uncertainty bound is then used in a robust, outer-loop controller to ensure stability of the overall system (see Figure 1). Moreover, we show that using our proposed strategy, the tracking error converges to a

The authors are with the Dynamic Systems Lab (www.dynsyslab.org), Institute for Aerospace Studies, University of Toronto, Canada. E-mail: mohamed.helwa@robotics.utias.utoronto.ca, adam.heins@robotics.utias.utoronto.ca, schoellig@utias.utoronto.ca.
This research was supported by NSERC grant RGPIN-2014-04634 and OCE/SOSCIP TalentEdge Project #27901.

Preprint submitted to 57th IEEE Conference on Decision and Control.
Received March 20, 2018.

ball with a radius that can be made arbitrarily small through appropriate control design, and hence, our proposed approach also achieves high-accuracy tracking. Furthermore, we verify the effectiveness of the proposed approach via simulations on *(i)* a 2 DOF planar manipulator using MATLAB Simulink, and *(ii)* a 6 DOF industrial manipulator using the Gazebo simulator.

This paper is organized as follows. Section II provides a summary of some recent related work. Section III describes the considered problem, and Section IV provides the proposed approach for solving the problem. Section V provides theoretical guarantees of the effectiveness of the proposed approach. Section VI provides the simulation results, and Section VII concludes the paper.

Notation and Basic Definitions: For a set S, $\bar{S}$ denotes its closure and S° its interior. The notation $B_\delta(y)$ denotes a ball of radius δ centered at a point y. A matrix P is *positive definite* if it is symmetric and all its eigenvalues are positive. For a vector x, $\|x\|$ denotes its Euclidean norm. We say a function $f(x)$ is *smooth* if its partial derivatives of all orders exist and are continuous. We say that the solutions of $\dot{x} = f(t,x)$ are *uniformly ultimately bounded with ultimate bound* b if there exist positive constants $b,\ c$, and for every $0 < a < c$, there exists $T(a,b) \geq 0$ such that $\|x(t_0)\| \leq a$ implies $\|x(t)\| \leq b$, for all $t \geq (T + t_0)$, where t_0 is the initial time instant. A *kernel* is a symmetric function $k : \mathcal{A} \times \mathcal{A} \to \mathbb{R}$. A *reproducing kernel Hilbert space (RKHS)* corresponding to a kernel $k(.,.)$ includes functions of the form $f(a) = \sum_{j=1}^{m} \alpha_j k(a, a_j)$ with $m \in \mathbb{N}$, $\alpha_j \in \mathbb{R}$ and representing points $a_j \in \mathcal{A}$.

II. Related Work

The study of safe learning dates back to the beginning of this century [9]. In [10] and [11], Lyapunov-based reinforcement learning is used to allow a learning agent to safely switch between pre-computed baseline controllers. Then, in [12], risk-sensitive reinforcement learning is proposed, in which the expected return is heuristically weighted with the probability of reaching an error state. In several other papers, including [13], [14] and [15], safe exploration methods are utilized to allow the learning modules to achieve a desired balance between ensuring safe operation and exploring new states for improved performance. In [9], a general framework is proposed for ensuring safety of learning-based control strategies for uncertain robotic systems. In this framework, robust reachability guarantees from control theory are combined with Bayesian analysis based on empirical observations. The result is a safety-preserving, supervisory controller of the learning module that allows the system to freely execute its learning policy almost everywhere, but imposes control actions to ensure safety at critical states. Despite its effectiveness for ensuring safety, the supervisory controller in this approach has no role in reducing tracking errors.

Focusing our attention on safe, learning-based inverse dynamics control, we refer to [16], [17]. In these papers, Gaussian processes (GPs) are utilized to learn the errors in the output torques of the inverse dynamics model online. In [16], the GP learning is combined with a state-of-the-art gradient descent method for learning feedback terms online. The main idea behind this approach is that the offsets learnt online by the gradient descent method would correct for fast perturbations, while the GP is responsible for correcting slow perturbations. This allows for exponential smoothing of the GP hyperparameters, which increases the robustness of the GP at the cost of having slower reactiveness. Nevertheless, [16] does not provide a proof of the robust stability of the closed-loop system. In [17], the variance of the GP prediction is utilized to adapt the parameters of an outer-loop PD controller online, and the uniform ultimate boundedness of the tracking error is proved under some assumptions on the structure of the PD controller (e.g., the gain matrix was assumed to be diagonal, which imposes a decentralized gain control scheme). The results of [17] are verified via simulations on a 2 DOF manipulator.

Our approach differs from [17] in several aspects. First, we do not use an adaptive PD controller in the outer loop, but add a robustness term to the output of the outer-loop controller. Second, while [17] uses the GP to learn the error in the estimated torque from the nominal inverse dynamics, in our approach, we learn the error between the commanded and actual accelerations. This can be beneficial in two aspects: *(i)* This makes our approach applicable to industrial manipulators that have onboard controllers for calculating the torque and allow the user to only send commanded acceleration/velocity; *(ii)* this makes our approach also applicable beyond inverse dynamics control of manipulators; indeed, our proposed approach can be applied to any Lagrangian system for which feedforward/feedback linearization can be used to convert the nonlinear dynamics of the system to a set of decoupled double integrators, such as a quadrotor under a feedforward linearization, see Section 5.3 of [18]. Third, while [17] shows uniform ultimate boundedness of the tracking error, it does not provide discussions on the size of the ultimate ball. In this work, we show that using our proposed approach, the size of the ball can be made arbitrarily small through the control design. Fourth, in our approach, we do not impose any assumption on the structure of the outer-loop PD controller and decentralized, outer-loop control is not needed for our proof. Finally, we verify our approach using simulations on a 6 DOF manipulator.

