Autonomous Timed Movement based on Attractor Dynamics in a Ball Hitting Task

Farid Oubbati and Gregor Schöner
Institut für Neuroinformatik, Ruhr–Universität Bochum, Bochum, Germany

Keywords: Attractor Dynamics, Timed Movement, Behavioral Organization.

Abstract: Timed robotic actions so that they are initiated or terminated just in time can be crucial in many tasks and scenarios in which the robot has to coordinate with other robotic agents or to interact with external entities such as moving objects. The analogy with human movement coordination has motivated an approach in which timed movements are generated from stable periodic solutions of dynamical systems, which are turned on and off in time to initiate and terminate a timed motor act. Here we extend this approach to generate sequences of timed motor actions required to intercept and hit a rolling ball on an inclined plane. The proposed system combines attractor dynamics for the robot's end-effector heading direction, fixed point attractor for end-effector postural states, limit cycle attractor dynamics for the end-effector speed and competitive neural dynamics to organize the different behaviors and movement phases. The ball interception point and time to contact are predicted based on a Kalman estimate of the ball's kinematics. The work is implemented on a redundant manipulator CoRA platform and the ball motion is monitored by the manipulator's vision system.

1 INTRODUCTION

Timed movement is defined in terms of the stability of its temporal structure, in such a way that an inherent acceleration or deceleration actions are performed so as to restore as much as possible the overall movement time (Schoener, 2002). Such behavior can be observed in most of the biological movements. For instance, in human reaching tasks, the hand opening is adapted to the reaching movement, i.e accelerated or decelerated accordingly (Jeannerod, 1984). This principle of coordination is crucial in many action-perception human tasks such as juggling, catching, hitting (Warren, 2006). In rhythmic movement, timing can be achieved by inserting couplings into a dynamical system that stabilize the phase relationships among periodic attractors, exemplified by the concept of central pattern generators and their coordination (Ijspeert, 2008). To control the timing of individual discrete motor acts, similar ideas can be used (Schoener, 1990). The problem is more complicated, as the stable periodic solutions must be initiated and terminated autonomously, requiring a form of behavioral organization (Steinhage and Schoener, 1998).

In this work, we propose an architecture that combines the timing of discrete motor acts directed toward moving objects and the behavioral organization of this timed movements in a ball hitting task scenario. The different task movements are planned by a non linear dynamical system in which fixed points attractors models postural states, and limit cycles attractors are used for movement states. The behavioral organization of these different states is achieved through a set of switching variables controlled by coupled nonlinear dynamics. The autonomous initiation and termination of the different task movements through the switching dynamics is itself controlled by some perceptual variables or parameters received from a tracking and prediction mechanism that involves a kalman filter for ball position and speed estimations and a process that delivers an estimate of the ball point of impact (p_{2c}) and time to contact (τ_{2c}). According to these perceptual parameters, the system initiate an interception and then a hitting movement 'just in time' returning after that to a base line ready to initiate the next hitting task. The proposed architecture is able to handle different situations like aborting the hitting sequence if this may lead the manipulator out of its reachable work space or reseting the hole system if the maximum speed of the manipulator is reached. This behavioral flexibility emerges from the continuous-time dynamics of the variables.

Similar efforts to exploit attractor dynamics to generate timed movements have been made in the

Oubbati F. and Schöner G..
Autonomous Timed Movement based on Attractor Dynamics in a Ball Hitting Task.
DOI: 10.5220/0004324003040311
In *Proceedings of the 5th International Conference on Agents and Artificial Intelligence* (ICAART-2013), pages 304-311
ISBN: 978-989-8565-38-9

realm of the rhythmic movement underlying legged locomotion (Raibert, 1986). A network of coupled central pattern generators whose output could be superposed with discrete trajectories has been proposed as an integration of discrete and rhythmic movement (S. Degallier, 2006), demonstrated in a drumming task. The discrete movement was not, however, timed in the sense we define this here. The learning of movement patterns in imitation scenarios has been successfully demonstrated using an oscillatory dynamical system as the substrate (A. J. Ijspeert, 2002; S. Schaal, 2003). Comparing the attractor dynamics approach to the potential field approach (Khatib, 1986) reveals a number of differences (B. R. Fajen, 2003). By choice of variable, movement is generated while the system is in an attractor rather than while the system is relaxing to an attractor. This facilitates the design of the dynamical systems, making it easier to avoid spurious attractors, and reduces oscillations. Because the motor plan has stability properties in the attractor dynamics approach, coupling to on-line sensory input is unproblematic. The generation of trajectory plans from limit cycle attractors supports stable timing. The notion of stabilizing movement time through a mechanism of on-line updating has been addressed in scenarios of target tracking and obstacle avoidance by a mobile robot (M. Tuma, 2009).

