

## Nonlinear Dynamical Systems as Movement Primitives

Stefan Schaal<sup>‡, #</sup>

sschaal@usc.edu  
http://www-slab.usc.edu/sschaal

Shinya Kotosaka<sup>‡</sup>

kotosaka@erato.atr.co.jp  
http://www.erato.atr.co.jp/~kotosaka

Dagmar Sternad<sup>\*</sup>

dxs48@psu.edu  
http://www.psu.edu/faculty/d/x/dxs48

<sup>#</sup>Computer Science and Neuroscience, University of Southern California, Los Angeles, CA 90089-2520, USA

<sup>\*</sup>Department of Kinesiology, Pennsylvania State University, 266 Recreation Building, University Park, PA 16802, USA

<sup>‡</sup>Kawato Dynamic Brain Project (ERATO/JST), 2-2 Hikaridai, Seika-cho, Soraku-gun, 619-02 Kyoto, Japan

**Abstract.** This paper explores the idea to create complex human-like movements from movement primitives based on nonlinear attractor dynamics. Each degree-of-freedom of a limb is assumed to have two independent abilities to create movement, one through a discrete dynamic system, and one through a rhythmic system. The discrete system creates point-to-point movements based on internal or external target specifications. The rhythmic system can add an additional oscillatory movement relative to the current position of the discrete system. In the present study, we develop appropriate dynamic systems that can realize the above model, motivate the particular choice of the systems from a biological and engineering point of view, and present simulation results of the performance of such movement primitives. The model was implemented for a drumming task on a humanoid robot.

### Introduction

When searching for a general framework of how to formalize the learning of coordinated movement, some of the ideas developed in the middle of the 20th century still remain useful. At this time, theories from optimization theory, in particular in the context of dynamic programming (Bellman, 1957; Dyer & McReynold, 1970), described the goal of learning control in learning a policy. A policy is formalized as a function that maps the continuous state vector  $\mathbf{x}$  of a control system and its environment, possibly in a time dependent way, to a continuous control vector  $\mathbf{u}$ :

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

The parameter vector  $\alpha$  denotes the problem specific adjustable parameters in the policy  $\pi$ —not unlike the parameters in neural network learning. At the first glance, one might suspect that not much was gained by this overly general formulation. However, 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  $\pi$  for complex nonlinear control problems. Unfortunately, as already noted in Bellman's original work, learning of  $\pi$  becomes computationally intractable for even moderately high dimensional state-action spaces. Although recent developments in reinforcement learning increased the range of complexity that can be dealt with (e.g., [1]; [2]; [3]), it still seems that there is a long if not impossible way to go to apply general policy learning to complex control problems.

In most robotics applications, the full complexity of learning a control policy is strongly reduced by providing prior information about the policy. The most common priors are in terms of a desired trajectory,  $[\mathbf{x}_d(t), \dot{\mathbf{x}}_d(t)]$ , usually hand-crafted by the insights of a human expert. For instance, by using a PD controller, a (explicitly time dependent) control policy can be written as:

$$\begin{aligned} \mathbf{u} &= \pi(\mathbf{x}, \alpha(t), t) = \pi(\mathbf{x}, [\mathbf{x}_d(t), \dot{\mathbf{x}}_d(t)], t) \\ &= \mathbf{K}_x(\mathbf{x}_d(t) - \mathbf{x}) + \mathbf{K}_x(\dot{\mathbf{x}}_d(t) - \dot{\mathbf{x}}) \end{aligned} \quad (2)$$

For problems in which the desired trajectory is easily generated and in which the environment is static or fully predictable, as in many industrial applications, such a shortcut through the problem of policy generation is highly successful. However, since policies like in (2) are usually valid only in a local vicinity of the time course of the desired trajectory, they are not very flexible. When dealing with a dynamically changing environment in which substantial and reactive modifications of control commands are required, one needs to modify trajectories appropriately, or even generate entirely new trajectories by generalizing from previously learned

knowledge. In certain cases, it is possible to apply scaling laws in time and space to desired trajectories ([4]; [5]), but those can provide only limited flexibility, as similarly recognized in related theories in psychology ([6]). Thus, for general-purpose reactive movement, the “desired trajectory” approach seems to be too restricted.

From the viewpoint of statistical learning, Equation (1) constitutes a nonlinear function approximation problem. A typical approach to learning complex nonlinear functions is to compose them out of basis functions of reduced complexity. The same line of thinking generalizes to learning policies: a complicated policy could be learned from the combination of simpler (ideally globally valid) policies, i.e., policy primitives or movement primitives, as for instance:

$$\mathbf{u} = \pi(\mathbf{x}, \alpha, t) = \sum_{k=1}^K \pi_k(\mathbf{x}, \alpha_k, t) \quad (3)$$

Indeed, related ideas have been suggested in various fields of research, for instance in computational neuroscience as Schema Theory ([7]) and in mobile robotics as behavior-based or reactive robotics ([8]). In particular the latter approach also emphasized to remove the explicit time dependency of  $\pi$ , such that complicated “clocking” and “reset clock” mechanisms could be avoided, and the combination of policy primitives became simplified. Despite the successful application of policy primitives in the mobile robotics domain, so far, it still remains unclear how to generate and combine those primitives in a principled and autonomous way, and how such an approach generalizes to complex movement systems, like human arms and legs.

