Learning Movement Primitives

Stefan Schaal[1,2], Jan Peters[1], Jun Nakanishi[2], and Auke Ijspeert[1,3]

[1] Computational Learning and Motor Control Laboratory, Computer Science and Neuroscience, University of Southern California, Los Angeles, CA 90089-2520, USA
[2] Dept. of Humanoid Robotics and Computational Neuroscience, ATR Computational Neuroscience Laboratory, 2-2-2 Hikaridai, Seika-cho, Soraku-gun, 619-0288 Kyoto, Japan
[3] School of Computer and Communication Sciences, EPFL, Swiss Federal Institute of Technology Lausanne, CH 1015 Lausanne, Switzerland

Abstract. This paper discusses a comprehensive framework for modular motor control based on a recently developed theory of dynamic movement primitives (DMP). DMPs are a formulation of movement primitives with autonomous nonlinear differential equations, whose time evolution creates smooth kinematic control policies. Model-based control theory is used to convert the outputs of these policies into motor commands. By means of coupling terms, on-line modifications can be incorporated into the time evolution of the differential equations, thus providing a rather flexible and reactive framework for motor planning and execution. The linear parameterization of DMPs lends itself naturally to supervised learning from demonstration. Moreover, the temporal, scale, and translation invariance of the differential equations with respect to these parameters provides a useful means for movement recognition. A novel reinforcement learning technique based on natural stochastic policy gradients allows a general approach of improving DMPs by trial and error learning with respect to almost arbitrary optimization criteria. We demonstrate the different ingredients of the DMP approach in various examples, involving skill learning from demonstration on the humanoid robot DB, and learning biped walking from demonstration in simulation, including self-improvement of the movement patterns towards energy efficiency through resonance tuning.

1 Introduction

With the advent of anthropomorphic and humanoid robots [e.g., 1], a large number of new challenges have been posed to the field of robotics and intelligent systems. Lightweight, highly complex, high degree-of-freedom (DOF) bodies defy accurate analytical modeling such that movement execution requires novel methods of nonlinear control based on learned feedforward controllers [e.g., 2], a control strategy that is also particularly needed since high gain control is not a viable alternative due to frequent contacts of the robot with an unknown environment. Movement planning in high dimensional motor systems offers another challenge. While efficient planning in typical low dimensional industrial robots, usually characterized by three to six DOFs, is already a complex issue [3, 4], optimal planning in 30 to 50 DOF systems with uncertain geometric and dynamic models is quite daunting, especially in the light of the required real-time performance in a reactive robotic system. As one more point, advanced sensing, using vision, tactile sensors, acoustic sensors, and potentially many other sources like olfaction, dis-

P. Dario and R. Chatila (Eds.): Robotics Research, STAR 15, pp. 561–572, 2005.

Fig. 1. The humanoid robot DB

tributed sensor networks, nanosensors, etc., play a crucial role in advanced robotics. Besides finding reliable methods of processing in such sensor rich environments, incorporating the resulting information into the motor and planning loops increases the above-mentioned complexity of planning and control even more.

One of the fundamental questions, common to many of the above issues, revolves around identifying movement primitives [e.g., 5]. The existence of movement primitives seems, so far, the only possibility how one could conceive that autonomous systems can cope with the complexity of motor control and motor learning [5-7]. Developing a theory of control, planning, learning, and imitation with movement primitives is therefore currently a rather prominent topic in both biological and robotic sciences. In the following sections, we will first sketch our idea of Dynamic Movement Primitives, originally introduced in [8-10], illustrate their potential for planning, movement recognition, perception-action coupling, imitation learning, and general reinforcement learning, and exemplify this framework in various applications from humanoid robotics with the humanoid robot DB (Figure 1) and simulation studies.

2 Dynamic Movement Primitives

The goal of motor learning can generally be formulated in terms of finding a task-specific control policy:

$$\mathbf{u} = \pi(\mathbf{x}, t, \alpha) \tag{1}$$

that maps the continuous state vector $\mathbf{x}$ of a control system and its environment, possibly in a time t dependent way, to a continuous control vector $\mathbf{u}$. The parameter vector α denotes the problem specific adjustable parameters in the policy π. Given some cost criterion that can evaluate the quality of an action $\mathbf{u}$ in a particular state $\mathbf{x}$, dynamic programming, and especially its modern relative, reinforcement learning, provide a well founded set of algorithms of how to compute the policy π for complex nonlinear control problems. Unfortunately, as already noted in Bellman's original work, learning of π becomes computationally intractable for even moderately high dimensional state-action spaces.