In this paper, we sketch how the attractor dynamics approach to autonomous robotics that was originally developed in the domain of vehicle motion (Schoener and Dose, 1992)(E. Bicho, 1996) and has been extended to trajectory generation in robotic manipulators (Iossifidis and Schoener, 2004) can be used to organize flexible and complex sequences of timed actions in a relatively complex ball hitting task scenario. The proposed architecture is able to initiate and terminate the different task movements through coupling to on-line sensory information and is implemented on a redundant manipulator platform.

The remainder of the paper is organized as follows : in Sec.2 the hardware setup, composed of a robotic and vision systems is presented. A detailed description of the robotic architecture including the timed trajectory generation is given in Sec.3. Experimental results are shown in Sec.4. Finally, a conclusion and outlook can be found in Sec.5.

2 SETUP

This section introduces the two main components of the experimental setup : the robotic system and the vision system.

2.1 Robotic System

The robotic demonstration of timed movement coordinated with perceived object motion employs the 8 (DoF) CoRA manipulator Fig. 1. CoRA is equipped with a vision system for tracking and prediction of a ball that moves on an inclined plane. The robot arm holds a paddle of 3.5[cm] in radius that serves as end-effector to intercept and hit the ball with a timed movement.

The end-effector trajectory is generated in the task space from which the corresponding joint angles are computed through the inverse kinematics closed form solution (Iossifidis and Schoener, 2004). The orientation of the end-effector is kept constant and an elevated elbow posture is chosen to accommodate CoRA's work space constraints. For real time performance, the implementaion was done in C++ under Linux environment.

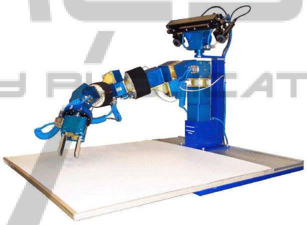


Figure 1: The anthropomorphic robotic assistant CoRA.

2.2 Vision System

The sensory information is acquired by a FireWire camera (SONY DFW VL500) mounted as part of the CoRA stereo vision system. The camera is oriented toward the inclined plane (Pan = 0[°], Tilt = 50[°]) and operated with a frame rate of 30 [f/s] providing a new measurement estimate every ≈ 33 [ms].

3 ROBOTIC ARCHITECTURE

The overall structure of the robotic demonstration of timed movement coordinated with perceived object motion is depicted in Fig. 2. it has three main modules : ball tracking, ball prediction and robot timed trajectory generation.

In the first module, ball tracking is performed based on visual sensor information. In the second module, a dynamical model of the ball is used to predict the necessary perceptual parameters **time_to_contact** (τ_{2c}) and **point_to_contact** (p_{2c}) based on the sensor feedback. The third module goes through the dynamics that genertate the different movement components that compose the ball hit-

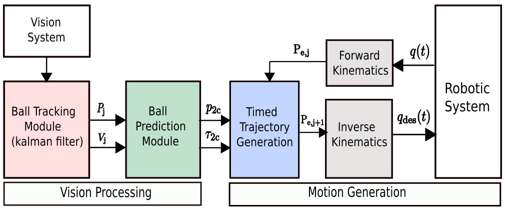


Figure 2: The robotic architecture.

ting task. The three modules will be described in the following subsections.

3.1 Ball Tracking Module

Different vision based techniques for object tracking exists (A. Yilmaz, 2006), reflecting different constraints on object motion and different processing time requirements. To achieve a robust ball tracking in the presence of image noise and non uniform lighting conditions, we use a color based object tracking process: first, by building an HSV histogram-based color model in HSV color space as an observation system and then we used a color-based segmentation process to estimate the ball position $p_b(t)$ in image plane (Fig. 3).

The position estimate is projected to the world coordinates by the camera transformation matrices. The position measurements are then fed to a Kalman filter (Schutter, 1999) to estimate the speed of the ball along the two movement axes of the inclined plane. The tracking process was implemented in a multiprocessing pipeline so that the vision system runs in parallel to the robot control process enabling real time performance with a relatively low processing time in the control loop.

3.2 Ball Prodiction Module

The ball position and speed estimates provided by the tracking module is then fed to the dynamics model of the ball Eq. 1 in order to predict the **time_to_contact** (τ_{2c}) and **point_to_contact** (p_{2c}).

$$\begin{aligned} V_{j+1} &= V_j + a_j \cdot \triangle t, \\ P_{j+1} &= P_j + V_j \cdot \triangle t + a_j \cdot \frac{\triangle t^2}{2}, \end{aligned} \tag{1}$$

Where a_j denotes the acceleration vector, V_j and P_j are the velocity and position vectors respectively. The main movement acceleration is along the y axes pointing along the inclination of the plane:

$$a_y = \frac{g \cdot \sin(\alpha) \cdot m \cdot r^2}{m \cdot r^2 + I} \tag{2}$$

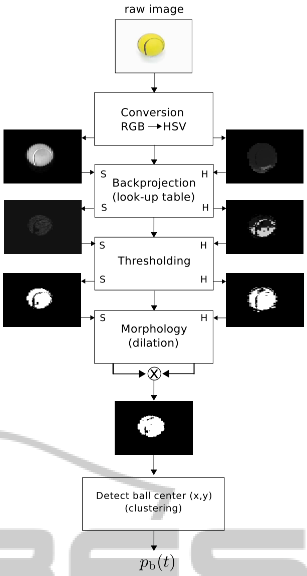


Figure 3: Color-based segmentation process.

