

Stable Gaussian Process based Tracking Control of Lagrangian Systems

Thomas Beckers¹, Jonas Umlauf¹, Dana Kulić² and Sandra Hirche¹

Abstract—High performance tracking control can only be achieved if a good model of the dynamics is available. However, such a model is often difficult to obtain from first order physics only. In this paper, we develop a data-driven control law that ensures closed loop stability of Lagrangian systems. For this purpose, we use Gaussian Process regression for the feed-forward compensation of the unknown dynamics of the system. The gains of the feedback part are adapted based on the uncertainty of the learned model. Thus, the feedback gains are kept low as long as the learned model describes the true system sufficiently precisely. We show how to select a suitable gain adaption law that incorporates the uncertainty of the model to guarantee a globally bounded tracking error. A simulation with a robot manipulator demonstrates the efficacy of the proposed control law.

I. INTRODUCTION

Lagrangian systems represent a very important and large class of dynamical systems, for which the equations of motion can be derived via the Euler-Lagrange equation. In the past decades, various control schemes for this class have been proposed, most of them can be considered as a subset of computed torque control laws. With computed torque control, it is possible to derive very effective controllers within robust, adaptive and learning control schemes [1]. The controller is separated into a feed-forward and a feedback part. An exact model of the system is necessary for the feed-forward part to compensate the dynamics to achieve a low gain feedback term. This is beneficial in several ways: it favors disturbance attenuation in the presence of noise, avoids the saturation of the actuators, and allows safe interaction [2]. Since the accuracy of the compensation depends on the precision of the model, all generalized external forces must be incorporated as precisely as possible [3]. However, an accurate model of these uncertainties is hard to obtain by classical first principles based techniques. Especially in modern applications of Lagrangian systems such as service robotics, the interaction with unstructured and a priori unknown environments further increases the uncertainty.

A suitable approach to avoid the time-consuming or even infeasible physical modeling is provided by Gaussian Process regression (GPR) which is a promising, data-driven learning approach [4]. GPR is a supervised learning technique which combines several advantages: It requires only a minimum of prior knowledge for arbitrary complex functions, generalizes well even for small training data

sets and has a precise trade-off between fitting the data and smoothing [5]. In addition, GPR provides not only a mean function but also a predicted variance, and therefore a measure of the model fidelity based on the distance to the training data. This is a significant benefit also for the feedback part of the control law since the model fidelity can be used to adapt the feedback gains to keep the gains as low as possible. For this purpose, the gains are kept low in state space regions with high model accuracy and increased otherwise.

In [6] an online adaptation law for the control of robotic manipulators is presented so that the tracking error converges to zero. However, this approach is based on an underlying parametric model and is limited to dynamics which are linear in terms of a suitably selected set of parameters. In [7] a stable feedback linearization with online learned GPR is proposed, but without adapting the feedback gains. The authors of [8] present a hybrid learning approach which incorporates model knowledge. The classical adaptive control approach with varying feedback gains based on the tracking error and the consequences for stability are analyzed in [9], but without considering the model fidelity. To the best of the authors' knowledge, no result is yet available for Gaussian Process based tracking control with adaptive feedback gains and stability guarantees.

The contribution of this paper is a Gaussian Process based control law for Lagrangian systems, that adapts feedback gains based on the model fidelity. For this purpose, the data-driven GPR learns the difference between an estimated model and the true system dynamics from training data. Afterwards, the control law uses the mean of the GPR to compensate the unknown dynamics and the model fidelity to adapt the feedback gains. The derived method guarantees that the tracking error is uniformly ultimately bounded and exponentially converges to a ball for a given probability.

The remainder of the paper starts with Section II where we introduce Lagrangian systems and GPR. Section III and IV describe the computation of the model error and the bounded tracking error. The method is validated in Section V.

Notation: Vectors and vector-valued functions are denoted with bold characters. Matrices are described with capital letters. The term $A_{:,i}$ denotes the i -th column of the matrix A . The expression $\mathcal{N}(\mu, \Sigma)$ describes a normal distribution with mean μ and covariance Σ . The Euclidean norm is given by $\|\cdot\|$ and the norm of a matrix A by $\|A\| = \bar{\lambda}(A^\top A)^{1/2}$.

¹ are with the Chair of Information-oriented Control (ITR), Department of Electrical and Computer Engineering, Technical University of Munich, 80333 Munich, Germany

{t.beckers, jonas.umlauft, hirche}@tum.de

² is with the University of Waterloo, Waterloo, ON N2L 3G1, Canada
dana.kulic@uwaterloo.ca

II. DEFINITIONS

This section starts with the necessary background on Gaussian Process regression. Afterwards, the class of Lagrangian system and stability concepts used are introduced.

A. Gaussian Process regression

Let (Ω, \mathcal{F}, P) be a probability space with the sample space Ω , the corresponding σ -algebra \mathcal{F} and the probability measure P . The set $\mathcal{X} \subseteq \mathbb{R}^n$ with $n \in \mathbb{N}_{>0}$ denotes the index set. A stochastic process is a discrete or real valued function $f(\mathbf{x}, \omega)$ which is a measurable function of $\omega \in \Omega$ with $\mathbf{x} \in \mathcal{X}$. A Gaussian Process is a stochastic process with

$$\begin{aligned} f(\mathbf{x}) &\sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')), & \mathbf{x}, \mathbf{x}' \in \mathcal{X} \\ m(\mathbf{x}): \mathcal{X} &\rightarrow \mathbb{R}, k(\mathbf{x}, \mathbf{x}'): \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}, \end{aligned}$$

which is fully described by a mean function $m(\mathbf{x})$ and a covariance function $k(\mathbf{x}, \mathbf{x}')$, since with fixed \mathbf{x} it is Gaussian distributed. The mean function is usually defined to be zero if no prior knowledge regarding f is available [5]. The covariance function is a measure of the correlation of data two points $(\mathbf{x}, \mathbf{x}')$. The covariance function depends on hyperparameters, which are function dependent. In this paper, we use the Gaussian Process

$$\mathbf{f}(\mathbf{x}) \sim \mathcal{GP}(\mathbf{m}(\mathbf{x}), \mathbf{k}(\mathbf{x}, \mathbf{x}')), \mathbf{f}: \mathbb{R}^n \rightarrow \mathbb{R}^n$$

