Derivative-Free Semiparametric Bayesian Models for Robot Learning

Romeres, Diego; Jha, Devesh K.; Dalla Libera, Alberto; Chiuso, Alessandro; Nikovski, Daniel N.

TR2018-165 December 07, 2018

Abstract

Model-Based Reinforcement Learning (MBRL) is gaining much interest in the robot learning community; in MBRL, the model serves as a representation which is largely task-invariant, and thus can facilitate transfer of knowledge across multiple tasks in the same domain. Learning reliable models for physical systems, however, remains a challenging problem. This paper summarizes recent semiparametric and derivative-free modelling techniques, and presents some key points for a new methodology to formulate derivative-free semiparametric Bayesian models with applications to robot learning. The modeling technique is demonstrated using a real robotic system, and is shown to consistently perform better than other state-ofthe-art techniques.

Advances in Neural Information Processing Systems (NIPS)

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Derivative-Free Semiparametric Bayesian Models for Robot Learning

Diego Romeres***, Devesh K. Jha*****, Alberto Dalla Libera**†**, Alessandro Chiuso**†**, Daniel Nikovski***

*MERL, Cambridge, MA. {romeres, jha, nikovski}@merl.com

†University of Padova, Padova, Italy. {dallalibera, chiuso}@dei.unipd.it

Abstract

Model-Based Reinforcement Learning (MBRL) is gaining much interest in the robot learning community; in MBRL, the model serves as a representation which is largely task-invariant, and thus can facilitate transfer of knowledge across multiple tasks in the same domain. Learning reliable models for physical systems, however, remains a challenging problem. This paper summarizes recent semiparametric and derivative-free modelling techniques, and presents some key points for a new methodology to formulate derivative-free semiparametric Bayesian models with applications to robot learning. The modeling technique is demonstrated using a real robotic system, and is shown to consistently perform better than other state-of-the-art techniques.

1 Motivation and Introduction

Model-Based Reinforcement Learning is increasingly gaining attention, due to the ability of learned models to efficiently compress information in the data that can also be transferred across multiple tasks [1, 2, 3, 4]. However, learning accurate global models for long-term predictions from limited measured data is a challenging problem, and even more so in use cases encountered in robotics where models should account for causal constraints between variables, non-linearities such as non-viscous friction, contacts etc. Oftentimes, models are built with derivative variables as inputs [1, 5] (e.g. velocities, accelerations); the latter can often be difficult or impossible to measure, and ad-hoc filters are used to estimate these derivatives are used. By doing so, often delays and non-ideal noise rejection are introduced during the modelling phase, which can negatively affect the modelling effort, or even cause it to fail altogether.

Contributions. The purpose of this work is to present a brief overview and analysis of some recently introduced derivative-free Bayesian models for learning inverse dynamics of robotic systems [6], extend them for learning forward dynamics and hopefully lead to discussion. Moreover, inspired by the recent success of semiparametric models in robot dynamics learning [7, 8, 9], which combine existing physics knowledge with a data-driven component, we propose a new methodology to describe physical models in a derivative-free fashion, and thus combine semiparametric models and derivative-free techniques. The effectiveness of this technique is demonstrated by the accuracy in long-term predictions, and in controlling a real Furuta pendulum system in an MBRL setting.

1.1 Problem Formulation

Consider a robotic system with n_{dof} degrees of freedom (DoF) and state vector $\mathbf{x}$. From first-principle physical laws, it can be argued that the state vector $\mathbf{x}$ consists of positions and their derivatives (velocities and accelerations) for all joints of the robot. Then, the forward dynamics of the robot $\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k)$ at time instant k is the state evolution map given the control input $\mathbf{u}_k$, whereas the inverse dynamics $\mathbf{u}_k = g(\mathbf{x}_{k+1}, \mathbf{x}_k)$ provides control inputs following a specific trajectory of the robot. In this paper, we shall consider Bayesian models that combine data-driven

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

techniques and physics-based prior structure, called semiparametric models. The generic form of the models under consideration is

$$y = f(x) + \epsilon \tag{1}$$

where $x \in \mathbb{R}^m$ are the input variables, $y \in \mathbb{R}$ are the outputs, $\epsilon \ \sim \mathcal{N}(0, \sigma^2 I_m)$. The inputs x and the outputs y are left in generic form, because we are interested in learning both the forward as well as inverse dynamic models, which have different input and output variables.
Moreover, we are interested in the choice of the input variables x. Indeed, in many robotic systems, velocities and accelerations cannot be sensed directly by the robot, and instead they need to be estimated by numerical differentiation using pre-processing filters on the position measurements. It is well known that applying filter operations to noisy data might lead to an increasing amount of error and to fictitious delays in the derivative signals. In this work, we investigate how to derive semiparametric Bayesian models on the past history of measurements, without the need to compute the derivatives of the measured state variables (and thus, in a derivative-free fashion).

