

Generating Trajectories With Temporal Constraints for an Autonomous Robot

Jorge Bruno Silva

Industrial Electronic Department, University of Minho
Guimares, Portugal
jbruno@dei.uminho.pt

Vitor Matos

University of Minho
Guimares, Portugal
vmatos@dei.uminho.pt

Cristina Santos

University of Minho
Guimares, Portugal
cristina@dei.uminho.pt

Abstract — Trajectory modulation and generation are two fundamental issues in the path planning problem in autonomous robotics, specially considering temporal stabilization of the generated movements. This is a very critical issue in several robotic tasks including: catching, hitting, and human-robot scenarios.

In this work, we address these problems and focus on generating movement for a mobile robot, whose goal is to reach a target within a constant time. We use an Hopf oscillator whose solution controls velocity, adapted according to temporal feedback. We have also proposed an adaptive mechanism for frequency modulation of the velocity profile that enables setting different times for acceleration and deceleration.

This approach is demonstrated on a DRK8000 mobile robot in order to confirm the system's reliability with low-level sensors.

Keywords: *Timing, Nonlinear dynamical systems, Adaptive modulation*

I. INTRODUCTION

An additional but fundamental issue in the path planning problem is temporal stabilization of movements, particularly in discrete movements. Temporal stabilization means to keep as much as possible the planned movement time (MT), despite varying environmental conditions or perturbations. Movement must be compensated for when these disturbances either hold up, accelerate or decelerate it.

This is a very critical issue in robotic tasks including: avoidance of moving objects; catching; hitting; juggling; coordination between degrees of freedom; human-robot scenarios and generating sequentially structured actions. These tasks may include external temporal boundary conditions, which elicit and drive movement generation.

In this contribution we continue previous attempts [1], [2], [3] to tackle these spatial-temporal problems. We propose a controller inspired in the vertebrate biological motor systems [4], [5], [6], [7], structured in two functional hierarchical layers according to their level of abstraction.

We apply autonomous nonlinear dynamical systems to formulate each of these layers, since their intrinsic properties apply ideally to the problem. These include: (1) low computational cost; (2) smooth online modulation while keeping the general features of the original movements; (3) robustness against small perturbations; (4) and allows integration of sensory feedback [8], [9].

However, a major drawback in the dynamical systems approach is the complexity inherent to their design. In order to tackle this drawback, we follow a modular approach by assuming that complex movement can be generated from the combination of simpler motor primitives, discrete and rhythmic, implemented as dynamical systems. Additionally, this modularity is assumed in terms of motor behaviors, stored as motor programs in the nervous system [10]. These assumptions turn a possibly high dimensional trajectory generation problem into a simple selection between pre-defined behaviors.

The lower level of the proposed controller acts out at the level of heading direction and velocity by formulating two separate dynamical systems. An attractor-based dynamics is formulated for the heading direction variable [11]. The velocity dynamics is based on past work for modeling a CPG [9], [1], [8]: a unique set of (oscillator-based) differential equations is able to produce complex movements modeled as periodic movements around time-varying offsets. The generated velocity profile and the switch between movement primitives can be easily modulated according to changes in values of the dynamical parameters.

The second level is responsible for selecting the most adequate motor behavior; map it onto the corresponding set of dynamical parameters and send them to the lower level at the right timing. A neural competitive dynamics models this second level. Its outcome are neural variables whose states encode the adequate set of parameters for the corresponding behavior. This switching dynamics autonomously bifurcate among possible behaviors, when relatively low level, noisy sensory information is used to initiate and steer action.

A great advantage of the proposed approach is that the proposed controller is simpler and more stable compared to others [2], [12], [1], it is generic enough to be implementable on a wide range of robots and allows to include other behaviors and thus generate more complex behavior. Further, the developed approach is implemented in a lower level robot, with modest computational resources.

This approach is demonstrated on a DRK8000 mobile robot in order to confirm the system's reliability with low-level sensors. The robot navigates towards a target in a world and is confronted with unexpected disturbances, such as obstacles or

sudden movements of the target. Results show that the system is able to deal with these disturbances and reaches the target still respecting the initial desired movement time.