III. Problem Statement

In this paper, we consider Lagrangian systems, which represent a wide class of mechanical systems [19]. In what follows, we focus our attention on a class of Lagrangian systems represented by:

$$M(q(t))\ddot{q}(t) + C(q(t), \dot{q}(t))\dot{q}(t) + g(q(t)) = u(t), \quad (1)$$

where $q = (q_1, \cdots, q_N)$ is the vector of generalized coordinates (displacements or angles), $\dot{q} = (\dot{q}_1, \cdots, \dot{q}_N)$ is the vector of generalized velocities, $u = (u_1, \cdots, u_N)$ is the vector of generalized forces (forces or torques), N is the system's degree of freedom, M, C, and g are matrices of proper dimensions and smooth functions, and $M(q)$ is a

Preprint submitted to 57th IEEE Conference on Decision and Control.
Received March 20, 2018.

positive definite matrix. Fully-actuated robotic manipulators are an example of Lagrangian systems that can be expressed by (1). Despite focusing our discussion on systems represented by (1), we emphasize that our results in this paper can be easily generalized to a wider class of nonlinear Lagrangian systems for which feedback/feedforward linearization can be utilized to convert the dynamics of the system into a set of decoupled double integrators, as we will discuss in detail in Remark 5.3.

For the nonlinear system (1) with uncertain matrices M, C, and g, we aim to make the system positions and velocities $(q(t), \dot{q}(t))$ track a desired smooth trajectory $(q_d(t), \dot{q}_d(t))$. For simplicity of notation, in our discussion, we drop the dependency on t from q, q_d, their derivatives, and u. Our goal is to design a novel, learning-based control strategy that is easy to interpret and implement, and that satisfies the following desired objectives:

(O1) *Robustness:* The overall, closed-loop control system satisfies robust stability in the sense that the tracking error has an upper bound under the system uncertainties.

(O2) *High-Accuracy Tracking:* For feasible desired trajectories, the tracking error converges to a ball around the origin that can be made arbitrarily small through the control design. For the ideal case, where the pre-assumed system parameters are correct, the tracking error should converge asymptotically to the origin.

(O3) *Adaptability:* The proposed strategy should incorporate online learning to continuously adapt to online changes of the system parameters and disturbances.

(O4) *Generalizability of the Approach:* The proposed approach should be general enough to be also applicable to industrial robots that have onboard controllers for calculating the forces/torques and allow the user to send only commanded acceleration/velocity. It is also desirable that the approach is general enough to be implementable on other Lagrangian systems beyond (1).

IV. Methodology

In this section, we present our proposed methodology, and then in the next section, we show that it satisfies objectives (O1)-(O4).

A standard approach for solving the tracking control problem for (1) is inverse dynamics control. In particular, if the matrices M, C, and g are all known, then the following inverse dynamics control law

$$u = C(q, \dot{q})\dot{q} + g(q) + M(q)a_q \tag{2}$$

converts the complex nonlinear dynamic system (1) into

$$\ddot{q} = a_q, \tag{3}$$

where a_q is the commanded acceleration, a new input to the linearized system (3) to be calculated by an outer-loop control law, e.g., a PD controller (see Figure 1). However, the standard inverse dynamics control (2) heavily depends on accurate knowledge of the system parameters. In practice, the matrices M, C, and g are not perfectly known, and consequently, one has to use estimated values of these matrices $\hat{M}$, $\hat{C}$, and $\hat{g}$, respectively, where $\hat{M}$, $\hat{C}$, and $\hat{g}$ are composed of smooth functions. Hence, in practice, the control law (2) should be replaced with

$$u = \hat{C}(q, \dot{q})\dot{q} + \hat{g}(q) + \hat{M}(q)a_q. \tag{4}$$

Now by plugging (4) into the system model (1), we get

$$\ddot{q} = a_q + \eta(q, \dot{q}, a_q), \tag{5}$$

where $\eta(q, \dot{q}, a_q) = M^{-1}(q)(\tilde{M}(q)a_q + \tilde{C}(q, \dot{q})\dot{q} + \tilde{g}(q))$, with $\tilde{M} = \hat{M} - M$, $\tilde{C} = \hat{C} - C$, and $\tilde{g} = \hat{g} - g$. Because of the term η, the dynamics (5) resulting from the inverse dynamics control are still nonlinear and coupled. To control the uncertain system (5), on the one hand, robust control methods are typically very conservative, while on the other hand, learning methods do not provide robust stability guarantees.

Hence, in this paper, we combine ideas from robust control theory with ideas from machine learning, particularly Gaussian processes (GPs) for regression, to provide a robust, learning-based control strategy that satisfies objectives (O1)-(O4). The main idea behind our proposed approach is to use GPs to learn the uncertainty vector $\eta(q, \dot{q}, a_q)$ in (5) online. Following [17], we use a set of N GPs, one for learning each element of η, to reduce the complexity of the regression. A main advantage of GP regression is that it does not only provide an estimated value of the mean μ, but it also provides an expected variance σ^2, which represents the accuracy of the regression model based on the distance to the training data. The punchline here is that one can use both the mean and variance of the GP to calculate an upper bound ρ on $\|\eta\|$ that is guaranteed to be correct with high probability, as we will show later in this section. One can then use this upper bound to design a robust, outer-loop controller that ensures robust stability of the overall system. Hence, our proposed strategy consists of three parts:

(i) Inner-loop controller: We use the inverse dynamics control law (4), where $\hat{M}$, $\hat{C}$, and $\hat{g}$ are estimated values of the system matrices from an a-priori model.

(ii) GPs for learning the uncertainty: We use a set of N GPs to learn the uncertainty vector η in (5). We start by reviewing GP regression [15], [20]. A GP is a nonparametric regression model that is used to approximate a nonlinear function $J(a) : \mathcal{A} \to \mathbb{R}$, where $a \in \mathcal{A}$ is the input vector. The ability of the GP to approximate the function is based on the assumptions that function values $J(a)$ associated with different values of a are random variables, and that any finite number of these variables have a joint Gaussian distribution. The GP predicts the value of the function, $J(a^*)$, at an arbitrary input $a^* \in \mathcal{A}$ from a set of n observations $D_n := \{a_j, \hat{J}(a_j)\}_{j=1}^n$, where $\hat{J}(a_j)$, $j \in \{1, \cdots, n\}$, are assumed to be noisy measurements of the function's true values. That is, $\hat{J}(a_j) = J(a_j) + \omega'$, where ω' is a zero mean Gaussian noise with variance σ_ω^2. Assuming, without loss of generality (w.l.o.g.), a zero prior mean of the GP and conditioned on the previous observations, the mean and variance of the GP prediction are given by:

$$\mu_n(a^*) = \mathbf{k}_n(a^*)(\mathbf{K}_n + \mathbf{I}_n\sigma_\omega^2)^{-1}\hat{\mathbf{J}}_n, \tag{6}$$

