Model-Free Reinforcement Learning of Impedance Control in Stochastic Environments

Freek Stulp, Jonas Buchli, Alice Ellmer, Michael Mistry, Evangelos A. Theodorou, and Stefan Schaal

***Abstract*—For humans and robots, variable impedance control is an essential component for ensuring robust and safe physical interaction with the environment. Humans learn to adapt their impedance to specific tasks and environments; a capability which we continually develop and improve until we are well into our twenties. In this article, we reproduce functionally interesting aspects of learning impedance control in humans on a simulated robot platform. As demonstrated in numerous force field tasks, humans combine two strategies to adapt their impedance to perturbations, thereby minimizing position error and energy consumption: 1) if perturbations are unpredictable, subjects increase their impedance through cocontraction; and 2) if perturbations are predictable, subjects learn a feed-forward command to offset the perturbation. We show how a 7-DOF simulated robot demonstrates similar behavior with our model-free reinforcement learning algorithm $\mathbf{PI^2}$, by applying deterministic and stochastic force fields to the robot's end-effector. We show the qualitative similarity between the robot and human movements. Our results provide a biologically plausible approach to learning appropriate impedances purely from experience, without requiring a model of either body or environment dynamics. Not requiring models also facilitates autonomous development for robots, as prespecified models cannot be provided for each environment a robot might encounter.**

***Index Terms*—Force field experiments, motion primitives, motor system and development, reinforcement learning, robots with development and learning skills, using robots to study development and learning, variable impedance control.**

I. Introduction

TO ACHIEVE robustness towards stochastic disturbances, humans adapt the impedance of their biomechanical system. The ability to adapt impedance is developed "*over the course of years [as] the stability of particular behaviors is reinforced and is more robust to perturbations*." [3]. On a smaller time scale, humans are also able to learn task-specific adaptations to achieve robustness against the specific perturbations arising in a particular task. In windsurfing or skiing for instance, we must be compliant enough to deal with small, high frequency perturbations caused by small waves or uneven snow, but still have a high enough impedance to be able to control the board or skis. Learning this balance is one of the aspects that make these sports difficult to master. In summary, impedance control in humans is the result of learning and development; the aim of this article is to reproduce functionally interesting aspects of this learning process on a simulated robot platform.

The standard paradigm for studying task-specific perturbation rejection in humans is by applying a force field to a subject's hands with a robotic manipulandum. Numerous force field experiments have demonstrated that humans develop two strategies to deal with perturbations [2], [4], [14], [17]–[19]. Deterministic, and thus predictable, perturbations are countered by learning a feed-forward term, whereas stochastic perturbations lead to an increase in stiffness through muscle cocontraction.

The general rule of thumb thus seems to be "be compliant when possible; stiffen up only when the task requires it." Task-specific adaptation of impedance allows humans to combine the advantages of high stiffness (accurate tracking, stable under unforeseen perturbations) and compliance (lower energy consumption, safer interaction with the environment, decoupling from perturbations). It takes humans more than two decades of experience to develop and tune this rule [3].

In contrast, robots have traditionally been controlled with constant high gain negative error feedback control [20]. Especially for industrial robots, the rule of thumb is rather "be stiff". Achieving high position accuracy has thus come at the cost of high energy consumption, and the necessity to build cages

Manuscript received November 08, 2011; revised February 21, 2012; accepted June 16, 2012. Date of publication June 28, 2012; date of current version December 07, 2012. This research was supported in part by National Science Foundation under Grants ECS-0326095, IIS-0535282, IIS-1017134, CNS-0619937, IIS-0917318, CBET-0922784, EECS-0926052, CNS-0960061, the DARPA program on Advanced Robotic Manipulation, the DARPA program on Learning Locomotion, the Okawa Foundation, the ATR Computational Neuroscience Laboratories, the Max-Planck-Society, EC Grant no. FP7-ICT-215181 CogX, and the MACSi project ANR-2010-BLAN-0216-03. F.S. was supported by a Research Fellowship and a Return Grant from the German Research Foundation (DFG). J.B. was supported by an advanced researcher fellowship from the Swiss National Science Foundation. E.T. was supported by a Myronis Fellowship.

F. Stulp was with the Computational Learning and Motor Control Laboratory, University of Southern California, Los Angeles, CA 90089 USA, and also with the Cognitive Robotics Department , École Nationale Supérieure de Techniques Avancées-ParisTech, 75015 Paris, France. He is also with the FLOWERS Research Team, INRIA Bordeaux Sud-Ouest, 33405 Talence, France (e-mail: freek.stulp@ensta-paristech.fr).

J. Buchli is with the Computational Learning and Motor Control Lab, University of Southern California, Los Angeles, CA 90089 USA, and also with the Department of Advanced Robotics, Istituto Italiano di Tecnologia, 16163 Genova, Italy (e-mail: jonas@buchli.org).

A. Ellmer is with the Computational Learning and Motor Control Laboratory, University of Southern California, Los Angeles, CA 90089 USA (e-mail: alice.ellmer@googlemail.com).

M. Mistry is with the School of Computer Science, University of Birmingham, Birmingham B15 2TT, UK (e-mail: mmistry@disneyresearch.com).

E. A. Theodorou is with the Computational Learning and Motor Control Laboratory, University of Southern California, Los Angeles, CA 90089 USA, and also with the Computer Science and Engineering Department, University of Washington, Seattle, USA (e-mail: etheodor@usc.edu).

S. Schaal is with the Computational Learning and Motor Control Laboratory, University of Southern California, Los Angeles, CA 90089 USA, and also with the Max-Planck-Insitute for Intelligent Systems, 72076 Tübingen, Germany (e-mail: sschaal@usc.edu)

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAMD.2012.2205924

around robots to avoid human-robot contact. For autonomous mobile robots operating in human environments, safety and energy efficiency requirements are very different, and low-gain variable impedance control will be an essential characteristic of such robots.

We have recently shown that the PI^2 reinforcement learning algorithm is able to learn such variable impedance control on real robots in high-dimensional tasks [1]. In this paper, as in [22], we use PI^2 to learn variable impedance controllers in deterministic and stochastic force fields. In particular, we show that the robot learns behavioral adaptations similar to those of humans, i.e., the two-fold strategy of combining a feed-forward term for deterministic perturbations with increased impedance for stochastic perturbations.

In contrast to model-based optimal control, which is commonly used to simulate human behavior in computational motor control [11], PI^2 does not require a model and scales to high-dimensional tasks [23], thus making it a biologically more plausible learning algorithm. Not requiring models also facilitates autonomous development for robots, as prespecified models cannot be provided for each environment a robot might encounter. Furthermore, as PI^2 converges to local optimal solutions [27], our results lends support to the idea that learning based on stochastic optimality is a good model for human learning and development of motor skills.

The rest of this article is structured as follows. In the next section, we discuss related work from computational motor control and robotics. We present a analysis of variable impedance control in stochastic force fields in Section III, and use this to make several predictions about learned behaviors for perturbation rejection. In Section IV, we summarize the experimental design and models used in this article, being Dynamic Movement Primitives [8] and the PI^2 reinforcement learning algorithm [27]. The results of our empirical evaluation are presented in Section V. We conclude with Section VI.

II. Related Work

In this section, we discuss related work on biophysics experiments that investigate the role of human stiffness adaptations to perturbations, as well as state-of-the-art in variable impedance control for robots.

A. Variable Impedance Control in Force Field Experiments

The standard paradigm for studying perturbation rejection in humans is by applying a force field to a subject's hands with a robotic manipulandum. With this paradigm, Shadmehr and Mussa–Ivaldi [19] demonstrated that, with practice, humans learn to compensate for external perturbations caused by a *deterministic* force field. When suddenly removing the force field, the observed trajectories (after-effects) were approximately mirror images of the trajectories before learning. This suggests that humans learn an internal model of the force field and compensate for it.

Further experiments demonstrated that in *stochastic* force fields, subjects learn to adapt to the mean disturbance regardless of the statistical distribution of disturbances [17], and that the after-effects decrease with increasing stochasticity [2]. Takahashi *et al.* [2] suggest that the nervous system adopts a dual strategy: learning an internal model of the mean of the random environment, while simultaneously increasing arm stiffness to minimize the consequence of errors. They thus conclude that "*the results of this study suggest that impedance control can coexist with the application of internal models for control*" [2]. The analytical and empirical results presented in this article further support this hypothesis. A similar dual effect was shown by Mistry *et al.* [14]. Here, it was also demonstrated that a small amount of stochasticity in force field strength leads to better learning of an internal model than in the deterministic case. This suggests that some noise is beneficial to exploration.

Franklin *et al.* [4] present a model which combines three principles to learn stable, accurate and efficient movements: 1) positive error (the muscle was stretched more than expected) leads to an increase in feedforward muscle activity; 2) negative error (muscle shortening) leads to a similar increase; and 3) the feedforward activation of a muscle is reduced if the error is below some threshold. The three principles are combined in a simple algorithm, which is able to accurately model empirical data gathered in force field experiments.

B. Variable Impedance Control in Robotics

Since variable impedance control allows a tradeoff between the minimization of error and energy, it is a desirable property for robots as well. We believe that compliant robots which interact safely with the environment (by avoiding high contact forces), are especially relevant to developmental robotics, which emphasizes the role of continual interaction with the environment. When learning continually through trial-and-error interaction, it is important that exploration does not lead to dangerously high forces due to (unexpected) physical contact with the environment—a prerequisite for developmental robotics is that an error may never be so grave that no more trials can be performed by the robot.

