Generating timed trajectories for an autonomous vehicle: A non-linear Dynamical Systems Approach

Cristina Manuela Peixoto Santos
Dept. of Industrial Electronics
University of Minho
phone:+351253510190 fax:+351253510189
Email: cristina@dei.uminho.pt

***Abstract*— The timing of movements and of action sequences is important when particular events must be achieved in time-varying environments, avoiding moving obstacles or coordinating multiple robots. However, timing is difficult when it must be compatible with continuous on-line coupling to low-level and often noisy sensory information which is used to initiate and steer action. We extended the Dynamic Approach to Behavior Generation to account for timing constraints. We proposed a solution that uses a dynamical system architecture to autonomously generate timed trajectories and sequences of movements as attractor solutions of dynamic systems. The model consists on a two layer architecture, in which a competitive "neural" dynamics layer controls the qualitative dynamics of a second, "timing" layer. The second layer generates both stable oscillations and stationary states, such that periodic attractors generate timed movement. The first layer controls the switching between the limit cycle and the fixed points, allowing for discrete movements and movement sequences. This model was integrated with another dynamical system without timing constraints. The complete dynamical architecture was demonstrated on a vision-guided mobile robot in real time, whose goal is to reach a target in approximately constant time within a non-structured environment. The obtained results illustrated the stability and flexibility properties of the timing architecture as well as the robustness of the proposed decision-making mechanism.**

I. Introduction

The classical organization of autonomous robots separates both conceptually and in implementation task planning, trajectory planning and control (see, e.g., [8]). This separation implies that space and time constraints on robot motion must be known before-hand with the high degree of precision typical for non-autonomous robot operation, making it very difficult to work in unknown or natural environments. Moreover, such systems remain inflexible, cannot correct plans online, and thus fail both in non-static environments such as those in which robots interact with humans, or are mounted on mobile platforms, and in dynamic tasks or time-varying environments which are not highly controlled and may change over time, such as those involving interception, impact or compliance.

A reasonable requirement is that robust behavior must be generated in face of uncertain sensors and a dynamically changing environment, where there is a continuous online coupling to sensory information. Such requirement is particularly relevant in *Behavior-based approaches to autonomous robotics*, in which linkage between perception and action is attempted at low levels of sensory information [1]. In [6], real-time collision avoidance using potential fields has been achieved as well.

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 is not stabilized against perturbations. For instance, within the classical problem of moving in ordinary non-engineered environments, tasks commonly only require the vehicle to work at a convenient overall speed. If an obstacle has been circumnavigated such 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 in robot arm motion which requires control of timing in addition to control of spatial path. It is also critical in tasks which involve sequentially structured actions. This requires predictive planning, and thus inherently control of timing.

One type of solution for developing time schedules is to generate time structure at the level of control. Ref. [11], for instance, generated rhythmic action by inserting into dynamic control models terms that stabilized oscillatory solutions. Ref. [12] 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. Earlier, Ref. [4] obtained juggling in a simple manipulator by inserting into the control laws terms that endowed the complete system with a limit cycle attractor. Ref. [5] used limit cycle attractors to generate light seeking behavior and obstacle avoidance in a robotic vehicle. Ref. [20] exploits the properties of a simple oscillator circuit to obtain robust rhythmic robot motion control in a wide variety of tasks. More generally, the nonlinear control approach to locomotion pioneered by [11] amounts to using limit cycle attractors that emerge from the coupling of a nonlinear dynamical control system with the physical environment of the robot. A limitation of such approaches is that they essentially generate a single ongoing pattern of rhythmic movement. However, Ref. [13] has, like us, been able to generate temporally discrete movement as well. 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.

