2012 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 7-12, 2012. Vilamoura, Algarve, Portugal

# Semi-parametric Gaussian Process for Robot System Identification

Tingfan Wu Javier Movellan
Machine Perception Lab, UCSD
{ting,movellan}@mplab.ucsd.edu

***Abstract*— One reason why control of biomimetic robots is so difficult is the fact that we do not have sufficiently accurate mathematical models of their system dynamics. Recent non-parametric machine learning approaches to system identification have shown good promise, outperforming parameterized mathematical models when applied to complex robot system identification problems. Unfortunately, non-parametric methods perform poorly when applied to regions of the state space that are not densely covered by the training dataset. This problem becomes particularly critical as the state space grows. Parametric methods use the available data very efficiently but, on the flip side, they only provide crude approximations to the actual system dynamics. In practice the systematic deviations between the parametric mathematical model and its physical realization results in control laws that do not take advantage of the compliance and complex dynamics of the robot. Here we present an approach to robot system identification, named Semi-Parametric Gaussian Processes (SGP), that elegantly combines the advantages of parametric and non-parametric approaches. Computer simulations and a physical implementation of an underactuated robot system identification problem show very promising results. We also demonstrate the applicability of SGP to articulated tree-structured robots of arbitrary complexity. In all experiments, SGP significantly outperformed previous parametric and non-parametric approaches as well as previous methods for combining the two approaches.**

## I. Introduction

System identification has been one of the major challenges for progress in the control of biologically inspired robots. Obtaining a good model is crucial for finding control laws that can take advantage of the compliance and complexity of these robots. For example, when using computed torque control with an inaccurate model, the un-modeled dynamics are treated as noise. The results of this are stiff control laws to deal with any un-modeled deviations from the system dynamics. More sophisticated approaches, like differential dynamic programming or iterative LQR [16] can take advantage of the dynamic properties of complex robots but these approaches require accurate models of the robot dynamics. In practice, accurate dynamical models are critical to develop control laws that are compliant, energy efficient and safe.

There are two major approaches for system identification: parametric and non-parametric. Parametric approaches rely on parameterized Newtonian physics models of the robot's dynamics. The advantage of these models is that they capture a great deal of prior knowledge that does not need to be learned from data. For example, we know that robots are subject to gravitational forces, viscous forces and joint constraints. It is indeed wasteful to have to go through a laborious data-gathering and machine learning process to discover these well known constraints. The disadvantage of parametric models is that they are only crude idealizations of the actual system dynamics. For traditional industrial robots these un-modeled dynamics are often ignorable. However, for modern biologically inspired robots these errors result in significant control inefficiencies.

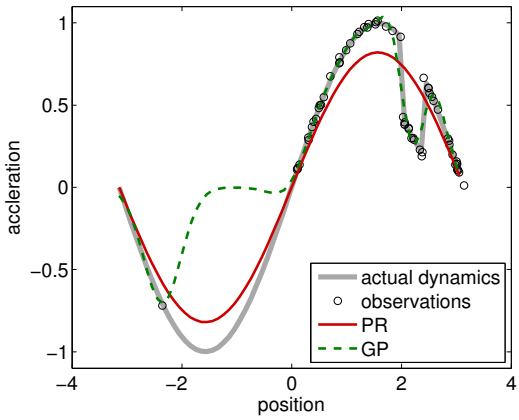



Fig. 1. Comparing parametric (PR) model and non-parametric model, Gaussian process (GP) on learning acceleration of a pendulum.

Non-parametric machine learning approaches avoid the model under-specification problem by directly learning from training sample data. Two popular non-parametric approaches include Locally Weighted Projection Regression (LWPR) [17], [14] and Gaussian Process (GP) [10] and neural networks [15]. The major advantage of these models is the ability to fit virtually any dynamics as long as they are locally smooth. The predictions are then based on interpolation between similar (or nearby) instances in the training data. However, since non-parametric models rely on local neighborhood training data to make predictions, they do not generalize well to unexplored state regions with little or no training data. Covering the entire state space becomes exponentially harder as the complexity and number of degrees of freedom in the robot increases. Thus it appears quite desirable to combine the benefits of parametric and non-parametric approaches. However, doing so in an efficient way is not trivial. A reasonable approach would be to first fit a parametric model and then fit a non-parametric model to the errors made by the parametric model. However such an approach is not ideal: it sequentially optimizes two models, rather than jointly optimizing them. Here we propose an

approach that allows joint optimal inference with parametric and non-parametric models that efficiently uses the available data. The approach is formalized in the framework of what we call Semi-parametric Gaussian Process (SGP), a type of Gaussian process designed for optimal inference with combinations of parametric and non-parametric models.

