MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

A Lagrangian Inspired Polynomial Kernel for Robot Dynamics Identification

Giacomuzzo, Giulio; Dalla Libera, Alberto; Carli, Ruggero; Romeres, Diego

TR2023-063 June 01, 2023

Abstract

In this paper, we propose a novel kernel for the identification of the inverse dynamics of robotic manipulators based on Gaussian Process Regression. The proposed kernel, called Lagrangian Inspired Polynomial (LIP) kernel is based on two main ideas. First, instead of directly modeling the joint torques, we model as GPs the kinetic and potential energy of the system. To this aim, we prove a polynomial characterization of the kinetic and potential energy and we define a polynomial kernel that encodes this property. Second, we derive the GP prior on the joint torques by leveraging on the knowledge of Lagrange's equations and by applying the properties of GPs under linear operators. We validated our method on a Franka Emika Panda robot, a 7 DOF cobot. The collected results show that the proposed method outperforms state-of-the-art black-box estimators based on Gaussian Processes in terms of prediction accuracy and generalization.

ICRA 2023 Workshop on Effective Representations, Abstractions, and Priors for Robot Learning (RAP4Robots)

A Lagrangian Inspired Polynomial Kernel for Robot Dynamics Identification

Giulio Giacomuzzo[1], Alberto Dalla Libera[1], Ruggero Carli[1], Diego Romeres[2]

***Abstract*— In this paper, we propose a novel kernel for the identification of the inverse dynamics of robotic manipulators based on Gaussian Process Regression. The proposed kernel, called *Lagrangian Inspired Polynomial* (LIP) kernel is based on two main ideas. First, instead of directly modeling the joint torques, we model as GPs the kinetic and potential energy of the system. To this aim, we prove a polynomial characterization of the kinetic and potential energy and we define a polynomial kernel that encodes this property. Second, we derive the GP prior on the joint torques by leveraging on the knowledge of Lagrange's equations and by applying the properties of GPs under linear operators. We validated our method on a Franka Emika Panda robot, a 7 DOF cobot. The collected results show that the proposed method outperforms state-of-the-art black-box estimators based on Gaussian Processes in terms of prediction accuracy and generalization.**

I. Introduction

In many robotics applications, control performance strongly benefits from the presence of accurate models. Inverse dynamics models expresse joint torques as a function of joint positions, velocities and accelerations, and they are crucial in several control problems, such as high-precision trajectory tracking [1], [2] and contact detection [3]–[5].

Learning the inverse dynamics in a black-box fashion represents an appealing solution. Differently from traditional model-based approaches [6]–[9], black-box solution learn inverse dynamics models directly from collected data, without the need of prior knowledge of the physical system.

Several black-box methods have been proposed, for instance relying on deep neural networks (NN) [10]- [11], [12] - and Gaussian Process Regression (GPR) [13], see [14]–[17]. Despite their ability to approximate even complex nonlinear dynamics, black-box methods typically exhibit low data efficiency and poor generalization properties: obtained estimators perform well only within a neighborhood of the training trajectories, and require a large amount of samples to generalize.

Several works tried to improve generalization by embedding insights from physics as a prior in black-box models [18]–[21], so that physical properties are embedded in the model. In this context, GPR is an attractive regression framework since prior on the physical properties can be encoded through a proper definition of the kernel function. In this manuscript, we propose an inverse dynamics GP estimator based on a novel kernel defined starting from the laws of Lagrangian mechanics, with the aim of improving generalization and data efficiency.

[1]Department of Information Engineering, Università di Padova, Italy {giacomuzzo,dallaliber,carlirug}@dei.unipd.it
[2]Mitsubishi Electric Research Lab - MERL
romeres@merl.com

Typically, when GPR is applied to the inverse dynamics identification, each joint torque is modeled with a distinct independent Gaussian Process (GP). This choice simplifies the regression problem but ignores the correlations between the different joint torques, which instead are present as explained by the Lagrange's equations, possibly limiting generalization and data efficiency.

In contrast, we propose a multi-output kernel function, named *Lagrangian Inspired Polynomial kernel* (LIP), which correlates the different joint torques by exploiting Lagrangian mechanics. Our method is based on two main ideas: first, we model as GPs the kinetic and potential energy of the system. Driven by the fact that the kinetic and potential energy are polynomial functions in a suitable input space, we derive two polynomial kernels that encode this property. Second, inspired by the fact that Lagrangian mechanics derives the dynamics equations as a linear transformation of the Lagrangian function, we obtain the torques GPs by applying a set of linear operators to the GPs of the potential and kinetic energy.