Over the last few years, an approach to planning with the help of dynamical systems, the *Dynamical Systems approach* *to autonomous robotics*, has been developed for the control of autonomous vehicles [17], [19], [7], [3]. This paper focuses on the extension of this approach to the timing of motor acts. Specifically, the aim is to autonomously generate timed trajectories and sequences of movements as attractor solutions of dynamic systems. The dynamic approach provides the theoretical concepts to integrate in a single model a theory of movement initiation, of trajectory generation over time and also provides for their control. These ideas have been formulated and tested as models of biological motor control in [14], [16] inspired by analogies with nervous systems and by the way the rhythmic movement patterns in legged locomotion are generated (e.g., [2], [5]). 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 [14]. The on-line linkage to sensory information can be understood through the coupling of these oscillators to time-varying sensory information [16]. Limited attempts to extend these theoretical ideas to temporally discrete movements (e.g., reaching) have been made [15].

We build on previous work, where we proposed an attractor based two-layer dynamics that autonomously generated timed movements and sequences of movements stably adapted to changing online sensory information [18]. In this paper, we integrate a timing architecture with another dynamical architecture which do not explicitly parameterize timing requirements. The timing model consists of 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 are obtained. System integration and behavioral organization is achieved through local sensor control and global task constraints expressed by logical interdependencies.

As an implementation of the approach, the capacity of a low level vehicle to navigate in a non-structured environment while being capable of reaching a target in an approximately constant time is chosen. The robot, that initially rests on the origin of an allocentric reference frame facing on a fixed direction, starts to orient towards target location, which is internally acquired by a visual system. After a certain time, the robot starts its timed trajectory towards the target while continuously avoiding sensed obstacles in its path. Robot velocity is controlled such that the vehicle has a fixed time to reach the target. The evaluation results illustrate the stability and flexibility properties of the timing architecture robustly adapted to changing online sensory information.

II. Attractor dynamics of heading direction

The robot action of turning is generated by letting the robot's heading direction, ϕ_h, measured relative to some allocentric reference frame, vary by making ϕ_h the behavioral variable of a dynamical system (for a full discussion see [17]). This behavioral variable is governed by a nonlinear vector field in which task constraints contribute independently by modelling desired behaviors as attractors and undesired behaviors as repellers of the overall behavioral dynamics. Integration of the *target acquisition* and *obstacle avoidance* contributions is achieved by adding each of them to the vector field that governs heading direction dynamics

$$\frac{d\phi_h}{dt} = F_{\mathrm{obs}}(\phi_h) + f_{\mathrm{tar}}(\phi_h) + F_{\mathrm{stoch}}(\phi_h). \qquad (1)$$

We add a stochastic component force, F_{stoch}, to ensure escape from unstable states within a limited time. The complete behavioral dynamics for heading direction has been implemented and evaluated in detail on a physical mobile robot [3].

III. The dynamical systems of driving speed

The path velocity, v of the vehicle is controlled through a dynamical system architecture that generates timed trajectories for the vehicle. We set two spatially fixed coordinates frames both centered on the initial posture, which is the origin of the allocentric reference frame: one for the x and the other for the y spatial coordinates of robot movement. A complete system of timing and neural dynamics is defined for each of these fixed coordinate frames. Each model consists of a timing layer [18], which generate both stable oscillations and two stationary states

$$\begin{aligned}\begin{pmatrix}\dot{x}_i\\ \dot{y}_i\end{pmatrix} &= 5\,|u_{\mathrm{init,i}}|\begin{pmatrix}x_i\\ y_i\end{pmatrix} + |u_{\mathrm{hopf,i}}| f_{\mathrm{hopf,i}}\\ &+ 5\,|u_{\mathrm{final,i}}|\begin{pmatrix}x_i - A_{\mathrm{ic}}\\ y_i\end{pmatrix} + \mathrm{gwn}, \qquad (2)\end{aligned}$$

where the index $i = x, y$ refers to timing dynamics of x and y spatial coordinates of robot movement. A neural dynamics controls the switching between the three regimes through three "neurons" $u_{j,i}$ ($j = \mathrm{init, hopf, final}$). Although only the variable, x_i, will be used to control motion of a relevant robotic task variable, a second auxiliary variable, y_i, is needed to represent oscillatory states. The "init" and "final" contributions generate stable stationary solutions at $x_i = 0$ for "init" and A_{ic} for "final" with $y_i = 0$ for both. These states are characterized by a time scale of $\tau = 1/5 = 0.2$.