2 Model Learning

In this Section, we first briefly review Bayesian model classes used to learn the inverse and forward dynamics of a robot. Then, we review the possibilities for learning derivative-free nonparametric models that the authors already proposed in previous work. Finally, we propose new guidelines in order to make also the physical models and therefore also the physical component of the semiparametric models derivative-free.

The model function $f(\cdot)$ in (1) is modeled a priori as a Gaussian process with zero mean and a covariance function also called the kernel function $k(x_i, x_j) = \text{cov}\,[f(x_i), f(x_j)]$. The characterization of the kernel function is what will define and differentiate the following models.

Physically inspired kernels. The physical model of a system is derived by first principles, and the model information might be used to identify a feature space over which the evolution of the system is linear. More precisely, assume that model (1) can be written in the form $f(x) = \phi(x)^T \boldsymbol{w}$, where $\phi(x) : \mathbb{R}^m \to \mathbb{R}^q$ is a known nonlinear function obtained by first principles that maps the input vector x in the physically inspired features space and $\boldsymbol{w}$ is the vector of unknown parameters. Modelling the unknown parameter with a Gaussian distribution a priori, $p(\boldsymbol{w}) \sim \mathcal{N}(0, \Sigma_{PI})$ with $\Sigma_{PI} \in \mathbb{R}^{q \times q}$ the covariance matrix, leads to the expression of the physically inspired kernel (PI)

$$k(x_i, x_j) = \phi(x_i)^T \Sigma_{PI} \phi(x_j) \tag{2}$$

The kernel defined in (2) is a linear kernel in the features $\phi(\cdot)$. For later convenience, we define also a homogeneous (without the bias constant) polynomial kernel in the feature space

$$k^p_{poly}(\phi(x_i), \phi(x_j)) = \left(\phi(x_i)^T \Sigma_{PI} \phi(x_j)\right)^p \tag{3}$$

which is a general case of the linear kernel.

Nonparametric kernel. Following the Gaussian process regression framework [10], the structure of the kernel has to be chosen by the user according to their beliefs about the process to be estimated. A common option is the Radial Basis Function kernel (RBF), $k_{NP}(x_i, x_j) = k_{RBF}(x_i, x_j)$.

Semiparametric kernel. This approach combines the physically-inspired and the nonparametric kernels. Several semiparametric kernels have been proposed [7, 9], and here we follow the method that considers the kernel function to be the sum of the covariances:

$$k(x_i, x_j) = \phi(x_i)^T \Sigma_{PI} \phi(x_j) + k_{NP}(x_i, x_j). \tag{4}$$

The semiparametric model takes advantage of the global property of the parametric model as well as of the flexibility of the nonparametric model. The advantage of using this semiparametric (SP) kernel over k_{NP} and k_{PI} has been shown in [7, 9] for inverse dynamics modelling, and in [11] for forward dynamics modeling. The reader is referred to these papers and the literature referred to therein.

2.1 Derivative-Free Semiparametric Models

The robot dynamics (both inverse and forward) computed from physical first principles are functions of joint positions, velocities, and accelerations. However, it is often the case that joint velocities and accelerations cannot be measured by the robot sensors, and computing them by numerical differentiation might severely hamper the final solution. This is a well known and highly discussed problem, see e.g., [12, 13, 14, 15, 16] and it is usually partially addressed by ad-hoc filter design. However, this requires users' knowledge and experience in tuning the filters' parameters, and still introduces fictitious delays. Alternatively, the past measurements of the positions can be considered. So we define