Our contribution is twofold. First, we derive the LIP kernel, a multi-output kernel which encodes the symmetries typical of Lagrangian systems and the polynomial nature of the kinetic and potential energy. Second, we compare the performance of the LIP model against baselines and state-of-the-art algorithms on a real Franka Emika Panda. The collected results show that the LIP estimator outperforms state-of-the-art black-box GP estimators, obtaining better generalization performance. This fact confirms that encoding physical properties in black-box models is a promising strategy to improve model accuracy and data efficiency.

The paper is structured as follows. In Section II we provide a review of robot dynamics and GPR theory. Section III presents our proposed kernel. First, we derive the GP prior for torques from the GP prior for energies by utilizing the principles of Lagrangian mechanics. Then, we present the polynomial kernels that we exploit to model system energies. Section IV outlines the experiments on the Panda robot, while Section V concludes the paper.

II. Background

In this section, we describe the inverse dynamics identification problem, and we concisely review GPR.

A. Inverse dynamics

Consider an n-degrees of freedom (DOF) manipulator composed of $n+1$ links connected by n joints. Let $\boldsymbol{q}_t = \left[q_t^1, \dots, q_t^n\right]^T \in \mathbb{R}^n$ and $\boldsymbol{\tau}_t = \left[\tau_t^1, \dots, \tau_t^n\right]^T \in \mathbb{R}^n$ be the vectors collecting, respectively, the joint positions and generalized torques at time t, where q_t^i and τ_t^i denote, respectively, the joint position and the torque of joint i. Moreover, we denote with $\dot{\boldsymbol{q}}_t$ and $\ddot{\boldsymbol{q}}_t$ the joints velocity and acceleration. In the following, we will denote explicitly the dependence on t only when strictly necessary. The inverse dynamics identification problem consists in identifying the map that relates $\boldsymbol{q}_t, \dot{\boldsymbol{q}}_t, \ddot{\boldsymbol{q}}_t$ with $\boldsymbol{\tau}_t$, given a dataset of input output measures $\mathscr{D}$. Under rigid body assumptions, the dynamics equations are described by the following matrix equation

$$B(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) + \tilde{\boldsymbol{\tau}} = \boldsymbol{\tau}, \tag{1}$$

where $B(\boldsymbol{q})$ is the inertia matrix, while $\boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ and $\boldsymbol{g}(\boldsymbol{q})$ model, respectively, the fictitious forces and gravity. Finally, $\tilde{\boldsymbol{\tau}}$ is the torque due to friction and unknown dynamical effects [2].

B. Gaussian Process Regression for multi-output models

GPR is a principled probabilistic framework for regression problems that allows estimating an unknown function $f : \mathbb{R}^m \to \mathbb{R}^d$ given a dataset of input-output observations $\mathscr{D} = \{X, Y\}$, composed of the input set $X = \{\boldsymbol{x}_1 \dots \boldsymbol{x}_N\}$ and the output set $Y = \{\boldsymbol{y}_1, \dots, \boldsymbol{y}_N\}$. We assume the following measurement model

$$\boldsymbol{y}_i = f(\boldsymbol{x}_i) + \boldsymbol{e}_i, \qquad i = 1, \dots, N, \tag{2}$$

where $\boldsymbol{e}_i$ is a zero-mean Gaussian noise with variance $\Sigma_{e_i} \in \mathbb{R}^d \times \mathbb{R}^d$, i.e., $\boldsymbol{e}_i \sim \mathscr{N}(0, \Sigma_{e_i})$. We assume that Σ_{e_i} is a diagonal matrix, namely, $\Sigma_{e_i} = \text{diag}(\sigma_{e_1}^2, \dots, \sigma_{e_d}^2)$, where $\sigma_{e_i}^2$ denotes the variance of the noise affecting the i-th component of f. By letting $\boldsymbol{y} = \left[\boldsymbol{y}_1^T, \dots, \boldsymbol{y}_N^T\right]^T$ and $\boldsymbol{e} = \left[\boldsymbol{e}_1^T, \dots, \boldsymbol{e}_N^T\right]^T$ we can write