However "*[t]he selection of good impedance parameters* [. . .] *is not an easy task*" [20]. The main reason is that deriving useful impedance controllers usually involves models of both the environment and the robot, as well as deep knowledge about designing and parameterizing such controllers [7]. Therefore, recent work in variable impedance control for robotics has investigated learning these models, or using methods that are all together model-free.

Mitrovic *et al.* [15] make a strong case for the limitations of analytical dynamic models. First, the accuracy of the model is limited to the level of detail in the physical model, and the amount of effort put in the system identification process. Second, it is not obvious how changes in the dynamics over time can be modelled. Finally, the dynamic stochasticity, incorporated in the noise model, often depends on the task and physical interactions with the environments. These problems already arise for the 1-dimensional antagonistic actuator Mitrovic *et al.* consider, and only become more severe when considering high-dimensional systems such as humanoid robots [23]. Therefore, Mitrovic *et al.* propose to *learn* a model of the dynamics *and* the noise through supervised learning, using locally weighted projection regression [28]. This model is then used in a model-based stochastic optimal controller to control a 1-DOF antagonistic actuator. Our goal is rather to circumvent

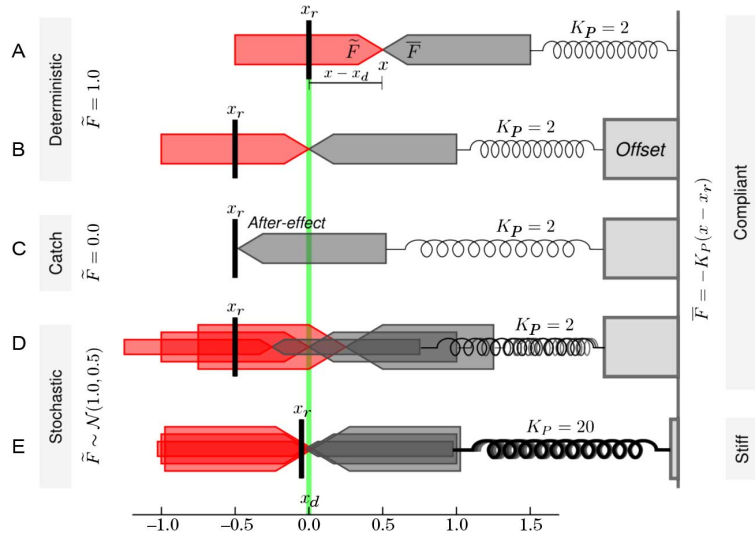


Fig. 1. Illustration of using different impedances (gains) in deterministic and stochastic perturbations. The perturbation pushes from the left; the magnitude of the perturbation force $\widetilde{F}$ is indicated by the width of the left (red) arrow. The 'robot manipulator' is depicted as a (gray) arrow pushing from the right; its force $\overline{F}$ is computed with $\overline{F} = -K_P(x - x_r)$, i.e., a proportional controller based on the position error between the reference position x_r and actual position x. The gain $K_P \in \{2, 20\}$ is represented by the thickness of the spring to the right. The robot's goal is to minimize the error $(x_d - x)$ between the actual (x) and desired (x_d) position. In this illustration, all forces are in equilibrium, i.e., $\overline{F} = -\widetilde{F}$ and $\dot{x} = 0$.

modelling all together, by learning directly in the space of the policy parameters.

An early model-free impedance learning approach was presented in [10], where a simulated 2DOF simulated robot arm with antagonistic muscle learns to reach for a target object whilst minimizing motor commands. The reinforcement learning algorithm is implemented as an actor-critic architecture, where the critic learns the value function by minimizing the temporal-difference error, and the actor determines the muscle forces. Both are implemented as feed-forward neural networks. The learning problem is simplified by generating exploration noise along two subspaces, one in which the joint stiffness remains constant, and another in which it does not. Most relevant to this article is the fact that the agent learns to increase the impedance through cocontraction when the arm is perturbed with a force of randomly varying orientation [9] or strength [10]. However, applying this approach to higher-dimensional systems or real robots might be challenging, as it requires tuning of the time-constant for learning, appropriately setting the bias for the initial stiffness, and determining the appropriate neural network structure. A similar more recent actor-critic approach is presented in [13]. In this work, the reference trajectory of the end-effector is fixed. Therefore this approach cannot be used to simulate the dual strategy seen in humans, as it relies adapting the reference trajectory.

We recently proposed the use of *model-free reinforcement learning* to simultaneously learn reference trajectories *and* variable impedance controllers [1], [22]. To do so, we use the policy improvement with path integrals (PI^2) algorithm; a very general algorithm which scales up to very high-dimensional problems [23]. This approach, which will be described in detail in Section IV, has been applied to simulated and real robots in viapoint tasks and tasks involving physical contact with the environment [1], for instance flipping a light switch or opening a door. This article presents the first results of applying PI^2 to tasks involving stochastic perturbations, and demonstrating how the resulting movements are similar to those observed in humans.

Most closely related to our work is the implementation of the model by Franklin *et al.* [4] (described in the previous section) in the context of a 1-DOF robot [5]. Here, cocontraction of muscles is also modelled by increasing the gains of the robot, and the robot learns to adapt its impedance, the forces it applies and reference trajectories. Our work differs in that PI^2 is a generic policy improvement algorithm, which can not only be used to learn gain schedules, but also for very different tasks, i.e., learning to grasp under uncertainty [25] or pick-and-place manipulation tasks [24]. Also, whereas [5] has been applied to 1-DOF robots, PI^2 has been shown to work on action spaces of very high dimensionality [23].

In summary, we observe that computational motor control models and robotics are converging towards the same solution: adaptively learning impedance control. The quote "*motor adaptation should incorporate . . . two adaptive processes: internal model formation as well as impedance regulation*. [2]" applies equally well to both fields. The motivation behind this article is to demonstrate that model-free reinforcement learning based on stochastic optimality is: 1) an excellent basis for controlling high-dimensional robot systems, as demonstrated in [1]; and 2) able to qualitatively reproduce the two adaptive processes as seen in human behavior.

III. Analysis

In this section, we model the force field and robot as a simple 1-dimensional spring system, depicted in Fig. 1. The aim of this section is two-fold: 1) introduce the concepts used in the experimental design in Section IV; and 2) demonstrate that, despite the simplicity of the static model, the model is able to make predictions that have been observed in dynamic biophysics experiments. This supports the hypothesis that variable gain sched-

uling is an appropriate paradigm for modelling variable stiffness in humans.

A. Deterministic Force Fields

We consider a scenario in which the forces are in equilibrium; the model thus represents a steady state, rather than a movement over time. The 'robot' is simply a spring, which models a resistance to the force field with a proportional controller.[1] Without loss of generality, we assume that the desired position is zero, i.e., $x_d = 0$. In Fig. 1, this corresponds to the tip of the gray arrow coinciding with the green line

$$\overline{F} = -\widetilde{F} \quad \text{Assume equilibrium} \quad (1)$$
$$\overline{F} = -K_P(x - x_r) \quad \text{Proportional controller} \quad (2)$$
$$\widetilde{F} = K_P(x - x_r) \quad \text{Combine 1 and 2} \quad (3)$$
$$x = x_r + \widetilde{F}/K_P \quad \text{Solve for } x. \quad (4)$$

Given this scenario, we can now readily compute the actual position x from the force field strength $\widetilde{F}$ with (4). An example is given in Fig. 1A, where $\widetilde{F} = 1$, $K_P = 2$, $x_r = x_d = 0$, and thus $x = 0 + 1/2 = 0.5$.

Since x represents the deviation from the desired position $x_d = 0$, we would like to keep $x = 0$ to minimize the position error

$$x = 0 \quad \text{Position error should be zero} \quad (5)$$
$$x_r + \widetilde{F}/K_P = 0 \quad \text{From Eq. 4} \quad (6)$$
$$x_r = -\widetilde{F}/K_P \quad \text{Solve for } x_r. \quad (7)$$

For the running example, the offset is $x_r = -1/2 = -0.5$, as depicted in Fig. 1B. Thus, to achieve a zero deviation $x = 0$ in deterministic force fields, $x_r = -\widetilde{F}/K_P$. Learning to move the reference position can be considered the learning of an internal model to counter the force field. However, the force field or the robot are not modeled directly, but rather implicitly in the policy parameter x_r. Note that with the constant offset term, the equilibrium will move to $x = x_r = -0.5$ if the force field is removed ($\widetilde{F} = 0$), as depicted in Fig. 1(C).

Prediction 1: Eq. (7) shows that deterministic force fields can be countered with a constant offset term. This leads to after-effects in catch trials, as observed in [19].

B. Stochastic Force Fields

In stochastic force fields, $\widetilde{F}$ should be rather interpreted as a random variable. The position x is therefore also a random variable, denoted X

$$\widetilde{F} \sim \mathcal{N}(\mu, \sigma^2) \quad \text{Gaussian force field} \quad (8)$$
$$X \sim x_r + \mathcal{N}(\mu, \sigma^2)/K_P \quad \text{Combine (4) and (8)} \quad (9)$$
$$X \sim \mathcal{N}\left(x_r + \mu/K_P, (\sigma/K_P)^2\right) \quad \text{Gauss. transform. rule.} \quad (10)$$

[1]In Section IV, we will rather use a proportional-derivative controller $\overline{F} = -K_P(x - x_r) - K_D(\dot{x} - \dot{x}_r)$ to implement damping. In this article, we assume the reference velocity $\dot{x}_r$ is zero. Since the system is assumed to be in equilibrium in this section, $\dot{x}$ is also zero. Hence, the derivative term drops, and is ignored in this section for simplicity.