$$\boldsymbol{x}_{k^-} := [\, q_{k^-}^{1^\top} \, \dots \, q_{k^-}^{n_{dof}^\top} \,]^\top \in \mathbb{R}^{(M+1)n_{dof}}, \quad q_{k^-}^i := [\, q_k^i \, q_{k-1}^i \, \dots \, q_{k-M}^i \,]^\top \in \mathbb{R}^{M+1}, \tag{5}$$

to be the vector of the past joint positions and the vector of the past of the i-th joint position, respectively, in the time window $[t-M, t]$, where M, sufficiently large, has been fixed. Postulate that the output y_k can be written as a non-linear function of a "features vector" $\xi_k := [\, \xi_k^{1^\top} \, \dots \, \xi_k^{n_{dof}^\top} \,]^\top$, defined as a linear function of past measurements $q_{k^-}^i$ for $i = [1 \, \dots \, n_{dof}]$

$$\xi_k^{i^\top} = R\, q_{k^-}^{i^\top}, \quad \text{where } R \in \mathbb{R}^{k \times (M+1)}. \tag{6}$$

Thus, $f(\cdot)$ can be defined as a Gaussian process with kernel function $k(\xi_i, \xi_j) = k_{NP}(\xi_i, \xi_j)$. The rows of the matrix R may compute an approximation of the derivatives of q_k^i. We may thus say that the approach considered in this section generalizes the standard case. This technique was recently introduced in [6], where the authors discussed three types of parametrization of the feature matrix R which plays a crucial role. The discussed options are that the matrix R can be diagonal or low rank or explicitly parametrize the velocity and acceleration. The reader is referred to [6] for details and a more detailed summary is presented in Appendix 4.

This derivative-free technique works only for the NP estimator, and cannot be applied directly to the physics-based model, because the basis functions $\phi(\cdot)$ are functions of the physical quantities position, velocity, and acceleration. In order to handle this limitation we propose a set of guidelines to design a kernel-based model completely derivative-free.

The new methodology and its effectiveness is explained by modelling the forward dynamics of a benchmark robotic system in control theory, the Furuta Pendulum (FP), [17, 18]. The FP is an underactuated system known to have highly nonlinear dynamics. In particular, it is composed of two arms: the base arm, which is constrained by an actuated revolute joint to perform circular movements in a plane parallel to the ground, and the pendulum arm, which rotates around the principal axis of the base arm. The base arm and the pendulum arm angles are denoted by α and θ. Further details of the FP system, explanation of its complex dynamic and of the delays present in this system are explained in Section 5. When neglecting the behaviors causing delays (see Sections 5), the expression of the discretized forward dynamics $\Delta_{\theta_{k+1}} = \theta_{k+1} - \theta_k$ is approximated by the physical linear model

$$\Delta_{\theta_{k+1}} = \begin{bmatrix} -\ddot{\alpha}\cos(\theta) & \dot{\alpha}^2 \sin(2\theta) & \dot{\theta} & \sin(\theta) \end{bmatrix} \boldsymbol{w'} = \phi_{\ddot{\theta}}(\ddot{\alpha}, \dot{\alpha}, \dot{\theta}, \theta)^T \boldsymbol{w'}. \tag{7}$$

The last equation provides an approximated model of the FP evolution as function of $\mathbf{x}_k = \left[\theta_k, \dot{\theta}_k, \ddot{\theta}_k, \alpha_k, \dot{\alpha}_k, \ddot{\alpha}_k\right]$. However, model (7) does not account for the presence of the unknown delay and other unmodeled long-term effects dependent on the past history of the state. To overcome these problematic aspects, the state is defined in a derivative-free fashion $\boldsymbol{x}_{k^-} := [\, \alpha_{k^-}^\top \, \theta_{k^-}^\top \,]^\top$, and we propose to define the PI kernel through a combination of polynomial kernels inspired by the forward physical model (7). While the resulting kernel will be shown for the specific model (7), we present here a set of general guidelines that can be applied to derive a free-derivative PI kernel robust to delay for a wide range of given physical models. The guidelines to derive a robust PI kernel are:

- Each and every position, velocity, or acceleration term in $\phi_{\ddot{\theta}}(\cdot)$ is replaced by a distinct polynomial kernel $k_{poly}^p(\cdot, \cdot)$ of degree p, where p is equal to the degree of the original term;
- The input of the kernel is either α_{k^-} or θ_{k^-}, depending on whether $k_{poly}^p(\cdot, \cdot)$ is substituting a term that is a function of θ or α;
- If a state variable appears into the model transformed by a function $g(\cdot)$, e.g., $\sin(\theta_{k^-})$, the input to $k_{poly}^p(\cdot, \cdot)$ becomes the variable transformed by the same function $g(\cdot)$, e.g., $k_{poly}^p(g(\theta_{k^-}), g(\theta_{k^-}))$.