$$\boldsymbol{y} = \begin{bmatrix} \boldsymbol{y}_1 \\ \vdots \\ \boldsymbol{y}_N \end{bmatrix} = \begin{bmatrix} f(\boldsymbol{x}_1) \\ \vdots \\ f(\boldsymbol{x}_N) \end{bmatrix} + \begin{bmatrix} \boldsymbol{e}_1 \\ \vdots \\ \boldsymbol{e}_N \end{bmatrix} = f(X) + \boldsymbol{e}, \tag{3}$$

where the noises $\boldsymbol{e}_1, \dots, \boldsymbol{e}_N$ are assumed independent and identically distributed. It turns out that the variance of $\boldsymbol{e}$ is a block diagonal matrix with equal diagonal blocks, namely

$$\Sigma_e = \text{diag}(\Sigma_{e_1}, \dots, \Sigma_{e_N}),$$

with $\Sigma_{e_1} = \Sigma_{e_2} = \dots = \Sigma_{e_N}$.

The unknown function f is defined a priori as a GP, that is, $f \sim GP(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}'))$, where $m(\cdot) : \mathbb{R}^m \to \mathbb{R}^d$ is the prior mean and $k(\cdot, \cdot) : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}^{d \times d}$ is the prior covariance, also named kernel. The kernel $k(\cdot, \cdot)$ determines the covariance between the values of the unknown function at different input locations, that is, $\text{Cov}[f(\boldsymbol{x}_p), f(\boldsymbol{x}_q)] = k(\boldsymbol{x}_p, \boldsymbol{x}_q)$. For instance, in the scalar case ($d = 1$), a common choice for $k(\cdot, \cdot)$ is the Square Exponential (SE) kernel, defined as

$$k_{SE}(\boldsymbol{x}, \boldsymbol{x}') = \lambda e^{-\|\boldsymbol{x} - \boldsymbol{x}'\|^2_{\Sigma^{-1}}}, \tag{4}$$

where λ and Σ are the kernel hyperparameters.

Under the Gaussian assumption, the posterior distribution of f given $\mathscr{D}$ in a general input location $\boldsymbol{x}$ is still Gaussian, with mean and variance given by the following expressions:

$$\mathbb{E}[f(\boldsymbol{x})|\mathscr{D}] = m(\boldsymbol{x}) + K_{\boldsymbol{x}X}(K_{XX} + \Sigma_e)^{-1}(\boldsymbol{y} - \boldsymbol{m}_X), \tag{5a}$$

$$\text{Cov}[f(\boldsymbol{x})|\mathscr{D}] = k(\boldsymbol{x}, \boldsymbol{x}) - K_{\boldsymbol{x}X}(K_{XX} + \Sigma_e)^{-1}K_{X\boldsymbol{x}}, \tag{5b}$$

where $K_{\boldsymbol{x}X} \in \mathbb{R}^{d \times dN}$ is given by

$$K_{\boldsymbol{x}X} = K_{X\boldsymbol{x}}^T = \left[k(\boldsymbol{x}, \boldsymbol{x}_1), \dots, k(\boldsymbol{x}, \boldsymbol{x}_N)\right], \tag{6}$$

and $K_{XX} \in \mathbb{R}^{dN \times dN}$ is the block matrix

$$K_{XX} = \begin{bmatrix} k(\boldsymbol{x}_1, \boldsymbol{x}_1) & \dots & k(\boldsymbol{x}_1, \boldsymbol{x}_N) \\ \vdots & \ddots & \vdots \\ k(\boldsymbol{x}_N, \boldsymbol{x}_1) & \dots & k(\boldsymbol{x}_N, \boldsymbol{x}_N) \end{bmatrix}. \tag{7}$$

See [13] for a detailed derivation of formulas in (5). The posterior mean (5a) is used as an estimate of f, that is, $\hat{f} = \mathbb{E}[f(\boldsymbol{x})|\mathscr{D}]$, while (5b) is useful to derive confidence intervals of $\hat{f}$.

When GPR is applied to inverse dynamics identification, the inverse dynamics map is treated as an unknown function and modeled a priori as a GP. The GP-input at time t is $\boldsymbol{x}_t = (\boldsymbol{q}_t, \dot{\boldsymbol{q}}_t, \ddot{\boldsymbol{q}}_t)$, while outputs are torques. The standard approach consists in defining the GP prior directly on the inverse dynamics function, by assuming its n components to be conditionally independent given the GP input $\boldsymbol{x}_t$. As a consequence, the overall inverse dynamics identification problem is split into a set of n scalar and independent GPR sub-problems,

$$y_t^i = f^i(\boldsymbol{x}_t) + e_t^i$$

where the i-th torque component $f^i : \mathbb{R}^{3n} \to \mathbb{R}$ is estimated independently of the others as in (5) with $d = 1$ and $\boldsymbol{y} = \boldsymbol{y}^i = [y_1^i \dots y_N^i]^T$, being y_t^i a measure of the i-th torque at time t.