with $\mathcal{X} = \mathbb{R}^n$ for regression of vector-valued, nonlinear functions. Since the output of a Gaussian Process is one dimensional, a n -dimensional function \mathbf{f} requires n GPs. Therefore, the vector valued function $\mathbf{m}(\cdot) = [m_1(\cdot), \dots, m_n(\cdot)]^\top$ describes the mean functions for each component of $\mathbf{f}(\mathbf{x})$. The Gaussian Process for each state depends on the corresponding mean and covariance function and is given by

$$\mathbf{f}(\mathbf{x}) = \begin{cases} f_1(\mathbf{x}) \sim \mathcal{GP}(m_1(\mathbf{x}), k_{\varphi_1}(\mathbf{x}, \mathbf{x}')) \\ \vdots & \vdots & \vdots \\ f_n(\mathbf{x}) \sim \mathcal{GP}(m_n(\mathbf{x}), k_{\varphi_n}(\mathbf{x}, \mathbf{x}')), \end{cases}$$

with the set of hyperparameters φ_i . The GPR is trained with an input and a corresponding output set which is generated by

$$\begin{aligned} \mathbf{y} &= \mathbf{f}(\mathbf{x}) + \boldsymbol{\eta}, \mathbf{y} \in \mathbb{R}^n \\ \boldsymbol{\eta} &\sim \mathcal{N}(\mathbf{0}, \text{diag}(\sigma_{n,1}^2, \dots, \sigma_{n,n}^2)) \end{aligned}$$

with Gaussian noise $\boldsymbol{\eta} \in \mathbb{R}^n$. The elements $\sigma_{n,i}^2$ are the variances of the noise of the output data for all $i \in \{1, \dots, n\}$. The m training inputs $\{\mathbf{x}^{\{j\}}\}_{j=1}^m$ and corresponding outputs $\{\mathbf{y}^{\{j\}}\}_{j=1}^m$ are concatenated in an input training matrix $X = [\mathbf{x}^{\{1\}}, \mathbf{x}^{\{2\}}, \dots, \mathbf{x}^{\{m\}}] \in \mathbb{R}^{n \times m}$ and an output training matrix $Y = [\mathbf{y}^{\{1\}}, \mathbf{y}^{\{2\}}, \dots, \mathbf{y}^{\{m\}}]^\top \in \mathbb{R}^{m \times n}$. Accordingly, the training data set is described by $\mathcal{D} = \{X, Y\}$. The prediction at a test input $\mathbf{x}^* \in \mathbb{R}^n$ for each component i of a new output vector $y_i^* \in \mathbb{R}$ is calculated as a Gaussian distributed variable with the conditional mean $\mu(y_i^* | \mathbf{x}^*, \mathcal{D})$

and the conditional variance $\text{var}(y_i^* | \mathbf{x}^*, \mathcal{D})$. The joint distribution of the i -th component of the vector y_i^* and the corresponding vector of the training outputs Y is

$$\begin{bmatrix} Y_{:,i} \\ y_i^* \end{bmatrix} \sim \mathcal{N} \left(\mathbf{0}, \begin{bmatrix} K_{\varphi_i}(X, X) & \mathbf{k}_{\varphi_i}(\mathbf{x}, X) \\ \mathbf{k}_{\varphi_i}(\mathbf{x}, X)^\top & k_{\varphi_i}(\mathbf{x}, \mathbf{x}) \end{bmatrix} \right) \quad (1)$$

where $Y_{:,i}$ is the i -th column of the matrix Y . The function $K_{\varphi_i}(X, X)$ is called the covariance matrix and $\mathbf{k}_{\varphi_i}(\mathbf{x}, X)$ the vector-valued extended covariance function with the set of hyperparameters φ_i

$$\begin{aligned} K_{\varphi_i}(X, X): \mathbb{R}^m \times \mathbb{R}^m &\rightarrow \mathbb{R}^{m \times m} \\ K_{j',j} &= k_{\varphi_i}(X_{:,j'}, X_{:,j}) \\ \mathbf{k}_{\varphi_i}(\mathbf{x}, X): \mathbb{R} \times \mathbb{R}^m &\rightarrow \mathbb{R}^m, k_{\varphi_i,j} = k_{\varphi_i}(\mathbf{x}, X_{:,j}) \\ \forall j', j \in \{1, \dots, m\}, i &\in \{1, \dots, n\}. \end{aligned}$$

Assuming the mean functions of the GPs are set to zero, a prediction of the i -th component of $\mathbf{y}^* \in \mathbb{R}^n$ is derived from the joint distribution (1). A nonzero mean function can be easily included, see [5] for details. The conditional probability distribution is Gaussian with the conditional mean

$$\mu_i(\mathbf{y}^* | \mathbf{x}^*, \mathcal{D}) = \mathbf{k}_{\varphi_i}(\mathbf{x}^*, X)^\top (K_{\varphi_i} + I\sigma_{n,i}^2)^{-1} Y_{:,i}$$

where I is the identity matrix while the variance of the prediction is given by

$$\begin{aligned} \text{var}_i(\mathbf{y}^* | \mathbf{x}^*, \mathcal{D}) &= k_{\varphi_i}(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{k}_{\varphi_i}(\mathbf{x}^*, X)^\top \\ &\quad (K_{\varphi_i} + I\sigma_{n,i}^2)^{-1} \mathbf{k}_{\varphi_i}(\mathbf{x}^*, X). \end{aligned}$$

The set of hyperparameters φ_i are optimized by means of the likelihood function, thus by

$$\varphi_i^* = \arg \max_{\varphi_i} \log P(Y_{:,i} | X, \varphi_i), \forall i = 1, \dots, n.$$

The n normally distributed components of $y_i^* | \mathbf{x}^*, \mathcal{D}$ are combined in a multi-variable Gaussian distribution

$$\begin{aligned} \mathbf{y}^* | \mathbf{x}^*, \mathcal{D} &\sim \mathcal{N}(\boldsymbol{\mu}(\cdot), \Sigma(\cdot)) \\ \boldsymbol{\mu}(\mathbf{y}^* | \mathbf{x}^*, \mathcal{D}) &= [\mu_1(\cdot), \dots, \mu_n(\cdot)]^\top \\ \Sigma(\mathbf{y}^* | \mathbf{x}^*, \mathcal{D}) &= \text{diag}(\text{var}_1(\cdot), \dots, \text{var}_n(\cdot)). \end{aligned}$$

