2008 IEEE/RSJ International Conference on Intelligent Robot Systems (IROS)

Learning perceptual Coupling for Motor Primitives

Jens Kober, Betty Mohler, Jan Peters
Max-Planck-Institute for Biological Cybernetics
Spemannstr. 38, 72076 Tuebingen, Germany
Email: {kober,mohler,jrpeters}@tuebingen.mpg.de

***Abstract*—Dynamic system-based motor primitives [1] have enabled robots to learn complex tasks ranging from Tennis-swings to locomotion. However, to date there have been only few extensions which have incorporated perceptual coupling to variables of external focus, and, furthermore, these modifications have relied upon handcrafted solutions. Humans learn how to couple their movement primitives with external variables. Clearly, such a solution is needed in robotics.**

In this paper, we propose an augmented version of the dynamic systems motor primitives which incorporates perceptual coupling to an external variable. The resulting perceptually driven motor primitives include the previous primitives as a special case and can inherit some of their interesting properties. We show that these motor primitives can perform complex tasks such a Ball-in-a-Cup or Kendama task even with large variances in the initial conditions where a skilled human player would be challenged. For doing so, we initialize the motor primitives in the traditional way by imitation learning without perceptual coupling. Subsequently, we improve the motor primitives using a novel reinforcement learning method which is particularly well-suited for motor primitives.

I. Introduction

The recent introduction of motor primitives based on dynamic systems [1], [2], [3], [4] have allowed both imitation learning and Reinforcement Learning to acquire new behaviors fast and reliable. Resulting successes have shown that it is possible to rapidly learn motor primitives for complex behaviors such as tennis swings [1], [2], T-ball batting [5], drumming [6], biped locomotion [3], [7] and even in tasks with potential industrial application [8]. However, in their current form these motor primitives are generated in such a way that they are either only coupled to internal variables [1], [2] or only include manually tuned phase-locking, e.g., with an external beat [6] or between the gait-generating primitive and the contact time of the feet [3], [7]. In many human motor control tasks, more complex perceptual coupling is needed in order to perform the task. Using handcrafted coupling based on human insight will in most cases no longer suffice.

In this paper, it is our goal to augment the Ijspeert-Nakanishi-Schaal approach [1], [2] of using dynamic systems as motor primitives in such a way that it includes perceptual coupling with external variables. Similar to the biokinesiological literature on motor learning (see e.g., [9]), we assume that there is an object of internal focus described by a state $\mathbf{x}$ and one of external focus $\mathbf{y}$. The coupling between both foci usually depends on the phase of the movement and, sometimes, the coupling only exists in short phases, e.g., in a catching movement, this could be at initiation of the movement (which is largely predictive) and during the last moment when the object is close to the hand (which is largely prospective or reactive and includes movement correction). Often, it is also important that the internal focus is in a different space than the external one. Fast movements, such as a Tennis-swing, always follow a similar pattern in joint-space of the arm while the external focus is clearly on an object in Cartesian space or fovea-space. As a result, we have augmented the motor primitive framework in such a way that the coupling to the external, perceptual focus is phase-variant and both foci $\mathbf{y}$ and $\mathbf{x}$ can be in completely different spaces.

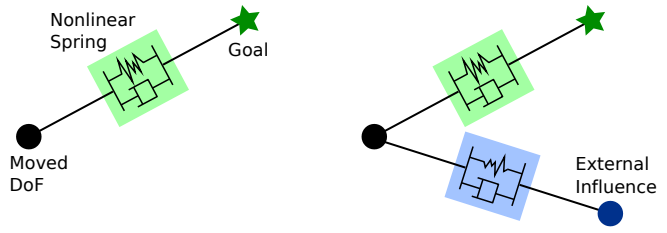
Integrating the perceptual coupling requires additional function approximation, and, as a result, the number of parameters of the motor primitives grows significantly. It becomes increasingly harder to manually tune these parameters to high performance and a learning approach for perceptual coupling is needed. The need for learning perceptual coupling in motor primitives has long been recognized in the motor primitive community [4]. However, learning perceptual coupling to an external variable is not as straightforward. It requires many trials in order to properly determine the connections from external to internal focus. It is straightforward to grasp a general movement by imitation and a human can produce a Ball-in-a-Cup movement or a Tennis-swing after a single or few observed trials of a teacher but he will never have a robust coupling to the ball. Furthermore, small differences between the kinematics of teacher and student amplify in the perceptual coupling. This part is the reason why perceptually driven motor primitives can be initialized by imitation learning but will usually require self-improvement by reinforcement learning. This is analogous to the case of a human learning tennis: a teacher can show a forehand but a lot of self-practice is needed for a proper tennis game.

II. Augmented Motor Primitives with Perceptual Coupling

In this section, we first introduce the general idea behind dynamic system motor primitives as suggested in [1], [2] and, subsequently, show how perceptual coupling can be introduced. Subsequently, we show how the perceptual coupling can be realized by augmenting the acceleration-based framework from [4].