As before, we would like to minimize the position error. Since X represents the deviation from the desired position $x_d = 0$, this now corresponds to $\mathbf{E}(X) = 0$

$$\mathbf{E}(X) = 0 \quad \text{As (5)} \quad (11)$$
$$\mathbf{E}\left(\mathcal{N}\left(x_r + \mu/K_P, (\sigma/K_P)^2\right)\right) = 0 \quad \text{From (10)} \quad (12)$$
$$x_r + \mu/K_P = 0 \quad \mathbf{E}\left(\mathcal{N}(a, b)\right) = a \quad (13)$$
$$-\mu/K_P = x_r \quad \text{Solve for } x_r. \quad (14)$$

Thus, to achieve an expected X of zero, $x_r = -\mu/K_P$. This is analogous to (7), except now we use the *expected* value of the force field μ, rather than the *known* value. For our running example we again have $x_r = -1/2 = -0.5$, as depicted in Fig. 1(D).

Prediction 2: Eq. (14) predicts that the offset should be adapted to the mean strength of the force field. This adaptation to the mean was observed in [17].

Now let's plug the offset term $x_r = -\mu/K_P$ back into (10)

$$X \sim \mathcal{N}\left(-\mu/K_P + \mu/K_P, (\sigma/K_P)^2\right) \quad (10) \text{ and } (14) \quad (15)$$
$$X \sim \mathcal{N}\left(0, (\sigma/K_P)^2\right) \quad \text{Simplify.} \quad (16)$$

With the appropriate offset term, the deviation is thus a zero mean Gaussian with variance $(\sigma/K_P)^2$. Therefore, the only way to decrease the variance in the deviation is by increasing the gain K_P. This can be seen when comparing Fig. 1(D) and (E). In Fig. 1(D), the gain is low ($K_P = 2$), and the variance in the position error is thus high ($X \sim \mathcal{N}(0, (0.5/2)^2) \sim \mathcal{N}(0, 0.25^2)$). In D, gains are high ($K_P = 20$), and the variance in the position error is thus low ($X \sim \mathcal{N}(0, (0.5/20)^2) \sim \mathcal{N}(0, 0.025^2)$).

Prediction 3: Eq. (16) predicts that if we want to enforce an upper bound on the variance in the deviation, we are required to increase K_P when the force field stochasticity σ increases. This effect is observed in humans, where impedance is increased when stochasticity of the force field is increased [2].

Prediction 4: Another interesting effect predicted by this analysis is that with an increasing gain K_P, the magnitude of the offset term $x_r = -\mu/K_P$ decreases, and thus the after-effect are predicted to decrease. This is seen in Fig. 1(E), where x_r is much closer to x_d than in Fig. 1(D). In humans it was also observed that increased stochasticity in force fields leads to smaller after effects [14].

Prediction 5: Finally, this analysis predicts that to achieve a prespecified error variance, the gains must increase linearly with increasing stochasticity. This is supported by our empirical results presented in Section V, but partially contradicts observations made in [14]. An possible explanation for this contradiction is given in Section V-A.

IV. Methods

Our robot "subject" is a 7-DOF Barret arm, depicted in Fig. 2. We use an accurate physical simulation of the robot with the SL software package [16]. Note that although the visualizations in this article are in 2D for ease of interpretation, the robot is simulated in full 3-D space.

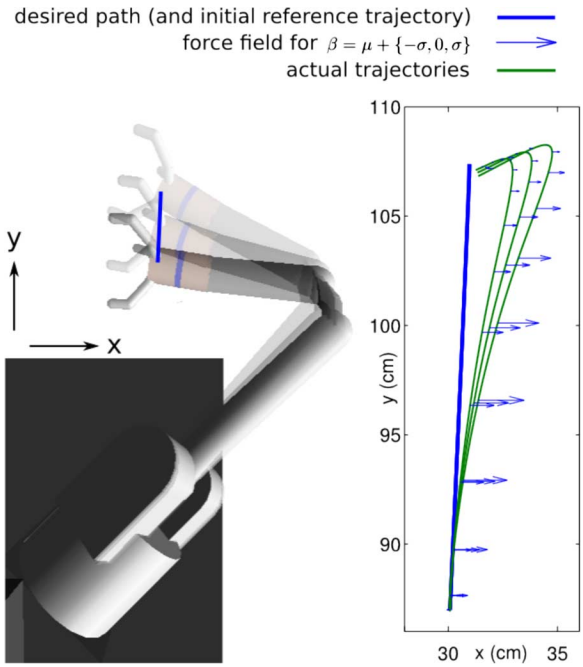


Fig. 2. 7-DOF robotic arm used in this article, simulated in SL. The reaching trajectory and force fields are depicted in the right graph (for $\sigma = 0.2886$).

A. Experimental Protocol

In this paper, we consider the learning of reaching movements to a discrete, specified goal, and follow the experimental protocols in [4], [11], [14]. In particular, the experimental parameters below are taken from Experiment 1 in [14], and Experiment 3 in [11].

Initially the robot makes a straight movement with minimum-jerk velocity profile along the x-axis (distance 0.2 m, duration 1 s), away from its body [11]. This movement is depicted in Fig. 2. We use a velocity dependent force field $\begin{bmatrix} F_x \\ F_y \end{bmatrix} = \beta \begin{bmatrix} 0 & 10 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}$, where F_x, F_y is the force applied to the subject's end-effector along the x/y-axis respectively, and $\dot{x}/\dot{y}$ is the velocity of the end-effector along the x/y-axis [14]. The strength of the force field β is sampled at the beginning of each trial from a Gaussian distribution $\mathcal{N}(1, \sigma)$. We apply four different force fields with $\sigma = \{0.0000, 0.1442, 0.2886, 0.4330\}$ [14]. The effect of the force field with $\sigma = 0.2886$ is depicted in Fig. 2. The deterministic case is represented by $\sigma = 0$

The robot 'subject' receives feedback about joint positions, end-effector position, and joint torques. After each trial, feedback on task achievement is given by the cost function[2]

$$J(\boldsymbol{\tau}_i) = \int_{t_i}^{t_N} 10^3 d(\mathbf{x}_t) + 10^{-2} \sum_{j=1}^{7} \left(K_{P,t}^{j} - K_{P,t}^{j,min} \right) + 10^{-3} |\ddot{\mathbf{x}}_t|. \tag{17}$$

This cost functions consists of the following components:

- **Position error**. $d(\mathbf{x})$ is the distance in meters from the end-effector to the line connecting start and end-point of the movement, i.e., similar to (2) in [4]; this cost expresses that we do not want large errors in position from the straight desired path. Note that this desired path is invariant, and is not the same as the reference trajectory which is adapted through learning (i.e., compare x_r and x_d in Fig. 1). Using such an invariant desired path is also frequently assumed in modelling of human behavior [4], [5], [29].
- **Gains**. $\sum_{j=1}^{7} (K_P^j - K_P^{j,min})$ is the sum of the proportional gains (minus their minimum values) over all joints j; this cost expresses that we prefer low gains, as they lead to lower torque commands and safer human-robot interaction. In principle, we would expect similar results when penalizing motor commands directly, as high gains generally lead to higher motor commands, cf. (25). Thus penalizing motor commands should also lead to lower gains. Penalizing the gains stems from our explicit goal of achieving compliant robots with low-gain control [1].
- **End-effector acceleration**. $|\ddot{\mathbf{x}}|$ is the end-effector acceleration in $\mathrm{m/s^2}$; this cost expresses that we do not want motions with high accelerations.

We now describe how the reference trajectory (see Section IV-B) and variable gain schedules (see Section IV-C) are represented. In Section IV-D, we present the model-free reinforcement algorithm that learns to minimize the cost function in (17), by adapting the reference trajectories and gain schedules.

B. Movement Representation

The reaching trajectory is represented by a Dynamic Movement Primitive (DMP) [8], which consists of a set of dynamic system equations ((18)–(22)) which are visualized, and explained in Fig. 3

Dynamic Movement Primitives

$$\frac{1}{\tau}\ddot{x}_t = f_t + \mathbf{g}_t^T \boldsymbol{\theta} \qquad \text{Transform. system} \tag{18}$$

$$f_t = \alpha \left(\beta (g - x_t) - \dot{x}_t \right) \qquad \text{Linear system} \tag{19}$$

$$[\mathbf{g}_t]_j = \frac{w_j(s_t) \cdot s_t}{\sum_{k=1}^{p} w_k(s_t)} (g - x_0) \qquad \text{Basis functions} \tag{20}$$

$$w_j = \exp\left(-0.5 h_j (s_t - c_j)^2 \right) \qquad \text{Gaussian kernel} \tag{21}$$

$$\frac{1}{\tau}\dot{s}_t = -\alpha s_t \qquad \text{Canonical system} \tag{22}$$

Gain schedules

$$K_{P,t} = \mathbf{g}_{t,K}^T \boldsymbol{\theta}_K \qquad \text{Time-dependent gain} \tag{23}$$

$$[\mathbf{g}_{t,K}]_j = \frac{w_j(s_t)}{\sum_{k=1}^{p} w_k(s_t)} \qquad \text{Basis functions.} \tag{24}$$

The intuition behind this approach is to generate trajectories $[x_t, \dot{x}_t, \ddot{x}_t]$ out of the time evolution of a nonlinear attractor system, where the goal g is a point attractor and x_0 the start state.[3]

The two main components in (18) are a linear (critically damped) spring system f_t, and a nonlinear component consisting of a set of Gaussian basis functions $\mathbf{g}_t^T$, multiplied with the parameter vector $\boldsymbol{\theta}$. The activation of the basis functions

[2]These cost are all part of r_t in the generic PI^2 cost function to be discussed in Section IV-D

[3]Having to start at the same initial condition is a limitation of the DMP. Since force field experiments typically involve repeated movements from the start to the same goal position, this limitation of DMPs coincides with the constraints of the experiments.

