Toward Fast Policy Search for Learning Legged Locomotion

Marc Peter Deisenroth[1] Roberto Calandra[1] André Seyfarth[2] Jan Peters[1,3]

***Abstract*— Legged locomotion is one of the most versatile forms of mobility. However, despite the importance of legged locomotion and the large number of legged robotics studies, no biped or quadruped matches the agility and versatility of their biological counterparts to date. Approaches to designing controllers for legged locomotion systems are often based on either the assumption of perfectly known dynamics or mechanical designs that substantially reduce the dimensionality of the problem. The few existing approaches for learning controllers for legged systems either require exhaustive real-world data or they improve controllers only conservatively, leading to slow learning. We present a data-efficient approach to learning feedback controllers for legged locomotive systems, based on learned probabilistic forward models for generating walking policies. On a compass walker, we show that our approach allows for learning gait policies from very little data. Moreover, we analyze learned locomotion models of a biomechanically inspired biped. Our approach has the potential to scale to high-dimensional humanoid robots with little loss in efficiency.**

I. Introduction

Legged locomotion is one of the most versatile forms of mobility for robots with prospective applications, e.g., rescue robotics, disaster site exploration, prosthetics [8]. Despite the importance of legged locomotion and the large number of corresponding studies, no biped or quadruped reproduces the agility and versatility of their biological counterparts to date.

Two key challenges have been particularly problematic for developing more dexterous bipedal robot locomotion. First, the dimensionality of fully actuated dynamic walkers is too high to manually design controllers. Second, current robots are frequently not built to allow for versatile forms of locomotion, based on compliant joints with muscle-like action generation for energy storage or inter-joint mechanical coupling. Instead, robots are built to be straightforward to control. They are usually either optimized for accurate trajectory tracking, such as Honda's Asimo, or having passive dynamics with only a limited access for actuation [18]. Reducing the complexity resulting from the high dimensionality of the state-action space has been at the core of most legged locomotion systems and is accomplished by smart design of either the control system [10], [25] or the mechanics [13].

Neuromuscular legged systems [7] are clearly capable of using biomechanics that simplify the control problem such that a neural control system with its long delays (a signal from the brain-stem to the foot takes about $120\,\mathrm{ms}$ [12]) is still capable of versatile and agile movements. While muscle-based biomechanical systems have favorable properties [28] and shield the control system from a variety of problems inherent to technical systems [27], the reduction of dimensionality is not amongst them. There is evidence that the human motor control system uses internal models [26] to deal with this complexity, possibly aided by dimensionality reduction in the spinal cord [21].

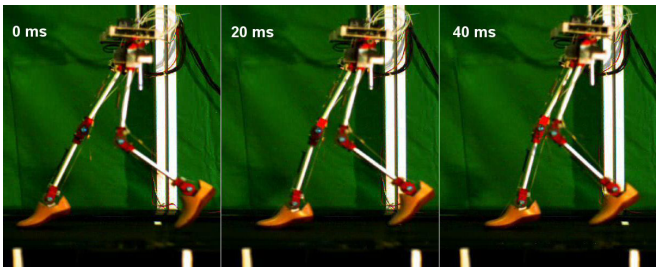


Fig. 1: The biomechanically inspired JenaWalker II [23] is controlled by adjusting muscle stiffness.

In this paper, we show that we can reproduce several features of the neuromuscular design: (i) Dimensionality reduction can deal with the complexity of a biomechanically-inspired design and (ii) bipedal walking can be learned efficiently using forward models acquired from data.

There have been a series of applications of machine learning approaches in legged locomotion such as *Imitation Learning* [20] and *Reinforcement Learning* (RL) [24], [22]. However, model-free policy gradient approaches, as in [24], require a well-structured policy parametrization with few parameters to be feasible and cannot re-use data. Similarly, model-free Q-learning [22] is data inefficient due to the continuous state-action spaces. In [15], it was shown that model-based RL with deterministic models does not suffice for learning locomotion as it cannot cope with the variability in the movement and the optimization bias.

In this paper, we show that learning probabilistic forward models for generating walking policies in conjunction with dimensionality reduction is tractable and data efficient. The presented approach is based on [4] and can efficiently learn feedback controllers from scratch while dealing with model uncertainties in a principled way. Hence, our approach has the potential to scale to high-dimensional humanoid robots with little loss in efficiency. We demonstrate the speed of learning gait policies for a compass walker [3]. Moreover, we detail modeling aspects for simulation data from the biomechanically inspired biped shown in Fig. 1.

[1]Intelligent-Autonomous-Systems Lab, Department of Computer Science, Technische Universität Darmstadt, Germany.

[2]Locomotion Lab, Department of Sport Science, Technische Universität Darmstadt, Germany.

[3]MPI for Intelligent Systems, Tübingen, Germany.

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007–2013) under grant agreement #270327 and by DFG within the grant #SE1042/1.

II. Policy Search for Learning Locomotion

Finding good locomotion policies is a generically challenging problem. For idealized models, a large number of planning and optimization methods can yield a solution. However, the dynamics of such muscle-based bipeds are very hard to model. Based on an inexact uncertain model, controller design for the real locomotive system is challenging, especially in high-dimensional state spaces. In this case, data-driven approaches, such as RL, may be better suited.