II. RELATED WORK

Some approaches using the dynamical systems theory, have addressed this timing problem. The framework proposed in [3] extends the attractor dynamics approach of behavior generation to the timing domain. This framework is applied in [1] to an autonomous vehicle generating timed movement, where a temporal stabilization mechanism was proposed. However, [2] argues that the included temporal stabilization mechanism does not guarantee invariant movement time, and propose some changes including: (1) to use the dynamic solution to directly control the robot's velocity; (2) to use a full oscillator cycle; and (3) propose an adaptive rule that theoretically warrants invariant movement time.

In this work, a few novel adaptation mechanisms and properties of the system were proposed by further exploring the intrinsic properties of the used oscillator.

We propose an adaptive mechanism for frequency modulation of the velocity profile based on the oscillators current state, that enables setting different times for acceleration and deceleration.

Previously, it was assumed that local bifurcation theory could not be used and hence was difficult to build a general model [13] for generation of more complex movements that include discrete; rhythmic and superimposition of both. Therefore, proposed solutions include application of a neural competitive dynamics to switch among different contributions, modeled as dynamical systems. However, care must be taken such that this switch occurs where vector fields of contributions are almost similar, such that the state is continuous. Otherwise, there will be a discontinuity. Further, the neuronal dynamics may present some multistability, meaning that more than one neuron is "on" and the vector field results from the sum of the active contributions. Herein, these problems were eliminated by using bifurcation theory to switch the qualitative dynamics of the Hopf nonlinear system, instead of switching among different dynamical systems.

III. SYSTEM OVERVIEW

The presented system is divided hierarchically in two functional levels according to their level of abstraction. Both of the levels are implemented through sets of interacting dynamical systems.

The second level selects and sequences the most appropriate motor behaviors accordingly to external conditions and current states of the task at hand. Three motor behaviors are available in the level's repertoire:

- 1) Stop: The robot does not move, as it awaits for the beginning of the task or has reached the target.
- 2) Execution: The robot performs the movement task. It starts and executes the pursuit of the target, avoiding eventual obstacles, while adapting its velocity to maintain the task's timing.

- 3) Rescue: In the eventual case of the target becoming out of reach within the time constraint due to an excessive disturbance, the robot maintains a constant, stable velocity to the target.

The first level is composed by the dynamics of two behavioral variables that control the 2D motion of the robot, its heading direction ϕ and forward velocity v . The directional dynamics governs the heading direction according to: the perceived target position as an attractor, and repellers erected by the detection of obstacles; safely steering the robot around obstacles into the target.

The velocity v of the robot is the major determinant to the success of the movement task within the specified timing constraints. Its value is the solution of the stable limit-cycle generated by the Hopf oscillator. Commands are sent from the second level, changing qualitatively the dynamic solution by taking advantage of the Hopf bifurcation, resulting in the three presented motor behaviors. The performed velocity consists of a single oscillation cycle, adapted in order to accelerate or decelerate in face of disturbances.

IV. HEADING DIRECTION CONTROL

The robot's heading direction, ϕ_h , in angular space and in an allocentric coordinate, is controlled by a nonlinear vector field in which task constraints contribute independently. The task of reaching the target, $F_{tar}(\phi_h)$, attracts ϕ_h towards the direction in which the target lies. The task of avoiding obstacles, $F_{obs}(\phi_h)$, repels ϕ_h from the direction in which obstacles are perceived.

Integration of these tasks is achieved by adding each of them to the vector field that governs heading direction dynamics.

$$\dot{\phi}_h = F_{obs}(\phi_h) + F_{tar}(\phi_h) + F_{stoch}. \quad (1)$$

A stochastic component, F_{stoch} , is added to ensure an escape from unstable states. For a full discussion see [14], [11], [1] for examples.

This approach differs from the potentially field approach basically in the aspect that the state of the behavioral system must be in or near an attractor state of the dynamical system during operation.