Where m is the ball mass, g is the gravitational constant, α is the plane's inclination, r is the ball radius and I is the ball inertia. Horizontal acceleration, a_x, is mainly determined by the rolling friction Eq. 3 related to movement speed and direction. Elastic bounces of the ball off the side walls of the inclined plane are modeled through a coefficient of restitution C_r representing the ratio of speeds before and after the bounce and determined experimentally. The ball spin is difficult to observe and model, so its contribution is neglected as is the air drag which has negligible effects. The rolling friction, on the other hand, is modeled and is empirically tuned with a non negligible effect on the accelerations in both movement axes:

$$F_r = N \cdot \frac{U_r}{r} \tag{3}$$

Here N is the Normal force, r is the ball radius and Ur is the coefficient of rolling friction.

3.3 Timed Trajectory Generation

Robot trajectories are generated as stable solutions of a non linear dynamical system formulated on two layers of description.

The first level is a switching dynamics for the sequential organization of behaviors representing the start, execution, and termination phases of the movement, each calculated depending on sensor signals and internal measures.

At the second level of description, a dynamics for all behavioral variables that define the robot's kinematic state is integrated. The behavioral variables comprise the end-effector heading directions ϕ (azimuth) and θ (elevation) and the end-effector velocity $V_{\mathrm{e,j}}$. The generated end-effector trajectories $P_{\mathrm{e,j}} = (x_{\mathrm{e,j}}, y_{\mathrm{e,j}}, z_{\mathrm{e,j}})$ are transformed into joint angles trajectories $q_{\mathrm{des}}(t)$ using the inverse kinematics and a feedback measurement is accomplisched through the forward kinematics.

Running the vision system and the trajectory generator in parallel processes allows a robot movement step time of 20 [ms] with a negligible effect of the tiny dis-synchronization.

3.3.1 Movement Components

The ball hitting task consists of the following movement components:

- **The Interception Movement:** executed at constant speed along a predefined base line at the bottom and parallel to the inclined plane. This movement must bring the robot end-effector as close as possible to the predicted point_to_contact ($p_{2\mathrm{c}}$).
- **The Hitting Movement:** a timed movement generated on the basis of the predicted end-effector posture to intercept the approaching ball.
- **The Return Movement:** a movement taking a fixed amount of time and brings the robot end-effector back to the base line, ready to start another inteception-hitting-return task sequence whenever sensory information flags a predicted point_to_contact and time_to_contact within a defined window.

The hitting movement must be initiated just in time to hit the ball just before it has reached the bottom of the inclined plane. Different measures and factors are continuously monitored or updated by the visual system and included as parameters in the dynamical systems for trajectory generation and behavioral organization. These include reachability, **time_to_contact** ($\tau_{2\mathrm{c}}$) and **point_to_contact** ($p_{2\mathrm{c}}$). These parameters control the initiation of the hitting movement when the ball point_to_contact time is within the time_to_contact criterion and the ball is inside the robot's work space. The hitting movement is stopped if the ball falls out of the inclined plane after an unsuccessful hitting attempt so that the ball is no longer detected by the visual sensor.

3.3.2 Switching Dynamics

The switching dynamics are formulated for each of the three "neural" activation variables u_{i} which control the different behavioral states, $i = \mathrm{init}, \mathrm{hopf}, \mathrm{final}$ (see below), by specifying which contribution to the behavioral dynamics is active. Values near ± 1 mean the corresponding behavior is active, values near 0 mean the corresponding behavior is inactive. The switch between the different behavioral states of the trajectory generator dynamics depend on sensory information and internal conditions (e.g., reaching a critical time_to_contact, target reached,...). As these factors are continuously updated and may fluctuate, the activation states are stabilized. The switching dynamics is based on the normal form of a degenerate pitchfork bifurcation for each variable, coupled competitively :

$$\tau_{\mathrm{u}} \dot{u}_{\mathrm{i}} = \mu_{\mathrm{i}} u_{\mathrm{i}} - |\mu_{\mathrm{i}}| u_{\mathrm{i}}^3 - \nu \sum_{a \neq i} u_{\mathrm{a}}^2 u_{\mathrm{i}} \qquad (4)$$

Where the competitive advantages, μ_{i}, determine which activation variable is eligible for being turned on. In general, the activation variable with the highest positive competitive advantage is activated. The parameter, ν, controls the strength of competition and τ_{u} is the dynamics time constant. The values of $\nu = 2.1$ and $1.5 \leq \mu_{\mathrm{i}} \leq 3.5$ lead to a reasonable trade-off between stability and flexibility. Neuronal weights are linked to sensory and internal signals through a set of quasi-boolean parameters $b_{\mathrm{i}} \in [0, 1]$. These parameters are normalized to keep the competitive advantages within the desired range:

$$\mu_{\mathrm{i}} = 1.5 + 2 \cdot b_{\mathrm{i}} \qquad (5)$$

For a more detailed illustration of this switching mechanisms and its use in movement (re)initialization and termination see (Steinhage and Schoener, 1998).

