Control of Musculoskeletal Systems using Learned Dynamics Models

Dieter Büchler[1,2], Roberto Calandra[3], Bernhard Schölkopf[1], Jan Peters[1,2]

***Abstract*—Controlling musculoskeletal systems, especially robots actuated by pneumatic artificial muscles, is a challenging task due to nonlinearities, hysteresis effects, massive actuator delay and unobservable dependencies such as temperature. Despite such difficulties, muscular systems offer many beneficial properties to achieve human-comparable performance in uncertain and fast-changing tasks. For example, muscles are backdrivable and provide variable stiffness while offering high forces to reach high accelerations. In addition, the *embodied intelligence* deriving from the compliance might reduce the control demands for specific tasks. In this paper, we address the problem of how to accurately control musculoskeletal robots. To address this issue, we propose to learn probabilistic forward dynamics models using Gaussian processes and, subsequently, to employ these models for control. However, Gaussian processes dynamics models cannot be set-up for our musculoskeletal robot as for traditional motor-driven robots because of unclear state composition etc. We hence empirically study and discuss in detail how to tune these approaches to complex musculoskeletal robots and their specific challenges. Moreover, we show that our model can be used to accurately control an antagonistic pair of pneumatic artificial muscles for a trajectory tracking task while considering only one-step-ahead predictions of the forward model and incorporating model uncertainty.**

***Index Terms*—Model Learning for Control; Biologically-Inspired Robots; Hydraulic/Pneumatic Actuators**

I. Introduction

MANY dynamic activities that appear straightforward for humans, such as walking, grasping or ball games, are still fundamental challenges in robotics. Uncertainty, the requirement of fast reactions, and dynamic movements with high accelerations – without damaging the system and environment – still pose big hurdles. Despite the existence of learning algorithms that outperform humans in non-robotics tasks [1], the transfer of super-human performance to robots in dynamic tasks has not been shown yet.

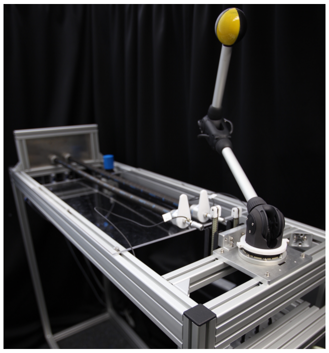


Fig. 1: 4-DoF robot arm actuated by eight PAMs. 700 g moving masses and PAMs with max. forces of 1200 N lead to angular accelerations of up to 28k deg/s^2.

Using robots actuated by muscle-like actuators, also called muscular robots, can be a way to achieve human-level performance in robotics. In this paper, we try to leverage antagonistic pairs of pneumatic artificial muscles (PAMs) instead of traditional motors. PAMs are the nearest replica of skeletal muscles available as robotics hardware and exhibit many of their desired properties [2]. First, PAMs are backdrivable, thus, damages due to low velocity impacts with humans, external objects and the robot itself are reduced (although not completely prevented). On the other hand, changing co-contraction levels adjust the compliance in the antagonistic pair, offering flexibility if the task requires it. Second, high accelerations, provided by PAMs, enable the robot to reach desired states in less time and allow for fast flick-movements due to energy storage and release as observed in fast human arm movements. Third, learning dynamic tasks, e.g. table tennis, is more feasible with antagonistic actuation as damage due to exploration at higher velocities can be minimized [3]. Fourth, it has been shown that the demands on the control algorithm are reduced for tasks where contact with external objects is required, e.g. opening a door [4]. Also muscular actuation assures gait stability in spite of the presence of unmeasured disturbances [5], a desirable property that might cope with uncertainties of dynamic tasks. These insights illustrate what is known as embodied intelligence and may – in combination with a learning control approaches – pave the way to human-level movements.

Yet, muscular robots are not widely used. Many of the issues with pneumatic muscles ultimately derive from the lack of good dynamics models that generally describe the relationship between the action $\mathbf{u}$ taken in state $\mathbf{s}$ and the successor state $\mathbf{s}'$. Severely non-linear behavior, unobservable dependencies such as temperature, wear-and-tear effects and hysteresis [2] render PAM systems considerably challenging to model. Another reason that muscular robots are scarcely in use is overactuation. An infinite set of air pressure combinations in an antagonistic PAM pair lead to the same joint angle but with different compliance levels. Compliance $c = \delta q / \delta F_{\text{ext}}$ or its inverse the stiffness k represents the external force F_{ext} applied to the joint required to cause a change of δq in the joint angle. Overactuation is critical for two reasons. First, overactuation principally rules out acquiring inverse models $\mathbf{u} = f(\mathbf{s}, \mathbf{s}')$ of musculoskeletal systems by traditional regression. Inverse dy-

Manuscript received: February, 24, 2018; Revised May, 14, 2018; Accepted June, 4, 2018.

This paper was recommended for publication by Editor Yu Sun upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the Max Planck Institute for Intelligent Systems Tübingen, Germany.

[1]Empirical Inference Department, Max Planck Institute for Intelligent Systems Tübingen, Germany

[2]Department of Computer Science, Technische Universität Darmstadt, Germany

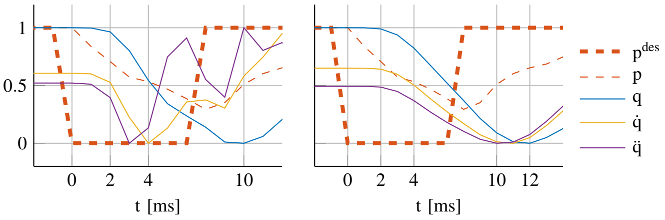
[3]University of California, Berkeley, United States
dbuechler@tue.mpg.de, mail@jan-peters.net, bs@tue.mpg.de, roberto.calandra@berkeley.edu

Digital Object Identifier (DOI): see top of this page.

namics models are only unique for traditionally actuated robots and have to be approximated locally for over- and under-actuated systems [6]. The second reason is that overactuation expands the state-action space, thus learning such dynamics models requires more data or additional assumptions about the underlying mechanisms. Using forward models $\mathbf{s}' = f(\mathbf{s}, \mathbf{u})$, however, does not fully solve the issue despite describing the true causal relationship and hence always existing. In case the controller uses the forward model to decide on the actions by minimizing the control error, overactuated systems offer multiple equally optimal solutions. Without any additional constraints, such a control framework results in optimization over loss functions with an additional severe source of non-convexity.

The straightforward constraint is minimal energy, hence always choosing the minimal pressures that lead to the minimal control error. In the well-known linear quadratic regulator setting this is indicated by the $\mathbf{u}^T\mathbf{R}\mathbf{u}$ term. Another possibility is to change stiffness or compliance according to the task, e.g., for grasping of delicate objects. The main idea of our approach is to pose a constraint that facilitates the use of a learned non-parametric probabilistic model. In the face of the above-mentioned difficulties of retrieving good models, learning a flexible model purely from data seems to be a promising approach. Recent successes in Gaussian process (GP) dynamics models [7]–[9] are especially encouraging to follow this line of research. However, the application of nonparametric models in an online setting requires – among other considerations – the selection of a small set of informative training data points from a highly time-correlated data stream. The GP return meaningful predictions only in the vicinity of the training data. Hence, it is possible to deviate into unknown state-action regions, especially with time-variant systems like pneumatic muscles. In order to enable nonparametric model learning for antagonistically actuated systems despite higher dimensionality due to overactuation, we utilize the uncertainty of the GP model as an additional constraint.