V. VELOCITY CONTROL

Robot velocity is generated by the m variable of the following Hopf oscillator:

$$\dot{m} = \alpha(\mu - r^2)(m - O_m) - \omega n, \quad (2)$$

$$\dot{n} = \alpha(\mu - r^2)n + \omega(m - O_m), \quad (3)$$

$$r = \sqrt{(m - O_m)^2 + n^2}, \quad (4)$$

where m and n are the state variables, amplitude of the oscillations (limit cycle radius) is given by $A = \sqrt{\mu}$ for $\mu > 0$ and ω specifies the oscillations frequency (rads^{-1}). Variable O_m is used to control the m solution offset.

This oscillator contains an Hopf bifurcation from a stable fixed point at $(m, n) = (O_m, 0)$ (when $\mu < 0$) to a structurally stable, harmonic limit cycle, for $\mu > 0$. Relaxation to these solutions is given by $\frac{1}{2\alpha\mu}$ (s).

In summary, this system is able to generate: (1) A discrete movement to a offset O_m , if $\mu < 0$; (2) a rhythmic movement around O_m , if $\mu > 0$; and (3) the superimposition of both movements, resulting in a more complex movement, if $\mu > 0$ and the offset is defined as a time-changing variable [8].

Additionally, we motivate the choice of this Hopf oscillator because it enables to explicitly modulate the generated trajectories with respect to their amplitude, frequency and offset, according to small parameter changes while keeping the general features of the original movements.

This is interesting for trajectory generation in a robot [8].

A. Profile modulation

The oscillator described by eqs. (2)(3)(4), generates a m harmonic solution which period equals the movement time $MT = \frac{2\pi}{\omega}$. Herein, MT is the time that velocity goes from zero to twice the oscillator radius, A , and back to zero again, performing a full sinusoidal cycle (fig. 1 (top)). However, ascending and descending parts of the oscillatory cycle have equal durations, meaning that a large amount of time is spent accelerating towards the maximum required velocity and decelerating back to zero again. From a robot physical perspective, it would be beneficial if minor top velocities are requested, keeping the profile approximately constant as long as possible. Therefore, we understand that ideally it should be possible to choose the acceleration/deceleration time durations within the task's movement time.

Consider the velocity profile depicted in fig. 1 (top). The velocity evolves as follows:

$$v(t) = A(1 - \cos(\omega t)) \quad (5)$$

Lets now consider the same task of reaching a target, but subdividing the required velocity profile in three time intervals, each with different durations, such their sum results in the movement time $MT = T_1 + T_2 + T_3$. During 0 to t_1 the oscillator covers the first quarter of the limit cycle (T_1), half of the limit cycle is covered from t_1 to t_2 (T_2), and the last quarter from t_2 to t_3 (T_3).

Angular frequency ω is calculated for each of the three time intervals, such that in the overall they are performed within the correct timing, as follows:

$$\omega_1 = \frac{\pi}{2T_1}, \quad \omega_2 = \frac{\pi}{T_2}, \quad \omega_3 = \frac{\pi}{2T_3}. \quad (6)$$

The distance s covered during each time interval is calculated by integrating eq. (5) during the given time intervals, for a fixed radius cycle A :

$$s_1 = \frac{A(\frac{\pi}{2} - 1)}{\omega_1}, \quad s_2 = \frac{A(\pi + 2)}{\omega_2}, \quad s_3 = \frac{A(\frac{\pi}{2} - 1)}{\omega_3}. \quad (7)$$

The total distance covered by the robot, is the distance needed to reach the target.

$$D(t=0) = s_1 + s_2 + s_3, \quad (8)$$

where $D(t=0)$ is the distance between the robot's initial position and the initial target coordinates at instant $t = 0s$.