Observe that the conditionally independence assumption is a strong approximation, which might limit generalization and data efficiency. As described in the next section, we propose a multi-output GP model that naturally correlates the different torque dimensions, thus obtaining the following generative model,

$$\boldsymbol{y}_t = f(\boldsymbol{x}_t) + \boldsymbol{e}_t,$$

with $f : \mathbb{R}^{3n} \to \mathbb{R}^n$, where n is the number of DOF of the considered mechanical system, $\boldsymbol{y}_t \in \mathbb{R}^n$ is the vector of torque measurements at time t and $\boldsymbol{e}_t \in \mathbb{R}^n$ is the noise at time t modeled as in Sec. II-B. The estimate of f can now be computed as described in Sec. II-B, with $d = n$.

III. Lagrangian Inspired Polynomial Kernel

In this section, we derive the *Lagrangian Inspired Polynomial* (LIP) kernel. We model the kinetic and potential energies as two different GPs and derive the GP prior of the torques exploiting the laws of Lagrangian mechanics and the properties of GPs under linear operators. First, in Section III-A we introduce a polynomial characterization of the system energies and we accordingly define two polynomial kernel functions. Then, in Section III-B we derive the inverse dynamics GP-models from the GPs of the kinetic and potential energies.

A. Kinetic and potential energy polynomial priors

Let $\mathscr{T}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ and $\mathscr{V}(\boldsymbol{q})$ be the kinetic and potential energy of a n-DOF system of the form (1). We model $\mathscr{T}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ and $\mathscr{V}(\boldsymbol{q})$ as two independent zero-mean GPs with kernel functions $k^{\mathscr{T}}(\boldsymbol{x}, \boldsymbol{x}')$ and $k^{\mathscr{V}}(\boldsymbol{x}, \boldsymbol{x}')$, where $\boldsymbol{x} = (\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$, as defined in Section II-B.

Before deriving the kernel functions $k^{\mathscr{V}}$ and $k^{\mathscr{T}}$ adopted as model prior for $\mathscr{T}$ and $\mathscr{V}$, we introduce some useful notations. Let $\boldsymbol{q}^i$ and $\dot{\boldsymbol{q}}^i$ be the vectors containing the positions and velocities of the joints up to index i, respectively. Then, assume the considered system to be composed of N_r revolute joints and N_p prismatic joints, with $N_r + N_p = n$. We denote with $I_r = \{r_1, \dots, r_{N_r}\}$ and $I_p = \{p_1, \dots, p_{N_p}\}$ the sets containing the revolute and prismatic joints indexes, respectively. We introduce the vectors $\boldsymbol{q}_c$ and $\boldsymbol{q}_s$, containing the cosine and sine of revolute joint positions, and the vector $\boldsymbol{q}_p$, containing the prismatic joint positions. By q_c^b, q_s^b and q_p^b we denote the b-th element of $\boldsymbol{q}_c$, $\boldsymbol{q}_s$ and $\boldsymbol{q}_p$, respectively. Next, let I_r^i (resp. I_p^i) be the subset of I_r (resp. I_p) composed by the indexes lower or equal to i and let us define the vectors $\boldsymbol{q}_c^i$, $\boldsymbol{q}_s^i$, (resp. $\boldsymbol{q}_p^i$) as the restriction of $\boldsymbol{q}_c$, $\boldsymbol{q}_s$ (resp. $\boldsymbol{q}_p$) to I_r^i (resp. I_p^i). To conclude let $\boldsymbol{q}_{cs_b}$ be the vector concatenating the b-th elements of $\boldsymbol{q}_c$ and $\boldsymbol{q}_s$, that is, $\boldsymbol{q}_{cs_b} = [q_c^b, q_s^b]^T$.

We continue our analysis by considering first the design of $k^{\mathscr{V}}$ and then the design of $k^{\mathscr{T}}$.

1) Potential energy: The following proposition establishes that the potential energy is polynomial w.r.t. the set of variables $(\boldsymbol{q}_c, \boldsymbol{q}_s, \boldsymbol{q}_p)$, which is function of $\boldsymbol{q}$.