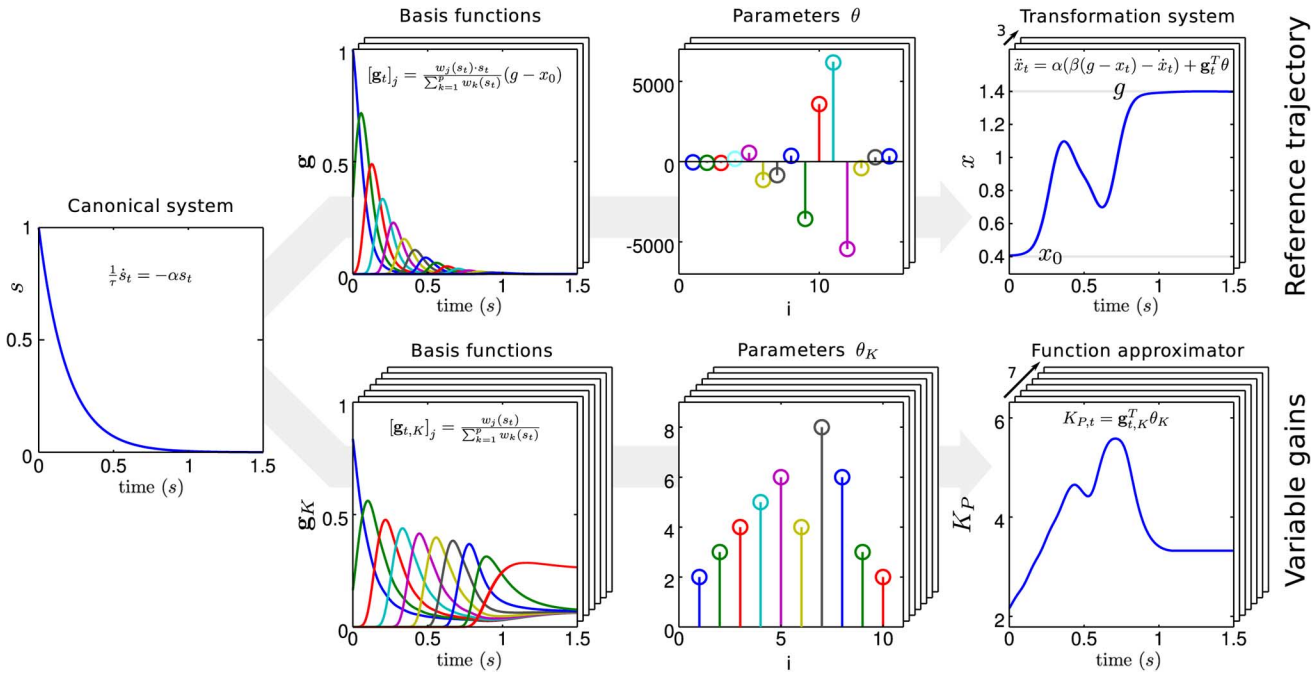


Fig. 3. Dynamic Movement Primitives (DMPs). The core idea behind DMPs is to perturb a simple linear dynamical system with a nonlinear component ($\mathbf{g}_t^T\boldsymbol{\theta}$) to acquire smooth movements of arbitrary shape. The nonlinear component consists of basis functions $\mathbf{g}_t$, multiplied with a parameter vector $\boldsymbol{\theta}$. The canonical system s_t represents the phase variable, which is 1 at the beginning of the movement, and 0 at the end. The movement (upper row) has a duration of 1 second, after which x has reached the goal, i.e., $x_{t>1.0} = g$. Proportional gain schedules $K_{P,t}$ (lower row) are not transformation systems, but rather represented directly with the function approximator $\mathbf{g}_{t,K}^T\boldsymbol{\theta}_K$.

during the movement is determined by the phase variable s_t, which exponentially decays over time from 1 to 0 as the movement progresses, as depicted in the left graph in Fig. 3. The centers c_j of the basis functions (i.e., their maximum activations), are positioned in phase space such that they are equally spaced in time. Finally, the parameters $\boldsymbol{\theta}$ determine the shape of the attractor landscape. For each basis function $[\mathbf{g}_t]_j$, there is one scalar value $\boldsymbol{\theta}_j$. This formulation allows a DMP to represent almost arbitrary smooth trajectories, e.g., a tennis swing, a reaching movement, or a complex dance movement. On the other hand, convergence to the goal g is guaranteed as it can be shown that f_t converges to g, and $\mathbf{g}_t$ converges to 0 [8].

We leave the details to [8], [27]. For this article, the important features of DMPs are that • When integrating a DMP over time, it generates a 1-dimensional output trajectory $[x_t\ \dot{x}_t\ \ddot{x}_t]$ • DMPs converge from the initial value x_0 towards the goal parameter g. • The general shape of the trajectory is determined by the parameters $\boldsymbol{\theta}$ • Multidimensional DMPs are represented by coupling several transformation systems as in (18) with one shared phase variable s.

Parameters for the Experiment: In this article, the DMP has three transformation systems, which represent the 3-dimensional reference trajectory of the robot's end-effector in Cartesian space $[\mathbf{x}_{r,t}\ \dot{\mathbf{x}}_{r,t}\ \ddot{\mathbf{x}}_{r,t}]$. The initial parameters $\boldsymbol{\theta}$ are trained with supervised learning [8], so that the reference trajectory has a minimum-jerk velocity profile, and generates the trajectory depicted in Fig. 2. The reference Cartesian end-effector velocities are converted into joint space using the Jacobian pseudo-inverse. The resulting joint velocities $\dot{\mathbf{q}}_{r,t}$ are integrated and differentiated, to get joint positions $\mathbf{q}_{r,t}$ and accelerations $\ddot{\mathbf{q}}_{r,t}$ respectively.

C. Variable Gain Schedule Representation

Given the reference joint trajectory $[\mathbf{q}_{r,t}\ \dot{\mathbf{q}}_{r,t}\ \ddot{\mathbf{q}}_{r,t}]$ generated above, the motor command torques $\mathbf{u}$ for our robot are calculated via a PD/feed-forward control law

$$\mathbf{u} = -\mathbf{K}_P(\mathbf{q}_t - \mathbf{q}_{r,t}) - \mathbf{K}_D(\dot{\mathbf{q}}_t - \dot{\mathbf{q}}_{r,t}) + \mathbf{u}_{ID} \quad (25)$$

$$\mathbf{K}_D = C\sqrt{\mathbf{K}_P} \quad (26)$$

where $\mathbf{K}_P, \mathbf{K}_D$ are the positive definite position and velocity gain matrices, and C is a constant scale factor set manually for each joint. The feed-forward control term $\mathbf{u}_{ID}$ is computed with an inverse dynamics (ID) controller based on a Newton-Euler algorithm. The inverse dynamics feed-forward torques $\mathbf{u}_{ID}$ only compensate for forces due to gravity, inertia and Coriolis effects, but *not* for the force fields we generate. Therefore, the feed-forward term which is learned to compensate for the force field perturbation $\widetilde{F}$ is completely independent of $\mathbf{u}_{ID}$. Our main motivation for using the ID controller is that humans also use feed-forward control to cancel the dynamics of their arm [29]; the results of the biophysics experiments we model thus also depend on human subjects that use inverse dynamics models. Note that although our learning method is model-free, our inverse dynamics controller *does* require a model of the robot. Although humans learn this model rather than it being model-based, we believe this is not relevant to the task we consider. Second, decoupling the inverse dynamics from the task dynamics would allow the learned motion to generalize better to similar tasks. For instance, the movement could be moved 10 cm to the right, and the same movement would still arise. If the movement would involve both the compensation of the

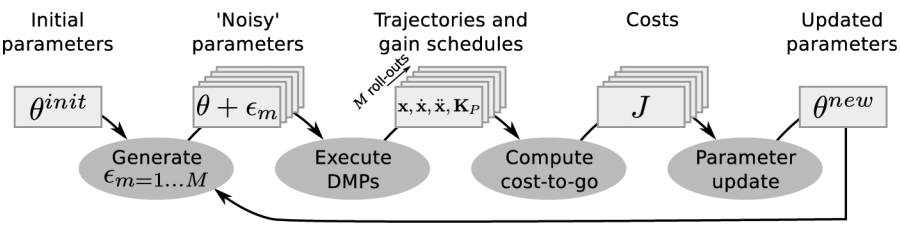


Fig. 4. Generic loop of policy improvement algorithms.

task dynamics *and* arm dynamics this generalization would no be possible.

In summary, the impedance of a joint is parameterized by the choice of the gains $\mathbf{K}_P$ (stiffness) and $\mathbf{K}_D$ (damping). The key to variable impedance control is to allow $\mathbf{K}_P$ to vary as the movement is executed. As introduced in [1], this can be achieved by representing the gain schedules as extra dimensions in the DMP. Since the gains do not have a specific goal value, the proportional gains are not represented as transformation systems that converge to g, but computed directly as a function approximator $K_{P,t} = \mathbf{g}_{t,K}^T \boldsymbol{\theta}_K$, as depicted in Fig. 3.