The contribution of this paper is twofold. First, we discuss and empirically evaluate the important aspects of learning GP forward dynamics models for muscular systems as they are substantially different to traditional robots. Second, we introduce a novel formulation that by incorporating the uncertainty of the GP dynamics model into our control framework, allows to controls the system towards the area of the state-space that are known from the training data. This capability is made possible by the use of a muscular systems and by exploiting its overactuation property, i.e., by choosing the set of muscle pressures from the infinite set that minimizes the control error. With this novel control scheme, we can ensure that the model-based controller remains in the vicinity of the training data of the GP where we have reasonable prediction.



(a) (b)

Fig. 2: System responses to a step in control signal at $t = 0$. All values have been normalized to be in $[0, 1]$. It can bee seen that $\ddot{q}$ reaches its minimum faster than $\dot{q}$ and q. (a) Unfiltered sensor values show a faster response as any filtering adds delay. The first substantial change occurs at $t = 2$. (b) Filtered values reach their respective minimum slower than in (a). The first substantial changes occur at $t = 3$, hence 1 time step later than the unfiltered signals. Generally, higher order time-derivatives react faster. However, filtering inhibits this effect.

II. Pneumatic Muscle Robot

We use the PAM-actuated robot arm from [3] which is shown in Fig. 1. Its key features are a) its lightweight structure with only 700 g moving masses, b) powerful PAMs that can lift up to 1.2 kN each and generate angular accelerations of up to 28k deg$/s^2$ and c) its design that aims to reduce difficulties for control, e.g., minimal bending of cables etc. This four DoF robot arm has eight PAMs in total with each DoF actuated by an antagonistic pair. Each PAM is equipped with an air pressure sensor and each joint angle is read by an incremental encoder. The pressure in each muscle is changed via Festo proportional valves that regulate the airflow according to the input voltage.

All signals are fed into a National Instruments FPGA PCIe 7842R card. The inputs and outputs of this card can be accessed via a C/C++ API from the main C++ code. On the FPGA, the encoder values are translated into joint angles $\mathbf{q}$ and a pressure controller is implemented for each PAM. This inner control loop compares the desired pressures $\mathbf{p}^{\text{des}}$ sent from the C++ code with the current air pressures within the PAMs $\mathbf{p}$ and sets the voltage to the proportional valves $\mathbf{v}$ accordingly. With this disentangled setup, we can bound $\mathbf{p}^{\text{des}}$ within the FPGA to be maximally 3 bar thus practically ensuring that $\mathbf{p}$ never reaches the allowed pressure limit of 6 bar.

III. Model Learning & Control

Learning flexible and probabilistic forward dynamics models and using them for control is a promising way to achieve higher performance for muscular robots. Gaussian process regression is a non-parametric and probabilistic model that is often used to represent robot dynamics [6]. Here, we address how to adapt such a GP model for muscular systems and discuss the additional difficulties that arise compared to GP dynamics modeling for traditionally actuated robots. We subsequently show how to use uncertainty estimates of the GP model during control to exploit the overactuation inherent to musculoskeletal systems in order to generate trajectories while staying close to the training data.

A. Traditional Model Learning with Gaussian Processes

Muscular robots are substantially different from motor-driven systems. After giving some background on forward dynamics modeling of motor-driven robots with GPs, we briefly describe the Hill-muscle model to illustrate our adaption to the GP model setup.

1) Rigid Body Dynamics: Generally, the dynamics of a robot representing the relationship between joint angles $\mathbf{q}$ and its derivatives $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ to the torque vector τ are modeled by differential equations derived by applying Newton's second law and assuming rigid links

$$\mathrm{M}(\mathbf{q})\ddot{\mathbf{q}}+\mathrm{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}+\mathrm{g}(\mathbf{q})=\tau\,, \tag{1}$$

with M being the joint space inertia matrix, C representing the Coriolis and centrifugal forces and g gravitation. For traditional motor-based robots the torque τ is proportional to the input current. Hence the relationships $(\mathbf{q},\dot{\mathbf{q}},\tau)\rightarrow\ddot{\mathbf{q}}$ and $(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}})\rightarrow\tau$ represent forward and inverse dynamics respectively. The state and actions here are clearly defined to be $\mathbf{s}=[\mathbf{q},\dot{\mathbf{q}}]$ and $\mathbf{u}=[\tau]$. The joint angle accelerations $\ddot{\mathbf{q}}$ can be used as successor state $\mathbf{s}'$ because the actions τ directly influence $\ddot{\mathbf{q}}$ and the state entities can be recovered by differentiation. For muscular robots the state composition is not obvious. The torque $\tau_i=\mathbf{F}_i\times\mathbf{r}_i$ depends on the combined force of all muscles acting on joint i, $\mathbf{F}_i$, as well as the geometry of the joint, specifically the moment arm $\mathbf{r}_i$. For an antagonistic muscle pair with a radial joint, the torque reduces to

$$\tau_i=(\mathrm{F}_i^{(a)}-\mathrm{F}_i^{(b)})r_i\,, \tag{2}$$

given the forces of both muscles $\mathrm{F}_i^{(a)}$ and $\mathrm{F}_i^{(b)}$ and the joint radius r_i that is independent of the joint angle.

2) Analytical Muscle Model: Many analytical force models for skeletal muscles exist. One of the most widely used is the Hill muscle model, see [10] for a detailed description. It has been shown that the Hill model reflects the properties of PAMs to some extent [11]. The total muscle force

$$\mathrm{F}_M=\mathrm{F}_{CE}+\mathrm{F}_{PEE}=\mathrm{F}_{SEE}\,, \tag{3}$$

derived from this model is based on an active contractile element F_{CE} that depends on the activation a and passive parallel and serial elastic elements F_{PEE} and F_{SEE} that both change with the muscle length l_M. The quantity F_M would enter Eq. (2) as one of the forces $F_i^{(a)}$ or $F_i^{(b)}$. The active part of the total force

$$\mathrm{F}_{CE}=a\mathrm{F}_{max}\mathrm{f}_L(\mathrm{l}_{CE})\mathrm{f}_V(\mathrm{v}_{CE})\,, \tag{4}$$

depends not only on the activation a but also on the force-length f_L and the force-velocity f_V relationship and is parameterized by the maximum isometric force F_{max}. Typically, f_L is bell-shaped whereas a sigmoid-like function constitutes f_V. Eq. (4) can then be used to form the dynamics of the muscle length

$$\frac{\partial \mathrm{l}_{CE}}{\partial \mathrm{t}}=\mathrm{f}_V^{-1}\left(\frac{\mathrm{F}_{SEE}-\mathrm{F}_{PEE}}{a\mathrm{F}_{max}\mathrm{f}_L(\mathrm{l}_{CE})}\right)\,, \tag{5}$$

that takes the activation a and l_{CE} as parameters, resulting from the interaction with τ in Eq. (1). Often the activation is modeled as a non-instantaneous process based on a neural excitation signal u

$$\frac{\partial a}{\partial \mathrm{t}}=\mathrm{c}_a(a-\mathrm{u})\,, \tag{6}$$

in which c_a is the constant activation and deactivation rate.