Proposition 1: The potential energy $\mathscr{V}(\boldsymbol{q})$ is a polynomial function in $(\boldsymbol{q}_c, \boldsymbol{q}_s, \boldsymbol{q}_p)$ of degree not greater than n, such that each element of $\boldsymbol{q}_c$, $\boldsymbol{q}_s$ and $\boldsymbol{q}_p$ appears with degree not greater than 1. Moreover, for any monomial of the aforementioned polynomial, the sum of the degrees of q_c^b and q_s^b is equal or lower than 1, namely, it holds

$$deg(q_c^b) + deg(q_s^b) \leq 1. \tag{8}$$

To comply with the properties introduced by Proposition 1, we define the $k^{\mathscr{V}}(\boldsymbol{x}, \boldsymbol{x}')$ kernel as the product of $N_r + N_p$ inhomogeneous polynomial kernels of degree 1 [13], namely

$$k^{\mathscr{V}}(\boldsymbol{x}, \boldsymbol{x}') = \prod_{b \in I_r} k_{pk}^{(1)}(\boldsymbol{q}_{cs_b}, \boldsymbol{q}'_{cs_b}) \prod_{b \in I_p} k_{pk}^{(1)}(q_p^b, q_p'^b), \tag{9}$$

with the general inhomogeneous polynomial kernel of degree p defined as $k_{pk}^{(p)}(\boldsymbol{x}, \boldsymbol{x}') = (\boldsymbol{x}^T \Sigma_{pk} \boldsymbol{x}' + \sigma_{pk})^p$, where Σ_{pk} and σ_{pk} are the hyperparameters.

2) Kinetic energy: The kinetic energy is the sum of the kinetic energies relative to each link, that is, $\mathscr{T}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \sum_{i=1}^n \mathscr{T}_i(\boldsymbol{q}, \dot{\boldsymbol{q}})$, where $\mathscr{T}_i(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is the kinetic energy of Link i. The following proposition establishes that $\mathscr{T}_i$ is polynomial w.r.t. the set of variables $(\boldsymbol{q}_c^i, \boldsymbol{q}_s^i, \boldsymbol{q}_p^i, \dot{\boldsymbol{q}}^i)$, which are functions of $\boldsymbol{q}^i$ and $\dot{\boldsymbol{q}}^i$.

Proposition 2: The kinetic energy of the i-th link $\mathscr{T}_i(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is a polynomial function in $(\boldsymbol{q}_c^i, \boldsymbol{q}_s^i, \boldsymbol{q}_p^i, \dot{\boldsymbol{q}}^i)$ of degree not greater than $2i + 2$, such that each element of $\boldsymbol{q}_c^i$, $\boldsymbol{q}_s^i$, $\boldsymbol{q}_p^i$ and $\dot{\boldsymbol{q}}^i$ appears with degree not greater than 2. Moreover, in each monomial, the sum of the degrees of q_c^b and q_s^b is lower than or equal to 2, namely

$$deg(q_c^b) + deg(q_s^b) \leq 2. \tag{10}$$

To comply with the constraints and properties stated in the above Proposition, we define the $k^{\mathscr{T}}(\boldsymbol{x}, \boldsymbol{x}')$ kernel as the product of i inhomogeneous polynomial kernels of degree 2, and 1 homogeneous kernel of degree 2, namely

$$\begin{aligned} k_i^{\mathscr{T}}(\boldsymbol{x}, \boldsymbol{x}') = & k_{hpk}^{(2)}(\dot{\boldsymbol{q}}^i, \dot{\boldsymbol{q}}'^i) \cdot \\ & \prod_{b \in I_r^i} k_{pk}^{(2)}(\boldsymbol{q}_{cs_b}, \boldsymbol{q}'_{cs_b}) \cdot \prod_{b \in I_p^i} k_{pk}^{(2)}(q_p^b, q_p'^b), \end{aligned} \tag{11}$$

where the homogeneous polynomial kernel of degree p is defined as the inhomegeneous one but with $\sigma_{pk} = 0$.

We finally define kinetic energy kernel as $k^{\mathscr{T}}(\boldsymbol{x}, \boldsymbol{x}') = \sum_{i=1}^n k_i^{\mathscr{T}}(\boldsymbol{x}, \boldsymbol{x}')$.