As a potential way to simplify learning control policies, research has turned towards a more macroscopic representation of policies in form of movement primitives, or, more precisely policy primitives [11]. Movement primitives are parameterized policies that can achieve a complete movement behavior. Planning complex movement based on a small number of such movement primitives can avoid the combinatorial complexity of motor learning in high dimensional movement systems. Thus, a key research topic, both in biological and artificial motor control, revolves around the topics: what is a good set of primitives, how can they be formalized, how can they interact with perceptual input, how can they be adjusted autonomously, how can they be combined task specifically, and what is the origin

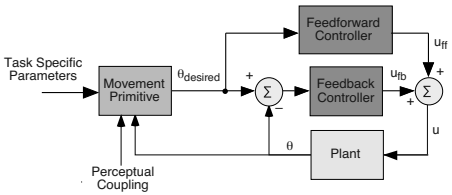

Fig. 2. Sketch of a control diagram with dynamic movement primitives.

of primitives? In order to address these questions, we suggest to resort to some of the most basic ideas of dynamic systems theory. A dynamic system can generally be written as a differential equation:

$$\dot{\mathbf{x}} = f(\mathbf{x}, \alpha, t) \tag{2}$$

which is almost identical to Equation (1), except that the left-hand-side denotes a change-of-state, not a motor command. Such a kinematic formulation is, however, quite suitable for motor control if we conceive of this dynamic system as a kinematic planning policy, whose outputs are subsequently converted to motor commands by an appropriate standard controller (Figure 2) [12]. It should be noted, however, that a kinematic representation of movement primitives is not necessarily independent of the dynamic properties of the limb. Proprioceptive feedback can be used to on-line modify the attractor landscape of a DMP in the same way as perceptual information [13-15]. Figure 2 indicates this property with the "perceptual coupling" arrow — the example in Section 3.2 will clarify this issue.

The two most elementary behaviors of nonlinear dynamic systems are point attractive and limit cycle behaviors, paralleled by discrete and rhythmic movement in motor control. The idea of dynamic movement primitives (DMP) is to exploit well-known simple formulations of such attractor equations to code the basic behavioral pattern (i.e., rhythmic or discrete), and to use statistical learning to adjust the attractor landscape of the DMP to the detailed needs of the task. As will be outlined in the next section, several appealing properties, such as perception-action coupling and reusability of the primitives, can be accomplished in this framework.

2.1 Planning with DMPs

The key question of DMPs is how to formalize nonlinear dynamic equations such that they can be flexibly adjusted to represent arbitrarily complex motor behaviors without the need for manual parameter tuning and the danger of instability of the equations. We will sketch our approach in the example of a discrete dynamic system for reaching movements — an analogous development holds for rhythmic systems.

Assume we have a basic point attractive system, for instance, instantiated by the second order dynamics

$$\tau \dot{z} = \alpha_z \left(\beta_z (g - y) - z \right), \quad \tau \dot{y} = z + f \tag{3}$$

where g is a known goal state, α_z and β_z are time constants, τ is a temporal scaling factor (see below) and $y, \dot{y}$ correspond to the desired position and velocity generated by the equations, interpreted as a movement plan. For appropriate parameter settings and f=0, these equations form a globally stable linear dynamic system with g as a unique point attractor. Could we find a nonlinear function f in