### A. Toy Problem

Here we illustrate the properties of parametric and non-parametric approaches, with a simple toy problem. We collect data from a pendulum that happens to have a significant friction irregularity when it reaches angles between 2 and 2.5 radians. We use a parametric model derived from the classic viscous-free equations of motion

$$\ddot{\theta} = w \sin(\theta),$$

where $w$ is an unknown parameter. Note this model encodes a great deal of information about the forces and constraints operating on the system, however it does not know about the fact that there are some significant friction forces that happen to be a function of $\theta$. To test this model we collected a sample of data and found the value of $w$ that best fit this data. We call this approach Parametric Regression (PR). Fig.1 shows that PR successfully captured the general trend of the data. It was also able to make reasonable predictions around state regions that had very little training data. However, as expected, PR could not capture the irregularities due to friction in the interval between 2 and 2.5 radians. We also tried a standard non-parametric model (GP) shown to perform well in the robotics system identification literature. GP did a remarkably good job at capturing the effects of state dependent friction. However it did not generalize well in regions that had little or no training data.

Below we first review related work in semi-parametric and non-parametric approaches for system identification. Next, we present our formalization of the problem of optimally combining parametric and non-parametric models, in what we call a Semi-parametric Gaussian Process. After deriving the optimal inference (system identification) equations we study its performance on the toy problem presented above and in actual physical system identification problems, including a simple reaction wheel and a 3-link robotic arm.

## II. Related Work

Semi-parametric regression has been a popular class of methods when partial knowledge is available [13]. The idea of combining Gaussian Process Regression with a global linear model was first explored in [2] where they use a GP to model the residual from a polynomial regression.

Gaussian Process has been used widely for system identification in robotics. However, most work used GP as a pure non-parametric model [12], [5], [6], [10], [3], which did not utilize the prior knowledge of the system and applied GP to learn the function directly. Such blackbox approaches are convenient but the lack of global model[1] unavoidably leads to poor generalization performance in unseen space and thus requires large amount of training data to cover the operational space.

Recent work [9] on modeling the dynamics of a blimp suggested using GP to learn the residual of the parametric Newtonion differential model. Their combined model achieved significant performance improvement over pure Newtonian model, which further demonstrated the benefit of semi-parametric model. However, since their model was not linear-in-parameter. They could not benefit from simultaneous optimization of parametric and non-parametric model. Instead, the non-parametric model is applied after parametric identification which may result in suboptimal model as shown in Sec. IV-A.

[1]In fact, GP probably should be classified as global model. However, due to the typical use of local basis function (eg. radial basis function), the impact of a data point is restricted to its neighborhood.

## III. Semi-parametric Gaussian Process

We define an SGP as a collection of stochastic processes indexed by $u \in \mathcal{R}^p$ and subject to the following constraints

$$W = \mu + R \tag{1}$$

$$X(u) = f(u)W + Z(u), \text{ for all } u \in \mathcal{R}^p \tag{2}$$

$$Y(u) = g(u)X(u) + V(u), \text{ for all } u \in \mathcal{R}^p \tag{3}$$

where

- $\mu \in \mathcal{R}^{n_w}$ is the prior mean of $W$
- $R$ is an $n_w$-dimensional Gaussian random vector with zero mean and variance matrix $\sigma_w \in \mathcal{R}^{n_w \times n_w}$.
- $X(u), Y(u)$ take values in $\mathcal{R}^{n_x}$ and $\mathcal{R}^{n_y}$ respectively.
- $f : \mathcal{R}^p \rightarrow \mathcal{R}^{n_x \times n_w}$ is a matrix function of a vector.
- $Z$ is a $n_x$-dimensional Gaussian process with zero mean and with covariance structure specified below.
- $g : \mathcal{R}^p \rightarrow \mathcal{R}^{n_y \times n_x}$ is a matrix function.
- $V$ is a $n_y$ dimensional Gaussian process with zero mean and with covariance structure specified below

We developed the SGPs specifically for system identification in robotics problems. $u$ represents the current state: joint angles, angular velocities and either control signals (in forward models) or accelerations (in inverse dynamics models). $f(u)$ represents the prediction made by the parametric model. It is either the predicted acceleration (in a forward model) or the torque that produced the observed acceleration (in an inverse model), and $W$ represents the unknown model parameters. Here we take advantage of the fact that the inverse dynamics models of any articulated robots are linear-in-parameter [8], [7]. The prior knowledge about $W$ is modeled by a prior mean $\mu$ and a prior covariance matrix $\sigma_w$. $Z$ represents structural deviations from the parametric model. $Z$ is independent of $R$ and its structure is captured via a matrix function $\kappa : \mathcal{R}^{p \times p} \rightarrow \mathcal{R}^{n_x \times n_x}$ that controls the covariance structure of $Z$,

$$\mathbb{C}(Z(u), Z(\tilde{u})) = \kappa(u, \tilde{u}), \text{ for } u, \tilde{u}$$

A popular choice is the squared exponential kernel

$$\kappa(u, \tilde{u})_{ij} = \delta(i, j)\sigma_z \exp(-(u - \tilde{u})^T P^{-1}(u - \tilde{u})/2). \tag{4}$$

where $\sigma_z > 0$ and $P = \mathbf{diag}(\ell_1^2, \ell_2^2, \ldots, \ell_p^2)$ are model parameters. $X(u)$ represents the fused prediction made by

the parametric part of the model $f(u)W$ and the non-parametric part of the model $Z(u)$. $Y(u)$ is a vector of sensory observations. It is assumed to be a known function of the state $X(u)$ plus additive white sensor noise $V(u)$. Thus the $V(u)$ vectors are independent of $Z$, $R$ and of each other, i.e., for all $u, \tilde{u} \in \mathcal{R}^{n_u}$, the covariances are

$$\mathbb{C}(V(u), V(\tilde{u})) = \delta(u, \tilde{u})\ \sigma_v \tag{5}$$
$$\mathbb{C}(Z(u), V(\tilde{u})) = 0 \tag{6}$$
$$\mathbb{C}(Z(u), R) = 0. \tag{7}$$

