Proceedings of the 2002 IEEE/RSJ
Intl. Conference on Intelligent Robots and Systems
EPFL, Lausanne, Switzerland • October 2002

Learning Rhythmic Movements by Demonstration using Nonlinear Oscillators

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

[1] *Dept. of Computer Science, University of Southern California, Los Angeles, USA*
[2] *ATR Human Information Science Laboratories, Kyoto, Japan*
Email: ijspeert@usc.edu, jun@his.atr.co.jp, sschaal@usc.edu

Abstract

This paper presents a new approach to the generation of rhythmic movement patterns with nonlinear dynamical systems. Starting from a canonical limit cycle oscillator with well-defined stability properties, we modify the attractor landscape of the canonical system by means of statistical learning methods to embed arbitrary smooth target patterns, however, without losing the stability properties of the canonical system. In contrast to non-autonomous movement representations like splines, the learned pattern generators remain autonomous dynamical systems which robustly cope with external perturbations that disrupt the time flow of the original pattern, and which can also be modified on-line by additional perceptual variables. A simple extension allows to cope with multiple degrees-of-freedom (DOF) patterns, where all DOFs share the same fundamental frequency but, otherwise, can move in arbitrary phase and amplitude offsets to each other. We evaluate our methods in learning from demonstration with an actual 30 DOF humanoid robot. Figure-8 and drumming movements are demonstrated by a human, recorded in joint angle space with an exoskeleton, and embedded in multi-dimensional rhythmic pattern generators. The learned patterns can be used by the robot in various workspace locations and from arbitrary initial conditions. Spatial and temporal invariance of the pattern generators allow easy amplitude and speed scaling without losing the qualitative signature of a movement. This novel way of creating rhythmic patterns could tremendously facilitate rhythmic movement generation, in particular in locomotion of robots and neural prosthetics in clinical applications.

1 Introduction

There has been a growing interest in nonlinear oscillators in both robotics and the modeling of animal motor control [3, 6, 9, 10, 13, 16], particularly for the control of rhythmic movements such as locomotion, juggling, or drumming. Nonlinear oscillators have interesting properties for rhythmic motor control, including robust limit cycle behavior, on-line adaptation through coupling signals, synchronization with other rhythmic systems, and the possibility to efficiently exploit the natural dynamics of mechanical systems through resonance tuning. However, it is difficult to design oscillator controllers for a designated task, e.g. when specific frequencies, phases, and signal shapes are needed. Attention has therefore been given to derive learning algorithms for adjusting nonlinear oscillators automatically (e.g., [4, 5, 12]). So far, most work has focused on learning the frequencies and phase relations of multiple rhythmic patterns, rather than on complete signal shaping.

This paper focuses on how to learn complex rhythmic patterns from a desired target signal, e.g., as obtained from movement recordings or learning from demonstration. We build on previous work, where we proposed to learn attractor landscapes of point attractors to form control policies (CPs) for discrete movements (e.g., a pointing motion, tennis swings toward a ball) [7]. The essence of this approach was to use a canonical simple dynamical system with well defined point attractor properties, to anchor Gaussian basis functions in phase space of this system, and to learn a weight vector that multiplies the basis functions and creates a nonlinear modulation of the canonical dynamics to form arbitrary smooth new attractor landscapes. The statistical learning system employed was based on nonparametric regression techniques [14], a method that allows a theoretically sound way of determining the number of basis function needed for accurate learning.

In this paper, we extend these ideas to learning rhythmic movements. A simple nonlinear oscillator with stable limit cycle dynamics is used as a canonical system to provide a phase signal to anchor the nonlinear basis functions, and a learned weight vector, multiplying the basis functions, together with the amplitude signal of the canonical system are used to create new, complex rhythmic patterns. Given a sample trajectory of a desired rhythmic movement, e.g., from a human demonstration, statistical learning methods [14] can be used to embed this desired pattern as a limit cycle attractor in our dynamical systems.

In the following, Section 2 first introduces our methods to learn rhythmic pattern generators for 1 DOF systems, discusses the theoretical properties of the systems, and demonstrates a possible extension to multi-dimensional pattern generators. Afterwards, in Section 3 , we will illustrate the application of our methods in experimental evaluations of learning from demonstration with an actual 30 DOF humanoid robot.