Thus, a key research topic, both in biological and artificial motor control, revolves around the question of movement primitives: 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 of primitives? In order to address the first four of these questions, we suggest to resort to some of the most basic ideas of dynamic systems theory. The two most elementary behaviors of a nonlinear dynamic system are point attractive and limit cycle behaviors, paralleled by discrete and rhythmic movement in motor control. Would it be possible to generate complex movement just out of these two basic elements? The idea of using dynamic systems for movement generation is not new: motor pattern generators in neurobiology ([9]), pattern generators for locomotion ([10]; [11]), potential field approaches for planning (e.g., [12]), and more recently basis field approaches for limb movement ([13]) have been published. Additionally, work in the dynamic systems approach in psychology ([14]; [15]) has emphasized the usefulness of autonomous nonlinear differential equations to describe movement behavior. However, rarely have these ideas addressed both rhythmic and discrete movement in one framework, task specific planning that can exploit both intrinsic (e.g., joint) coordinates and extrinsic (e.g., Cartesian) coordinate frames, and more general purpose behavior, in particular for multi-joint arm movements. It is in these domains, that the present study offers a novel framework of how movement primitives can be formalized and used, both in the context of biological research and robotics research.

## Programmable Pattern Generators

Using nonlinear dynamic systems as policy primitives is the most closely related to the original idea of motor pattern generators (MPG) in neurobiology. MPGs are largely thought to be hardwired with only moderately modifiable properties. In order to allow for the large flexibility of human limb control, the MPG concept needs to be augmented by a component that can be adjusted task specifically, thus leading to programmable pattern generators (PPG). Given a set of parameters  $\alpha$ , a PPG realizes a policy primitive that implements a globally stable attracting regime whose specifics are determined by the particular values of  $\alpha$ .

We assume that the attractor landscape of PPGs represent desired kinematic state of a limb, e.g., positions, velocities, and accelerations. This approach deviates from MPGs which are usually assumed to code motor commands, and is strongly related to the idea developed in the context of “mirror laws” by Bühler, Rizzi, and Koditschek ([16]; [17]). In our current scheme, kinematic variables are converted to motor commands through an inverse dynamics model and stabilized by low gain feedback control. The motivation for this approach is largely inspired by data from neurobiology that demonstrated strong evidence for the representation of kinematic trajectory plans in parietal cortex ([18]) and inverse dynamics models in the cerebellum ([19]; [20]). Kinematic trajectory plans are equally backed up by the discovery of the principle of motor equivalence in

psychology (e.g., [21]), demonstrating that different limbs (e.g., fingers, arms, legs) can produce kinematically similar patterns despite having very different dynamical properties; these findings are incompatible with direct planning in motor command space. Kinematic trajectory plans, of course, are also well known in robotics from the computed torque control schemes ([22]). From the view point of policy primitives, kinematic representations are more advantageous than direct motor command coding since this allows for workspace independent planning, and, importantly, for the possibility to superimpose PPGs. However, it should be noted that a kinematic representation of policy primitives is not necessarily independent of dynamic properties of the limb. Proprioceptive feedback can be used to modify the attractor landscape of a PPG in the same way as perceptual information ([17]; [23]; [24]).

## Formalization of PPGs

In order to accommodate discrete and rhythmic movements, two kinds of PPGs are needed, a point attractive PPG and a limit cycle PPG. Although it is possible to construct nonlinear differential equations that could realize both these behaviors in one set of equations (e.g., [25]), for reasons of robustness, simplicity, functionality, and biological realism, we chose an approach that separates these two regimes. Every degree-of-freedom (DOF) of a limb is described by two variables, a rest position  $\theta_o$  and a superimposed oscillatory position,  $\theta_r$ , as shown in Figure 1. By moving the rest position, discrete motion is generated. The change of rest position can be anchored in joint space or, by means of inverse kinematics transformations, in external space. In contrast, the rhythmic movement is produced in joint space, relative to the rest position. This dual strategy permits to exploit two different coordinate systems: joint space, which is the most efficient for rhythmic movement, and external (e.g., Cartesian) space, which is needed to reference a task to the external world. For example, it is now possible to bounce a ball on a racket by producing an oscillatory up-and-down movement in joint space, but using the discrete system to make sure the oscillatory movement remains under the ball such that the task can be accomplished—this task actually motivated our current research ([26]).

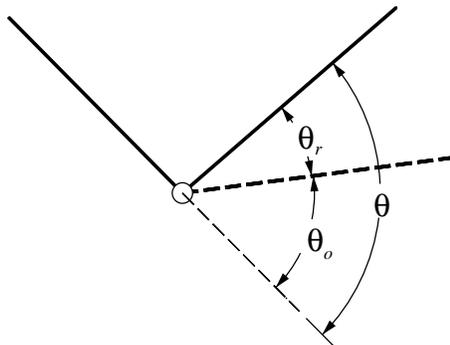


Figure 1: Each degree-of-freedom of a limb has a rest state  $\theta_o$  and an oscillatory state  $\theta_r$ .