The predicted variance with respect to just a part of $\mathbf{x}^* \in \mathbb{R}^n$ can be done by marginalization. Assume $\mathbf{x}^* = [\mathbf{x}_1^{*\top}, \mathbf{x}_2^{*\top}]^\top$ with $\mathbf{x}_1^* \in \mathbb{R}^{n_1}$, $\mathbf{x}_2^* \in \mathbb{R}^{n_2}$ and $\mathbf{x}^* \in \mathbb{R}^{n=n_1+n_2}$. Then, the marginal variance of the prediction based on \mathbf{x}_1^* is given by

$$\begin{aligned} \text{var}_i(\mathbf{y}^* | \mathbf{x}_1^*, \mathcal{D}) &= k_{\phi_i}(\mathbf{x}_1^*, \mathbf{x}_1^*) - \mathbf{k}_{\phi_i}(\mathbf{x}_1^*, X_{1:n_1,:})^\top \\ &\quad (K_{\phi_i} + I\sigma_{n,i}^2)^{-1} \mathbf{k}_{\phi_i}(\mathbf{x}_1^*, X_{1:n_1,:}) \quad (2) \end{aligned}$$

with the covariance function defined on the input space \mathbb{R}^{n_1} . Since the dimension of the input data is reduced, i.e. from $n_1 + n_2$ to n_1 , the necessary set of hyperparameters ϕ_i is a subset of the original set of hyperparameters φ_i . The combined marginal variance is rewritten as

$$\Sigma(\mathbf{y}^* | \mathbf{x}_1^*, \mathcal{D}) = \text{diag}(\text{var}_1(\cdot), \dots, \text{var}_{n_1}(\cdot)).$$

B. Class of Lagrangian systems

The assumed class of Lagrangian systems is described by

$$H(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + \boldsymbol{\kappa}(\check{\mathbf{q}}) = \boldsymbol{\tau}, \quad (3)$$

where $\mathbf{q} \in \mathbb{R}^n$ are the generalized coordinates with their time derivatives $\ddot{\mathbf{q}}, \dot{\mathbf{q}} \in \mathbb{R}^n$ and the generalized input $\boldsymbol{\tau} \in \mathbb{R}^n$. An additional unknown dynamic $\boldsymbol{\kappa}(\check{\mathbf{q}}) : \mathbb{R}^{3n} \rightarrow \mathbb{R}^n$ which depends on $\check{\mathbf{q}} = [\ddot{\mathbf{q}}^\top, \dot{\mathbf{q}}^\top, \mathbf{q}^\top]^\top$ affects the system as a generalized force. The generalized inertia $H(\mathbf{q}) : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$, the matrix $C(\mathbf{q}, \dot{\mathbf{q}}) : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$, and $\mathbf{g}(\mathbf{q}) : \mathbb{R}^n \rightarrow \mathbb{R}^n$ are assumed to have the following properties:

Property 1 (Structural properties)

The matrix $H(\mathbf{q})$ is symmetric and positive definite and there is a parameterization of matrix $C(\mathbf{q}, \dot{\mathbf{q}})$ so that $\forall \dot{\mathbf{q}}, \mathbf{q} \in \mathbb{R}^n$

- $\dot{H}(\mathbf{q}) = C(\mathbf{q}, \dot{\mathbf{q}}) + C(\mathbf{q}, \dot{\mathbf{q}})^\top \in \mathbb{R}^{n \times n}$ and thus,
- $\dot{H}(\mathbf{q}) - 2C(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{n \times n}$ is skew-symmetric.

Property 2 (Boundedness and Linearity)

- The matrix $H(\mathbf{q})$ is bounded, i.e. there exists two constants $h_1 \in \mathbb{R}_{>0}$ and $h_2 \in \mathbb{R}_{>0}$ such that $H(\mathbf{q})$ is bounded with $h_1 \|\mathbf{x}\|^2 \leq \mathbf{x}^\top H(\mathbf{q}) \mathbf{x} \leq h_2 \|\mathbf{x}\|^2$ for all $\mathbf{x}, \mathbf{q} \in \mathbb{R}^n$.
- The matrix $C(\mathbf{q}, \dot{\mathbf{q}})$ is bounded in \mathbf{q} and linear in $\dot{\mathbf{q}}$, i.e. there exists a $k_C \in \mathbb{R}_{>0}$ such that $\|C(\mathbf{q}, \dot{\mathbf{q}})\| \leq k_C \|\dot{\mathbf{q}}\|$, and $C(\mathbf{q}, \dot{\mathbf{q}})\mathbf{p} = C(\mathbf{q}, \mathbf{p})\dot{\mathbf{q}}$ for all $\mathbf{q}, \dot{\mathbf{q}}, \mathbf{p} \in \mathbb{R}^n$.

As shown in [10], these properties hold, for example, for the class of robotic manipulators with revolute joints.

III. MODELING

In this section, we introduce the modeling procedure and the error estimation between the learned and the true dynamics. Consider the Lagrangian system in (3) for which the following is assumed:

Assumption 1

The generalized external force $\boldsymbol{\kappa}(\check{\mathbf{q}})$ has a bounded reproducing kernel Hilbert space (RKHS) norm, i.e. $\|\boldsymbol{\kappa}\|_k < \infty$, with respect to the covariance function $k(x, x')$ of a GPR.

This assumption ensures that the covariance function is suitable for the approximation of the function $\boldsymbol{\kappa}(\check{\mathbf{q}})$. A comparison of different covariance functions and recommendation for Lagrangian systems can be found in [11], [12].