2 Nonlinear Oscillators as Control Policies

Nonlinear dynamical systems can be conceived of as control policies (CP), as they prescribe a change-of-state in a particular state in the same way as control policies prescribe a motor command in a particular state. The interesting property of dynamical systems as control policies lies in their ability to be on-line modified by external coupling variables, as well-known from the theory of coupled oscillators. The research question we pursue is whether differential equations can be used as a general tool to create control policies. For the purpose of learning a control policy from sample trajectories, we represent CPs in kinematic coordinates (similar as in [13]), e.g. joint angles of a robot, thus assuming that an appropriate controller exists to convert outputs of the kinematic policies into motor commands.

To create rhythmic control policies (RCPs), the central element of our modeling is the canonical nonlinear oscillator (see [8, 11] for very similar oscillators):

$$\dot{z} = -\frac{\mu}{E_0}\left(E - E_0\right) z - k^2 u \quad (1)$$

$$\dot{u} = z \quad (2)$$

which has a stable limit cycle characterized by the closed trajectory $\frac{z^2}{2} + \frac{k^2 u^2}{2} = E_0$ $(const)$ on the phase plane (see Figure 1), where μ, E_0 and k are positive parameters and $E(u, z) = \frac{z^2}{2} + \frac{k^2 u^2}{2}$ denotes the energy of the oscillator. Note that the first term in (1) can be considered as a nonlinear damping term which regulates the total energy of the system. The parameter k corresponds to the frequency of the oscillator, and E_0 corresponds to the desired total energy and determines the amplitude of the oscillation. μ determines the convergence rate to the limit cycle.

Based on this limit cycle oscillator, we introduce a control policy of first order dynamics with a nonlinear function f to produce a desired position y from

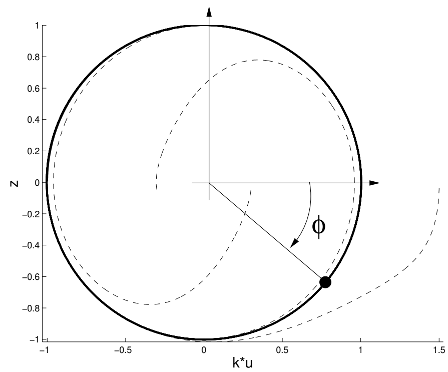

Figure 1: *Phase plot of the nonlinear oscillator* (u, z) *with* μ*=10,* k*=*2π *and* E_0*=0.5. The continuous line corresponds to the limit cycle behavior. The dotted lines show different trajectories starting with different initial conditions.*

$$\dot{y} = \beta(y_m - y) + f \quad (3)$$

where

$$f = \frac{\sum_{i=1}^{N} \Psi_i \mathbf{w}_i^T \tilde{\mathbf{z}}}{\sum_{i=1}^{N} \Psi_i} \quad (4)$$

with Gaussian kernel functions

$$\Psi_i = \exp\left(-0.5 h_i (\phi - c_i)^2\right). \quad (5)$$

β is a positive constant, and y_m is a parameter which determines the baseline around which y oscillates. The phase variable $\phi = \text{atan2}(z, ku)$ anchors the Gaussian kernel functions in the phase space of the canonical system (1,2), and $\tilde{\mathbf{z}} = [z, \sqrt{E_0}]^T$ is the driving signal for (3), generated by the limit cycle dynamics and including a bias term for better function approximation. The parameters h_i and c_i determine the location and width, respectively, of each basis function in phase space. The choice of $\tilde{\mathbf{z}}$ is motivated by our desire to have a differential equation with spatial scale invariance (see Section 2.2) and the need that $\dot{y}$ should be zero for $E_0 = 0$. The parameters $\mathbf{w}_i$ are learned to fit a given sample trajectories[1], as explained in Section 2.4. Figure 2 shows an exemplary time evolution of the complete system.

2.1 Stability of the RCP

First, we show that the (u, z) oscillator has a stable limit cycle characterized by the closed trajectory $\frac{z^2}{2} + \frac{k^2 u^2}{2} = E_0$ on the phase plane. Consider the time derivative of the energy of the oscillator

$$\dot{V} = -\frac{\mu}{E_0}(E - E_0) z^2. \quad (6)$$

[1]Note that to ensure smooth transitions from one period to the other, each Ψ_i function is in practice the sum of three gaussian functions centered on $c_i - 2\pi$, c_i, and $c_i + 2\pi$, rather than a single gaussian function centered on c_i.