Notation: The actions $\mathbf{u}$ of the system from Section II correspond to the desired pressures $\mathbf{p}^{\mathrm{des}}$ and the state $\mathbf{s}$ can be composed of any combination of the sensed signals $\mathbf{v}$, $\mathbf{q}$ and $\mathbf{p}$. The activation a and the neural excitation signal u from the analytical Hill muscle model in Eq. (6) coincide with the current and desired pressure $\mathbf{p}$ and $\mathbf{p}^{\mathrm{des}}$.

3) Gaussian Processes Forward Dynamics Models: In this paper, we learn a probabilistic Gaussian process [12] forward dynamics model in discrete time

$$\mathbf{s}_{t+1}=\mathbf{f}(\mathbf{s}_t,\mathbf{u}_t)+\varepsilon\,, \tag{7}$$

with $\mathbf{s}\in\mathbb{R}^D$ being the state of dimensionality D, $\mathbf{u}\in\mathbb{R}^M$ the action of dimensionality M and $\varepsilon\sim\mathcal{N}(0,\Sigma_n)$ independent and identically distributed (i.i.d.) Gaussian measurement noise with a diagonal covariance noise matrix Σ_n. The state transfer function $\mathbf{f}$ is modeled by a Gaussian process with a squared exponential kernel and automatic relevance determination (ARD)

$$k(\mathbf{x}_a,\mathbf{x}_b)=\sigma_f^2\exp\left(-\frac{1}{2}(\mathbf{x}_a-\mathbf{x}_b)^T\Lambda^{-1}(\mathbf{x}_a-\mathbf{x}_b)\right)\,, \tag{8}$$

having signal variance σ_f^2 and squared lengthscales $\Lambda=\mathrm{diag}(l_1^2,\ldots,l_{D+M}^2)$. The dataset $\mathcal{D}=\{X,\mathbf{y}\}$ consists of a design matrix $X\in\mathbb{R}^{N\times(D+M)}$ with each row being the nth training input $\mathbf{x}_n^T=[\mathbf{s}_n,\mathbf{u}_n]^T$ and $\mathbf{y}\in\mathbb{R}^N$ the target values. Hence, one GP is established for each element of $\mathbf{s}'$. A GP can be seen as a distribution over functions and is queried using the conditional (posterior) probability

$$p(\mathbf{f}|\mathrm{X},\mathbf{y},\mathbf{x}_*)=\mathcal{N}(\mathbf{k}_*^T\alpha,k_{**}-\mathbf{k}_*^T\tilde{\mathrm{K}}\mathbf{k}_*)\,, \tag{9}$$

where $[\mathbf{k}_*]_n=k(\mathbf{x}_*,\mathbf{x}_n)$, $k_{**}=k(\mathbf{x}_*,\mathbf{x}_*)$, $\alpha=\tilde{\mathrm{K}}\mathbf{y}$, $\tilde{\mathrm{K}}=(\mathrm{K}+\sigma_n\mathrm{I})^{-1}$ and I being the identity matrix. The hyperparameters σ_f,σ_n and Λ are optimized by maximizing the marginal likelihood $p(\mathbf{y}|\mathbf{X})$. The GP provides a flexible model by assuming only smoothness of the underlying function f, depending on the choice of the covariance function. On the other hand, non-parametric methods require a training dataset to make predictions. The training time complexity rises cubically $\mathcal{O}(N^3)$ and prediction time complexity linearly $\mathcal{O}(N)$ with the number of training data points N [13], [14]. Many approximations schemes exist [15], [16] that are relevant to future work but are not used here as they rely on approximations.

B. Model Adaptations to Muscular Systems

Many properties of pneumatic muscular systems render modeling difficult. We mention some of these issues from which we subsequently derive adaptations to the GP forward dynamics model.

1) Modeling Issues: Modeling of PAMs is still a key problem for control [2]. The reason for this is that analytical models derived from physics, including the Hill muscle model from Eq. (4), do not sufficiently describe the properties of real PAMs. Unobserved influences such as volume change of the muscle while moving, tip effects and temperature as well as hysteresis and nonlinearities require the model to be extraordinarily flexible.

Another important issue with pneumatics is actuator delay. In every system a short time passes between applying control until a reaction is sensed. The time in between sums up all delays in the control loop. For instance, a magnetic field has to increase in a motor or – as in our case – valves have to open and air pressure has to rise within the PAM. For PAM-actuated robots, the sequence of actuation

$$\mathbf{p}^{\text{des}} \rightarrow \mathbf{v} \rightarrow \text{air flow} \rightarrow \mathbf{p} \rightarrow \tau \rightarrow \ddot{\mathbf{q}} \rightarrow \dot{\mathbf{q}} \rightarrow \mathbf{q}\,,$$

contains more sources of delay compared to motor-driven systems, e.g., mechanical opening and closing of the valve and the generation of air pressure in the muscles. This process is further delayed by the compressibility of air. Fig. 2 shows an experiment where the joint angles $\mathbf{q}$, velocities $\dot{\mathbf{q}}$ and accelerations $\ddot{\mathbf{q}}$ are recorded in response to a step in desired pressure signal $\mathbf{p}^{\text{des}}$ for unfiltered and filtered cases. Unfiltered entities respond faster to the excitation but inhere more noise, especially higher time-derivatives such as joint velocities $\dot{\mathbf{q}}$ and accelerations $\ddot{\mathbf{q}}$ because they need to be estimated from $\mathbf{q}$. Strong noise complicates modeling with GPs as only output noise is assumed.

2) State Composition: Actuator delay is a sign of unobserved dependencies as the successor state $\mathbf{s}'$ cannot be fully explained by the current state $\mathbf{s}$ and action $\mathbf{u}$. This requirement is a key aspect of a Markov decision process (MDP, [17]) that generally describes a control task. More formally, the state has to bear the Markov property

$$p(\mathbf{s}_t|\mathbf{u}_{0:t-1},\mathbf{s}_{0:t-1}) = p(\mathbf{s}_t|\mathbf{u}_{t-1},\mathbf{s}_{t-1}). \tag{10}$$

The Markov assumption together with a reward r that the agent receives upon taking action $\mathbf{u}$ in $\mathbf{s}$ complete the MDP. The relationship between states from one time step to the next is governed by a transition function $(\mathbf{s},\mathbf{u}) \rightarrow \mathbf{s}'$ corresponding to the forward dynamics. As the GP is a probability distribution over functions, the prediction quality is reduced the more the Markov assumption is violated. Hence by forcing the state to be Markov, the modeling of the forward dynamics with a GP is facilitated.