To achieve a controller with a good feed-forward compensation, the system (3) must be identified. Since partial a priori knowledge of the system is often available, we use a so-called hybrid learning approach which is a combination of a parametric and a data-driven model. For this purpose, a Gaussian Process learns the difference between the real and the estimated dynamics

$$\hat{\boldsymbol{\tau}} = \hat{H}(\mathbf{q})\ddot{\mathbf{q}} + \hat{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \hat{\mathbf{g}}(\mathbf{q}) \quad (4)$$

of the system where \hat{H} , \hat{C} and $\hat{\mathbf{g}}$ are the estimates of the true matrices, thus

$$\begin{aligned} \tilde{\boldsymbol{\tau}} = \boldsymbol{\tau} - \hat{\boldsymbol{\tau}} = & H(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + \boldsymbol{\kappa}(\check{\mathbf{q}}) \\ & - \hat{H}(\mathbf{q})\ddot{\mathbf{q}} - \hat{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - \hat{\mathbf{g}}(\mathbf{q}). \end{aligned} \quad (5)$$

Without loss of generality, we assume in the following analysis that $H = \hat{H}$, $C = \hat{C}$, and $\mathbf{g} = \hat{\mathbf{g}}$ since the error of the estimation can be included in $\boldsymbol{\kappa}(\check{\mathbf{q}})$. For the generation of training data, a set of $\{\check{\mathbf{q}}, \dot{\mathbf{q}}, \mathbf{q}\}$ as training inputs X and $\tilde{\boldsymbol{\tau}}$ as training output Y is necessary, which could be generated through any suitably well-behaving control. For the stability analysis of the closed loop, the error between the model and the true dynamics is necessary. A probabilistic upper bound for the distance between the mean prediction $\boldsymbol{\mu}(\tilde{\boldsymbol{\tau}})$ of the Gaussian Process regression and the true function is given in [13] and is extended for multidimensional functions in the following lemma.

Lemma 1

Consider a Lagrangian System (3) and the estimated dynamics (4). A Gaussian Process is trained with the difference between the true and the estimated dynamics. Then the model error is bounded by

$$P\{\|\boldsymbol{\mu}(\tilde{\boldsymbol{\tau}}|\check{\mathbf{q}}, \mathcal{D}) - \boldsymbol{\kappa}(\check{\mathbf{q}})\| \leq \|\boldsymbol{\beta}^\top \Sigma(\tilde{\boldsymbol{\tau}}|\check{\mathbf{q}}, \mathcal{D})\| \} \leq (1 - \delta)^n$$

for $\check{\mathbf{q}} \in D$ on a compact set $D \subset \mathbb{R}^{3n}$ where each element of $\boldsymbol{\beta} \in \mathbb{R}^n$ is defined as

$$\beta_j = \sqrt{2\|\boldsymbol{\kappa}_j\|_k^2 + 300\gamma_j \ln^3\left(\frac{m+1}{\delta}\right)}.$$

The variable $\gamma_j \in \mathbb{R}$ is the maximum information gain

$$\gamma_j = \max_{\check{\mathbf{q}}^{\{1\}}, \dots, \check{\mathbf{q}}^{\{m+1\}} \in D} \frac{1}{2} \log |I + \sigma^{-2} K_{\varphi_j}(\mathbf{x}, \mathbf{x}')|$$

$$\mathbf{x}, \mathbf{x}' \in \{\check{\mathbf{q}}^{\{1\}}, \dots, \check{\mathbf{q}}^{\{m+1\}}\}$$

with the covariance matrix K_{φ_j} .

Proof: The result is a direct consequence of [13, Theorem 6] and the fact that $\boldsymbol{\eta}$ is uncorrelated. Therefore,

$$\begin{aligned} & P\{\forall \check{\mathbf{q}} \in D, \\ & |\boldsymbol{\mu}(\tilde{\boldsymbol{\tau}}_1|\check{\mathbf{q}}, \mathcal{D}) - f_1(\check{\mathbf{q}})| \leq |\beta_1 \text{var}_1(\tilde{\boldsymbol{\tau}}_1|\check{\mathbf{q}}, \mathcal{D}) \cap \dots \cap \\ & |\boldsymbol{\mu}(\tilde{\boldsymbol{\tau}}_n|\check{\mathbf{q}}, \mathcal{D}) - f_n(\check{\mathbf{q}})| \leq |\beta_n \text{var}_n(\tilde{\boldsymbol{\tau}}_n|\check{\mathbf{q}}, \mathcal{D})| \} \geq (1 - \delta)^n \\ \Rightarrow & P\{\forall \check{\mathbf{q}} \in D, \Delta \leq \|\boldsymbol{\beta}^\top \Sigma(\tilde{\boldsymbol{\tau}}|\check{\mathbf{q}}, \mathcal{D})\| \} \geq (1 - \delta)^n \end{aligned}$$

provides an upper bound for the norm of the model error with a probability of at least $(1 - \delta)^n$. ■

The information capacity γ has a sublinear dependence on the number of training points for many commonly used covariance functions and can be approximated by a constant, e.g. shown in [13]. Therefore, even though $\boldsymbol{\beta}$ is increasing with the number of training samples, it is possible to learn the true function $\mathbf{f}(\mathbf{x})$ arbitrarily exactly.

IV. PD CONTROL WITH VARIABLE GAINS

Classical computed torque control employs static feedback gains. Low feedback gains are desirable to avoid saturation of the actuators and achieve a good noise attenuation. However, the unknown dynamics usually requires a specific minimal feedback gain to keep the tracking error under a defined limit. After a training procedure, we use the mean of the

Gaussian Process regression to compensate $\kappa(\check{q})$ with the feed-forward part and adapt the gains based on the model fidelity. For this purpose, the uncertainty of the regression is used to scale the feedback gains.

We start with the following natural assumption for the desired trajectory.

Assumption 2

The desired trajectory is bounded by $\|\mathbf{q}_d\| \leq \bar{q}_d$, $\|\dot{\mathbf{q}}_d\| \leq \bar{\dot{q}}_d$ with $\bar{q}_d, \bar{\dot{q}}_d \in \mathbb{R}_{\geq 0}$.

In the next step, an assumption for the varying gains is introduced.