Preprint submitted to 57th IEEE Conference on Decision and Control.
Received March 20, 2018.

$$\sigma_n^2(a^*) = k(a^*, a^*) - \mathbf{k}_n(a^*)(\mathbf{K}_n + \mathbf{I}_n\sigma_\omega^2)^{-1}\mathbf{k}_n^T(a^*), \quad (7)$$

respectively, where $\hat{\mathbf{J}}_n = [\hat{J}(a_1), \cdots, \hat{J}(a_n)]^T$ is the vector of observed, noisy function values. The matrix $\mathbf{K}_n \in \mathbb{R}^{n\times n}$ is the covariance matrix with entries $[\mathbf{K}_n]_{(i,j)} = k(a_i, a_j)$, $i,\ j \in \{1, \cdots, n\}$, where $k(a_i, a_j)$ is the covariance function defining the covariance between two function values $J(a_i),\ J(a_j)$ (also called the kernel). The vector $\mathbf{k}_n(a^*) = [k(a^*, a_1), \cdots, k(a^*, a_n)]$ contains the covariances between the new input and the observed data points, and $\mathbf{I}_n \in \mathbb{R}^{n\times n}$ is the identity matrix. The tuning of the GP is typically done through the selection of the kernel function and the tuning of its hyperparameters. For information about different standard kernel functions, please refer to [20].

We next discuss our implementation of the GPs. The GPs run in discrete time with sampling interval T_s. The inputs to each GP regression model are the same $(q_k, \dot{q}_k, a_{q_k})$, and the output is an estimated value of an element of the η vector at the sampling instant k. For the training data for each GP, n observations of $(q, \dot{q}, a_q)$ are used as the labeled input together with n observations of an element of the vector $\ddot{q} - a_q + \omega$ as the labeled output, where $\omega \in \mathbb{R}^N$ is Gaussian noise with zero mean and variance $diag(\sigma_{\omega_1}^2, \cdots, \sigma_{\omega_N}^2)$; see (5). In this paper, we use the squared exponential kernel

$$k(a_i, a_j) = \sigma_\eta^2 e^{-\frac{1}{2}(a_i - a_j)^T \bar{M}^{-2}(a_i - a_j)}, \quad (8)$$

which is parameterized by the hyperparameters: σ_η^2, the prior variance, and the positive length scales $l_1, \cdots, l_{3N}$ which are the diagonal elements of the diagonal matrix $\bar{M}$. Hence, the expected mean and variance of each GP can be obtained through equations (6)-(8). Guidelines for tuning the GP hyperparameters $\sigma_\omega^2, \sigma_\eta^2, l_1, \cdots, l_{3N}$ can be found in [15].

As stated before, a main advantage of GP regression is that the GP provides a variance, which represents the accuracy of the regression model based on the distance between the new input and the training data. One can then use the predicted mean and variance of the GP to provide a confidence interval around the mean that is guaranteed to be correct with high probability. There are several comprehensive studies in the machine learning literature on calculating these confidence intervals. For completeness, we review one of these results, particularly Theorem 6 of [20]. Let $a_{aug} = (q, \dot{q}, a_q)$, and $\eta_i(a_{aug})$ denote the i-th element of the unknown vector function η.

Assumption 4.1: The function $\eta_i(a_{aug})$, $i \in \{1, \cdots, N\}$, has a bounded RKHS norm $\|\eta_i\|_k$ with respect to the covariance function $k(a, a')$ of the GP, and the noise ω_i added to the output observations, $i \in \{1, \cdots, N\}$, is uniformly bounded by $\bar{\sigma}$.

The boundedness of the RKHS norm implies that the functions are well-behaved in the sense that they are regular with respect to the kernel [20].

Lemma 4.1 (Theorem 6 of [20]): Suppose that Assumption 4.1 holds. Let $\delta_p \in (0, 1)$. Then, $\Pr\{\forall a_{aug} \in \mathcal{A}, \|\mu(a_{aug}) - \eta_i(a_{aug})\| \leq \beta^{1/2}\sigma(a_{aug})\} \geq 1 - \delta_p$, where Pr stands for the probability, $\mathcal{A} \subset \mathbb{R}^{3N}$ is compact, $\mu,\ \sigma^2$ are the GP mean and variance evaluated at a_{aug} conditioned on n past observations, and $\beta = 2\|\eta_i\|_k^2 + 300\gamma \ln^3((n+1)/\delta_p)$. The variable $\gamma \in \mathbb{R}$ is the maximum information gain and is given by $\gamma = \max_{\{a_{aug,1}, \cdots, a_{aug,n+1}\} \in \mathcal{A}} 0.5\ \log(\det(I + \bar{\sigma}^{-2}\mathbf{K}_{n+1}))$, where det is the matrix determinant, $I \in \mathbb{R}^{(n+1)\times(n+1)}$ is the identity matrix, $\mathbf{K}_{n+1} \in \mathbb{R}^{(n+1)\times(n+1)}$ is the covariance matrix given by $[\mathbf{K}_{n+1}]_{(i,j)} = k(a_{aug,i}, a_{aug,j})$, $i,\ j \in \{1, \cdots, n+1\}$.

The punchline here is that we know from Lemma 4.1 that one can define for each GP a confidence interval around the mean that is guaranteed to be correct for all points $a_{aug} \in \mathcal{A}$, a compact set, with probability higher than $(1 - \delta_p)$, where δ_p is typically picked very small. Let $\mu_{k,i}$, $\sigma_{k,i}^2$, and $\beta_{k,i}$ represent the expected mean, variance, and β parameter in Lemma 4.1 of the i-th GP at the sampling instant k, respectively, where $i \in \{1, \cdots, N\}$. We select the upper bound on the absolute value of η_i at k to be

$$\rho_{k,i}(\mu_{k,i}, \sigma_{k,i}) = \max(|\mu_{k,i} - \beta_{k,i}^{1/2}\sigma_{k,i}|, |\mu_{k,i} + \beta_{k,i}^{1/2}\sigma_{k,i}|). \quad (9)$$