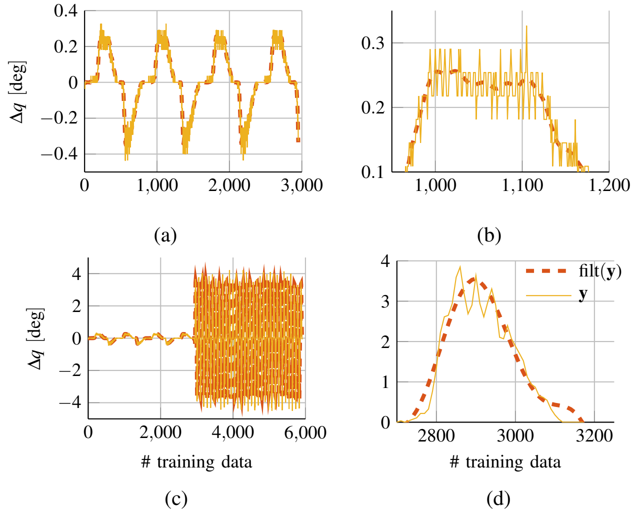
The Markov requirement, however, has often been violated by weak dependence on the past, leading to models that are adequate but sub-optimal. On real systems, dependencies often remain unmeasured due to the lack of sensors or tedious measurement procedures. Examples are 1) estimating the temperature in PAMs, 2) stiction and friction effects as well as 3) slack of the cables. Nonetheless, the information of such unobserved effects is captured in the transition $(\mathbf{s}_t,\mathbf{u}_t) \rightarrow \mathbf{s}_{t+1}$ as different successor states will be reached when applying the same action in the same state at different times. Such a problem is described as a partially observable MDP (POMDP).

A possible solution to this problem is to approximate this POMDP with a K-th order MDP by concatenating the previous states and actions into an augmented state

$$\mathbf{s}_t^{\text{aug}} = [\mathbf{s}_t, \mathbf{s}_{t-1}, \dots, \mathbf{s}_{t-k}, \mathbf{u}_t, \mathbf{u}_{t-1}, \dots, \mathbf{u}_{t-k}]\,, \tag{11}$$

an idea closely related to the NARX concept [18].

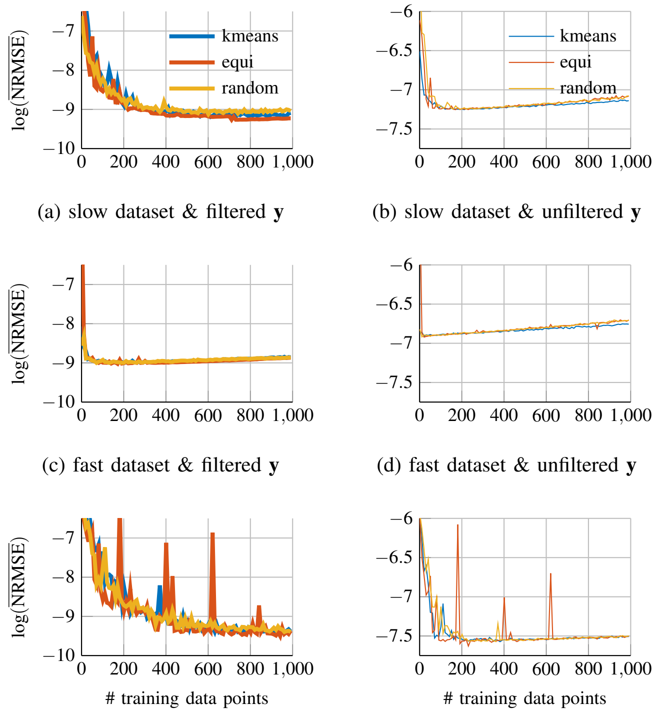
a) State Elements based on Analytical Models: Dependencies that should be incorporated into the state can be obtained for a motor-driven robot from the rigid body dynamics Eq. (1). As some dynamic properties of PAMs are also captured in the analytical model Eqs. (5) and (6), it is highly likely that some form of the variables governing Eqs. (5) and (6) need to be included into the state. Hence, the joint angle $\mathbf{q}$ and angle velocities $\dot{\mathbf{q}}$ as well as muscle lengths $\mathbf{l}$, muscle contraction speeds $\dot{\mathbf{l}}$, current PAM pressures $\mathbf{p}$ and current pressure rates of change $\dot{\mathbf{p}}$ should be part of the state. Incorporating the pressures $\mathbf{p}$ into the state also helps to resolve hysteresis in the $\mathbf{p} \rightarrow \mathbf{q}$ relationship because rising and falling curves can be discriminated according to $\mathbf{p}_t \leq \mathbf{p}_t^{\text{des}}$. The order of the system determines how many time-derivatives need to be added to the state. While the rigid body dynamics Eq. (1) suggest that the robot arm dynamics is a second order system (neglecting cable dynamics), the lack of well-established analytical models for PAMs means that the order of PAM dynamics is unclear. For instance, [10] indicates that the activation from Eq. (6) should possibly be modeled as a second order system but do not take the higher order into account to reduce computational load. In addition, for slower movements higher order derivatives might not play such an essential role in the model prediction and can be held out from the state to avoid higher input dimensionality of the GP. For this reason, we test slow and fast trajectory datasets in the experiments in Section IV and check how performance is influenced by the orders of the system.



(a) (b) (c) (d)

Fig. 3: Datasets under consideration in this work depicted for illustration. All datasets were created by applying two antiphase periodic pressure trajectories to the antagonistic PAM pair and recording the resulting sensor values. The target Δq represents the difference to the next state and is depicted in filtered and raw sensor form. A non-causal 10th order Butterworth lowpass filter was used for filtering. (a) 'slow' dataset. (b) Closer view of (a) illustrates that the relative angle encoders add considerable noise for slow motions. (c) 'slow' and 'fast' datasets form the 'mixed' dataset. (d) Closer view of fast part of (c). For faster movements the noise is smaller. Hence, the 'mixed' dataset includes two types of noise. Faster movements excite higher order dynamics components which need to be expressed by the model.

b) Estimated State Elements: Another issue with time derivatives is that they are not sensed directly but have to be estimated. A common way is to use finite differences $\dot{m}_t = (m_{t+1} - m_{t-1})/(2\Delta t)$ where Δt is the sampling period and m_t a measurement. However, the noise on $\dot{m}_t$ would in this case multiply. A GP only assumes output noise and would experience major complications. Unfortunately, from our experience, the delay introduced by filtering the sensor values online on the real robot, even with a second order Butterworth lowpass



(a) slow dataset & filtered **y**

(b) slow dataset & unfiltered **y**

(c) fast dataset & filtered **y**

(d) fast dataset & unfiltered **y**

(e) mixed dataset & filtered **y**

(f) mixed dataset & unfiltered **y**

Fig. 4: The influence of random, k-means and equidistant training data selections model is illustrated with increasing size of training dataset on the (a) & (b)'slow', (c) & (d) 'fast' and (e) & (f) 'mixed' dataset. Experiments have been performed five times and the mean of the logarithm of the normalized root mean squared error $\log(\overline{NRMSE})$ is depicted. Subfigures (a), (c) and (e) have been tested against filtered test targets **y** and (b), (d) and (f) against unfiltered **y**. The 'fast' dataset is composed of similar periodic curves and hence needs the least number of data. In case the $\log(\overline{NRMSE})$ is calculated against unfiltered test targets, the $\log(\overline{NRMSE})$ tends to rise for more data points provided because the prediction of the GP becomes smoother as the noise modeling improves. The graphs for most realistic 'mixed' dataset decrease slowest. The regularly-spaced spikes in (e) & (f) occur due to sampling of the equidistant scheme at the same location in the periodically repeating data.

filter, substantially impairs control performance. We hence test how adding previous joint angles $\mathbf{q}_{t:t-h}$ and pressures $\mathbf{p}_{t:t-o}$ instead of their respective time-derivatives alter the prediction performance without filtering the input training data. For instance, the sequence $[\mathbf{q}_t, \mathbf{q}_{t-1}, \mathbf{q}_{t-2}]$ contains the same information as $[\mathbf{q}_t, \dot{\mathbf{q}}_t, \ddot{\mathbf{q}}_t]$ but corrupted by less noise.

