Control of movement time and sequential action through attractor dynamics: A simulation study demonstrating object interception and coordination

Gregor Schöner* **and Cristina Santos****

C.N.R.S. – Centre de Recherche en Neurosciences Cognitives
31, ch. Joseph Aiguier, 13402 Marseille Cedex 20, France

Abstract. The timing of movements and of action sequences is difficult when on-line coupling to sensory information is a requirement. That requirement arises in most behavior-based robot architectures, in which relatively low-level and often noisy sensor input is used to initiate and steer action. We show how an attractor dynamics approach to the generation of behavior in such architectures can be extended to the timing of motor acts. We propose a two-layer architecture, in which a competitive "neural" dynamics controls the qualitative dynamics of a second, "timing" layer. At that second layer, periodic attractors generate timed movement. By activating such limit cycles over limited time intervals, discrete movements and movement sequences can be obtained. We demonstrate the approach by simulating two tasks that involve control of timing: the interception of moving objects by a simple two-degree-of-freedom robot arm and the temporal coordination of the end-effector motions of two six-degree-of-freedom robot arms.

1 Introduction

The classical organization of autonomous robots separates task planning, path planning, trajectory planning, and control (e.g., Latombe, 91). The time structure of movement is primarily concentrated at the trajectory planning level, at which time courses of relevant effector variables are specified. These time courses may be planned beforehand, based on prior knowledge about space and time constraints. In the potential field approach, some of this planning may take place on-line in reaction to changing sensory information as well (Khatib, 1986). Continuous on-line coupling to sensory information is, however, an important requirement for robots that interact with humans, that are mounted on mobile platforms, or that work in "natural" environments which are not highly controlled and may change over time. This aspect is particularly emphasized in behavior-based approaches to autonomous robotics (e.g., Arkin, 1998), in which linkage between action and perception is attempted at particularly low levels of sensory information.

Most current demonstrations of behavior-based robotics do not address timing: The time when a particular action is initiated and terminated is not a controlled variable, and not stabilized against perturbations. When a vehicle, for instance, takes longer to arrive at a goal because it needed to circumnavigate an obstacle, this change of timing is not compensated for by accelerating the vehicle along its path. Timed actions, by contrast, involve stable temporal relationships. Stable timing is important when particular events must be achieved in time-varying environments such as hitting or catching moving objects, avoiding moving obstacles, or coordinating multiple robots. Moreover, timing is critical in tasks involving sequentially structured actions, in which subsequent actions must be initiated only once previous actions have terminated or reached a particular phase.

While time schedules can be developped within classical approaches (e.g., through configuration-time space representations), timing is more difficult to control when it must be compatible with continuous on-line coupling to sensory information. One type of solution is to generate time structure at the level of control. Raibert (1986), for instance, generated rhythmic action by inserting into dynamic control models terms that stabilized oscillatory solutions. Similarly, Schaal and Atkeson (1993) generated rhythmic movements in a robot arm that supported juggling of a ball by inserting into the control system a model of the bouncing ball together with terms that stabilized stable limit cycles. A limitation of such approaches is that they

* Email: gregor@lnf.cnrs-mrs.fr
** Email: csantos@lnf.cnrs-mrs.fr

essentially generate a single motor act in rhythmic fashion. The flexible activation of different motor acts in response to user demands or sensed environmental conditions is more difficult to achieve from the control level.

These control level solutions were inspired, in part, by analogies with nervous systems, in particular, by the way the rhythmic movement patterns in legged locomotion are generated (e.g., Beer, Chiel, Sterling, 1990). The timing of rhythmic activities in nervous systems is typically based on the autonomous generation of rhythms in specialized neural networks ("central pattern generators"), which can be mathematically described as nonlinear dynamical systems with stable limit cycle (periodic) solutions. Coordination among limbs can be modelled through mutual coupling of such nonlinear oscillators (Schöner, Kelso, 1986). The on-line linkage to sensory information can be understood through the coupling of these oscillators to time-varying sensory information (Schöner, 1994). Limited attempts to extend these theoretical ideas to temporally discrete movements (e.g., reaching) have been made (Schöner, 1990).

The dynamic approach to autonomous robotics (Schöner, Dose, 1992; Schöner, Dose, Engels, 1995; Steinhage, Schöner, 1998; Large, Christensen, Bajcsy, 1999; Bicho, Mallet, Schöner, 2000) extends these ideas to the level of planning. Plans are generated from stable states of nonlinear dynamical systems, into which sensory information is fed. Intelligent choice of planning variables makes it possible to obtain complex trajectories and action sequences from stationary stable states, which shift and may even go through instabilities as sensory information changes. For the control of vehicle motion, for instance, a dynamical system of heading direction may generate paths that circumnavigate obstacles and find their way to a target, while at all times the planning variable "heading direction" sits in a fixed point attractor, which may shift as the vehicle moves and sensory information changes. The possibility of integrating multiple constraints and generating decisions through instabilities and multistability makes such systems much more flexible than nonlinear controllers.

The generation of trajectories with stable timing had not yet been attempted within this approach (but see Schaal, Kotosaka, Sternad, 2000, for a related attempt). In this paper we propose a dynamical systems architecture that generates timed trajectories of manipulators. The model consists of a timing layer, which can generate both stable oscillations and stationary states. A "neural" dynamics controls the switching between these two regimes. Incoupling of sensory information enables sensor driven initiation and termination of movement. Coupling among several such systems enables temporal coordination of multiple effectors.