3.3.3 Behavioral Dynamics

Based on estimates of the point_to_contact $p_{2\mathrm{c}} = (x_{\mathrm{tar}}, y_{\mathrm{tar}}, z_{\mathrm{tar}})$ and the current end-effector position $P_{\mathrm{e,j}} = (x_{\mathrm{e,j}}, y_{\mathrm{e,j}}, z_{\mathrm{e,j}})$, two heading directions ϕ_{tar} (azimuth) and θ_{tar} (elevation) are defined under which the target point is seen from the current position of the end-effector. These set attractor states for the end-effector, azimuthal and elevational, heading directions dynamics:

$$\dot{\phi} = \lambda_{\mathrm{tar}} \cdot sin(\phi - \phi_{\mathrm{tar}}) \qquad (6)$$

$$\dot{\theta} = \lambda_{\mathrm{tar}} \cdot sin(\theta - \theta_{\mathrm{tar}}) \qquad (7)$$

For more details see (Iossifidis and Schoener, 2004). At each time step, the arm end-effector velocity is set as the state variable x that evolves jointly with an auxiliary state variable y according to the following dynamical system:

$$\begin{aligned} \tau_{\mathrm{v}} \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} &= -c_1 \cdot u_{\mathrm{init}}^2 \begin{pmatrix} x \\ y \end{pmatrix} \\ &\quad + u_{\mathrm{hopf}}^2 \cdot f_{\mathrm{hopf}}(x - R_{\mathrm{h}}, y) \\ &\quad - c_2 \cdot u_{\mathrm{final}}^2 \begin{pmatrix} x \\ y \end{pmatrix} \qquad (8) \end{aligned}$$

Here c_1 and c_2 are scaling parameters and f_{hopf} is a Hopf oscillator of radius R_{h} (time dependant, see Eq. 11). The activation variables, $u_{\mathrm{i}} \in (u_{\mathrm{init}}, u_{\mathrm{hopf}}, u_{\mathrm{final}})$, defined in eq. 4 are used to switch between the three different regimes : stable oscillation (Hopf oscillation) activated by u_{hopf} and two initial and final fixed point attractors activated by u_{init} and u_{final}. The postural states are $(x,y) = (0,0)$, keeping the robot at rest before and after the movement. During the movement phase, the velocity dynamics is governed by a Hopf oscillator described by the normal form of the Hopf bifurcation:

$$\begin{aligned} f_{\mathrm{hopf}}(x - R_{\mathrm{h}}, y) = &\begin{pmatrix} \lambda & -\omega \\ \omega & \lambda \end{pmatrix} \begin{pmatrix} x - R_{\mathrm{h}} \\ y \end{pmatrix} \\ &- \gamma(x - R_{\mathrm{h}})^2 \\ &- \gamma y^2 \begin{pmatrix} x - R_{\mathrm{h}} \\ y \end{pmatrix} \end{aligned} \qquad (9)$$

The fact that the limit cycle attractor can be determined analytically facilitates the interpretation of the parameters: The angular frequency ω defines the cycle time $T = \frac{2\pi}{\omega}$ and hence the total movement time. The parameters $\lambda > 0$ and $\gamma > 0$ together set the oscillator radius R_{h}:

$$R_{\mathrm{h}} = \sqrt{\frac{\lambda}{\gamma}} \qquad (10)$$

The time scale of the velocity dynamics is set to a relaxation time of $\tau_{\mathrm{v}} = 0.02$, ten times slower than the relaxation time of the neuronal switching dynamics. Generating speed profiles instead of position trajectories, gives complete control over the robot movement speed for the hitting and return movements, a very important constraint in order to achieve safe and stable movement within the limits of the hardware. Combining the heading and velocity dynamics, the manipulator end-effector trajectories become fully defined. Timing and competitive neural dynamics of the trajectory generator are numerically integrated using the Euler method.