A similar transformation as finite differences is required to attain the lengths of the PAMs. The lengths of the muscles are $l_{a,b} = l_0 \pm (q/2\pi)r$ where l_0 is the muscle length for $q = 0$ deg assuming a radial joint with fixed radius r and cables that are always under tension (so that any change in the joint angle is solely due to a change in muscle length and not slack in the cables). The lengths of the PAMs can then be inferred immediately from the joint angles. An interesting question is whether the GP is able to recover such transformations in addition to unobserved dependencies such as elongation and slack of the cables. Based on the previous discussion, we propose to test the following state compositions : 1) $\mathbf{s}_t^{\text{noisy}} = [\mathbf{q}_t, \dot{\mathbf{q}}_t, \mathbf{p}_t, \dot{\mathbf{p}}_t]$, 2) $\mathbf{s}_t^{\text{padded}} = [\mathbf{q}_{t:t-h}, \mathbf{p}_{t:t-o}]$, and 3) $\mathbf{s}_t^{\text{action}} = [\mathbf{s}_t^{\text{x}}, \mathbf{p}_{t:t-h}^{\text{des}}]$ where $\mathbf{s}_t^{\text{x}}$ is a placeholder for either $\mathbf{s}_t^{\text{noisy}}$ or $\mathbf{s}_t^{\text{padded}}$. The state compositions are empirically tested in Section IV.

3) Model Validation: According to [19] a model can only be good enough to fulfill its purpose. [20] coins this aspect purposiveness. The process of disentangling the contribution of the model and the controller to the overall tracking performance is quite involved. Hence, often the model is assessed separately by the long-term predictions abilities [7], [8]. This procedure involves a previously collected data sequence consisting of a state $[\mathbf{s}_t]_{t=0}^{T}$ and a corresponding action trajectory $[\mathbf{u}_0]_{t=1}^{T-1}$. Starting with the initial input $\mathbf{x}_0 = [\mathbf{s}_0, \mathbf{u}_0]$ to the GP, the prediction of the next state is fed together with the next action as input for the next time step. This iterative procedure is continued until the predicted state deviates sufficiently from the known state trajectory, e.g. $||\mathbf{s}_\text{h}^{\text{des}} - \mathbf{f}(\mathbf{s}_{\text{h}-1}, \mathbf{u}_{\text{h}-1})||^2 > \xi$. The horizon h then represents how well the roll-outs can be simulated and is important, for instance, for model predictive control type of control frameworks. For muscular systems, the state is required to contain the pressures inside the PAMs. Long-term predictions are then always corrupted by the quality of the pressure model that, in return, is not predefined by the desired trajectory. PAM pressures play a role only when stiffness should be modulated along with the position trajectory, which is not the focus of this work. For this reason, we decide to resort to one-step-ahead predictions and only predict entities that are essential to calculate a control error. Many criteria exist to measure model quality for one-step-ahead predictions, see [20] for a detailed description. Here, we employ a normalized version of the root mean squared error

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{\text{N}} (\text{f}^{(k)}(\mathbf{s}_{i-1}, \mathbf{u}_{i-1}) - s_i^{(\text{k})})^2}{\text{N}}}\,, \qquad (12)$$

which addresses datasets with different magnitudes

$$\text{NRMSE} = \frac{\text{RMSE}}{\text{s}_{\text{max}}^{(k)} - \text{s}_{\text{min}}^{(k)}}\,, \qquad (13)$$

where $\text{f}^{(k)}$ is the k-th element of the prediction and $k \in [1, D]$, $\text{s}_{\text{max}}^{(k)}$ and $\text{s}_{\text{min}}^{(k)}$ the maximal and minimal value of the state component k in the dataset comprising N data points. The predictions can be tested against either the filtered or raw targets from the dataset. The filtering of the target signal can be done with a non-causal filter, e.g. zero-phase Butterworth lowpass filter which does not add additional delay. Filtered outputs have the benefit of being closer to the latent true data as can be seen in Fig. 3 (b). We test both cases in Section IV.

4) Filtering Training Outputs: We decided to filter the training outputs although a GP in principle handles output noise. However, we found state dependent noise in our training dataset which requires heteroscedastic noise treatment. Fig. 3 (b) and (d) illustrate that the unfiltered graph changes more randomly for the slow dataset than for the fast dataset. Heteroscedastic datasets often occur in real applications. Unfortunately, inference in heteroscedastic GPs is based on approximations and training is computationally more demanding. For this reason, we decided to filter the training targets such that different noise levels of the dataset are balanced to some extent.

Using approaches to handle input noise for GPs [21], [22] is a valid alternative and subject to future work. In this project, we aim at highlighting problems and give basic answers. Thus, we try to avoid making approximations as it is hard to quantify how doing so influences the result.

5) Subset Selection: Robots accumulate large amounts of correlated data at high frequencies. Keeping all this data in the training dataset for non-parametric data is not possible. Hence, the data need to be further subsampled. We do not incorporate approximations to the GP in order to realize bigger training datasets, e.g. with pseudo inputs [15] as such approaches also rely on approximations.

Hence, we pick a subset of data from the complete dataset instead. As a baseline we consider random subsampling, over which any sensible subsampling approach should improve. The default choice is often equidistant subsampling where N data points are picked from the dataset that are equally far apart. If, however, the resulting sampling frequency is less than twice the maximum frequency of the time series, some aspects of the signal cannot be captured according to the Nyquist-Shannon sampling theorem. On the other hand, areas where little change happens could be sampled too often. As an alternative, we use the k-means algorithm to find N clusters among the complete dataset and pick the data points that are closest to each cluster. Thus, data points are selected so as to maximize the difference among the training set according to the Euclidean distance. The GP is hence more expressive in regions of the training data exhibiting greater variation.