B. From energies to torques GP models

GPR cannot be applied directly on $\mathscr{T}$ and $\mathscr{V}$, as these quantities are not directly measured. However, starting from the prior on the two energies, we can derive a GP prior for the torques leveraging on Lagrangian mechanics. Lagrangian mechanics derives the inverse dynamics equations in (1) (with $\tilde{\boldsymbol{\tau}} = 0$) as the solution of a set of differential equations of the Lagrangian function $\mathscr{L} = \mathscr{T}(\boldsymbol{q}, \dot{\boldsymbol{q}}) - \mathscr{V}(\boldsymbol{q})$ [2]. The i-th differential equation of (1) is

$$\frac{d\mathscr{L}}{dt}\left(\frac{\partial \mathscr{L}}{\partial \dot{q}^i}\right) - \frac{\partial \mathscr{L}}{\partial q^i} = \tau^i, \tag{12}$$

where q^i, $\dot{q}^i$, and τ^i are, respectively, the i-th component of $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, and $\boldsymbol{\tau}$. Exploiting the chain rule, equation (12) can be rewritten as

$$\sum_{j=1}^n \left(\frac{\partial^2 \mathscr{L}}{\partial \dot{q}^i \partial \dot{q}^j}\ddot{q}^j + \frac{\partial^2 \mathscr{L}}{\partial \dot{q}^i \partial q^j}\dot{q}^j\right) - \frac{\partial \mathscr{L}}{\partial q^i} = \tau^i. \tag{13}$$

It is convenient to introduce the linear operator $\mathscr{G}_i$, that maps $\mathscr{L}$ in the left-hand side of (13), namely $\tau^i = \mathscr{G}_i \mathscr{L}$. In compact form we can write $\boldsymbol{\tau} = \mathscr{G}\mathscr{L} = [\mathscr{G}_1\mathscr{L} \ \dots \ \mathscr{G}_n\mathscr{L}]^T$, which defines the linear operator $\mathscr{G}$ mapping $\mathscr{L}$ into $\boldsymbol{\tau}$.

Since we modeled $\mathscr{T}$ and $\mathscr{V}$ as independent GPs, $\mathscr{L}$ is a GP itself. The sum of two independent GPs, indeed, is a still a GP, and its kernel is the sum of the kernels [13], namely,

$$\mathscr{L} \sim GP(0, k^{\mathscr{L}}(\boldsymbol{x}, \boldsymbol{x}')), \tag{14a}$$

$$k^{\mathscr{L}}(\boldsymbol{x}, \boldsymbol{x}') = k^{\mathscr{T}}(\boldsymbol{x}, \boldsymbol{x}') + k^{\mathscr{V}}(\boldsymbol{x}, \boldsymbol{x}'). \tag{14b}$$

Moreover, note that the inverse dynamics map is the result of the application of the linear operator $\mathscr{G}$ to the GP defined in (14a) and (14b) modeling $\mathscr{L}$. As the GPs are close under linear operators [22], also the inverse dynamics is a GP in our setup. In detail, it turns out that $\boldsymbol{\tau} \sim GP(0, k^{\tau}(\boldsymbol{x}, \boldsymbol{x}'))$ with the multi-output kernel $k^{\tau} \in \mathbb{R}^{n \times n}$ defined as

$$k^{\tau}(\boldsymbol{x}, \boldsymbol{x}') = \begin{bmatrix} \mathscr{G}_1 \mathscr{G}_1' k^{\mathscr{L}}(\boldsymbol{x}, \boldsymbol{x}') & \dots & \mathscr{G}_1 \mathscr{G}_n' k^{\mathscr{L}}(\boldsymbol{x}, \boldsymbol{x}') \\ \vdots & \ddots & \vdots \\ \mathscr{G}_n \mathscr{G}_1' k^{\mathscr{L}}(\boldsymbol{x}, \boldsymbol{x}') & \dots & \mathscr{G}_n \mathscr{G}_n' k^{\mathscr{L}}(\boldsymbol{x}, \boldsymbol{x}') \end{bmatrix}, \tag{15}$$

where $\mathscr{G}_i'$ is the same linear operator as $\mathscr{G}_i$ but applied to the input variable $\boldsymbol{x}'$. We refer the interested reader to [22] for a more detailed discussion on GPs and linear operators.

To summarize, the LIP kernel consists of the multi-output torque prior k^{τ} expressed in (15), where $k^{\mathscr{L}}$ is defined in (14b) and $k^{\mathscr{V}}$ and $k^{\mathscr{T}}$ are the polynomial kernels presented in Section III-A.