Equation (3) to change the rather trivial exponential convergence of y to allow more complex trajectories on the way to the goal? As such a change of Equation (3) enters the domain of nonlinear dynamics, an arbitrary complexity of the resulting equations can be expected. To the best of our knowledge, this problem has prevented research from employing generic learning in nonlinear dynamic systems so far. However, the introduction of an additional canonical dynamic system (x,v)

$$\tau \dot{v} = \alpha_v\left(\beta_v\left(g - x\right) - v\right), \quad \tau \dot{x} = v \tag{4}$$

and the nonlinear function f

$$f\left(x, v, g\right) = \sum_{i=1}^{N} \psi_i w_i v \Big/ \sum_{i=1}^{N} \psi_i \text{, where } \psi_i = \exp\left(-h_i\left(\frac{x}{g} - c_i\right)^2\right) \tag{5}$$

alleviates this issue. Equation (4) is a second order dynamic system similar to Equation (3), however, it is linear and not modulated by a nonlinear function. Thus, its monotonic global convergence to g can be guaranteed with a proper choice of α_v and β_v, e.g., such that (4) is critically damped. Assuming that all initial conditions of the state variables x, v, y, z are zero, the quotient $x / g \in [0,1]$ can serve as a phase variable to anchor the Gaussian basis functions ψ_i (characterized by a center c_i and bandwidth h_i), and v can act as a "gating term" in the nonlinear function (5) such that the influence of this function vanishes at the end of the movement. If the weights w_i in (5) are bounded, the combined system in (3), (4), and (5) asymptotically converges to the unique point attractor g.

It is not the particular instantiation in Equations (3), (4), and (5) what is the most important idea of DMPs, but rather it is design principle. A DMP consists of two sets of differential equations: a *canonical* system

$$\tau \dot{\mathbf{x}} = \mathbf{h}\left(\mathbf{x}\right) \tag{6}$$

and a *transformation* system

$$\tau \dot{\mathbf{y}} = \mathbf{g}\left(\mathbf{y}, f\left(\mathbf{x}\right)\right) \tag{7}$$

The canonical system needs to generate two quantities: a phase variable x, and a phase velocity v, i.e., $\mathbf{x} = [x \quad v]^T$. The phase x is a substitute for time and allows us anchoring our spatially localized basis functions (5). The appealing property of using a phase variable instead of an explicit time representation is that we can manipulate the time evolution of phase, e.g., by additive coupling terms or phase resetting (cf. Section 3.2) — in contrast, time cannot be manipulated easily. The phase velocity v is a multiplicative term in the nonlinearity (5). If v is set to zero, the influence of the nonlinearity vanishes in the transformation system, and the dynamics of the transformation system with f=0 dominate its time evolution. In the design of a DMP, we usually choose a structure for canonical and transformation systems that are analytically easy to understand, such that the stability properties of the DMP can be guaranteed. The transformation system makes use of f to generate the desired movement represented in the variables $\mathbf{y}$. In the following, we give the equations for the canonical and transformation systems for another formulation of a discrete system, and also a rhythmic system.

A Discrete Acceleration DMP

The canonical system and the function approximator are identical to Equation (4) and Equation (5), respectively. The transformation system is:

$$\tau \dot{z} = \alpha_z\left(\beta_z\left(r - y\right) - z\right) + f, \quad \tau \dot{y} = z, \quad \tau \dot{r} = \alpha_g\left(g - r\right) \tag{8}$$

This set of equations moved the nonlinear function into the differential equation of $\dot{z}$. Thus, $\dot{z}, z, y$ can be interpreted as the desired acceleration, velocity, and position, i.e., $\ddot{y}, \dot{y}, y$, generated by the DMP. In order to ensure a continuous acceleration profile for $\ddot{y}$, we had to introduce a simple first order filter for the goal state in the $\dot{r}$ equation – if α_z and β_z are chosen for critical damping, i.e., $\beta_z = \alpha_z / 4$, then $\alpha_g = \alpha_z / 2$ is a suitable time constant for the $\dot{r}$ equation for a three-fold repeated eigenvalue of $\lambda_{1,2,3} = \alpha_z / 2$ of the linear system in Equation (8) with f=0. The advantage of this "acceleration" DMP is that it can easily be used in conjunction with an inverse model controller that requires a continuous desired acceleration signal.