C. Variance-Regularized Control

The control goal is to track a desired trajectory $[\mathbf{s}_t^{\text{des}}]_{t=1}^T$ with a control trajectory $[\mathbf{u}_t]_{t=1}^T$ with minimal deviation. A simple way to express this desire is to define the squared error

$$\mathrm{e}_t^2 = ||(\mathbf{s}_t^{\text{des}} - \mathbf{s}_t)||_{\mathrm{Q}}^2 \,, \tag{14}$$

for each time step t. The mean of the forward dynamics model $\bar{\mathbf{f}}$ from Eq. (7) can be incorporated for the prediction of e_{t+1}^2 in the next time step

$$\begin{aligned} \bar{\mathrm{e}}_{t+1}^2 &= (\mathbf{s}_{t+1}^{\text{des}} - \mathbf{s}_{t+1})^T \mathrm{Q} (\mathbf{s}_{t+1}^{\text{des}} - \mathbf{s}_{t+1}) \\ \to \bar{\mathrm{e}}_{t+1}^2(\mathbf{u}_t) &= (\mathbf{s}_{t+1}^{\text{des}} - \bar{\mathbf{f}}(\mathbf{s}_t, \mathbf{u}_t))^T \mathrm{Q} (\mathbf{s}_{t+1}^{\text{des}} - \bar{\mathbf{f}}(\mathbf{s}_t, \mathbf{u}_t)) . \end{aligned} \tag{15}$$

	slow	fast	mixed
$\mathbf{s}_t^{(1)}$	$2.6e-4 \pm 1.1e-4$ $2.6e-4 \pm 3.6e-5$ $2.0e-4 \pm 0.0e+0$	$1.3e-4 \pm 2.2e-6$ $1.3e-4 \pm 1.1e-5$ $1.3e-4 \pm 0.0e+0$	$4.8e-4 \pm 2.7e-4$ $4.8e-4 \pm 3.5e-4$ $3.6e-4 \pm 0.0e+0$
$\mathbf{s}_t^{(2)}$	$2.8e-4 \pm 3.7e-5$ $2.0e-4 \pm 1.3e-5$ $2.9e-4 \pm 0.0e+0$	$1.3e-4 \pm 6.5e-6$ $1.3e-4 \pm 3.7e-6$ $1.3e-4 \pm 0.0e+0$	$4.6e-4 \pm 3.0e-4$ $3.1e-4 \pm 1.2e-4$ $3.1e-4 \pm 0.0e+0$
$\mathbf{s}_t^{(3)}$	$2.8e-4 \pm 9.0e-5$ $2.6e-4 \pm 4.4e-5$ $2.9e-4 \pm 0.0e+0$	$1.3e-4 \pm 2.2e-6$ $1.3e-4 \pm 3.4e-6$ $1.3e-4 \pm 0.0e+0$	$3.0e-4 \pm 6.9e-5$ $2.8e-4 \pm 8.3e-5$ $2.7e-4 \pm 0.0e+0$
$\mathbf{s}_t^{(4)}$	$2.8e-4 \pm 6.4e-5$ $2.3e-4 \pm 3.4e-5$ $2.9e-4 \pm 0.0e+0$	$1.3e-4 \pm 2.6e-6$ $1.3e-4 \pm 5.3e-6$ $1.3e-4 \pm 0.0e+0$	$1.7e-4 \pm 9.0e-5$ $9.9e-5 \pm 2.7e-5$ $2.6e-4 \pm 0.0e+0$
$\mathbf{s}_t^{(5)}$	$3.4e-4 \pm 9.0e-5$ $3.0e-4 \pm 3.6e-5$ $3.2e-4 \pm 0.0e+0$	$1.3e-4 \pm 1.3e-5$ $1.3e-4 \pm 5.3e-6$ $1.3e-4 \pm 0.0e+0$	$4.3e-4 \pm 1.3e-4$ $2.7e-4 \pm 1.0e-4$ $3.1e-4 \pm 0.0e+$

TABLE I: Normalized root mean squared error from Eq. (13) calculated for the prediction of each model with different state compositions (rows) against filtered target values from the datasets (columns) depicted in Fig. 3. $\mathbf{s}_t^{(1)} = [q_t]$, $\mathbf{s}_t^{(2)} = [q_t, q_{t-1}, q_{t-2}]$, $\mathbf{s}_t^{(3)} = [q_t, q_{t-1}, q_{t-2}, p_t^a, p_t^b]$, $\mathbf{s}_t^{(4)} = [q_t, q_{t-1}, q_{t-2}, p_t^a, p_t^b, p_{t-1}^{a,\text{des}}, p_{t-1}^{b,\text{des}}]$, $\mathbf{s}_t^{(5)} = [q_t, \dot{q}_t, p_t^a, p_t^b]$ Experiments have been performed five times under same conditions illustrated by the mean NRMSE and one standard deviation. 100 data points have been selected to the training set by random, k-means and equidistant selection. Equidistant selection is deterministic and thus has always zero standard deviation. The best performance on the 'slow' dataset is achieved with $\mathbf{s}_t^{(2)}$ and k-means selection. 100 training data points are sufficient to be independent of state composition as well as data selection type for the 'fast' dataset as a low amount of information is contained. The 'mixed' dataset is corrupted by heteroscedastic noise and requires modeling of higher order dynamics. Best performance is achieved with more complex state representations, $\mathbf{s}_t^{(4)}$, as well as k-means selection that is capable to select more distinctive data points compared to the other selection types.

The controls in each time step t can then be extracted using

$$\mathbf{u}_t = \underset{\mathbf{u}}{\operatorname{argmin}} \, \bar{\mathrm{e}}_{t+1}^2(\mathbf{u}) \,. \tag{16}$$

The main idea of our control framework is to make use of the full posterior distribution that is given by our probabilistic forward model. Thus, we optimize for the expected value of the loss, similar to [23]. In this case, Eq. (14) becomes

$$\begin{aligned} \mathrm{E}[\mathrm{e}_{t+1}^2(\mathbf{u})] &= \mathbf{s}_{t+1}^{\text{des}\,T} \mathrm{Q} \mathbf{s}_{t+1}^{\text{des}} - 2 \mathbf{s}_{t+1}^{\text{des}\,T} \mathrm{Q} \mathrm{E}[\mathbf{f}] + \mathrm{E}[\mathbf{f}^T \mathrm{Q} \mathbf{f}] \\ &= \mathbf{s}_{t+1}^{\text{des}\,T} \mathrm{Q} \mathbf{s}_{t+1}^{\text{des}} - 2 \mathbf{s}_{t+1}^{\text{des}\,T} \mathrm{Q} \bar{\mathbf{f}} + \bar{\mathbf{f}}^T \mathrm{Q} \bar{\mathbf{f}} + \mathrm{tr}(\mathrm{Q}\Sigma) \\ &= \bar{\mathrm{e}}_{t+1}^2(\mathbf{u}) + \mathrm{tr}(\mathrm{Q}\Sigma(\mathbf{u})) \end{aligned} \tag{17}$$

with $[\Sigma]_{i,j} = \mathrm{cov}(\mathrm{f}_i, \mathrm{f}_j)$ being the covariance of each GP in the forward model vector $\mathbf{f}$ and $\mathrm{diag}(\Sigma) = \sigma^2$. We have left out the dependence on $\mathbf{u}$ in the derivation of Eq. (17) to keep the math uncluttered and added $(\mathbf{u})$ in the last line where appropriate. Eq. (17) can be interpreted as the sum of the mean of the squared error from Eq. (15) and regularized by the variances of each element of the forward model vector $\mathbf{f}$ weighted by the diagonal elements of Q. This regularization poses a constraint to the overactuation of antagonistic actuation. Hence, by using this control framework, the controls are chosen such that the successor state is near to the training dataset.