A. Perceptual Coupling for Motor Primitives

The basic idea in the original work of Ijspeert, Nakanishi and Schaal [1], [2] is that motor primitives can be parted into



(i) Ijspeert Model (ii) New Model

Figure 1. Illustration of the behavior of the motor primitives (i) and the augmented motor primitives (ii).

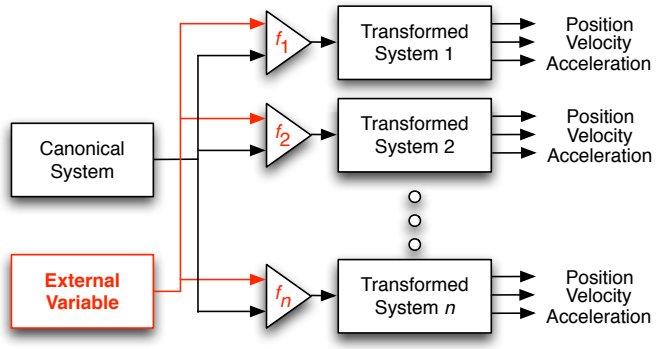


Figure 2. General schematic illustrating both the original motor primitive framework by [2], [4] in black and the augmentation for perceptual coupling in red.

two components, i.e., a canonical system $\mathbf{h}$ which drives transformed systems $\mathbf{g}_k$ for every considered degree of freedom k. As a result, we have system of differential equations given by

$$\dot{\mathbf{z}} = \mathbf{h}(\mathbf{z}), \quad (1)$$
$$\dot{\mathbf{x}} = \mathbf{g}(\mathbf{x}, \mathbf{z}, \mathbf{w}), \quad (2)$$

which determines the variables of internal focus $\mathbf{x}$. Here, $\mathbf{z}$ denotes the state of the canonical system and $\mathbf{w}$ the internal parameters for transforming the output of the canonical system. The schematic in Figure 2 illustrates this traditional setup in black. In Section II-B, we will discuss good choices for these dynamical systems as well as their coupling based on the most current formulation [4].

When taking an external variable $\mathbf{y}$ into account, there are three different ways how this variable influences the motor primitive system which one can consider, i.e., (i) it could only influence Eq.(1) which would be appropriate for synchronization problems and phase-locking (similar as in [6], [10]), (ii) only affect Eq.(2) which allows the continuous modification of the current state of the system by another variable and (iii) the combination of (i) and (ii). While (i) and (iii) are the right solution if phase-locking or synchronization are needed, the coupling in the canonical system will destroy many of the nice properties of the system and make it prohibitively hard to learn in practice. Furthermore, as we focus on discrete movements in this paper, we focus on the case (ii) which has not been used to date. In this case, we have a modified dynamical system

$$\dot{\mathbf{z}} = \mathbf{h}(\mathbf{z}), \quad (3)$$
$$\dot{\mathbf{x}} = \hat{\mathbf{g}}(\mathbf{x}, \mathbf{y}, \bar{\mathbf{y}}, \mathbf{z}, \mathbf{v}), \quad (4)$$
$$\dot{\bar{\mathbf{y}}} = \overline{\mathbf{g}}(\bar{\mathbf{y}}, \mathbf{z}, \mathbf{w}), \quad (5)$$

where $\mathbf{y}$ denotes the state of the external variable, $\bar{\mathbf{y}}$ the expected state of the external variable and $\dot{\bar{\mathbf{y}}}$ its derivative. This architecture inherits most positive properties from the original work while allowing the incorporation of external feedback. We will show that we can incorporate previous work with ease and that the resulting framework resembles the one in [4] while allowing to couple the external variables into the system.

B. Realization for Discrete Movements

The original formulation in [1], [2] was a major breakthrough as the right choice of the dynamical systems in Equations (1, 2) allows determining the stability of the movement, choosing between a rhythmic and a discrete movement and is invariant under rescaling in both time and movement amplitude. With the right choice of function approximator (in our case locally-weighted regression), fast learning from a teachers presentation is possible. In this section, we first discuss how the most current formulation from the motor primitives as discussed in [4] is instantiated from Section II-A. Subsequently, we show how it can be augmented in order to incorporate perceptual coupling.