A Phase Oscillator DMP

By replacing the point attractor in the canonical system with a limit cycle oscillator, a rhythmic DMP is obtained [9]. Among the simplest limit cycle oscillators is a phase-amplitude representation:

$$\tau \dot{r} = \alpha_r\left(A - r\right), \quad \tau \dot{\phi} = 1 \tag{9}$$

where r is the amplitude of the oscillator, A the desired amplitude, and ϕ its phase. For this case, Equation (5) is modified to

$$f\left(r, \phi\right) = \sum_{i=1}^{N} \psi_i \mathbf{w}_i^T \mathbf{v} \Big/ \sum_{i=1}^{N} \psi_i \text{, where } \mathbf{v} = \left[r\cos\phi, r\sin\phi\right]^T$$
$$\text{and } \psi_i = \exp\left(-h_i\left(\text{mod}\left(\phi, 2\pi\right) - c_i\right)^2\right) \tag{10}$$

The transformation system in Equations (3) remains the same, except that we now identify the goal state g with a setpoint around which the oscillation takes place. Thus, by means of A, τ, and g, we can independently control amplitude, frequency, and setpoint of an oscillation.

2.2 Imitation Learning with DMPs

An important issue is how to learn the weights $\mathbf{w}_i$ in the nonlinear function f that characterize the spatiotemporal path of a DMP. Given that f is a normalized basis function representation with linear parameterization [e.g., 16], a variety of learning algorithms exist to find $\mathbf{w}_i$. Let us assume we are given a sample trajectory $y_{demo}(t), \dot{y}_{demo}(t), \ddot{y}_{demo}(t)$ with duration T, e.g., as typical in imitation learning [5]. Based on this information, a supervised learning problem results with the following target for f:

- *For the transformation system in Equation (3), using* $g = y_{demo}(T)$*:*

$$f_{target} = \tau \dot{y}_{demo} - z_{demo} \text{ where } \tau \dot{z}_{demo} = \alpha_z\left(\beta_z\left(g - y_{demo}\right) - z_{demo}\right) \tag{11}$$

- *For the transformation system in Equation (8)*

$$f_{target} = \tau \ddot{y}_{demo} - \alpha_z\left(\beta_z\left(r - y_{demo}\right) - \dot{y}_{demo}\right) \tag{12}$$

In order to obtain a matching input for f_{target}, the canonical system needs to be integrated. For this purpose, in Equation (4), the initial state of the canonical system is set to $v = 0, x = y_{demo}(0)$ before integration. An analogous procedure is performed for the rhythmic DMP. The time constant τ is chosen such that the DMP with f=0 achieves 95% convergence at t=T. With this procedure, a clean supervised learning problem is obtained over the time course of the movement to be approximated with training samples $(\mathbf{v}, f_{target})$ (cf. Equations (5) and (10)). For solving the function approximation problem, we chose a nonparametric regression technique from locally weighted learning (LWPR) [17] as it allows us to determine the necessary number of basis functions N, their centers c_i, and bandwidth h_i automatically.

2.3 Reinforcement Learning of DMPs

While imitation learning provides an excellent means to start a movement skill at a high performance level, many movement tasks require trial-and-error refinement until a satisfying skill level is accomplished. From the viewpoint of DMPs, we need non-supervised learning methods to further improve the weights $\mathbf{w}=\{\mathbf{w}_i\}$, guided by a general reward or optimization criterion. We assume that an arbitrary optimization criterion J governs our learning process, such that it is not possible to obtain analytical gradients $dJ/d\mathbf{w}$. Thus, the gradient needs to be estimated from empirical data. Levenberg-Marquardt and Gauss-Newton algorithms are a possible choice for this task [18] – however, both algorithms often dare large jumps in parameter space under the assumption that an aggressive exploration of parameters is permissible, and are not very robust in stochastic settings. As an alternative, we developed a novel reinforcement learning algorithm, the Natural-Actor Critic (NAC) [19]. The NAC is a stochastic gradient method, i.e., it injects noise in the control policy to provide the necessary exploration for learning. We will illustrate the NAC algorithm in the context of the acceleration DMP in Section 3.2.

The DMP in Equation (8) can be interpreted as creating an acceleration command $\ddot{\bar{y}} = \dot{z}$ in the top equation of (8). In the NAC, this acceleration command is treated as the mean of a Gaussian control policy

$$\pi\left(\ddot{y} \mid x, v, z, y\right) = 1\Big/\sqrt{2\pi\sigma^2}\,\exp\left(-\frac{1}{2\sigma^2}\left(\ddot{y} - \ddot{\bar{y}}\right)^2\right) \tag{13}$$