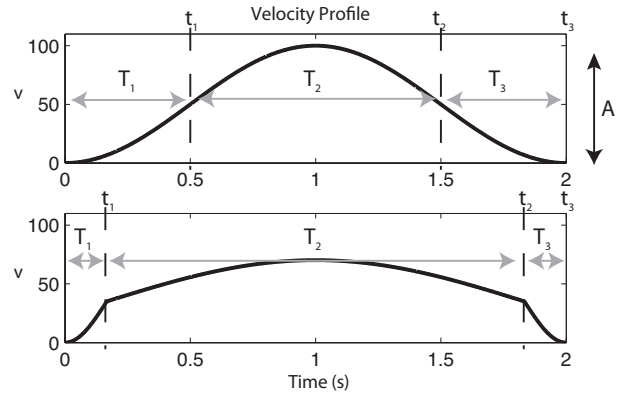


Fig. 1. Top: simple oscillation profile generated by the Hopf oscillator. On this profile, T_1 and T_2 are longer, resulting in a curve with a higher top velocity. Bottom: modulated oscillation profile where the acceleration and deceleration times are smaller, resulting in a smaller top velocity.

In an undisturbed scenario in which no disturbances (obstacles, target displacement, etc) occur, the fixed radius A can be calculated as follows:

$$A = \frac{D(t=0)}{\frac{(\frac{\pi}{2}-1)}{\omega_1} + \frac{(\pi+2)}{\omega_2} + \frac{(\frac{\pi}{2}-1)}{\omega_3}} \quad (9)$$

B. Timing adaptation

However, in a real implementation disturbances of many types may occur which disturb the online planned robot time course. Hence, the robot must be able to accelerate or decelerate along its path, and still reach the target within the defined movement time. In [2] was proposed an online adaptation rule for the Hopf radius which was able to guarantee that the remaining distance is traversed within the remaining time.

Building on the same formulation, we integrate eq. (5) during each of the three time intervals, thus determining an online updating rule for the Hopf offset and radius cycle A , that considers both the current distance to the target and the remaining time to do it. The adaptive rule is calculated by different formulae, depending on the current instant of time, For $0 < t < t_1$:

$$A_1(t) = \frac{D(t)}{\frac{\frac{\pi}{2}-1+\sin(\omega_1 t)}{\omega_1} + \frac{\pi+2}{\omega_2} + \frac{\frac{\pi}{2}-1}{\omega_3} - t} \quad (10)$$

For $t_1 < t < t_2$:

$$A_2(t) = \frac{D(t)}{\frac{\frac{\pi}{2}}{\omega_1} + \frac{\pi+1+\cos(\omega_2(t-T_1))}{\omega_2} + \frac{\frac{\pi}{2}-1}{\omega_3} - t} \quad (11)$$

For $t_2 < t < t_3$:

$$A_3(t) = \frac{D(t)}{\frac{\frac{\pi}{2}}{\omega_1} + \frac{\pi}{\omega_2} + \frac{\frac{\pi}{2}-\cos(\omega_3(t-T_1-T_2))}{\omega_3} - t} \quad (12)$$

C. Frequency and amplitude modulation

The velocity profile is modulated in amplitude and frequency by simply changing both A and ω parameters, respectively. The idea is to explicitly change these parameters

according to the oscillator current state, as follows:

$$A = \frac{A_1}{(1 + e^{b(m-A)})(1 + e^{bn})} + \frac{A_2}{1 + e^{-b(m-A)}} \quad (13)$$

$$+ \frac{A_3}{(1 + e^{b(m-A)})(1 + e^{-bn})}$$

where A_1 , A_2 and A_3 are as defined in eq.(10-12).

A alternates between three different values, A_1 , A_2 and A_3 , depending on the current values of the m and n variables. The alternation speed between these values is controlled by b . The same procedure is used for the ω parameters.

D. Dealing with obstacles

In this work, we specifically deal with the disturbance of obstacles along the path. These oblige the robot to change its path, and therefore it is required to compensate for the corresponding change of timing.