While the original formulation in [1], [2] used a second-order canonical system, it has since then been shown that a single first order system suffices [4], i.e., we have

$$\dot{z} = h(z) = -\tau \alpha_h z,$$

which represents the phase of the trajectory. It has a time constant τ and a parameter α_h which is chosen such that the system is stable. We can now choose our internal state such that position of degree of freedom k is given by $q_k = \mathbf{x}_{2k}$, i.e., the $2k$-th component of $\mathbf{x}_i$, the velocity by $\dot{q}_k = \tau \mathbf{x}_{2k+1} = \dot{\mathbf{x}}_{2k}$ and the acceleration by $\ddot{q}_k = \tau \dot{\mathbf{x}}_{2k+1}$. Upon these assumptions, we can express the motor primitives function $\mathbf{g}$ in the following form

$$\dot{\mathbf{x}}_{2k+1} = \tau \alpha_g \left(\beta_g \left(\mathbf{t}_k - \mathbf{x}_{2k} \right) - z \right) + \tau \left(\left(\mathbf{t}_k - \mathbf{x}^0_{2k} \right) + \mathbf{a}_k \right) \mathbf{f}_k, \quad (6)$$
$$\dot{\mathbf{x}}_{2k} = \tau \mathbf{x}_{2k+1}. \quad (7)$$

This function has the same time constant τ as the canonical system, appropriately set parameters α_g, β_g, a goal parameter $\mathbf{t}_k$, an amplitude modifier $\mathbf{a}_k$, and a transformation function $\mathbf{f}_k$. This transformation function transforms the output of the canonical system so that the transformed system can represent complex nonlinear patterns and is given by

$$\mathbf{f}_k(\mathbf{z}) = \sum_{i=1}^{N} \psi_i(z) \mathbf{w}_i \mathbf{z}_k, \quad (8)$$

where $\mathbf{w}$ are adjustable parameters and uses normalized Gaus-

sian kernels without scaling such as

$$\boldsymbol{\psi}_i = \frac{\exp\left(-\mathbf{h}_i \left(z - \mathbf{c}_i\right)^2\right)}{\sum_{j=1}^{N} \exp\left(-\mathbf{h}_j \left(z - \mathbf{c}_j\right)^2\right)} \tag{9}$$

for localizing the interaction in phase space where we have centers $\mathbf{c}_i$ and width $\mathbf{h}_i$.

In order to learn a motor primitive with perceptual coupling, we need two components. First, we need to learn the normal or average behavior $\bar{\mathbf{y}}$ of the variable of external focus $\mathbf{y}$ which can be represented by a single motor primitive $\bar{\mathbf{g}}$, i.e., we can use the same type of function from Equations (2, 5) for $\bar{\mathbf{g}}$ which are learned based on the same $\mathbf{z}$ and given by Equations (6, 7). Additionally, we have the system $\hat{\mathbf{g}}$ for the variable of internal focus $\mathbf{x}$ which determines our actual movements which incorporates the inputs of the normal behavior $\bar{\mathbf{y}}$ as well as the current state $\mathbf{y}$ of the external variable. We obtain the system $\hat{\mathbf{g}}$ by inserting a modified coupling function $\hat{\mathbf{f}}_k(\mathbf{z}, \mathbf{y}, \bar{\mathbf{y}})$ instead of the original $\mathbf{f}_k(\mathbf{z})$ in $\mathbf{g}$. This new function $\hat{\mathbf{f}}_k(\mathbf{z}, \mathbf{y}, \bar{\mathbf{y}})$ in order to include perceptual coupling to $\mathbf{y}$ and obtain

$$\begin{aligned}\hat{\mathbf{f}}_k\left(\mathbf{z}, \mathbf{y}, \bar{\mathbf{y}}\right) &= \sum_{i=1}^{N} \boldsymbol{\psi}_i(z) \hat{\mathbf{w}}_i \mathbf{z}_k \\ &+ \sum_{j=1}^{M} \hat{\boldsymbol{\psi}}_j(z) \left(\boldsymbol{\kappa}_{jk}^T(\mathbf{y} - \bar{\mathbf{y}}) + \boldsymbol{\delta}_{jk}^T(\dot{\mathbf{y}} - \dot{\bar{\mathbf{y}}})\right),\end{aligned}$$

where $\hat{\boldsymbol{\psi}}_j(z)$ denote Gaussian kernels as in Equation (9) with centers $\hat{\mathbf{c}}_j$ and width $\hat{\mathbf{h}}_j$. Note, that it can be useful to set $N > M$ for reducing the number of parameters. All parameters are given by $\mathbf{v} = [\hat{\mathbf{w}}, \boldsymbol{\kappa}, \boldsymbol{\delta}]$. Here, $\hat{\mathbf{w}}$ are just the standard transformation parameters while $\boldsymbol{\kappa}_{jk}$ and $\boldsymbol{\delta}_{jk}$ are the local coupling factors which can be interpreted as gains acting on the difference between the desired behavior of the external variable and its actual behavior. Note that for noise-free behavior and perfect initial positions, such coupling would never play a role; thus, the approach would simplify to the original approach. However, in the noisy, imperfect case, this perceptual coupling can ensure success even in extreme cases.

III. Learning for Perceptually Coupled Motor Primitives

While the transformation function $\mathbf{f}_k(\mathbf{z})$ can be learned from few or even just a single trial, this simplicity no longer transfers to learning the new function $\hat{\mathbf{f}}_k(\mathbf{z}, \mathbf{y}, \bar{\mathbf{y}})$ as perceptual coupling requires that the coupling to an uncertain external variable is learned. While imitation learning approaches are feasible, they require larger numbers of presentations of a teacher with very similar kinematics for learning the behavior sufficiently well. As an alternative, we could follow "Nature as our teacher", and create a concerted approach of imitation and self-improvement by trial-and-error. For doing so, we first have a teacher who presents several trials and, subsequently, we improve our behavior by reinforcement learning.