We present a model-based RL approach to *learning* state-feedback controllers for locomotive systems, without strong prior assumptions on the dynamics. Instead, we learn purely data-driven forward models that explicitly express their uncertainty about the true underlying locomotive dynamics. By taking this uncertainty into account during long-term planning and controller learning, our feedback controller is robust to model errors.

A. Problem Setup and Notation

Throughout this paper, we assume the locomotive dynamics follow the discrete-time Markov chain $\mathbf{x}_t = f(\mathbf{x}_{t-1}, \mathbf{u}_{t-1}) + \mathbf{w}$ with continuous-valued states $\mathbf{x} \in \mathbb{R}^D$ and controls $\mathbf{u} \in \mathbb{R}^F$ and unknown transition dynamics f. The measurement noise term $\mathbf{w}$ is assumed zero-mean i.i.d. Gaussian with unknown (diagonal) covariance matrix $\mathbf{\Sigma}_w$. The objective is to find a parametrized *policy* $\pi : \mathbf{x} \mapsto \pi(\mathbf{x}, \boldsymbol{\theta}) = \mathbf{u}$ that minimizes the *expected cost-to-go*

$$J^\pi(\boldsymbol{\theta}) = \sum_{t=0}^{T} \mathbb{E}_{\mathbf{x}_t}[c(\mathbf{x}_t)], \quad \mathbf{x}_0 \sim \mathcal{N}(\boldsymbol{\mu}_0, \mathbf{\Sigma}_0), \qquad (1)$$

of following π for T steps from an initial state distribution $p(\mathbf{x}_0) = \mathcal{N}(\boldsymbol{\mu}_0, \mathbf{\Sigma}_0)$, where $c(\mathbf{x}_t)$ is the cost of being in state $\mathbf{x}$ at time t. The policy parameters are denoted by $\boldsymbol{\theta}$.

Policy search RL methods have been receiving much attention in the last decade [1], [16], [4]. They scale relatively well to high-dimensional states and controls. For continuous controls $\mathbf{u} \in \mathbb{R}^F$, indirect policy search methods estimate the gradient $\partial J(\boldsymbol{\theta})/\partial \boldsymbol{\theta}$ of the expected long-term cost defined in Eq. (1) w.r.t. the policy parameters $\boldsymbol{\theta}$. Then, the policy parameters are updated using this gradient information.

The policy gradients can be estimated using finite difference methods of the long-term cost $J(\boldsymbol{\theta})$ based on trajectory sampling, which is exact in the limit [16]. Alternatively, $J(\boldsymbol{\theta})$ can be approximated, but the corresponding policy gradients can be computed analytically [4]. The latter approach often allows for a large number of policy parameters $\boldsymbol{\theta}$, and, therefore, more flexible controllers.

In this paper, we use the PILCO (probabilistic inference for learning control) framework [4], [5] for learning to control biologically inspired walkers. PILCO is a model-based policy search method for learning optimal state-feedback controllers. PILCO is data efficient and can learn from scratch. Hence, expert knowledge, e.g., in the form of demonstrations, is not necessary. Fig. 2 sketches the main steps of the algorithm. The policy parameters are initialized randomly, and the initial small data set for learning the dynamics model is generated by applying random actions.

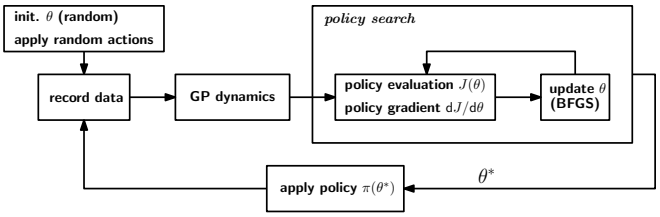


Fig. 2: The main steps of the PILCO policy search framework. Initially, the policy parameters $\boldsymbol{\theta}$ and a small data set are created randomly. After each policy search, the learned policy with parameters $\boldsymbol{\theta}^*$ is applied to the robot and the resulting trajectory is used to update the GP dynamics model.

Since PILCO can learn from little data, it is a promising RL approach to learn controllers in robotics. Typically, RL is not directly applicable in a robotics context if no informative prior knowledge is available: Expensive exploration with the robot can seriously damage the robot hardware.

PILCO's learning speed is largely due to its robustness to model errors during long-term planning and policy evaluation. This robustness stems from the use of a probabilistic forward dynamics model that explicitly describes model uncertainties. Since PILCO explicitly averages over these uncertainties, its performance is not usually degraded seriously by model errors.