3.3.4 Temporal Stabilization

The presence of sensory noise and measurement errors in addition to the time varying prediction of the point_to_contact requires that the movement parameters are updated continuously. Therefore, the hitting movement must be parametrically updating while it is in execution. Continuously updating the target position ($p_{2\mathrm{c}}$) in the heading dynamics requires adapting the velocity dynamics to accelerate or decelerate according to the remaining distance to be traveled by the end-effector with the objective of keeping the hitting time as invariant as possible. We use this update rule for the amplitude of the Hopf oscillation R_{h} in Eq. 9:

$$R_{\mathrm{h}}(t) = \frac{\omega}{2\pi} \cdot D(t) / (1 - \frac{t}{T} + \frac{\sin(2\pi \cdot \frac{t}{T})}{2\pi}) \qquad (11)$$

Here, $D(t)$ is the remained distance to be traversed at the current speed. For a complete explanation of this update rule and a detailed illustration of its performance see (M. Tuma, 2009).

3.3.5 Behaviors Specifications

The autonomous initiation, continuation, and termination of the three movement components, **interception movement**, **hitting movement** and **return movement** are controlled by two sensory signals time_to_contact ($\tau_{2\mathrm{c}}$) and point_to_contact ($p_{2\mathrm{c}}$) as well as a number of internal conditions. All these factors are expressed through the quasi-boolean parameters. The conditions make use of a sigmoid function $\sigma(.)$ which returns values near 0 for negative argument and near 1 for positive argument. The predicted time_to_contact is used to predict the total rolling time of the ball on the inclined plane: (1) $\tau_{2\mathrm{c}} < 0$ if the ball is not detected or the ball trajectory cannot be predicted immediately after a hit; (2) $\tau_{2\mathrm{c}} > \tau_{\mathrm{crit}} > 0$ if the ball contact is not within a criterion time_to_contact; (3) $0 < \tau_{2\mathrm{c}} < \tau_{\mathrm{crit}}$ if the ball contact is predicted within a criterion time.

1) Interception Movement. The interception movement occurs at constant speed along the base line parallel to and at the bottom of the inclined plane, $(y = 379.6[\mathrm{mm}], z = 80[\mathrm{mm}])$. The movement is initiated and stopped by the parameter b_{inter} taking on values of 1 or 0, respectively.

$$\begin{aligned} b_{\mathrm{inter}} = &\,\sigma(d_{\mathrm{target}} - d_{\mathrm{offset}}) \\ &\cdot \sigma(\tau_{2\mathrm{c}} - \tau_{\mathrm{crit}}) \cdot \sigma(\tau_{2\mathrm{c}}) \end{aligned} \qquad (12)$$

Here the parameter d_{target} is the distance from the current end-effector position to the the target or point_to_contact $p_{2\mathrm{c}}$ computed continuously and d_{offset} is a small offset distance indicating that the target is reached. The choice of using a constant speed movement instead of a timed movement was motivated by the relatively major fluctuation of the early predicted landing position. Such strong fluctuations may induce an oscillator with the update rule to generate very large speeds that exceed the capacity of our hardware. The speed of the interception movement, I_{speed}, is generated through a simple attractor dynamics that switches smoothly between 0 and a safe maximum speed, max_{speed}:

$$\dot{I}_{\mathrm{speed}} = \lambda_{\mathrm{inter}} \cdot (I_{\mathrm{speed}} - (b_{\mathrm{inter}} \cdot max_{\mathrm{speed}})) \qquad (13)$$

Here λ_{inter} sets the strength of attraction.

2) Hitting Movement. When the ball contact is within a criterion time_to_contact $0 < \tau_{2c} < \tau_{\text{crit}}$, the hitting movement is initiated through the quasi-boolean parameter:

$$\begin{aligned} b_{\text{hit}} &= b_{\text{detected}} \cdot b_{\text{reachable}} \cdot (1 - \sigma(\tau_{2c} - \tau_{\text{crit}})) \\ &\quad \cdot (1 - m_{\text{ret}}) \cdot \sigma(\tau_{2c}) \end{aligned} \tag{14}$$

The parameter, b_{detected}, is a flag set to 1 by the vision system to indicate that the ball is being tracked. The parameter, $b_{\text{reachable}}$, is a flag set internally to 1 if the ball can be hit without crashing the end-effector into the borders of the workspace. m_{ret} is a flag set to 0 initially and set to 1 dynamically when the hitting movement is finished. Once the hitting movement is initiated, the Hopf oscillator generate the speed profile and together with heading dynamics, the hitting movement is started. Two other parameters $b_{\underline{\text{co}}\text{hit}}$ and $b_{\underline{\text{st}}\text{hit}}$, standing for continue the hitting movement and stop the hitting movement respectively, are evaluated continuously during movement generation:

$$b_{\text{cohit}} = b_{\text{detected}} \cdot (1 - f_{\text{maxspeed}}) \cdot \sigma(\tau_{2c}) \tag{15}$$

$$\begin{aligned} b_{\text{sthit}} &= (1 - \sigma(d_{\text{target}} - d_{\text{offset}})) \\ &\quad + \sigma(d_{\text{target}} - d_{\text{offset}}) \\ &\quad \cdot \sigma(t_{\text{current}} - hit_{\text{cycletime}}) \end{aligned} \tag{16}$$