The "Hopf" contribution to the timing dynamics is defined as follows:

$$\begin{aligned}\mathbf{f}_{\mathrm{hopf,i}} &= \begin{pmatrix}\alpha_h & -\omega\\ \omega & \alpha_h\end{pmatrix}\begin{pmatrix}x_i - \frac{A_{ic}}{2}\\ y_i\end{pmatrix}\\ &- \gamma_i\left(\left(x_i - \frac{A_{ic}}{2}\right)^2 + y_i^2\right)\begin{pmatrix}x_i - \frac{A_{ic}}{2}\\ y_i\end{pmatrix}, \qquad (3)\end{aligned}$$

where $\gamma_i = \frac{4\,\alpha_h}{A_{ic}^2}$ defines amplitude of Hopf i contribution. This "Hopf" contribution is the normal form of the Hopf bifurcation [10]. It generates a periodic solution (limit cycle attractor) with cycle time $T = 2\pi/\omega$ and amplitude $A_{ic}/2$ (x_i timing variables vary between the initial posture state at zero and the final postural state at A_{ic}). We use it because it can be completely analytically solved, providing complete control over its stable states [18].

The "neuronal" dynamics of $u_{\text{j,i}} \in [-1, 1]$ ($j = \text{init}, \text{final}, \text{hopf}$) switches each timing dynamics from the initial and final postural states into the oscillatory regime and back. The competitive dynamics are given by

$$\alpha_u \, \dot{u}_{\text{j,i}} = \mu_{\text{j,i}} \; u_{\text{j,i}} - |\mu_{\text{j,i}}| \; u_{\text{j,i}}^3 - \nu \sum_{a \neq j} u_{\text{a,i}}^2 \; u_{\text{j,i}} + \text{gwn}. \qquad (4)$$

This dynamics enforces competition among task constraints within the timing level depending on the neural competitive parameters ("competitive advantages"), μ_i. As the sensory information changes, the competitive parameters change, and a bifurcation occurs. 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). These parameters are explicitly designed such that their functions reflect the current sensorial context and the global constraints expressing which states are more applicable to the current situation.

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 [19], for a general framework for sequence generation based on these ideas and [18] for a description). Herein, the time, t, and target location, fully control the neural dynamics through the quasi-boolean parameters.

The competitive advantage of the initial postural state is controlled by the parameter b_{init}. This parameter must be "on" ($= 1$) when either of the following is true: (1) time, t, is bellow the initial time, t_{init}, set by the user ($t < t_{\text{init}}$); (2) timing variable x_i is close to the initial state 0 ($b_{\text{x}_\text{i} \text{ close } \text{x}_\text{init}}(x_i)$); **and** time exceeds t_{init} ($t \geq t_{\text{init}}$); **and** target has not been reached.

We consider that the target has not been reached when the distance, d_{tar}, from the actual robot position (as internally calculated through dead-reckoning) and the $(x_{\text{target}}, y_{\text{target}})$ position is higher than a specified value, d_{margin}. This logical condition is expressed by the quasi-boolean factor, $b_{\text{x}_\text{i} \text{ has not reached target}}(d_{\text{tar}}) = \sigma(d_{\text{tar}} - d_{\text{margin}})$, where $\sigma(\cdot)$ is a sigmoid function that ranges from 0 for negative argument to 1 for positive argument, chosen here as

$$\sigma(x) = [\tanh(10x) + 1]/2, \qquad (5)$$

although any other functional form will work as well. Note that this switch is driven from the sensed actual position of the robot.

The factor $b_{\text{x}_\text{i} \text{ close } \text{x}_\text{init}}(x_i) = \sigma(x_{\text{crit}} - x_i)$ has values close to one while the timing variable x_i is bellow $0.15A_{ic}$ and switches to values close to zero elsewhere.