B. Learning a Locomotion Dynamics Model

PILCO's probabilistic dynamics model is implemented as a non-parametric Gaussian process (GP) [19]. In the context of regression, a GP is a prior over an unknown function f, where for given inputs $\mathbf{x} \in \mathbb{R}^D$, function values $y_i = f(\mathbf{x}_i) + \epsilon$, $i = 1, \ldots, n$, $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$, are observed. A GP is fully specified by a mean function m and a covariance function k, also called a *kernel*. We consider a GP prior mean function of $m(\cdot) \equiv 0$ and the sum of a Gaussian kernel with automatic relevance determination and a noise kernel, i.e.,

$$k(\tilde{\mathbf{x}}_p, \tilde{\mathbf{x}}_q) = \sigma_f^2 \exp\big(-\tfrac{1}{2}\|\tilde{\mathbf{x}}_p - \tilde{\mathbf{x}}_q\|^2_{\mathbf{\Lambda}^{-1}}\big) + \sigma_w^2 \delta_{pq} \qquad (2)$$

where $\tilde{\mathbf{x}} := [\mathbf{x}^\top \mathbf{u}^\top]^\top$ is the control-augmented state. In Eq. (2), we define σ_f^2 as the variance of the latent function f and $\mathbf{\Lambda} := \operatorname{diag}([\ell_1^2, \ldots, \ell_D^2])$, which depends on the characteristic length-scales ℓ_i. There are n training inputs $\tilde{\mathbf{X}} = [\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_n]$ and corresponding training targets $\mathbf{y} = [\Delta_1, \ldots, \Delta_n]^\top$ with $\Delta_i = \mathbf{x}_{i+1} - \mathbf{x}_i$.

Due to its non-parametric form, the GP is flexible, i.e., it is not required to specify a parametric form of f in advance, which is often based on idealized modeling assumptions. Moreover, a GP possesses only a few hyper-parameters that are automatically learned from the data using evidence maximization [19]. As a result of evidence maximization, a GP does not tend to overfit since it automatically obeys Occam's razor. Hence, using a non-parametric GP is promising for modeling the dynamics of a biologically inspired biped, such as shown in Fig. 1: Compliance due to the presence of muscles and delays are hard to parametrize.

PILCO's probabilistic dynamics model is implemented as a GP, where we use tuples $\tilde{\mathbf{x}}_{t-1} = (\mathbf{x}_{t-1}, \mathbf{u}_{t-1}) \in \mathbb{R}^{D+F}$ as training inputs and state differences $\Delta_t = \mathbf{x}_t - \mathbf{x}_{t-1} + \mathbf{w} \in$

$\mathbb{R}^D$, $\mathbf{w} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_w)$, $\boldsymbol{\Sigma}_w = \mathrm{diag}([\sigma_{w_1}^2, \ldots, \sigma_{w_D}^2])$, as training targets. For given $\tilde{\mathbf{x}}_{t-1}$ the successor state distribution is $p(\mathbf{x}_t|\tilde{\mathbf{x}}_{t-1}) = \mathcal{N}(\mathbf{x}_t \,|\, \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$, where

$$\boldsymbol{\mu}_t = \boldsymbol{\mu}_{t-1} + \mathbb{E}_f[\Delta_t]\,, \quad \boldsymbol{\Sigma}_t = \mathrm{var}_f[\Delta_t]\,. \tag{3}$$

With $\boldsymbol{\beta} = \mathbf{K}^{-1}\mathbf{y}$, the predictive distribution $p(\Delta_*|\tilde{\mathbf{x}}_*)$ at a test input $\tilde{\mathbf{x}}_*$ is Gaussian with mean and variance

$$m_f(\tilde{\mathbf{x}}_*) = \mathbb{E}_f[\Delta_*] = \mathbf{k}_*^\top \mathbf{K}^{-1}\mathbf{y} = \mathbf{k}_*^\top \boldsymbol{\beta}\,, \tag{4}$$

$$\sigma_f^2(\Delta_*) = \mathrm{var}_f[\Delta_*] = k_{**} - \mathbf{k}_*^\top \mathbf{K}^{-1}\mathbf{k}_*\,, \tag{5}$$

respectively, where $\mathbf{k}_* \coloneqq k(\tilde{\mathbf{X}}, \tilde{\mathbf{x}}_*)$, $k_{**} \coloneqq k(\tilde{\mathbf{x}}_*, \tilde{\mathbf{x}}_*)$, and the entries of $\mathbf{K}$ are $K_{ij} = k(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j)$, see Eq. (2).

Multivariate targets are treated as independent. Thus, a separate GP is trained for each target dimension. This model implies that the predictive state dimensions do not covary as long as the test input is deterministically given.

C. Approximate Inference for Long-Term Planning

Following the description of the locomotion dynamics model, we now detail how PILCO performs policy search to learn a state-feedback controller for the locomotive system.