Then, a good estimate of the upper bound on $\|\eta\|$ at k is

$$\rho_k = \sqrt{\rho_{k,1}(\mu_{k,1}, \sigma_{k,1})^2 + \cdots + \rho_{k,N}(\mu_{k,N}, \sigma_{k,N})^2}. \quad (10)$$

(iii) Robust, outer-loop controller: We use the estimated upper bound ρ_k to design a robust, outer-loop controller. In particular, for a smooth, bounded desired trajectory $q_d(t)$, we use the outer-loop control law

$$a_q(t) = \ddot{q}_d(t) + K_P(q_d(t) - q(t)) + K_D(\dot{q}_d(t) - \dot{q}(t)) + r(t), \quad (11)$$

where $K_P \in \mathbb{R}^{N\times N}$ and $K_D \in \mathbb{R}^{N\times N}$ are the proportional and derivative matrices of the PD control law, respectively, and $r \in \mathbb{R}^N$ is an added vector to the PD control law that will be designed to achieve robustness. Let $e(t) := (q(t) - q_d(t), \dot{q}(t) - \dot{q}_d(t))$ denote the tracking error vector. From (11) and (5), it can be shown that the tracking error dynamics are

$$\dot{e}(t) = Ae(t) + B(r(t) + \eta(q(t), q_d(t), a_q(t))), \quad (12)$$

where

$$A = \begin{bmatrix} 0 & I \\ -K_P & -K_D \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ I \end{bmatrix}, \quad (13)$$

and $I \in \mathbb{R}^{N\times N}$ is the identity matrix. From (12) and (13), it is clear that the controller matrices K_P and K_D should be designed to make A a Hurwitz matrix.

We now discuss how to design the robustness vector $r(t)$. To that end, let P be the unique positive definite matrix satisfying $A^T P + PA = -Q$, where Q is a positive definite matrix. We define $r(t)$ as follows

$$r(t) = \begin{cases} -\rho(t)\frac{B^T P e(t)}{\|B^T P e(t)\|} & \|B^T P e(t)\| > \epsilon, \\ -\rho(t)\frac{B^T P e(t)}{\epsilon} & \|B^T P e(t)\| \leq \epsilon, \end{cases} \quad (14)$$

where $\rho(t)$ is the last received upper bound on $\|\eta\|$ from the GPs, i.e., we use

$$\rho(t) = \rho_k, \forall t \in [kT_s, (k+1)T_s], \quad (15)$$

Preprint submitted to 57th IEEE Conference on Decision and Control.
Received March 20, 2018.

and ϵ is a small positive number. It should be noted that ϵ is a design parameter that can be selected to ensure high-accuracy tracking, as we will discuss in the next section.

V. Theoretical Guarantees

After discussing the proposed control strategy, we now justify that it satisfies the desired objectives (O1)-(O4). To that end, we require the following reasonable assumptions:

Assumption 5.1: The uncertainty vector $\eta(q, \dot{q}, a_q)$ in (5) can be modeled by N independent GPs.

Assumption 5.2: The GPs run at a sufficiently fast sampling rate such that the calculated upper bound on $\|\eta\|$ is accurate between two consecutive sampling instants.

Next, we impose another assumption to ensure that the added robustness vector $r(t)$ will not cause the uncertainty vector norm $\|\eta(q(t), \dot{q}(t), a_q(t))\|$ to blow up. It is easy to show that the uncertainty function $\eta(q, \dot{q}, a_q)$ is smooth, and so $\|\eta\|$ attains a maximum value on any compact set in its input space $(q, \dot{q}, a_q)$. However, since from (11) and (14), a_q is a function of $\rho(t)$, an upper bound on $\|\eta\|$, one still needs to ensure the boundedness of $\|\eta\|$ for bounded $q,\ \dot{q}$ or bounded tracking error e. Hence, we present the following assumption.

Assumption 5.3: For a given, smooth, bounded desired trajectory $(q_d(t), \dot{q}_d(t))$, there exists $\bar{\rho} > 0$ such that $\|\eta\| < \bar{\rho}$ for each $e \in D$, where D is a compact set containing $\{e \in \mathbb{R}^{2N} : e^T P e \leq e(0)^T P e(0)\}$, and $e(0)$ is the initial tracking error.

We now justify that Assumption 5.3 is reasonable, and that it is automatically satisfied for small uncertainties in the system matrices, particularly in the inertia matrix $M(q)$. In this discussion, we suppose that $\rho(t)$ in (14) satisfies $\rho(t) \leq \|\eta(t)\| + c$, where c is a positive scalar, and study whether imposing $r(t)$ into (11) can make $\|\eta(t)\|$ blow up. Recall that $\eta(q, \dot{q}, a_q) = M^{-1}(q)(\tilde{M}(q)a_q + \tilde{C}(q, \dot{q})\dot{q} + \tilde{g}(q))$. From (11), we have $\eta(q, \dot{q}, a_q) = M^{-1}(q)(\tilde{M}(q)r + \tilde{M}(\ddot{q}_d + K_P(q_d - q) + K_D(\dot{q}_d - \dot{q})) + \tilde{C}(q, \dot{q})\dot{q} + \tilde{g}(q))$. It is evident that $\|\eta\| \leq \|M^{-1}(q)\tilde{M}(q)\|\|r\| + \|M(\ddot{q}_d + K_P(q_d - q) + K_D(\dot{q}_d - \dot{q})) + \tilde{C}(q, \dot{q})\dot{q} + \tilde{g}(q))\|$. From (14), it is easy to verify $\|r(t)\| \leq \|\rho(t)\| = \rho(t)$, and so $\|r(t)\| \leq (\|\eta(t)\| + c)$. Hence, $\|\eta\| \leq \|M^{-1}(q)\tilde{M}(q)\|\|\eta\| + \|M^{-1}(q)\tilde{M}(q)\|c + \|\tilde{M}(\ddot{q}_d + K_P(q_d - q) + K_D(\dot{q}_d - \dot{q})) + \tilde{C}(q, \dot{q})\dot{q} + \tilde{g}(q))\|$. Now if the uncertainty in the matrix $M(q)$, $\tilde{M}(q)$, is sufficiently small such that $\|M^{-1}(q)\tilde{M}(q)\| < 1$ is satisfied, then we have $\|\eta\| \leq \frac{1}{1 - \|M^{-1}(q)\tilde{M}(q)\|}(\|M^{-1}(q)\tilde{M}(q)\|c + \|\tilde{M}(\ddot{q}_d + K_P(q_d - q) + K_D(\dot{q}_d - \dot{q})) + \tilde{C}(q, \dot{q})\dot{q} + \tilde{g}(q))\|)$. Since $q_d,\ \dot{q}_d,\ \ddot{q}_d$ are all bounded by assumption, if $e(t) \in D$, a compact set, then $q(t),\ \dot{q}(t)$ are also bounded. Then, it is straightforward to show that there exists a fixed upper bound $\bar{\rho}$ on $\|\eta\|$ that is valid for each $e \in D$, and Assumption 5.3 is satisfied.