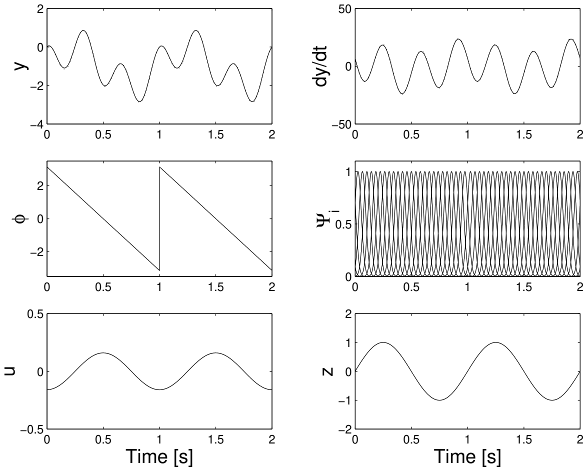

Figure 2: *Time evolution over two periods of the dynamical systems (limit cycle behavior).* N*=25,* μ*=20,* β*=3,* σ_i*=0.0386 for* i*=1,...,*N*, The* c_i *are equally spaced between* $c_1=-\pi$ *and* $c_N=\pi$*. The same parameters will be used throughout the paper. In this particular example,* $k=2\pi$*,* E_0*=0.5,* $y_m=-1.0$*, and the parameters* $\mathbf{w}_i$ *have been adjusted to fit* $y_{demo}(t) = sin(2\pi t) + cos(6\pi t)$*.*

This implies that the energy of the oscillator E monotonically converges to E_0 from any initial condition (except for the origin) and therefore $E(u, z) = E_0$ is a unique stable limit cycle of the system. The $\dot{y}$ dynamics in (3) can be interpreted simply as a first order low pass filter of a periodic input f. If the input is bounded, the output of this equation will be bounded, too.

2.2 Spatial Scale Invariance

An interesting property of the RCPs is that they are spatially invariant. Scaling of the energy of the oscillator E_0 by a factor c does not affect the topology of the attractor landscape of the policy. It can be shown that the attractor landscape with $[E_0, z, u, \phi, y_m, y]$ is topologically equivalent to the landscape with $[cE_0, \sqrt{c}z, \sqrt{c}u, \phi, \sqrt{c}y_m, \sqrt{c}y]$. Similarly, the frequency of the signal y is directly determined by the parameter k. We will exploit these properties to modulate the frequency and amplitude of learned rhythmic patterns in section 3.3.

2.3 Robustness against Perturbations

When considering applications of our approach to physical systems, e.g., robots and humanoids, interactions with the environment may require an on-line modification of the policy. As outlined in [7], the dynamical system formulation allows feeding back an error term between actual and desired positions into CPs such that the time evolution of the CP can be paused during a perturbation. In this section, we extend this idea to rhythmic control policies by modulating the original equations by the error between

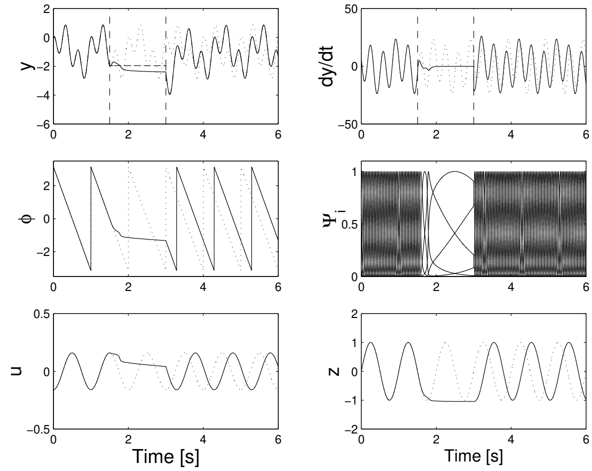

Figure 3: *Time evolution of the dynamical systems under a perturbation. The dotted and continuous lines correspond, respectively, to the unperturbed and perturbed dynamics. The actual position* $\tilde{y}$ *is frozen between 1.5 and 3.0s (dashed line in top left figure). For this example,* α_u *= 200, and* α_y *= 50.*

the planned position y and the actual position of the physical system denoted by $\tilde{y}$:

$$\dot{u} = z\left(1 + \alpha_u(\tilde{y} - y)^2\right)^{-1} \quad (7)$$

$$\dot{y} = \frac{\sum_{i=1}^{N} \Psi_i \mathbf{w}_i^T \tilde{\mathbf{z}}}{\sum_{i=1}^{N} \Psi_i} + \beta(y_m - y) + \alpha_y(\tilde{y} - y) \quad (8)$$