### A. Learning/Inference

Given a data set $\mathcal{D} = \{(u^{[1]}, y^{[1]}), \cdots, (u^{[s]}, y^{[s]})\}$ of input-output pairs, our goal is to make inferences about the value of $X(q)$ for arbitrary query points $q \in \mathcal{R}^p$. This can be accomplished using the standard equations for conditional Gaussian distributions. The expected value of $X(q)$ given the available training data is

$$\mathbb{E}[X(q) \mid \mathcal{D}] = \mathbb{E}[X(q)] + k\,(y - \mathbb{E}[Y]) \tag{8}$$

where

$$\mathbb{E}[X(q)] = f(q)\mu \tag{9}$$
$$\mathbb{E}[Y(u^{[i]})] = g(u)f(u^{[i]})\mu \tag{10}$$

$$Y = \begin{pmatrix} Y(u^{[1]}) \\ \vdots \\ Y(u^{[s]}) \end{pmatrix}, \qquad y = \begin{pmatrix} y^{[1]} \\ \vdots \\ y^{[s]} \end{pmatrix} \tag{11}$$

Next,

$$k = \sigma_{qd}\sigma_{dd}^{-1} \tag{12}$$

where

$$\sigma_{dd} = \begin{pmatrix} (\sigma_{dd})_{11} & (\sigma_{dd})_{12} & \cdots & (\sigma_{dd})_{1s} \\ (\sigma_{dd})_{21} & (\sigma_{dd})_{22} & \cdots & (\sigma_{dd})_{2s} \\ \vdots & \vdots & \vdots & \vdots \\ (\sigma_{dd})_{s1} & (\sigma_{dd})_{s2} & \cdots & (\sigma_{dd})_{ss} \end{pmatrix} \tag{13}$$

with

$$\begin{aligned}(\sigma_{dd})_{ij} &= \mathbb{C}[Y(u^{[i]}), Y(u^{[j]})] \\ &= g(u)\mathbb{C}[X(u^{[i]}), X(u^{[j]})]g(u)^T + \mathbb{C}[V(u^{[i]}), V(u^{[j]})]\end{aligned} \tag{14}$$

and

$$\mathbb{C}[X(u^{[i]}), X(u^{[j]})] = f(u^{[i]})\sigma_w f(u^{[j]})^T + \kappa(u^{[i]}, u^{[j]}) \tag{15}$$

$\sigma_{qd}$ be defined as follows

$$\sigma_{qd} = \begin{pmatrix} (\sigma_{qd})_1 & (\sigma_{qd})_2 & \cdots & (\sigma_{qd})_s \end{pmatrix} \tag{16}$$

with

$$(\sigma_{qd})_j = \mathbb{C}[X(q), Y(u^{[j]})] \tag{17}$$
$$= \mathbb{C}[X(q), g(u^{[j]})X(u^{[j]}) + V(u^{[j]})]$$
$$= \mathbb{C}[X(q), X(u^{[j]})]g(u^{[j]})^T \tag{18}$$
$$= \Big(f(q)\sigma_w f(u^{[j]})^T + \kappa(q, u^{[j]})\Big)g(u^{[j]})^T \tag{19}$$

The uncertainty about the model's prediction (its variance matrix) is given by the following formula

$$\mathbb{V}[X(q) \mid \mathcal{D}] = \sigma_{qq} - k\sigma_{dd}k^T \tag{20}$$

These equations provide the optimal way to combine the prior parametric knowledge about the system, the non-parametric knowledge of structural deviations from the parametric model, and the available training data.

Since we assume the parametric model is just a rough approximation to the actual robot dynamics we are not particularly interested on making inferences about its parameters. Instead we want to use our knowledge and uncertainty about the parameters as a component to make better predictions. Thus we marginalize over the posterior distribution of $W$ given the training data. Note however that our beliefs about $W$, both the posterior mean and posterior variance have an effect in the inference process. Automatically, as we get more data we may become more certain about the value of $W$, thus, changing our inferences as we marginalize over $W$. This becomes particularly important in regions with sparse training data.

### B. Relationship to Gaussian Processes(GP)

A standard Gaussian process is typically defined as a collection of unidimensional Gaussian random variables $X(u)$ with zero mean and a covariance structure defined by a kernel function, i.e.,

$$\mathbb{C}(X(u), X(\tilde{u})) = \kappa(u, \tilde{u}) \tag{21}$$

This can be seen as a special type of SGP in which the parameter $W$ is visible and known to be zero ($\mathbb{E}[W] = 0$) with no uncertainty ($\mathbb{V}[W] = 0$), and $X$ is also directly visible, i.e., $Y(u) = X(u)$. In the SGP notation this corresponds to $g(u) = I_{n_y}$, the $n_y$-dimensional identity matrix and $\mathbb{V}[V(u)] = 0$ for all $u \in \mathcal{I}$.