A. Imitation Learning with Perceptual Coupling

For imitation learning, we can largely follow the original work in [1], [2] and only need minor modifications. We also make use of locally-weighted regression in order to determine the optimal motor primitives, use the same weighting and compute the targets based on the dynamic systems. However, unlike in [1], [2], we need a bootstrapping step as we determine first the parameters for the system described by Equation (5) and, subsequently, use the learned results in the learning of the system in Equation (4). For doing so, we can compute the regression targets for the first system by taking Equation (6) , replacing $\bar{\mathbf{y}}$ and $\dot{\bar{\mathbf{y}}}$ by samples of $\mathbf{y}$ and $\dot{\mathbf{y}}$, and solving for $\mathbf{f}_k(\mathbf{z})$ as discussed in [1], [2]. A local regression yields good values for the parameters of $\mathbf{f}_k(\mathbf{z})$. Subsequently, we can perform the exact same step for $\hat{\mathbf{f}}_k(\mathbf{z}, \mathbf{y}, \bar{\mathbf{y}})$ where only the number of variables has increased but the resulting regression follows analogously. However, note that while a single demonstration suffices for the parameter vector $\mathbf{w}$ and $\hat{\mathbf{w}}$, the parameters $\boldsymbol{\kappa}$ and $\boldsymbol{\delta}$ cannot be learned by imitation as these require deviation from the nominal behavior for the external variable.

However, as discussed before, pure imitation for perceptual coupling can be difficult for learning the coupling parameters as well as the best nominal behavior for a robot with kinematics different from the human, many different initial conditions and in the presence of significant noise. Thus, we suggest to improve the policy by trial-and-error using reinforcement learning upon an initial imitation.

B. Reinforcement Learning for Perceptually Coupled Motor Primitives

Reinforcement learning [11] of discrete motor primitives is a very specific type of learning problem where it is hard to apply generic reinforcement learning algorithms [5], [12]. For this reason, the focus of this paper is largely on domain-appropriate reinforcement learning algorithms which operate on parametrized policies for episodic control problems.

1) Reinforcement Learning Setup: When modeling our problem as a reinforcement learning problem, we always have a state $\mathbf{s} = [\mathbf{z}, \mathbf{y}, \bar{\mathbf{y}}, \mathbf{x}]$ with high dimensions (as a result, standard RL methods which discretize the state-space can no longer be applied), and the action $\mathbf{a} = [\mathbf{f}\left(\mathbf{z}\right) + \boldsymbol{\epsilon}, \hat{\mathbf{f}}_k(\mathbf{z}, \mathbf{y}, \bar{\mathbf{y}}) + \hat{\boldsymbol{\epsilon}}]$ is the output of our motor primitives. Here, the exploration is denoted by $\boldsymbol{\epsilon}$ and $\hat{\boldsymbol{\epsilon}}$, and we can give a stochastic policy $\mathbf{a} \sim \pi(\mathbf{s})$ as distribution over the states with parameters $\boldsymbol{\theta} = [\mathbf{w}, \mathbf{v}] \in \mathbb{R}^n$. After a next time-step δt, the actor transfers to a state $\mathbf{s}_{t+1}$ and receives a reward r_t. As we are interested in learning complex motor tasks consisting of a single stroke [9], [4], we focus on finite horizons of length T with episodic restarts [11] and learn the optimal parametrized policy for such problems. The general goal in reinforcement learning is to optimize the *expected return* of the policy with parameters $\boldsymbol{\theta}$ defined by

$$J(\boldsymbol{\theta}) = \int_{\mathbb{T}} p(\boldsymbol{\tau}) R(\boldsymbol{\tau}) d\boldsymbol{\tau}, \tag{10}$$

where $\boldsymbol{\tau} = [\mathbf{s}_{1:T+1}, \mathbf{a}_{1:T}]$ denotes a sequence of states $\mathbf{s}_{1:T+1} = [\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_{T+1}]$ and actions $\mathbf{a}_{1:T} = [\mathbf{a}_1,$