IV. Experimental results

We tested proposed estimator on a Franka Emika Panda robot, which is a 7 DOF manipulator with only revolute joints. We compared the performance of the LIP model with those of three different state-of-the-art GP-based estimators. Two of them are obtained with the standard approach, namely modeling joint torques as independent GPs. One is based on the Square Exponential (SE) kernel in (4) , while the other one is based on the GIP kernel presented in [18]. The third solution, instead, exploits the multi-output kernel presented in [20], hereafter denoted as LSE. The LSE estimator models directly the Lagrangian function, instead of modeling the kinetic and potential energy separately. The Lagrangian is modeled using a SE kernel defined on an augmented input space obtained by substituting the positions of the revolute joints with their sine and cosine. All the considered estimators have been implemented in Python, using the functionalities provided by the library PyTorch [23]. The hyperparameters of all the GP-based estimators have been optimized by marginal likelihood maximization [13]. In order to compensate the friction affecting the real system, we integrate the considered estimators with a GP model linear in the features $\dot{\boldsymbol{q}}$ and sign$(\dot{\boldsymbol{q}})$.

On the real Panda robot, we collected joint positions, velocities, and torques through the ROS interface provided by the robot manufacturer. To mitigate the effect of measurement noise, we filtered the collected positions, velocities, and torques with a low pass filter with a cut-off frequency of $4\,\mathrm{Hz}$. We obtained joint accelerations from joint velocities by means of acausal differentiation.

We collected 10 training and 16 test datasets, following different sum of sinusoids reference trajectories. In detail, each dataset is obtained imposing to the i-th joint a trajectory defined as

$$q_i(t) = \sum_{l=1}^{N_s} \frac{a}{\omega_f\, l} \sin(\omega_f\, l\, t) - \frac{b}{\omega_f\, l} \cos(\omega_f\, l\, t), \tag{16}$$

with $\omega_f = 0.02\,\mathrm{rad/s}$, while a and b are sampled from a uniform distribution ranging in $[-c,\, c]$, with c chosen in order to respect the limits on joint position, velocity and acceleration. The test datasets has a wider range of frequencies, $N_s = 100$, than the training trajectories, $N_s = 50$, to stress the generalization properties. All the generated datasets are composed of 500 samples, collected at a frequency of $10Hz$. The GP-based estimators are learned in each training trajectory and tested on each and every of the 16 test trajectories. Figure 1 reports the distribution of the normalized Mean Sqaured Error (nMSE) for each joint, on the test datasets. The nMSE is defined as

$$nMSE(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{1}{N} \sum_{j=1}^{N} (y_j - \hat{y}_j)^2 / Var(\boldsymbol{y}),$$

where $\boldsymbol{y}$ is the vector of measures and $\hat{\boldsymbol{y}}$ contains the corresponding estimates. The estimator with the LIP kernel predicts the joint torques better than the other GP estimators on each and every joint, which proves the advantages of our approach.

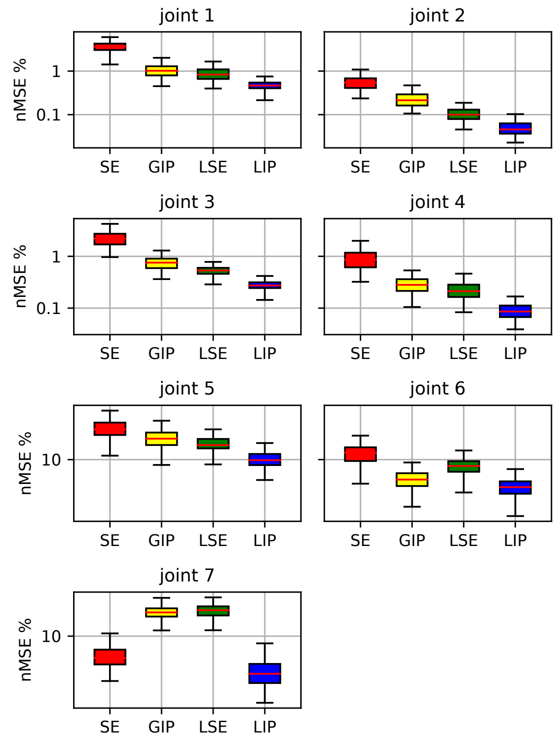

Fig. 1: Box plot of the torque nMSE obtained with the experiment on the PANDA robot described in section IV.