with variance σ^2. Given a reward $r(\ddot{y}, x, v, y, z)$ at every time step of the movement, the goal of learning is to optimize the expected accumulated reward $J(\mathbf{w}) = E\left\{\sum_{i=0}^{T} r_t\right\}$ where the expectation is take with respect to all trajectories starting at the same start state and following the stochastic policy above. As developed in detail in [19], the *natural* gradient $d\tilde{J}/d\mathbf{w}_i$ can be estimated by a linear regression procedure:

$$\begin{aligned} &\text{Define for one roll-out } r: && R_r = \sum_{t=0}^{T} r_{t,r} \text{ and } \varphi_r = \sum_{t=0}^{T} \left[\begin{array}{c} \partial \log \pi\left(\ddot{y}_t \mid, x_t, v_t, y_t, z_t\right) / \partial \mathbf{w} \\ 1 \end{array} \right]_r \\ &\text{After multiple roll-outs, compute } \partial \tilde{J} / d\mathbf{w}: && \Phi = \left[\begin{array}{c} \varphi_1^T \\ \varphi_2^T \\ \vdots \end{array} \right], \mathbf{R} = \left[\begin{array}{c} R_1 \\ R_2 \\ \vdots \end{array} \right], \left[\begin{array}{c} \partial \tilde{J} / d\mathbf{w} \\ c \end{array} \right] = \left(\Phi^T \Phi\right)^{-1} \Phi^T \mathbf{R} \end{aligned} \quad (14)$$

where c is the regression parameter corresponding to the constant offset in the regression. The *natural* gradient is a more efficient version of the regular gradient that takes into account the Riemannian structure of the space in which the optimization surface lies [20]. The above algorithm can also be formulated as a recursive estimation method using recursive least squares [21], and the variance σ^2 of the stochastic policy can be included in the parameters to be optimized. By updating $\mathbf{w}$ with gradient ascent (or descent) according to the natural gradient estimate, fast convergence to a locally optimal parameterization can be accomplished.

2.4 Miscellaneous Issues of DMPs

Several other issues of DMPs are worth mentioning:

- ***Invariance Properties of DMPs:*** In all the different DMP variants above, the weights $\mathbf{w}_i$ determine the particular shape of the trajectories realized by the DMP, the parameter τ the speed of a movement, and some other parameters like g or A the amplitude of the movement. In order to exploit the property that DMPs code kinematic control policies, i.e., the plans can theoretically be re-used in many parts of the workspace, it is desirable that DMPs retain their qualitative behavior if translated and if scaled in space or time. From a dynamic systems point of view, we wish that the attractor landscape of a DMP does not change qualitatively after scaling, a topic addressed in the framework of "topological equivalence" [22]. It can easily be verified, that if a new DMP is created by multiplying τ, g, or A in any of the DMPs above by a factor c, a simple multiplication of all state variables and change-of-state variables by a factor c or $\sqrt{c}$ constitutes the required homeomorphism to proof topological equivalence.
- ***Multiple Degree-of-Freedom DMPs:*** So far, our treatment of DMPs focused on single degree-of-freedom (DOF) systems. An extension to multiple DOFs is rather straightforward. The simplest form employs a separate DMP for every DOF. Alternatively, all DMPs could share the same canonical system, but have separate transformation systems, as realized in [23]. In this case, every DOF learns its own function f. By sharing the same canonical system, very complex phase relationships between individual DOFs can be realized and stabilized, for instance, as needed for biped locomotion in the examples in Section 3.3.
- ***Superposition of DMPs:*** Given that DMPs create kinematic control polices, a superposition of DMPs to generate behaviors that are more complex is possible. For instance, a discrete DMP could be employed to shift the setpoint of a rhythmic DMP, thus generating a point-to-point movement with a superimposed periodic pattern. For example, with this strategy is possible to bounce a ball on a racket by producing an oscillatory up-and-down movement in joint space of the arm, and use the discrete system to make sure the oscillatory movement remains under the ball such that the task can be accomplished [e.g., 13, 24]. Other forms of superposition are conceivable, and future work will evaluate the most promising strategies.

- ***Movement Recognition with DMPs:*** The invariance properties described above render the parameters **w** of a DMP insensitive towards movement translation and spatial and temporal scaling. Thus, the **w** vector can serve as a classifier for DMPs, e.g., by using nearest neighbor classification or more advanced classification techniques [e.g., 16]. In [10], we demonstrated how DMPs can be used for character recognition of the Palm Pilot graffiti alphabet.