## The Discrete PPG

Discrete movement is generated by a set of weakly nonlinear differential equations, closely related to the VITE model by Bullock and Grossberg ([27]). The modeling strategy is to use first-order differential equations (“leaky integrators”) as basis for the development—similar to abstract models of biological neurons—and to augment these equations with nonlinear terms such that an attractor landscape is created that produces smooth trajectory profiles between start and target states. In contrast to VITE, our dynamic system does not require artificial resetting of certain states of the attractor model after each movement, as all states of the dynamic system converge to their initial states after the movement terminates. Future work will address how to learn such dynamical systems from unstructured networks—however, the scope of this paper is to demonstrate which ingredients are needed in a dynamic network to produce the desired attractor landscapes. With muscle-based actuation in mind, the following equations model the discrete PPG for an antagonistically actuated 1 DOF joint:

$$\begin{aligned}\Delta v_i &= [t_i - \theta_{o,i}]^+ \\ \dot{v}_i &= a_v(-v_i + \Delta v_i)\end{aligned}\tag{4}$$

$$\dot{x}_i = -a_x x_i + (v_i - x_i + C_i) c_o \quad (5)$$

$$\dot{y}_i = -a_y y_i + (x_i - y_i) c_o$$

$$\dot{r}_i = a_r (-r_i + (1 - r_i) b v_i) \quad (6)$$

$$\dot{z}_i = -a_z z_i + (y_i - z_i)(1 - r_i) c_o \quad (7)$$

$$\dot{\theta}_{o,i} = a_p ([z_i]^+ - [z_j]^+) c_o \quad (8)$$

where  $i \in \{1, 2\}$  and  $j \in \{2, 1\}$ , indicating the agonist and antagonist and their reciprocal influence, and where  $[\cdot]^+$  denotes a threshold function that sets negative values to 0 while not affecting positive values. Equations (4) build a difference vector  $\Delta v_i$  between the target position  $t_i$  and the current position  $p_i$  of each muscle and pass this difference vector through a first order differential equation, thus simulating an activation pattern in  $v_i$  that resembles signals observed in the primate cortex ([27]). Equations (5) accomplish a double smoothing of  $v_i$  with  $c_o$  acting as an amplifier of the time constants  $a_x$  and  $a_y$ . Indeed,  $c_o$  will allow adjusting the speed of the movement, as shown below.  $C_i$  stands for the possibility to couple additional external signals to these differential equation—for the purpose of this paper,  $C_i$  can be assumed to be zero. The goal of the discrete PPG is to achieve a trajectory with a roughly symmetric bell-shaped velocity profile, similar to those observed in humans (e.g., [28]). At the stage of Equation (5), we interpret  $y_i$  as a velocity signal which, due to the exponential convergence of the first-order dynamics of Equations (4) and (5), displays a smooth but quite asymmetric profile. Equation (6) provides a signal that can correct this behavior. Given appropriate parameters  $a_r$  and  $b$ ,  $r$  very quickly “jumps” to a value of almost “1” and then decreases smoothly back to zero. This signal can be used as a time constant adjustment in Equation (7): initially it reduces the time constant of this equation and later it causes an increase. This effect exactly counteracts the initially fast and subsequent slow phase of first-order differential equations. The signal  $z_i$  is interpreted as an unscaled desired velocity signal that is finally adjusted by the pure integrator in Equation (8). Figure 3 shows all the signals of the discrete PPG in a 0.7 s point-to-point movement of 1 rad distance. The time course of the signals should be compared with the description above. For simplification, we assume that the current position and target of each muscle are identical but just of the opposite sign.

Figure 2 shows the output of the discrete PPG for three different movement speeds, otherwise using the same parameters as in Figure 3. With increasing movement speed, some transient overshoot of the target starts to appear. This effect is quite similar as in human reaching movement and, for many movement tasks, does not cause any problems.

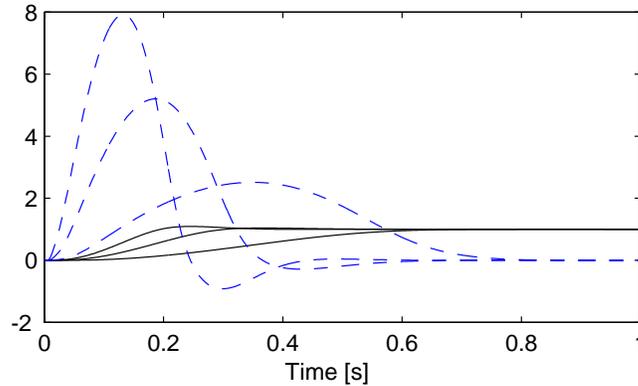


Figure 2: Position (solid lines) and velocity (dashed lines) traces for three different movement velocities, accomplished by setting  $c_o$  to 50, 100, and 150, respectively.

Extending the discrete PPG to multiple DOFs is easily accomplished by allocating one antagonistic PPG per degree-of-freedom.  $c_o$  is kept the same for all DOFs, while the target positions, of course, would vary for every DOF. Such a scheme would produce a multi-joint PPG that generates a “joint-interpolation” policy primitive ([29]). The formulation of the discrete PPG in terms of directional signals (cf. Equation (4)) bears the advantage that it is straightforward to use goals defined in Cartesian space ([30]). By using a Jacobian-based inverse kinematics scheme, e.g., the pseudo-inverse or the Extended Jacobian method ([31]), it is possible to transform a difference vector  $\Delta v_c$  in Cartesian space into the difference vector  $\Delta v_i$  in joint space for every DOF. This

inverse kinematics-based difference vector would replace the first equation in Equation (4), after appropriately adding a sign adjustment for the antagonistic formulation.