These logical conditions are expressed through the mathematical function:

$$\begin{aligned} b_{\text{init}} &= 1 - \{(t \geq t_{\text{init}}) \\ &\quad [1 - (b_{\text{x}_\text{i} \text{ close } \text{x}_\text{init}}(x_i) \; (t \geq t_{\text{init}}) \\ &\quad\quad b_{\text{x}_\text{i} \text{ has not reached target}}(d_{\text{tar}}) \;) \;] \; \} \, . \end{aligned} \qquad (6)$$

A similar analysis derives the b_{hopf} and b_{final} parameters:

$$\begin{aligned} b_{\text{hopf}} &= (t \geq t_{\text{init}}) \; b_{\text{x}_\text{i} \text{ not close } \text{x}_\text{final}}(x_i) \qquad (7) \\ &\quad b_{\text{x}_\text{i} \text{ has not reached target}}(d_{\text{tar}}) \; \sigma\left(b_{\text{update A}_\text{ic}}\right) \\ b_{\text{final}} &= (t \geq t_{\text{init}}) \, [\; b_{\text{x}_\text{i} \text{ not close } \text{x}_\text{final}}(x_i) \; + \\ &\quad b_{\text{x}_\text{i} \text{ reached target}}(d_{\text{tar}}) \; + \; b_{\text{x}_\text{i} \text{ not close } \text{x}_\text{init}}(x_i) \\ &\quad + \; (1 - \sigma\left(b_{\text{update A}_\text{ic}}\right)) \;] \, . \qquad (8) \end{aligned}$$

We algorithmically turn off the update of the i timed target location, ${}^{T}x_{\text{target}}$ or ${}^{T}y_{\text{target}}$, once this changes sign relatively to the previous update and the corresponding timing level is in the initial postural state.

The factor $b_{\text{x}_\text{i} \text{ not close } \text{x}_\text{final}}(x_i) = \sigma(d_{\text{switch}} - d_{\text{crit}})$ is specified based on absolute values, where d_{switch} represents the distance between the timing variable x_i and the final postural state, A_{ic} and d_{crit} is tuned empirically.

The competitive dynamics are the faster dynamics of the all system. Its relaxation time, τ_u, is set ten times faster than the relaxation time of the timing variables ($\tau_u = 0.02$). This difference in time scale guarantees 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, this difference in time scales makes it possible to treat x as a parameter in the neural dynamics. Conversely, from the view point of the timing dynamics, the neural weights can be assumed to have already relaxed to their corresponding fixed points (adiabatic elimination).

The system is designed such that the planning variable is in or near a resulting attractor of the dynamical system most of the time. The maximal rate of shift, $\dot{\psi}_{max}$, of the fixed points is a function of the object's relative velocity and the distance d between the robot and the object (target or obstacle) (see [3] for full discussion). Thus, if we control the driving velocity, v, of the vehicle, we limit the rate of such shifts and the system is able to track the moving attractor.

The velocities we want to achieve, V_{obs} or V_{timing}, are imposed by a dynamics similar to that described by [3]. V_{obs}, is computed as a function of distance, where $\dot{\psi}_{max}$ is a design parameter. The path velocity, V_{timing}, as planned by the timing dynamical level is set as:

$$V_{\text{timing}} = \sqrt{\dot{x}_x + \dot{x}_y} \; . \qquad (9)$$

Suppose that at $t = 0$ s the robot is resting at an initial fixed position. The robot rotates in the spot in order to orient towards the target direction. At time t_{init}, the quasi-boolean for motion, b_{hopf}, becomes one, triggering activation of the corresponding neuron, u_{hopf}, and movement initiation. Movement initiation is accomplished by setting the driving speed, v, different from zero. During periodic movement, the target location in time is updated each time step based on error, $x_\mathrm{R} - {}^{T}x_x$, such that

$$^{T}x_\mathrm{target} \quad = \quad x_\mathrm{target} - \left(x_\mathrm{R} - {}^{T}x_x\right), \tag{10}$$

where ${}^{T}x_x$ is the current timing variable x_x. The periodic motion's amplitude, A_{xc}, is set as the distance between ${}^{T}x_\mathrm{target}$ and the origin of the allocentric reference frame (which is coincident with the x robot position previously to movement initiation), such that

$$A_{xc}(t) \quad = \quad {}^{T}x_\mathrm{target}(t). \tag{11}$$

In case an obstacle is detected, velocity is set according to the current distance to the obstacle.