2 The dynamical systems trajectory generator

The timing level consists of a dynamical system for a pair of timing variables, (x, y). Generating oscillatory solutions requires at least two dynamical degrees of freedom. Thus, although only the variable, x, will be used to control motion of a relevant robotic task variable, a second auxiliary variable, y, is needed to represent oscillatory states. The time courses of these two variables are generated from a dynamical system

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = -5\ |u_{\text{init}}| \begin{pmatrix} x - x_{\text{init}} \\ y \end{pmatrix} + |u_{\text{hopf}}\ |\mathbf{f}_{\text{hopf}}\ - 5\ |u_{\text{final}}| \begin{pmatrix} x - x_{\text{final}} \\ y \end{pmatrix} + \text{gwn} \tag{1}$$

that can operate in three dynamic regimes controlled by the three "neurons" u_i ($i = \text{init}, \text{hopf}, \text{final}$). The "init" and "final" contributions generate stable stationary solutions at $x = -1$ for "init" and $+1$ for "final" with $y = 0$ for both. These states are characterized by a time scale of $\tau = 1/5 = 0.2$. The "hopf" term

$$\mathbf{f}_{\text{hopf}} = \begin{pmatrix} 2.5 & -\omega \\ \omega & 2.5 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} - 2.5\ (x^2 + y^2) \begin{pmatrix} x \\ y \end{pmatrix} \tag{2}$$

is the normal form of the Hopf bifurcation (Perko, 1991), that is, the simplest polynomial equation containing a bifurcation to a limit cycle. We use it because it can be completely analytically solved, providing complete control over its stable states. The "hopf" term in isolation ($u_{\text{init}} = u_{\text{final}} = 0$; $|u_{\text{hopf}}| = 1$) provides a stable periodic solution (limit cycle attractor)

$$x(t) = \sin(\omega t) \tag{3}$$

with cycle time $T = 2\pi/\omega$ and amplitude 1. Relaxation to that stable solution occurs at a time scale of $1/(2. * 2.5) = 0.2$ time units.

Gaussian white noise gwn is added to the timing dynamics to guarantees escape from unstable states.

The "neuronal" dynamics of u_i ($i = \text{init}, \text{final}, \text{hopf}$) switches the timing dynamics from the fixed point regimes into the oscillatory regime and back. Thus, a single discrete movement act is generated by starting out with neuron $|u_{\text{init}}| = 1$ activated, the other neurons deactivated ($|u_{\text{hopf}}| = |u_{\text{final}}| = 0$), so that the system is in a postural state. The oscillatory solution is then stabilized ($|u_{\text{init}}| = 0;\ |u_{\text{hopf}}| = 1$). This oscillatory solution is deactivated again when the effector reaches its target state, after approximately a half-cycle of the oscillation, turning on the final postural state instead ($|u_{\text{hopf}}| = 0;\ |u_{\text{final}}| = 1$). These various switches are generated from the following competitive dynamics:

$$\alpha\, \dot{u}_{\text{init}} = \mu_{\text{init}}\ u_{\text{init}} - |\mu_{\text{init}}|\ u_{\text{init}}^3 - 2.1\ (u_{\text{final}}^2 + u_{\text{hopf}}^2)\ u_{\text{init}} + \text{gwn} \tag{4}$$

$$\alpha\, \dot{u}_{\text{hopf}} = \mu_{\text{hopf}}\ u_{\text{hopf}} - |\mu_{\text{hopf}}|\ u_{\text{hopf}}^3 - 2.1\ (u_{\text{init}}^2 + u_{\text{final}}^2)\ u_{\text{hopf}} + \text{gwn} \tag{5}$$

$$\alpha\, \dot{u}_{\text{final}} = \mu_{\text{final}}\ u_{\text{final}} - |\mu_{\text{final}}|\ u_{\text{final}}^3 - 2.1\ (u_{\text{init}}^2 + u_{\text{hopf}}^2)\ u_{\text{final}} + \text{gwn} \tag{6}$$

The first two terms of each equation represent the normal form of a degenerate pitchfork bifurcation: A single attractor at $u = 0$ for negative μ_i becomes unstable for positive μ_i, and two new attractors at $u_i = 1$ and $u_i = -1$ form. We use the absolute value of u_i as a weight factor in the timing dynamics, so that $+1$ and -1 are equivalent "on" states of a neuron, while $u = 0$ is the "off" state.

The last term in each equation is a competitive term, which destabilizes any attractors in which more than one neuron is "on". For positive μ_i, all attractors of this competitive dynamics have one neuron in an "on" state, and the other two neurons in the "off" state (Schöner, Dose, 1992; Large, Christensen, Bajczy, 1999).

The neuron, u_i, with the largest competitive advantage, $\mu_i > 0$, is likely to win the competition, although for sufficiently small differences between the different μ_i values multiple outcomes are possible (the system is multistable). To control switching, the parameters, μ_i (competitive advantages) are therefore defined as functions of user commands, sensory events, or internal states (Steinhage, Schöner, 1998). Here, we assure that one neuron is always "on" by varying the μ-parameters between the values 1.5 and 3.5: $\mu_i = 1.5 + 2b_i$, where b_i are "quasi-boolean" factors taking on values between 0 and 1 (with a tendency to have values either close to 0 or close to 1). These quasi-booleans express logical or sensory conditions controlling the sequential activation of the different neurons (see Steinhage, Schöner, 1998, for a general framework for sequence generation based on these ideas):