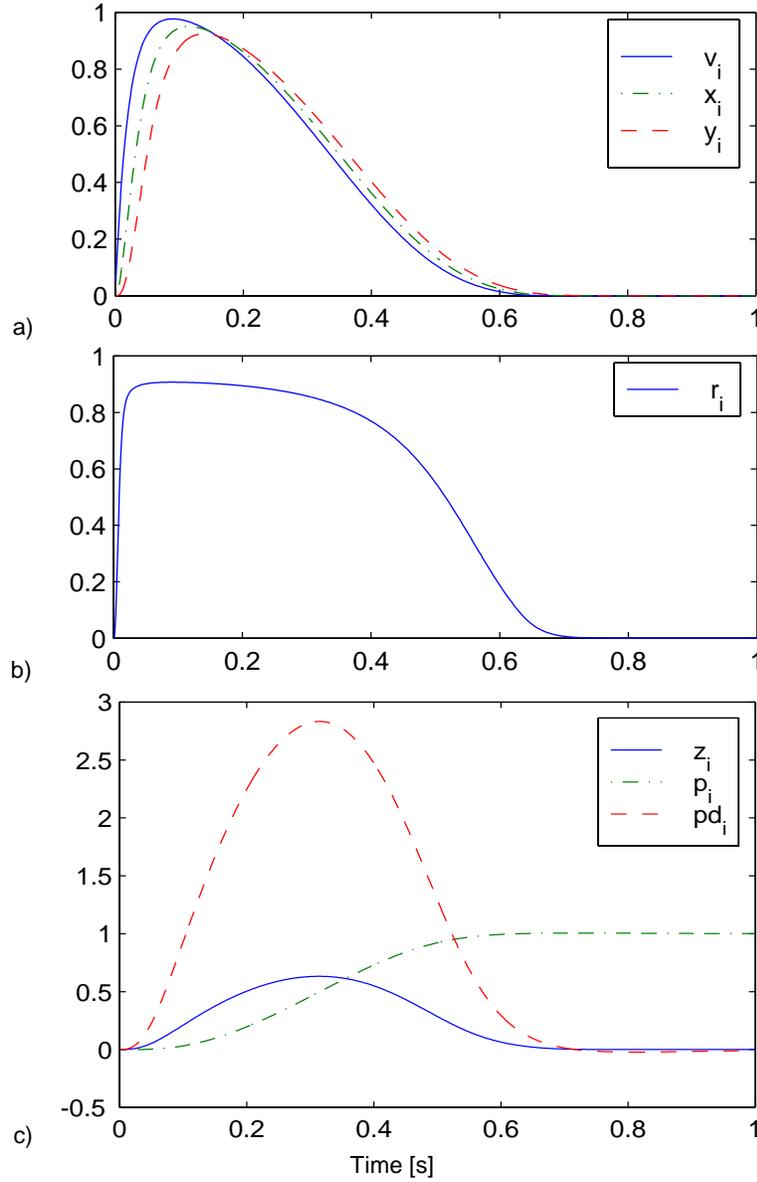


Figure 3: Time course of discrete PPG for a 0.7 s movement. Parameter settings are:  $a_v = a_r = 50$ ,  $a_x = a_y = 1$ ,  $a_z = 0.01$ ,  $a_p = 0.08$ , and  $b = 10$ . The target was  $t_1 = -t_2 = 1$ , starting from an initial position  $\theta_o = \theta_{o,1} = -\theta_{o,2} = 0$ . The speed factor was set to  $c_o = 60$ .

### The Rhythmic PPG

Using the same modeling strategy as in the previous section, a dynamic policy primitive can be created that displays limit cycle behavior. The following equations are based on a half-centered oscillator model (i.e., two mutually inhibitory units, Brown, 1914) suggested by Matsuoka (1985, 1987), and similarly employed in ([23]; [24]):

$$\Delta\omega_i = [A - \theta_{r,i}]^+ \quad (9)$$

$$\dot{\xi}_i = a_\xi(-\xi_i + \Delta\omega_i)$$

$$\dot{\psi}_i = -a_\psi\psi_i + \left(\xi_i - \psi_i - \beta\zeta_i - w[\psi_j]^+ + K_i\right)c_r \quad (10)$$

$$\dot{\zeta}_i = -\frac{a_\psi}{5}\zeta_i + \left([\psi_i]^+ - \zeta_i\right)\frac{c_r}{5}$$