### C. Relationship to Parametric Regression(PR)

In a standard parametric regression model $X$ is a unidimensional variable, $n_x = 1$. The goal is to predict $X$ as a linear combination of feature variables, i.e.,

$$X(u) = f(u)w \tag{22}$$

and there is additive i.i.d. Gaussian noise $V(u)$ to the observations, i.e.,

$$Y(u) = X(u) + V(u), \tag{23}$$

where $\mathbb{V}[V(u), V(\tilde{u})] = \delta(u, \tilde{u})\sigma_v$. The parameter $w$ is typically chosen to minimize the squared error using a dataset of $f(u), Y(u)$ values,

$$\hat{w} = \operatorname{argmin} \sum_i \|Y(u^{[i]}) - f(u^{[i]})w\|^2 \tag{24}$$

The solution takes the following form

$$\hat{w} = (\sum_i f(u^{[i]})^T f(u^{[i]}) + \sigma_v \mathbf{I})^{-1} (\sum_i f(u^{[i]})^T y_i). \tag{25}$$

Once $\hat{w}$ is found, $X$ can be estimated for new query points, i.e.,

$$\hat{X}(q) = f(q)\hat{w} \tag{26}$$

This can be seen as a special type of SGP in which the term $Z$ is a constant process, $\mathbb{E}[Z] = 0$, $\mathbb{V}[Z] = 0$, and both $\sigma_w$ and the observation function $g(u)$ are identity matrices.

### D. Hyper Parameter Selection

As in GPs, the hyper-parameters of an SGP ( $\sigma_w$, $\sigma_v$ and the parameters in kernel function $\kappa$) can be set using cross-validation methods or maximum likelihood estimation. For the popular squared exponential kernel, the kernel parameters include $\sigma_z$ and $\ell_i, i = 1, 2, \ldots, p$. For problem with larger input dimension $p$, the space of hyper-parameters can be too large for grid search in cross-validation. Therefore, we take the maximum likelihood approach to find an (locally) optimal set of hyper-parameters maximizing the likelihood of the training data,

$$\begin{aligned}
&\log p(Y|X, f(\cdot), g(\cdot)) \\
&= -\frac{1}{2}\left(g(u)f(u)\mathbb{E}[W] - Y(u)\right)^T \left(\sigma_{qd}\sigma_{dd}^{-1}\sigma_{qd}^T + \sigma_v\right)^{-1} \\
&\left(g(u)f(u)\mathbb{E}[W] - Y(u)\right) - \frac{1}{2}\log\det(\sigma_{qd}\sigma_{dd}^{-1}\sigma_{qd}^T + \sigma_v) \\
&- \frac{n}{2}\log 2\pi.
\end{aligned}$$

In practice, we put uninformative zero-mean prior over $W$, $\mu = \mathbf{0}$, which simplifies the likelihood to

$$\begin{aligned}
&\log p(Y|X, f(\cdot), g(\cdot)) \\
&= -\frac{1}{2}Y(u)^T(\sigma_{qd}\sigma_{dd}^{-1}\sigma_{qd}^T + \sigma_v)^{-1}Y(u) \\
&- \frac{1}{2}\log\det(\sigma_{qd}\sigma_{dd}^{-1}\sigma_{qd}^T + \sigma_v) - \frac{n}{2}\log 2\pi.
\end{aligned}$$

Then, we initialize the parameter with some heuristic guess:

- $\sigma_v$: $1/100$ of sample variance of $y^{[i]}$'s.
- $\sigma_z$: sample variance of $y^{[i]}$'s.
- $\ell_i$: sample variance of $u^{[i]}$'s.
- $\sigma_w$: some large number denoting the uninformative prior over $W$.

Finally, we optimize the likelihood using conjugate gradient ascent as in [11] to find a local maximum.

## IV. Experiments

### A. Toy Problem

A popular and reasonable way to combine parametric and non-parametric approaches uses a two-step approach: first the parametric model is fit to the available data and then a non-parametric model is fit to the errors made by the parametric model. Fig. 2 shows that SGPs combine parametric and non-parametric prior knowledge in a smarter way. The curve shows the result of the two-step approach (PRGP) and the SGP approach on the simple pendulum task. Both approaches perform well in the region densely populated with training data. However, in the region with sparse training data SGP achieves better prediction than PRGP. The reason is that SGP jointly fits the global parameter $W$ and Gaussian process $Z$ at the same time, thus allowing them to interact and complement each other: The non-parametric part of the SGP "explains away" the irregular bump, and allows for the parametric model parameter $W$ to be less affected by it. The result is a more accurate global model. In the two-phase approach, the non-parametric model (GP) cannot affect the predictions made by the parametric model, resulting in worse predictions around the regions with sparse training data.