Figure 3 illustrates the effect of a perturbation where the actual motion of the physical system is blocked during a short period of time. During the perturbation, the time evolution of the states of the policy is gradually halted. The desired position y is modified to remain close to the actual position $\tilde{y}$, and, as soon as the perturbation stops, rapidly resumes performing the (time-delayed) planned trajectory. Note that other (task-specific) ways to cope with perturbations can be designed, which will be addressed in our future work. Such on-line modifications are one of the most interesting properties of using autonomous differential equations for control policies.

2.4 Learning of Rhythmic Control Policies

Assume we are given a target trajectory y_{demo}, e.g., from a human demonstration. For learning the attractor landscape of the control policy from the given sampled trajectory, we solve a nonlinear function approximation problem to find the parameters $\mathbf{w}_i$ in (3).

Given a sampled data point $(f_{target}, \tilde{\mathbf{z}})$ at t where

$$f_{target} = \dot{y}_{demo} - \beta(y_m - y_{demo}) \quad (9)$$

(cf. (3)), the learning problem is formulated to find the parameters $\mathbf{w}_i$ using incremental locally

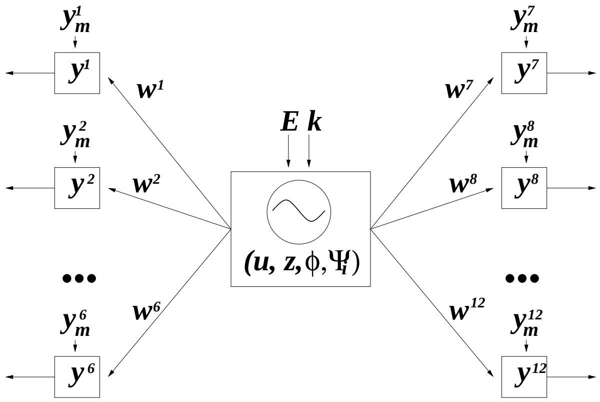

Figure 4: *Schematic view of an organization of a multiple DOF RCP. A unique (u, z) oscillator is used to generate the rhythmic drive for 12 DOFs (as will be needed in our drumming experiment below).*

weighted regression technique [14] in which $\mathbf{w}_i$ is updated by

$$\mathbf{w}_i^{t+1} = \mathbf{w}_i^t + \mathbf{P}_i^{t+1}\tilde{\mathbf{z}}e_i \tag{10}$$

where

$$\mathbf{P}_i^{t+1} = \frac{1}{\lambda}\left(\mathbf{P}_i^t - \frac{\mathbf{P}_i^t\tilde{\mathbf{z}}\tilde{\mathbf{z}}^T\mathbf{P}_i^t}{\frac{\lambda}{\Psi_i} + \tilde{\mathbf{z}}^T\mathbf{P}_i^t\tilde{\mathbf{z}}}\right),\ e_i = f_{target} - \mathbf{w}_i^T\tilde{\mathbf{z}}$$

and $\lambda \in [0, 1]$ is the forgetting factor. We chose this locally weighted regression framework as it can automatically find the correct number of necessary basis function, can tune the h_i parameters of each Gaussian basis function (5) to achieve higher function approximation accuracy, and, most importantly, learns the parameters $\mathbf{w}_i$ of every local model i totally independently of all other local models. The latter property creates very consistent parameters for similar patterns that can be use for classification of different rhythmic movements [7].

2.5 Multi-dimensional Rhythmic Patterns

For multi-dimensional rhythmic patterns, it is important to also provide a mechanism for the stable coordination between the individual DOFs participating in the pattern. A possible solution is suggested in Figure 4. Here, the canonical oscillator (1, 2) is used to generate the rhythmic (u, z) dynamics for *all* DOFs—the parameters $\mathbf{w}_i$ are learned individually for each DOF, and each DOF has its own equation (3). By manipulating E_0 and k in the oscillator, the amplitude and frequency of all DOFs can be modulated, while keeping the same phase relation between the DOFs. This form of modulation might be particularly useful, for instance, in a drumming task in order to replay the same beat pattern at different speeds.