The PILCO policy search framework computes analytic policy gradients $\partial\tilde{J}(\boldsymbol{\theta})/\partial\boldsymbol{\theta}$ of an approximation $\tilde{J}(\boldsymbol{\theta})$ to the expected long-term cost $J(\boldsymbol{\theta})$ in Eq. (1), requiring approximate long-term predictions $p(\mathbf{x}_1), \ldots, p(\mathbf{x}_T)$. We use a moment-matching approach that iteratively approximates

$$p(\mathbf{x}_{t+1}) = \iiint p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t) p(\mathbf{x}_t, \mathbf{u}_t)\, \mathrm{d}f\, \mathrm{d}\mathbf{x}_t\, \mathrm{d}\mathbf{u}_t \tag{6}$$

by a Gaussian with the exact mean $\boldsymbol{\mu}_{t+1}$ and the exact covariance $\boldsymbol{\Sigma}_{t+1}$. The integration in Eq. (6) is analytically intractable: The conditional distribution $p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t)$ is given by the GP predictive distribution in Eq.(3). The use of the nonlinear Gaussian covariance function in Eq. (2) makes determining $p(\mathbf{x}_{t+1})$ in Eq. (6) computationally intractable. As an alternative to moment matching, approximations based on linearization are conceivable [11]. Although they are computationally more advantageous, their approximation performance can suffer from their inappropriate treatment of model uncertainties. An extreme example of the resulting underestimation of the predictive variance is given in Fig. 3. For mathematical details on how to compute the predictive means and covariance matrices, we refer to [4], [5].

Having determined Gaussian approximations to the predictive state distributions $p(\mathbf{x}_1), \ldots, p(\mathbf{x}_T)$, an approximation $\tilde{J}(\boldsymbol{\theta})$ to the expected long-term cost $J(\boldsymbol{\theta})$ in Eq. (1) can be computed by summing up the expected immediate costs $\mathbb{E}_{\mathbf{x}_t}[c(\mathbf{x}_t)] = \int c(\mathbf{x}_t)\mathcal{N}(\mathbf{x}_t \,|\, \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)\, \mathrm{d}\mathbf{x}_t$, where we assume that these integrals can be computed analytically.

To perform a gradient-based policy search, the gradients $\partial\tilde{J}(\boldsymbol{\theta})/\partial\boldsymbol{\theta}$ of the approximated expected long-term cost with respect to the policy parameters $\boldsymbol{\theta}$ are computed *analytically*: Approximate inference to determine $p(\mathbf{x}_t)$, $t = 1, \ldots, T$ in Eq. (6), is performed analytically, i.e., neither sampling nor numerical integration is required [4], [5].

By explicitly averaging out the posterior uncertainty about the latent locomotion dynamics f, our approximate inference method and, thus, the policy search, is robust to model errors. PILCO's increased robustness leads to data efficient RL from scratch that can be applied to robotic tasks where executing many trials is infeasible [5]. Hence, our policy search framework is promising for learning locomotion models and controllers.

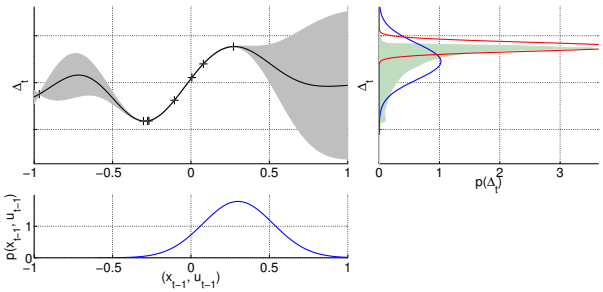


Fig. 3: Predictive distributions based on moment matching (blue) and linearization (red). The black markers denote the training targets, the black solid line is the posterior GP mean function. The Gaussian distribution in the bottom panel is the test input distribution, the shaded distribution in the right-hand side panel is the true predictive distribution. Using linearization for approximate inference can lead to predictive distributions that are too tight.

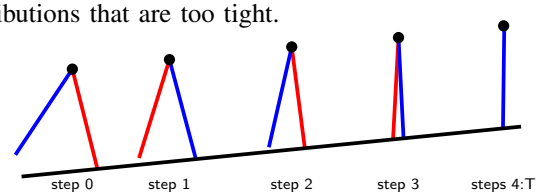


Fig. 4: Simulated compass gait walker on an inclining slope. The left leg is shown in blue, the right leg is shown in red. When solely penalizing falling, the walker learned to stop and to stand upright. When additionally penalizing small hip velocities, the walker learned to climb the slope.

The moment-matching approximation of the predictive distributions, see Eq. (6), captures the variability in the gait. Hence, by iteratively computing predictive state distributions $p(\mathbf{x}_{t+1})$, $t = 0, \ldots, T$, we obtain a distribution over predicted state trajectories that can be encountered when applying a control policy. Ideally, the corresponding trajectories can be controlled with a feedback policy into a limit cycle behavior. Our approach makes it feasible to actually learn these feedback controllers. Hence, we can learn locomotion state-feedback controllers that are robust to variability in the trajectory, which might occur due to, for example, noise, model errors, or gait variability.

III. Experimental Results

In the following, we demonstrate that our policy search approach quickly learns gait policies for a compass gait walker [3]. Furthermore, we discuss modeling and dimensionality reduction aspects using data from the biomechanically inspired biped shown in Fig. 1.