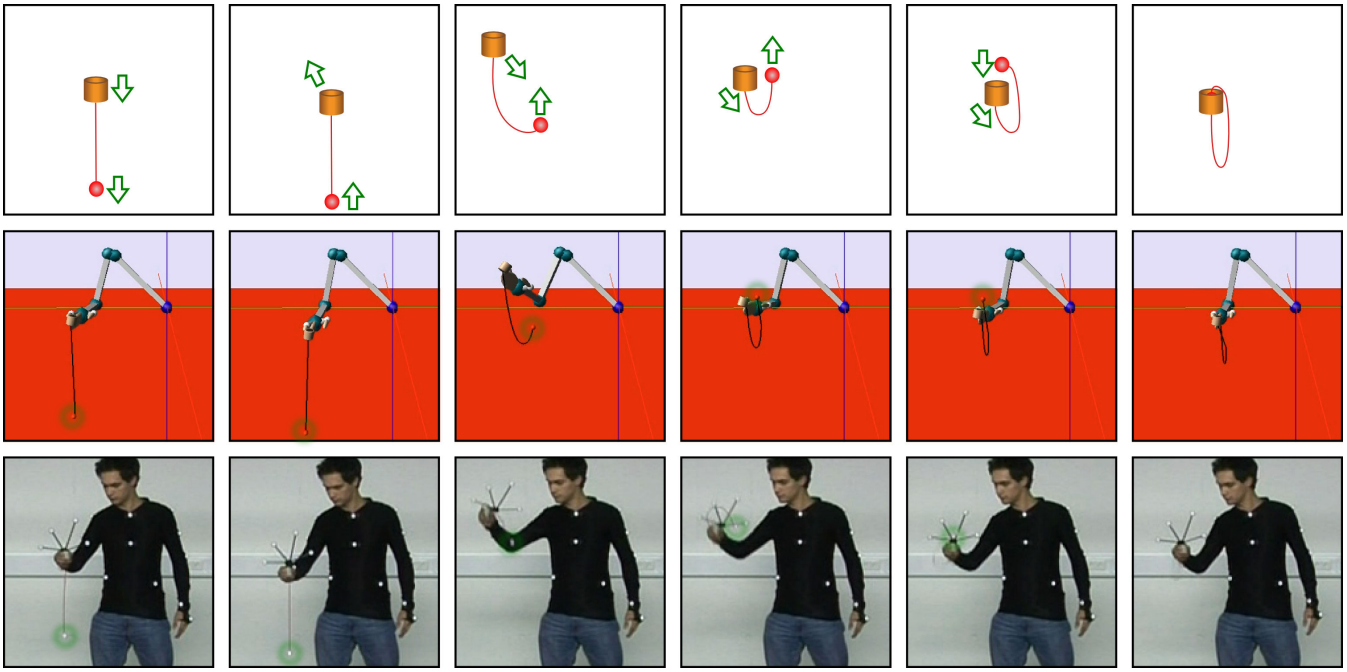


Figure 3. This figure shows schematic drawings of the Ball-in-a-Cup motion, the final learned robot motion as well as a motion-captured human motion. The green arrows show the directions of the momentary movements. The human cup motion was taught to the robot by imitation learning with 91 parameters for 1.5 seconds. Also see the supplementary video in the proceedings.

$\mathbf{a}_2, \ldots, \mathbf{a}_T]$, the probability of an episode $\boldsymbol{\tau}$ is denoted by $p(\boldsymbol{\tau})$ and $R(\boldsymbol{\tau})$ refers to the return of an episode $\boldsymbol{\tau}$. Using Markov assumption, we can write the path distribution as $p(\boldsymbol{\tau}) = p(\mathbf{x}_1)\prod_{t=1}^{T+1} p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)\pi(\mathbf{a}_t|\mathbf{s}_t, t)$ where $p(\mathbf{s}_1)$ denotes the initial state distribution and $p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$ is the next state distribution conditioned on last state and action. Similarly, if we assume additive, accumulated rewards, the return of a path is given by $R(\boldsymbol{\tau}) = \frac{1}{T}\sum_{t=1}^{T} r(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}, t)$, where $r(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}, t)$ denotes the immediate reward.

While episodic Reinforcement Learning (RL) problems with finite horizons are common in motor control, few methods exist in the RL literature (c.f., model-free method such as Episodic REINFORCE [13] and the Episodic Natural Actor-Critic eNAC [5] as well as model-based methods, e.g., using differential-dynamic programming [14]). In order to avoid learning of complex models, we focus on model-free methods and, to reduce the number of open parameters, we rather use a novel Reinforcement Learning algorithm which is based on expectation-maximization. Our new algorithm is called Policy learning by Weighting Exploration with the Returns (PoWER) and can be derived from the same higher principle as previous policy gradient approaches, see [15] for details.