Following these guidelines, the expression of the physically inspired kernel for model (7) is

$$\begin{aligned} k_{PI}(\boldsymbol{x}_{i^-}, \boldsymbol{x}_{j^-}) := {} & k_{poly}^1(\alpha_{i^-}, \alpha_{j^-}) k_{poly}^1(\cos(\theta_{i^-}), \cos(\theta_{j^-})) + k_{poly}^1(\theta_{i^-}, \theta_{j^-}) \\ & + k_{poly}^1(\sin(\theta_{i^-}), \sin(\theta_{j^-})) + k_{poly}^2(\alpha_{i^-}, \alpha_{j^-}) k_{poly}^1(\sin(2\alpha_{i^-}), \sin(2\alpha_{j^-})) \end{aligned} \tag{8}$$

In this way, a large set of suitable basis function is defined, which GP regression can use to capture the dynamics of the system, even the ones that depend on velocity and accelerations.
Finally, the derivative-free semiprametric kernel can be defined as:

$$k_{SP}(\boldsymbol{x}_{i^-}, \boldsymbol{x}_{j^-}) = k_{PI}(\boldsymbol{x}_{i^-}, \boldsymbol{x}_{j^-}) + k_{NP}(\xi_i, \xi_j). \tag{9}$$

The estimator obtained with this kernel,f_{SP}, is now compared with the other derivative-free models f_{NP}, f_{PI}, and a standard NP model with inputs position, velocity, and acceleration f_{der}. The detailed explanation of the experiments is out of the scope of this paper. Figure 1 represents the performance in prediction on validation data: in the top sub-figure, the data, D_{sin}, are of the same kind as the training data, and in the bottom sub-figure, the data, D_{trw}, have different trajectories. The model f_{SP} outperforms the others, and also exhibits almost identical performance in the two cases. In Figure 2, the performance for k-steps ahead, the model is evolving in open loop, are presented for the derivative-free models in D_{trw}. Accurate performance is achieved by f_{SP} for 100steps ahead, outperforming the other methods. The accuracy of the model is confirmed since we were able to swing up the FP with an open-loop controller obtained in an MBRL setting as described in Appendix 5.1.

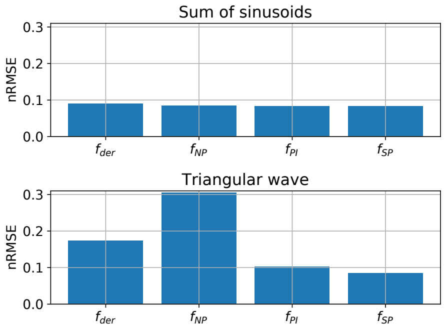

Figure 1: Bar-plot with the nRMSE of f_{der}, f_{NP}, f_{PI} and f_{SP} on two test datasets D_{sin},D_{trw}.
D_{sin}: $nRMSE = [0.091, 0.085, 0.084, 0.083]$
D_{trw}: $nRMSE = [0.174, 0.305, 0.102, 0.085]$

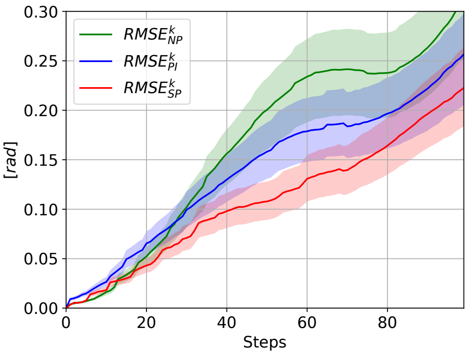

Figure 2: Performance of the derivative-free models f_{NP}, f_{PI} and f_{SP} at k-steps ahead in terms of RMSE, $RMSE^k$, with relative confidence intervals.

3 Discussion

In this paper, we summarized the recent results for semiparametric and derivative-free Bayesian models. Moreover, we presented a new general methodology to describe physical dynamical models in a derivative-free kernel fashion, which leads to the definintion of a new model class of Derivative-Free Semiparametric Bayesian (DFSB) models.