A. Compass Gait Walker on an Inclining Slope

We considered learning to control the under-actuated compass gait walker presented in [3]. The compass gait walker

is an idealized mathematical model of two rod-like legs joined by a hinge hip and with masses at the joint and the center of each leg, respectively, see Fig. 4 for a sketch. The hip can exert a torque of $[-20, 20]$ Nm. The swing-leg dynamics of the compass gait model are given as [9] $\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \mathbf{B}\mathbf{u}$, where $\mathbf{M}$ is the mass matrix, $\mathbf{C}$ subsumes the centripetal and Coriolis forces, and $\mathbf{G}$ is the gravity matrix. The contact point of the stance leg with the ground is a zero-torque, frictionless joint. As in [3], ground collisions are assumed instantaneous and perfectly inelastic. For the legs, we used identical masses of $5\,\mathrm{kg}$ and lengths of $1\,\mathrm{m}$, respectively. The hip mass was set to $10\,\mathrm{kg}$. Generally, changing these parameters does not substantially change the learning results as the locomotion forward model is purely data driven. For details about the dynamics and contact modeling, we refer to [3].

The objective was to automatically learn a controller that allowed the walker to walk uphill on an inclining slope ($1°$), as shown in Fig. 4. Unlike [3], we did not assume that the environment or the dynamics of the walker were known. Instead, the walking behavior in the environment was acquired from experimental data only.

The initial state distribution was Gaussian with mean $\boldsymbol{\mu}_0 = [q_{l_0}, q_{r_0}, \dot{q}_{l_0}, \dot{q}_{r_0}]^\top = [-0.36, 0.23, 0, -0.8]^\top$, where q_l, q_r are the angles of the left and right legs with respect to the y-coordinate, respectively. Initially, q_l is the stance leg and q_r is the swing leg. The speed $\dot{q}_r$ in the mean of $p(\mathbf{x}_0)$ corresponds to the swing leg swinging forward toward the incline. The initial covariance matrix $\boldsymbol{\Sigma}_0$ was set to $10^{-4}\mathbf{I}$.

The feedback controller was implemented as a radial-basis-function network with 50 basis functions, such that $\mathbf{u} = \pi(\mathbf{x}, \boldsymbol{\theta}) = \sum_{i=1}^{50} w_i \Phi_i(\mathbf{x})$, where Φ_i were axes-aligned Gaussian-shaped basis functions located at $\boldsymbol{\mu}_i$ and shared widths $\mathbf{W}$. The policy parameters are the weights w_i, the locations $\boldsymbol{\mu}_i$ of the basis functions, and the diagonal elements of $\mathbf{W}$, resulting in 305 parameters.

We chose an immediate cost function c that penalized the distance of the y-coordinate of the hip from fully upright and small hip velocities in x-direction to encourage walking ahead. The learned controller successfully walked the simulated robot uphill. Solely penalizing the hip's y-coordinate resulted in a learned controller that took energy out of the system and the walker stopped in an upright position, illustrated in Fig. 4.

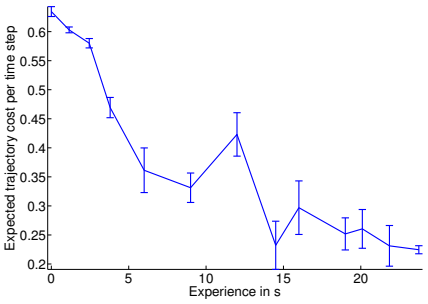
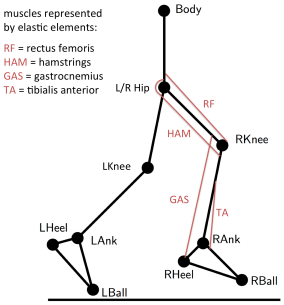


Fig. 5: Learning curve. Expected trajectory cost per time step with twice the standard error of the mean. The horizontal axis shows the total amount of data available to the learner. 0 seconds of experience correspond to the result of the random initialization, see Fig. 2.

(a) JenaWalker II.



(b) Simulated JenaWalker II.

Fig. 6: (a) The high-dimensional biomechanically inspired JenaWalker II is controlled by adjusting muscle stiffness [23]. (b) 23-dimensional simulated JenaWalker II.

Learning was initialized by applying random torques to the hip and sampling the policy parameters from a standard normal distribution. The learned GP locomotion model mapped pairs $(\mathbf{x}_t, \mathbf{u}_t) \in \mathbb{R}^5$ to $\mathbf{x}_{t+1} - \mathbf{x}_t \in \mathbb{R}^4$. Policy search was performed according to Sec. II-C. After convergence, the learned controller was applied to the simulated walker. The additional data from this trial was used to update the GP locomotion model and to re-learn the policy parameters. After a few iterations in PILCO (see Fig. 2), using less than $30\,\mathrm{s}$ of total data, PILCO successfully learned a feedback controller walked compass gait walker uphill.

Fig. 5 shows the corresponding learning curves: After about $15\,\mathrm{s}$ of data, the average step cost per trajectory was close to the optimum for the learned controller, which corresponds to a relatively fast walk up the inclined slope.

B. Nonparametric Dynamics Model of a Biomechanically Inspired Biped

Fig. 6a shows the JenaWalker II, a biomechanically inspired biped that is used to investigate human walking and running [23]. The biped's gaits are controlled by adjusting the muscle stiffness. Here, the muscles of a human leg are realized as elastic structures spanning the joints, see Fig. 6b. A limitation of our biped, compared to human walking, is that 3D walking is impossible as lateral stability and pitch stability of the trunk are not provided. The robot would need to be able to adjust the legs vertically and laterally [17].