3 Experimental Evaluations

We tested the proposed RCPs in a learning by demonstration task using a humanoid robot, which is a 1.9-meter tall 30 DOF hydraulic anthropomorphic robot with legs, arms, a jointed torso, and a head [2]. We recorded a set of rhythmic movements such as tracing a figure-8, or a drumming sequence on a bongo performed by a human subject using a joint-angle recording system, the Sarcos Sensuit (see Figure 5 left). Six degrees of freedom for both arms were recorded (three at the shoulder, one at the elbow, and two at the wrist). The recorded joint-angle trajectories are fitted by the RCPs, with one set of weights per DOF using the multi-DOF RCP in Figure 4. The RCPs are then used by the humanoid robot to imitate the movement. Sections 3.1 and 3.2 show the experimental results of learning by imitation of figure-8 and drumming tasks. Section 3.3 discusses how the learned movements can be modulated using the proposed RCPs.

3.1 Learning of Figure-8 Movements

Figure 5 (left) shows the human demonstration of a figure-8 movement, and Figure 5 (right) shows the robot performance of the learned movement using the proposed RCPs. Figure 6 illustrates the recorded trajectories of the figure-8 performed by the robot. Movies of the human demonstration and the robot performance can be found at [1]. The movies also demonstrate robustness against perturbations imposed by a human interfering with the robot's execution of the pattern: the robot pauses the figure-8 motion when its arm is blocked, and resumes the motion where it was stopped when the perturbation vanishes.

3.2 Learning of Drumming Movements

In this experiment, we recorded movements which look like a drumming sequence on a bongo (i.e. without drumming sticks). Figure 7 shows the joint trajectories over one period of an exemplary drumming beat. Demonstrated and learned trajectories are superposed. For learning, the base frequency was extracted manually such as to provide the parameter k to the RCP. Note that the beat includes higher frequency components for the right arm, with the right hand hitting twice the bongo during the baseline period. Movies of the human demonstration and the robot performance can be found at [1].

3.3 Modulation of Learned Movements

Once a rhythmic movement has been learned by the RCP, it can be modulated in several ways. Figure 8 illustrates an example of different modulations of the recorded trajectory (A), which were generated by

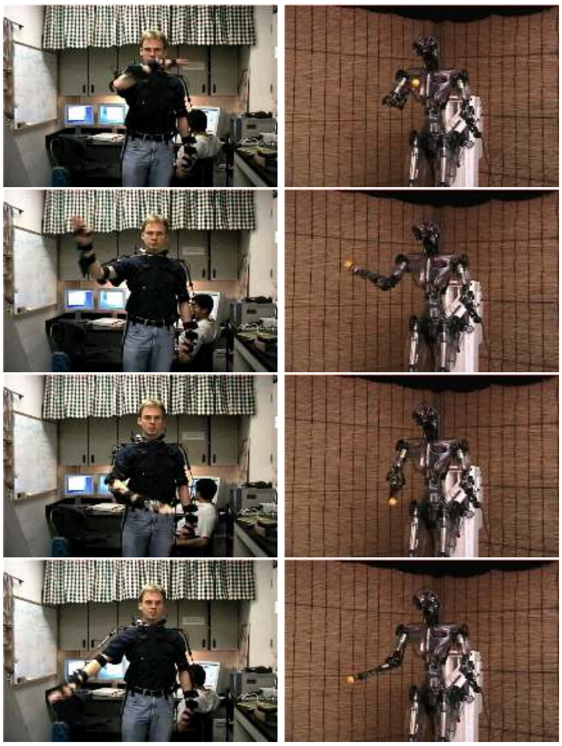

Figure 5: Humanoid robot learning a figure-8 movement from a human demonstration. Left: human demonstration. Right: robot execution. Recording of the robot performance is shown in Figure 6.

changing in the time window between 3-7 seconds the RCP's amplitude (B), frequency (C) and spatial mid point (D), respectively—due to space constraints, only one degree of freedom is shown. In all examples, changes are smooth and rapid, as it is desirable for such behaviors. Movies showing the effect of frequency and amplitude modulations on the figure-8 and drumming movements can be viewed at [1].