Semiparametric Bayesian models (which, roughly speaking, are nonparametric models equipped also with basis functions given by the physics), have been shown to achieve accurate performance and outperform physical and nonparametric models in several situations. First, while modelling both inverse dynamics [7, 9] and forward dynamics in [11]. Second, in both cases when the physical model was poorly or highly performing in [7] and in this work, respectively. Third, in prediction of n-steps ahead even if the cost function was designed only for 1-step ahead prediction. Derivative-free models have also been shown to be efficient in all these cases combining the results of [6] and the one obtained in this paper. Moreover, since we are considering the past history of the measurements, the method is robust to system delays, and can autonomously select which temporal instants are most relevant. These characteristics make the DFSB models a promising highly accurate and robust method for robot learning.

The downside of these models is that they might be harder to train, because of their complexity and a higher number of hyperparameters to estimate. Therefore, a larger training data set might be required. This effect is mitigated by the fact that semiparametric models are more data efficient than their nonparametric counterparts, as shown for example in [11].

The design of the matrix R offers the possibility of future variations, and adaptability to different applications. For example, the rows can be parameterized as band-pass filters in order to detect delays in the system, and use only the appropriate temporal lags. Moreover, to be robust to highly noisy data, this work can be extended to heteroscedastic noise models.

4 APPENDIX: Derivative-free features structure

A detailed description on the possible structures of R can be found in [6], and is briefly summarized here.

Derivative-free features. The simplest choice is to take $R = I_{M+1}$, that is, the feature vector coincides with the M past measured joint positions. As an alternative, it is possible to choose

$$R = diag\left(r_1, \ldots, r_{M+1}\right), \quad \text{with } r_i \in \mathcal{R}, i = 1, .., M+1. \tag{10}$$

That is, the feature vector is a weighted version of the past measured joint positions.

Derivative-free features with reduced rank. Alternatively, R can be fully parameterized with a number of features k smaller than $M+1$. For instance, the physics suggests that the right number of features should be equal to 3 (position, velocity, acceleration). The role of the features is to compress the useful information available in $\boldsymbol{x}_{k^-}$ so as to render the learning procedure more robust.

$$R = \left[r_1, \ldots, r_k\right]^\top, \quad \|r_i\| = 1, \quad \text{where } k < M, r_i \in \mathbb{R}^{M+1} \tag{11}$$

These features include all the possible linear and causal numerical differentiation and filtering operations. The price of this generality is a large number of hyperparameters $(k-1)(M+1)$ to estimate.

Structured derivative-free input locations with reduced rank. Matrix R can be parameterized to explicitly estimate the physical quantities, with $k = 3$ features: the first is composed of the measurement variables $q(t)$ (e.g. the position) while the other two rows will attempt to estimate explicitly velocities and accelerations. Since the velocities and accelerations can be computed by a first order backward difference and by a second order backward difference, respectively, both filtered by a first order low pass filter, that is

$$\dot{q}_i(t) \approx \frac{1 - z^{-1}}{T_s} \frac{1}{1 - \beta_1 z^{-1}} q_i(t), \qquad \ddot{q}_i(t) \approx \frac{1 - 2z^{-1} + z^{-2}}{T_s^2} \frac{1}{1 - \beta_2 z^{-1}} q_i(t)$$

where z^{-1} is the backward shift operator, $T_s > 0$ is the sampling time and $0 < \beta_1, \beta_2 < 1$ represent the poles of the filters. We resort to a partial fraction decomposition to rewrite the above expressions as a function of $q(s^-)$, that is:

$$\dot{q}_i(t) \approx \alpha_1 q_i(t) + \sum_{t=1}^{M} \alpha_1 \beta_1^{t-1}(\beta_1 - 1) q_i(s-t)$$

$$\ddot{q}_i(t) \approx \alpha_2 q_i(t) + \alpha_2(\beta_2 - 2) q_i(s-1) + \sum_{t=2}^{M} \alpha_2 \beta_2^{t-2}(\beta_2^2 - 2\beta_2 + 1) q_i(s-t)$$