Assumption 3

Let $G_d(\check{\mathbf{q}}, \mathbf{q}): \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$ and $G_p(\mathbf{q}): \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$.
i) Let $K_d: \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^{n \times n}$ be a positive definite and symmetric matrix such that $(K_d \circ G_d)$ is continuous and that there exists a lower and upper quadratic bound

$$k_{d1}\|\mathbf{x}\|^2 \leq \mathbf{x}^\top K_d(G_d(\check{\mathbf{q}}, \mathbf{q}))\mathbf{x} \leq k_{d2}\|\mathbf{x}\|^2, \forall \check{\mathbf{q}}, \mathbf{q}, \mathbf{x} \in \mathbb{R}^n,$$

with $k_{d1} \in \mathbb{R}_{\geq 0}$ and $k_{d2} \in \mathbb{R}_{> 0}$.

ii) Let $K_p: \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^{n \times n}$ be a positive definite diagonal matrix. Each diagonal element $K_{p,ii}(G_p(\mathbf{q}))$ is continuous and bounded by $0 < \underline{k}_{p,ii} \leq K_{p,ii}(G_p(\mathbf{q})) \leq \bar{k}_{p,ii}$ for all $\mathbf{q} \in \mathbb{R}^n, i = 1, \dots, n$, so that

$$k_{p1}\|\mathbf{x}\|^2 \leq \mathbf{x}^\top K_p(G_p(\mathbf{q}))\mathbf{x} \leq k_{p2}\|\mathbf{x}\|^2, \forall \mathbf{q}, \mathbf{x} \in \mathbb{R}^n$$

with $k_{p1} = \min_i \underline{k}_{p,ii} \in \mathbb{R}_{\geq 0}$ and $k_{p2} = \max_i \bar{k}_{p,ii} \in \mathbb{R}_{> 0}$ for all $i = 1, \dots, n$.

Assumption 3 restricts the matrix K_p to be diagonal which results in a decentralized feedback of the tracking error. The symmetric form of K_p and K_d is a common assumption which does not restrict the applicability of the approach but must be kept in mind during the design of the controller. Before the control law is proposed, the following definition and lemma are introduced.

Definition 1

Assume a Gaussian Process trained with the difference between the true and the estimated dynamics of a Lagrangian system (5). The marginal variances $\Sigma_d: \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$ and $\Sigma_p: \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$ are defined with (2) by

$$\Sigma_d = \Sigma(\tilde{\tau}|\check{\mathbf{q}}, \mathbf{q}, \mathcal{D}), \Sigma_p = \Sigma(\tilde{\tau}|\mathbf{q}, \mathcal{D}).$$

Lemma 2

There exists an $\epsilon > 0$ such that the matrix $A \in \mathbb{R}^{2n \times 2n}$ given by¹

$$A = \begin{bmatrix} -K_d(\Sigma_d) + \epsilon H & \frac{\epsilon}{2}(-K_d^\top(\Sigma_d) + C) \\ \frac{\epsilon}{2}(-K_d(\Sigma_d) + C^\top) & -\epsilon K_p(\Sigma_p) \end{bmatrix},$$

which is negative definite under Property 1 and Assumption 3 for all $\check{\mathbf{q}}, \mathbf{q} \in \mathbb{R}^n$.

¹For notational convenience, the dependencies of H, C, \mathbf{g} are dropped here.

Proof: According to the Schur's Lemma, A is negative definite if

$$\begin{aligned} A_{11} &= -K_d(\Sigma_d) + \epsilon H \\ S &= -\epsilon K_p(\Sigma_p) + \frac{\epsilon^2}{4}(K_d(\Sigma_d) - C^\top) \\ &\quad (K_d(\Sigma_d) - \epsilon H)^{-1}(K_d^\top(\Sigma_d) - C) \end{aligned}$$

are negative definite where $A_{11} \in \mathbb{R}^{n \times n}$ is the upper left block of A and $S \in \mathbb{R}^{n \times n}$ is the Schur complement. Since K_d, H , and K_p are positive definite and bounded, ϵ can be chosen sufficiently small to obtain the negative definiteness of A_{11} . The second summand of the Schur complement S is quadratic in ϵ and positive definite while the first summand is linear in ϵ and negative. Thus, for every $\mathbf{q}, \check{\mathbf{q}} \in \mathbb{R}^n$, an ϵ can be found which guarantees the negative definiteness of the Schur complement. Therefore, the matrix A is negative definite. ■

The next theorem introduces the control law with guaranteed boundedness of the tracking error which is the main contribution of the paper.

Theorem 1

Consider the Lagrangian system (3) which satisfies the Properties 1 and 2 and Assumptions 1 to 3. A Gaussian Process is trained with m data pairs of the set $\mathcal{D} = \{\check{\mathbf{q}}^{\{i\}}, \tilde{\tau}^{\{i\}}\}_{i=1}^m$ with

$$\tilde{\tau} = \tau - H(\mathbf{q})\ddot{\mathbf{q}} - C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - \mathbf{g}(\mathbf{q}).$$

Let $\mathbf{e} = \mathbf{q} - \mathbf{q}_d, \dot{\mathbf{e}} = \dot{\mathbf{q}} - \dot{\mathbf{q}}_d$ be the tracking error. The control law

$$\begin{aligned} \tau &= H(\mathbf{q})\ddot{\mathbf{q}}_d + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}_d + \mathbf{g}(\mathbf{q}) + \boldsymbol{\mu}(\tilde{\tau}|\check{\mathbf{q}}, \mathcal{D}) \\ &\quad - K_d(\Sigma_d)\dot{\mathbf{e}} - K_p(\Sigma_p)\mathbf{e} \end{aligned} \quad (6)$$

guarantees that the tracking error is uniformly ultimately bounded and exponentially converges to a ball with a probability of at least $(1 - \delta)^n, \delta \in (0, 1)$.