3 Evaluations

The following sections give some examples of the abilities of DMPs in the context of humanoid robotics. We implemented our DMP system on a 30 DOF Sarcos Humanoid robot (Figure 1). Desired position, velocity, and acceleration information was derived from the states of the DMPs to realize a computed-torque controller (Figure 2). All necessary computations run in real-time at 420Hz on a multiple processor VME bus operated by VxWorks.

3.1 Imitation Learning

In [10], we demonstrated how a complex tennis forehand and tennis backhand swing can be learned from a human teacher, whose movements were captured at the joint level with an exoskeleton. In [9], imitation learning for a rhythmic trajectory using the phase oscillator DMP from Section 2.1 was evaluated. All DMPs referred to the same canonical system (cf. Section 2.4). Very complex phase relationships between the individual DOFs could be realized, and switching between different movement amplitudes and frequencies was easily accomplished by changing the appropriate parameters of the DMP. Due to space constraints, we ask to refer the reader to [8, 9, 23] for detailed explanations and illustrations.

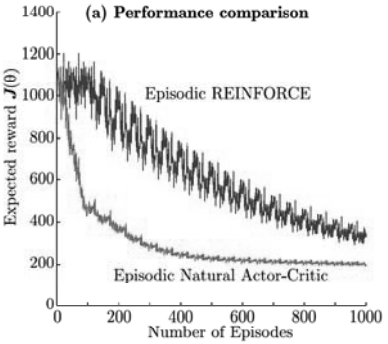

Fig. 3: Convergence of Natural Actor Critic reinforcement learning for learning the weights of a DMP.

3.2 Reinforcement Learning

In a preliminary application of the reinforcement learning method of Section 2.3, we optimized the weights of the acceleration DMP to create smooth joint-level trajectories in the spirit of 5th order polynomials [e.g., 25, 26]. The reward per trajectory was

$$R = 1000\left(\left(y(T) - g\right)^2 + \dot{y}^2(T)\right) + \sum_{t=0}^{T} \ddot{y}^2(t)$$

Weights of each DMP were initialized to zero. Each movement started at y=0 and moved within 500ms to g=1. Figure 3 illustrates the convergence of the Natural Actor Critic algorithm in comparison to a non-natural gradient method, *Episodic Reinforce* [27]. The NAC algorithm converges smoothly with about one order of magnitude faster performance than *Episodic Reinforce*. Smooth bell-shaped velocity profiles of the DMP are reached after about 150-200 trials, which is compa-

rable to human learning in related tasks [28]. This initial evaluation demonstrates the efficiency of the NAC algorithm for optimizing DMPs, although more thorough evaluations are needed for various reward criteria and also multi-DOF tasks.

3.3 Learning Resonance Tuning in Biped Locomotion

As a last evaluation of DMPs, we applied the phase oscillator DMP to simulated biped locomotion [29] of a planar biped (Figure 4). Motion capture data from human locomotion was employed to learn an initial trajectory pattern, which, after some modest tuning of the speed and amplitude parameters of the DMP achieved stable locomotion. Consider the following update law for the phase and frequency of the canonical system of the DMP in at the moment of heel-strike:

$$\dot{\phi} = \hat{\omega}^n + \delta\left(t - t_{heel-strike}\right)\left(\phi^{robot}_{heel-strike} - \phi\right); \quad \hat{\omega}^{n+1} = \hat{\omega}^n + K\left(\omega^n_{measured} - \hat{\omega}^n\right) \quad (16)$$

where δ is the Dirac delta function, n is the number of steps, $\omega = 1/\tau$, and $\phi^{robot}_{heel-strike}$ is the phase of the mechanical oscillator (robot) at heel strike defined as $\phi^{robot}_{heel-strike} = 0$ at the heel strike of the leg with the corresponding oscillator, and $\phi^{robot}_{heel-strike} = \pi$ at the heel strike of the other leg. $\omega^n_{measured}$ is the measured frequency of locomotion defined by , where T_n is the time for one step of locomotion (half period with respect to the oscillator). This equation introduces phase resetting of the DMP at heel-strike as long as the natural frequency of the robot does not correspond to the natural frequency of the canonical system of the DMP; more details can be found in [29]. Figure 5 depicts the time course of adaptation of the movement frequency of the DMPs for the right leg DOFs due to the update law above. The frequency gradually increases until it reaches approximately resonance frequency of the simulated robot legs. This simulation provides a nice example how imitation learning can initially be used to start a movement skill, and self-improvement can optimize the pattern for the particular inertial and geometric structure of the robot.