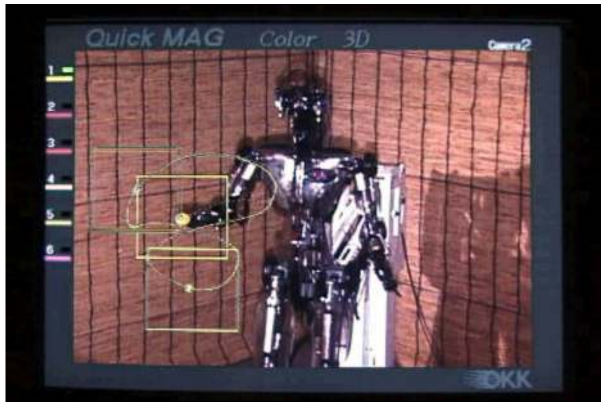

Figure 6: Recorded tip trajectory of the hand of robot performing the learned figure-8 movement shown in Figure 5.

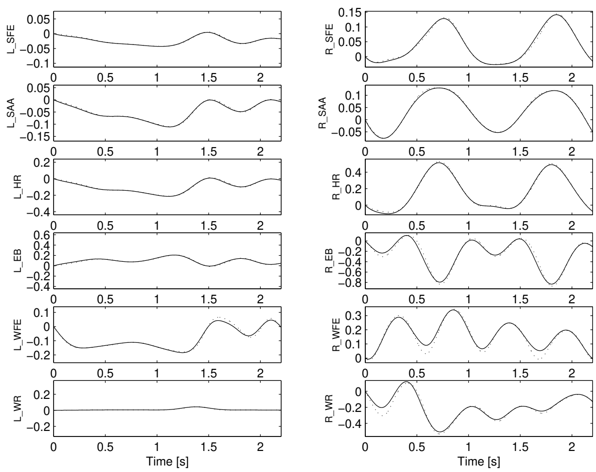

Figure 7: Recorded drumming movement performed with both arms (6 degrees of freedom per arm). The dotted lines and continuous lines correspond to the demonstrated and learned trajectories, respectively.

4 Discussion

This paper presented a method for learning rhythmic patterns based on nonlinear oscillators. The central idea of our approach is to encode complex oscillatory patterns based on modulating a canonical simple limit cycle system with statistical learning methods, using nonlinear basis functions anchored in phase space of the canonical system. Despite the resulting dynamical system becomes strongly nonlinear, stability properties of the canonical system are preserved. The final dynamical system can be conceived of as a kinematic control policy with a global limit cycle attractor.

Learning control policies based on a dynamical systems approach has various desirable properties. Firstly, since movement plans generated by the dynamical systems are not explicitly indexed by time, i.e, develop out of the time evolution of autonomous differential equations, flexible on-line modification of the control policies can be accomplished by means of coupling terms to the differential equations. We demonstrated one such example by incorporating the tracking error of a robotic systems as an inhibitory variable in the control policies, thus ensuring that the control policy cannot create movement plans that are not realizable by the robot. Other perceptual variables could be used to create different forms of on-line modifications, e.g., based on contact forces in locomotion or perceptual variables in juggling. Secondly, although the output of the canonical limit cycle oscillator has a very simple signal shape, the proposed framework can fit almost arbitrarily complex smooth signals using the function

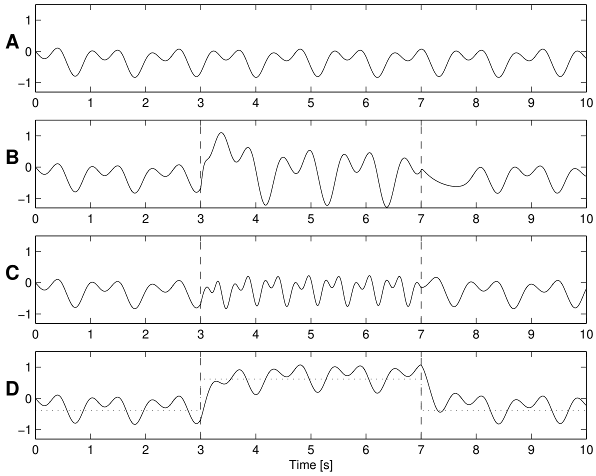

Figure 8: *Modification of the learned rhythmic pattern (flexion/extension of the right elbow, R_EB, see Figure 7).* **A:** *trajectory learned by the RCP,* **B:** *amplitude modification with* $\tilde{E}_0 = 4E_0$*,* **C:** *frequency modification with* $\tilde{E}_0 = 4E_0$ *and* $\tilde{k} = 2k$*,* **D:** *spatial modification with* $\tilde{y_m} = y_m+1$ *(dotted line), where* $\tilde{E}$*,* $\tilde{k}$*, and* $\tilde{y_m}$ *correspond to modified parameters between 3 and 7s.*