In our experiments, we used a realistic simulator of the JenaWalker II. The state of the simulated 7-segment biped is 23-dimensional and consists of its speed (1D) and the (x, y)-coordinates of the body (2D) and the six inner joints (hip, knee, ankle, heel, ball) for each leg ($2\times$10D), see Fig. 6b. The controls are two torques that can be applied at the hip to either of the legs.

In the following, we present results on modeling the dynamics of the biped from the corresponding simulator and approaches to dimensionality reduction.

1) Learning High-Dimensional Locomotion Models: For modeling purposes, we collected data from a time series of $7.5\,\mathrm{s}$, where the simulated robot started in a standing configuration and then transitioned into a running gait. The biped's state was measured at approximately $3.2\,\mathrm{kHz}$.

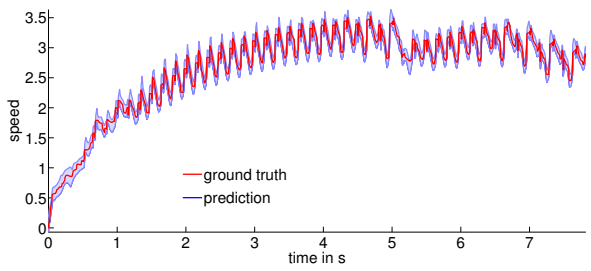


Fig. 7: Even with only 450 data points in a 25-dimensional input space, the learned GP locomotion model can accurately predict all 23 dimensions of the state variable. The 95% confidence bounds of the predictive speed are shown in blue, together with the ground truth (red).

For training the GP dynamics models, we (uniformly) randomly selected $n = 450$ states $\mathbf{x}_i$ and torques $\mathbf{u}_i$ from the time series. The training targets were the corresponding state variables $1/60\,\mathrm{s}$ later, i.e., we subsampled the data with a physiologically more realistic frequency of $60\,\mathrm{Hz}$ [12]. The test set was chosen to be equally spaced along the time axis from $0\,\mathrm{s}$ to $7.5\,\mathrm{s}$. Fig. 7 gives an example that the learned dynamics model can predict the state variables very accurately, although the learned GP locomotion mapping $f : \mathbb{R}^{25} \to \mathbb{R}^{23}$ was learned using 450 data points only. We show the most difficult prediction, the speed of the biped, in Fig. 7. It is easy to see that the robot was initially standing still, then sped up, until it finally ended up in a running gait.

2) Dimensionality Reduction: Policy search, as described in Sec. II-C, can be computationally demanding in high-dimensional spaces. Hence, for learning a feedback controller, a lower-dimensional representation of the 23-dimensional state is advantageous. Lower-dimensional embeddings using, e.g., factor analysis (FA) or (kernel) probabilistic component analysis (PCA) [2] are conceivable; unfortunately, the physiological interpretability of either the lower-dimensional feature space or the data space—both of which we want to maintain—gets lost. Thus, we propose finding a generative mapping from the low-dimensional feature space $\mathcal{F}$ into the original 23-dimensional state space that uses as features a subset of the original state dimensions. Hence, a vector in $\mathcal{F}$ has the same meaning as the corresponding state variables in $\mathbb{R}^{23}$, e.g., "speed".

To find low-dimensional features for predicting the high-dimensional states, we correlated the GP training inputs and the GP training targets. In the statistics literature, this approach is known as *Sure Independence Screening (SIS)* [6]. SIS is very fast and applicable to ultra-high dimensional spaces. Furthermore, it possesses theoretical properties, such as oracle properties and accuracy bounds [6]. In our context, SIS computes the correlation matrix between the GP training inputs and targets. The idea is to select those input dimensions of the state that are strongly correlated with the target dimensions. If there is a strong correlation between input dimension d and target dimension e, the GP transfers knowledge from the input to the target, see Eqs. (4)–(5).

Fig. 8 displays the correlations between the GP training

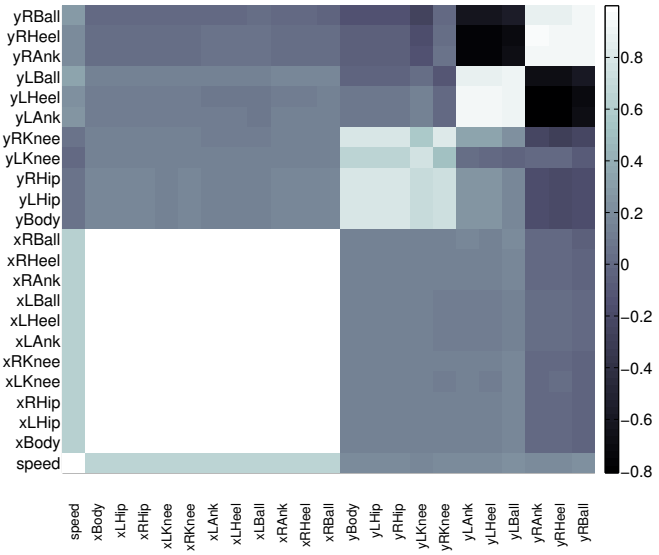


Fig. 8: Correlation between GP training inputs and training targets. Light colors indicate strong positive correlation, dark colors strong negative correlation, respectively. Weak correlations are shown in gray.