The periodic solution is deactivated again when the vehicle comes into the vicinity of the x timed target, and the final postural state is turned on instead (neurons $|u_\mathrm{hopf}| = 0$; $|u_\mathrm{final}| = 1$). At this moment in time, the x timed target location is no longer updated in the timing dynamics level. The same behavior applies for the timing level defined for the y spatial coordinate.

IV. Experimental results

The dynamic architecture was implemented and evaluated on an autonomous wheeled vehicle [3]. Image processing has been simplified by working in a structured environment, where a red ball lies at coordinates $(x_B, y_B) = (-0.8, 3.2)$ m on the top of a table at approximately 0.9m tall. The sensed obstacles do not block vision.

The dynamics of heading direction, timing, competitive neural, path velocity and dead-reckoning equations are numerically integrated using the Euler method. An image is acquired only every 10 sensorial cycles such that the cycle time is 70 ms, which yields a movement time (MT) of 14s. Forward movement only starts for $t_\mathrm{init} = 3s$.

A. Properties of the generated timed trajectory

The sequence of video images shown in Fig. 1 illustrates the robot motion in a very simple scenario: during its path towards the target, the robot faces two obstacles separated of 0.7m, which is a distance larger enough for the robot to pass in between. The detailed time courses of the relevant variables and parameters are shown in Fig. 2. In case timing dynamics stabilize the velocity dynamics the robot is strongly accelerated in order to compensate for the object circumnavigation.

Fig. 3 illustrates the robot trajectory as recorded by the dead-reckoned robot position when the distance between the two obstacles (0.3 m) is not enough for the robot to pass in between them. In such case, the path followed by the robot is qualitatively different. Fig. 4 illustrate the robot's behavior when the environment is more complex.

These trajectories display a number of properties of dynamical decision making. The system is able to make decisions such that it flexibly responds to the demands of any given situation while keeping timing stable. Stability is displayed in the sense that the approach is robust to the presence of noise. Finally, Fig. 2(a) shows how the hysteresis property

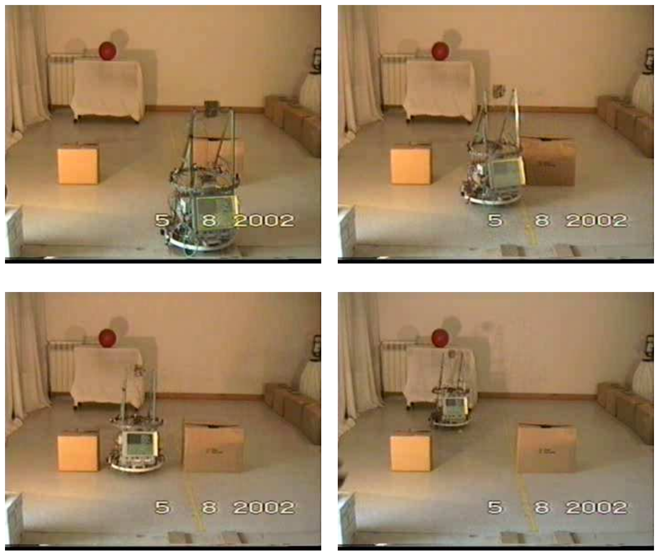


Fig. 1. A sequence of video images illustrates robot motion when the robot faces two objects separated of 0.7m during its path. The robot successfully passes through the narrow passage towards the target and comes to rest at a distance of 0.9m near the red ball. The overall generated timed trajectory takes $t = 16.6 - 3$ s to reach the target. The effective movement time is 13.6 s for a movement time of 14s.

allows for a special kind of behavioral stability. At $t = 15$s the quasi-boolean parameter $b_\mathrm{x,hopf}$ becomes zero but the $u_\mathrm{x,hopf}$ neuron remains active until the neuron $u_\mathrm{x,final}$ is more stable, what happens around $t = 16.2$ s. At this time, the x periodic motion is turned off. Thus, hysteresis leads to a simple kind of memory which determines system performance depending on its past history and enables the system to be robust to ambiguity in the environment.