IV. Experiments and Evaluations

The goal of this paper is to illustrate how to set up a GP dynamics model and subsequently use it for control of a muscular system. In this section we validate the model considerations from Section III with experiments. First, we illustrate how equidistant, random and k-means subsampling influences the prediction performance with an increasing number of training data points. The next experiments test state compositions candidates on different datasets with characteristic challenges and with distinct subsampling types. In the third experiment, we show that the best performing model set-up from the previous experiments can be used to control one real pneumatic muscle pair. We further demonstrate that leveraging the full posterior of our stochastic forward model can help to keep the system near the training data. Our new control approach outperforms our previous PID controller from [3].

A. Subset Selection and Model Validation

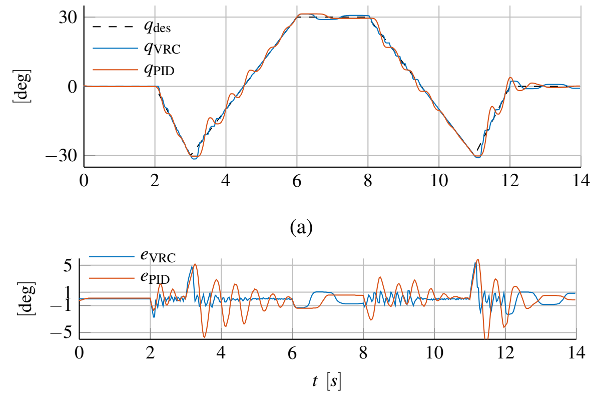
In this experiment, we test how the prediction performance changes with the number of training data points as well as with the subset of data selection strategy on the slow, fast and mixed datasets from Fig. 3. The slow dataset challenges the model to accurately distinguish noise from signal. As the relative angle encoders of our PAM-actuated robot have a resolution of 0.036°, relatively high quantization noise corrupts the signal as can be seen in Fig. 3 (b). The fast dataset consists of periodic movements with higher frequency. Thus, more data represents the same curve compared to the slow datasets but higher order dynamics components are excited. The challenge of the mixed dataset is its heteroscedastic noise that often occurs in real scenarios. The prediction performance is measured by the NRMSE from (13) against filtered and unfiltered training data targets $\mathbf{y}$. The models always predict the difference to the next joint angle Δq.

Fig. 4 depicts these experiments. One can observe that in all graphs the mean of the NRMSE decreases with increasing number of training data points when tested against the filtered test targets. In contrast, the graphs for tests against unfiltered $\mathbf{y}$ start to rise. This fact illustrates that testing against filtered test data outputs gives a better estimate of the model's quality as the model is guaranteed to improve with more data. Another observation is that the fast dataset can be learned with a smaller number of training data as it comprises less information. In Fig. 4 (e), (f) the graph for equidistant subsampling is corrupted by spikes that occur periodically. These spikes happen when the same location on the periodically repeating signal is sampled as can happen with equidistant subsampling. Hence, not all information is captured in the dataset and leads to worse prediction performance.

B. Evaluation of State Composition

Section III-B discusses important issues regarding the state composition like 1) actuator delay, 2) non-Markovian states and 3) noisy time-derivatives and derives possible solutions. In this section, these proposed states are evaluated by measuring the NRMSE on the three different datasets from Fig. 3 with their specific characteristics and challenges. We also compare how the choice of subsampling type from Section III-B5 affects the prediction performance. Table I displays the results. Each experiment is performed with 100 training data points and repeated five times under identical settings. The NRMSE is hence indicated as mean and one standard deviation.

It is noteworthy that the standard deviation for all experiments using equidistant subsampling is zero as no source of randomness is present. For k-means subsampling, the stochasticity is introduced by initial random seeds of the clusters. Also striking is that the prediction performance for the fast dataset is independent of the state composition and subsampling type. The reason is that the fast dataset comprises less information compared to the other datasets. Thus, 100 data points are enough to learn the model well. Further conclusions can be made from Table I. First, higher order components are excited in the mixed dataset as it includes the fast dataset. Comparing the prediction performance for $\mathbf{s}_t^{(1)}$ and $\mathbf{s}_t^{(2)}$ with k-means subsampling, one can see that higher order elements contained in $\mathbf{s}_t^{(2)}$ lead to improved performance. Second, states $\mathbf{s}_t^{(3)}$ contains time-derivatives that were estimated using finite differences whereas $\mathbf{s}_t^{(5)}$ instead includes time-padded elements as suggested in section III-B2b. Hence, the NMRSE is generally lower for $\mathbf{s}_t^{(3)}$ on the slow dataset that is corrupted by relatively strong noise. The best overall performance on the mixed dataset is achieved with the most involved state representation of $\mathbf{s}_t^{(4)}$ with k-means subsampling. This dataset is the most realistic as it contains heteroscedastic noise as well as excites higher order terms. K-means subsampling assures that the most different data points are selected to be in the training dataset. In this manner, the flexibility introduced by state $\mathbf{s}_t^{(4)}$ has a better chance to be fully exploited. In general, for rich datasets, the state representation is required to be expressive and hence needs to avoid noisy input data and contain higher order terms. For simpler datasets, such as data collected locally as in [13], a simpler state representation can be used in order to speed up computation.



(a)

(b)

Fig. 5: Tracking experiment that validates our model on a real one antagonistic PAM pair and employs the variance-regularized control (VRC). A dataset is collected using a manually tuned PID controller along the des. trajectory and subsampled with the k-means strategy. The VRC settles faster to the error bound of ±1 degree after sharp changes in the des. trajectory compared the PID.

C. Evaluation of Control Performance

We now test the use of the learned model in the variance-regularized control framework (VRC) from Section III-C on the robot arm. We do so by controlling one DoF of our system described in Section II as a showcase that our model derived from the preceding discussion and experiments can be used to control a real PAM system. Separate control laws for each DoF can be designed as the dynamics of our light-weight rigid body structure can be neglected in comparison to the friction of the cable drives and dynamics of the pneumatic muscles. We employ the state representation $\mathbf{s}_t^{(4)}$ from the preceding experiment and the GP predicts the difference to the current joint angle Δq. 100 training data points are subsampled using the k-means approach from the complete dataset. This dataset is generated by tracking multiple reference trajectories with the PID controller from our previous work [3]. The PID parameters are manually tuned and need to be adapted for significantly different desired trajectories. In order to decide on two desired pressures based on one control signal that the PID provides, we employ the 'symmetrical co-contraction approach' from [24].