From Assumption 5.3 and our discussion in the previous paragraph, we know that $\|\eta(t)\| \leq \bar{\rho}$ if $e(t) \in D$, and consequently, it is reasonable to saturate any estimate of $\rho(t)$ beyond $\bar{\rho}$. Hence, we suppose that the estimation of ρ is slightly modified to be

$$\rho(t) = \min(\rho_{GP}(t), \bar{\rho}), \tag{16}$$

where $\rho_{GP}(t)$ is the upper bound on the uncertainty norm, $\|\eta\|$, calculated from the GPs in (9), (10), and (15). It is straightforward to show that with the choice of $\rho(t)$ in (16) and for bounded smooth trajectories, the condition $e(t) \in D$ for all $t \geq 0$ implies that $a_q(t)$ in (11) is always bounded, and so $a_{aug} = (q, \dot{q}, a_q)$ always lies in a compact set. To be able to provide theoretical guarantees, we also assume w.l.o.g. that the small positive number ϵ in (14) is selected sufficiently small such that

$$\sqrt{\frac{\epsilon\bar{\rho}}{2\lambda_{min}(Q)}} \ll \delta_1, \tag{17}$$

where $\delta_1 > 0$ is such that $B_{\delta_1}(0) \subset \{e \in \mathbb{R}^{2N} : e^T P e < e(0)^T P e(0)\}$, and $\lambda_{min}(Q) > 0$ is the smallest eigenvalue of the positive definite matrix Q.

Based on Assumptions 4.1 and 5.1-5.3, we provide the following main result.

Theorem 5.1: Consider the Lagrangian System (1) and a smooth, bounded desired trajectory $(q_d(t), \dot{q}_d(t))$. Suppose that Assumptions 4.1, 5.1, 5.2, and 5.3 hold. Then, the proposed, robust, learning-based control strategy in (4), (11), and (14), with the uncertainty upper bound ρ calculated by (16) and the design parameter ϵ satisfying (17), ensures with high probability of at least $(1 - \delta_p)^N$ that the tracking error $e(t)$ is uniformly ultimately bounded with an ultimate bound that can be made arbitrarily small through the selection of the design parameter ϵ.

Proof: From Assumption 5.3, we know that $\|\eta(t)\| \leq \bar{\rho}$ when $e(t) \in D$, where D is a compact set containing $\{e \in \mathbb{R}^{2N} : e^T P e \leq e(0)^T P e(0)\}$. In the first part of the proof, we assume that the upper bound $\rho_{GP}(t)$ calculated by (9), (10) and (15) is a correct upper bound on $\|\eta(t)\|$ when $e(t) \in D$. Thus, in the first part of the proof, we know that $\rho(t)$ calculated by (16) is a correct upper bound on $\|\eta(t)\|$ when $e(t) \in D$, and we use Lyapunov stability analysis to prove that $e(t)$ is uniformly ultimately bounded. Then, in the second part of the proof, we use Lemma 4.1 to evaluate the probability of satisfying the assumption that $\rho_{GP}(t)$ is a correct upper bound on $\|\eta(t)\|$ when $e(t) \in D$, and hence, the probability that the provided guarantees hold.

The first part of the proof closely follows the proof of the effectiveness of the robust controller in Theorem 3 of Chapter 8 of [1], and we include the main steps of the proof here for convenience. Consider a candidate Lyapunov function $V(e) = e^T P e$. From (12), it can be shown that $\dot{V} = -e^T Q e + 2w^T(\eta + r)$, where $w = B^T P e$. Then, from (14), we need to study two cases.

For the case where $\|w\| > \epsilon$, we have $w^T(\eta + r) = w^T(\eta - \rho\frac{w}{\|w\|}) \leq -\rho\|w\| + \|\eta\|\|w\|$ from the Cauchy-Schwartz inequality. Since $\{e \in \mathbb{R}^{2N} : e^T P e \leq e(0)^T P e(0)\} \subset D$ by definition and from Assumption 5.3, we know that $\|\eta\| \leq \bar{\rho}$. Also, by our assumption in this part of the proof, $\|\eta\| \leq \rho_{GP}$. Then, from (16), $\|\eta\| \leq \rho$, and $w^T(\eta + r) \leq 0$. Thus, for

Preprint submitted to 57th IEEE Conference on Decision and Control.
Received March 20, 2018.

this case, $\dot{V} \leq -e^T Qe$, which ensures exponential decrease of the Lyapunov function.