Proof: For the stability analysis, we use the following Lyapunov candidate

$$V(\dot{\mathbf{e}}, \mathbf{e}) = \frac{1}{2}\dot{\mathbf{e}}^\top H(\mathbf{q})\dot{\mathbf{e}} + \int_0^{\mathbf{e}} \mathbf{z}^\top K_p(\Sigma_p)d\mathbf{z} + \epsilon \mathbf{e}^\top H(\mathbf{q})\dot{\mathbf{e}}$$

with a parameter $\epsilon > 0$. To ensure that the Lyapunov candidate is positive definite, we analyze the domain of the integral

$$\int_0^{\mathbf{e}} \mathbf{z}^\top K_p(\Sigma_p)d\mathbf{z} \geq \frac{1}{2} \sum_{i=1}^n k_{p,ii} e_i^2 \geq \frac{1}{2} k_{p1} \|\mathbf{e}\|^2,$$

where each component of the sum has a lower bound and thus, the whole integral is lower bounded. An upper quadratic bound can be found in a similar way as presented in [9]. Since the integral is lower bounded and $H(\mathbf{q})$ is always positive definite, the parameter ϵ can be chosen sufficiently small to achieve a positive definite and radially unbounded Lyapunov function. The valid interval for ϵ can be determined by the lower bound of the Lyapunov function

$$V(\dot{\mathbf{e}}, \mathbf{e}) \geq \frac{1}{2} h_1 \|\dot{\mathbf{e}}\|^2 + \frac{1}{2} k_{p1} \|\mathbf{e}\|^2 - \frac{1}{2} \epsilon h_2 (\|\dot{\mathbf{e}}\|^2 + \|\mathbf{e}\|^2)$$

which is positive for $0 < \varepsilon < \min\{k_{p1}/h_2, h_1/h_2\}$. In the next step, we investigate the time derivative of the Lyapunov function to establish stability of the closed loop. With Property 1 and (6), it can be written as

$$\begin{aligned} \dot{V} = & \begin{bmatrix} \dot{e} \\ e \end{bmatrix}^\top \underbrace{\begin{bmatrix} -K_d(\Sigma_d) + \varepsilon H & \frac{\varepsilon}{2}(-K_d^\top(\Sigma_d) + C) \\ \frac{\varepsilon}{2}(-K_d(\Sigma_d) + C^\top) & -\varepsilon K_p(\Sigma_p) \end{bmatrix}}_A \begin{bmatrix} \dot{e} \\ e \end{bmatrix} \\ & + \begin{bmatrix} \dot{e}^\top & \varepsilon e^\top \end{bmatrix} \underbrace{\begin{bmatrix} \boldsymbol{\mu}(\tilde{\boldsymbol{\tau}}) - \boldsymbol{\kappa}(\check{\boldsymbol{q}}) \\ \boldsymbol{\mu}(\tilde{\boldsymbol{\tau}}) - \boldsymbol{\kappa}(\check{\boldsymbol{q}}) \end{bmatrix}}_b. \end{aligned} \quad (7)$$

For the analysis, we define the two parts of the equation as $A \in \mathbb{R}^{2n \times 2n}$ and $b \in \mathbb{R}^{2n}$. The following statements can be made for the matrix A : The submatrix A_{11} is bounded with

$$\dot{e}^\top A_{11} \dot{e} = \dot{e}^\top (-K_d(\Sigma_d)\varepsilon H) \dot{e} \leq (-k_{d1} + \varepsilon h_2) \|\dot{e}\|^2.$$

With Assumptions 2 and 3, and Property 2, the submatrix A_{21} is bounded by

$$e^\top A_{21} \dot{e} \leq \varepsilon (k_C \|\dot{e}\| + k_C \bar{q}_d + k_{d2}) \|\dot{e}\| \|e\|.$$

Then, using Lemma 1 and Assumption 3, the overall upper bound for the time derivative of the Lyapunov function is given by

$$\begin{aligned} \dot{V}(\dot{e}, e) \leq & (\varepsilon h_2 - k_{d1}) \|\dot{e}\|^2 - \varepsilon k_{p1} \|e\|^2 \\ & + \varepsilon (k_C \|\dot{e}\| + k_C \bar{q}_d + k_{d2}) \|\dot{e}\| \|e\| \\ & + (\|\dot{e}\| + \varepsilon \|e\|) \|\boldsymbol{\beta}^\top \Sigma(\tilde{\boldsymbol{\tau}}|\check{\boldsymbol{q}}, \mathcal{D})\|. \end{aligned} \quad (8)$$

Considering the inequality

$$\|\dot{e}\| \|e\| \leq \frac{1}{2} \left(\rho \|\dot{e}\|^2 + \frac{e^2}{\rho} \right)$$

that holds for all $\dot{e}, e \in \mathbb{R}^n$ and $\rho \in \mathbb{R}_{\geq 0}$, (8) can be rewritten as

$$\begin{aligned} \dot{V}(\dot{e}, e) \leq & \left(\varepsilon h_2 - k_{d1} + \frac{\varepsilon \rho}{2} (k_C \bar{q}_d + k_{d2}) \right) \|\dot{e}\|^2 \\ & - \varepsilon k_{p1} \frac{\varepsilon_2}{1 + \varepsilon_2} \|e\|^2 + \varepsilon k_C \|\dot{e}\|^2 \|e\| \\ & + (\|\dot{e}\| + \varepsilon \|e\|) \|\boldsymbol{\beta}^\top \Sigma(\tilde{\boldsymbol{\tau}}|\check{\boldsymbol{q}}, \mathcal{D})\| \end{aligned} \quad (9)$$

$$\text{with } \rho = (1 + \varepsilon_2) \frac{k_C \bar{q}_d + k_{d2}}{2k_{p1}}, \varepsilon_2 \in \mathbb{R}_{>0}.$$

The choice of ρ guarantees that the factors of the quadratic parts are still negative. The linear part of (9) can be bounded by a quadratic function with $v_1 \|\boldsymbol{x}\| \leq v_1^2/v_2 + v_2/4 \|\boldsymbol{x}\|^2$ that holds for all $\boldsymbol{x} \in \mathbb{R}^n$ and $v_1, v_2 \in \mathbb{R}_{\geq 0}$

$$\begin{aligned} & (\|\dot{e}\| + \varepsilon \|e\|) \|\boldsymbol{\beta}^\top \Sigma(\tilde{\boldsymbol{\tau}}|\check{\boldsymbol{q}}, \mathcal{D})\| \\ & \leq \frac{\bar{\Delta}^2}{v_1} + \frac{v_1}{4} \|\dot{e}\| + \frac{\varepsilon^2 \bar{\Delta}^2}{\varepsilon v_2} + \frac{\varepsilon v_2}{4} \|e\|. \end{aligned} \quad (10)$$

with $\|\boldsymbol{\beta}^\top \Sigma(\tilde{\boldsymbol{\tau}}|\check{\boldsymbol{q}}, \mathcal{D})\| \leq \bar{\Delta} \in \mathbb{R}_{>0}$ of Lemma 1 and

$$v_1 := -\varepsilon h_2 + k_{d1} - \frac{\varepsilon \rho}{2} (k_C \bar{q}_d + k_{d2}), v_2 := k_{p1} \frac{\varepsilon_2}{1 + \varepsilon_2}.$$