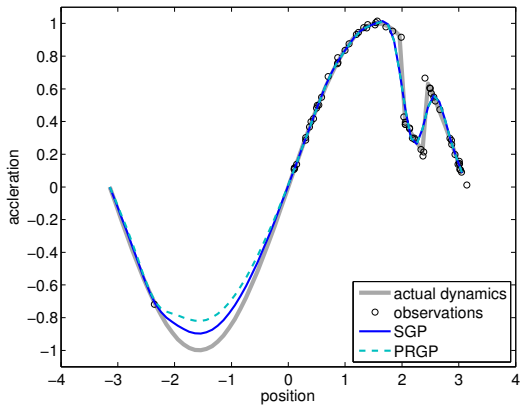



Fig. 2. Comparing the two-stage method (PRGP) model and SGP on learning acceleration of a pendulum. Note the predictions from SGP are closer to the desired dynamics in the region with sparse data (position $[-2, 0]$).

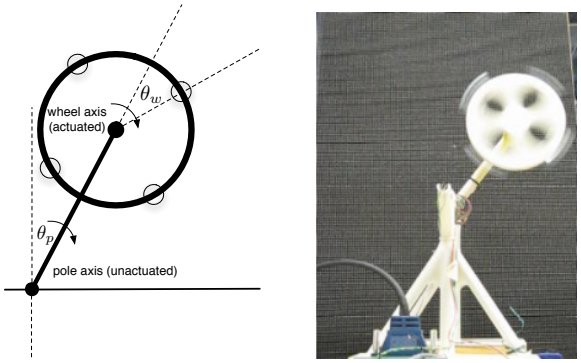
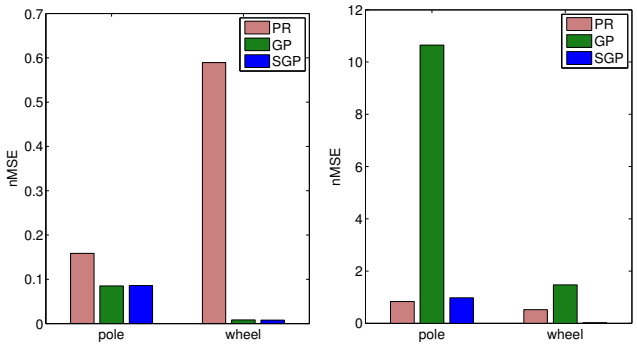



Fig. 3. The reaction wheel and our physical implementation.

### B. Simulated Reaction Wheel

We compared SGP with PR and GP on a simulated reaction wheel problem. A reaction wheel is a pendulum with a flywheel attached to the end of its pole (see Figure 3). It has two parallel axes: the pole axis and the wheel axis. The pole axis is un-actuated while the wheel axis has an electric motor. Both axes are equipped with sensors which measure the angles $\theta_p$ and $\theta_w$. The angular velocities $\dot{\theta}_p, \dot{\theta}_w$ are obtained via finite differences. The state space of the system is given by $x = [\theta_p, \theta_w, \dot{\theta}_p, \dot{\theta}_w]^T$. In this case the




(a) pole down (b) pole up

Fig. 4. System identification error for **simulated** reaction wheel in pole-down and pole-up testing conditions.

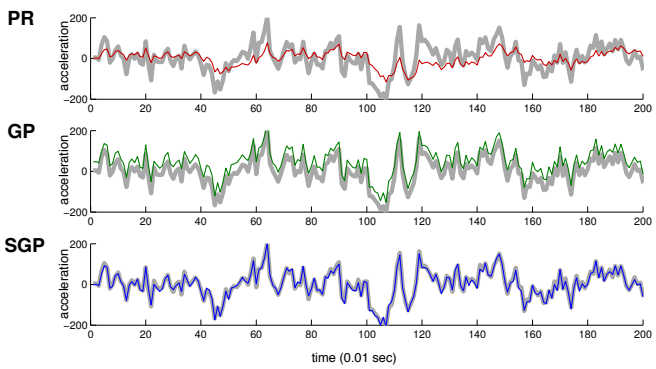



Fig. 5. The predicted (color-narrowband) versus actual (gray-wideband) accelerations of the wheel axis in the pole-up condition of **simulated** reaction wheel.

Newtonian equation of motion is as follows

$$\frac{d}{dt}\begin{bmatrix}\theta_p\\ \theta_w\\ \dot{\theta}_p\\ \dot{\theta}_w\end{bmatrix} = \begin{bmatrix}0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ w_1 & w_2 & w_3 & w_4\\ w_5 & w_6 & w_7 & w_8\end{bmatrix}\begin{bmatrix}\sin\theta_p\\ \theta_w\\ \dot{\theta}_p\\ \dot{\theta}_w\end{bmatrix} + \begin{bmatrix}0\\ 0\\ w_9\\ w_{10}\end{bmatrix} m, \tag{27}$$

where $w \in \mathcal{R}^{10}$ is the model parameter and $m \in \mathcal{R}$ is the control (voltage to the motor driver) applied to the flywheel. Note the dynamics are a linear function of the parameters and therefore we can apply the SGP approach to model the forward dynamics. The task was to predict the acceleration of both axes, $y = [\frac{d}{dt}\dot{\theta}_p, \frac{d}{dt}\dot{\theta}_w]$, given the current state $x$ and control signal $m$. The performance criterion was normalized mean square error(nMSE). The normalization was with respect to the variance of the ground truth of target $y$. The goal of this experiment was to evaluate how well PR, GP and SGP work when the parametric model provides only a "rough" approximation to the real system dynamics. To do so, we obtained training data from the full Newtonian model but fitted a parametric model with no viscous friction. Thus the training data came from a model with 10 parameters but the parametric model only had 6 parameters,i.e., $W \in \mathcal{R}^6$. The feature matrix of the parametric model looked as follows:

$$f(x,m) = \begin{bmatrix}\sin(\theta_p) & \theta_w & m & 0 & 0 & 0\\ 0 & 0 & 0 & \sin(\theta_p) & \theta_w & m\end{bmatrix} \tag{28}$$