Herein, d_{target} is the current distance from the current end-effector position to a new target position chosen to be beyond the actual predicted point_to_contact ($p_{2c} = (x_{\text{tar}}, y_{\text{tar}}, z_{\text{tar}})$) but at equal distance with the initial end-effector position before the hitting initiation $P_{e,0} = (x_{e,0}, y_{e,0}, z_{e,0})$ in order to hit the ball around the maximum speed. t_{current} is the current execution time and $hit_{\text{cycletime}} = 1.8[\text{s}]$ is the fixed hitting cycle time. Stopping is indicated when b_{cohit} is set to 0, for example, when the ball becomes invisible $b_{\text{detected}} = 0$ or when maximum speed exceeded by the movement plan, $f_{\text{maxspeed}} = 1$. The condition b_{sthit} is set to 1 if the target is reached or if the oscillator time is elapsed.

3) Return Movement. The return movement back to the base line is initiated when the hitting is finished or terminated:

$$b_{\text{ret}} = (1 - \sigma(\tau_{2c} - \tau_{\text{crit}})) \cdot (1 - m_{\text{hit}}) \tag{17}$$

The flag, m_{hit}, is set 1 initially and set to 0 dynamically when the return movement is finished. Like in the hitting movement, two other parameters b_{coret} and b_{stret} are tested continuously during speed generation:

$$b_{\text{coret}} = (1 - f_{\text{maxspeed}}) \tag{18}$$

4) Switching between Hitting and Return Movements. The flags m_{hit} and m_{ret} permit a correct sequencing between hitting and return movements so that the return movement is initiated only once the hitting has been initiated and terminated. Similarly, the hitting movement is executed only after returning to the base line. During the hitting movement, switching the flags m_{hit} to 0 and m_{ret} to 1 is done using two neural variables in a competitive dynamics Eq. 4 with these conditions:

$$b_{\text{mhit}} = b_{\text{cohit}} \cdot (1 - b_{\text{sthit}}) \tag{19}$$

$$b_{\text{mret}} = (1 - b_{\text{cohit}}) + b_{\text{cohit}} \cdot b_{\text{sthit}} \tag{20}$$

Similarly, during the return movement, switching the flags m_{ret} to 0 and m_{hit} to 1 is related to these conditions:

$$b_{\text{mret}} = b_{\text{coret}} \cdot (1 - b_{\text{stret}}) \tag{21}$$

$$b_{\text{mhit}} = (1 - b_{\text{coret}}) + b_{\text{coret}} \cdot b_{\text{stret}} \tag{22}$$

5) Maximum Speed Handling. The movement speed during hitting and return phases is constantly monitored and when the maximum speed is reached, the flag f_{maxspeed} is set to 1 through a neuron. If the flag f_{maxspeed} is set to 1 during the hitting movement, the Hopf oscillator is turned off by setting b_{hopf} to 0 and b_{final} to 1 as well as by setting the flags, m_{hit} to 0 and m_{ret} to 1, initiating the return movement. If the flag, f_{maxspeed}, is set to 1 during the return movement, an initialization movement is started through the parameter:

$$b_{\text{initmov}} = f_{\text{maxspeed}} \tag{23}$$

This movement brings autonomously the robot to its initial position (in the middle of the base line) through an oscillator with a relatively large cycle time. An internal $b_{\text{stinitmov}}$ flag controls the termination of this phase, and the flag f_{maxspeed} is set to 0 again. To organize the sequence of movement phases in Eq. 4 of the three movement types, the competitive advantages μ_{i} in Eq. 5 were chosen to depend on a set of logical conditions corresponding to the movement component, here we show the conditions for the hitting movement but the same conditions hold for the return and initialization movements just by replacing the parameters b_{hit}, b_{cohit} and d_{target} by the corresponding in each movement component:

$$b_{\text{init}} = (1 - b_{\text{hit}}) \tag{24}$$

$$\begin{aligned} b_{\text{hopf}} &= b_{\text{cohit}} \cdot (1 - \sigma(t_{\text{current}} - t_{\text{offset}})) \\ &\quad \cdot \sigma(d_{\text{target}} - d_{\text{offset}}) \end{aligned} \tag{25}$$

$$\begin{aligned} b_{\text{final}} &= (1 - b_{\text{cohit}}) \\ &\quad + b_{\text{cohit}} \cdot \sigma(t_{\text{current}} - t_{\text{offset}}) \\ &\quad + b_{\text{cohit}} \cdot (1 - \sigma(t_{\text{current}} - t_{\text{offset}})) \\ &\quad \cdot (1 - \sigma(d_{\text{target}} - d_{\text{offset}})) \end{aligned} \tag{26}$$

4 RESULTS & EVALUATION

In order to evaluate the overal robotic architecture for the ball hitting task, the following experiment was conducted : a ball is lauched on the inclined plane toward the top by a human operator. The ball trajectory was tracked by the vision system. Based on the predicted perceptual parameters τ_{2c} and p_{2c}, a timed trajectory for the end-effector is generated.