where $\alpha_1 = 1/T_s$ and $\alpha_2 = 1/T_s^2$. Here, we exploited the fact that a (stable) low-pass filter can be approximated by a finite impulse response (FIR) filter with length M, where the latter is chosen to be sufficiently large. Accordingly, we have

$$R = \begin{bmatrix} 1 & 0 & \ldots & \ldots & 0 \\ \alpha_1 & \alpha_1(\beta_1 - 1) & \ldots & \alpha_1 \beta_1^{t-1}(\beta_1 - 1) & \ldots \\ \alpha_2 & \alpha_2(\beta_2 - 2) & \ldots & \alpha_2 \beta_2^{t-2}(\beta_2^2 - 2\beta_2 + 1) & \ldots \end{bmatrix}.$$

A nice property of this characterization is that the number of hyperparameters in R is small $[\alpha_1, \alpha_2, \beta_1, \beta_2] \in \mathbb{R}^4$, independently of the length of the past temporal lags M, which can be arbitrarily chosen.

5 APPENDIX: Furuta Pendulum

In this Section we provide a brief description of the FP and the setup adopted in the experiments.

A symbolic description of the FP is reported in Figure 4, where we refer to "Arm-1" and "Arm-2" as the base arm and the pendulum arm respectively. Mechanically, the FP is a normal gravitational pendulum of length L_2 (Arm-2 in Figure 4), suspended from the end of a turntable arm (Arm-1 in Figure 4). The pendulum arm is attached with a perpendicular frictionless bearing to the base arm, at

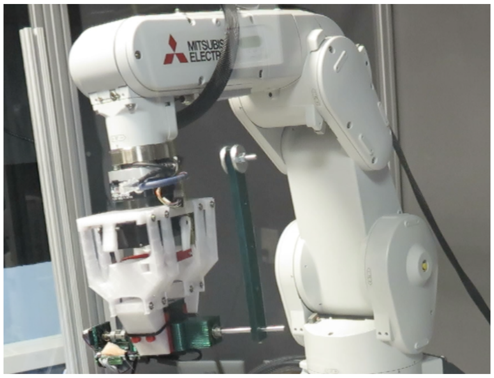

Figure 3: The Furuta Pendulum held in the wrist joint of the robotic arm by the (white) 3D printed gripper at the swing-up position.

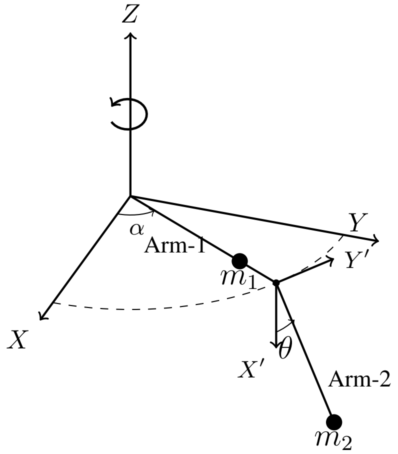

Figure 4: A schematic diagram of the FP with various system parameters and state variables. Arm-1 has length L_1 and mass m_1, and Arm-2 has length L_2 and mass m_2. The center of mass for the two arms are at l_1 and l_2.

distance L_1 from the center of the base arm's actuation source. The torque actuation axis is vertical, so the actuation arm sweeps a flat horizontal plane, while the direction of the pendulum rotation axis is always aligned with the turntable arm.

In [19], the authors derived an accurate physical model of the FP using the Lagrangian approach. Let m_i, l_i, and J_i be respectively the mass, the center of mass and the inertia of the i-th arm, with $i = 1, 2$, while α and θ are the variables pointing out the angles of the two arms. The froward dynamics equation of the pendulum are:

$$\begin{aligned}\ddot{\theta} &= \frac{-\ddot{\alpha} m_2 L_1 l_2 \cos(\theta) + \frac{1}{2}\dot{\alpha}^2 \hat{J}_2 \sin(2\theta) + b_2 \dot{\theta} + g m_2 l_2 \sin(\theta)}{\hat{J}_2} \\ &= \begin{bmatrix} -\ddot{\alpha}\cos(\theta) & \dot{\alpha}^2 \sin(2\theta) & \dot{\theta} & \sin(\theta) \end{bmatrix} \boldsymbol{w} = \phi_{\ddot{\theta}}(\ddot{\alpha}, \dot{\alpha}, \dot{\theta}, \theta)^T \boldsymbol{w}, \end{aligned} \tag{12}$$

where $\hat{J}_1 = J_1 + m_1 l_1^2 + m_2 L_1^2$ and $\hat{J}_2 = J_2 + m_2 l_2^2$.