inputs and targets. Clear positive correlations can be seen amongst all x-coordinates since the biped moves along the x-axis. In the upper right corner, the y-coordinates of the biped's left foot (ankle, heel, ball) are strongly negatively correlated with their right leg's counterparts. The biped's upper joints' (hip and knee) y-coordinates are positively correlated amongst themselves, independent of whether the left or right leg are considered. This makes sense, since the upper body's movements are fairly restricted. The x and y-coordinates are essentially uncorrelated.

From Fig. 8, we can already guess that only about half the coordinates are relevant for reasonably accurate predictions. This intuition was confirmed when we conducted the following experiment, which can be considered a data preprocessing step: First, we trained a full GP model for the locomotion dynamics mapping from 25-dimensional inputs (23D state plus 2D controls) to 23-dimensional targets. In order to predict *all* 23 dimensions, we gave the GP models a budget of a d-dimensional feature space $\mathcal{F}$. For our analysis, we varied $d = 1, \ldots, 23$. The 2D controls were augmented for learning the mapping. For each target dimension $e = 1, \ldots, 23$, we greedily selected the d most relevant state dimensions in the training set according to the maximum absolute correlation between the input dimensions and the eth target dimension (SIS). All other input dimensions were discarded. At test time, we predicted all 23 target dimensions using only the information from the lower-dimensional feature space. To evaluate the predictive performance, we computed the negative log-likelihood (NLL) values

$$-\ln p(\mathbf{y}_*|\mathbf{m}_*, \mathbf{\Sigma}_*) = \tfrac{1}{2}(D\ln(2\pi) + \ln|\mathbf{\Sigma}_*| + \|\mathbf{m}_* - \mathbf{y}_*\|^2_{\mathbf{\Sigma}_*^{-1}})$$

for $D = 23$, where $\mathbf{m}_*$ and $\mathbf{\Sigma}_*$ are the mean and covariance of the GP prediction at the test input $\mathbf{x}_* \in \mathcal{F}$, respectively. The test target is denoted by $\mathbf{y}_* \in \mathbb{R}^{23}$. The NLL performance measure penalizes incoherence of the predictive

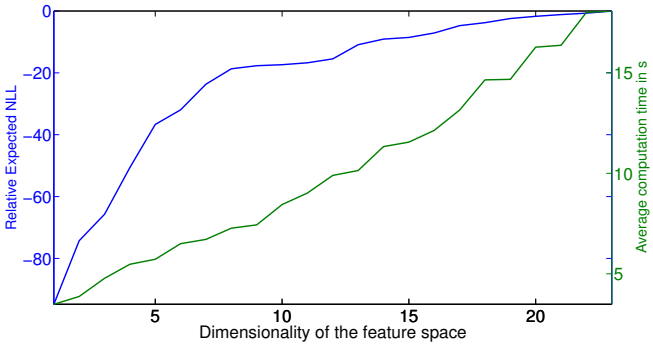


Fig. 9: Expected relative predictive NLL w.r.t. the model using all features (blue) and expected computation time (green) as functions of the feature space dimensionality. The left vertical axis shows the average negative log-likelihoods when predicting the full 23-dimensional stat at 1000 test inputs. Higher values are better. The right vertical axis shows the average computation time for the moment-matching approximation and the policy gradients for a single time step.

distribution: deviations of the predictive mean from the observed values (weighted by the inverse covariance) and predictive uncertainty (determinant of the covariance). Lower NLL values are better. The results shown in Fig. 9 suggest that, based on SIS, we can accurately model the intrinsic dynamics of the biped with an 8-dimensional state space (blue curve): The relative loss of a low-dimensional feature space with respect to the full model (23D) flattens out from $|\mathcal{F}| = 8$ onward. Fig. 9 also shows the duration for computing the moment-matching predictions and policy gradients (see Sec. II-C) for a single time slice as a function of the dimensionality of the embedded space, i.e., the number of features used for predictions (green curve). It might be possible to embed the high-dimensional state into even lower dimensions using other dimensionality reduction techniques, such as FA or PCA. However, with either of these approaches we lose the interpretability of the data.

IV. Conclusion

We have presented a novel and very promising approach to learning models and state-feedback controllers for legged locomotion. Our controller is learned fully automatically from very small data sets and can even learn from scratch, with no initial model information given. Unlike other approaches to learning locomotion, our successful feedback controller is based on predictive distributions over limit cycles. Hence, our approach allows for learning state-feedback controllers for locomotion that are robust to variations in the trajectory. Such variability can be a result of real-world considerations, such as noise, model errors, or gait variability, for instance.