In the following, experiemntal results for one hitting task performed by the robot Fig. 1 is illustrated as well as numerical results of multiple trials performed in simulation.

4.1 Robot System Results

The vision system provides the trajectory generator with a new τ_{2c} and p_{2c} predictions every ≈ 33 [ms]. The latter typically reaches the necessary precision of < 4 [cm] for hitting only about a τ_{2c} of 100 [ms]. The maximum speed of the end-effector is set to 400 [mm/s] observed to be the maximum limit for a safe and stable movements. Fig. 4 shows the trajectories of the paddle and the ball when the CoRA manipulator successfully hits an approaching ball.

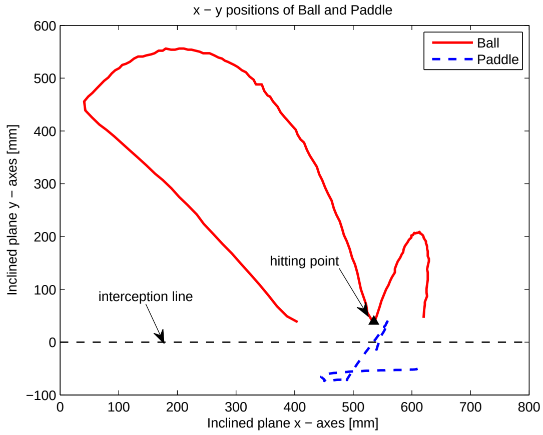


Figure 4: The movements of the paddle and the ball on the real setup for a successful hit. The hitting point is indicated by the black triangle.

The interception movement is started as soon as the ball is detected by the vision system and predicted by the kalman filter. When τ_{2c} becomes smaller than a critical value (here 1 [s]), the interception movement is stopped and the hitting movement is initiated by the parameter b_{hit} setting the boolean parameter b_{hopf} to 1 which triggers the corresponding neuron u_{hopf} to start the speed generation and with the heading dynamics, the end-effector is driven toward the p_{2c} to hit the ball.

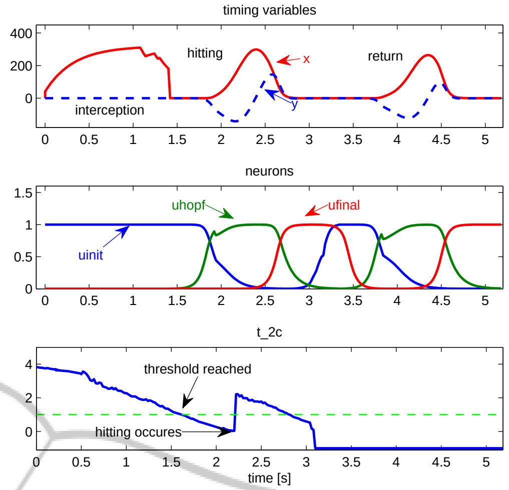


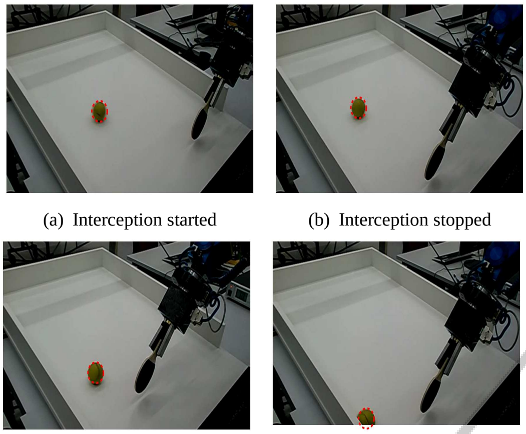
Figure 5: Trajectories of the timing and neural variables of an autonomous hit and return movements. The top two panels represent timing and neural variables. The bottom panel shows the time_to_contact, which crosses a threshold at about 1 [s]. At this moment, the arm initiates its timed movement.

The hitting is accomplished approximately at maximum movement speed, here 310 [mm/s], to provide the maximum momentum during the hit. When the hitting movement ends, the return movement, in a similar manner, is initiated immediately, b_{ret} set to 1, bringing the robot to the base line ready to initiate the next hitting sequence. These results show an inherent resistance of the timing sequence against sensor noise and the noisy prediction of the time_to_contact and point_to_contact which demonstrates the robustness of the approach. Fig. 5 shows the time courses of the relevant dynamics variables for this successful hit. Fig. 6 shows a sequence of snapshots of the movements involved to execute the ball hitting task.