1. b_{init} may be controlled by user input: the command "move" sets b_{init} from the default value 1 to 0 to destabilize the initial posture. b_{init} may also be controlled by sensory input, such that, for instance, b_{init} changes from 1 to 0 when a particular sensory event is detected. Below we demonstrate how the time-to-contact of an approaching object computed from sensory information can be used to initiate movement in this manner.
2. b_{hopf} is set from 0 to 1 under the same conditions. This term is multiplied, however, with a second factor $b_{\text{has not reached target}}(x) = \sigma(x_{\text{crit}} - x)$ that resets b_{hopf} to zero when the effector has reached its final state. Herein, $\sigma(x)$ is a sigmoid function that ranges from 0 for negative argument to 1 for positive argument, chosen here as
$$\sigma(x) = [\tanh(10x) + 1]/2 \tag{7}$$
although any other functional form will work as well. The factor, $b_{\text{has not reached target}}(x)$ has values close to one while the timing variable x is below $x_{\text{crit}} = 0.7$ and switches to values close to zero when x comes within 0.3 of the target state $x = 1$. Multiplying two quasi-booleans means connecting the corresponding logical conditions with an "and" operation. Thus, as soon as the timing variable has come within the vicinity of the final state, it autonomously turns the oscillatory state off. In actual implementation, this switch can be driven from the sensed actual position of an effector rather than from the timing dynamics.
3. b_{final} is, conversely, set from 0 to 1 when the timing variable comes into the vicinity of the target: $b_{\text{final}} = 1 - b_{\text{has not reached target}}$.

The time scale of the neuronal dynamics is controlled by $\alpha = 45.45$, which leads to a typical relaxation time of $\tau_u = 0.02$, ten times faster than the relaxation time of the timing variables. This difference in time scale guarantee that the analysis of the attractor structure of the neural dynamics is unaffected by the dependence of its parameters, μ_i on the timing variable, x, which is a dynamical variable as well. (Strictly speaking, the neural and timing dynamics are thus mutually coupled. The difference in time scale makes it possible to treat x as a parameter in the neural dynamics. Conversely, the neural weights can be assumed to have relaxed to their corresponding fixed points when analyzing the timing dynamics.)

Solutions. Periodic movement can be trivially generated from the timing and neural dynamics by selecting u_{hopf} "on" through the corresponding quasi-booleans. A timed, but temporally discrete movement act, is autonomously generated by these two coupled levels of nonlinear dynamics through a sequence of neural switches, such that an oscillatory state exists during an appropriate time interval of about a half-cycle. This is illustrated in Figure 1. The timing variable, x, which is used to generate effector movement, is initially in a postural state at -1, the corresponding neuron u_{init} being "on". When the user initiates movement, the quasi-booleans, b_{init} and b_{hopf} exchange values, which leads, after a short delay, to the activation of the "hopf" neuron. This switch initiates movement, with x evolving along a harmonic trajectory, until it approaches the final state at $+1$. At that point, the quasi-boolean b_{final} goes to one, while b_{hopf} changes to zero. The neurons switch accordingly, activating the final postural state, so that x relaxes to its terminal level $x = 1$. The movement time is approximately a half cycle time, here $MT = 2$.

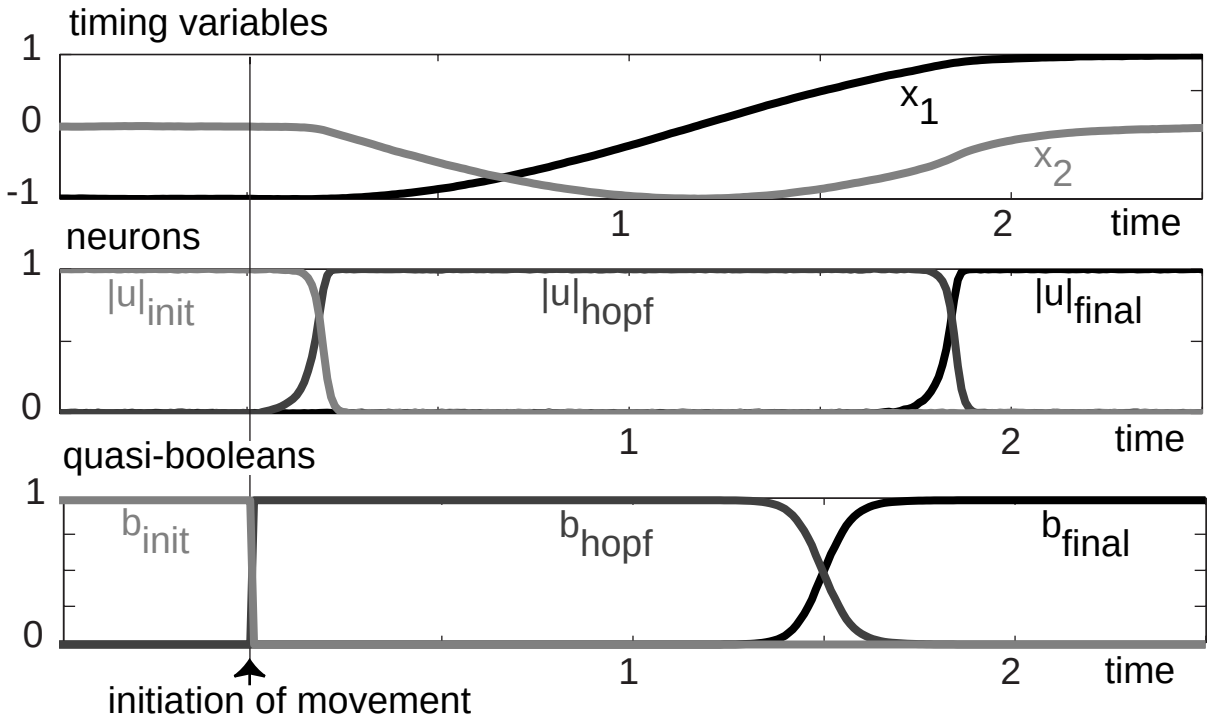

Fig. 1. Simulation of a user initiated temporally discrete movement represented by the timing variable, x, which is plotted together with the auxiliary variable, y, in the top panel. The time courses of the three neural activation variables, u_{init}, u_{hopf}, and u_{final}, which control the timing dynamics, are shown in the middle panel. The quasi-boolean parameters, b_{init}, b_{hopf}, and b_{final}, plotted on bottom, determine the competitive advantage of each neuron.