2) Policy learning by Weighting Exploration with the Returns (PoWER): When learning motor primitives, we intend to learn a deterministic mean policy $\bar{\mathbf{a}} = \boldsymbol{\theta}^T\boldsymbol{\mu}(\mathbf{s}) = [\mathbf{f}(\mathbf{z}), \hat{\mathbf{f}}_k(\mathbf{z}, \mathbf{y}, \bar{\mathbf{y}})]$ which is linear in parameters $\boldsymbol{\theta}$ and augmented by additive exploration $\boldsymbol{\varepsilon}(\mathbf{s}, t) = [\hat{\boldsymbol{\epsilon}}, \boldsymbol{\epsilon}]$ in order to make model-free reinforcement learning possible. As a result, the explorative policy can be given in the form $\mathbf{a} = \boldsymbol{\theta}^T\boldsymbol{\mu}(\mathbf{s}, t) + \boldsymbol{\epsilon}(\boldsymbol{\mu}(\mathbf{s}, t))$. Previous work in [5], [12] has focused on state-independent, white Gaussian exploration, i.e., $\boldsymbol{\epsilon}(\boldsymbol{\mu}(\mathbf{s}, t)) \sim \mathcal{N}(0, \Sigma)$, and has resulted into applications such as T-Ball batting [5] and operational space control [12]. However, such unstructured exploration at every step has several disadvantages, i.e., (i) it causes a large variance which grows with the number of time-steps [5], (ii) it perturbs actions too frequently, thus, 'washing' out their effects and (iii) can damage the system executing the trajectory.

Alternatively, one could generate a form of structured, state-dependent exploration $\boldsymbol{\epsilon}(\boldsymbol{\mu}(\mathbf{s}, t)) = \boldsymbol{\varepsilon}_t^T\boldsymbol{\mu}(\mathbf{s}, t)$ with $[\boldsymbol{\varepsilon}_t]_{ij} \sim \mathcal{N}(0, \sigma_{ij}^2)$, where σ_{ij}^2 are meta-parameters of the exploration that can also be optimized. This argument results into the policy $\mathbf{a} \sim \pi(\mathbf{a}_t|\mathbf{s}_t, t) = \mathcal{N}(\mathbf{a}|\boldsymbol{\mu}(\mathbf{s}, \mathbf{t}), \hat{\boldsymbol{\Sigma}}(\mathbf{s}, t))$. Based on the EM updates for Reinforcement Learning as suggested in [12], [15], we can derive the update rule

$$\boldsymbol{\theta}' = \boldsymbol{\theta} + \frac{E_{\boldsymbol{\tau}}\left\{\sum_{t=1}^{T} \boldsymbol{\varepsilon}_t Q^{\pi}(\mathbf{s}_t, \mathbf{a}_t, t\right\}}{E_{\boldsymbol{\tau}}\left\{\sum_{t=1}^{T} Q^{\pi}(\mathbf{s}_t, \mathbf{a}_t, t)\right\}}.$$

In order to reduce the number of trials in this on-policy scenario, we reuse the trials through importance sampling [16], [11]. To avoid the fragility sometimes resulting from importance sampling in reinforcement learning, samples with very small importance weights are discarded.

IV. Evaluation & Application

In this section, we demonstrate the effectiveness of the augmented framework for perceptually coupled motor primitives as presented in Section II and show that our concerted approach of using imitation for initialization and reinforcement learning for improvement works well in practice, particularly

with our novel PoWER algorithm from Section III. We show that this method can be used in learning a complex, real-life motor learning problem Ball-in-a-Cup in a physically realistic simulation of an anthropomorphic robot arm. This problem is a good benchmark for testing the motor learning performance and we show that we can learn the problem roughly at the efficiency of a young child. This algorithm successfully creates a perceptual coupling even to perturbations that are very challenging for a skilled adult player.

A. Robot Application: Ball-in-a-Cup

We have applied the presented algorithm in order to teach a physically-realistic simulation of an anthropomorphic SARCOS robot arm how to perform the traditional American children's game Ball-in-a-Cup, also known as Balero, Bilboquet or Kendama. The toy consists of a ball which is attached to a wooden cup by a string. The initial position is the ball hanging down vertically on the string and the player has to toss the ball into the cup by jerking his arm [17], see Figure 3(top) for an illustrative figure. The state of the system is described in Cartesian coordinates of the cup (i.e., the operational space) and the Cartesian coordinates of the ball. The actions are the cup accelerations in Cartesian coordinates with each direction represented by a motor primitive. An operational space control law [18] is used in order to transform accelerations in the operational space of the cup into joint-space torques. All motor primitives are perturbed separately but employ the same joint reward which is $r_t = \exp(-\alpha(x_c - x_b)^2 - \alpha(y_c - y_b)^2)$ the moment where the ball passes the rim of the cup with a downward direction and $r_t = 0$ all other times. The cup position is denoted by $[x_c, y_c, z_c] \in \mathbb{R}^3$, the ball position $[x_b, y_b, z_b] \in \mathbb{R}^3$ and a scaling parameter $\alpha = 10000$. The task is quite complex as the reward is not modified solely by the movements of the cup but foremost by the movements of the ball and the movements of the ball are very sensitive to perturbations. A small perturbation of the initial condition or the trajectory will drastically change the movement of the ball and hence the outcome of the trial if we do not use any form of perceptual coupling to the external variable "ball".