Obstacle presence is indicated by a potential function, $U(\phi_h)$ (see [11], [14] for details): if $U(\phi_h)$ has negative values, the repulsion from obstacles contribution is weak for the current heading direction value; if it has positive values, the current heading direction ϕ_h , is on a repulsion zone of sufficient strength and the robot must avoid the obstacle. In this last situation the velocity should decrease by reducing A . This is achieved by modifying eq.(14) as follows:

$$A = (\text{eq.14}) \left(1 - \frac{TF - d}{1 + e^{-b(U(\phi)-1/b)}} \right) \quad (14)$$

where TF is the maximum range of the robot's sensors and d the minimum distance to an obstacle measured by any of the sensors.

This set of equations constitutes the lower level responsible for setting the robot velocity at each time step. It receives from higher levels and at the right timing, sets of parameters that specify and modulate in a simple and straightforward manner the generated trajectories.

In summary, the set of parameters sent by the second layer, is composed by:

- 1) μ , switches on/off the rhythmic output. If $\mu > 0$ it also encodes the amplitude of rhythmic activity, $A = \sqrt{\mu}$;
- 2) O_m , modulates the oscillations' offsets, i.e. the goal for the discrete movement;
- 3) T_1 , T_2 , T_3 , specifies the velocity profile's shape.

The parameters α and b are set a priori.

VI. BEHAVIOR SWITCHING

In this work, we consider that there should exist three different behaviors: *stop*; *execution*; and *rescue*. Further, the switch between these behaviors should be easily and autonomously elicited, according to sensory information, such that action itself is elicited by perception.

A competitive dynamical system implements this switching mechanism.

A. Neural Competitive Dynamics

Each of the possible behaviors is represented by a "neural" variable $u_i \in [-1, 1]$ ($i = \text{stop}, \text{execution}, \text{rescue}$). A competitive dynamics is formulated for these variables as follows:

$$\alpha_\mu \dot{u}_i = \beta_i u_i - |\beta_i| u_i^3 - \nu \sum_{a \neq i} u_a^2 u_i + g_{wn} \quad (15)$$

where neurons u_i can go "on" (=1) or "off" (=0). The neuron u_i , with the largest competitive advantage, $\beta_i > 0$, is likely to win the competition, although for sufficiently small differences between the different β_i values multiple outcomes are possible, so that the system is effectively multistable.

Parameter ν is a competitive term which destabilizes any attractors in which more than one neuron is "on". α_μ defines the time scale of the dynamics.

We vary β_i parameters by : $\beta_i = 1.5 + 2b_i$, where b_i are "quasi - boolean" variables, varying between 0 and 1 (with a tendency to have values either close to 0 or close to 1). A sequence of neural switches and hence behavior switching, is generated by translating sensory conditions and logical constraints into values for these parameters ([15], [3], [1], [12] for examples).

Neuron *stop* is "on" (=1) when one of the following is true: (1) time, t , is below the initial time, t_{init} , set by user; (2) target has been reached.

Neuron *execution* is "on" (=1) when: (1) $t > t_{\text{init}}$; **and** (2) target is reachable; **and** (3) target has not been reached. A target is considered reachable when it is possible to reach the target in the remaining time.

Similarly, neuron *rescue* is "on" (=1) when: (1) $t > t_{\text{init}}$; **and** (2) target is not reachable.

B. Parameter modulation

Different values of triplets of neurons ($u_{\text{stop}}, u_{\text{execution}}, u_{\text{rescue}}$) lead to different behaviors, namely: no movement, timed movement and constant movement. Each triplet must then be mapped onto different values for the set of parameters. This is achieved as follows.

1) *Offset* O_m : The offset is set as follows

$$O_m = |u_{\text{stop}}| O_s + |u_{\text{execution}}| O_e + |u_{\text{rescue}}| O_r, \quad (16)$$

where O_s, O_e, O_r are respectively the offsets for behaviors *stop*, *execution* and *rescue*. The values are set according to desired behaviors, $O_s = 0$, $O_e = A$ and $O_r = 0.1$.

2) *Oscillatory Activity*: Qualitatively, by modifying on the fly the μ parameter, the system switches between a stable fixed point at $m = O_m$ (for $\mu < 0$) and a purely rhythmic movement (for $\mu > 0$). Hence, the μ parameter controls whether or not there are oscillations and thus, timed movement.