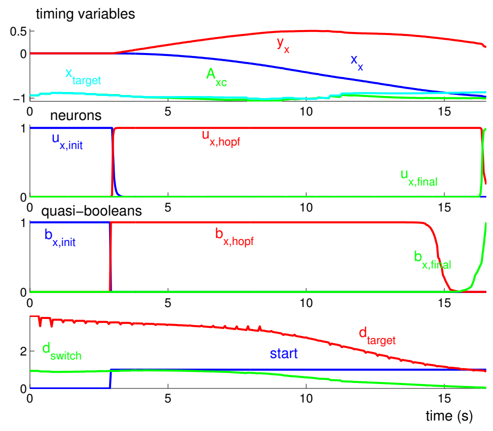
B. Trajectories generated with and without timing control

Table I surveys the time the robot takes to reach this target for several configurations when path velocity is controlled with and without timing control. In the latter, path velocity is specified differently: when no obstructions are detected the robot velocity is stabilized by an attractor, which is set proportional to the distance to the target [3]. Note that forward movement starts immediately. Conversely, forward movement only happens at $t = 3$ s when there is timing control. The specified movement time is 14s. We observe that both controllers have stably reached the target but the former is capable of doing it in an approximately constant time independently of the environment configuration.

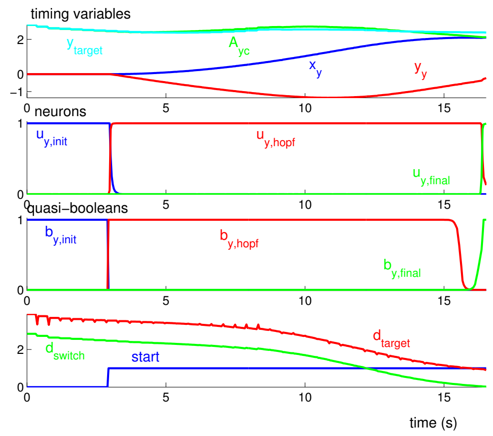
We have also compared the time the robot takes to reach the target for different target locations. The results have shown that the achieved movement time is approximately constant and independent of the distance to the target.

V. Discussion

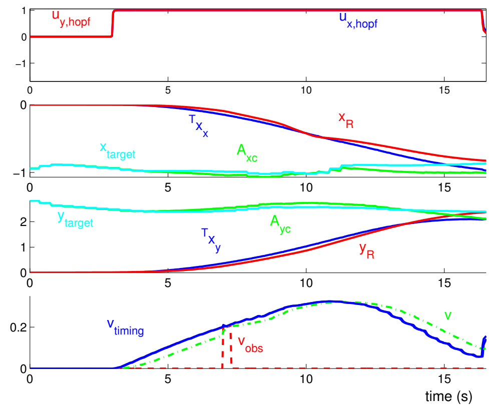
This paper addressed the problem of generating timed trajectories and sequences of movements for autonomous vehicles when relatively low-level, noisy sensorial information is used to steer action. The developed architectures are fully



(a) Timing and neural dynamics in x coordinate.



(b) Timing and neural dynamics in y coordinate.



(c) Top panel depicts both neural variables. The next two panels represent timing variables, robot trajectories, the real target locations and the periodic motion amplitudes. The bottom panel depicts velocity variables.

Fig. 2. Time courses of variables and parameters for the robot trajectory depicted in Fig. 1.

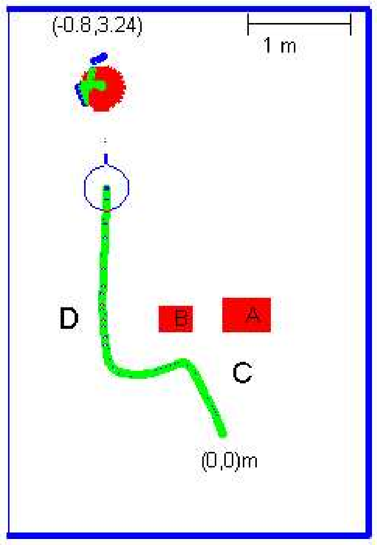


Fig. 3. A timed trajectory as recorded by the dead-reckoning mechanism. During its trajectory to the target, the robot is faced with two obstacles which are separated of 0.3m. The robot smoothly circumnavigates the obstacles and comes to rest near the ball. The ball position as calculated by the visual system slight differs from the real robot position (indicated by the line over the ball).

TABLE I

TIME (IN SECONDS) THE ROBOT TAKES TO REACH THE TARGET BOTH FOR A DYNAMIC ARCHITECTURE WITH AND WITHOUT TIMING CONTROL.

Experiments	**timing (MT)**	**without timing**
No obstacles	16.8 (13.8)	18.6
one obstacle	16.6 (13.6)	19.0
obstacles separated 0.8m	16.5 (13.5)	18.9
obstacles separated 0.7m	16.6 (13.6)	19.0
obstacles separated 0.3m	18.8 (15.8)	23.6
Complex configuration 1	19.3 (16.3)	22.5
Complex configuration 2	17.2 (14.2)	19.2
Complex configuration 3	17.4 (14.4)	19.6
Complex configuration 4	16.8 (13.8)	19.7
Complex configuration 5	17.3 (14.3)	18.3
Complex configuration 6	17.7 (14.7)	19.8
Complex configuration 7	23.3 (20.3)	28.0

formulated in terms of nonlinear dynamical systems which lead to a flexible timed behavior stably adapted to changing online sensory information. The model consists of a timing layer with either stable fixed points or a stable limit cycle. The qualitative dynamics of this layer is controlled by a "neural" competitive dynamics. By switching between the limit cycle and the fixed points, discrete movements and sequences of movements are obtained. These switches are controlled by the parameters of the neural dynamics which express sensory information and logical conditions. Further, we have shown that by manipulating the timing of a limit cycle the system performed well tasks with complex timing constraints.

The dynamical systems approach has various desirable properties. Firstly, we guarantee the stability and the controllability of the overall system by obeying the time scale separation principle. In addition, we can use properties, such as stability,

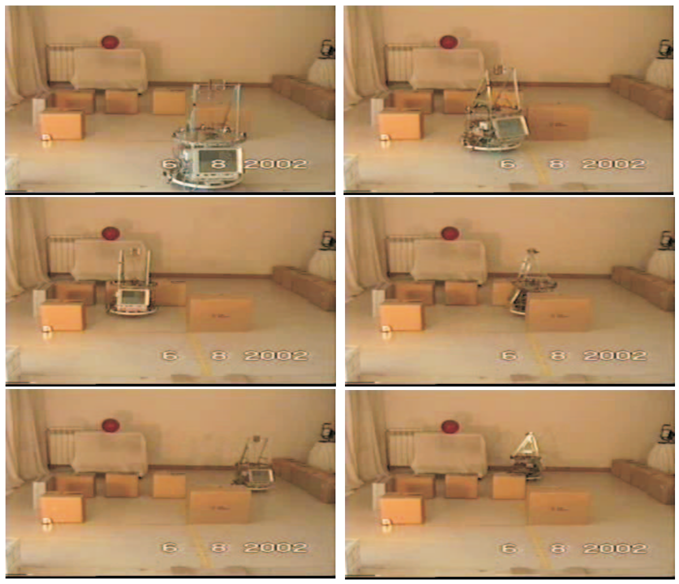


Fig. 4. Robot motion in a more complex environment named of *complex configuration 7* in which the robot is faced with a barrier of obstacles behind which lies the ball. The robot comes to rest at 0.9m from the target, at $t = 23.29$ s (the effective movement time is around 20s).

bifurcation, and hysteresis, which enable planning decisions to be made and carried out in a flexible, yet stable, way even if noisy sensory information is used to steer action. Secondly, a globally optimized behavior is achieved through local sensor control and global task constraints expressed by logics contained in the parameters of the differential equations.

The ease with which the system is integrated into larger architectures for behavioral organization that do not necessarily explicitly represent timing requirements is a specific advantage of our formulation. This scalability property implies a high modularity. Another advantage is the fact that it is possible to parameterize the system by analytic approximation, which facilitates the specification of parameters. Not only we have generated discrete movement as well as we provide a theoretically based way of tuning the dynamical parameters to fix a specific movement time or extent.

As a case study, we have addressed the generation of trajectories with stable timing for a low-level autonomous vehicle which must reach a target within a certain time independently of the environment configuration or the distance to the target. The implemented decision making mechanism allowed the system to flexibly respond to the demands of the sensed environment at any given situation. The generated sequences were stable and a decision maintained stable by the hysteresis property. Thus, stable coupling to unreliable timing information is achieved. The described implementation in hardware probes how the inherent stability properties of neural and timing dynamics play out when the sensory information is noisy and unreliable.

Future work will address how to endow the timed dynamical model discussed here with cognitive capabilities, how to extend the described model to achieve more complex behavior, how to integrate this approach with the dynamical systems approach to generate formation control and how to incorporate the ability of using learning. 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 [9].

Acknowledgment

This project was supported through grant PRAXIS XXI/BD/16163/98 from the FNICT (Portugal).

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] Estela Bicho, Pierre Mallet, and Gregor Schöner. Target representation on an autonomous vehicle with low-level sensors. *The International Journal of Robotics Research*, 19(5):424–447, May 2000.

[4] M Bühler, D E Koditscheck, and R D Skinner. Planning and control of a juggling robot. *International Journal of Robotics Research*, 13(2):101–118, 1994.

[5] M R Clark, G T Anderson, and R D Skinner. Coupled oscillator control of autonomous mobile robots. *Autonomous Robots*, 9:189–198, 2000.

[6] O Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *International Journal Robotics Research*, 5(1):90–98, 1986.

[7] E W Large, H I Christensen, and R Bajcsy. Scaling the dynamic approach to path planning and control: Competition amoung behavioral constrains. *International Journal of Robotics Research*, 18(1):37–58, 1999.

[8] J. C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1998 fifth edition, 1991.

[9] 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.

[10] L. Perko. *Differential Equations and Dynamical Systems*. Springer-Verlag, Berlin, second edition, 1991.

[11] M H Raibert. *Legged robots that balance*. MIT Press, Cambridge, Massachusetts, 1986.

[12] Stefan Schaal and Christopher G. Atkeson. Open loop stable control strategies for robot juggling. In *IEEE International Conference on Robotics and Automation*, pages 913–918, Georgia, Atlanta, 1990.

[13] Stefan Schaal, Shinya Kotosaka, and Dagmar Sternad. Nonlinear dynamical systems as movement primitives. In *IEEE International Conference on Humanoid Robotics*. IEEE, Cambridge, MA, 2000.

[14] G Schöner and J A S Kelso. Dynamic pattern generation in behavioral and neural systems. *Science*, 239:1513–1520, 1988.

[15] Gregor Schöner. A dynamic theory of coordination of discrete movement. *Biological Cybernetics*, 63:257–270, 1990.

[16] Gregor Schöner. Dynamic theory of action - perception patterns: The time-before-contact-paradigm. *Human Movement Science*, 3:415–439, 1994.

[17] Gregor Schöner and Michael 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.

[18] Gregor Schöner and Cristina Santos. Control of movement time and sequential action through attractor dynamics: A simulation study demonstrating object interception and coordination. In *9th Intelligent Symposium on Intelligent Robotic Systems - SIRS'2001*, Toulouse, France, 18-20,July 2001.

[19] Axel Steinhage and Gregor Schöner. Dynamical systems for the behavioral organization of autonomous robot navigation. In In McKee G T Schenker PS, editor, *Sensor Fusion and Decentralized Control in Robotic Systems: Proceedings of Spie-Intelligent Systems Manufactors*, pages 169–180, Boston, 1998. SPIE-publishing.

[20] Matthew Williamson. Rhythmic robot arm control using oscillators. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'98)*, Victoria, B.C., Canada, October 1998.