Since the covariance function is bounded on the closed interval D , the variance $\Sigma(\tilde{\boldsymbol{\tau}}|\check{\boldsymbol{q}}, \mathcal{D})$ is bounded, for more details

see [14]. Thus, there exists an upper bound $\bar{\Delta}$ for the model error relating to \dot{e}, e . To ensure that the variables v_1, v_2 are positive, the restriction for ε must be extended to

$$0 < \varepsilon < \min \left\{ \frac{k_{p1}}{h_2}, \frac{h_1}{h_2}, \frac{2k_{d1}}{2h_2 + \rho(k_C \bar{q}_d + k_{d2})} \right\}. \quad (11)$$

With (10), equation (9) can be rewritten as

$$\begin{aligned} \dot{V}(\dot{e}, e) \leq & -\frac{3}{4} v_1 \|\dot{e}\|^2 - \frac{3}{4} \varepsilon v_2 \|e\| + \varepsilon k_C \|\dot{e}\|^2 \|e\| \\ & + \frac{\bar{\Delta}^2}{v_1} + \varepsilon \frac{\bar{\Delta}^2}{v_2}. \end{aligned}$$

According to [9] and Lemma 1, there exists a $\xi \in \mathbb{R}_{\geq 0}$ and a $\varrho \in \mathbb{R}_{\geq 0}$ for (8) such that

$$P \left\{ \dot{V}(\dot{e}, e) \leq -\xi V(\dot{e}, e) + \varrho \right\} \geq (1 - \delta)^n$$

holds with

$$\begin{aligned} \xi &= \frac{2}{3} \frac{\min \left\{ \varepsilon v_2, v_1 - \frac{4}{3} \varepsilon k_C \sqrt{\frac{2V_0}{k_{p1} - \varepsilon h_2}} \right\}}{\max \{ \varepsilon h_2 + k_{p2}, (1 + \varepsilon) h_2 \}} \\ \varrho &= \frac{\bar{\Delta}^2}{v_1} + \varepsilon \frac{\bar{\Delta}^2}{v_2} \end{aligned}$$

and the extension of (11)

$$0 < \varepsilon < \min \left\{ \frac{k_{p1}}{h_2}, \frac{h_1}{h_2}, \frac{2k_{d1}}{2h_2 + \frac{2k_{p1}\rho^2}{1 + \varepsilon_2} + \frac{8}{3} k_C \sqrt{\frac{2V_0}{k_{p1} - \varepsilon h_2}}} \right\}.$$

Finally, Lemma 1 requires that $[\check{\boldsymbol{q}}, \dot{\boldsymbol{q}}, \boldsymbol{q}]$ is always an element of the set D . Therefore, it must be chosen so that

$$\{\forall \check{\boldsymbol{q}}, \dot{\boldsymbol{q}}, \boldsymbol{q} \in \mathbb{R}^n : V(\dot{e}, e) \leq V_0\} \in D$$

$$\text{and } \{\forall \check{\boldsymbol{q}}, \dot{\boldsymbol{q}}, \boldsymbol{q} \in \mathbb{R}^n : \|\dot{e}^\top, e^\top\| \leq \sqrt{2\varrho/(\xi\bar{c})}\} \in D$$

with $\bar{c} = \min\{k_{p1} - \varepsilon h_2, h_1 - \varepsilon h_2\}$ which guarantees that the trajectory stays inside the set D , [9]. Thus, the tracking error is uniformly ultimately bounded and converge to a ball with a probability of at least $(1 - \delta)^n$. ■

Remark 1

The first summand of (7) contains the influence of the controller on the system while the second summand captures the model error. If a perfect model was available, such that $\boldsymbol{\mu}(\tilde{\boldsymbol{\tau}}) = \boldsymbol{\kappa}(\check{\boldsymbol{q}})$, equation (7) with Lemma 2 would show that the closed loop system is asymptotically stable.

Remark 2

A similar idea of GPR-based computed torque control is presented in [15], however, without stability analysis.

V. SIMULATION

For the simulation, we apply Lagrange's equations to the common model of a 2-link planar manipulator [16]. We assume a point mass for the links of 1 kg which are located in the center of each link. The length of the links is set to 1 m. The joints are without mass and not influenced by any friction. Gravity is assumed to be 10 ms^{-2} . Here, the generalized coordinates q_1 and q_2 are the joint angles. The initial values are set to $\boldsymbol{q}_0 = [0, 0]^\top$. The unknown dynamics $\boldsymbol{\kappa}(\cdot)$

is simulated by a sample path of a Gaussian Process with a squared exponential covariance function which is acting here as ground truth. This approach guarantees that $\kappa(\cdot)$ has a bounded RKHS norm regarding to the squared exponential function. This Gaussian Process is trained by 50 data values of the arbitrary chosen nonlinear function

$$f(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} -\dot{q}_1 + 2 \sin(q_2) + |q_1| \\ -\dot{q}_2 + 2 \sin(q_2) \end{bmatrix}.$$

Now, the proposed control law of Theorem 1 is used. We assume that the estimated matrices of the Lagrangian system are equal to the true matrices. A Gaussian Process with a squared exponential covariance function learns the difference between the estimated model and the true system, thus the unknown dynamics $\kappa(\cdot)$. For this purpose, we generate 225 pairs of $\{\tilde{\tau}\}$ and states $\{\tilde{\mathbf{q}}, \tilde{\dot{\mathbf{q}}}, \tilde{\mathbf{q}}\}$ as training data on the domain $\tilde{\mathbf{q}}, \tilde{\dot{\mathbf{q}}} \in [-1, 1]^2$, $\tilde{\mathbf{q}} \in [0, 1]$ to generate a set \mathcal{D} of training points. The hyperparameters are optimized by means of the likelihood function. The desired trajectory is a sinusoidal function with $\mathbf{q}_0 = [0, 1]^\top$.