Let $\Delta_{\theta_{k+1}} = \theta_{k+1} - \theta_k$. Assuming that $\ddot{\theta}_k$ is constant over all the sampling interval time δ_t, the $\Delta_{\theta_{k+1}}$ equation is:

$$\Delta_{\theta_{k+1}} = \dot{\theta}_k \delta_t + \frac{\delta_t^2}{2} \ddot{\theta}_k = \dot{\theta}_k \delta_t + \frac{\delta_t^2}{2} \phi_{\ddot{\theta}}(\ddot{\alpha}_k, \dot{\alpha}_k, \dot{\theta}_k, \theta_k)^T \boldsymbol{w} = \phi_{\ddot{\theta}}(\ddot{\alpha}_k, \dot{\alpha}_k, \dot{\theta}_k, \theta_k)^T \boldsymbol{w'}. \tag{13}$$

In the setup adopted, the source of actuation is the wrist joint of an industrial robotic arm (the MELFA RV-4FL, see Figure 3). This robot can only be controlled in a position-control mode, by sending a sequence of desired set-points. Since the pendulum inertia is considerably lower than the one of the robot joint, it is reasonable to assume that the α dynamics evolves independently from the one of θ. More precisely, we noticed that, in order to guarantee repeatability, the inner controller of the MELFA robot controls the evolution of α in deterministic fashion based on the sequence of set-points. Therefore, there we know the forward dynamic of this variable exactly and only the forward dynamic of θ needs to be estimated.

Moreover, the FP is held by a 3D-printed gripper, and is not rigidly fixed to the wrist joint of the robot. This results in significant interplay between the gripper and the FP base link leading to vibration of the base arm along with the rotational motion. More importantly, the base arm of the FP is not rigid, and has significant elasticity. This results in significant delay in actuation of the pendulum arm. These factors make the dynamics of the FP presented in this paper more involved when compared to the one in Equations (13).

5.1 Control using iLQG

We present some results of implementing the iLQG [20] controller on the real FP. The learned SP model was linearized using numerical differentiation and used for trajectory optimization using iLQG. The cost structure imposed for input saturation was able to contain the input within the allowable control volume and thus, feasible trajectories were obtained during the optimization process. The trajectories obtained by the iLQG algorithm were implemented in an open-loop fashion on the real system, and the results are shown in Figure 5. The FP is able to swing up with near-zero velocity to the goal position; however, as expected, the open-loop control sequence was not able to stabilize it. In Figure 5, we obtained an accurate agreement between the trajectories on θ obtained from the iLQG control sequence using the SP model and the real robot, which shows the long-horizon predictive accuracy of the learned model. Notice that the models might be less accurate around the unstable equilibrium point, because of the lack of data which are harder to collect for training purposes. At this point, the authors would like to point out that previous results on using GP-based MBRL were not able to swing-up a simpler class of inverted pendulum (cart-pole system) as the learned models were not accurate for long-term prediction [21]. Due to the high control rate (140 Hz) for our FP, we were not able to implement the iLQG in a model-predictive control (MPC) fashion which could also stabilize the FP at the unstable equilibrium point. In-depth research of these topics has been left as future research.