V. Conclusions

In this work we presented the LIP kernel, a novel multi-output kernel designed to model the kinetic and potential energy of robotics systems. Leveraging on the laws of Lagrangian mechanics and on a polynomial characterization of potential and kinetic energy, the proposed kernel provides black-box inverse dynamics models which respect the symmetries imposed by the Lagrange's equations, while improving generalization and data efficiency. The method has been validated on a real setup involving a Franka Emika Panda robot. The collected results showed that it outperforms state-of-the-art black-box estimators based on Gaussian Processes in term of accuracy and generalization.

References

[1] P. K. Khosla and T. Kanade, "Experimental evaluation of nonlinear feedback and feedforward control schemes for manipulators," *The International Journal of Robotics Research*, vol. 7, no. 1, pp. 18–28, 1988.

[2] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. Springer Publishing Company, Incorporated, 2010.

[3] A. Dalla Libera, E. Tosello, G. Pillonetto, S. Ghidoni, and R. Carli, "Proprioceptive robot collision detection through gaussian process regression," in *2019 American Control Conference (ACC)*, 2019, pp. 19–24.

[4] S. Haddadin, A. De Luca, and A. Albu-Schäffer, "Robot collisions: A survey on detection, isolation, and identification," *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1292–1312, 2017.

[5] A. De Santis, B. Siciliano, A. De Luca, and A. Bicchi, "An atlas of physical human–robot interaction," *Mechanism and Machine Theory*, vol. 43, no. 3, pp. 253–270, 2008.

[6] J. Hollerbach, W. Khalil, and M. Gautier, "Model identification," in *Springer Handbook of Robotics*. Springer, 2008, pp. 321–344.

[7] C. Gaz, M. Cognetti, A. Oliva, P. Robuffo Giordano, and A. De Luca, "Dynamic identification of the franka emika panda robot with retrieval of feasible parameters using penalty-based optimization," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4147–4154, 2019.

[8] C. D. Sousa and R. Cortesao, "Physical feasibility of robot base inertial parameter identification: A linear matrix inequality approach," *The International Journal of Robotics Research*, vol. 33, no. 6, pp. 931–944, 2014.

[9] J. Kwon, K. Choi, and F. C. Park, "Kinodynamic model identification: A unified geometric approach," *IEEE Transactions on Robotics*, vol. 37, no. 4, pp. 1100–1114, 2021.

[10] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[11] E. Rueckert, M. Nakatenus, S. Tosatto, and J. Peters, "Learning inverse dynamics models in o(n) time with lstm networks," in *Humanoids*, Nov 2017, pp. 811–816.

[12] A. S. Polydoros, L. Nalpantidis, and V. Krüger, "Real-time deep learning of robotic manipulator inverse dynamics," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 3442–3448.

[13] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer school on machine learning*. Springer, 2003, pp. 63–71.

[14] J. Schreiter, P. Englert, D. Nguyen-Tuong, and M. Toussaint, "Sparse gaussian process regression for compliant, real-time robot control," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 2586–2591.

[15] A. Gijsberts and G. Metta, "Incremental learning of robot dynamics using random features," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 951–956.

[16] D. Nguyen-Tuong, M. Seeger, and J. Peters, "Model learning with local gaussian process regression," *Advanced Robotics*, vol. 23, no. 15, pp. 2015–2034, 2009.

[17] S. Rezaei-Shoshtari, D. Meger, and I. Sharf, "Cascaded gaussian processes for data-efficient robot dynamics learning," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 6871–6877.

[18] A. Dalla Libera and R. Carli, "A data-efficient geometrically inspired polynomial kernel for robot inverse dynamic," *IEEE Robotics and Automation Letters*, vol. 5, no. 1, pp. 24–31, 2019.

[19] M. Lutter, C. Ritter, and J. Peters, "Deep lagrangian networks: Using physics as model prior for deep learning," in *7th International Conference on Learning Representations (ICLR)*. ICLR, May 2019. [Online]. Available: https://openreview.net/pdf?id=BklHpjCqKm

[20] 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.

[21] G. Evangelisti and S. Hirche, "Physically consistent learning of conservative lagrangian systems with gaussian processes," in *2022 IEEE 61st Conference on Decision and Control (CDC)*. IEEE, 2022, pp. 4078–4085.

[22] S. Särkkä, "Linear operators and stochastic partial differential equations in gaussian process regression," in *International Conference on Artificial Neural Networks*. Springer, 2011, pp. 151–158.

[23] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf