Data-Driven Model Predictive Control for Trajectory Tracking With a Robotic Arm

Andrea Carron, Elena Arcari, Martin Wermelinger, Lukas Hewing, Marco Hutter, and Melanie N. Zeilinger

Abstract—**High-precision trajectory tracking is fundamental in robotic manipulation. While industrial robots address this through stiffness and high-performance hardware, compliant and cost-effective robots require advanced control to achieve accurate position tracking. In this letter, we present a model-based control approach, which makes use of data gathered during operation to improve the model of the robotic arm and thereby the tracking performance. The proposed scheme is based on an inverse dynamics feedback linearization and a data-driven error model, which are integrated into a model predictive control formulation. In particular, we show how offset-free tracking can be achieved by augmenting a nominal model with both a Gaussian process, which makes use of offline data, and an additive disturbance model suitable for efficient online estimation of the residual disturbance via an extended Kalman filter. The performance of the proposed offset-free GPMPC scheme is demonstrated on a compliant 6 degrees of freedom robotic arm, showing significant performance improvements compared to other robot control algorithms.**

Index Terms—**Learning and adaptive systems, predictive control, model learning for control, model predictive control, robotics.**

Fig. 1. Anypulator, the 6DoF compliant robot used for the experiments and for validating the controller.

I. Introduction

HIGH-PRECISION trajectory tracking with non-industrial robotic arms is a major challenge, since, in contrast to industrial robots, they are non-stiff, have limited actuation power, and the components can exhibit large variations with respect to their specifications. This limits the use of compliant or inexpensive robots for tasks that require high accuracy, or in environments where conditions can quickly change, e.g. construction sites [1] where robots are exposed to severe weather and working conditions. A simple way to deal with these scenarios is to operate robots at low speed, where the dynamics coupling among joints is low, at the expense of limiting overall performance and productivity. The goal addressed in this letter is to achieve both high precision and speed, in order to leverage the full potential of compliant and cost-effective robots.

The combination of data-driven models and Model Predictive Control (MPC) has shown great potential for addressing this challenge and has gained significant popularity in recent years [2]–[5]. On one hand, MPC enables optimal operation while providing constraint satisfaction, on the other hand, data-driven algorithms can enhance performance and adapt to system changes.

In this work, we propose a controller that combines the use of Gaussian Processes (GP) to improve the model of a robotic arm with an additive disturbance model for offset-free control. The GP is used as an *offline* estimator of the expected model mismatch, based on data collected during previous experiments. In order to also provide offset-free tracking in regions of the state space where little or no previous data is available, we make use of an extended Kalman filter to *online* estimate the residual deviation between measurements and the nominal model combined with the Gaussian Process.

The main contribution of the letter is therefore to extend previously presented GP-based MPC methods [6]–[9] to an offset-free tracking approach and the adaptation of this scheme for trajectory tracking with a robotic manipulator, as well as its demonstration in hardware experiments. Previous results have focused on online updates of the GP in order to achieve high controller performance, which is computationally infeasible in

Manuscript received February 24, 2019; accepted June 21, 2019. Date of publication July 22, 2019; date of current version August 1, 2019. This letter was recommended for publication by Associate Editor K. Yamane and Editor P. Rocco upon evaluation of the reviewers' comments. This work was supported by the Swiss National Science Foundation under Grant PP00P2 157601/1. The work of A. Carron and M. Wermelinger were supported by the Swiss National Centre of Competence in Research NCCR Digital Fabrication. *(Corresponding author: Andrea Carron.)*

A. Carron, E. Arcari, L. Hewing, and M. N. Zeilinger are with the Institute for Dynamical Systems and Control, ETH Zürich, 8092 Zürich, Switzerland (e-mail: carrona@ethz.ch; earcari@ethz.ch; lhewing@ethz.ch; mzeilinger@ethz.ch).

M. Wermelinger and M. Hutter are with the Robotic System Laboratory, ETH Zürich, 8092 Zürich, Switzerland (e-mail: martiwer@ethz.ch; mahutter@ethz.ch).

This letter has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors. The video provides a high level overview of the presented approach, as well as a demonstration of the experimental results. The size of the video is 13.2 MB.

Digital Object Identifier 10.1109/LRA.2019.2929987

the considered application. A distinct difference of the proposed method with respect to available learning-based robot control techniques is that we do not fit the GP to the inverse dynamics, but rather to the error between a double integrator model and the robot controlled via feedback linearization. This structure permits us to exploit the given approximate model knowledge, i.e. a parametric inverse dynamics model, and the GP uncertainty.

We support the proposed algorithm with an experimental validation on a compliant 6DoF robotic arm, shown in Figure 1, which makes use of low-power series elastic actuators [10]. The results demonstrate that the controller improves the tracking performance in terms of root mean square tracking error compared to a PID controller, the offset-free MPC scheme presented in [11], the nonlinear MPC scheme [12], and the three learning-based schemes presented in [13], [14]. In particular, the proposed controller reduces the root mean square tracking error by up to 53% compared with the best-practice PID controller.

The remainder of the letter is structured as follows: In Section II and III, we review the literature, and provide a short introduction to Gaussian Processes, respectively. The problem formulation is introduced in Section IV. In Section V we introduce the disturbance model and discuss state and offset estimation, while in Section VI we illustrate the control method. Experimental results are shown in Section VII. Finally, Section VIII concludes the letter.

II. Related Work

Model-based control of robotic arms has been well studied in the literature, including control in joint-space [15]–[18], and task-space [19]–[21].

In recent years, learning techniques and learning-based controllers for robotic manipulation have gained increasing interest, due to their potential for improving control performance compared to traditional control approaches and potentially permitting the to use of inexpensive hardware, or compliant robots, in tasks where high-precision trajectory tracking is required.

One of the first non-parametric estimation approaches applied to robotic arms has been presented in [22], where non-linear functions are approximated by many local linear models. More recently, deep learning techniques have been used to estimate inverse dynamics, see for example [23], [24]. Two recent results that exploit non-parametric Gaussian Processes in the field of robot control are outlined in [25] and [26]. In particular, [25] presents a new approach called Manifold Gaussian Process with the goal of alleviating the smoothness assumptions on the function to be modeled. The results in [26] show how to combine direct and indirect learning to speed up the learning of the inverse dynamics. Examples of non-parametric GP approaches used within an MPC optimization scheme are found in [6]–[9], [27], [28]. The work in [6], [7] presents an MPC approach that integrates a nominal system with an additive nonlinear part of the dynamics modeled as a GP. Approximation techniques for propagating the state distribution and formulating chance constraints are also discussed. A similar model is employed in [8], [9], where the authors study a learning-based path-tracking controller that exploits a GP to improve the kinematic model. In [27], a reinforcement learning approach is shown for collecting data to update the GP model of a ship. The approach in [28] exploits Pontryagin's maximum principle to deal with state and input constraints by reformulating the optimal control problem with uncertainty propagation as a deterministic problem.

When prior model knowledge is available, it can often be incorporated in data-driven approaches. In the case of inverse dynamics learning with Gaussian Processes, this can be reflected by including the parametric inverse dynamics model in the mean prior, see [14]. Over the years, many extensions have been proposed, e.g. [13] presents an inverse dynamics controller based on semi-parametric Gaussian Processes augmented with adaptive feedback. Online inverse dynamics learning techniques are also studied in [29] and [30], with the aim of rendering computations on large-scale data sets tractable.

Reinforcement learning and iterative learning control are two other important classes of learning-based control. In model-free approaches, the goal is to directly learn a control policy by repeating the same task multiple times to collect information about the unmodeled uncertainty and update the controller to achieve better performance, see e.g. [31]–[34]. A reinforcement learning approach exploiting prior model information can be found in [35], that uses GPs to model the residual dynamics and finds an optimal policy using standard policy search methods.

To the best of our knowledge, no combination of an expressive data-based model that is trained offline with a simple additive disturbance model estimated online via an EKF for offset-free tracking has been proposed in the literature.

III. Gaussian Processes

We briefly state the basic concepts of non-parametric Gaussian Process regression. Let $f : \mathscr{X} \to \mathscr{R}$ be a zero-mean Gaussian field with kernel $K : \mathscr{X} \times \mathscr{X} \mapsto \mathscr{R}$, where $\mathscr{X}$ is a compact set. Consider a set of $N \in \mathbb{N}_{>0}$ noisy measurements of the form

$$y_i = f(x_i) + \nu_i,$$

where ν_i is zero-mean Gaussian noise with variance σ^2, i.e. $\nu_i \sim \mathscr{N}(0, \sigma^2)$. The predictive mean estimate μ of f and its posterior covariance Σ for any input $x \in \mathscr{X}$ are given by

$$\mu(x) = \mathbb{E}\left[f(x)|\{x_i, y_i\}_{i=1}^N\right]$$
$$\Sigma(x) = \mathrm{Var}\left[f(x)|\{x_i, y_i\}_{i=1}^N\right],$$

where $\{x_i, y_i\}_{i=1}^N$ is the collected dataset.

IV. Problem Formulation

In the following, we derive a model predictive control formulation for trajectory tracking with a robotic arm. The proposed scheme is composed of three parts: a feedback linearization inner-loop that exploits a priori knowledge of the robot dynamics, a model for the closed-loop system with the feedback linearization, which accounts for two types of uncertainty: a state and input dependent uncertainty estimated offline using a GP, and a constant disturbance estimated online using an extended

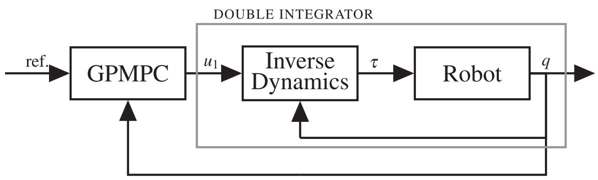

Fig. 2. Control scheme leveraging inverse dynamics model.

Kalman filter in order to perform offset-free tracking also in regions of the state space where little or no data is available, an offset-free nonlinear MPC control scheme for the outer-loop leveraging the data-driven model and uncertainty.

In order to derive this control scheme, we first introduce the underlying model of the robot.

The rigid body dynamics of the robot are modeled as a second-order differential equation

$$M(q)\ddot{q} + n(q, \dot{q}) = \tau, \tag{1}$$

where q, $\dot{q}$, $\ddot{q}$ represent the joint angles, joint angular velocities, and joint angular accelerations, respectively, τ the torques applied by the actuators, $M(q)$ is the inertia matrix, and $n(q, \dot{q})$ includes gravitational, Coriolis, frictional and centripetal forces. Since (1) is linear in τ, it is possible to perform a feedback linearization via the following control law if the matrix $M(q)$ is full rank for all possible q [15]

$$\tau = M(q)u + n(q, \dot{q}), \tag{2}$$

where u is the applied input signal. If the inverse dynamics is exact, then the feedback linearized system behaves like D decoupled double integrators, where D is the number of degrees of freedom of the robotic arm.

Since the inverse dynamics is usually not perfectly known, it is common to add an outer-loop controller, which in our case is designed as a nonlinear MPC, see Figure 2.

The discrete-time model employed in the MPC scheme has the following form

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + B_d(w(x(k), u(k))) \\ y(k) &= Cx(k) + \nu(k), \end{aligned} \tag{3}$$

where $x(k) = [q(k),\ \dot{q}(k)]^T$, the matrices A and B model the discrete-time double integrators, B_d and C are the process-noise to state matrix and the output matrix, $w(x(k), u(k))$ is a nonlinear function modeling the uncertainty, and $\nu(k)$ represents the measurement noise and is i.i.d., Gaussian with zero mean and covariance $\Sigma_\nu = \sigma^2\mathbb{I}$. In addition, system (1) is subject to state and input constraints of the form

$$\begin{aligned} x(k) &\in \mathbb{X} = \{x | A_x x \leq b_x\} \\ u(k) &\in \mathbb{U} = \{u | A_u u \leq b_u\}, \end{aligned}$$

where $\mathbb{X}$ and $\mathbb{U}$ are polytopic sets representing the position, velocity and acceleration limits of the robotic arm.

Remark 1: In robotic arms, velocity and acceleration constraints are sometimes non-linear and state-dependent. The proposed control framework can be directly extended to address non-linear constraints [36]. For the sake of clarity, we will only consider polytopic constraints in this letter, which can be computed considering the worst-case velocity and acceleration scenarios for the given trajectory.

In the next sections we will first introduce the disturbance model composed of a Gaussian Process and a constant disturbance, together with a state-disturbance estimator. We will then introduce the non-linear MPC controller that exploits the improved model and the estimated disturbance to optimize the control input. Finally, we will discuss some implementation details.

V. Disturbance Model

We assume that the uncertainty in (3) consists of two components, one is modeled as a GP and the other as a constant offset, i.e.,

$$w(x(k), u(k)) = d(x(k), u(k)) + \bar{d}. \tag{4}$$

Differently to the other approaches which fit the GP to the inverse dynamics, see [14], [29], [30], we use the GP to model the error with respect to the double integrator model. In this way, it is possible to directly exploit the GP uncertainty estimate in the state, without the need to perform a transformation of the probability distributions through the forward dynamics. The GP provides the residual model uncertainty in domains where little data is available, which can be used to ensure constraint satisfaction, however, it can still result in poor tracking performance and offsets. While this could be addressed in a GP approach by updating the data for the GP online, it is computationally prohibitive to train and evaluate the GP model in the considered dimensions at frequency rates of around 1 kHz. To mitigate this limitation, we take an offset-free tracking MPC approach and introduce a constant disturbance term $\bar{d}$, which can be efficiently estimated online using an extended Kalman filter [11]. The GP can be thought of as a feedforward compensation, while the constant $\bar{d}$ estimated through the extended Kalman filter as a feedback compensation. The reaction speed of the constant term depends on how quickly the extended Kalman filter converges to a steady-state estimate compared with the changes in the reference trajectory. The robot model (3) is augmented with the disturbance model as follows

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + B_d\left(d(x(k), u(k)) + \bar{d}(k)\right) \\ \bar{d}(k+1) &= \bar{d}(k) \\ y(k) &= Cx(k) + \nu(k), \end{aligned} \tag{5}$$

where we assume the augmented model to be observable, such that both the state and offset $\bar{d}$ can be estimated using an extended Kalman filter. The details of the control scheme are presented in the following section.

VI. Control Scheme

We propose a non-linear MPC controller for system (5) that exploits the disturbance model to provide offset-free tracking and ensures satisfaction of state and input constraints. Since we consider noise with infinite support, we specify state constraints

Algorithm 1: Offset-free GPMPC.

1: **Input:** Extended Kalman Filter matrices, GP training data
2: Optimize GP hyperparameters
3: **for** $k = 1, 2, \dots$ **do**
4: Extended Kalman prediction:
$\hat{s}_{k+1|k} = \bar{A}\hat{s}_{k|k} + \bar{B}u(k) + \bar{B}_d d(x(k), u(k))$
$\Sigma_{k+1|k} = \bar{A}\Sigma_{k|k}\bar{A}^T + Q$
5: Solve approximation of problem (7)
6: Extended Kalman update:
$\hat{s}_{k+1|k+1} = \hat{s}_{k+1|k} + L_{k+1}(y_{k+1} - \bar{C}\hat{s}_{k+1|k})$
$\Sigma_{k+1|k+1} = \Sigma_{k+1|k} - L_{k+1}\bar{C}\Sigma_{k+1|k}$
$L_{k+1} = \Sigma_{k+1|k}\bar{C}^T(\bar{C}\Sigma_{k+1|k})\bar{C}^T + R^T)^{-1}$
7: **end for**

in a probabilistic manner in the form of chance constraints

$$\Pr\left[x(k) \in \mathbb{X}\right] \geq 1 - \epsilon \tag{6}$$

where ϵ is the violation probability. The resulting optimization problem is given by

$$\min\ l_f(x_N, r_{k+N}) + \sum_{i=0}^{N_{MPC}-1} l(x_i, u_i, r_{k+i}) \tag{7a}$$

$$\text{s.t. } x_0 = \hat{x}(k) \tag{7b}$$

$$\bar{d}_0 = \hat{\bar{d}}(k) \tag{7c}$$

$$x_{i+1} = Ax_i + Bu_i + B_d\left(d(x_i, u_i) + \bar{d}_i\right) \tag{7d}$$

$$\bar{d}_{i+1} = \bar{d}_i \tag{7e}$$

$$\Pr\left[x_{i+1} \in \mathbb{X}\right] \geq 1 - \epsilon, \tag{7f}$$

$$u_i \in \mathbb{U}, \tag{7g}$$

$$\forall i = 0, \dots, N_{MPC} - 1$$

where $l(x_i, u_i, r_{k+i})$ and $l_f(x_i, r_{k+N_{MPC}})$ are appropriate tracking stage and terminal cost functions, respectively, that depend on the state, input, and the reference trajectory $r_k = [r_k^x, r_k^u]$. The initial condition of the optimization problem is obtained from the extended Kalman filter, i.e., $\hat{x}(k)$ and $\hat{\bar{d}}(k)$. In the following we show how to derive a tractable approximation of problem (7). In particular, we show how to propagate the model and uncertainty over the prediction horizon, discuss suitable cost functions, formulate state and input constraints, and finally discuss how to reduce the computational burden of the Gaussian Process estimation. The overall control algorithm is summarized in Algorithm 1, where $s(k) = \left[x(k)^T, \bar{d}(k)^T\right]^T$ is the extended state, and $\bar{A}$, $\bar{B}$, $\bar{B}_d$, and $\bar{C}$ are the extended system matrices. Steps 4 and 6 of Algorithm 1 describe the update of the extended Kalman filter. Step 4 is the open-loop prediction based on the nonlinear model given by the combination of the linear system and the GP. Step 6 is the update of the state and covariance based on the measurements collected.

Remark 2: Once new data is collected, the GP hyperparameters can be retrained on an enlarged dataset to further improve the model and hence the tracking performance. It is, however, important to note that the online disturbance estimation via the Kalman filter and the offset-free MPC formulation improve tracking performance independent of the GP and thereby for trajectories for which no prior data is available.

A. Approximate Uncertainty Propagation

Due to the representation of the nonlinearity by a GP and the presence of noise, the predicted states are given as stochastic distributions. The evaluation of the posterior distribution of the GP from an input distribution, however, requires the computation of an integral that is analytically intractable. A common approach is to approximate state, input and the GP describing the nonlinearity as jointly Gaussian [37]–[39], i.e.,

$$\begin{bmatrix} x_i \\ d_i \end{bmatrix} \sim \mathcal{N}(\mu_i, \Sigma_i) = \mathcal{N}\left(\begin{bmatrix} \mu_i^x \\ \mu_i^d \end{bmatrix}, \begin{bmatrix} \Sigma_i^x & \Sigma_i^{xd} \\ \Sigma_i^{dx} & \Sigma_i^d + \Sigma_\nu \end{bmatrix}\right).$$

The update equation is then given by a simple linear transformation

$$\mu_{i+1}^x = \begin{bmatrix} A & B_d \end{bmatrix} \mu_i + Bu_i$$
$$\Sigma_{i+1}^x = \begin{bmatrix} A & B_d \end{bmatrix} \Sigma_i \begin{bmatrix} A & B_d \end{bmatrix}^T.$$

We adopt a first-order Taylor approximation [37] of the estimate to evaluate the GP at a Gaussian input and approximate the variance and covariance matrices, i.e.

$$\mu_i^d = \mu_i^d(\mu_i^x, u_i)$$
$$\Sigma_i^d = \Sigma_i^d(\mu_i^x, u_i) + \nabla\mu_i^d(\mu_i^x, u_i)\Sigma_i^x(\nabla\mu_i^d(\mu_i^x, u_i))^T$$
$$\Sigma_i^{dx} = \Sigma_i^x(\nabla\mu_i^d(\mu_i^x, u_i))^T.$$

B. Cost Function

We propose two possible cost functions that can be used in optimization problem (7) in the following, however, any other cost that makes use of the first and second moments of state could be similarly employed. One option is to take a certainty equivalence approach, and use a quadratic cost function evaluated at the GP mean prediction, namely,

$$\begin{aligned} l(x_i, u_i, r_{k+i}) &= \|\mu_i^x - r_{k+i}^x\|_Q^2 + \|u_i - r_{k+i}^u\|_R^2 \\ l_f(x_{N_{MPC}}, r_{k+N_{MPC}}) &= \|\mu_{N_{MPC}}^x - r_{k+N_{MPC}}^x\|_P^2, \end{aligned} \tag{8}$$

where Q, R, and P are the cost weights. A second possibility is the expected value of a quadratic cost, which results in

$$\begin{aligned} l(x_i, u_i, r_{k+i}) &= \mathbb{E}\left[\|x_i - r_{k+i}^x\|_Q^2\right] + \|u_i - r_{k+i}^u\|_R^2 \\ &= \|\mu_i^x - r_{k+i}^x\|_Q^2 + \text{tr}(Q\Sigma_i^x) \\ &\quad + \|u_i - r_{k+i}^u\|_R^2 \\ l_f(x_{N_{MPC}}, r_{k+N_{MPC}}) &= \mathbb{E}\left[\|x_{N_{MPC}} - r_{k+N_{MPC}}^x\|_Q^2\right] \\ &= \|\mu_N^x - r_{k+N_{MPC}}^x\|_P^2 + \text{tr}(P\Sigma_{N_{MPC}}^x). \end{aligned}$$

Note that both options result in convex cost functions for positive semidefinite Q, R and P.

C. Chance Constraints

Building on the approximation of the state distribution discussed in subsection VI-A, state mean and covariance can be also exploited to approximate the chance constraints. By approximating the predicted distribution with a Gaussian multivariate distribution and considering polytopic state constraints, chance constraint (6) is given by

$$\Pr\left[\eta(A_x\mu_i^x, A_x\Sigma_i^X A_x^T) \le b_x\right] \ge 1-\epsilon, \tag{9}$$

where $\eta(\mu, \Sigma)$ denotes a normal random vector with mean μ and covariance Σ. Equation (9) can be further approximated using the union bound,

$$\Pr\left[\eta(A_x\mu_i^x, A_x\Sigma_i^X A_x^T) \le b_x\right] \ge$$
$$\sum_{j=1}^{n_c}\Pr\left[\eta([A_x]_j\mu_i^x, [A_x]_j\Sigma_i^X [A_x]_j^T) \le [b_x]_j\right] \ge \sum_{j=1}^{n_c} 1-\epsilon_j,$$

where $\epsilon_j = \frac{\epsilon}{n_c}$. The individual constraints can then be imposed by

$$[A_x]_j\mu_i^x + \Phi^{-1}(1-\epsilon_j)\sqrt{[A_x]_j\Sigma_i^X[A_x]_j^T} \le [b_x]_j,$$

where Φ^{-1} denotes the quantile function of the standard normal distribution.

D. Sparse GP

When using a GP model, the size of the available dataset directly affects the complexity of inference. The computationally most expensive operation is the inversion of the sampled kernel $K_{N,N}$, which is however performed offline and therefore does not affect the computation time of the controller. Other expensive operations are mean and covariance evaluation, which scale as $O(N^2)$, where N is the number of training examples, due to the multiplication between the inverted sampled kernel and the kernel evaluated at an input location of interest. Different techniques have been proposed in the literature to tackle this problem, generally referred to as approximate or sparse GP methods [40], [41], [42]. The approximation is derived by choosing a small set of $M \ll N$ "virtual" data points, typically optimizing over the location of virtual points, reducing the complexity from $O(N^2)$ to $O(NM)$. For the proposed offset-free GPMPC controller, we utilize a technique based on variational learning of inducing inputs and kernel hyperparameters [40], [43], which are selected by minimizing the Kullback-Leibler divergence between the exact posterior GP and its variational approximation.

VII. Experimental Results

In order to validate the tracking capabilities of the proposed method, we implemented the algorithm on a compliant 6DoF robotic arm [10]. All joints are equipped with the same high-performance Series Elastic Actuator (SEA) units. These actuators, called ANYdrives [44], are composed of a brushless motor, harmonic drive gear, a torsional spring, and integrated control electronics and sensors. They have a maximum torque of 40 Nm and can reach a speed of 12 rad/s. The drive features a torque control bandwidth of 70 Hz for low amplitudes and more than 20 Hz for large amplitudes. The arm has a reach of 0.75 m and an end-effector payload of approximately 3 kg in nominal configuration. The approximation of problem (7) was solved using the interior-point-based NLP solver FORCES PRO [45]. We employ a double integrator model for each link, with states given by joint position and velocity. We assume the model mismatch to act only on the velocity states. We estimated the process and measurement matrices of the extended Kalman filter using the Autocovariance Least-square method presented in [46]. The stage and terminal cost functions of problem (7) were chosen to be quadratic as in (8), with weights $Q = \mathrm{diag}(50, 1)$ and $R = 1$ for every double integrator model corresponding to a link. The terminal cost is chosen as the solution of the Algebraic Riccati Equation with weights Q and R for the double integrator system. The offset-free GPMPC prediction horizon length is chosen to $N_{MPC} = 50$. The proposed approach is run in a receding horizon fashion, i.e., problem (7) is solved at every sampling time and the optimal solution obtained, together with current position and velocity, is used for the inverse dynamics computation.

The training data for the Gaussian Process was collected using the offset-free MPC scheme [11], and in particular we chose position, velocity and the acceleration computed by [11] as features. We then trained the GP with all the data collected, without performing any kind of data pre-selection. It was observed that the training of the Gaussian Process is a critical part of the proposed control scheme and overfitting to the data should be avoided to achieve good control performance. To cope with this issue, we modeled the GP with zero mean and Gaussian kernel and estimated the hyperparameters using leave-one-out cross-validation, followed by a fine manual tuning on the hardware. Finally, the inducing points used in the variational approach [40] were chosen to be a subset of the measurements. Interestingly, even though the GP is trained offline, the sparse GP approximation is required to render the MPC prediction with the GP model computationally feasible. It was in fact not possible to generate the solver with the full GP model, whereas with the sparse GP it is possible to run the solver in less than 1 ms.

We performed two different tracking experiments. The first task was to track a 6th-order polynomial trajectory generated in the joint-space, the second task to track a lemniscate trajectory generated in the task-space and then mapped to the joint-space through the inverse kinematics. To train the GP, using [11] we collected about 1,000 samples for the first experiment and 12,000 for the second, which correspond to about 1 and 12 seconds of data, respectively, given a sampling time of 1 ms. The performance of the proposed offset-free GPMPC algorithm is compared with the offset-free MPC scheme presented in [11], the nonlinear MPC scheme presented in [12], a PID controller, the GP-based inverse dynamics control presented in [14], and the two learning-based algorithms presented in [13], all running at a sampling time of 1 ms. The tuning of the PID controller was performed empirically on the robot, for each joint individually. The goal was to achieve a critically damped behavior in the nominal configuration for a typical step input. The algorithm presented in [14], hereafter called GPRBD, estimates the inverse dynamics with a GP where the prior mean is

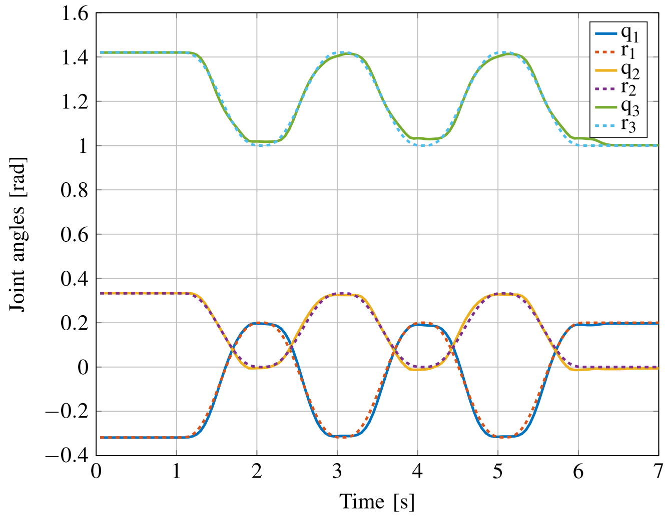

Fig. 3. Polynomial reference trajectory (dashed) and closed-loop trajectories under the offset-free GPMPC controller (solid) for the first three links.

a parametric model of the inverse dynamics. The loop is closed using an LQR controller. The first method presented in [13], hereafter called OL, is a gradient-based learning algorithm that estimates constant offsets of the inverse dynamics model and uses them to correct the control action of an LQR controller. The second variant of this method presented in [13], called OL+GP, fits a GP to the model error and is used together with the gradient-based learning algorithm to correct the LQR control action. While with the PID, MPC, offset-free GPMPC and the three learning-based schemes we run all experiments on six links, this was not possible with the NMPC approach, being computationally too demanding for the given hardware. For this reason we performed all comparisons only on the first three links.

The trajectory of the first task for the first three links is shown in Figure 3. As the performance index we use the Root Mean Square Error (RMSE) between the desired and tracked positions of the three joints over the entire trajectory, i.e.

$$\mathrm{RMSE} = \sqrt{\frac{1}{T}\sum\nolimits_{k=1}^{T} \|q_k - r_k\|_2^2},$$

where T is the length of the trajectory, together with the 25th and 75th percentile of the 2-norm of the error, i.e., $\|q_k - r_k\|_2$. The results of this experiment are shown in Figure 4. Offset-free GPMPC outperforms all other techniques with respect to tracking RMSE, with an improvement of about a factor of 3 with respect to the offset-free MPC [11], highlighting the performance gain through the improved model accuracy provided by the GP, and an improvement of about a factor of 2 compared with the best-practice PID controller and the other learning-based controllers. Figure 5 shows the offset estimated by the extended Kalman filter for the second joint. According to the model and considering only the second link, the states of which are joint position and velocity, the disturbance only affects the joint velocity, and is therefore scalar. We noticed that for the offset-free GPMPC this offset is smaller than for the offset-free MPC [11], because the prediction model used in the extended

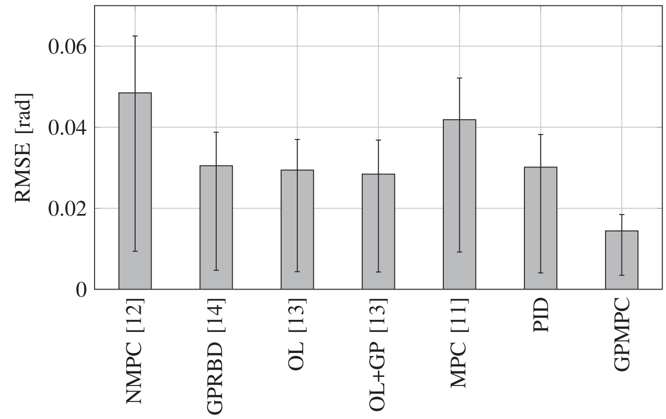

Fig. 4. RMSE over the entire trajectory, 25th and 75th percentiles of the 2-norm of the error at each time step for the three MPC schemes, three learning-based controllers and the PID controller on a 6th order polynomial trajectory in joint space, for the first three joints.

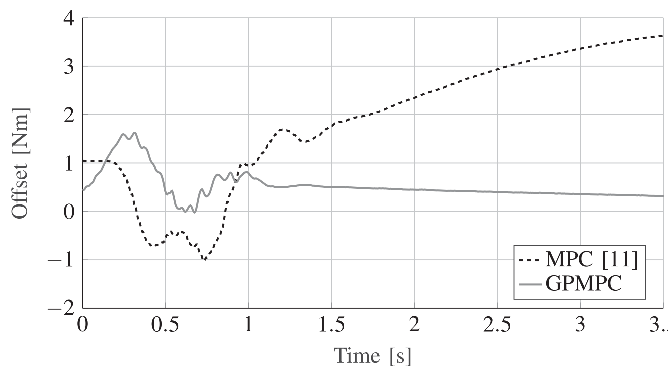

Fig. 5. Estimated offset for the second link obtained from the extended Kalman filter for the MPC scheme [11] and the GPMPC.

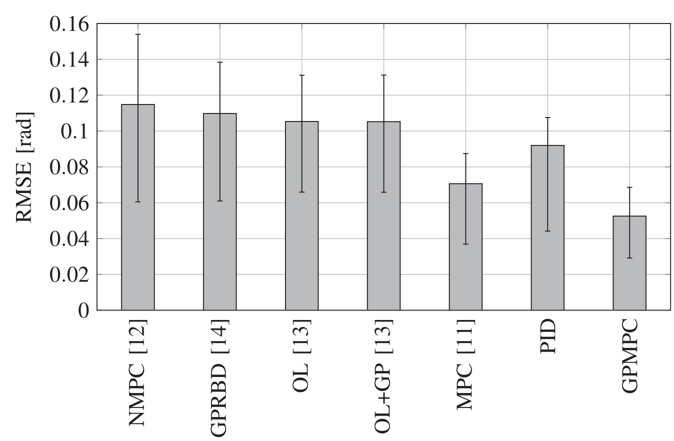

Fig. 6. RMSE over the entire trajectory, 25th and 75th percentiles of the 2-norm of the error at each time step for the three MPC schemes, the three learning-based controllers and the PID controller on a lemniscate trajectory, for the first three joints.

Kalman filter is more accurate due to the use of the Gaussian Process in the prediction step.

In the second experiment, the robot tracked a lemniscate trajectory in task-space. The associated joint angles are generated through the inverse kinematics and fed as reference to

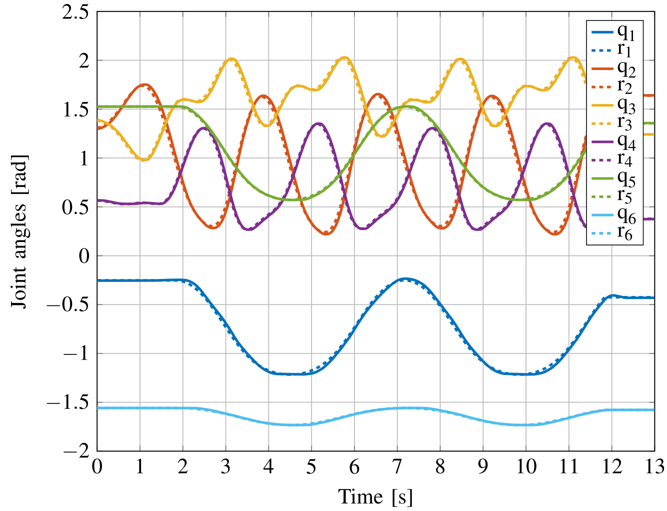

Fig. 7. Lemniscate reference trajectory (dashed) and closed-loop trajectories under the offset-free GPMPC controller (solid).

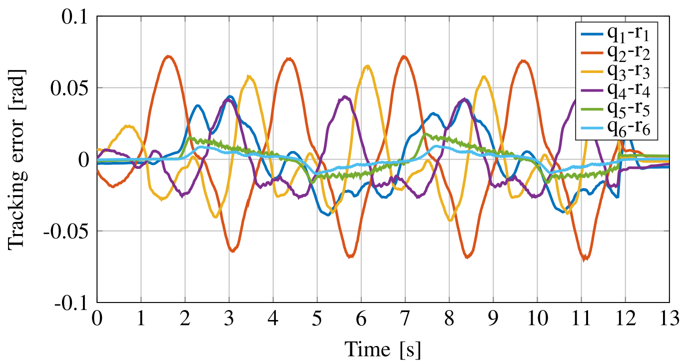

Fig. 8. Error between the reference trajectories and closed-loop trajectories under the offset-free GPMPC controller.

the control scheme. Figure 6 shows the tracking RMSE over the whole trajectory in joint-space together with the 25th and 75th percentile of the 2-norm of the error. Figure 7 and 8 show the reference trajectories for all links together with the closed-loop trajectory obtained under the offset-free GPMPC control scheme, and the associated tracking errors, respectively. Also in this experiment offset-free GPMPC outperforms all other techniques with respect to tracking RMSE showing a 28% and 44% performance improvement of the offset-free MPC and the best-practice PID, respectively. Figure 9 shows the offset estimate by the extended Kalman filter and the GP for the second link. We noticed that most of the correction was performed by the GP, while a finer correction was performed by the disturbance estimated via the extended Kalman filter.

We performed a third experiment showing the disturbance rejection capabilities of the offset-free GPMPC, and the result is shown in Figure 10. We applied an external disturbance to the end-effector while the controller was tracking a constant reference. The figure shows how the second link of the robot quickly recovers the tracked reference after the application of the disturbance. A video of the experiments performed can be found at the following link https://youtu.be/D46Eh59K_eU.

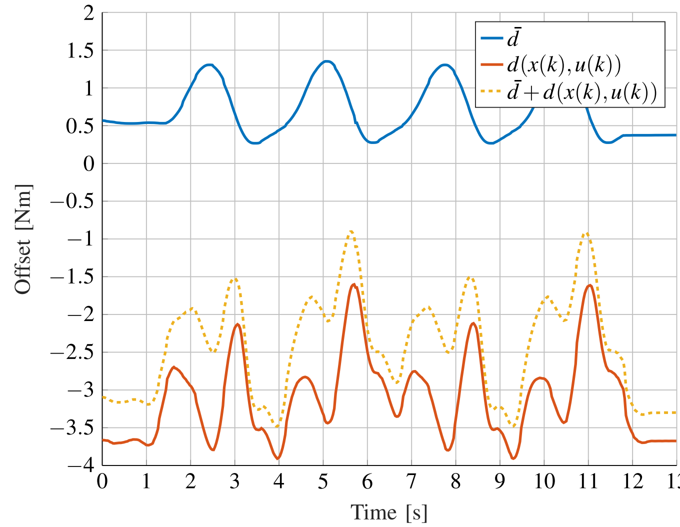

Fig. 9. Estimated offset for the second link computed by the extended Kalman filter $\bar{d}$, GP offset estimate $d(x(k), u(k))$, and sum of the two.

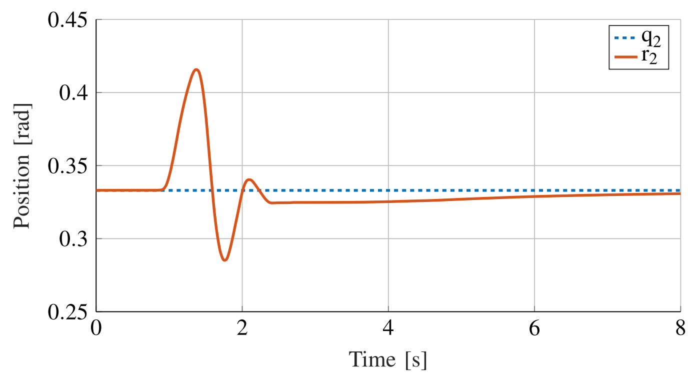

Fig. 10. Constant trajectory to track (dotted) and tracked trajectory (solid) under the offset-free GPMPC for the second link. After 1 second a disturbance is applied to the end-effector, and the controller quickly recovers the original position.

VIII. CONCLUSIONS

This letter presents a novel approach for controlling a robotic arm based on a data-driven model predictive controller. The control scheme exploits a Gaussian Process trained offline to estimate the mismatch between the actual and estimated model, and an extended Kalman filter to estimate the residual model mismatch online, and provides offset-free tracking. The proposed algorithm was shown to improve tracking capabilities on a 6DoF robotic arm compared with two other MPC schemes and a PID controller.

REFERENCES

[1] M. Giftthaler *et al.*, "Mobile robotic fabrication at 1:1 scale: The in situ fabricator," *Construction Robot.*, vol. 1, no. 1, pp. 3–14, Dec. 2017.

[2] A. Aswani, H. Gonzalez, S. S. Sastry, and C. Tomlin, "Provably safe and robust learning-based model predictive control," *Automatica*, vol. 49, no. 5, pp. 1216–1226, 2013.

[3] D. Limon, J. Calliess, and J. Maciejowski, "Learning-based nonlinear model predictive control," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 7769–7776, 2017.

[4] K. P. Wabersich and M. N. Zeilinger, "Linear model predictive safety certification for learning-based control," in *Proc. IEEE Conf. Decis. Control*, Miami Beach, FL, 2018, pp. pp. 7130–7135.

[5] T. Koller, F. Berkenkamp, M. Turchetta, and A. Krause, "Learning-based model predictive control for safe exploration and reinforcement learning," 2018, *arXiv:1803.08287*.

[6] L. Hewing and M. N. Zeilinger, "Cautious model predictive control using Gaussian process regression," *CoRR*, 2017, *arXiv:1705.10702*.

[7] L. Hewing, A. Liniger, and M. N. Zeilinger, "Cautious NMPC with Gaussian process dynamics for miniature race cars," *CoRR*, 2017, *arXiv:1711.06586*.

[8] C. J. Ostafew, A. P. Schoellig, and T. D. Barfoot, "Learning-based nonlinear model predictive control to improve vision-based mobile robot path-tracking in challenging outdoor environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2014, pp. 4029–4036.

[9] C. J. Ostafew, A. P. Schoellig, and T. D. Barfoot, "Robust constrained learning-based NMPC enabling reliable mobile robot path tracking," *Int. J. Robot. Res.*, vol. 35, no. 13, pp. 1547–1563, 2016. [Online]. Available: https://doi.org/10.1177/0278364916645661

[10] J. Carius, M. Wermelinger, B. Rajasekaran, K. Holtmann, and M. Hutter, "Deployment of an autonomous mobile manipulator at MBZIRC," *J. Field Robot.*, vol. 35, no. 8, pp. 1342–1357, 2018.

[11] F. Borrelli and M. Morari, "Offset free model predictive control," in *Proc. 46th IEEE Conf. Decis. Control*, Dec. 2007, pp. 1245–1250.

[12] M. Giftthaler, "Towards a unified framework of efficient algorithms for numerical optimal robot control," Ph.D. dissertation, ETH Zürich, Zürich, Switzerland, 2018.

[13] F. Meier, D. Kappler, N. Ratliff, and S. Schaal, "Towards robust online inverse dynamics learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2016, pp. 4034–4039.

[14] D. Nguyen-Tuong and J. Peters, "Using model knowledge for learning inverse dynamics," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2010, pp. 2677–2682.

[15] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control* (ser. Advanced Textbooks in Control and Signal Processing). London, U.K.: Springer, 2008.

[16] J.-J. E. Slotine and W. Li, "On the adaptive control of robot manipulators," *Int. J. Robot. Res.*, vol. 6, no. 3, pp. 49–59, 1987.