$$\dot{\theta}_{r,i} = \psi_i \quad (11)$$

$$\theta_r = c_r[\theta_{r,1}]^+ - c_r[\theta_{r,2}]^+$$

Equations (9) are the equivalent of Equations (4): given an amplitude signal  $A$ , the difference between the current position and the desired amplitude is calculated and passed through a first order differential equation. Equations (10) are the original Matsuoka equations, except that we formulated them such that  $\psi_i$  is interpreted as a velocity signal instead of a position signal as in Matsuoka's original concept ([32]). Important in these equations are the inhibitory coupling from the antagonistic unit,  $\psi_j$ , the coupling term to external signals,  $K_i$ , and the velocity factor  $c_r$  that determines the frequency of the oscillator. The second equation in (10) can be interpreted in terms of an adaptation of the activation  $\psi_i$  as observed in biological units. Matsuoka ([33]) explains the motivation and stability properties of Equations (10) in detail. Equations (11) have a pure integrator for the position  $\theta_{i,r}$  of the oscillator, equivalent to Equation (8), and combine the positive parts of agonist and antagonist to result in the rhythmic position signal  $\theta_r$ . Figure 4 shows position and velocity of a 1 DOF oscillatory movement. Since the limit cycle oscillator codes velocity, position traces become quite smooth and sinusoidal due to the integration of the more uneven velocity signal. It should be noted how quickly a steady limit cycle oscillation is achieved starting from time  $t=0$ . This behavior is accomplished by "priming" one unit of the rhythmic PPG with a small inhibitory (negative) signal through the coupling term  $K_i$ , thus breaking the symmetry in the equations. Such priming also allows determining whether the oscillation should start in a positive direction or negative direction.

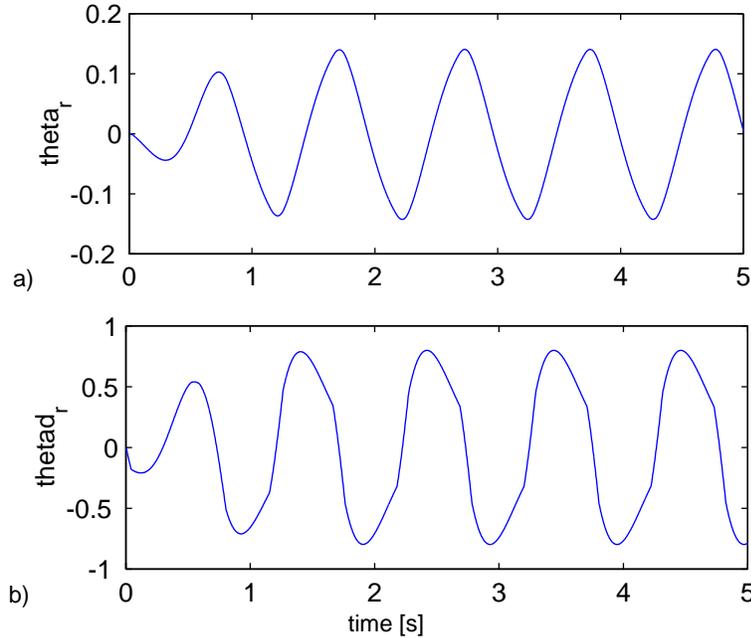


Figure 4: Time course of the rhythmic PPG: a) position of oscillator, b) velocity of oscillator. Parameter settings were  $a_\xi = 50$ ,  $a_\psi = 1$ ,  $\beta = 1$ ,  $w = 2$ , for a speed parameter of  $c_r = 5$  and  $A = 0.3$ .

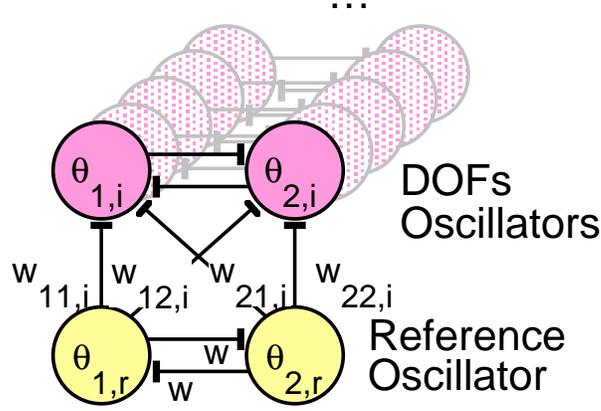


Figure 5: Multi-joint rhythmic PPG

### A Multi-Joint Rhythmic PPG

In order to generate rhythmic movement with a multi-joint limb, it is necessary to couple the individual rhythmic PPGs such that they remain phase-locked. Moreover, parameters have to be provided to adjust the phase-offset between individual DOFs, as needed for certain rhythmic movements and also observed in human behavior ([34]). These two requirements can be fulfilled by an appropriate coupling structure between the individual oscillators, illustrated in Figure 5. In this oscillator network, we introduced a “Reference Oscillator” to which every DOF refers in order to adjust its phase offset, and only a unidirectional influence from the reference oscillator to the DOFs exists. The connection scheme bears some important advantages over alternatives, e.g., all-to-all coupling, or chain-like coupling. All-to-all coupling requires a highly redundant set of phase offset parameters: for a 7 DOF arm, 7 phase-offsets uniquely determine the oscillatory pattern—all-to-all coupling would specify an overcomplete set of  $7^2 = 49$  parameters. Chain-like coupling avoids this problem, however, if one intermediate oscillator in the chain has small or no amplitude, the synchronization process would be interrupted and not proceed to the end of the chain. The idea of a reference Figure 5 avoids these problems. By means of the four connection weights, a range of different phase offsets can be achieved. The phase information from the reference oscillator enters Equation (10) through the external coupling  $K_{j,i}$ :