3 Simulation of a two degree-of-freedom arm intercepting a ball

As a first (toy) example of how the dynamical systems approach to timing can be put to use to solve robotic problems, consider a two degree-of-freedom robot arm moving in a plane (Fig. 2). The task is to generate a timed movement from an initial posture to intercept an approaching ball. Movement with a fixed movement time (reflecting manipulator constraints) must be initiated in time to reach the ball before it arrives within the plane in which the arm moves. Factors such as reachability and approach path of the ball are continuously monitored, leading to a return to the resting posting when interception becomes impossible (e.g., because the ball hits outside the workspace of the arm, the ball is no longer visible, or ball contact is no longer expected within a criterion time-to-contact). After the ball was intercepted, the arm moves back to its resting position, ready to initiate a new movement whenever appropriate sensory information arives.

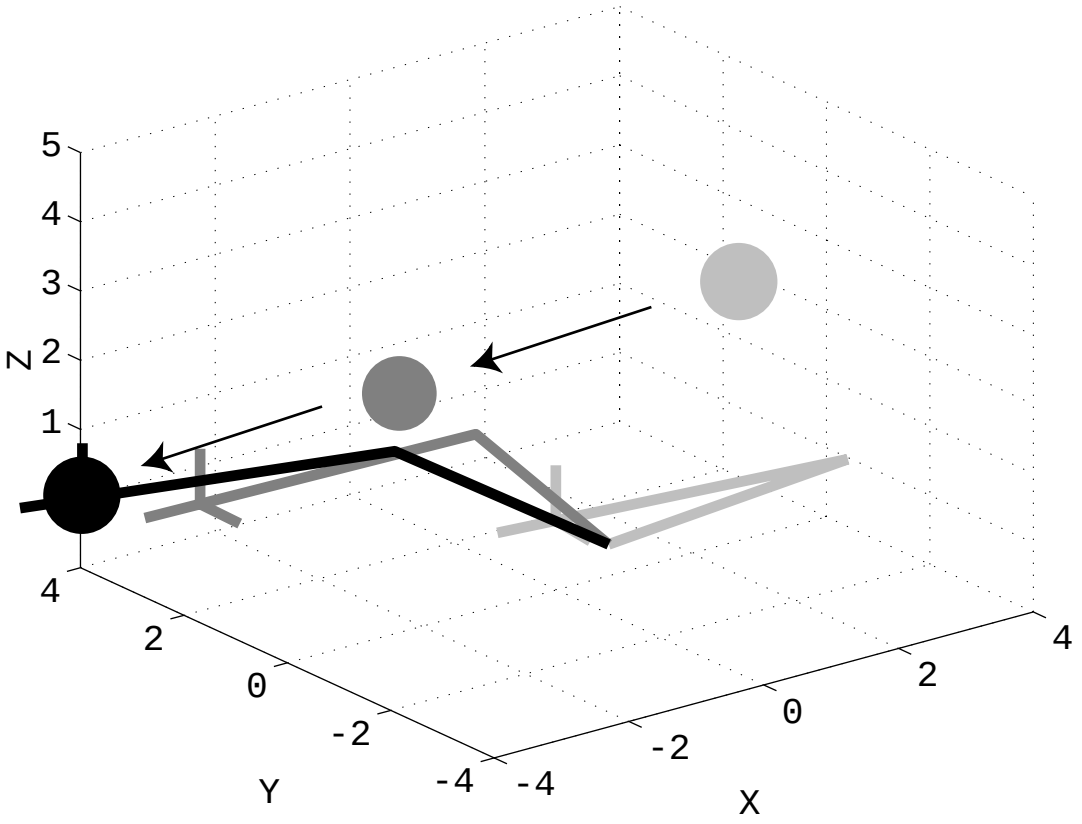

Fig. 2. A two degree-of-freedom arm intercepts an approaching ball. Corresponding ball and arm positions are illustrated by using the same grey-scale. The first position (light grey) is close to the critical time-to-contact, where arm motion starts. The last position (dark grey) is close to actual contact. The black arrows indicate the ball's movement.

In this simulation we have extracted from a simulated ball trajectory two measures: the time-to-contact, τ_{t2c}, based on constant approach velocity and the point of collision in the plane in which the robot arm moves. The time-to-contact can be extracted from segmented visual information without having estimated the full cartesisan trajectory of the ball (Lee, 1976). The point of impact can be computed along similar lines if the ball size is assumed known and can be measured in the image.