For $\mu > 0$ this parameter also encodes the amplitude of rhythmic activity. This parameter depends on the neural competitive dynamics as follows:

$$\mu = -(|u_{\text{stop}}| + |u_{\text{rescue}}|) \frac{A^2}{2} + |u_{\text{execution}}| A^2 \quad (17)$$

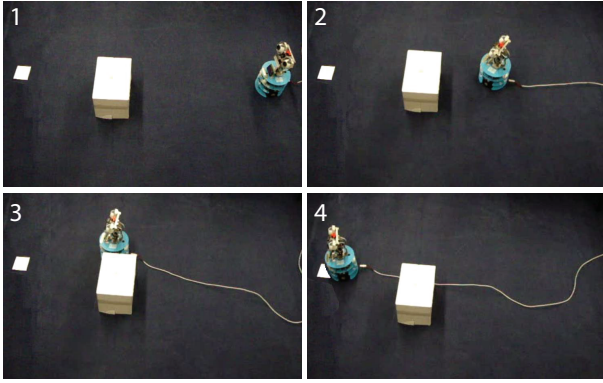


Fig. 2. Robot motion when the robot is moving to a 1.9 m target but during its path is faced with an obstacle.

This means that the timed movement is performed when $u_{\text{execution}}$ is “on”, by making $\mu > 0$, and that the velocity is constant when u_{stop} or u_{rescue} are ”on“, by making $\mu < 0$.

VII. EXPERIMENTAL RESULTS

In this section we describe some experiments done in a real mobile robot: *DRK8000*. Initially the robot is stopped and it must reach a target in three different world configurations within a specified time. The robot has no previous knowledge of any of the obstacles in the world, or of any other disturbance that may occur. Targets are directly represented through coordinates.

At each sensorial cycle, sensory information is acquired, dynamic equations are calculated and integrated using an Euler method with time step of 9 ms. The maximum robot velocity is 0.15 m/s.

A. Experiment 1

In this experiment the robot is faced with an obstacle during its path towards a target positioned at 1.9m away (fig. 2). The obstacle is at 1 m from the robot initial position. During the first $t_{\text{init}} = 3\text{s}$, the robot only turns towards the target but no timed forward movement is generated. Despite this disturbance, we expect that the robot will avoid the obstacle and reach the target without exceeding the specified movement time, 30 s.

The velocity profile of this experiment is presented on fig. 3. At $t = 12$ s, when robot is close to the obstacle, the amplitude of oscillator is decreased according to eq. (14), consequently reducing the robot’s velocity. This enforces a safe circumnavigation of the obstacle. After steering around the obstacle ($t = 15$ s), the amplitude is raised in order to compensate for the provoked delay.

Fig.4 shows the resulting path in phase-space. The oscillator radius A and offset are reduced during obstacle circumnavigation but then increase to higher values in order to reach the target within the specified movement time.

In fig. 5 we can observe the path taken by the robot throughout the experiment. Markers depict robot position at

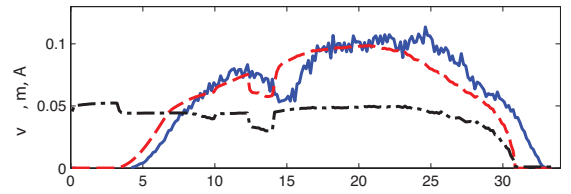


Fig. 3. Robot real velocity (blue continuous line), the timing velocity (red dashed line) and the amplitude of the oscillator (black mixed line).

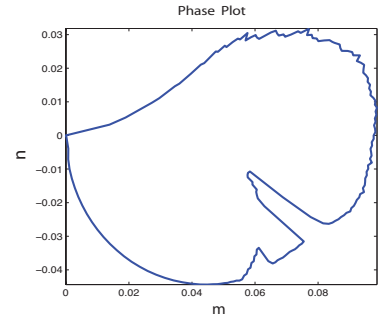


Fig. 4. Trajectories in the phase plane space for simulation depicted in 2.

instants of time, thus giving an indication of the robot velocity.