The real-time optimization of the VRC scheme from Eq. (16) is numerically approximated. The method-of-moving-asymptotes algorithm (MMA, [25]) has proven to practically work well, and in our experiments we use the implementation from the C++ NLOPT toolbox [26]. We found helpful to iterate through all algorithms implemented in NLOPT as well as plot the error surface to be optimized (for VRC this is Eq. (17)) offline after tracking with the PID controller to find a good parameter setting for VRC. The parameters in Q from Eq. (17) can be conveniently set by $\mathrm{Q}_{i,i} = c_i/\sigma_f^{(i)}$ where

$\sigma_f^{(i)}$ is the signal variance of the GP predicting the i-th state element of successor state $\mathbf{s}'$. In this manner the contribution of the variance regularization is not changing in case the GP is retrained as $\sigma_f^{(i)}$ is the maximum variance prediction of the GP (in case the system is far away from the training data). The parameter c_i can then be chosen in order to balance between minimizing the squared control error $\bar{e}_{t+1}^2$ and the variance regularization term.

We chose to evaluate the performance of our controller, and a baseline PID controller, on a challenging trajectory that includes rapid changes in direction. From Fig. 5, we notice how the control performance of the VRC approach remained within a control error bound of ±1 deg most of the time. The PID, however, oscillates outside this error bound in response to the sudden change in the des. trajectory. One possible reason why VRC performs better is that the GP captures essential information from this oscillations and inhibits oscillation of VRC due to the prediction one time step ahead. VRC can then choose controls that lead the system towards the desired trajectory. The regularization term helps to stay near the training data, thus, helps to keep the GP predictions valid. The PID has fixed parameters, VRC is in this sense more adaptive.

V. Conclusion

This paper aims at modeling the complex dynamics of PAM-actuated systems with Gaussian process dynamics models and using these for control. To accomplish this goal, we discussed and tested important problems that arise for muscular systems such as state-dependent and hence heteroscedastic sensor noise, overactuation and actuator delay. We elaborated on the implications of such issues and related them to model learning such as 1) state components that have been derived from analytical dynamics equations to construct a Markov state, 2) validating the model with the NRMSE instead of long-term predictions as they are less significant in this scenario, 3) using filtered training targets for model validation and 4) to use time-padded state elements instead of noisy time-derivatives estimated from finite differences. Experiments were performed to confirm these implications.

Moreover, to solve the issues deriving from overactuation (which offers multiple solution to the control problem as an infinite set of PAM pressures to the desired joint angle in the next time step) we propose a novel control scheme that includes as additional objective to minimizes the variance of the controller (VRC). After discussing important technical details on how to practically implement VRC, we evaluate our approach on a real pneumatic arm and show that VRC performs better than a PID controller.

In future work, we want to integrate the GP forward dynamics model into model-based Reinforcement Learning approaches. Furthermore, we plan to design models that take the future horizon into account and employ such control schemes to dynamic and uncertain tasks such as table tennis.

References

[1] V. Mnih and et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.

[2] B. Tondu, "Modelling of the McKibben artificial muscle: A review," *Journal of Intelligent Material Systems and Structures*, vol. 23, no. 3, pp. 225–253, Feb. 2012.

[3] D. Büchler, H. Ott, and J. Peters, "A Lightweight Robotic Arm with Pneumatic Muscles for Robot Learning," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2016.

[4] K. Hosoda, S. Sekimoto, Y. Nishigori, S. Takamuku, and S. Ikemoto, "Anthropomorphic Muscular–Skeletal Robotic Upper Limb for Understanding Embodied Intelligence," *Advanced Robotics*, vol. 26, no. 7, pp. 729–744, Jan. 2012.

[5] R. Blickhan, A. Seyfarth, H. Geyer, S. Grimmer, H. Wagner, and M. Günther, "Intelligence by mechanics," *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 365, no. 1850, pp. 199–220, 2007.

[6] D. Nguyen-Tuong and J. Peters, "Model learning for robot control: a survey," *Cognitive processing*, vol. 12, no. 4, pp. 319–340, 2011.

[7] A. Doerr, C. Daniel, D. Nguyen-Tuong, A. Marco, S. Schaal, T. Marc, and S. Trimpe, "Optimizing Long-term Predictions for Model-based Policy Search," in *Conference on Robot Learning*, 2017, pp. 227–238.

[8] S. Eleftheriadis, T. Nicholson, M. Deisenroth, and J. Hensman, "Identification of Gaussian process state space models," in *Advances in Neural Information Processing Systems*, 2017, pp. 5315–5325.

[9] J. Vinogradska, B. Bischoff, D. Nguyen-Tuong, and J. Peters, "Stability of Controllers for Gaussian Process Dynamics," *Journal of Machine Learning Research*, vol. 18, no. 100, pp. 1–37, 2017.

[10] F. E. Zajac, "Muscle and tendon: properties, models, scaling, and application to biomechanics and motor control." *Critical reviews in biomedical engineering*, vol. 17, no. 4, pp. 359–411, Dec. 1988.

[11] C.-P. Chou and B. Hannaford, "Static and dynamic characteristics of McKibben pneumatic artificial muscles," in *IEEE International Conference on Robotics and Automation*. IEEE, 1994, pp. 281–286.

[12] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*. Cambridge, Mass.: MIT Press, 2006.

[13] D. Nguyen-Tuong, J. R. Peters, and M. Seeger, "Local Gaussian process regression for real time online model learning," in *Advances in Neural Information Processing Systems*, 2009.

[14] M. P. Deisenroth, D. Fox, and C. E. Rasmussen, "Gaussian processes for data-efficient learning in robotics and control," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 37, no. 2, pp. 408–423, 2015. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6654139

[15] E. Snelson and Z. Ghahramani, "Sparse Gaussian processes using pseudo-inputs," in *Advances in Neural Information Processing Systems 18*. MIT press, 2006, pp. 1257–1264.

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

[17] Richard S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[18] S. A. Billings, *Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains*. John Wiley & Sons, 2013.

[19] L. Ljung, *System identification: theory for the user*. Prentice Hall PTR, USA, 1999.

[20] J. Kocijan, "System Identification with GP Models," in *Modelling and Control of Dynamic Systems Using Gaussian Process Models*, ser. Advances in Industrial Control. Springer, 2016, pp. 21–102.

[21] A. McHutchon and C. E. Rasmussen, "Gaussian process training with input noise," in *Advances in Neural Information Processing Systems*, 2011, pp. 1341–1349.

[22] A. C. Damianou, M. K. Titsias, and N. D. Lawrence, "Variational inference for latent variables and uncertain inputs in Gaussian processes," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1425–1486, 2016.

[23] D. Sbarbaro, R. Murray-Smith, and A. Valdes, "Multivariable Generalized Minimum Variance Control Based on Artificial Neural Networks and Gaussian Process Models," in *Advances in Neural Networks*. Springer, Aug. 2004, pp. 52–58.

[24] B. Tondu and P. Lopez, "Modeling and control of McKibben artificial muscle robot actuators," *IEEE Control Systems*, 2000.

[25] K. Svanberg, "A class of globally convergent optimization methods based on conservative convex separable approximations," *SIAM journal on optimization*, vol. 12, no. 2, pp. 555–573, 2002.

[26] Steven G. Johnson, "The NLopt nonlinear-optimization package." [Online]. Available: http://ab-initio.mit.edu/nlopt