These two measures fully control the neural dynamics through the quasi-boolean parameters. A sequence of neural switches is generated by translating sensory conditions and logical constraints into values for these parameters (Steinhage, Schöner, 1998). For instance, the parameter, b_{init}, controlling the competitive advantage of the initial postural state must be "on" ($= 1$) when the the timing variable x is close to the initial state -1, **and** either of the following is true: (1) ball not approaching or not visible ($\tau_{\mathrm{t2c}} \leq 0$); (2) ball contact not yet within a criterion time-to-contact ($\tau_{\mathrm{t2c}} > \tau_{\mathrm{crit}}$); (3) ball is approaching within criterion time-to-contact but is not reachable ($0 < \tau_{\mathrm{t2c}} < \tau_{\mathrm{crit}}$; $b_{\mathrm{reachable}} = 0$). These logical conditions can be expressed through this mathematical function:

$$b_{\mathrm{init}} = \sigma(-x_{\mathrm{crit}} - x)\,[\sigma(\tau_{\mathrm{t2c}} - \tau_{\mathrm{crit}}) + \sigma(\tau_{\mathrm{t2c}})\;\sigma(\tau_{\mathrm{crit}} - \tau_{\mathrm{t2c}})\;\sigma(1 - b_{\mathrm{reachable}})\;\; + \sigma(-\tau_{\mathrm{t2c}})] \tag{8}$$

where $\sigma(\cdot)$ is the threshold-function used earlier (Eq. 7). Multiplication of sigmoids expresses logical "and" operations. The "or" is realized by summing terms which are never simultaneously different from zero. In other cases, the "or" is expressed with the help of the "not" (subtracting from 1) and the "and". This is used in the following expressions for b_{hopf} and b_{final} which can be derived from a similar analysis:

$$\begin{aligned} b_{\mathrm{hopf}} \;&=\; 1 - (1 - [\sigma(x_{\mathrm{crit}} - x)\;\sigma(\tau_{\mathrm{t2c}})\;\sigma(\tau_{\mathrm{crit}} - \tau_{\mathrm{t2c}})\;\sigma(b_{\mathrm{reachable}})]) \\ &\quad \cdot (1 - [\sigma(x + x_{\mathrm{crit}})\{1 - (1 - \sigma(1 - b_{\mathrm{reachable}}))(1 - \sigma(-\tau_{\mathrm{t2c}}))(1 - \sigma(\tau_{\mathrm{t2c}} - \tau_{\mathrm{crit}}))(1 - \sigma(x_{\mathrm{crit}} - x))\}]) \end{aligned} \tag{9}$$

$$b_{\mathrm{final}} \;=\; \sigma(x - x_{\mathrm{crit}})\;\sigma(\tau_{\mathrm{t2c}})\;\;\sigma(\tau_{\mathrm{crit}} - \tau_{\mathrm{t2c}})\;\sigma(b_{\mathrm{reachable}}) \tag{10}$$

Finally, the three relevant coordinate frames must be linked: (a) The timing variable frame describes the end-effector position along a straight path from the initial position (reference posture) to the target position (computed coordinates of point of interceptance). (b) The task relevant frame is cartesian and describes the end-effector position of the arm and the ball position. (c) The arm kinematics is described by two joint angles. Frame (a) and (b) are linked through straightforward formulae, which depend on the predicted point of interceptance. Frame (b) and (c) are linked through the kinematic model of the robot arm and its inverse. During movement execution, the timing variables are continuously transformed into reference frame (b), from which joint angles are computed through the inverse kinematic transformation.

Figure 2 shows how this two degree-of-freedom arm intercepts an approaching ball. The detailed time courses of the relevant variables and parameters are shown in Figure 3. As the ball approaches, the current time-to-contact becomes smaller than a critical value (here 3), at which time the quasi-boolean for motion, b_{hopf} becomes one, triggering activation of the corresponding neuron, u_{hopf}, and movement initiation. Movement is completed (x reaches the final state of $+1$) well before actual ball contact is made. The arm waits in the target posture. In this simulation the ball is reflected upon contact. The negative time-to-contact observed then leads to autonomous initiation of the backward movement to the arm resting position.