approximation framework of locally weighted learning [14]. These learning methods also allow us to automatically determine the open parameters of the learning system, i.e., the number of kernel functions, their center location and bandwidth (a topic that we did not expand on due to space limitations). And lastly, due to the meaningful parameterization of the dynamical systems, the learned trajectories can easily be modified with respect to amplitude, frequency, and the midpoint of the rhythmic pattern. Spatial and temporal invariance of the differential equations ensure that such scaling does not affect the qualitative shape of the rhythmic patterns.

Future work will address how to automatically extract the base frequency of a sample trajectory (e.g., by FFT analyses, learning approaches, or oscillator synchronization [15]), how to integrate the rhythmic control policies suggested in this paper with discrete control policies from previous work for general imitation learning, and how to incorporate the ability of using reinforcement learning to further modify a learned pattern by trial and error.

Acknowledgments

This work was made possible by support from the US National Science Foundation (Awards 9710312 and 0082995), the ERATO Kawato Dynamic Brain Project funded by the Japan Science and Technology Corporation, the ATR Human Information Sciences Laboratories, and the Communications Research Laboratory (CRL).

References

[1] http://rana.usc.edu:8736/˜ijspeert/humanoid.html.

[2] C. G. Atkeson, J. Hale, M. Kawato, S. Kotosaka, F. Pollick, M. Riley, S. Schaal, S. Shibata, G. Tevatia, and A. Ude. Using humanoid robots to study human behaviour. *IEEE Intelligent Systems*, 15:46–56, 2000.

[3] R.D. Beer, H. J. Chiel, R. D. Quinn, and R. E. Ritzmann. Biorobotic approaches to the study of motor systems. *Curr. Opin. Neurobiol.*, 8:777–782, 1998.

[4] K. Doya and S. Yoshizawa. Adaptive synchronization of neural and physical oscillators. In J.E. Moody, S.J. Hanson, and R.P. Lippmann, editors, *Advances in Neural Information Processing Systems*. Morgan Kaufmann, 1992.

[5] B. Ermentrout and N. Kopell. Learning of phase lags in coupled neural oscillators. *Neural Computation*, 6:225–241, 1994.

[6] A.J. Ijspeert. A connectionist central pattern generator for the aquatic and terrestrial gaits of a simulated salamander. *Biological Cybernetics*, 85(5):331–348, 2001.

[7] A.J. Ijspeert, J. Nakanishi, and S. Schaal. Movement imitation with nonlinear dynamical systems in humanoid robots. In *IEEE International Conference on Robotics and Automation (ICRA2002)*, pages 1398–1403. 2002.

[8] E. Klavins and D. Koditschek. Stability of coupled hybrid oscillators. In *IEEE International Conference on Robotics and Automation*, pages 4200–4207. 2001.

[9] S. Kotosaka and S. Schaal. Synchronized robot drumming by neural oscillator. In *The International Symposium on Adaptive Motion of Animals and Machines. Montreal, Canada.* 2000.

[10] E. Marder. Motor pattern generation. *Curr. Opin. Neurobiol.*, 10:691–698, 2000.

[11] J. Nakanishi, T. Fukuda, and D.E. Koditschek. A brachiating robot controller. *IEEE Transactions on Robotics and Automation*, 16(2):109–123, 2000.

[12] J. Nishii. A learning model for oscillatory networks. *Neural Networks*, 11:249–257, 1998.

[13] A.A. Rizzi and D.E. Koditschek. Further progress in robot juggling: Solvable mirror laws. In *IEEE International Conference on Robotics and Automation*, pages 2935–2940. 1994.

[14] S. Schaal and C.G. Atkeson. Constructive incremental learning from only local information. *Neural Computation*, 10(8):2047–2084, 1998.

[15] S. Schaal, S. Kotosaka, and D. Sternad. Nonlinear dynamical systems as movement primitives. In *International Conference on Humanoid Robotics. Cambridge, MA, Sept 6-7, 2001.* 2001. in Press.

[16] G. Taga, Y. Yamaguchi, and H. Shimizu. Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment. *Biological Cybernetics*, 65:147–159, 1991.