As comparison, the proposed control law with static gains $K_{p,\text{static}} = \text{diag}(10, 10)$ and $K_{d,\text{static}} = \text{diag}(10, 10)$ is used. Figure 1 shows the resulting trajectory for the first joint. The system trajectory with static gains (red dotted) is close to the desired trajectory (blue dashed) while it is in the neighborhood of the training data. Outside this area, the tracking error increases. Now, the same control law of Theorem 1 with variable gains is used. In this example, the gains are adapted according to Assumption 3 with

$$K_p(\Sigma_p) = \text{diag}(10 + 30 \text{var}_{p,1}(\tilde{\tau}), 10 + 30 \text{var}_{p,2}(\tilde{\tau}))$$

$$K_d(\Sigma_d) = \text{diag}(10 + 30 \text{var}_{d,1}(\tilde{\tau}), 10 + 30 \text{var}_{d,2}(\tilde{\tau})).$$

In Fig. 1, the color of the trajectory indicates the norm of the current feedback gains. In the area close to the training data, the feedback gains remain low (blue color) while outside the training area the gains increase (red color). The result is that the tracking error is kept low and bounded even for areas where no training data is available.

CONCLUSION

In this paper, we propose a GPR-based control law for Lagrangian systems which guarantees a bounded tracking error. The feedback gains of the control law are adapted by the model fidelity to keep the feedback gains as low as possible. The main contribution is that the tracking error of the closed loop system with the data-driven GPR model is proven to be uniformly ultimately bounded and exponentially convergent to a ball with a given probability.

ACKNOWLEDGMENTS

The research leading to these results has received funding from the European Research Council under the European Union Seventh Framework Program (FP7/2007-2013) / ERC Starting Grant ‘‘Control based on Human Models (con-humo)’’ agreement n^o337654.

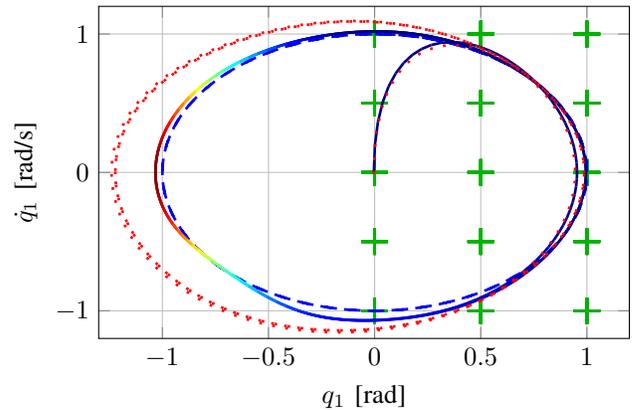


Fig. 1. Comparison between the proposed control law with static feedback gains (red dotted) and adapted feedback gains (solid line) regarding to the desired trajectory (blue dashed). The color of the adapted feedback trajectory indicates the norm of the current feedback gains. The green crosses mark the training data points.

REFERENCES

- [1] B. Siciliano, L. Sciacivco, L. Villani, and G. Oriolo, *Robotics: modelling, planning and control*. Springer Science+Business Media, 2010.
- [2] D. Nguyen-Tuong and J. Peters, ‘‘Learning robot dynamics for computed torque control using local Gaussian processes regression,’’ in *Symposium on Learning and Adaptive Behaviors for Robotic Systems*, pp. 59–64, 2008.
- [3] M. W. Spong and M. Vidyasagar, *Robot dynamics and control*. John Wiley & Sons, 2008.
- [4] M. P. Deisenroth, D. Fox, and C. E. Rasmussen, ‘‘Gaussian processes for data-efficient learning in robotics and control,’’ *Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 2, pp. 408–423, 2015.
- [5] C. E. Rasmussen and C. K. Williams, *Gaussian processes for machine learning*, vol. 1. MIT press Cambridge, 2006.
- [6] J.-J. E. Slotine and W. Li, ‘‘On the adaptive control of robot manipulators,’’ *The international journal of robotics research*, vol. 6, no. 3, pp. 49–59, 1987.
- [7] G. Chowdhary, H. A. Kingravi, J. P. How, and P. A. Vela, ‘‘Bayesian nonparametric adaptive control using gaussian processes,’’ *IEEE transactions on neural networks and learning systems*, vol. 26, no. 3, pp. 537–550, 2015.
- [8] D. Nguyen-Tuong and J. Peters, ‘‘Using model knowledge for learning inverse dynamics,’’ in *Proc. of the International Conference on Robotics and Automation*, pp. 2677–2682, 2010.
- [9] T. Ravichandran, D. Wang, and G. Heppner, ‘‘Stability and robustness of a class of nonlinear controllers for robot manipulators,’’ in *Proc. of the American Control Conference*, vol. 6, pp. 5262–5267, IEEE, 2004.
- [10] F. Ghorbel, B. Srinivasan, and M. W. Spong, ‘‘On the positive definiteness and uniform boundedness of the inertia matrix of robot manipulators,’’ in *Proc. of the Conference on Decision and Control*, pp. 1103–1108, IEEE, 1993.
- [11] C. M. Bishop *et al.*, *Pattern recognition and machine learning*, vol. 4. Springer New York, 2006.
- [12] C.-A. Cheng, H.-P. Huang, H.-K. Hsu, W.-Z. Lai, and C.-C. Cheng, ‘‘Learning the inverse dynamics of robotic manipulators in structured reproducing kernel hilbert space,’’ *IEEE transactions on cybernetics*, vol. 46, no. 7, pp. 1691–1703, 2016.
- [13] 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*, vol. 58, no. 5, pp. 3250–3265, 2012.
- [14] T. Beckers and S. Hirche, ‘‘Equilibrium distributions and stability analysis of gaussian process state space models,’’ in *Proc. of the Conference on Decision and Control*, 2016.
- [15] N. T. Alberto, M. Mistry, and F. Stulp, ‘‘Computed torque control with variable gains through Gaussian process regression,’’ in *Proc. of the International Conference on Humanoid Robots*, pp. 212–217, 2014.
- [16] R. M. Murray, Z. Li, S. S. Sastry, and S. S. Sastry, *A mathematical introduction to robotic manipulation*. CRC press, 1994.