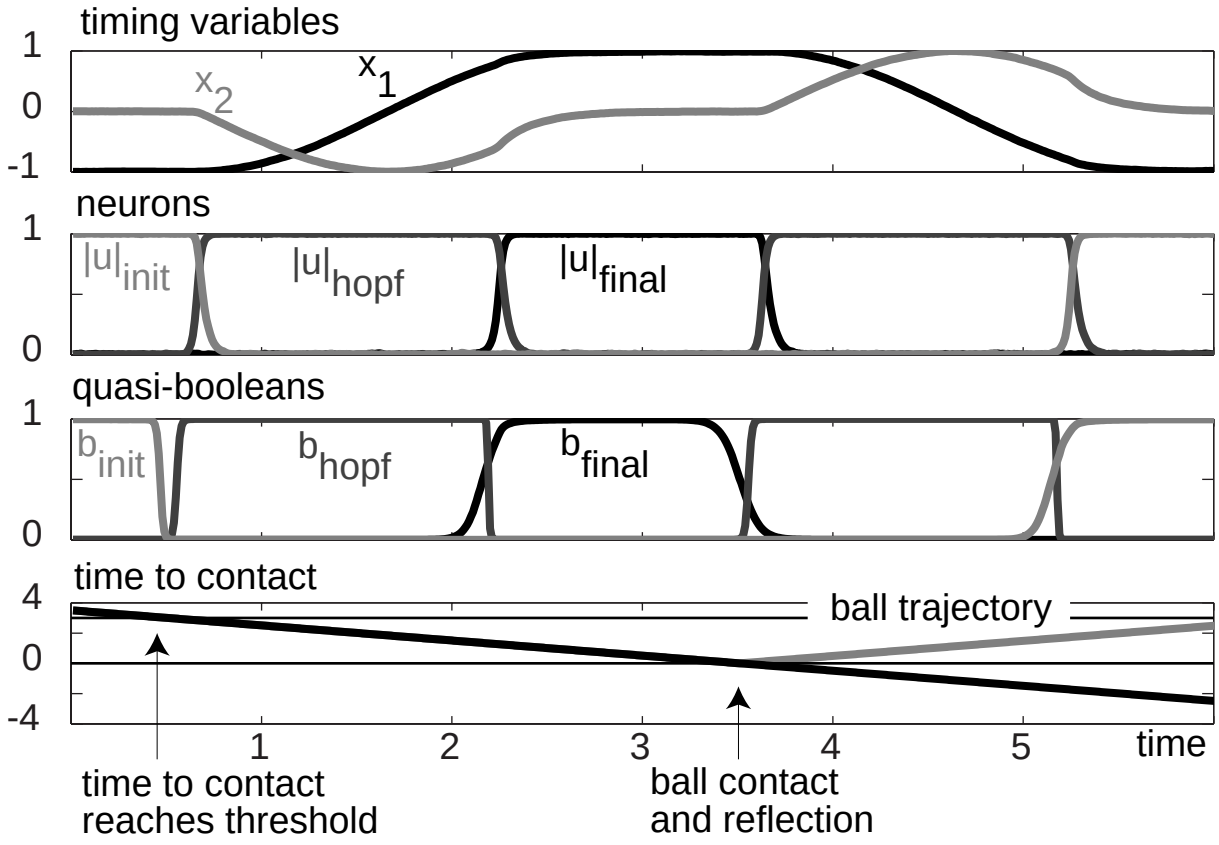

Fig. 3. Trajectories of variables and parameters in autonomous ball interception and return to resting position. The top three panels represent timing variables, neural variables, and quasi-booleans as in Fig 1. The bottom panel shows the time-to-contact, which crosses a threshold at about 0.5 time units. When contact is made, the ball is assumed to be reflected, leading to negative time-to-contact.

The fact that timed movement is generated from attractor solutions of a nonlinear dynamical system leads to a number of properties of this system, that are potentially useful to the real-world implementation of this form of autonomy. For instance, the autonomous sensor-driven initiation of movement is stabilized by the hysteresis properties of the competitive neural dynamics (see Schöner, Dose, 1992, for analysis). Moreover, when sensory conditions change, an appropriate new sequence of events emerges. When one of the sensory conditions for ball interception is invalided (e.g., ball becomes invisible, unreachable, or no longer

approaches with appropriate time-to-contact), then one of the following happens depending on the point within the sequence of events at which the change occurs: (1) If the change occurs during the initial postural stage, the system stays in that postural state. (2) If the change occurs during the movement, then the system continues on its trajectory, now going around a full cycle to return to the reference posture. (3) When the change occurs during posture in the target position, a discrete movement is initiated that takes the arm back to its resting position. As an illustration, Figure 4 shows how the arm continues on its movement cycle when during the motion phase a sudden change (ball trajectory is changed leading to much larger time-to-contact) invalidates the sensory conditions for ball contact.

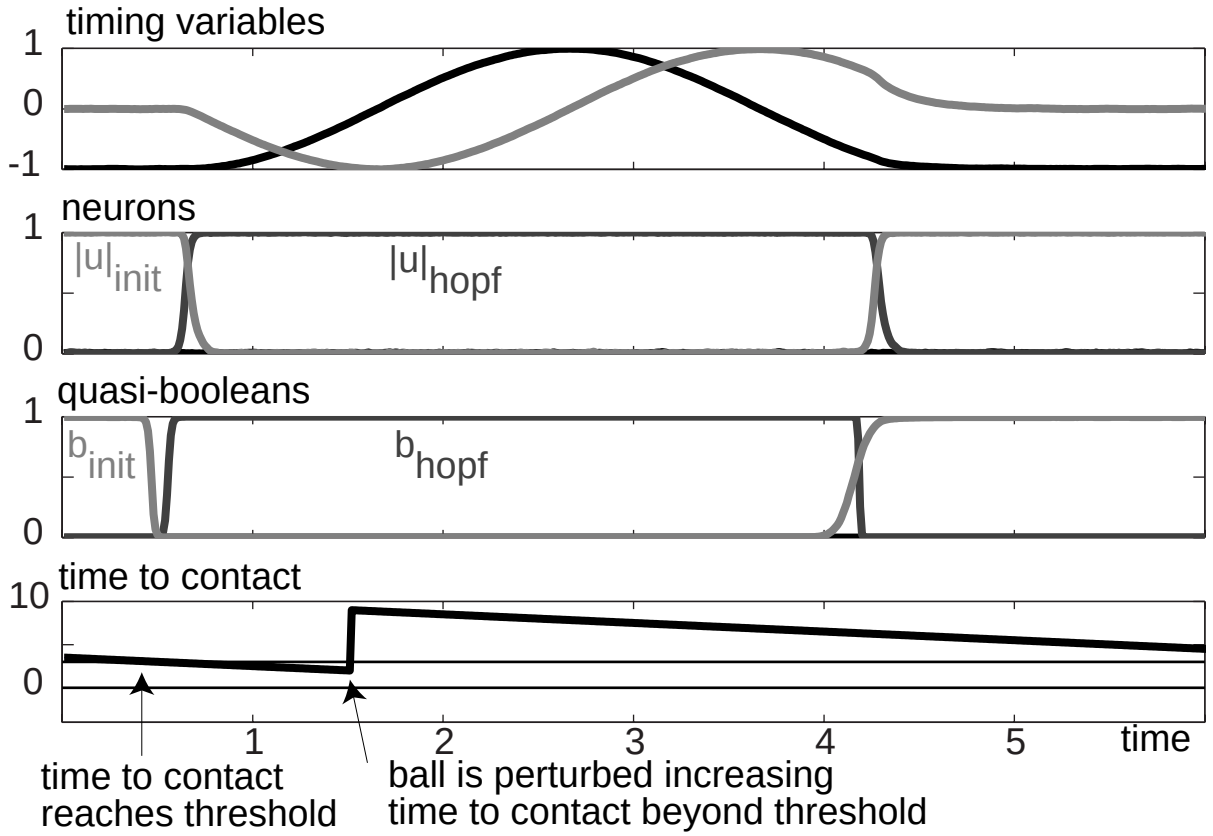

Fig. 4. Similar to Fig. 3, but now the ball is suddenly shifted away from arm at about 1.5 time units, leading to much increased time-to-contact, well beyond threshold for movement initiation. The arm is in the motion stage at this point and hence continues its movement a full cycle, until captured by the initial postural state when the arm is back to the reference position. This behavior emerges from the sensory conditions controlling the neuronal dynamics.