4.2 Simulation Results

The approach was evaluated in a simulation framework of the CoRA manipulator coupled to sensory information about the real ball motion and among 500 random throws of the ball on the inclined plane, the system was able in 61% of the trials to cross the predicted point_to_contact in time with a mean standard deviation of the paddle-ball contact point of 2.2 [cm].

In 25% of the trials, the hit was not successful because of the inaccurate point_to_contact prediction. 14% of the trials led to an initialization movement. Setting the end-effector maximum speed to 550 [m/s] increases the hitting performance to 75%, which illustrates the effect of the hardware limitations.



(a) Interception started

(b) Interception stopped

(c) Hitting

(d) Return

Figure 6: The different movements composing a ball hitting task in the real robot.

5 CONCLUSIONS

Based on attractor dynamics, an autonomous mechanism of timed movement generation was implemented for a ball hitting task. The approach was evaluated in simulation and successfully brought onto a real manipulator, enabling the robot to repeatedly intercept and hit a ball rolling on an inclined plane. Stability and targetted instabilities are inherent properties of the nonlinear dynamics for trajectory generation and behavioral organization through which the generated behavior can be updated online and adjusted depending on the current situation. This behavioral flexibility enabled the robot to perform the hitting task autonomously. The system is capable of responding quickly and flexibly to changes in the sensed environment or movement conditions while keeping timing stable. Limitations in the system's performance were primarily due to the small framerate imposed by the employed camera. The robot's modest maximal speed affected the hitting performance on the real robot.

We are currently working on overcoming these limitations by improving the vision system's prediction performance through a faster camera and so, an improved Kalman estimation system. We will also transfer the implementation to a new robotic platform, the KUKA lightweight arm, which permits much faster movements and is, therefore, a more suitable platform for this application.

REFERENCES

A. J. Ijspeert, J. Nakanishi, S. S. (2002). Movement imitation with nonlinear dynamical systems in humanoid robots. In *IEEE Int Conf on Robotics and Automation*. ICRA.

A. Yilmaz, O. Javed, M. S. (2006). Object tracking: A survey. In *ACM Journal of Computing Surveys*. Vol. 38, No. 4.

B. R. Fajen, W. H. Warren, S. T. L. P. K. (2003). A dynamical model of visually-guided steering, obstacle avoidance and route selection. In *International Journal of Computer Vision*. 54(1-2) vol 54 13-34.

E. Bicho, G. S. (1996). The dynamic approach to autonomous robotics demonstrated on a low-level vehicle platform. In *International Symposium on Intelligent Robotic Systems*. SIRS 96.

Ijspeert, A. J. (2008). Central pattern generators for locomotion control in animals and robots. In *Neural Networks Review*. vol. 21, no. 4, pp. 642-653.

Iossifidis, I. and Schoener, G. (2004). Autonomous reaching and obstacle avoidance with the anthropomorphic arm of a robotic assistant using the attractor dynamics approach. In *IEEE Int Conf on Robotics and Automation*. ICRA.

Jeannerod, M. (1984). The timing of natural prehension movements. In *Journal of Motor Behavior*. vol 16, 235-254.

Khatib, O. (1986). Real time obstacle avoidance for manipulators and mobile robots. In *International Journal Robotics Research*. vol. 5, pp. 90-98.

M. Tuma, I. Iossifidis, G. S. (2009). Temporal stabilization of discrete movement in variable environments an attractor dynamics approach. In *IEEE Int Conf on Robotics and Automation*. ICRA.

Raibert, M. H. (1986). *Legged robots that balance*. MIT Press, Massachusetts, 1st edition.

S. Degallier, C. Santos, L. R. A. I. (2006). Movement generation using dynamical systems: a humanoid robot performing a drumming task. In *International Conference on Humanoid Robots*. IEEE-RAS.

S. Schaal, A. Ijspeert, A. B. (2003). Computational approaches to motor learning by imitation. In *Philosophical Transactions of the Royal Society*. vol 358 537-547.

Schoener, G. (1990). A dynamic theory of coordination of discrete movement. In *Biological Cybernetics 63*. vol 63, 257-270.

Schoener, G. (2002). Timing, clocks and dynamical systems. In *Brain and Cognition*. vol 48, pp. 31-51.

Schoener, G. and Dose, M. (1992). A dynamical systems approach to task-level system integration used to plan and control autonomous vehicle motion. In *Robotics and Autonomous Systems*. vol 10, 253-267.

Schutter, J. D. (1999). Kalman filters: A tutorial. In *Journal of Computing Surveys*. vol. 40 (4).

Steinhage, A. and Schoener, G. (1998). Dynamical systems for the behavioral organization of autonomous robot navigation. In *Sensor Fusion and Decentralized Control in Robotic Systems*. Proc SPIE.

Warren, W. H. (2006). The dynamics of perception and action. In *Psychological Review 113(2)*. vol 113, 358-389.