Parameters for the Experiment: In our experiments, we use supervised learning to initialize $\boldsymbol{\theta}_K$ such that the proportional gains of the 7 joints are constant over time, and have the values $\mathbf{K}_P^{min} = \{60, 60, 16, 16, 6, 6, 1.6\}$. These are 0.4 times the default gains we use for this robot, and the minimum gains we allow during learning, as too low gains lead to poor tracking such that the robot frequently runs into its joint limits. Although we start out with a gain schedule that is constant over time, we shall see in the next section that varying $\boldsymbol{\theta}_K$ leads to varying gain schedules, which are adapted to external perturbations through reinforcement learning.

In summary, in the Dynamic Movement Primitive used in this article, there is one canonical system (representing the phase variable s_t) which drives 3 transformation systems (representing the 3-D end-effector position over time) and 7 function approximators (representing the gain schedules of the 7 joints). We chose to specify the reference trajectory in end-effector space since this is the element the 'subject' has to regulate to fulfill the task and receives feedback on. In contrast we chose to regulate gains in joint space to avoid to have to specify and additional, arbitrary null-space behavior.

D. Reinforcement Learning Algorithm

Given the DMP representation above, the goal is to learn the parameters $\boldsymbol{\theta}$ and $\boldsymbol{\theta}_K$ which minimize the cost function in (17). To do so, we use the policy improvement algorithm PI^2 [27]. Since PI^2 learns $\boldsymbol{\theta}$ and $\boldsymbol{\theta}_K$ simultaneously with the same method, from now on we simply denote these parameters for the end-effector trajectory and gain schedules as one parameter vector $\boldsymbol{\theta}$.

Cost functions for PI^2 take the generic form

$$J(\boldsymbol{\tau}_i) = \phi_{t_N} + \int_{t_i}^{t_N} \left(r_t + \frac{1}{2} \boldsymbol{\theta}^T \mathbf{R} \boldsymbol{\theta} \right) dt \qquad \text{Traj. cost} \quad (27)$$

where J is the finite horizon cost over a trajectory $\boldsymbol{\tau}_i$ starting at time t_i and ending at time t_N. This cost consists of a terminal cost ϕ_{t_N}, an immediate cost r_t, and an immediate control cost

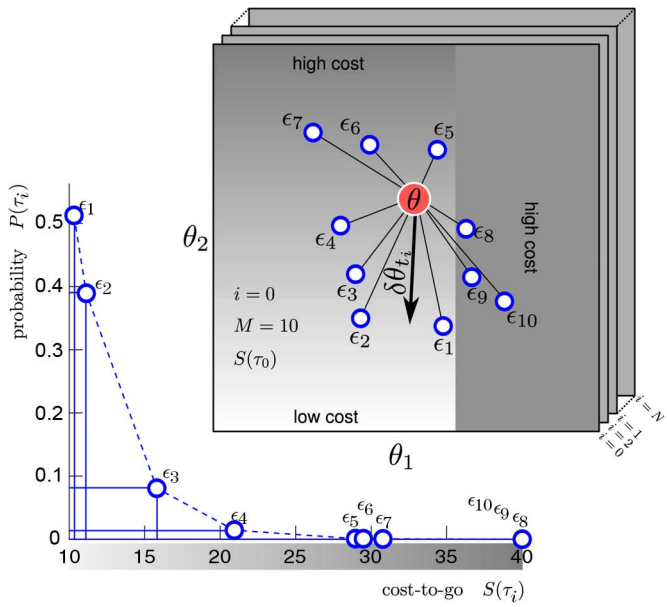


Fig. 5. Visualization of PI^2 exploration and the update rule. The upper right inset shows the parameters $\boldsymbol{\theta}$ and explorations $\boldsymbol{\theta} + \boldsymbol{\epsilon}_{m=1...10}$. The time slice for $t_{i=0}$ is in the foreground; those for $t_{i=1...N}$ are similar but hidden from view. For reasons of clarity, this figure visualizes a DMP with only one dimension and only two basis functions (and thus a parameter vector of length two).

$(1/2)\boldsymbol{\theta}^T \mathbf{R} \boldsymbol{\theta}$. The specific cost function for the task considered in this article was given in (17), and adheres to this generic format.

1) Generic Policy Improvement Loop: Policy improvement methods minimize cost functions through an iterative process of exploration and parameter updating, which we explain using Fig. 4. Exploration is done by executing a Dynamic Movement Primitive M times, each time with slightly different policy parameters $\boldsymbol{\theta} + \boldsymbol{\epsilon}_{t,k}$ which is added to explore the parameter space, as in (28). This noise is sampled from a Gaussian distribution with variance $\boldsymbol{\Sigma}^{\boldsymbol{\theta}}$. A similar Gaussian exploration is applied to the gain schedules as in (29)

$$\frac{1}{\tau}\ddot{x}_t = \alpha\left(\beta(g - x_t) - \dot{x}_t\right) + \mathbf{g}_t^T \left(\underbrace{\boldsymbol{\theta} + \boldsymbol{\epsilon}_{t,m}}_{\text{Shape exploration}} \right) \quad (28)$$

$$K_{P,t} = \mathbf{g}_{t,K}^T \left(\underbrace{\boldsymbol{\theta}_K + \epsilon_{t,m}^{\theta_K}}_{\text{Gain exploration}} \right) \quad (29)$$

These 'noisy' DMP parameters generate slightly different reference trajectories $\{\ddot{\mathbf{x}}_{r,t}, \dot{\mathbf{x}}_{r,t}, \mathbf{x}_{r,t}\}_{m=1...M}$ and gain schedules

PI^2 Parameter Update Rule

$$S(\boldsymbol{\tau}_{i,m}) = \phi_{t_N,m} + \sum_{j=i}^{N-1} r_{t_j,m} + \frac{1}{2}\sum_{j=i+1}^{N-1} (\boldsymbol{\theta} + \mathbf{M}_{t_j,m}\boldsymbol{\epsilon}_{t_j,m})^T \mathbf{R}(\boldsymbol{\theta} + \mathbf{M}_{t_j,m}\boldsymbol{\epsilon}_{t_j,m}) \quad (30)$$

$$\mathbf{M}_{t_j,m} = \frac{\mathbf{R}^{-1}\mathbf{g}_{t_j}\,\mathbf{g}_{t_j}^T}{\mathbf{g}_{t_j}^T\mathbf{R}^{-1}\mathbf{g}_{t_j}} \quad (31)$$

$$P(\boldsymbol{\tau}_{i,m}) = \frac{e^{-\frac{1}{\lambda}S(\boldsymbol{\tau}_{i,m})}}{\sum_{l=1}^{M}[e^{-\frac{1}{\lambda}S(\boldsymbol{\tau}_{i,l})}]} \quad (32)$$

$$\delta\boldsymbol{\theta}_{t_i} = \sum_{m=1}^{M} [P(\boldsymbol{\tau}_{i,m})\ \mathbf{M}_{t_i,m}\ \boldsymbol{\epsilon}_{t_i,m}] \quad (33)$$

$$[\delta\boldsymbol{\theta}]_j = \frac{\sum_{i=0}^{N-1}(N-i)\ w_{j,t_i}\ [\delta\boldsymbol{\theta}_{t_i}]_j}{\sum_{i=0}^{N-1} w_{j,t_i}(N-i)} \quad (34)$$

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \delta\boldsymbol{\theta} \quad (35)$$

Fig. 6. PI^2 parameter update rule: **Eq. 30—Determine cost-to-go of each roll-out** $S(\boldsymbol{\tau}_{i,m})$ at each time step i. This is an evaluation of the cost function $J(\boldsymbol{\tau}_i)$ in Eq. (27), which is task-dependent and provided by the user. The matrix $\mathbf{M}_{t_j,m}$ (Eq. 31) is needed to project the exploration noise onto the parameter space. Note how each exploration in Fig. 6 leads to a different cost in the cost landscape (Eq. (27)), depicted here in gray scale. **Eq. 32—Compute probability of each roll-out** $P(\boldsymbol{\tau}_{i,m})$ at each time step i by exponentiating the cost-to-go. This exponentiation is visualized in the lower left graph in Fig. 6. The intuition behind this step is that trajectories of lower cost should have higher probabilities. **Eq. 33—Average over roll-outs**. Compute the parameter update $\delta\boldsymbol{\theta}$ for each time step i through probability weighted averaging over the exploration $\boldsymbol{\epsilon}$ of all M roll-outs. Trajectories with higher probability, and thus lower cost, therefore contribute more to the parameter update. Again, $\mathbf{M}_{t_j,m}$ projects the exploration noise onto the parameter space. The resulting update vector $\delta\boldsymbol{\theta}_i$ is depicted as an arrow in the inset in Fig. 6. Note how $\boldsymbol{\theta}$ will move towards a lower cost in the landscape. **Eq. 34—Average over time-steps**. In this step (not visualized in the Fig. 6), we average the parameter update $\delta\boldsymbol{\theta}_{t_i}$ per time step i (every "slice" in the inset) over all time steps to acquire one update vector $\delta\boldsymbol{\theta}$. Each parameter update is weighted according to the number of steps left in the trajectory. This is to give earlier points in the trajectory higher weights, as they influence a larger part of the trajectory. They are also weighted with the activation of the corresponding basis function w_j at time t_i, as the influence of parameter θ_j is highest when w_j is highest. Finally, the actual parameter update is performed with Eq. 35. **Eq. 35—Update the parameters**. In the final step, $\delta\boldsymbol{\theta}$ is added to the parameters $\boldsymbol{\theta}$ to acquire the new parameters which will be used in the next round of explorations.