$$\begin{aligned}
 K_{1,i} &= -A_i \left( w_{11,i} [\theta_{r,1}]^+ + w_{21,i} [\theta_{r,2}]^+ \right) \\
 K_{2,i} &= -A_i \left( w_{12,i} [\theta_{r,1}]^+ + w_{22,i} [\theta_{r,2}]^+ \right) \\
 w_{11,i} &= \gamma_i w_c \quad w_{12,i} = (1 - \gamma_i) w_c \quad \text{where } \gamma_i \in [0,1] \\
 w_{21,i} &= (1 - \gamma_i) w_c \quad w_{22,i} = \gamma_i w_c
 \end{aligned}$$

where  $i$  indexes the DOF, and  $j$  the unit in an oscillator.  $\gamma_i = 0$  generates zero phase offset, while  $\gamma_i = 1$  will result in an offset of  $\pi$ . Intermediate offset values are achieved by intermediate values of  $\gamma_i$ ; Figure 6 provides some examples. The coupling base weight  $w_c$  is constant for all DOFs. The amplitude  $A_i$  adjusts the coupling weights according to the desired amplitude of each DOF since the reference oscillator is chosen to have unit amplitude. Because coupling from the reference oscillator to all DOFs is unidirectional, there is no interference between the individual DOFs, thus resulting a robust oscillatory network.

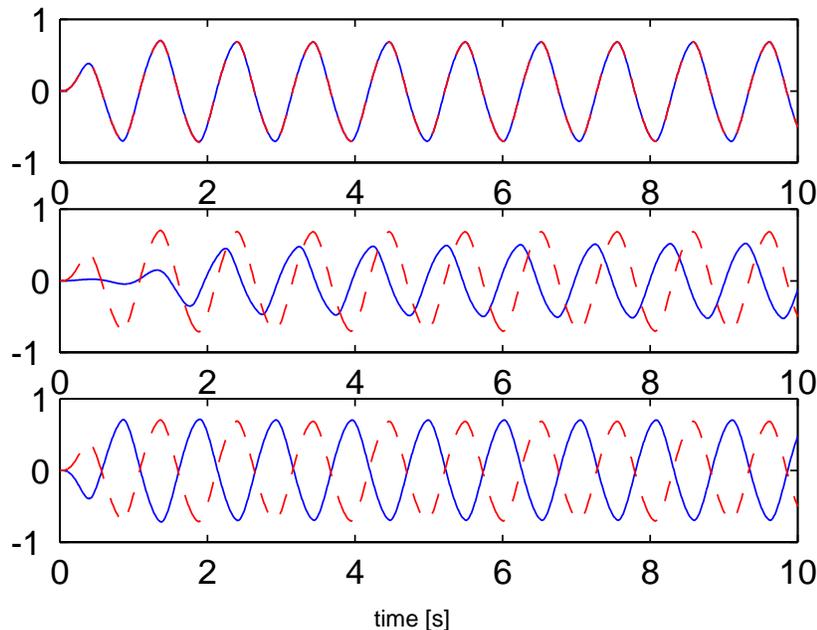


Figure 6: Realization of 0 (top),  $\pi/2$  (middle) and  $\pi$  (bottom) phase offsets in a 3 DOF rhythmic PPG. Parameters settings were the same as in Figure 4; the coupling parameters were  $w_c = 1$ , and all three oscillator amplitudes were chosen to be 1 rad.

## Robot Implementations

We implemented the discrete and rhythmic multi-joint PPGs on two robots, a 7 DOF Sarcos Dexterous Arm and a 30 DOF Sarcos Humanoid robot. Desired position, velocity, and acceleration information was derived from the states of the PPGs to realize a compute-torque controller. All necessary computations run in real-time at 480Hz on a multiple processor VME bus operated by VxWorks. We realized arbitrary rhythmic “3-D drawing” patterns, sequencing of point-to-point movements and rhythmic patterns like ball bouncing with a racket. Figure 7a shows our humanoid robot in a drumming task. The robot used both arms to generate a regular rhythm on a drum and a cymbal. The arms moved in 180-degree phase difference, primarily using the elbow and wrist joints, although even the entire body was driven with oscillators for reasons of natural appearance. The left arm hit the cymbal on beat 3, 5, and 7 based on an 8-beat pattern. The velocity zero crossings of the left drum stick at the moment of impact triggered the discrete movement to the cymbal. Figure 7b shows a trajectory piece of the left and the right elbow joint angles to illustrate the drumming pattern. Given the independence of a discrete and rhythmic movement primitives, it is very easy to create the demonstrated bimanual coordination without any problems to maintain a steady drumming rhythm.

Figure 7c illustrates how the robot drumming can also be synchronized with an external sound with zero phase offset. We used another drum connected to a microphone to manually create an external rhythmic signal that was added through the coupling constant  $K_i$  in Equation (10). In Figure 7c, the external sound undergoes a frequency shift, which is well tracked by the robot. This behavior is similar to the synchronization needed when playing in a music-band or orchestra.