4 Simulation of two six degree-of-freedom arms: temporal coordination

A second example illustrating uses of the dynamical systems approach to timing is the temporal coordination of two robot arms. In the simulations, we used two PUMA arms, whose inverse kinematics were based on the exact solution (Fu, Gonzalez, Lee, 1987; section. 2.3.2.). Each arm was driven by a complete system of timing and neural dynamics. The two timing dynamics were coupled in a way that generates phase-locking in the periodic regime. This was achieved by modifying the "Hopf" contribution to the timing dynamics as follows:

$$\begin{pmatrix} \dot{x}_1 \\ \dot{y}_1 \end{pmatrix} = \ldots |u_{\mathrm{hopf},1}| \left[\mathbf{f}_{\mathrm{hopf},1}(x_1, y_1) + |u_{\mathrm{hopf},2}| c \begin{pmatrix} x_2 - x_1 \\ y_2 - y_1 \end{pmatrix} \right] \ldots \tag{11}$$

$$\begin{pmatrix} \dot{x}_2 \\ \dot{y}_2 \end{pmatrix} = \ldots |u_{\text{hopf},2}| \left[\mathbf{f}_{\text{hopf},2}(x_2, y_2) + |u_{\text{hopf},1}| c \begin{pmatrix} x_1 - x_2 \\ y_1 - y_2 \end{pmatrix} \right] \ldots \tag{12}$$

where the index $i = 1, 2$ refers to arm 1 and arm 2. The coupling term is multiplied with the neuronal activation of the other system's hopf state so that coupling is effective only when both components are in the movement state.

In discrete motor acts, a coupling of this form tends to synchronize movement in the two components. Thus, even if the movement onsets are not perfectly synchronized, this coupling coordinates the two components so that movements terminate approximately simultaneously. This is illustrated in the top panel of Figure 5. Moreover, coupling two timing dynamics removes the need to compute exactly identical movement times for two component movements that must be temporally coordinated. Even if there is a discrepancy in the movement time programmed by the parameter, ω, of the timing dynamics, coupling generates identical effective movement times. This discrete analogue of frequency locking is illustrated in the bottom panel of Figure 5.

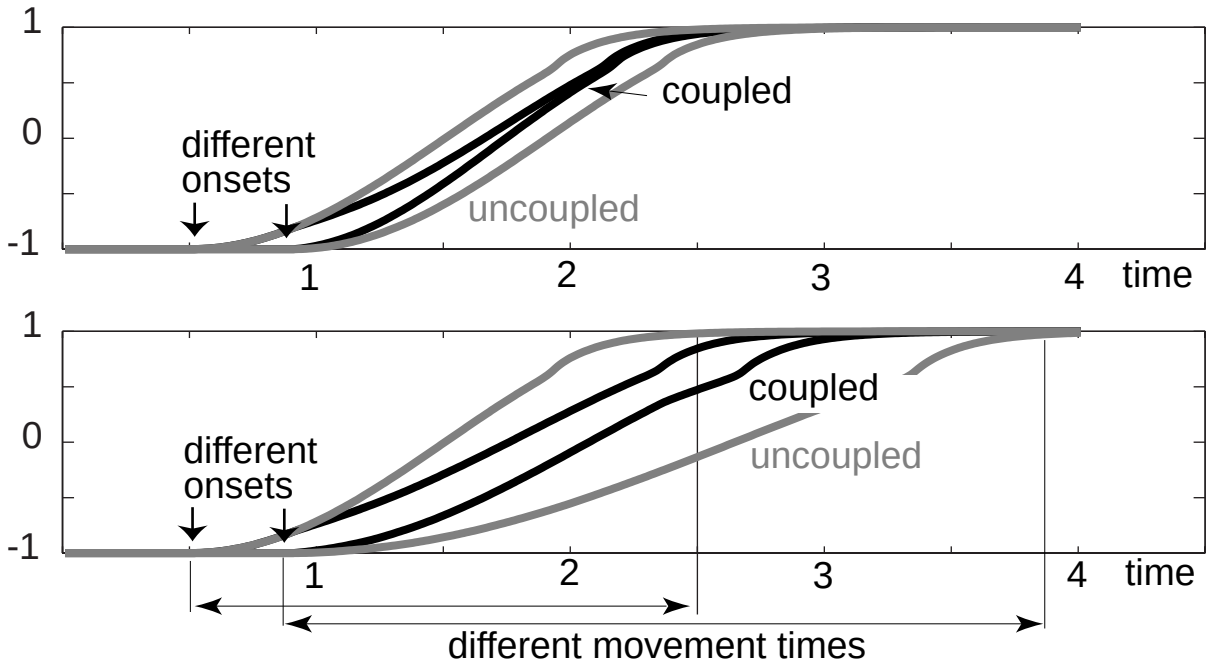

Fig. 5. Coordination between two timing dynamics through coupling leads to synchronization. The relevant x component of the timing dynamics is shown in each panel for both systems. In the top panel, movement initiation is slightly asynchronous. Without coupling (grey lines) this asynchrony would persist throughout the movement. With coupling (black lines), the initial asynchrony is diminished so that the two components terminate movement almost simultaneously. In the bottom panel, movement initiation is asynchronous as well, but in addition movement times differ (2 vs. 3). While in this case coupling does not achieve perfect synchrony, the movement time of the two coupled componets is equal.