Due to the complexity of the task, Ball-in-a-Cup is even a hard motor task for children who only succeed at it by observing another person playing or deducing from similar previously learned tasks how to maneuver the ball above the cup in such a way that it can be caught. Subsequently, a lot of improvement by trial-and-error is required until the desired solution can be achieved in practice. The child will have an initial success as the initial conditions and executed cup trajectory fit together by chance, afterwards the child still has to practice a lot until it is able to get the ball in the cup (almost) every time and so cancel various perturbations. Learning the necessary perceptual coupling to get the ball in the cup on a consistent basis is even a hard task for adults, as our whole lab can testify. In contrast to a tennis swing, where a human just needs to learn a goal function for the one moment the racket hits the ball, in Ball-in-a-Cup we need a complete dynamical system as cup and ball constantly interact.

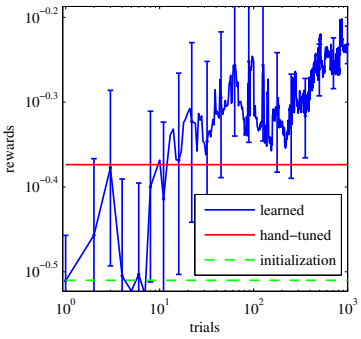


Figure 4. This figure shows the expected return for one specific perturbation of the learned policy in the Ball-in-a-Cup scenario (averaged over 3 runs with different random seeds and the standard deviation indicated by the error bars). Convergence is not uniform as the algorithm is optimizing the returns for a whole range of perturbations and not for this test case. Thus, the variance in the return as the improved policy might get worse for the test case but improve over all cases. Our algorithm rapidly improves, regularly beating a hand-tuned solution after less than fifty trials and converging after approximately 600 trials. Note that this plot is a double logarithmic plot and, thus, single unit changes are significant as they correspond to orders of magnitude.

Mimicking how children learn to play Ball-in-a-Cup, we first initialize the motor primitives by imitation and, subsequently, improve them by reinforcement learning in order to get an initial success. Afterwards we also acquire the perceptual coupling by reinforcement learning.

We recorded the motions of a human player using a VICON^{TM} motion-capture setup in order to obtain an example for imitation as shown in Figure 3(c). The extracted cup-trajectories were used to initialize the motor primitives using locally-weighted regression for imitation learning. The simulation of the Ball-in-a-Cup behavior was verified using the tracked movements. We used one of the recorded trajectories for which, when played back in simulation, the ball goes in but does not pass the center of the opening of the cup and thus does not optimize the reward. This movement is then used for initializing the motor primitives and determining their parametric structure where cross-validation indicates that 91 parameters per motor primitive are optimal from a bias-variance point of view. The trajectories are optimized by reinforcement learning using the PoWER algorithm on the parameters $\mathbf{w}$ for non perturbed initial conditions. The robot constantly succeeds at bringing the ball into the cup after approximately 60-80 iterations given no noise and perfect initial conditions.

One set of the found trajectories is then used to calculate the baseline $\bar{\mathbf{y}} = (\mathbf{h} - \mathbf{b})$ and $\dot{\bar{\mathbf{y}}} = (\dot{\mathbf{h}} - \dot{\mathbf{b}})$, where $\mathbf{h}$ and $\mathbf{b}$ are the hand and ball trajectories. This set is also used to set the standard cup trajectories.

Hand tuned coupling factors work quite well for small perturbations of the initial conditions. In order to make them more robust we use reinforcement learning using the same joint reward as before. The initial conditions (positions and velocities) of the ball are perturbed completely randomly (no PEGASUS Trick) using Gaussian random values with variances set according to the desired stability region. The PoWER algorithm converges after approximately 600-800 iterations.

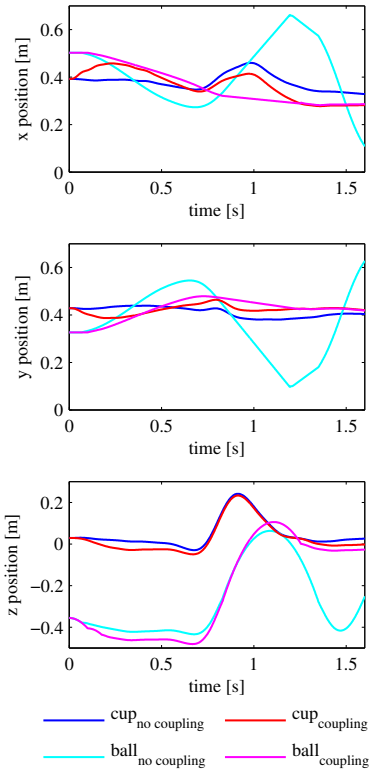


Figure 5. This figure compares cup and ball trajectories with and without perceptual coupling. The trajectories and different initial conditions are clearly distinguishable. The perceptual coupling cancels the swinging motion of the string and ball "pendulum" out. The successful trial is marked either by overlying (x and y) or parallel (z) trajectories of the ball and cup from 1.2 seconds on.