[17] M. W. Spong, R. Ortega, and R. Kelly, "Comments on" adaptive manipulator control: a case study" by J. Slotine and W. li," *IEEE Trans. Autom. Control*, vol. 35, no. 6, pp. 761–762, Jun. 1990.

[18] J. J. Slotine and S. S. Sastry, "Tracking control of non-linear systems using sliding surfaces, with application to robot manipulators," *Int. J. Control*, vol. 38, no. 2, pp. 465–492, 1983.

[19] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE J. Robot. Autom.*, vol. 3, no. 1, pp. 43–53, Feb. 1987.

[20] F. Caccavale, C. Natale, B. Siciliano, and L. Villani, "Resolved-acceleration control of robot manipulators: A critical review with experiments," *Robotica*, vol. 16, no. 5, pp. 565–573, Sep. 1998.

[21] P. Hsu, J. Hauser, and S. Sastry, "Dynamic control of redundant manipulators," in *Proc. Amer. Control Conf.*, Jun. 1988, pp. 2135–2139.

[22] S. Vijayakumar and S. Schaal, "Locally weighted projection regression: An o(n) algorithm for incremental real time learning in high dimensional space," in *Proc. 17th Int. Conf. Mach. Learn.*, 2000, pp. 1079–1086.

[23] A. S. Polydoros, L. Nalpantidis, and V. Krger, "Real-time deep learning of robotic manipulator inverse dynamics," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 3442–3448.

[24] E. Rueckert, M. Nakatenus, S. Tosatto, and J. Peters, "Learning inverse dynamics models in o(n) time with LSTM networks," in *Proc. Int. Conf. Humanoid Robots*, 2017, pp. 811–816.

[25] R. Calandra, J. Peters, C. Rasmussen, and M. Deisenroth, "Manifold Gaussian processes for regression," in *Proc. Int. Joint Conf. Neural Netw.*, 2016, pp. 3338–3345.

[26] D. Kappler, F. Meier, N. D. Ratliff, and S. Schaal, "A new data source for inverse dynamics learning," *CoRR*, 2017, *arXiv:1710.02513*.

[27] Y. Cui, S. Osaki, and T. Matsubara, "Reinforcement learning ship autopilot: Sample-efficient and model predictive control-based approach," *CoRR*, 2019, *arXiv:1901.07905*.

[28] S. Kamthe and M. Deisenroth, "Data-efficient reinforcement learning with probabilistic model predictive control," in *Proc. 21st Int. Conf. Artif. Intell. Statist.*, 2018, vol. 84, pp. 1701–1710.

[29] R. Camoriano, S. Traversaro, L. Rosasco, G. Metta, and F. Nori, "Incremental semiparametric inverse dynamics learning," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2016, pp. 544–550.

[30] D. Romeres, M. Zorzi, R. Camoriano, and A. Chiuso, "Online semiparametric learning for inverse dynamics modeling," in *Proc. IEEE 55th Conf. Decis. Control*, 2016, pp. 2945–2950.

[31] E. Theodorou, J. Buchli, and S. Schaal, "Reinforcement learning of motor skills in high dimensions: A path integral approach," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 2397–2403.

[32] M. Deisenroth and C. Rasmussen, "Pilco: A model-based and data-efficient approach to policy search," in *Proc. 28th Int. Conf. Mach. Learn.*, Omnipress, 2011, pp. 465–472.

[33] A. Doerr *et al.*, "Optimizing long-term predictions for model-based policy search," in *Proc. Mach. Learn. Res.*, Nov. 2017, vol. 78, pp. 227–238.

[34] H. Ahn, Y. Chen, and K. L. Moore, "Iterative learning control: Brief survey and categorization," *IEEE Trans. Syst., Man, Cybernet., C (Appl. Rev.)*, vol. 37, no. 6, pp. 1099–1121, Nov. 2007.

[35] M. Saveriano, Y. Yin, P. Falco, and D. Lee, "Data-efficient control policy search using residual dynamics learning," *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 4709–4715

[36] M. Wendt, P. Li, and G. Wozny, "Nonlinear chance-constrained process optimization under uncertainty," *Ind. Eng. Chemistry Res.*, vol. 41, no. 15, pp. 3621–3629, 2002.

[37] J. Quionero-Candela, A. Girard, and C. Rasmussen, "Prediction at an uncertain input for Gaussian processes and relevance vector machines application to multiple-step ahead time-series forecasting," Danish Technical University, Tech. Rep. IMM-2003-18, 2002.

[38] M. Kuss, "Gaussian process models for robust regression, classification, and reinforcement learning," Ph.D. dissertation, Biologische Kybernetik, Technische Universität Darmstadt, Darmstadt, Germany, Mar. 2006.

[39] M. Deisenroth, "Efficient reinforcement learning using Gaussian processes," Ph.D. dissertation, Karlsruhe Institute of Technology, Karlsruhe, Germany, 2010.

[40] M. Titsias, "Variational learning of inducing variables in sparse Gaussian processes," in *Proc. 12th Int. Conf. Artif. Intell. Statist.*, Apr. 16–18, 2009, pp. 567–574.

[41] A. Smola and P. Bartlett, "Sparse greedy Gaussian process regression," in *Proc. Adv. Neural Inf. Process. Syst.*, 2001, pp. 619–625.

[42] E. Snelson and Z. Ghahramani, "Sparse Gaussian processes using pseudo-inputs," in *Adv. Neural Inf. Process. Syst.*, 2006, pp. 1257–1264.

[43] M. Titsias, "Variational learning of inducing variables in sparse Gaussian processes," in *Proc. 12th Int. Conf. Artif. Intell. Statist.*, Jan. 2009, pp. 567–574.

[44] M. Hutter *et al.*, "Anymal-a highly mobile and dynamic quadrupedal robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2016, pp. 38–44.

[45] A. Domahidi and J. Jerez, "FORCES Professional," embotech GmbH. Jul. 2014. [Online]. Available: http://embotech.com/FORCES-Pro.

[46] B. J. Odelson, M. R. Rajamani, and J. B. Rawlings, "A new autocovariance least-squares method for estimating noise covariances," *Automatica*, vol. 42, no. 2, pp. 303–308, 2006.