## Conclusion

The present study describes research towards generating flexible movement primitives out of nonlinear dynamic attractor systems. We focused on motivating appropriate dynamic systems such that discrete and rhythmic movements could be generated with high-dimensional movement systems. We also described some implementations of our system of Programmable Pattern Generators on a complex anthropomorphic robot. Clearly, the presented work leaves open many questions that we raised at the beginning of this paper, as for instance learning with such dynamic movement primitives. However, we believe that our work provides a first

step towards pursuing new methods of perceptuomotor control that will finally result in successful autonomous and self-organizing machines and a better understanding of biology.

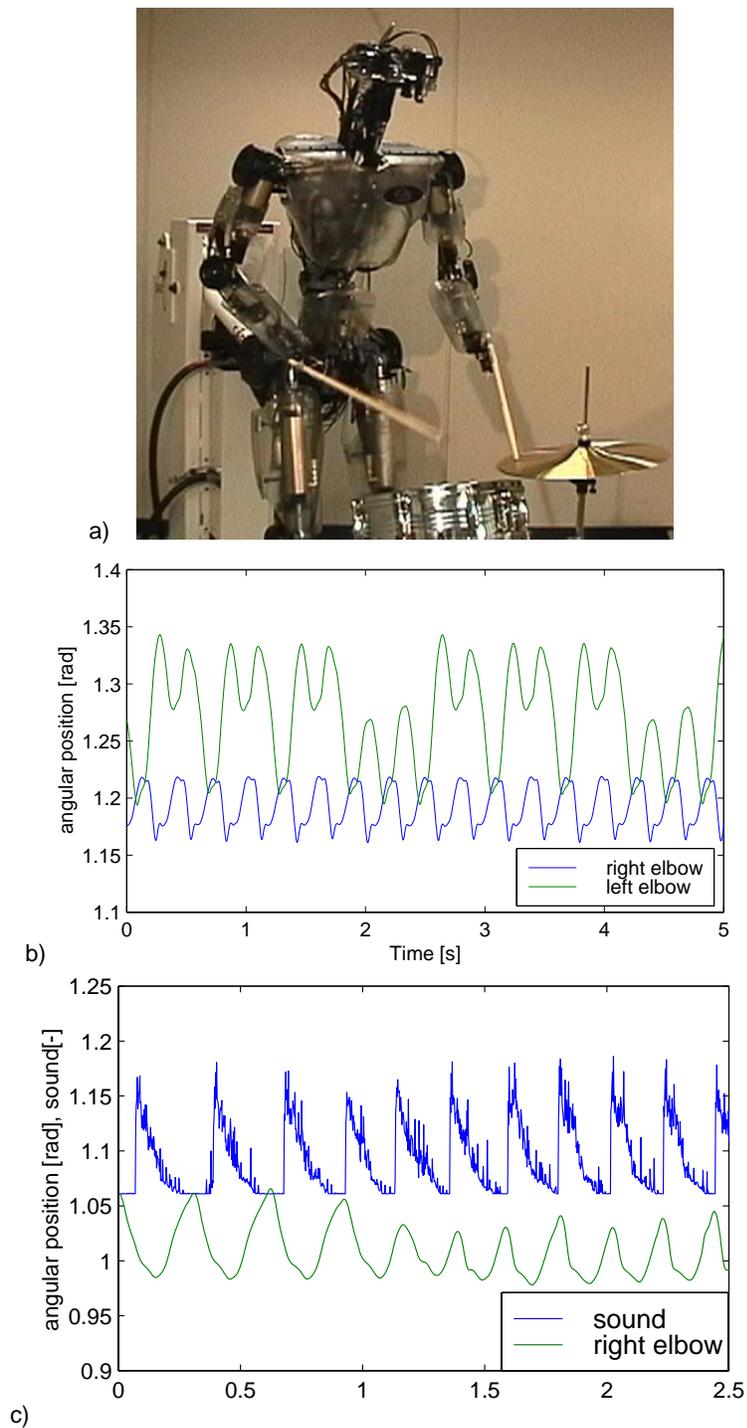


Figure 7: a) Humanoid robot in drumming task, b) coordination of left and right elbow, c) synchronization of right elbow with external sound source: the robot tracks the frequency shift of the sound

## Acknowledgments

This work was made possible by Award #9710312 of the National Science Foundation, the ERATO Kawato Dynamic Brain Project funded by the Japanese Science and Technology Cooperation, and the ATR Human Information Processing Research Laboratories.