This is roughly comparable to the learning speed of a 10 year old child (Figure 4). For the training we used concurrently standard deviations of 0.01m for x and y and of 0.1 m/s for $\dot{x}$ and $\dot{y}$. The learned perceptual coupling gets the ball in the cup for all tested cases where the hand-tuned coupling was also successful. The learned coupling pushes the limits of the canceled perturbations significantly further and still performs consistently well for double the standard deviations seen in the reinforcement learning process. Figure 5 shows an example of how the visual coupling adapts the hand trajectories in order to cancel perturbations and to get the ball in the cup.

V. Conclusion

Perceptual coupling for motor primitives is an important topic as it results in more general and more reliable solutions while it allows the application of the dynamic systems motor primitive framework to many other motor control problems. As manual tuning can only work in limited setups, an automatic acquisition of this perceptual coupling is essential.

In this paper, we have contributed an augmented version of the motor primitive framework originally suggested by [1], [2], [4] such that it incorporates perceptual coupling while keeping a distinctively similar structure to the original approach and, thus, preserving most of the important properties. We present a concerted learning approach which relies on an initialization by imitation learning and, subsequent, self-improvement by reinforcement learning. We introduce a particularly well-suited algorithm for this reinforcement learning problem called PoWER. The resulting framework works well for learning Ball-in-a-Cup on a simulated anthropomorphic SARCOS arm in setups where the original motor primitive framework would not suffice to fulfill the task.

References

[1] J. A. Ijspeert, J. Nakanishi, and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, Washinton, DC, May 11-15 2002, pp. 1398–1403.

[2] A. Ijspeert, J. Nakanishi, and S. Schaal, "Learning attractor landscapes for learning motor primitives," in *Advances in Neural Information Processing Systems*, S. Becker, S. Thrun, and K. Obermayer, Eds., vol. 15. Cambridge, MA: MIT Press, 2003, pp. 1547–1554.

[3] S. Schaal, J. Peters, J. Nakanishi, and A. Ijspeert, "Control, planning, learning, and imitation with dynamic movement primitives," in *Workshop on Bilateral Paradigms on Humans and Humanoids, IEEE International Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, NV, Oct. 27-31, 2003.

[4] S. Schaal, P. Mohajerian, and A. Ijspeert, "Dynamics systems vs. optimal control — a unifying view," *Progress in Brain Research*, vol. 165, no. 1, pp. 425–445, 2007.

[5] J. Peters and S. Schaal, "Policy gradient methods for robotics," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Beijing, China, 2006, pp. 2219 – 2225.

[6] D. Pongas, A. Billard, and S. Schaal, "Rapbid synchronization and accurate phase-locking of rhythmic motor primitives," in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS 2005)*, vol. 2005, 2005, pp. 2911–2916.

[7] J. Nakanishi, J. Morimoto, G. Endo, G. Cheng, S. Schaal, and M. Kawato, "Learning from demonstration and adaptation of biped locomotion," *Robotics and Autonomous Systems*, vol. 47, no. 2-3, pp. 79–91, 2004.

[8] H. Urbanek, A. Albu-Schäffer, and P.v.d.Smagt, "Learning from demonstration repetitive movements for autonomous service robotics," in *IEEE/RSL Int. Conf. on Intelligent Robots and Systems*, Sendai, Japan, 2004.

[9] G. Wulf, *Attention and motor skill learning*. Champaign, IL: Human Kinetics, 2007.

[10] J. Nakanishi, J. Morimoto, G. Endo, G. Cheng, S. Schaal, and M. Kawato, "A framework for learning biped locomotion with dynamic movement primitives," in *Proceedings of the IEEE-RAS International Conference on Humanoid Robots (HUMANOIDS)*. Los Angeles, CA: Nov.10-12, Santa Monica, CA: IEEE, 2004.

[11] R. Sutton and A. Barto, *Reinforcement Learning*. MIT PRESS, 1998.

[12] J. Peters and S. Schaal, "Reinforcement learning for operational space," in *Proceedings of the International Conference on Robotics and Automation (ICRA)*, Rome, Italy, 2007.

[13] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine Learning*, vol. 8, pp. 229–256, 1992.

[14] C. G. Atkeson, "Using local trajectory optimizers to speed up global optimization in dynamic programming," in *Advances in Neural Information Processing Systems 6*, J. E. Hanson, S. J. Moody, and R. P. Lippmann, Eds. Morgan Kaufmann, 1994, pp. 503–521.

[15] J. Kober and J. Peters, "Policy search for motor primitives in robotics," in *Submitted to NIPS08*, 2008.

[16] C. Andrieu, N. de Freitas, A. Doucet, and M. I. Jordan, "An introduction to MCMC for machine learning," *Machine Learning*, vol. 50, no. 1, pp. 5–43, 2003.

[17] Wikipedia, June 2008. [Online]. Available: http://en.wikipedia.org/wiki/Ball_in_a_cup

[18] J. Nakanishi, M. Mistry, J. Peters, and S. Schaal, "Experimental evaluation of task space position/orientation control towards compliant control for humanoid robots," in *IEEE International Conference on Intelligent Robotics Systems (IROS 2007)*, 2007.