Next, consider the case where $\|w\| \leq \epsilon$. If $w = 0$, then $\dot{V} = -e^T Qe < 0$. Then, for $\|w\| \leq \epsilon$ and $w \neq 0$, it is easy to show $\dot{V} = -e^T Qe + 2w^T(\eta + r) \leq -e^T Qe + 2w^T(\rho \frac{w}{\|w\|} + r)$. From (14), we have $\dot{V} \leq -e^T Qe + 2w^T(\rho \frac{w}{\|w\|} - \rho \frac{w}{\epsilon})$. It can be shown that the term $2w^T(\rho \frac{w}{\|w\|} - \rho \frac{w}{\epsilon})$ has a maximum value of $(\epsilon\rho)/2$ when $\|w\| = \epsilon/2$. Thus, $\dot{V} \leq -e^T Qe + (\epsilon\rho)/2$. From (16), $\rho \leq \bar{\rho}$, and consequently $\dot{V} \leq -e^T Qe + (\epsilon\bar{\rho})/2$. If the condition $e^T Qe > (\epsilon\bar{\rho})/2$ is satisfied, then $\dot{V} < 0$. Since Q is positive definite by definition, then $e^T Qe \geq \lambda_{min}(Q)\|e\|^2$, where $\lambda_{min}(Q) > 0$ is the smallest eigenvalue of Q. Hence, if $\lambda_{min}(Q)\|e\|^2 > (\epsilon\bar{\rho})/2$, then $\dot{V} < 0$. Thus, the Lyapunov function is strictly decreasing if $\|e\| > \sqrt{\frac{\epsilon\bar{\rho}}{2\lambda_{min}(Q)}}$. Let B_δ be the ball around the origin of radius $\delta := \sqrt{\frac{\epsilon\bar{\rho}}{2\lambda_{min}(Q)}}$, S_δ be a sufficiently small sublevel set of the Lyapunov function V satisfying $\bar{B}_\delta \subset S_\delta^\circ$, and B_c be the smallest ball around the origin satisfying $S_\delta \subset \bar{B}_c$. Since the Lyapunov function V is strictly decreasing outside $\bar{B}_\delta$, the tracking error $e(t)$ eventually reaches and remains in $S_\delta \subset \bar{B}_c$, and so the tracking error $e(t)$ is uniformly ultimately bounded, and its ultimate bound is the radius of B_c. Note that from (17), $B_\delta \subset \{e \in \mathbb{R}^{2N} : e^T Pe < e(0)^T Pe(0)\} \subset D$, and ρ is a correct upper bound on $\|\eta\|$. One can see that δ and hence the radius of B_c depend on the choice of the design parameter ϵ. Indeed, ϵ can be selected sufficiently small to make B_δ and B_c arbitrarily small.

In the second part of the proof, we calculate the probability of our assumption in the first part that $\rho_{GP}(t)$ is a correct upper bound on $\|\eta(t)\|$ when $e(t) \in D$. Recall that $e(t) \in D$ implies that $a_{aug}(t)$ is in a compact set, as discussed immediately after (16). From Assumption 5.2, our problem reduces to calculating the probability that ρ_{GP} is a correct upper bound on $\|\eta\|$ for all the sampling instants. Using the confidence region proposed in Lemma 4.1 for calculating the upper bound on the absolute value of each element of η, and under Assumption 4.1, the probability that this upper bound is correct for all samples is higher than $(1 - \delta_p)$ from Lemma 4.1. Since the N GPs are independent and the added noise to the output observations ω is uncorrelated, then the probability that the upper bounds on the absolute values of all the elments of η, and hence the upper bound on $\|\eta(t)\|$, are correct is higher than $(1 - \delta_p)^N$. ■

Remark 5.1: Although in practice it is difficult to estimate the upper bound $\bar{\rho}$ on $\|\eta\|$ used in (16), one can be conservative in this choice. Then, unlike robust control techniques that keep this conservative bound unchanged, (16) would relax the upper bound $\bar{\rho}$ when the GPs learn a less-conservative (lower) upper bound from collected data. Having a less-conservative upper bound ρ will result in a lower tracking error. Indeed, it can be shown that if $\rho(t) \leq \rho' < \bar{\rho}$ for all t, then the tracking error will converge to an ultimate ball $B_{c'}$ smaller than B_c.

Remark 5.2: While in theory ϵ can be selected sufficiently small to ensure arbitrarily accurate tracking as shown in the proof of Theorem 5.1, in practice, having a smaller ϵ will result in a higher $r(t)$ from (14), a higher $a_q(t)$, and a higher control action u. Thus, achieving arbitrarily accurate tracking may be limited by the actuation limits of the robots, which is a typical scenario in practice. Incorporating the actuation limits in the theoretical analysis is an interesting point for future research.

Remark 5.3: One advantage of the proposed approach is that it is generalizable in two different ways. First, since the GPs in our approach learn the error between the commanded and actual acceleration, and not the error in the force/torque, and since the GPs do not directly alter the onboard controller, our approach can be also applied to a wide class of industrial manipulators that have onboard controllers computing the joint torques and allow the user to send the commanded joint acceleration/velocity. Second, the idea of our approach can be easily generalized to any class of Lagrangian systems that can be expressed, possibly after a feedforward/feedback linearization, as a set of decoupled double integrators with an added uncertainty vector. The GPs can be used to learn the uncertainty vector resulting from the linearization. For instance, as shown in Section 5.3 of [18], the nonlinear dynamics of a quadrotor moving in the x, y plane at a fixed height can be converted using feedforward linearization into two decoupled double integrators. Thus, we can again assume $\ddot{q}(t) = a_q(t) + \eta(x(t), a_q(t))$, where η represents the uncertainty in the linearization, and x is the quadrotor state vector. GPs can be used to learn η, and an upper bound on $\|\eta\|$ can be then used to design a robust, outer-loop controller for calculating a_q. The details are beyond the scope of this paper.

VI. Simulation Results

In this section, we present several simulation results to demonstrate the effectiveness of the proposed approach. In particular, simulations of the proposed approach were conducted on a 2 DOF, planar manipulator using MATLAB Simulink as well as a 6 DOF, industrial manipulator using the standard Gazebo simulator. It should be noted that in the simulations, we follow a practical approach in which we always use a confidence interval $[\mu_k - 3\sigma_k, \mu_k + 3\sigma_k]$, which is simpler to implement and found to be effective in practice [15]. We also set the upper bound $\bar{\rho}$ in (16) to be a very high positive number. Hence, the simulation results mainly evaluate the effectiveness of the upper bound estimated by the GPs.

A. 2 DOF Planar Manipulator

In the 2 DOF simulations, we use the robot dynamics (1) for the system, where M, C, and g are as defined in Chapter 7 of [1]. For the system parameters, a value of $1\ \mathrm{kg}$ is used for each link mass and $1\ \mathrm{kg \cdot m^2}$ for each link inertia. The length of the first link is $2\ \mathrm{m}$ and that of the second link is $1\ \mathrm{m}$. The joints are assumed to have no mass and are not affected by friction. Then, it is assumed that these parameters are not perfectly known. Thus, in the inverse dynamics controller (4), we use parameters with different

Preprint submitted to 57th IEEE Conference on Decision and Control.
Received March 20, 2018.