B. Experiment 2

The purpose of this experiment illustrated in fig.6, is to show that the robot compensates not only disturbances when faced with obstacles, but also compensates the displacement of the target. Initially, the robot tries to reach within the specified movement time (30 s), a target positioned at 1.9m. However, at $t = 19\text{s}$ (panel 2 in fig.6), the target is displaced to a distance of 1.4 m relatively the robot’s initial position.

Fig. 7 (bottom), presents the distance to the target, initially at 1.9 m. At $t = 19$ s it decreases due to the displacement of the target to a closer position.

In fig.7 (top) it is noticeable the simultaneous adaptation of the oscillator solution and the real velocity of the robot (blue continuous line) at the moment of displacement.

Observing table VII-B we can see that independently of the complexity of the world and the disturbances that the robot faces (obeying the robot’s physical restrictions), the robot performs its movement task within the specified movement time.

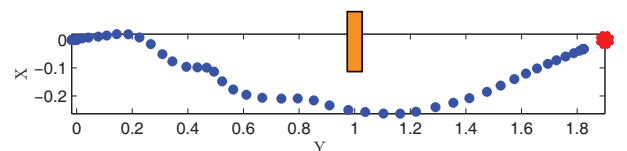


Fig. 5. The robot’s path as recorded by the dead-reckoning for the situation depicted in fig. 2. The red circle indicates the target location and the rectangle indicates the obstacle position in the world.

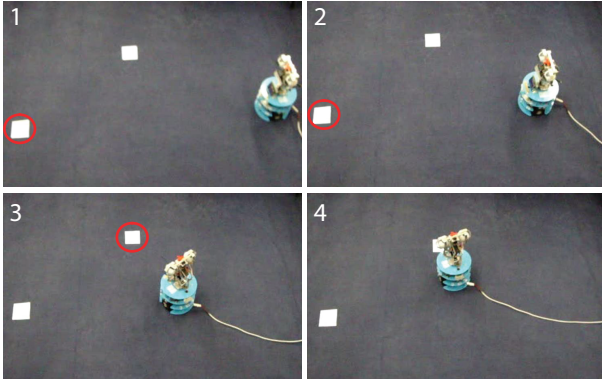


Fig. 6. Robot motion when the robot is moving to a target and suddenly at $t = 19$ s the target is displaced (panel B)

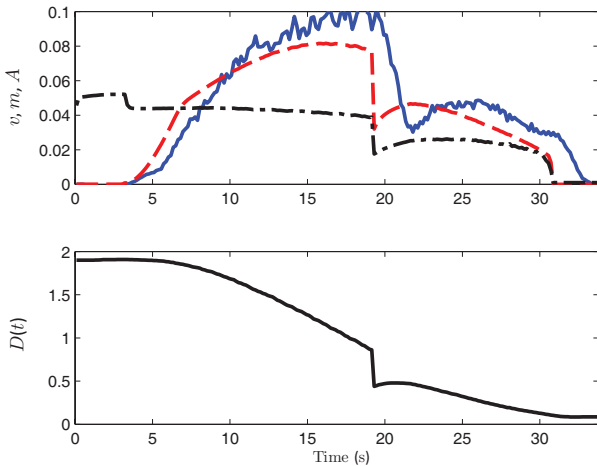


Fig. 7. Top: robot real velocity (blue continue line), the timing velocity (red dashed line) and the amplitude of the oscillator (black mixed line). Bottom: distance between the target and the robot along all trajectory.

VIII. CONCLUSION

In this paper we addressed the problem of generating timed trajectories for autonomous vehicles with noisy, low-level sensory information, which must reach a target within a stipulated time independently of the complexity of the environment and distance to the target, being able to detect and circumnavigate any obstacle and moreover compensate any delay or rush.