Additionally, we demonstrated that we can learn very good predictive models for a high-dimensional (23D) biomechanically inspired biped. Based on Sure Independence Screening, we automatically identified low-dimensional features of the state that preserve the interpretability of both the low-dimensional space and the high-dimensional data space.

Setting up a cost function for learning gaits is a non-trivial task. We saw in Sec. III that a poor cost function can lead to undesired effects. In the future, we will investigate whether we can learn a good cost function from human data using inverse RL techniques.

References

[1] J. A. Bagnell and J. G. Schneider. Autonomous Helicopter Control using Reinforcement Learning Policy Search Methods. In *ICRA*, 2001.

[2] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag, 2006.

[3] K. Byl and R. Tedrake. Approximate Optimal Control of the Compass Gait on Rough Terrain. In *ICRA*, 2008.

[4] M. P. Deisenroth and C. E. Rasmussen. PILCO: A Model-Based and Data-Efficient Approach to Policy Search. In *ICML*, 2011.

[5] M. P. Deisenroth, C. E. Rasmussen, and D. Fox. Learning to Control a Low-Cost Manipulator using Data-Efficient Reinforcement Learning. In *RSS*, 2011.

[6] J. Fan and J. Lv. Sure Independence Screening for Ultrahigh Dimensional Feature Space. *J. of the Roy. Stat. Soc.*, 70(5):849–911, 2008.

[7] H. Geyer and H. M. Herr. A Muscle-Reflex Model that Encodes Principles of Legged Mechanics Produces Human Walking Dynamics and Muscle Activities. *IEEE Trans. on Neur. Systm and Rehab. Eng.*, 18(3):263–273, 2010.

[8] X. Guo, L. Chen, Y. Zhang, P. Yang, and L. Zhang. A Study on Control Mechanism of above Knee Robotic Prosthesis based on CPG Model. In *RoBio*, pp. 283–287, 2010.

[9] F. Iida and R. Tedrake. Minimalistic Control of Biped walking in Rough Terrain. *Aut. Rob.*, 28(3):355–368, 2010.

[10] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa. The 3D Linear Inverted Pendulum Mode: A Simple Modeling for a Biped Walking Pattern Generation. In *IROS*, 2001.

[11] J. Ko and D. Fox. GP-BayesFilters: Bayesian Filtering using Gaussian Process Prediction and Observation Models. In *IROS*, 2008.

[12] D. Kraus. *Concepts in Modern Biology*. Globe Book Company, 1993.

[13] T. McGeer. Passive Dynamic Walking. *IJRR*, 9(2):62–82, 1990.

[14] J. Morimoto and C. G. Atkeson. Learning Biped Locomotion. *IEEE Robotics Automation Magazine*, 14(2):41–51, 2007.

[15] J. Morimoto, G. Zeglin, and C. G. Atkeson. Minimax Differential Dynamic Programming: Application to a Biped Walking Robot. In *IROS*, 2003.

[16] J. Peters and S. Schaal. Policy Gradient Methods for Robotics. In *IROS*, 2006.

[17] F. Peuker, C. Maufroy, and A. Seyfarth. Leg-Adjustment Strategies for Stable Running in Three Dimensions. *Bioinspiration & Biomimetics*, 7(3), 2012.

[18] J. E. Pratt and G. A. Pratt. Intuitive Control of a Planar Bipedal Walking Robot. In *ICRA*, 1998.

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

[20] N. Ratliff, J. A. Bagnell, and S. S. Srinivasa. Imitation Learning for Locomotion and Manipulation. In *Humanoids*, 2007.

[21] P. Saltiel, K. Wyler-Duda, A. D'Avella, M. C. Tresch, and E. Bizzi. Muscle Synergies Encoded Within the Spinal Cord: Evidence From Focal Intraspinal NMDA Iontophoresis in the Frog. *J. of Neurophys.*, 85(2):605–619, 2001.

[22] M. Sato, Y. Nakamura, and S. Ishii. Reinforcement Learning for Biped Locomotion. In *ICANN*, 2002.

[23] J. A. Smith and A. Seyfarth. *Autonome Mobile Systeme*, chapter Exploring Toe Walking in a Bipedal Robot. Springer-Verlag, 2007.

[24] R. Tedrake, T. Zhang, and H. Seung. Stochastic Policy Gradient Reinforcement Learning on a Simple 3D Biped. In *IROS*, 2004.

[25] M. Vukobratovic and B. Borovac. Zero-Moment Point—Thirty Five Years of its Life. *IJHR*, 1(1):157–173, 2004.

[26] D. M. Wolpert, R. C. Miall, and M. Kawato. Internal Models in the Cerebellum. *Trends in Cognitive Sciences*, 2(9):338–347, 1998.

[27] S. L.-Y. Woo, R. E. Debski, J. D. Withrow, and M. A. Janaushek. Biomechanics of Knee Ligaments. *Am. J. on Sports Medicine*, 27(4):533–543, 1999.

[28] F. E. Zajac, R. R. Neptune, and S. A. Kautz. Biomechanics and Muscle Coordination of Human Walking: Part I: Introduction to Concepts, Power Transfer, Dynamics and Simulations. *Gait & Posture*, 16(3):215–232, 2002.