We used a squared exponential kernel with independent length-scale hyper-parameter in each dimension. Since the sensors directly observe the angles of the axes, we have the $g(x)$ be an identity matrix. All the hyper-parameters were optimized by maximum marginal likelihood.

The training traces were obtained by controlling the reaction wheel with a mix of sinusoids signal of random frequencies for 1000 time steps at 100Hz. Due to the fact that this is an under-actuated problem, the distribution of $\theta_p$ in the training data was heavily biased toward the pole-down position. For testing, two different groups of data were collected:

- **pole-down:** traces collected in the same manner as training data.
- **pole-up:** traces collected while the pole was balanced by an LQR controller at the upward position. Note since there was noise in the simulated dynamics, the reaction wheel randomly wondered around the balancing point rather than being static.

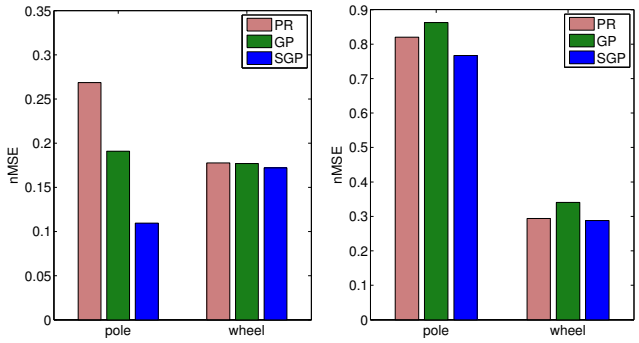
The purpose of the two test sets was to simulate queries in regions of dense and sparse training data. The simulation was repeated 5 times and the average was reported. Figure 4 shows the performance of parametric regression (PR), Gaussian Processes (GP) and Semi-parametric Gaussian Processes (SGP). Note in the pole down position, which has a high density of training data, GP and SGP outperformed PR. This shows that in this region better predictions could be made by interpolation between local training data. However in the pole up position, where there were sparse training data, the parametric model did much better and the non-parametric model (GP) did quite poorly. SGP learned to rely on the parametric model in this case for the pole acceleration, and on training data for the fly-wheel acceleration thus outperforming the GP and the PR approaches.

In some cases, SGP performed surprisingly better than both GP and PR. For example, for the wheel axis in the pole-up condition, SGP achieved almost perfect prediction while PR and GP performed much worse (Fig. 4(b)). We plot the traces for this specific condition in Fig. 5. Note the high accuracy of the SGP method when compared to he other two. The predicted acceleration of the other two method is either too small in magnitude (PR) or biased (GP).

### C. Physical Reaction Wheel

The data collection procedure was the same as in the simulated reaction wheel problem. Except that for the parametric model we used the full 10 parameter Newtonian model in (27). The hope was that the non-parametric model could capture the dynamics not modeled by the best known Newtonian model for our problem.

Figure 6 plots the performance the three methods. Overall, the trend resembles what we observed in the simulation. In the pole-down testing condition where there were dense




(a) pole down (b) pole up

Fig. 6. System identification error for **physical** reaction wheel in pole-down and pole-up testing conditions.

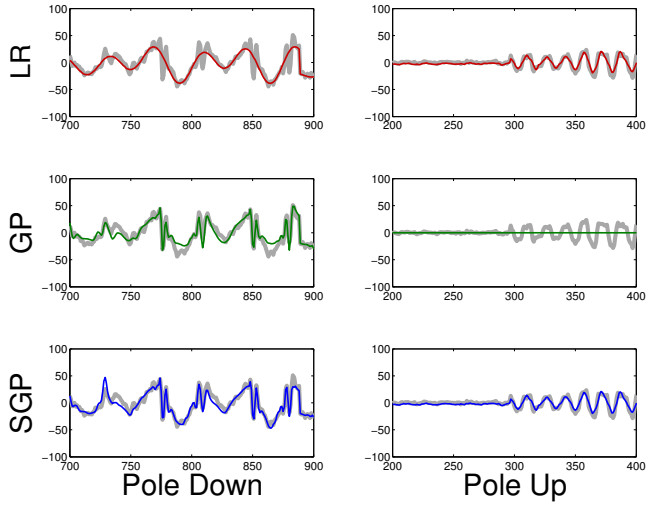



Fig. 7. The predicted (color-narrowband) versus actual (gray-wideband) accelerations of the pole axis of **physical** reaction wheel. We only plotted a small fraction of the total data to reveal the detail in closeup.

training data, SGP and GP outperformed PR. In the pole-up condition with sparse training data PR and SGP beat GP. In all cases SGP was the leading method.

Figure 7 shows some traces of pole-axis acceleration predicted by various models. First we examine the pole-down condition (left column): parametric model, PR, predicted very smooth curve but missed those small spikes. Some of these spikes might be state dependent noise that could be captured by the Gaussian Process in GP and SGP model. Indeed, both model captured some of the spikes. But overall, SGP predicted the data better.

As for the pole-up condition (right column), since the state space was quite distant from the training data collected in the pole-down condition, the non-parametric component contributes little to the final prediction. In this particular case, GP predicted basically a flat-line (the zero-mean prior of GP), which happened to be a good guess for pole-up balancing data. The prediction of SGP was dominated by its parametric model as the prediction followed PR closely.

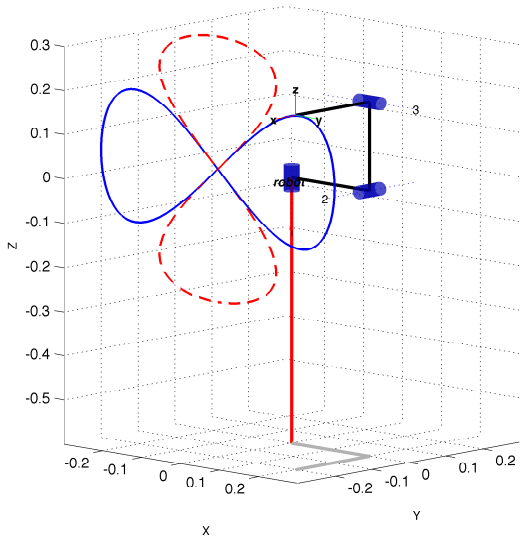



Fig. 8. The simulated 3-link robot used in the experiment. The figure-eight horizontal (blue-solid) and vertical (red-dash) trajectories are also plotted.

### D. Multi-Link Robot Arm

Next we experimented on learning the inverse dynamics model of a multi-link robot arm as a demonstration of the applicability of SGP to arbitrary tree-linked robot structure. The configuration of our simulated 3-link arm is shown in Fig. 8. The goal was to learn the dynamic model on the horizontal (blue-solid) figure-eight curve and then tested on vertical (red-dashed) one. In this way, we could compare how different algorithms fitted the training data and generalized to unseen curve. Although we experimented with a 3-link arm for simplicity, SGP scales to as many degree-of-freedom open-chain robot as GP scales.

For data collection, we drove the robot along these two figure-eight curves with a simple proportional controller. Meanwhile, we recorded the trace $\{\theta_i, \dot{\theta}_i, \ddot{\theta}_i, \tau_i\}_{i=1:t}$, where $\theta_i, \dot{\theta}_i, \ddot{\theta}_i, \tau_i \in \mathbb{R}^3$. In the process, artificial state dependent noise was injected by increasing the friction when joint 3 (the last joint that connects the end-effector) approached its joint limits. Such noise in real-life may be caused by cables between the links.

An inverse dynamics model predicts the required joint torque $\tau$ given the current joint angle $\theta$, joint angular velocity $\dot{\theta}$, and the desired angular acceleration $\ddot{\theta}$. The inverse dynamics of an articulated open-chain tree structured is the following:

$$\tau = -M(\theta)\ddot{\theta} - C(\theta, \dot{\theta})\dot{\theta} + \tau^{\nu} + \tau^{c} + \tau^{g}, \qquad (29)$$

where $M$ is the inertia matrix, $C$ is Coriolis and centripetal forces matrix, $\tau^{v} = \dot{\theta}\eta_{\nu}$ is the viscosity, $\tau^{c} = \mathbf{sign}[\dot{\theta}]\eta_{c}$ is the Coulomb friction and $\tau^{g}$ is the gravitational force.

To apply the regression algorithms, (29) can be written into in linear-in-parameter form [8], [1], [7],

$$\tau = f(\theta, \dot{\theta}, \ddot{\theta})w, \qquad (30)$$

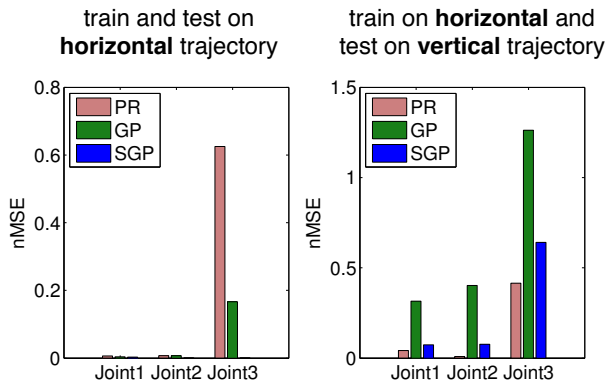



Fig. 9. System identification error for simulated 3-link arm. Models trained using horizontal figure-eight trajectory are tested on (left-panel) another similar horizontal figure-eight trajectory (right-panel) the vertical figure-eight trajectory.

where $w \in \mathbb{R}^3$ is the dynamics parameters to be identified, including mass, center-of-mass, inertia tensor for each link and viscosity/Coulomb parameters $(\eta_\nu, \eta_c)$ for each joint. Fig. 9 compares the prediction error of the three algorithms. In the left panel, all three model performed very well for the first two joints when tested on the data in the neighborhood of training data. However, for joint 3, due to the additional un-modeled noise, LR failed badly and GP also performed worse than SGP. Further inspection suggested that GP did not fit the discontinuities caused by friction very well. The optimal length-scale parameter $l_i$ selected by maximum likelihood was too large to fit the discontinuities. In comparison, the parametric model in SGP had absorbed these discontinuities which resulted in a better fit. Similar discontinuity fitting problem was also discovered in [4]. On the right panel, PR generalized to new vertical trajectory quite well, closely followed by SGP. GP failed badly in this case as the testing data were quite distant from the training data.