TABLE I
AVERAGE RMS TRACKING ERROR (IN RAD) OVER 12 TRAJECTORIES FOR DIFFERENT CONTROLLERS ON A 2 DOF MANIPULATOR

Uncertainty	Nominal	Fixed Robust	Robust Learning
10%	0.1557	0.0489	0.0065
20%	0.2807	0.0512	0.0101
30%	0.3793	0.0534	0.0148

levels of uncertainties, and for each uncertainty case, evaluate our proposed control strategy. The desired trajectories are sinusoidal trajectories with different amplitudes and frequencies. All the simulation runs are initialized at zero initial conditions.

Since the linearization by (4) is not perfect due to uncertainties, we use 2 GPs to learn the uncertainty vector η in (5). Each GP uses the squared exponential kernel, and is parameterized with $\sigma_{\eta,i} = 1$, $\sigma_{\omega,i} = 0.001$, and $l_{j,i} = 0.5$, for all $j \in \{1, \cdots, 6\}$ and $i \in \{1, 2\}$. The Gaussian processes use the past $n = 20$ observations for prediction. For the robust controller, ϵ is set to 0.001.

A sequence of *12 trajectories* was run for *3 different cases of model uncertainty*. Each of the three cases makes the $\hat{M}$ matrix differ from the M matrix by using values for the estimated link masses that differ from the true link mass values. In particular, in the three uncertainty cases, the estimated mass differs from the actual mass by 10%, 20%, and 30% for each link, respectively.

The tracking performance was compared between three controllers: a nominal controller with no robust control, a robust controller with a fixed upper bound on the uncertainty norm $\rho = 1000$, and our proposed, robust, learning-based controller. The root-mean-square (RMS) error of the joint angles was averaged over the 12 trajectories, and is presented for each controller and uncertainty case in Table I.

It is clear that while the robust controller with a high, fixed value for the upper bound on the uncertainty improves the tracking performance compared to the nominal controller, it is conservative, and thus, still causes considerable tracking errors. The tracking errors are significantly reduced by our proposed, robust, learning-based controller, which is able to learn a less conservative upper bound on the uncertainty norm. On average, our proposed controller reduces the tracking errors by 96.21% compared to the nominal controller and by 79.84% compared to the fixed, robust controller.

A comparison of the tracking error for each controller over a single desired trajectory is shown in Figure 2. The brief period at the beginning of the trajectory shows the learning period of the robust learning controller. The fixed robust controller outperforms the robust learning controller until the latter's Gaussian processes have observed enough data to make more accurate predictions.

B. 6 DOF Industrial Manipulator

The proposed approach is further tested on a 6 DOF industrial manipulator using the Robot Operating System (ROS) and Gazebo simulator. The model parameters are selected based on the UR10 robot manipulator's open-source Gazebo package in [21]. The saturation limits are set to sufficiently high values to evaluate the system performance when the desired trajectories are feasible to be perfectly tracked under the actuation limits.

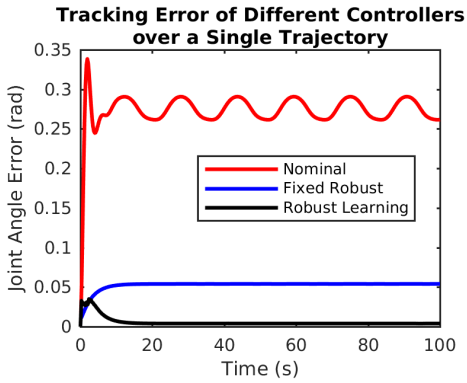

Fig. 2. The joint angle error over time for each controller for tracking a single trajectory $q_d(t) = (\frac{\pi}{18}\sin(0.4t), \frac{\pi}{18}\sin(0.4t))$ with 30% model uncertainty in the link masses.

TABLE II
AVERAGE RMS TRACKING ERROR (IN RAD) OVER 6 TRAJECTORIES ON A 6 DOF MANIPULATOR

Uncertainty	Nominal	Robust Learning	Improvement
30%	0.0275	0.0139	58.05%
60%	0.0434	0.0218	49.66%
90%	0.0539	0.0304	43.52%
Average	0.0416	0.0221	50.41%

Similar to our discussion in the previous subsection, we use in this example 6 GPs to learn the uncertainty vector η. Each GP uses the squared exponential kernel, and is parameterized with $\sigma_{\eta,i} = 8.0$, $\sigma_{\omega,i} = 0.001$, and $l_{j,i} = 0.05$, for all $j \in \{1, \cdots, 18\}$ and $i \in \{1, \cdots, 6\}$. Also, each GP uses the past $n = 20$ observations for prediction, and runs at a sampling rate of 125 Hz. For the robust controller, the parameter ϵ is set to 5×10^{-6}.

The performance of the robust learning controller is compared with that of the nominal, non-robust controller on *6 different trajectories and 3 different, large model uncertainties*. Model uncertainties were introduced in the same way as in the 2 DOF robot simulation; that is, the estimated link masses used in the inverse dynamics controller (4) were changed from the true values used in the simulation's physics environment. The three different uncertainty cases use estimated link masses that differ from the true link masses by 30%, 60%, and 90%, respectively. The results of the comparison are summarized in Table II.

It is clear that our proposed controller consistently reduces the tracking error in the presence of model uncertainties. The performance of each controller over all six trajectories with a fixed model uncertainty in the link masses of 60% is presented in Table III. This table shows how our proposed controller consistently provides a significant performance improvement over the nominal controller across a variety

Preprint submitted to 57th IEEE Conference on Decision and Control.
Received March 20, 2018.

TABLE III
RMS TRACKING ERROR (IN RAD) FOR 60% MODEL UNCERTAINTY AND DIFFERENT TRAJECTORIES

Trajectory	Nominal	Robust Learning	Improvement
#1	0.0389	0.0204	47.50%
#2	0.0416	0.0210	49.54%
#3	0.0450	0.0211	53.10%
#4	0.0466	0.0235	49.49%
#5	0.0444	0.0223	49.77%
#6	0.0437	0.0225	48.58%
Average	0.0434	0.0218	49.66%

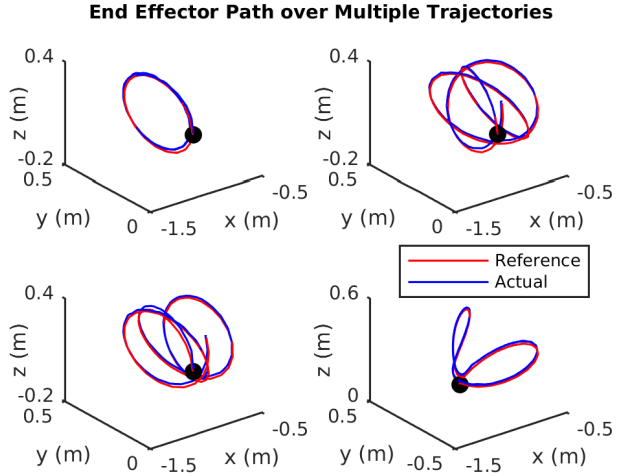

Fig. 3. The end effector position path for Trajectory #1, #2, #3, and #4 in Table III, respectively, under 60% model uncertainty in the masses and our proposed robust, learning-based controller. The black dots represent the initial positions of the end effector.

of trajectories.