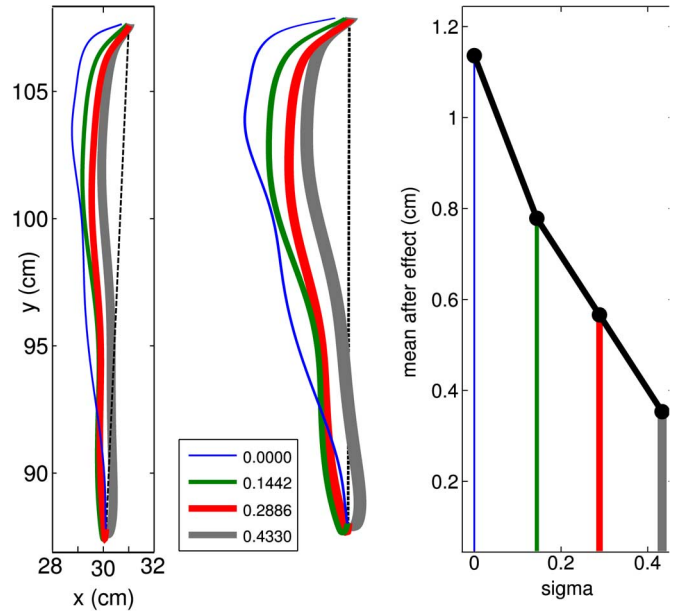


Fig. 7. Left: After-effects (i.e., reference trajectories) after 100 updates for each of the force fields. Center: Same, with x axis scaled $\times 2$ for visualization purposes. Right: Average after-effect (i.e., mean distance to the trajectory before learning) as a function of force field stochasticity.

$\{\mathbf{K}_{P,t}\}_{m=1\ldots M}$, which each lead to different costs. Given the costs and noisy parameters of the M DMP executions, called *roll-outs*, policy improvement methods then update the parameter vector $\boldsymbol{\theta}$ such that it is expected to generate movements that lead to lower costs in the future. The process then continues with the new $\boldsymbol{\theta}$ as the basis for exploration.

2) Policy Improvement With Path Integrals—PI^2: The most crucial part of the policy improvement loop in Fig. 4 is the parameter update; it is here that the key differences between PI^2 and other policy improvement methods lie.

The foundation of PI^2 comes from (model-based) stochastic optimal control for continuous time and continuous state-action systems. The derivation of PI^2 starts with the standard Hamilton–Jacobi Bellman equation [21], which is a nonlinear partial differential equation (PDE). This equation is first linearized by applying a log transformation and assuming that the exploration noise is inversely proportional to the control cost matrix [27]; the underlying intuition is that there should be less exploration in directions where command costs are high. The resulting linear PDE is then transformed into a path integral with the Feynman-Kac theorem [12]. The importance of this transformation is that it is now possible to evaluate the path integral with Monte Carlo roll-outs, i.e., the optimal control problem can be iteratively solved with the generic policy improvement loop depicted in Fig. 4. Finally, the update rule is applied it to Dynamical Movement Primitives, such that the linear component f_t and basis functions $\mathbf{g}_t$ constitute the "system model." Since f_t and $\mathbf{g}_t$ are known to the robot, this renders the algorithm model-free.

Rather than focussing on its derivation from first principles of stochastic optimal control, which is presented extensively in [27], we provide a posthoc interpretation of the resulting update rule in Fig. 6, with a visualization in Fig. 5.

As demonstrated in [27], PI^2 often outperforms previous RL algorithms for parameterized policy learning by at least one order of magnitude in learning speed and also lower final cost performance. It also scales up to very high-dimensional spaces, which enables PI^2 to learn full-body humanoid motor skills [23]. The main reasons for its superior performance are: • There is no need to calculate a gradient, which is sensitive to noise and large derivatives in the value function. The update is rather based on computing a weighted average (Eq. 33), which does not involve a gradient. • No backward propagation of approximations of the value function are required, which allows for a sampling (i.e., roll-out) based method. • Exploration is done is DMP parameter space, rather than state space. For high dimensional problems, it is simply not possible to sample the whole state space.

As an additional benefit, PI^2 has no open algorithmic parameters, except for the magnitude of the exploration noise $\boldsymbol{\epsilon}_t$ (the

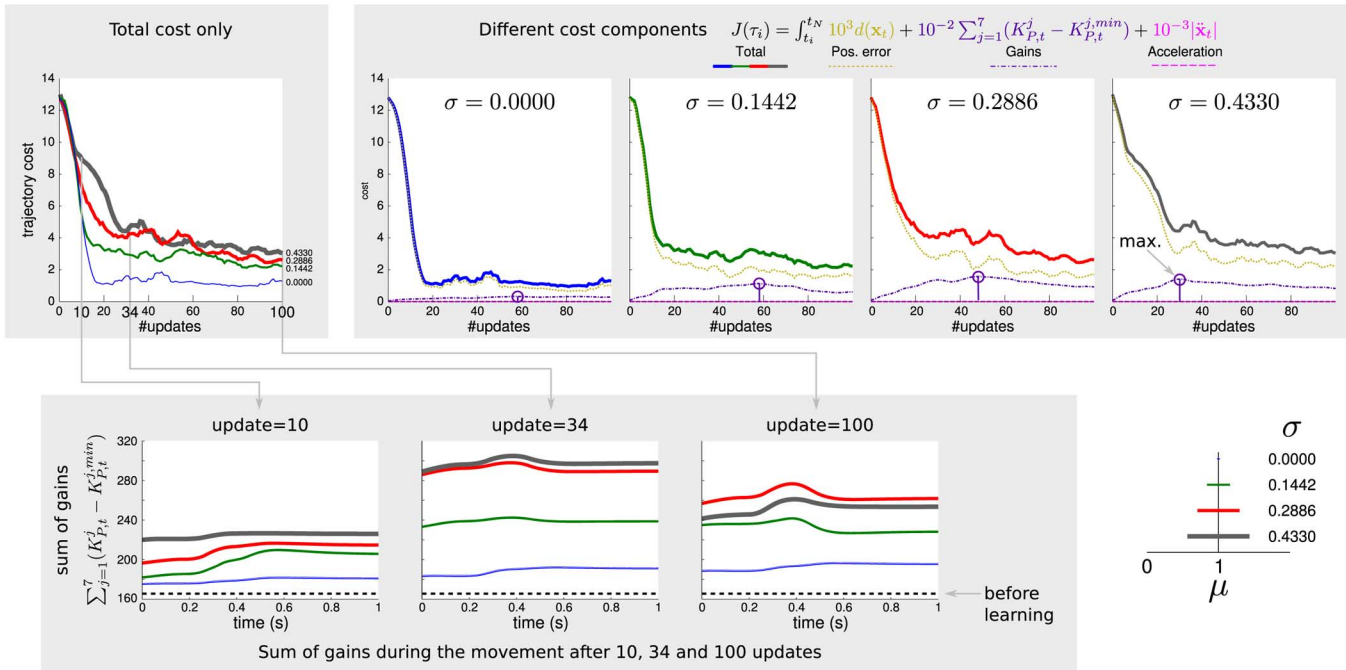


Fig. 8. Top right: Learning curves for the different force fields. The y-axis represents the average costs over the three evaluation roll-outs performed after each update. Top left: total costs over time for the four force fields. Right four graphs: costs for each force field, split up into the different cost components of Eq. (17). Bottom: Sum of the individual gain schedules of the 7 joints after 10, 34, and 100 updates. The gains before learning (which are also the minimum gains) are depicted as dashed lines.

parameter λ is set automatically, cf. [27]). We would like to emphasize that PI^2 is model-free, and *does not* require a model of the control system or the environment.

Parameters for the Experiment: In our experiments, we performed 100 PI^2 updates with $M = 10$ roll-outs per update for each of the four force fields. The exploration noise was $\boldsymbol{\Sigma}^{\boldsymbol{\theta}} = 10^2$ for the Cartesian positions, and $\boldsymbol{\Sigma}^{\boldsymbol{\theta}_K} = 10^{-3} \cdot \mathbf{K}_P^{min}$ for the 7 gain schedules,[4] where $\mathbf{K}_P^{min}$ are the minimum gains as listed in Section IV-C.

V. Results

After each PI^2 update, three roll-outs were executed *without* exploration noise for evaluation purposes. For these three roll-outs, force fields with strength $\beta = 1-\sigma, 1, 1+\sigma$ were used. For all four forcefields (each with a different level of stochasticity determined by σ), the reference trajectories, force fields and actual trajectories at various stages of learning are depicted on the last page in Fig. 9. For comparison, the reference trajectories after 100 updates for the four force fields are depicted together in Fig. 7. These correspond to the after-effects that occur when the force field is turned off, similar to the graphs in [14], [19].

The learning curves for the four force fields are depicted in the top row of Fig. 8. In this figure, the cost for each force field are also split up into the different cost components of (17). The cost due to acceleration is relatively low, and hardly visible in Fig. 8. The sum over all gain schedules at various stages during learning are depicted in the bottom row of Fig. 8.