5 Conclusion/Outlook

We have shown how timed movements and sequences of movements can be autonomously generated from an attractor based two-layer dynamics. The timing level has either stable fixed points or stable limit cycles. It is switched between these regimes by the neural dynamics, which is build entirely around fixed points, at which only one neuron is active. Parameters of the neural dynamics ("competitive advantage") express sensory and logical conditions for the activation of any particular neuron and the corresponding movement state.

We have illustrated how this attractor based dynamics may generate rhythmic and temporally discrete movements, movement sequences, and temporally coordinated movements of multiple effectors. We simulated two situations: In one, a simple robot arm intercepts a moving object and returns to a reference position thereafter. This sequence is stably adapted to changing online sensory information. Moreover, movement initiation is entirely sensor-driven in this example. A second simulation demonstrated how coupling amoung multiple timing systems helps synchronize systems and reduces the computational requirements for determining identical movement parameters across such components. This coordination through coupling approach resembles the generation of coordinated patterns of activation in locomotory behavior of nervous systems.

One advantage of our formulation is the ease with which the system is integrated into larger architectures for behavioral organization (Steinhage, Schöner, 1998) that do not necessarily explicitly represent timing requirements. At a technical level, our system is analytically treatable to a large extent, which facilitates the specification of parameters such as movement time, movement extent, maximal velocity, etc. Moreover, even relatively complex logical and sensory conditions can be expressed though the quasi-boolean parameters of the neuronal dynamics. This enables the generation of sequences. Schaal, Kotosaka, and Sternad (2000) present a related project, in which a neuronally inspired dynamical system is used to generate both rhythmic and temporally discrete movement. The analytical solvability of our dynamics and the generalization to sequence generation are two distinguishing features of our approach.

An implementation of this approach in hardware will provide a more rigorous test of its robustness. This will probe, in particular, how the inherent stability properties of neural and timing dynamics play out when the sensory information that serves to initiate movement is noisy and unreliable. We are currently planning an implementation of this nature making use of an experimental robot vehicle available in our lab (Bicho, Mallet, Schöner, 2000). Another direction in which these ideas could be developed lies in the domain of computer based animation, in which autonomy can help to reduce the amount of programmer effort to generate scenes and can provide user interactivity (Latombe, 1999).

References

1. R.C. Arkin. *Behavior-Based Robotics*. MIT Press, Cambridge, 1998.
2. R D Beer, H J Chiel, and L S Sterling. A biological perspective on autonomous agent design. *Robotics and Autonomous Systems*, 6:169–189, 1990.
3. E Bicho, P Mallet, and G Schöner. Target representation on an autonomous vehicle with low-level sensors. *The International Journal of Robotics Research*, 19, pages-424-447, 2000.
4. K S Fu, R C Gonzalez, and C S G Lee. *Robotics: Control, Sensing, Vision, and Intelligence*. McGraw-Hill, New York, 1987.
5. O Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *International Journal Robotics Research*, 5:90–98, 1986.
6. E W Large, H I Christensen, and R Bajcsy. Scaling the dynamic approach to path planning and control: Competition amoung behavioral constraints. *International Journal of Robotics Research*, 18(1):37–58, 1999.
7. J. C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.
8. Jean-Claude Latombe. Motion planning: A journey of robots, molecules, digital actors, and other artifacts. *The International Journal of Robotics Research*, 18(11):1119–1128, November 1999.
9. D N Lee. A theory of visual control of breaking based on information about time-to-collision. *Perception*, 5:437–459, 1976.
10. L Perko. *Differential Equations and Dynamical Systems*. Springer Verlag, Berlin, 1991.
11. M H Raibert. *Legged robots that balance*. MIT Press, Cambridge, Massachusetts, 1986.
12. S Schaal and C G Atkeson. Open loop stable control strategies for robot juggling. In *IEEE International Conference on Robotics and Automation*, volume 3, pages 913–918, 1993.
13. S Schaal, S Kotosaka, and D Sternad. Nonlinear dynamical systems as movement primitives. In *IEEE International Conference on Humanoid Robotics*, Cambridge, MA, 2000.
14. G Schöner. A dynamic theory of coordination of discrete movement. *Biological Cybernetics*, 63:257–270, 1990.
15. G Schöner. Dynamic theory of action-perception patterns: The time-before-contact paradigm. *Human Movement Science*, 3:415–439, 1994.
16. G Schöner and M Dose. A dynamical systems approach to task-level system integration used to plan and control autonomous vehicle motion. *Robotics and Autonomous Systems*, 10:253–267, 1992.
17. G Schöner, M Dose, and C Engels. Dynamics of behavior: Theory and applications for autonomous robot architectures. *Robotics and Autonomous Systems*, 16:213–245, 1995.
18. G Schöner and J A S Kelso. Dynamic pattern generation in behavioral and neural systems. *Science*, 239:1513–1520, 1988.

19. A Steinhage and G Schöner. Dynamical systems for the behavioral organization of autonomous robot navigation. In McKee G T Schenker P S, editor, *Sensor Fusion and Decentralized Control in Robotic Systems: Proceedings of SPIE*, volume 3523, pages 169–180. SPIE-publishing, 1998.

This article was processed using the TEX macro package with SIRS2001 style