The 3-dimensional paths followed by the manipulator's end effector under 60% model uncertainty and our proposed controller for 4 different desired trajectories are shown in Figure 3. As can be seen from the figure, despite the large uncertainty, the proposed controller makes the end effector follow the desired path very closely.

VII. CONCLUSIONS

For an important class of Lagrangian systems, we have provided a novel, learning-based control strategy based on Gaussian processes (GPs) that ensures stability of the closed-loop system and high-accuracy tracking of smooth trajectories. The main idea is to use GPs to estimate an upper bound on the uncertainty of the linearized system, and then use the uncertainty bound in a robust, outer-loop controller. Unlike most of the existing, learning-based inverse dynamics control techniques, we have provided a proof of the closed-loop stability of the system that takes into consideration the regression errors of the learning modules. Moreover, we have proved that the tracking error converges to a ball with a radius that can be made arbitrarily small through appropriate control design. Furthermore, we have verified the effectiveness of our approach via simulations on planar manipulators and 6 DOF industrial manipulators using the standard Gazebo simulator. A promising direction for future research is to formally incorporate the saturation limits of the system into the theoretical analysis of the proposed control system behavior.

REFERENCES

[1] M. W. Spong, S. Hutchinson, and M. Vidyasagar. *Robot Modeling and Control*. John Wiley & Sons, Inc., 2006.
[2] V. Hagenmeyer, and E. Delaleau. Exact feedforward linearization based on differential flatness. *International Journal of Control*, 76(6), pp. 537-556, 2003.
[3] C. Abdallah, D. M. Dawson, P. Dorato, and M. Jamshidi. Survey of robust control for rigid robots. *IEEE Control Systems Magazine*, 11(2), pp. 24-30, 1991.
[4] J. Sun de la Cruz. *Learning Inverse Dynamics for Robot Manipulator Control*. M.A.Sc. Thesis, University of Waterloo, 2011.
[5] R. Calandra, S. Ivaldi, M. P. Deisenroth, E. Rueckert, and J. Peters. Learning inverse dynamics models with contacts. *IEEE International Conference on Robotics and Automation*, Seattle, 2015, pp. 3186-3191.
[6] A. S. Polydoros, L. Nalpantidis, and V. Kruger. Real-time deep learning of robotic manipulator inverse dynamics. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Hamburg, 2015, pp. 3442-3448.
[7] S. Zhou, M. K. Helwa, and A. P. Schoellig. Design of deep neural networks as add-on blocks for improving impromptu trajectory tracking. *IEEE Conference on Decision and Control*, Melbourne, 2017, pp. 5201-5207.
[8] D. Nguyen-Tuong, J. Peters. Using model knowledge for learning inverse dynamics. *IEEE International Conference on Robotics and Automation*, Anchorage, 2010, pp. 2677-2682.
[9] J. F. Fisac, A. K. Akametalu, M. N. Zeilinger, S. Kaynama, J. Gillula, and C. J. Tomlin. A general safety framework for learning-based control in uncertain robotic systems. *arXiv preprint arXiv:1705.01292*, 2017.
[10] T. J. Perkins, and A. G. Barto. Lyapunov design for safe reinforcement learning. *The Journal of Machine Learning Research*, 3, pp. 803-832, 2003.
[11] J. W. Roberts, I. R. Manchester, and R. Tedrake. Feedback controller parameterizations for reinforcement learning. *IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*, Paris, 2011, pp. 310-317.
[12] P. Geibel, and F. Wysotzki. Risk-sensitive reinforcement learning applied to control under constraints. *Journal of Artificial Intelligence Research*, 24, pp. 81-108, 2005.
[13] M. Turchetta, F. Berkenkamp, and A. Krause. Safe exploration in finite markov decision processes with Gaussian processes. *Advances in Neural Information Processing Systems*, Barcelona, 2016, pp. 4312-4320.
[14] F. Berkenkamp, R. Moriconi, A. P. Schoellig, A. Krause. Safe learning of regions of attraction for uncertain, nonlinear systems with Gaussian processes. *IEEE Conference on Decision and Control*, Las Vegas, 2016, pp. 4661-4666.
[15] F. Berkenkamp, A. P. Schoellig, and A. Krause. Safe controller optimization for quadrotors with Gaussian processes. *IEEE International Conference on Robotics and Automation*, Stockholm, 2016, pp. 491-496.
[16] F. Meier, D. Kappler, N. Ratliff, and S. Schaal. Towards robust online inverse dynamics learning. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Daejeon, 2016, pp. 4034-4039.
[17] T. Beckers, J. Umlauft, D. Kulić, and S. Hirche. Stable Gaussian process based tracking control of Lagrangian systems. *IEEE Conference on Decision and Control*, Melbourne, 2017, pp. 5580-5585.
[18] M. K. Helwa, A. P. Schoellig. On the construction of safe controllable regions for affine systems with applications to robotics. *arXiv preprint arXiv:1610.01243*, 2016.
[19] R. M. Murray. Nonlinear control of mechanical systems: a Lagrangian perspective. *Annual Reviews in Control*, 21, pp. 31-42, 1997.
[20] N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger. Information-theoretic regret bounds for Gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory*, 58(5), pp. 3250-3265, 2012.
[21] universal_robot package, ver. 1.1.9, link: wiki.ros.org/universal_robot, 2014.

Preprint submitted to 57th IEEE Conference on Decision and Control.
Received March 20, 2018.