[4]The relatively low exploration noise for the gains does not express less exploration per se, but is rather due to numerical differences in using the function approximator to model the gains directly [see (23)] rather than as the nonlinear component of a transformation system [see (18)]

A. Discussion

A closer inspection of the individual cost components, reference trajectories and gain schedules leads to the following observations:

- After learning, higher stochasticity σ leads to higher peak values for the gains (correlation coefficient $R = 0.86$) and smaller after-effects (right graph Fig. 7, $R = -0.99$), which is consistent with human adaptations observed in [2], [14].
- The main adaptation of the reference trajectory happens in the first 10 updates. As can be seen in Fig. 9, the reference trajectories after 10 updates are already close to the shape they have at the end of learning after 100 updates. This suggests that the robot first learns the feed-forward term to compensate for the average force field with $\beta = 1$. Since the force field pushes from left to right, the reference trajectory is placed to the left, and is approximately a mirror image of the perturbed trajectory before learning. This is consistent with the observations on human adaptation in [2], [14], [19].
- In each of the four force field experiments, the gains initially go up, as reflected in the increasing costs due to the gains in Fig. 8. After reaching a maximum value, the gains then slowly decrease. This effect is consistent with our observations in several robotic tasks [1], [23]. There, we demonstrated that the algorithm initially increases the gains to minimize the main cost component (low position error), and then fine-tunes the gains to minimize the overall cost (low position error *and* low costs). In this article, this hypothesis is supported by the fact that the maximum cost

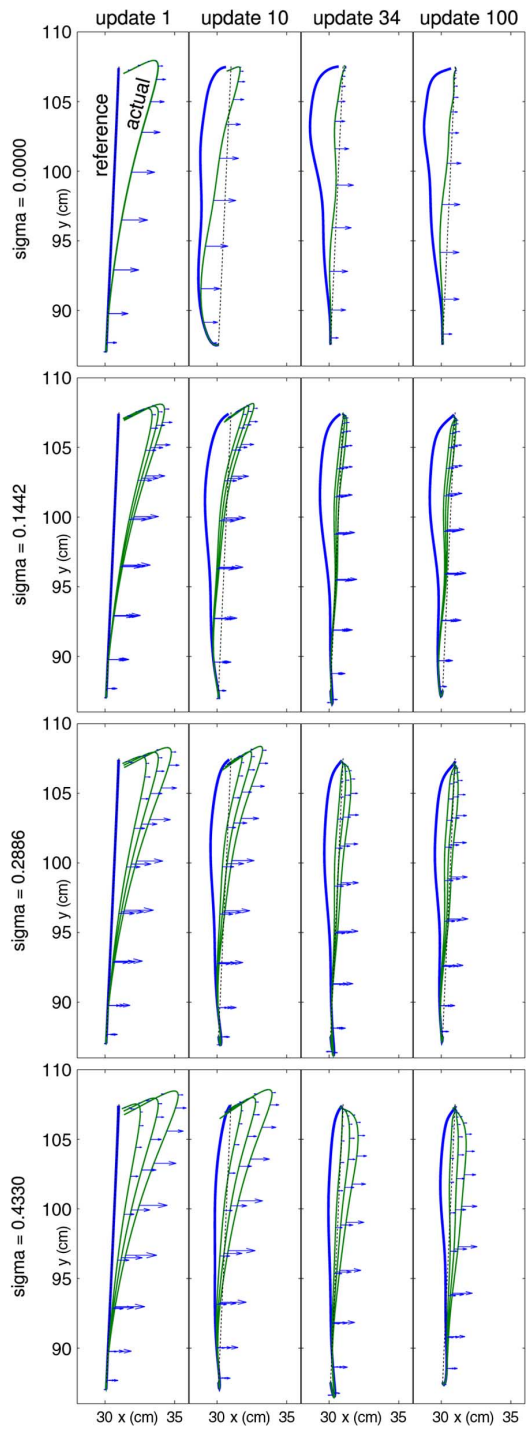


Fig. 9. Reference and actual hand trajectories as learning progresses, for the different force fields. Each row represents a force field with different σ. The actual trajectories represent the perturbed motions when executing the reference trajectories for force fields with strengths $\beta = \{1-\sigma, 1, 1+\sigma\}$.

due to the gains is achieved earlier if the force field is more stochastic.

- We are not able to reproduce the effect seen in [14], where a small amount of stochasticity leads to larger after-effect than in the deterministic case. This supports the hypothesis that a small amount of variance aids in learning the internal model. These effects are difficult to reproduce for PI^2, as it uses only the exploration noise it generates itself to update the policy parameters (see (28) and 33).

In summary, the robot adapts to the mean of perturbation by moving the reference trajectory in the opposite direction of the force field, and adapts to stochasticity in the perturbations by increasing the impedance. These are qualitatively similar results to those observed biophysics experiments [14].

For now, the results could not be modelled quantitatively, as there are some clear differences between humans and our robotic platform. In particular, the kinematics and dynamics of the robot are not the same as the human body; this alone may explain many of the quantitative differences. Also, humans learn in "muscle space," and higher impedance is caused by cocontraction of the muscles. In contrast, our robot learns in gain space, as in [5]. Finally, the robot learns much slower than humans (1000 trials versus 175 in [14]). We believe the main reason for this is that in contrast to humans, our controller and PI^2 have no built-in reflexes, and start without any initial knowledge about the domain. Also, humans learn continually during and after each trial, whereas PI^2 requires M trials ($M = 10$ in this article) to be performed before updating the parameters. Our current research focuses mainly on applying our methods to more human-like kinematics, and biologically plausible muscle models. We are also developing a version of PI^2 in which the last M trials are kept in a FIFO buffer, allowing updates to be performed after each trial, which would enable continual learning.

As in [4], [5], we assume a straight desired path to calculate the task reward, i.e., deviations from this (invariant) desired path are penalized. Without this position error penalty, we are not able to simulate the human movement data. The plausibility of a desired trajectory in biology [29] is still under debate, and the role of the desired trajectory in our system also deserves further investigation.

An important part of the robot's adaptation to the force field is achieved by changing the reference trajectory, i.e., in the analytical example in Fig. 1 this was $\overline{F} = -K_P(x - x_r)$, with $x_r = x_d - \widetilde{F}/K_P$. The position offset $\widetilde{F}/K_P$ leads to larger errors between the reference path and actual trajectory $(x_d - x)$ and thus larger forces, which counteract the force field. Changing the reference position to exert a force is known as *indirect force control* [20]. An alternative would be to directly learn a reference force F_r to compensate for the force field, and perform *direct force control*, i.e., $\overline{F} = -K_P(x - x_d) + F_r$. Since both direct and indirect force control will lead to similar after-effects, it is not clear which approach humans use. In our future work, we will compare the results of learning with direct and indirect force control on our robot platform. Any differences observed on the robot could assist us in designing experiments to verify which form of control humans use to compensate for stochastic perturbations.

VI. Conclusion

Variable impedance control is essential for safe physical interaction with the environment, and in humans, achieving it is the results of years of learning and development. Task-specific

adaptation of impedance in humans is commonly studied within the force field paradigm, where human reaching movements are perturbed by external forces, and compensation strategies used to counter the perturbations are analyzed.

In this article we demonstrate how our reinforcement learning algorithm PI^2 is able to find motor policies that qualitatively replicate human movement data in such stochastic force field experiments. Dynamic movement primitives and the PI^2 algorithm have previously been applied to learning reference trajectories and gain schedules for complex high-dimensional robotic tasks [1], [23]. One advantage of PI^2 is that it does not require a model of the system or environment, making it a more biologically plausible alternative to model-based optimal control methods. That our algorithms required no modifications to generate these results (only the cost function and environment are specific to the task) highlights the general applicability of model-free algorithms. That PI^2 finds locally optimal solutions without a model of the force field or the robot supports the hypothesis that human learning in such situations is based on stochastic optimality, and that such optimal policies can be learned from experience without requiring a model.

Model-free learning is a prerequisite for autonomous development, where models are extracted from observed data rather than prespecified by a designer. We also believe that learning variable impedance control is especially relevant to developmental robotics, as it allows for safe continual exploration with the environment, and thus facilitates trial-and-error learning.

Our current work is aimed at several research topics: First of all, we use covariance matrix updating, a well-known strategy in evolutionary optimization [6] to automatically determine the right level of exploration over time, which enables life-long reinforcement learning. In a second project, inspired by the HORDE architecture [26], we are using machine learning to determine which tasks can be used in which contexts, and how existing tasks can be reused for novel task context. Furthermore, we are continuing our modeling of biophysics results, and are directing our efforts to reproducing the impedance ellipsoids in [18].

ACKNOWLEDGMENT

The authors would like to thank the reviewers for their constructive suggestions for improvement of the paper.

REFERENCES

[1] J. Buchli, F. Stulp, E. Theodorou, and S. Schaal, "Learning variable impedance control," *Int. J. Robot. Res.*, vol. 30, no. 7, pp. 820–833, 2011.

[2] C. Takahashi, D. Scheidt, and R. Reinkensmeyer, "Impedance control and internal model formation when reach in a randomly varying dynamical environment," *J. Neurophys.*, vol. 86, pp. 1047–1051, 2001.

[3] S. Dayanidhi, A. Hedberg, I. Hägg, N. Lilja, H. Forssberg, and F. J. Valero-Cuevas, "Dynamic control of fingertip forces: Development in childhood and decline with aging," presented at the Annu. Amer. Soc. Biomech. Conf., 2011.

[4] D. W. Franklin, E. Burdet, K. P. Tee, R. Osu, C. M. Chew, T. E. Milner, and M. Kawato, "CNS learns stable, accurate, and efficient movements using a simple algorithm," *J. Neurosci.*, vol. 28, no. 44, pp. 11165–11173, 2008.

