

Learning Positive Definite Inertia Matrices in Black-Box Inverse Dynamics Models via Gaussian Processes: a Constraint Learning Approach

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

TR2025-090 June 24, 2025

Abstract

Inverse dynamics models are crucial for robotic control, but traditional physics-based models require precise system parameters that can be difficult to obtain. While data-driven black-box models offer a valid alternative, they usually lack physical plausibility, limiting their use with standard control methods. This paper presents a method for black-box inverse dynamics identification using Gaussian Processes Regression (GPR) that promotes physical consistency, by enforcing the positive definiteness of the inertia matrix. In particular, we unveil how to estimate the inertia matrix elements from black-box models, and we integrate positivity constraints into the empirical risk minimization problem. Experimental validation demonstrates that our approach significantly improves physical consistency with minimal loss in estimation accuracy, outperforming unconstrained models that may yield non-physical behaviors and consequently poor control performances.

European Control Conference (ECC) 2025

© 2025 MERL. This work may not be copied or reproduced in whole or in part for any commercial purpose. Permission to copy in whole or in part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of Mitsubishi Electric Research Laboratories, Inc.; an acknowledgment of the authors and individual contributions to the work; and all applicable portions of the copyright notice. Copying, reproduction, or republishing for any other purpose shall require a license with payment of fee to Mitsubishi Electric Research Laboratories, Inc. All rights reserved.

Learning Positive Definite Inertia Matrices in Black-Box Inverse Dynamics Models via Gaussian Processes: a Constraint Learning Approach

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

Abstract—Inverse dynamics models are crucial for robotic control, but traditional physics-based models require precise system parameters that can be difficult to obtain. While data-driven black-box models offer a valid alternative, they usually lack physical plausibility, limiting their use with standard control methods. This paper presents a method for black-box inverse dynamics identification using Gaussian Processes Regression (GPR) that promotes physical consistency, by enforcing the positive definiteness of the inertia matrix. In particular, we unveil how to estimate the inertia matrix elements from black-box models, and we integrate positivity constraints into the empirical risk minimization problem. Experimental validation demonstrates that our approach significantly improves physical consistency with minimal loss in estimation accuracy, outperforming unconstrained models that may yield non-physical behaviors and consequently poor control performances.

I. INTRODUCTION

Deriving accurate inverse dynamics models is a crucial but challenging task in robotics. Traditionally, these models are built from first-order principles of physics, with unknown parameters estimated using system identification techniques. While effective, this approach relies on precise knowledge of system parameters and dynamics, which is often difficult to obtain in practice. To overcome this limitation, learning inverse dynamics models directly from data, in a black-box fashion, represents a viable alternative. Several black-box solutions have been proposed in recent years, mostly based on Neural Networks (NN) [1]–[3] and Gaussian Process Regression (GPR) [4]–[7].

Ensuring physical consistency in data-driven inverse dynamics models is crucial. Properties such as energy conservation and symmetries originate from first principles of physics. The positive definiteness of the inertia matrix is particularly essential in several tasks, for example when deriving the forward dynamics from the inverse [8]. Such property is equally important for control applications, as traditional strategies like feedback linearization [9] require it for accurate and robust performance; non-positive definiteness can lead to divergent behaviors. However, guaranteeing

these properties when learning from limited and noisy data is challenging. This difficulty exists even in classical inverse dynamics identification approaches based on first principles; for instance, see solutions proposed in [10] to learn feasible parameters.

In the context of black-box estimators, the proposed solutions try to improve physics consistency by enforcing particular model structures. For example, in [11] the authors propose a novel NN structure that resembles the inverse dynamics equations. To impose the positivity of the inertia matrix, they parametrize its Cholesky decomposition as a NN and they add a small constant to its diagonal.

The inverse dynamics equations’ structure has also inspired several GPR solutions [12]–[16]. In particular, in [16], the authors define a GP prior on the potential energy and on the elements of the inertia matrix. As a result, they provide probabilistic guarantees on the inertia positivity, at the price of a potentially high number of parameters to train, which could compromise the model accuracy, especially in high dimensional settings with limited data availability.

In this paper, we propose a novel constrained learning strategy to promote the positivity of the inertia matrix in black-box inverse dynamics identification using GPR. Recognizing that the GPR solution arises from minimizing a regularized objective function, we modify the standard optimization problem — which typically focuses solely on minimizing the empirical risk — by incorporating constraints that enforce the positive definiteness of the inertia matrix.

Our contribution is threefold. First, we provide a method to unveil how to estimate the inertia matrix from black-box GPR models of the inverse dynamics, showing that these estimates can be written as a linear combination of the GPR posterior coefficients. Second, we modify the standard empirical risk minimization problem by incorporating constraints that enforce the positive definiteness of the inertia matrix. Third, we demonstrate the advantage of the proposed method both in terms of estimation and control on a 7 degrees-of-freedom (DOF) manipulator.

Experimental results show that the addition of the constraints is an effective solution to promote the positivity of the inertia matrix both on training and test trajectories, at the price of a minimal estimation accuracy loss. Moreover, we show that the unconstrained model, despite higher estimation accuracy, leads to unstable control due to non-positive inertia matrix estimates. In contrast, the proposed model guarantees precise and accurate tracking performance.

Giulio Giacomuzzo and Alberto Dalla Libera were supported by PNRR research activities of the consortium iNEST (Interconnected North-East Innovation Ecosystem) funded by the European Union Next-GenerationEU (Piano Nazionale di Ripresa e Resilienza (PNRR) – Missione 4 Componente 2, Investimento 1.5 – D.D. 1058 23/06/2022, ECS_00000043).

Giulio Giacomuzzo, Ruggero Carli and Alberto Dalla Libera are with the Department of Information Engineering, University of Padova, 35131 Padova, Italy (email: {giacomuzzo, carlirug, dallaliber}@dei.unipd.it).

Diego Romeres is with the Mitsubishi Electric Research Lab (MERL), Cambridge, MA 02139 USA (email: romeres@merl.com).

II. BACKGROUND

This section reviews background notions on the inverse dynamics model and its identification via GPR.

A. Inverse dynamics

Consider an n -degrees of freedom mechanical systems, and let $\mathbf{q}_t = [q_t^1 \dots q_t^n]^T \in \mathbb{R}^n$ be its generalized coordinates at time t . Accordingly, $\dot{\mathbf{q}}_t$ and $\ddot{\mathbf{q}}_t$ are, respectively, the velocity and acceleration vector at time t , while $\boldsymbol{\tau}_t$ is the torque vector; in the following, we will point out time-dependency only when necessary. Under rigid body assumptions, the dynamics equations of a mechanical system are described by the following matrix equation

$$M(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) + \boldsymbol{\varepsilon} = \boldsymbol{\tau}, \quad (1)$$

where $M(\mathbf{q})$ is the inertia matrix, $\mathbf{c}(\mathbf{q}, \dot{\mathbf{q}})$ and $\mathbf{g}(\mathbf{q})$ account for the contributions of fictitious forces and gravity, respectively, and $\boldsymbol{\varepsilon}$ is the torque due to friction and unknown dynamical effects. We refer the interested reader to [17] for a complete and detailed description and derivation of (1).

The inverse dynamics identification problem aims at learning from data a function that maps $\mathbf{x} = [\mathbf{q}^T \dot{\mathbf{q}}^T \ddot{\mathbf{q}}^T]^T$ in the correspondent $\boldsymbol{\tau}$. Importantly, the terms in (1) are subject to constraints necessary to guarantee global properties of the physical system, such as the positivity of the kinetic energy or energy conservation. In particular, the inertia matrix $M(\mathbf{q})$ must be positive definite, namely $M(\mathbf{q}) > 0$, that is $M(\mathbf{q}) = M(\mathbf{q})^T$ with all positive eigenvalues. Deriving models that estimate inertia matrices $\hat{M}(\mathbf{q}) > 0$ is crucial for control applications: non-symmetric or non-positive $\hat{M}(\mathbf{q})$ could compromise the stability of traditional control strategy, such as feedback linearization [17]. Indeed, as shown in our experiments, accurate black-box inverse dynamics estimators combined with traditional control strategies can lead to unstable controllers if $\hat{M}(\mathbf{q}) < 0$.

B. GPR for inverse dynamics identification

Several solutions proposed for inverse dynamics identification rely on GPR. In these approaches, (1) is modeled through an unknown function $f : \mathbb{R}^{3n} \rightarrow \mathbb{R}^n$ that takes as input $\mathbf{x} = (\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ and returns the correspondent $\boldsymbol{\tau}$. The function is learned from an input-output dataset $\mathcal{D} = \{X, Y\}$, where $X = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$, while the output $Y = \{\mathbf{y}_1, \dots, \mathbf{y}_N\}$ collects the $\boldsymbol{\tau}_1, \dots, \boldsymbol{\tau}_N$ measures.

GPR assumes the following probabilistic model

$$\mathbf{y}_i = f(\mathbf{x}_i) + \mathbf{e}_i \quad (2)$$

where \mathbf{e}_i is a zero-mean Gaussian noise with variance $\Sigma_{e_i} \in \mathbb{R}^n \times \mathbb{R}^n$. We consider the measurement noise independent and identically distributed, which is a reasonable choice in the inverse dynamics setup; this means that $\Sigma_{e_i} = \sigma_e^2 I_n$, where I_n is the n -dimensional identity matrix. Thus, by concatenating (2) applied to all the data in \mathcal{D} , we get

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

where noises $\mathbf{e}_1, \dots, \mathbf{e}_N$ are Gaussians independent and identically distributed, with zero mean and variance $\Sigma_e = \sigma_e^2 I_{nN}$.

The function f is defined a priori as a GP, namely, $f \sim GP(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$, where $m(\cdot) : \mathbb{R}^{3n} \rightarrow \mathbb{R}^n$ is the prior mean and $k(\cdot, \cdot) : \mathbb{R}^{3n} \times \mathbb{R}^{3n} \rightarrow \mathbb{R}^{n \times n}$ is the kernel function. In GPR, the kernel determines the GP covariance, so that $\text{Cov}[f(\mathbf{x}_p), f(\mathbf{x}_q)] = k(\mathbf{x}_p, \mathbf{x}_q)$. In the following, we will assume $m(\cdot) = 0$, which is a common choice in the black-box scenario, where no strong prior knowledge of the system dynamics is available.

Under the Gaussian assumption, the posterior distribution of f given \mathcal{D} in a general input location \mathbf{x} is still a Gaussian distribution, with mean and variance given by

$$\mathbb{E}[f(\mathbf{x})|\mathcal{D}] = K_{\mathbf{x}X}(K_{XX} + \Sigma_e)^{-1}\mathbf{y}, \quad (4a)$$

$$\text{Cov}[f(\mathbf{x})|\mathcal{D}] = k(\mathbf{x}, \mathbf{x}) - K_{\mathbf{x}X}(K_{XX} + \Sigma_e)^{-1}K_{X\mathbf{x}}, \quad (4b)$$

where matrix $K_{\mathbf{x}X} \in \mathbb{R}^{n \times nN}$ is

$$K_{\mathbf{x}X} = K_{X\mathbf{x}}^T = [k(\mathbf{x}, \mathbf{x}_1), \dots, k(\mathbf{x}, \mathbf{x}_N)], \quad (5)$$

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

$$K_{XX} = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & \dots & k(\mathbf{x}_1, \mathbf{x}_N) \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}_N, \mathbf{x}_1) & \dots & k(\mathbf{x}_N, \mathbf{x}_N) \end{bmatrix}. \quad (6)$$

See [4] for a detailed derivation of formulas in (4). Since the posterior is Gaussian distributed, the posterior mean (4a) corresponds to the Maximum a Posteriori Estimate of f , and is used as an estimator of the inverse dynamics, while the posterior variance (4b) measures the estimation confidence.

The GP estimate in (4a) can be rewritten as

$$\mathbb{E}[f(\mathbf{x})|\mathcal{D}] = K_{\mathbf{x}X}\boldsymbol{\alpha}^* := \hat{f}(\mathbf{x}) \quad (7)$$

$$\boldsymbol{\alpha}^* := (K_{XX} + \Sigma_e)^{-1}\mathbf{y} \quad (8)$$

where $\boldsymbol{\alpha}^*$ are denominated as posterior coefficients. For convenience, we introduced the multi-output GPR framework. However, it is worth mentioning that several solutions rely on single-output GPs to estimate the inverse dynamics, namely, f is learned using n independent GPs, one for each output. This setup is included in the GPR formulation introduced so far by defining the kernel function as

$$k(\mathbf{x}_p, \mathbf{x}_q) = \text{diag}(k^1(\mathbf{x}_p, \mathbf{x}_p) \dots k^n(\mathbf{x}_p, \mathbf{x}_q)), \quad (9)$$

where each $k^i(\mathbf{x}_p, \mathbf{x}_q) : \mathbb{R}^{3n} \rightarrow \mathbb{R}$ is a distinct single-output kernel.

C. Kernel selection for inverse dynamics models

Kernel selection plays an important role in GPR for inverse dynamics identification. In the case of the single-output framework, several options are available. A common choice is the Radial Basis Function (RBF) kernel, which defines the covariance between samples based on the distance between their input locations. More formally

$$k_{RBF}(\mathbf{x}, \mathbf{x}') = \lambda e^{-\|\mathbf{x} - \mathbf{x}'\|_{\Sigma}^2}, \quad (10)$$

where λ and Σ are the kernel hyperparameters.

The direct definition of multi-output kernels is challenging due to the complexity of explicitly specifying also the correlation between the different output dimensions. Typically, multi-output kernels, also named multi-task GP in the literature, are derived as linear combinations of standard scalar kernels. See, for instance, [18]. Recently, several solutions proposed to derive a multi-output kernel for inverse dynamics identification by exploiting Lagrange's equations, thus embedding in the GP model physical properties, with potential benefits as concerns consistency and data-efficiency. In the remainder of this section, we briefly review the Lagrangian Inspired Polynomial (LIP) kernel [15], which we will use in our experiments.

Instead of directly modeling joint torques, the LIP kernel defines a prior on the kinetic and potential energies of the system. Let $\mathcal{T}(\mathbf{q}, \dot{\mathbf{q}})$ and $\mathcal{V}(\mathbf{q})$ be, respectively, the kinetic and potential energy of a n -DOF system of the form (1). $\mathcal{T}(\mathbf{q}, \dot{\mathbf{q}})$ and $\mathcal{V}(\mathbf{q})$ are assumed to be two independent zero-mean GPs with covariances determined by the kernel functions $k^{\mathcal{T}}(\mathbf{x}, \mathbf{x}')$ and $k^{\mathcal{V}}(\mathbf{x}, \mathbf{x}')$, that is

$$\mathcal{T} \sim GP(0, k^{\mathcal{T}}(\mathbf{x}, \mathbf{x}')), \quad (11a)$$

$$\mathcal{V} \sim GP(0, k^{\mathcal{V}}(\mathbf{x}, \mathbf{x}')). \quad (11b)$$

where $\mathbf{x} = (\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ as in the previous Section. The GP prior on \mathcal{T} and \mathcal{V} cannot be used directly in GPR to compute posterior distributions since kinetic and potential energies are not measured. However, starting from the prior on the two energies, a GP prior for the torques can be computed by relying on Lagrangian mechanics. Lagrangian mechanics states that the inverse dynamics equations in (1) (with $\varepsilon = 0$), also named Lagrange's equations, are the solution of a set of differential equations of the Lagrangian function $\mathcal{L} = \mathcal{T}(\mathbf{q}, \dot{\mathbf{q}}) - \mathcal{V}(\mathbf{q})$ [17]. The i -th differential equation of (1) is

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

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

$$\mathcal{G}_i \mathcal{L} = \sum_{j=1}^n \left(\frac{\partial^2 \mathcal{L}}{\partial \dot{q}^i \partial \dot{q}^j} \ddot{q}^j + \frac{\partial^2 \mathcal{L}}{\partial \dot{q}^i \partial q^j} \dot{q}^j \right) - \frac{\partial \mathcal{L}}{\partial q^i} = \tau^i, \quad (13)$$

where we also introduced the linear operator \mathcal{G}_i , that maps \mathcal{L} in the left-hand side of (13). In view of the definition of \mathcal{G}_i , the inverse dynamics in (1) can be written as

$$\boldsymbol{\tau} = \mathcal{G} \mathcal{L} = [\mathcal{G}_1 \mathcal{L} \dots \mathcal{G}_n \mathcal{L}]^T. \quad (14)$$

where the above equality defines the linear operator \mathcal{G} mapping \mathcal{L} into $\boldsymbol{\tau}$.

Notice that, in such a framework, \mathcal{L} is a GP, since \mathcal{T} and \mathcal{V} are two independent GPs. The sum of two independent GPs is indeed a GP itself, with the kernel being the sum of the kernels [4], namely,

$$\mathcal{L} \sim GP(0, k^{\mathcal{L}}(\mathbf{x}, \mathbf{x}')), \quad (15a)$$

$$k^{\mathcal{L}}(\mathbf{x}, \mathbf{x}') = k^{\mathcal{T}}(\mathbf{x}, \mathbf{x}') + k^{\mathcal{V}}(\mathbf{x}, \mathbf{x}'). \quad (15b)$$

Equation (14) shows that the inverse dynamics map is the result of the application of the linear operator \mathcal{G} to the Lagrangian \mathcal{L} , that, in the LIP framework, is modeled as the GP defined in (15a) and (15b). It turns out that, under mild assumptions satisfied in this framework, linear transformations of GPs are still GPs, with mean and covariance that can be derived by applying the linear operator to the starting mean and covariance, see [19]. As a consequence, within the LIP framework, the inverse dynamics is itself a zero mean GP, with covariance function given by

$$k^{\boldsymbol{\tau}}(\mathbf{x}, \mathbf{x}') = \begin{bmatrix} \mathcal{G}_1 \mathcal{G}_1' k^{\mathcal{L}}(\mathbf{x}, \mathbf{x}') & \dots & \mathcal{G}_1 \mathcal{G}_n' k^{\mathcal{L}}(\mathbf{x}, \mathbf{x}') \\ \vdots & \ddots & \vdots \\ \mathcal{G}_n \mathcal{G}_1' k^{\mathcal{L}}(\mathbf{x}, \mathbf{x}') & \dots & \mathcal{G}_n \mathcal{G}_n' k^{\mathcal{L}}(\mathbf{x}, \mathbf{x}') \end{bmatrix}, \quad (16)$$

where \mathcal{G}_j' is the same operator as \mathcal{G}_j but applied to $k^{\mathcal{L}}(\mathbf{x}, \mathbf{x}')$ as function of \mathbf{x}' . In details, the notation $\mathcal{G}_i \mathcal{G}_j' k^{\mathcal{L}}(\mathbf{x}, \mathbf{x}')$ means that \mathcal{G}_j' is first applied to $k^{\mathcal{L}}(\mathbf{x}, \mathbf{x}')$ assuming \mathbf{x} constant and then \mathcal{G}_i is applied to the obtained result assuming \mathbf{x}' constant.

Regarding the choice of the kernels $k^{\mathcal{T}}(\mathbf{x}, \mathbf{x}')$ and $k^{\mathcal{V}}(\mathbf{x}, \mathbf{x}')$, the LIP kernel considers custom polynomial kernels, specifically designed to induce a limited set of basis functions that generate the inverse dynamics equations. Due to space limitations, we refer the interested reader to [15] for a deeper and more detailed derivation of the LIP kernel.

It is worth mentioning that, in the literature, there are approaches similar to the LIP. These approaches consider more general kernels, such as the RBF, see for example [14], [16]. However, thanks to the combination of Lagrange's equation embedding and the selection of tailored basis functions, the LIP kernel showed to be far more accurate than all the state-of-the-art black-box alternatives presented so far, with performance comparable to first-order principles models also in high-dimensional scenarios with limited data availability.

III. METHOD

This Section presents the constrained learning method we propose to promote the positivity of the inertia matrix. First, in Section III-A, we present possible ways of estimating the inertia matrix as a function of the posterior coefficients. Then, in Section III-B we describe the constrained optimization problem we propose.

A. Inertia Matrix estimates with black-box models

In the context of black-box data-driven models, the inverse dynamics is a function $\hat{\boldsymbol{\tau}}(\cdot) : \mathbb{R}^{3n} \rightarrow \mathbb{R}^n$ that takes as input the state $\mathbf{x} = (\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ and outputs the torques $\hat{\boldsymbol{\tau}}$, and it is learned without considering the structure described by the Lagrange's equations in (1). In this section, we describe how it is possible to unveil the inertia matrix estimate from the black-box model of the inverse dynamics.

A very general method that can be applied to any black-box estimator can be derived by inspection from (1). Due to space limitations, we refer the interested reader to [9].

In the case of multi-output kernels derived with the Lagrangian approach described at the end of section II-C,

a more elegant method allows us to estimate the inertia matrix exploiting the properties of linear transformations and GPs. To describe such a solution, we start by noting that the element in position ij of the inertia matrix M is

$$M^{ij}(\mathbf{q}) = \frac{\partial^2 \mathcal{F}(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{q}^i \partial \dot{q}^j} =: \mathcal{G}_{M^{ij}} \mathcal{F}(\mathbf{x}), \quad (17)$$

by definition of the kinetic energy. We refer to [17] for the derivation of (17). Additionally, we introduced the definition of the linear operator $\mathcal{G}_{M^{ij}}$, that maps the kinetic energy \mathcal{F} to M^{ij} . Notice that, as \mathcal{F} is modeled as a zero-mean GP, and exploiting the properties of GPs under linear operators [19], also M^{ij} is a GP. In particular, the covariance between M^{ij} and τ at general input locations \mathbf{x} and \mathbf{x}' can be computed as

$$\begin{aligned} \text{Cov}[M^{ij}(\mathbf{x}), \tau(\mathbf{x}')] &= \text{Cov}[\mathcal{G}_{M^{ij}} \mathcal{F}(\mathbf{x}), \mathcal{G} \mathcal{F}(\mathbf{x}')] \\ &= \mathcal{G}_{M^{ij}} \mathcal{G} k^{\mathcal{F}}(\mathbf{x}, \mathbf{x}') \\ &=: k^{M^{ij} \tau}(\mathbf{x}, \mathbf{x}'), \end{aligned} \quad (18)$$

where \mathcal{G} is defined in (14). This result implies that we can estimate M^{ij} at any input location \mathbf{x} , given measures \mathbf{y} of τ , as

$$\hat{M}^{ij}(\mathbf{x}) = K_{\mathbf{x}X}^{M^{ij} \tau} (K_{XX} + \Sigma_e)^{-1} \mathbf{y} = K_{\mathbf{x}X}^{M^{ij} \tau} \boldsymbol{\alpha}^*, \quad (19)$$

with $K_{\mathbf{x}X}^{M^{ij} \tau} = [k^{M^{ij} \tau}(\mathbf{x}, \mathbf{x}_1), \dots, k^{M^{ij} \tau}(\mathbf{x}, \mathbf{x}_N)]$.

At this point, some observations are necessary. Firstly, note that in the latter approach, the symmetry of the inertia matrix is imposed by design. Specifically, this stems from the operator $\mathcal{G}_{M^{ij}}$ being invariant to the swap of the indices i and j . This is not true for the former approach, which does not generally guarantee symmetry of the inertia matrix, for example for the commonly used single-output GP black box models. Secondly, Estimating M using (??) or (19) gives the same results when considering Lagrangian kernels. Finally, it is important to notice that, when $\hat{\tau}$ is estimated using GPR, both in (??) and (19) the elements of \hat{M} depends linearly on the posterior coefficients $\boldsymbol{\alpha}$.

B. GPR as an Empirical Risk Minimization Problem

In Section II, we pointed out that the posterior mean of a GP is the Maximum a Posteriori estimator of the inverse dynamics. Interestingly, the same solution can be obtained by interpreting GPR as the probabilistic counterpart of regression in the Reproducing Kernel Hilbert Space (RKHS) induced by the kernel k . We refer the interested reader to [20] for further details on RKHS, and to [21], [22] for the connections between regression in RKHS and GPR.

When performing regression in the RKHS framework, the estimate of f at any input location \mathbf{x} is of the form $\hat{f}(\mathbf{x}) = K_{\mathbf{x}X} \boldsymbol{\alpha}$ for $\boldsymbol{\alpha} \in \mathbb{R}^{nN}$ and $K_{\mathbf{x}X}$ defined as in 5. The regularized empirical risk is defined as $J(\boldsymbol{\alpha}) = J_{fit}(\boldsymbol{\alpha}) + \lambda J_{reg}(\boldsymbol{\alpha})$, where the fitting term J_{fit} measures how well the estimated function approximate the true one on the training points. The regularization term $J_{reg}(\boldsymbol{\alpha})$, instead, is

$$J_{reg} = \boldsymbol{\alpha}^T K_{XX} \boldsymbol{\alpha},$$

which corresponds to the norm of \hat{f} in the RKHS. This term is meant to penalize functions of high complexity, where the complexity is measured in terms of the norm in the RKHS.

It is trivial to show that, if $\lambda = \sigma_e^2$ and the standard squared error empirical risk is assumed, namely

$$J_{fit} = (\mathbf{y} - K_{XX} \boldsymbol{\alpha})^T (\mathbf{y} - K_{XX} \boldsymbol{\alpha}),$$

then the value of $\boldsymbol{\alpha}$ that minimizes the regularized empirical risk, that is

$$\boldsymbol{\alpha}^* = \arg \min_{\boldsymbol{\alpha} \in \mathbb{R}^{nN}} J(\boldsymbol{\alpha}), \quad (20)$$

corresponds to $\boldsymbol{\alpha}^* = (K_{XX} + \Sigma_e)^{-1} \mathbf{y}$, which leads exactly to the posterior mean in (4a).

The unconstrained solution $\boldsymbol{\alpha}^*$ is optimal in the sense of (20) but, it can lead to physically inconsistent solutions, estimating for example non-positive-definite inertia matrices. To promote the positive definiteness of the inertia matrix, we propose to modify the problem in (20) by adding positivity constraints. Let $\hat{M}(\mathbf{q}_r)$ denote the estimate of the inertia matrix at the constraint location \mathbf{q}_r , with $r = 1, \dots, N_c$. To make the dependence on $\boldsymbol{\alpha}$ explicit, in the following, we will denote $\hat{M}(\mathbf{q}_r)$ as $\hat{M}_r(\boldsymbol{\alpha})$. Then, we redefine (20) as

$$\begin{aligned} \boldsymbol{\alpha}_c^* &= \arg \min_{\boldsymbol{\alpha} \in \mathbb{R}^{nN}} J(\boldsymbol{\alpha}) \\ &\text{subject to } \hat{M}_r(\boldsymbol{\alpha}) \succ 0, \quad r = 1, \dots, N_c \end{aligned} \quad (21)$$

Notice that the constraint optimization problem (21) leaves complete freedom on the selection of the number and location of the constraint points. One could enforce the constraint on the training inputs, namely $N_c = N$ and \mathbf{q}_r corresponds to the training configurations \mathbf{q}_i . However, this is not the only possible choice. As an example, constraint points could be randomly located on regions of the space not covered by the training set, and the number of points can be increased to improve the coverage of the regions of interest. Differently from $\boldsymbol{\alpha}^*$, $\boldsymbol{\alpha}_c^*$ is not available in close form, and we have to resort to numerical optimizers to solve (21).

From the implementation point of view, we express the positivity constraints in terms of the principal minors of \hat{M} , namely we require all its principal minors to be positive. Denoting with $\Delta_r^d(\boldsymbol{\alpha})$ the principal minor of order d of the matrix $\hat{M}_r(\boldsymbol{\alpha})$, the constrained optimization problem becomes

$$\begin{aligned} \boldsymbol{\alpha}_c^* &= \arg \min_{\boldsymbol{\alpha} \in \mathbb{R}^{nN}} J(\boldsymbol{\alpha}) \\ &\text{subject to } \Delta_r^d(\boldsymbol{\alpha}) \succ 0, \quad d = 1, \dots, n \\ &\quad r = 1, \dots, N_c. \end{aligned} \quad (22)$$

As an alternative, one could consider the eigenvalues of the estimated inertia matrix. However, we experimentally verified that employing principal minors leads to superior performance, both in terms of reduced solving time and enhanced robustness of the numerical solutions obtained.

Finally, note that using $\boldsymbol{\alpha}_c^*$ in place of $\boldsymbol{\alpha}^*$ in (7) could lead to a sub-optimal solution, with a possible loss in model accuracy. However, our experimental evaluation suggests that in practice the loss in accuracy is negligible, while the improvements in terms of physical consistency are remarkable.

IV. EXPERIMENTS

This section presents the numerical experiments performed to validate the proposed approach. All the experiments are performed on a Franka Emika Panda robot, simulated using Pybullet. We consider three implementations of the constrained model we propose. All of them are based on the LIP estimator presented in [15] and reviewed in Section II-C, and they differ on the constraint number and location. In particular, the first model enforces the constraints on the training points, while the second and the third enforce the constraints respectively on 200 and 400 input points randomly sampled with uniform distribution in the same range of the training dataset. We compare the aforementioned models with the standard LIP and with a single-output model based on the RBF kernel, which is included as a baseline.

As training data, we collected a random trajectory, obtained by imposing to each joint a reference position defined as

$$q_i(t) = \sum_{l=1}^{N_s} \frac{a}{\omega_f l} \sin(\omega_f l t) - \frac{b}{\omega_f l} \cos(\omega_f l t), \quad (23)$$

with $N_s = 50$, $\omega_f = 0.02$ rad/s, while a and b are sampled from a uniform distribution ranging in $[-c, c]$, with c chosen to respect the limits on joint position, velocity, and acceleration. The training trajectory lasts for 50 seconds, sampled with a frequency of 100 Hz. To reduce the computational burden, we restrict the resulting dataset of 5000 points by selecting the 250 most informative ones. Collected data are used to select the kernels' hyperparameters through marginal likelihood optimization [4], as well as to compute K_{XX} in (22). When implementing the proposed models, we substitute the optimal α^* with α_c^* obtained solving (22), while maintaining the same kernel hyperparameters. We solve the constraint optimization in (22) using *ipopt* [23], exploiting the *jax* library¹ for efficient numeric differentiation.

A. Generalization Experiment

In the first set of experiments, we test the prediction accuracy of the compared models. To this aim, we perform a Monte Carlo experiment, where we tested the estimators on 50 noisy trajectories obtained as in (23). We downsampled the test trajectories with a constant rate, to obtain 100 samples each.

The distribution of the torque prediction error is reported in figure 1 in terms of normalized mean squared error (nMSE) percentage. First, we note that the LIP model is far more accurate than the RBF baseline, confirming the advantages of the multi-output approach in terms of generalization and data efficiency. Interestingly, the performance of the constrained models is comparable to that of the unconstrained one on almost all the joints. Furthermore, note how, as expected, the error slightly increases at the increase of the number of constraint points.

In turn, the positive-definiteness property of the estimated inertia matrix drastically improves. In figure 2, we report the

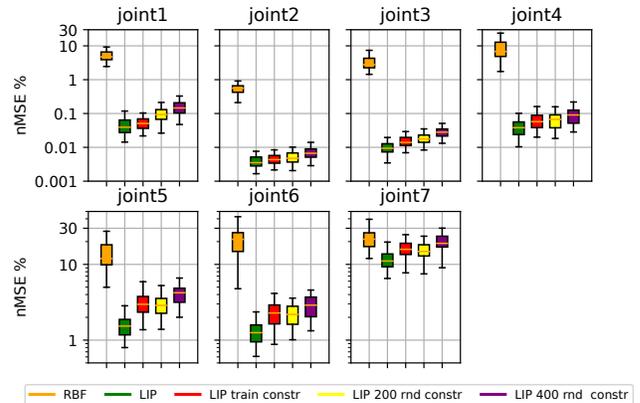


Fig. 1. Box plots of the torque nMSE percentage obtained with the simulations described in Section IV-A.

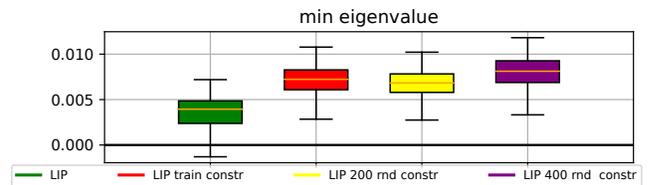


Fig. 2. Box plots of the minimum eigenvalue estimated on the trajectories of the Monte-Carlo experiment described in section IV-A.

distribution of the minimum eigenvalue estimated with the considered models along the 50 trajectories of the Monte Carlo experiment. We do not report the results of the RBF baseline, as it provides non-symmetric and highly inaccurate estimates. Note that, while the distribution of the LIP models comprises a non-negligible amount of non-positive inertias, the constrained models return positive eigenvalues on the test sets. These benefits will be further explored in the next Section.

B. Control Experiment

Assume $(q_d, \dot{q}_d, \ddot{q}_d)$ represents the reference joint trajectory. Moreover, let $e = q_d - q$ represent the tracking error. The feedback linearization control law is defined as

$$\tau = M(q)a + c(q, \dot{q}) + g(q) \quad (24)$$

with $a = \ddot{q}_d + K_P e + K_D \dot{e}$, where K_P and K_D are gain matrices. Assuming perfect knowledge of M , c , and g and provided that K_P and K_D are positive definite, the feedback linearization control law guarantees to asymptotically reach zero error tracking. In practice, if the uncertainties on M , c , and g are limited, tracking errors can be compensated for by increasing the gains K_P and K_D , assuming the positivity of M . If M is not positive definite, the control law may become unstable, and increasing K_P and K_D will further exacerbate the instability.

We exploit the considered models to derive the components required to implement the feedback linearization control. Due to its poor accuracy, we do not consider the

¹<http://github.com/jax-ml/jax>

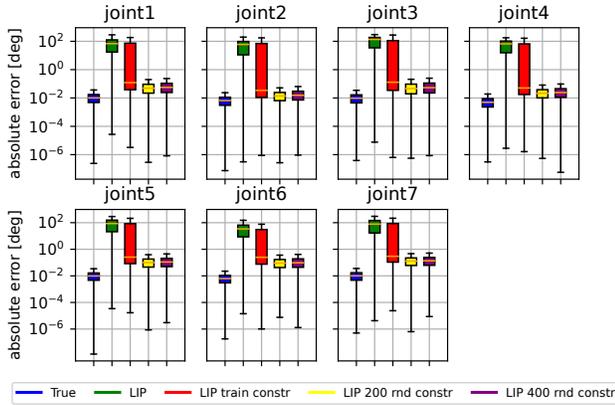


Fig. 3. Box plots of the absolute tracking error obtained with the simulations described in Section IV-B.

RBF model. As a reference, we include the performance of the true model.

Regarding the reference to track, we consider 10 trajectories, each one generated as in the MC test described in the previous Section. All the control loops run at 100 Hz and reference trajectories last for 50 seconds.

In figure 3, we report the distribution of the absolute value of the tracking error, obtained along the 10 reference trajectories. While the unconstrained LIP models always fail to track all the reference trajectories due to non-positive-definite inertia estimates arising, the constrained models show reliable performance on almost all the considered trajectories. The model with constraints only on the training points fails on two trajectories out of ten. We suggest that this happens when the reference explores positions that are very different from the training ones, even if they are sampled from the same distribution. Distributing the constraints uniformly in the region of interest, instead, guarantees correct tracking on all the ten considered trajectories.

V. CONCLUSIONS

We presented a constrained learning approach to promote positive definiteness of the inertia matrix in black-box inverse dynamics models based on GPR. By introducing constraints into the optimization process, our method produces models that maintain physical consistency, essential for stable control in robotics. Simulated experiments on a 7 DOF manipulator validated the effectiveness of our method both in terms of prediction accuracy and control performance, showing that constrained models outperform traditional black-box approaches in providing reliable estimates for standard control tasks.

REFERENCES

[1] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, <http://www.deeplearningbook.org>.
[2] E. Rueckert, M. Nakatenus, S. Tosatto, and J. Peters, "Learning inverse dynamics models in $\mathcal{O}(n)$ time with lstm networks," in *IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, 2017.

[3] 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.
[4] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer school on machine learning*. Springer, 2003, pp. 63–71.
[5] D. Romeres, M. Zorzi, R. Camoriano, S. Traversaro, and A. Chiuseo, "Derivative-free online learning of inverse dynamics models," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 3, 2020.
[6] D. Nguyen-Tuong, M. Seeger, and J. Peters, "Model learning with local gaussian process regression," *Advanced Robotics*, vol. 23, no. 15, 2009.
[7] S. Rezaei-Shoshtari, D. Meger, and I. Sharf, "Cascaded gaussian processes for data-efficient robot dynamics learning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.
[8] A. Dalla Libera, G. Giacomuzzo, R. Carli, D. Nikovski, and D. Romeres, "Forward dynamics estimation from data-driven inverse dynamics learning," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 519–524, 2023.
[9] A. D. Libera, F. Amadio, D. Nikovski, R. Carli, and D. Romeres, "Control of mechanical systems via feedback linearization based on black-box gaussian process models," *arXiv preprint arXiv:2104.12854*, 2021.
[10] C. Gaz, M. Cognetti, A. Oliva, P. R. 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.
[11] 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 2019*. OpenReview.net, 2019. [Online]. Available: <https://openreview.net/forum?id=BklHpjCqKm>
[12] 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, 2020.
[13] 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.
[14] C.-A. Cheng and H.-P. Huang, "Learn the lagrangian: A vector-valued rkhs approach to identifying lagrangian systems," *IEEE transactions on cybernetics*, vol. 46, no. 12, pp. 3247–3258, 2015.
[15] G. Giacomuzzo, R. Carli, D. Romeres, and A. Dalla Libera, "A black-box physics-informed estimator based on gaussian process regression for robot inverse dynamics identification," *IEEE Transactions on Robotics*, 2024.
[16] 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.
[17] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, "Modelling, planning and control," *Advanced Textbooks in Control and Signal Processing*. Springer, 2009.
[18] C. Williams, S. Klanke, S. Vijayakumar, and K. Chai, "Multi-task gaussian process learning of robot inverse dynamics," in *Advances in Neural Information Processing Systems*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds., vol. 21. Curran Associates, Inc., 2008.
[19] 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.
[20] N. Aronszajn, "Theory of reproducing kernels," *Trans. of the American Mathematical Society*, vol. 68, pp. 337–404, 1950.
[21] G. Pillonetto, F. Dinuzzo, T. Chen, G. De Nicolao, and L. Ljung, "Kernel methods in system identification, machine learning and function estimation: A survey," *Automatica*, vol. 50, no. 3, pp. 657–682, 2014.
[22] A. Carè, R. Carli, A. Dalla Libera, D. Romeres, and G. Pillonetto, "Kernel methods and gaussian processes for system identification and control: A road map on regularized kernel-based learning for control," *IEEE Control Systems Magazine*, vol. 43, no. 5, pp. 69–110, 2023.
[23] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, pp. 25–57, 2006.