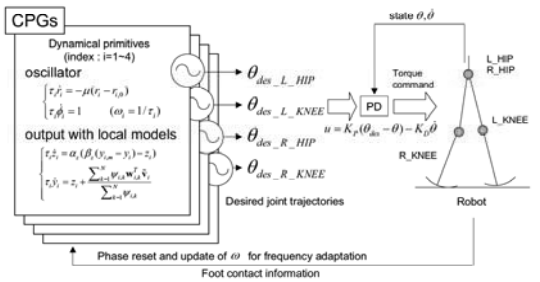

Fig. 4. Four rhythmic DMPs drive the four actuated joints of a planar biped simulation, consisting of two hip and two knee joints. A PD controller is used to track the trajectories generated by the DMPs.

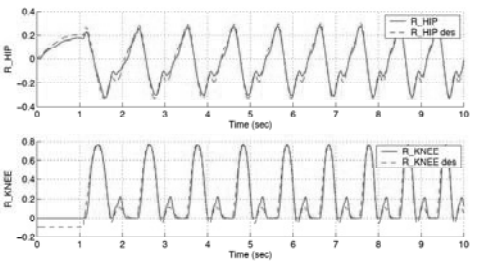

Fig. 5. Time course of trajectories of right leg DOFs using the phase resetting adaptation law. The movement frequency slowly increases until resonance tuning is accomplished.

4 Conclusion

This paper described research towards generating flexible movement primitives out of nonlinear dynamic attractor systems. We focused on motivating the design principle of appropriate dynamic systems such that discrete and rhythmic movements could be learned for high-dimensional movement systems. In particular, we emphasized methods of imitation learning and reinforcement learning for acquiring motor skills with movement primitives. We also described some implementations of our methods of dynamic movement primitives on a complex anthropomorphic robot, demonstrating imitation learning of complex rhythmic movement, re-using of learn skills in related situations, and resonance tuning for biped locomotion. We believe that the framework of dynamic movement primitives has a tremendous potential for understanding autonomous generation of complex motor behaviors in humans and humanoids.

Acknowledgments

This research was supported in part by National Science Foundation grants ECS-0325383, IIS-0312802, IIS-0082995, ECS-0326095, ANI-0224419, a NASA grant AC#98-516, an AFOSR grant on Intelligent Control, the ERATO Kawato Dynamic Brain Project funded by the Japanese Science and Technology Agency, and the ATR Computational Neuroscience Laboratories. A.I. is supported by a Young Professorship Award from the Swiss National Science Foundation.

References

1. Menzel, P. and F. D'Alusio, *Robosapiens: Evolution of a new species*. 2000: Cambridge, MA: MIT Press.
2. Nakanishi, J., J.A. Farrell, and S. Schaal, *Composite adaptive control with locally weighted statistcal learning.* International Journal of Robotics Research, submitted.
3. LaValle, S.M., *Planning algorithms*. 2003.
4. Latombe, J.-C., *Robot motion planning*. 1991, Boston: Kluwer Academic Publishers. xviii, 651 p.
5. Schaal, S., *Is imitation learning the route to humanoid robots?* Trends in Cognitive Sciences, 1999. **3**(6): p. 233-242.
6. Schaal, S., *Learning robot control*, in *The handbook of brain theory and neural networks, 2nd Edition*, M.A. Arbib, Editor. 2002, MIT Press: Cambridge, MA. p. 983-987.
7. Schaal, S., *Arm and hand movement control*, in *The handbook of brain theory and neural networks, 2nd Edition*, M.A. Arbib, Editor. 2002, MIT Press: Cambridge, MA. p. 110-113.
8. Ijspeert, A., J. Nakanishi, and S. Schaal. *Trajectory formation for imitation with nonlinear dynamical systems*. in *IEEE International Conference on Intelligent Robots and Systems (IROS 2001)*. 2001.