[5] G. Ganesh, A. Albu-Schäffer, M. Haruno, M. Kawato, and E. Burdet, "Biomimetic motor behavior for simultaneous adaptation of force, impedance and trajectory in interaction tasks," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2010, pp. 2705–2711.

[6] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evol. Comput.*, vol. 9, no. 2, pp. 159–195, 2001.

[7] N. Hogan, "Impedance control—An approach to manipulation. I—Theory. II—Implementation. III—Applications," *ASME Trans. J. Dynamic Syst., Meas., Contr.*, vol. 107, pp. 1–24, 1985.

[8] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," presented at the IEEE Int. Conf. Robot. Autom. (ICRA), 2002.

[9] J. Izawa, T. Kondo, and K. Ito, "Biological robot arm motion through reinforcement learning," presented at the IEEE Int. Conf. Robot. Autom. (ICRA), 2002.

[10] J. Izawa, T. Kondo, and K. Ito, "Biological arm motion through reinforcement learning," *Biol. Cybern.*, vol. 91, no. 1, pp. 10–22, 2004.

[11] J. Izawa, T. Rane, O. Donchin, and R. Shadmehr, "Motor adaptation as a process of reoptimization," *J. Neurosci.*, vol. 278, pp. 2325–2332, 2008.

[12] H. J. Kappen, "Path integrals and symmetry breaking for optimal control theory," *J. Statist. Mech.: Theory Exper.*, vol. 2005, no. 11, p. 11011, 2005.

[13] B. Kim, J. Park, S. Park, and S. Kang, "Impedance learning for robotic contact tasks using natural actor-critic algorithm," *IEEE Trans. Syst., Man, Cybern., Part B: Cybern.*, vol. 40, no. 2, pp. 433–443, 2009.

[14] M. Mistry, E. Theodorou, H. Hoffmann, and S. Schaal, "The dual role of uncertainty in force field learning," presented at the Abstracts 18th Annu. Meeting Neural Contr. Movement (NCM), 2008.

[15] D. Mitrovic, S. Klanke, M. Howard, and S. Vijayakumar, "Exploiting sensorimotor stochasticity for learning control of variable impedance actuators," presented at the IEEE-RAS Int. Conf. Human. Robots (Humanoids), 2010.

[16] S. Schaal, The SL Simulation and Real-Time Control Software Package Univ. Southern California, 2009, Unpublished.

[17] R. Scheidt, B. Dingwell, and F. Mussa-Ivaldi, "Learning to move amid uncertainty," *J. Neurophys.*, vol. 86, pp. 971–985, 2001.

[18] L. P. Selen, D. W. Franklin, and D. M. Wolpert, "Impedance control reduces instability that arises from motor noise," *J. Neurosci.*, vol. 7, no. 40, pp. 12606–12616, 2009.

[19] R. Shadmehr and F. A. Mussa-Ivaldi, "Adaptive representation of dynamics during learning of a motor task," *J. Neurosci.*, vol. 14, no. 5, pp. 3208–3224, 1994.

[20] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics—Modelling, Planning and Control*. Berlin, Germany: Springer-Verlag, 2009.

[21] R. F. Stengel, *Optimal Control and Estimation*. New York: Dover Publications, 1994.

[22] F. Stulp, J. Buchli, A. Ellmer, M. Mistry, E. Theodorou, and S. Schaal, "Reinforcement learning of impedance control in stochastic force fields," presented at the Int. Conf. Develop. Learn. (ICDL), 2011.

[23] F. Stulp, J. Buchli, E. Theodorou, and S. Schaal, "Reinforcement learning of full-body humanoid motor skills," in *Proc. 10th IEEE-RAS Int. Conf. Human. Robots*, 2010, pp. 405–410, Best paper finalist.

[24] F. Stulp and S. Schaal, "Hierarchical reinforcement learning with motion primitives," presented at the 11th IEEE-RAS Int. Conf. Humanoid Robots, 2011.

[25] F. Stulp, E. Theodorou, J. Buchli, and S. Schaal, "Learning to grasp under uncertainty," presented at the IEEE Int. Conf. Robot. Autom. (ICRA), 2011.

[26] R. S. Sutton, J. Modayil, M. Delp, T. Degris, P. M. Pilarski, A. White, and D. Precup, "Horde: A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction," presented at the 10th Int. Conf. Auton. Agents Multiagent Syst., 2011.

[27] E. Theodorou, J. Buchli, and S. Schaal, "A generalized path integral control approach to reinforcement learning," *J. Mach. Learn. Res.*, vol. 11, 2011.

[28] S. Vijayakumar, A. D'Souza, and S. Schaal, "Incremental online learning in high dimensions," *Neural Comput.*, vol. 17, no. 12, pp. 2602–2634, 2005.

[29] D. M. Wolpert, Z. Ghahramani, and M. I. Jordan, "Are arm trajectories planned in kinematic or dynamic coordinates? An adaptation study," *Exp. Brain Res.*, vol. 103, pp. 460–470, 1995, 10.1007/BF00241505.

Freek Stulp received the M.Sc. degree in cognitive science and engineering from the University of Groningen, and received his doctorate degree in Computer Science from the Technische Universität München in 2007.

He is currently an Assistant Professor at the École Nationale Supérieure de Techniques Avancées (ENSTA-ParisTech), Paris, France, and a Member of the FLOWERS team at INRIA Bordeaux. His research interests include robotics, reinforcement learning, motion primitives, and developmental principles for robot learning. His main application domain is autonomous manipulation in human environments.

Dr. Stulp was awarded Postdoctoral Research Fellowships from the Japanese Society for the Promotion of Science and the German Research Foundation (DFG), to pursue his research at the Advanced Telecommunications Research Institute International, Kyoto, Japan, and the University of Southern California, Los Angeles.

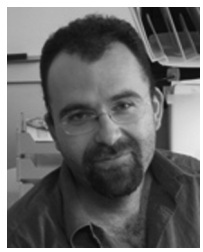
Jonas Buchli received the Dipl. degree in electrical engineering from ETH Zürich, Zürich, Switzerland, in 2003, and the Ph.D. degree from EPF Lausanne, Lausanne, Switzerland, in 2007.

From 2007 to 2011, he was a Postdoc at the Computational Learning and Motor Control Laboratory at the University of Southern California, Los Angeles, where he was the Team Leader of the USC Team for the DARPA Learning Locomotion challenge. Since 2010, he has been a Team Leader at the Advanced Robotics Department of the Italian Institute of Technology in Genova. In May 2012, he was appointed Assistant Professor at ETH Zürich. His research interests include model-based control of legged robotic and human locomotion and manipulation, machine learning and adaptive control, and dynamic, versatile service, and field robots.

Alice Ellmer received the B.Sc. degree in cognitive science from the University of Osnabrueck, in 2006., and the M.Sc. degrees in computer science and neuroscience from the University of Southern California, Los Angeles, in 2012.

Her research interests include machine learning, reinforcement learning, and free and open source software.

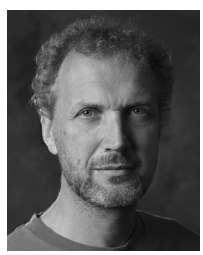
Michael Mistry is a Lecturer in robotics at the School of Computer Science, University of Birmingham, Birmingham, U.K., where he is also a Member of the Intelligent Robotics Laboratory and the Centre for Computational Neuroscience and Cognitive Robotics. His research focuses on issues relevant to dexterous movement in both humans and humanoid robots, including redundancy resolution and inverse kinematics, operational space control and manipulation, stochastic optimal control, and internal model learning and control, particularly in environmental contact. Previously, he was a Postdoc at the Disney Research Laboratory at Carnegie Mellon University, a Researcher at the ATR Computational Neuroscience Laboratory, and a Ph.D. student in Stefan Schaals CLMC Laboratory at the University of Southern California.



Evangelos A. Theodorou received the M.Sc. degree in electrical and computer engineering and the M.Sc. degree in production engineering from the Technical University of Crete, Athens, Greece, in 2001 and 2004, respectively. He received the M.Sc. degree in computer science and engineering from the University of Minnesota, Minneapolis, in 2007. He received the M.Sc. degree in electrical engineering and the Ph.D. degree in computer science from the Viterbi School of Engineering at the University of Southern California (USC), Los Angeles, in 2010 and 2011, respectively.

Since 2011, he has been a Postdoctoral Research Associate with the Computer Science and Engineering Department, University of Washington, Seattle. His research interest include the areas of control, estimation, and machine learning theory with focus on Stochastic Optimal Control, Stochastic Estimation, Reinforcement Learning, and applications to Robotics and Computational Neuroscience.

Dr. Theodorou has been a Recipient of a Myronis Fellowship for Engineering Graduate Students at USC.



Stefan Schaal is currently a Professor of Computer Science, Neuroscience, and Biomedical Engineering at the University of Southern California, Los Angeles, and a Founding Director of the Max-Planck-Institute for Intelligent Systems in Tübingen, Germany. He is also an Invited Researcher at the ATR Computational Neuroscience Laboratory in Japan, where he held an appointment as Head of the Computational Learning Group during an international ERATO project, the Kawato Dynamic Brain Project (ERATO/JST). His research interests include topics of statistical and machine learning, neural networks, computational neuroscience, functional brain imaging, nonlinear dynamics, nonlinear control theory, and biomimetic robotics. He applies his research to problems of artificial and biological motor control and motor learning, focusing on both theoretical investigations and experiments with human subjects and anthropomorphic robot equipment.