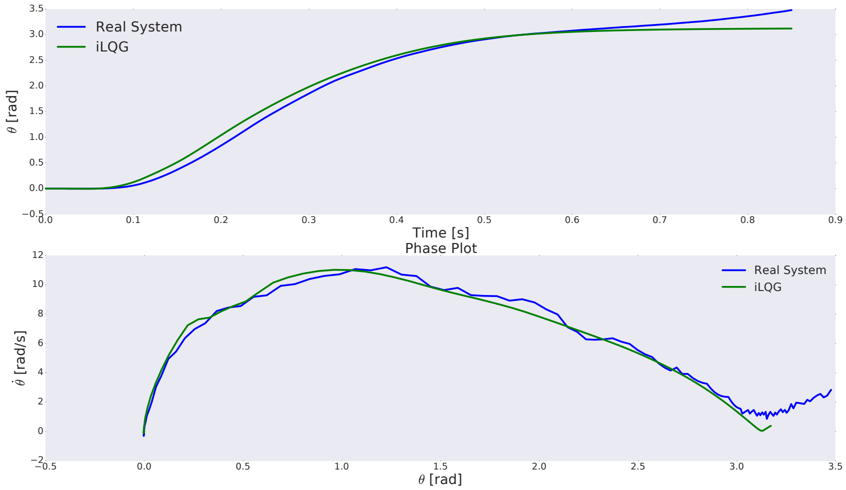

Figure 5: Performance of the trained iLQG trajectory on the swing-up control of the Furuta pendulum. The controller can swing up the pendulum and the model prediction works for about 0.7 s which is equal to 100 time steps at a control rate of 140 Hz.

References

[1] Marc Peter Deisenroth, Dieter Fox, and Carl Edward Rasmussen. Gaussian processes for data-efficient learning in robotics and control. *IEEE transactions on pattern analysis and machine intelligence*, 37(2):408–423, 2015.

[2] Sergey Levine and Pieter Abbeel. Learning neural network policies with guided policy search under unknown dynamics. In *Advances in Neural Information Processing Systems*, 2014.

[3] Sergey Levine, Nolan Wagener, and Pieter Abbeel. Learning contact-rich manipulation skills with guided policy search. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 156–163. IEEE, 2015.

[4] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[5] Marc Toussaint, Kelsey R Allen, Kevin A Smith, and Josh B Tenenbaum. Differentiable physics and stable modes for tool-use and manipulation planning. In *Proc. of Robotics: Science and Systems (R:SS 2018)*, 2018.

[6] D. Romeres, M. Zorzi, R. Camoriano, S. Traversaro, and A. Chiuso. Derivative-free online learning of inverse dynamics models. *ArXiv e-prints*, September 2018.

[7] Diego Romeres, Mattia Zorzi, R. Camoriano, and Alessandro Chiuso. Online semi-parametric learning for inverse dynamics modeling. In *Conference on Decision and Control*. IEEE, 2016.

[8] Tingfan Wu and Javier Movellan. Semi-parametric Gaussian process for robot system identification. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 725–731, 2012.

[9] D. Nguyen-Tuong and J. Peters. Using model knowledge for learning inverse dynamics. In *IEEE International Conference on Robotics and Automation*, 2010.

[10] C.E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[11] Diego Romeres, Devesh Jha, Alberto DallaLibera, Bill Yerazunis, and Daniel Nikovski. Learning hybrid models to control a ball in a circular maze. *arXiv preprint*, abs/1809.04993, 2018.

[12] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010.

[13] John Hollerbach, Wisama Khalil, and Maxime Gautier. Model identification. In *Springer Handbook of Robotics*, pages 321–344. Springer, 2008.

[14] Krzysztof R Kozlowski. *Modelling and identification in robotics*. Springer Science & Business Media, 2012.

[15] John J Craig. *Introduction to robotics: mechanics and control*, volume 3. Pearson Prentice Hall Upper Saddle River, 2005.

[16] Duy Nguyen-Tuong and Jan Peters. Model learning for robot control: a survey. *Cognitive Processing*, 12(4):319–340, 2011.

[17] Katsuhisa Furuta, M Yamakita, and S Kobayashi. Swing-up control of inverted pendulum using pseudo-state feedback. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 206(4):263–269, 1992.

[18] K Furuta, M Yamakita, S Kobayashi, and M Nishimura. A new inverted pendulum apparatus for education. In *Advances in Control Education 1991*, pages 133–138. Elsevier, 1992.

[19] Benjamin Seth Cazzolato and Zebb Prime. On the dynamics of the furuta pendulum. *Journal of Control Science and Engineering*, 2011:3, 2011.

[20] Yuval Tassa, Tom Erez, and Emanuel Todorov. Synthesis and stabilization of complex behaviors through online trajectory optimization. In *Intelligent Robots and Systems (IROS)*. IEEE, 2012.

[21] Joschka Boedecker, Jost Tobias Springenberg, Jan Wülfing, and Martin Riedmiller. Approximate real-time optimal control based on sparse gaussian process models. In *Adaptive Dynamic Programming and Reinforcement Learning (ADPRL), 2014 IEEE Symposium on*, pages 1–8. IEEE, 2014.