## References

- [1] G. Tesauro, "Temporal difference learning of backgammon strategy," in *Proceedings of the Ninth International Workshop Machine*, D. Sleeman and P. Edwards, Eds. San Mateo, CA: Morgan Kaufmann, 1992, pp. 9-18.
- [2] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-dynamic Programming*. Belmont, MA: Athena Scientific, 1996.
- [3] R. S. Sutton and A. G. Barto, *Reinforcement learning : an introduction*. Cambridge: MIT Press, 1998.
- [4] J. M. Hollerbach, "Dynamic scaling of manipulator trajectories," *Transactions of the ASME*, vol. 106, pp. 139-156, 1984.
- [5] S. Kawamura and N. Fukao, "Interpolation for input torque patterns obtained through learning control," presented at Icarcv'94, 1994.
- [6] R. A. Schmidt, *Motor control and learning*. Champaign, Illinois: Human Kinetics, 1988.
- [7] M. A. Arbib, "Perceptual structures and distributed motor control," in *Handbook of Physiology, Section 2: The Nervous System Vol. II, Motor Control, Part 1*, V. B. Brooks, Ed.: American Physiological Society, 1981, pp. 1449-1480.
- [8] R. A. Brooks, "A robust layered control system for a mobile robot," *IEEE Journal of Robotics and Automation*, vol. 2, pp. 14-23, 1986.
- [9] A. I. Selverston, "Are central pattern generators understandable?," *The Behavioral and Brain Sciences*, vol. 3, pp. 555-571, 1980.
- [10] M. Raibert, *Legged robots that balance*. Cambridge, MA: MIT Press, 1986.
- [11] G. Taga, Y. Yamaguchi, and H. Shimizu, "Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment," *Biological Cybernetics*, vol. 65, pp. 147-159, 1991.
- [12] D. E. Koditschek, "Exact robot navigation by means of potential functions: Some topological considerations," presented at Proceedings of the IEEE International Conference on Robotics and Automation, Raleigh, North Carolina, 1987.
- [13] F. A. Mussa-Ivaldi and E. Bizzi, "Learning Newtonian mechanics," in *Self-organization, Computational Maps, and Motor Control*, P. Morasso and V. Sanguineti, Eds. Amsterdam: Elsevier, 1997, pp. 491-501.
- [14] D. Sternad, M. T. Turvey, and R. C. Schmidt, "Average phase difference theory and 1:1 phase entrainment in interlimb coordination," *Biological Cybernetics*, vol. no.67, pp. 223-231, 1992.
- [15] J. A. S. Kelso, *Dynamic patterns: The self-organization of brain and behavior*. Cambridge, MA: MIT Press, 1995.
- [16] M. Bühler, "Robotic tasks with intermittent dynamics," : Yale University New Haven, 1990.
- [17] A. A. Rizzi and D. E. Koditschek, "Further progress in robot juggling: Solvable mirror laws," presented at IEEE International Conference on Robotics and Automation, San Diego, CA, 1994.
- [18] J. F. Kalaska, "What parameters of reaching are encoded by discharges of cortical cells?," in *Motor Control: Concepts and Issues*, D. R. Humphrey and H. J. Freund, Eds.: John Wiley & sons, 1991, pp. 307-330.
- [19] N. Schweighofer, M. A. Arbib, and M. Kawato, "Role of the cerebellum in reaching movements in humans. I. Distributed inverse dynamics control," *Eur J Neurosci*, vol. 10, pp. 86-94, 1998.
- [20] N. Schweighofer, J. Spoolstra, M. A. Arbib, and M. Kawato, "Role of the cerebellum in reaching movements in humans. II. A neural model of the intermediate cerebellum," *Eur J Neurosci*, vol. 10, pp. 95-105, 1998.
- [21] N. A. Bernstein, *The control and regulation of movements*. London: Pergamon Press, 1967.
- [22] J. J. Craig, *Introduction to robotics*. Reading, MA: Addison-Wesley, 1986.
- [23] S. Schaal and D. Sternad, "Programmable pattern generators," presented at 3rd International Conference on Computational Intelligence in Neuroscience, Research Triangle Park, NC, 1998.
- [24] M. Williamson, "Neural control of rhythmic arm movements," *Neural Networks*, vol. 11, pp. 1379-1394, 1998.
- [25] G. Schöner, "A dynamic theory of coordination of discrete movement," *Biological Cybernetics*, vol. 63, pp. 257-270, 1990.
- [26] S. Schaal, D. Sternad, and C. G. Atkeson, "One-handed juggling: A dynamical approach to a rhythmic movement task," *Journal of Motor Behavior*, vol. 28, pp. 165-183, 1996.
- [27] D. Bullock and S. Grossberg, "Neural dynamics of planned arm movements: Emergent invariants and speed-accuracy properties during trajectory formation," *Psychological Review*, vol. 95, pp. 49-90, 1988.
- [28] N. Hogan, "An organizing principle for a class of voluntary movements," *Journal of Neuroscience*, vol. 4, pp. 2745-2754, 1984.
- [29] J. M. Hollerbach and C. G. Atkeson, "Inferring limb coordination strategies from trajectory kinematics," *Journal of Neuroscience Methods*, vol. 21, pp. 181-194, 1987.
- [30] D. Bullock, S. Grossberg, and F. H. Guenther, "A self-organizing neural model of motor equivalent reaching and tool use by a multijoint arm," *Journal of Cognitive Neuroscience*, vol. 5, pp. 408-435, 1993.
- [31] J. Baillieul and D. P. Martin, "Resolution of kinematic redundancy," in *Proceedings of Symposia in Applied Mathematics*, vol. 41: American Mathematical Society, 1990, pp. 49-89.
- [32] K. Matsuoka, "Mechanisms of frequency and pattern control in the neural rhythm generators," *Biological Cybernetics*, vol. 56, pp. 345-353, 1987.

- [33] K. Matsuoka, "Sustained oscillations generated by mutually inhibiting neurons with adaptation," *Biological Cybernetics*, vol. 52, pp. 73-83, 1985.
- [34] J. F. Soechting and C. A. Terzuolo, "An algorithm for the generation of curvilinear wrist motion in an arbitrary plane in three-dimensional space," *Neuroscience*, vol. 19, pp. 1393-1405, 1986.