## V. Discussion

In all the experiments, we found that SGP performs comparable or better than the standard parametric (PR) and non-parametric (GP) approaches.

In the real world experiment, the non-parametric model (GP) performed substantially better than the parametric model (PR) when dense training data was available. This suggests that there were state dependent dynamics that the parametric Newtonian model could not pick up. SGP performed best in data rich and data poor conditions but the difference was not as large in the real world experiment as in the computer simulation.

## VI. Conclusions

We presented a principled approach, named Semi-Parametric Gaussian Processes, to combine parametric and non-parametric models in robot dynamics system identification problems. The method applies to articulated robots of arbitrary complexity as long as they can be expressed in tree-like structure. SGP flexibly fuses the prior knowledge available in Newtonian parametric models and the knowledge about local structure provided by non-parametric models. We showed computer simulations and physical implementation results that suggested SGPs capture the desirable properties of parametric and non-parametric models, both in terms of the accuracy of their predictions and the knowledge about their own uncertainty. One property that distinguishes SGPs from pure non-parametric approaches is the ability to make rough but useful generalizations over unexplored state regions. This may prove particularly important for solving underactuated robotics problems. In these problems it is difficult to develop a controller that explores, and thus provides training data about target regions of the state space. Without such training data it is difficult to train non-parametric models that could be used to develop effective controllers. The parametric component of SGP provides the ability to generalize to unseen regions of the state space. This knowledge while coarse, can prove useful to develop controllers that gather more data to train the fine-grain non-parametric component of the SGP.

## References


[1] C. An, C. Atkeson, and J. Hollerbach. Estimation of inertial parameters of rigid body links of manipulators. In *Decision and Control, 1985 24th IEEE Conference on*, volume 24, pages 990–995. IEEE, 1985.

[2] B. Blight and L. Ott. A bayesian approach to model inadequacy for polynomial regression. *Biometrika*, 62(1):79–88, 1975.

[3] K. Chai. Multi-task learning with gaussian processes. 2010.

[4] K. M. A. Chai. *Multi-task Learning with Gaussian Processes*. PhD thesis, Institute for Adaptive and Neural Computation School of Informatics University of Edinburgh, 2010.

[5] M. Deisenroth, J. Peters, and C. Rasmussen. Approximate dynamic programming with gaussian processes. In *American Control Conference, 2008*, pages 4480–4485. Ieee, 2008.

[6] M. Deisenroth and C. Rasmussen. Pilco: A model-based and data-efficient approach to policy search. In *Proceedings of the Twenty-Eighth International Conference on Machine Learning (ICML)*, 2011.

[7] M. Gautier and W. Khalil. A direct determination of minimum inertial parameters of robots. In *ICRA*, pages 1682–1687. IEEE, 1988.

[8] P. Khosla. Estimation of robot dynamics parameters: Theory and application. 1987.

[9] J. Ko and D. J. Klein. Gaussian processes and reinforcement learning for identification and control of an autonomous blimp. In *in IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2007.

[10] D. Nguyen-Tuong and J. Peters. Local gaussian process regression for real-time model-based robot control. In *International Conference on Intelligent Robots and Systems*, pages 380–385. IEEE, 2008.

[11] C. Rasmussen and H. Nickisch. Gaussian processes for machine learning (gpml) toolbox. *The Journal of Machine Learning Research*, 9999:3011–3015, 2010.

[12] A. Rottmann and W. Burgard. Learning non-stationary system dynamics online using gaussian processes. In *Pattern Recognition*, volume 6376 of *Lecture Notes in Computer Science*, pages 192–201. Springer Berlin / Heidelberg, 2010.

[13] D. Ruppert, M. Wand, and R. Carroll. *Semiparametric regression*, volume 12. Cambridge Univ Pr, 2003.

[14] S. Schaal, C. G. Atkeson, and S. Vijayakumar. Real-time robot learning with locally weighted statistical learning. In *ICRA*, pages 288–293. IEEE Press, 2000.

[15] J. Sjoberg, H. Hjalmarsson, and L. Ljung. Neural networks in system identification. 1994.

[16] E. Todorov and W. Li. A generalized iterative lqg method for locally-optimal feedback control of constrained nonlinear stochastic systems. In *American Control Conference, 2005*, pages 300–306. IEEE, 2005.

[17] S. Vijayakumar, A. D'souza, and S. Schaal. Incremental online learning in high dimensions. *Neural Computation*, 17(12):2602–2634, 2005.