In this work we presented several new capabilities of the system relatively to previous works: 1) it initiates and terminates the movement task through the exploitation of the oscillator's Hopf bifurcation; 2) velocity profile adaptation,

TABLE I
AVERAGE PERFORMED MOVEMENT TIME IN DIFFERENT WORLD CONFIGURATIONS

Experiments	Time (s)	MT (s)	Initial Distance (m)
Experiment 1	28	30	1.60
Experiment 2	28.4	30	1.90
Experiment 3	28.6	30	1.90

allowing to specify the duration of specific parts, and 3) respective temporal adaptation; 4) new mechanism to adapt the velocity in order to safely overcome obstacles.

We successfully demonstrated the reliability of the proposed system through three experiments in a *DRK8000* robot, even when depending on low-level sensory information.

Future work will address how to extend the described model to achieve more complex behavior and how to integrate this approach with the dynamical system approach to generate formation control and how to incorporate the ability of using learning with neuronal layers.

ACKNOWLEDGMENT

Work supported by the Portuguese Science Foundation (grant PTDC/EEA-CRO/100655/2008).

REFERENCES

- [1] Cristina Santos. Generating timed trajectories for an autonomous vehicle: a non-linear dynamical systems approach. in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2004.
- [2] J. Lipinski, Y. Sandamirskaya, and G. Schöner. Swing it to the left, swing it to the right: Enacting flexible spatial language using a neurodynamic framework. *Cognitive Neurodynamics, special issue on "Language Dynamics"*, 3(4), 2009.
- [3] G. Schöner and Cristina Santos. Control of movement time and sequential action through attractor dynamics: A simulation study demonstrating object interception and coordination. in *Proc. of the 9th Int. Symposium on Intelligent Robotic Systems (SIRS)*, 2001.
- [4] S. Grillner. Locomotion in vertebrates: central mechanisms and reflex interaction. *Physiological Reviews*, 55:247–304, 1975.
- [5] M. MacKay-Lyons. Central pattern generation of locomotion: a review of the evidence. *Phys Ther*, 82(1):69–83, January 2002.
- [6] S. Grillner, P. Wallna, K. Saitoha, A. Kozlova, and B. Robertson. Neural bases of goal-directed locomotion in vertebrates: An overview. *Brain Research Reviews*, 57(1):2–12, January 2008.
- [7] S. Grillner, J. Hellgren, A. Ménard, K. Saitoh, and M. A. Wikström. Mechanisms for selection of basic motor programs—roles for the striatum and pallidum. *Trends Neurosci*, 28(7):364–370, July 2005.
- [8] Sarah Degallier, Cristina Santos, Ludovic Righetti, and Auke Ijspeert. Movement generation using dynamical systems: a humanoid robot performing a drumming task. In *IEEE-RAS International Conference on Humanoid Robots*, 2006.
- [9] Luiz Castro, Cristina Santos, Miguel Oliveira, and Auke Ijspeert. Postural control on a quadruped robot using lateral tilt: A dynamical system approach. In *EUROS*, volume 44 of *Springer Tracts in Advanced Robotics*, pages 205–214. Springer, 2008.
- [10] S. Grillner, Ö. Ekeberg, A. El Manira, A. Lansner, D. Parker, J. Tegner, and P. Wallen. Intrinsic function of a neuronal network – a vertebrate central pattern generator. *Brain Research Reviews*, 1998.
- [11] Pierre Mallet Estela Bicho and Gregor Schöner. Target representation on an autonomous vehicle with low-level sensors. *The International Journal of Robotics Research*, (210):424–447, 2000.
- [12] Cristina P. Santos and Manuel Ferreira. Two vision-guided vehicles: temporal coordination using nonlinear dynamical systems. In *ICRA*, pages 14–19, 2007.
- [13] Gregor Schöner. A dynamic theory of coordination of discrete movement. in *Biological Cybernetics*, 63:257270, 1990.
- [14] 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(4):253–267, 1992.
- [15] Axel Steinhage and Gregor Schöner. Dynamical systems for the behavioral organization of autonomous robot navigation. In *McKee G T Schenker PS, editor, Sensor Fusion and Decentralized Control in Robotic Systems: Proceedings of Spie-Intelligent Systems Manufacturers, Boston*, page 169180, 1998.