9. Ijspeert, A., J. Nakanishi, and S. Schaal, *Learning attractor landscapes for learning motor primitives*, in *Advances in Neural Information Processing Systems 15*, S. Becker, S. Thrun, and K. Obermayer, Editors. 2003, Cambridge, MA: MIT Press.
10. Ijspeert, J.A., J. Nakanishi, and S. Schaal. *Movement imitation with nonlinear dynamical systems in humanoid robots*. in *International Conference on Robotics and Automation (ICRA2002)*. 2002. Washinton, May 11-15 2002.
11. Sutton, R.S., D. Precup, and S. Singh, *Between MDPs and semi-MDPs: a framework for temporal abstraction in reinforcement learning.* Artificial Intelligence, 1999. **112**: p. 181-211.
12. Craig, J.J., *Introduction to robotics*. 1986, Reading, MA: Addison-Wesley.
13. Rizzi, A.A. and D.E. Koditschek. *Further progress in robot juggling: Solvable mirror laws*. in *IEEE International Conference on Robotics and Automation*. 1994. San Diego, CA.
14. Schaal, S. and D. Sternad. *Programmable pattern generators*. in *3rd International Conference on Computational Intelligence in Neuroscience*. 1998. Research Triangle Park, NC.
15. Williamson, M., *Neural control of rhythmic arm movements.* Neural Networks, 1998. **11**(7-8): p. 1379-1394.
16. Bishop, C.M., *Neural networks for pattern recognition*. 1995, New York: Oxford University Press.
17. Vijayakumar, S. and S. Schaal. *Locally weighted projection regression: An O(n) algorithm for incremental real time learning in high dimensional spaces*. in *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000)*. 2000. Stanford, CA.
18. Press, W.P., et al., *Numerical recipes in C – The art of scientific computing*. 1989, Cambridge, MA: Press Syndiacate University of Cambridge.
19. Peters, J., S. Vijayakumar, and S. Schaal. *Reinforcement learning for humanoid robotics*. in *Humanoids2003, Third IEEE-RAS International Conference on Humanoid Robots*. 2003. Karlsruhe, Germany, Sept.29-30.
20. Amari, S., *Natural gradient learning for over- and undercomplete bases In ICA*. Neural Comput, 1999. **11**(8): p. 1875-83.
21. Ljung, L. and T. Söderström, *Theory and practice of recursive identification*. 1986: Cambridge MIT Press.
22. Jackson, E.A., *Perspectives of nonlinear dynamics, Vol.1*. 1989, New York: Cambridge University Press.
23. Ijspeert, J.A., J. Nakanishi, and S. Schaal. *Learning rhythmic movements by demonstration using nonlinear oscillators*. in *IEEE International Conference on Intelligent Robots and Systems (IROS 2002)*. 2002. Lausanne, Sept.30-Oct.4 2002: Piscataway, NJ: IEEE.
24. Schaal, S., D. Sternad, and C.G. Atkeson, *One-handed juggling: A dynamical approach to a rhythmic movement task.* Journal of Motor Behavior, 1996. **28**(2): p. 165-183.
25. Sciavicco, L. and B. Siciliano, *Modeling and control of robot manipulators*. 1996, New York: MacGraw-Hill.
26. Stein, R.B., M.N. Ogusztöreli, and C. Capaday, *What is optimized in muscular movements?*, in *Human Muscle Power*, N.L. Jones, N. McCartney, and A.J. McComas, Editors. 1986, Human Kinetics Publisher: Champaign, Illinois. p. 131-150.

27. Williams, R.J., *Simple statistical gradient-following algorithms for connectionist reinforcement learning.* Machine Learning, 1992. **8**: p. 229-256.
28. Shadmehr, R. and F.A. Mussa-Ivaldi, *Adaptive representation of dynamics during learning of a motor task.* Journal of Neuroscience, 1994. **14**(5): p. 3208-3224.
29. Nakanishi, J., et al. *Learning from demonstration and adaptation of biped locomotion with dynamical movement primitives*. in *Workshop on Robot Learning by Demonstration, IEEE International Conference on Intelligent Robots and Systems (IROS 2003)*. 2003. Las Vegas, NV, Oct. 27-31.