

# Two Vision-guided vehicles: temporal coordination using nonlinear dynamical systems

Cristina P. Santos and Manuel Ferreira

**Abstract**—This article addresses the problem of generating timed trajectories and temporally coordinated movements for two wheeled vehicles, when relatively low-level, noisy sensorial information is used to steer action. The generated trajectories have controlled and stable timing (limit cycle type solutions). Incoupling of sensory information enables sensor driven termination of movement. We build on a previously proposed solution in which timed trajectories and sequences of movements were generated as attractor solutions of dynamic systems. We present a novel system composed of two coupled dynamical architectures that temporally coordinate the solutions of these dynamical systems. The coupled dynamics enable synchronization of the different components providing an independence relatively to the specification of their individual parameters.

We apply this architecture to generate temporally coordinated trajectories for two vision-guided mobile robots in a non-structured simulated environment, whose goal is to reach a target within a certain time independently of the environment configuration or the distance to the target. The results illustrate the robustness of the proposed decision-making mechanism and show that the two vehicles are temporal coordinated: if a robot movement is affected by the environment configuration such that it will take longer to reach the target, the control level coordinates the two robots such that they terminate approximately simultaneously.

## I. INTRODUCTION

Trajectory planning has been extensively studied over the last few years, ranging from the addition of the time dimension to the robot's configuration space [1], visibility graph [2], cell decomposition [3] or neural networks [4]. There are several results for time-optimal trajectory planning [5].

Despite the efficient planning algorithms that have been developed and the advances in the control domain which validated dynamic, robust and adaptive control techniques, the path planning problem in autonomous robotics remains separated in theory from perception and control. This separation implies that space and time constraints on robot motion must be known before hand with the high degree of precision typically required for non-autonomous robot operation. In order to develop autonomous robot systems capable of operating in changing and uncertain environments it is required a tight coupling of planning, sensing and execution.

However, timing is more difficult to control when it must be compatible with the requirement of continuous coupling to sensory information. Some approaches have addressed this issue [6], but timing was not fully explored.

Industrial Electronics Department, University of Minho, Guimaraes, Portugal  
cristina@dei.uminho.pt, mjf@dei.uminho.pt

In this article, we propose an approach fully formulated in terms of nonlinear dynamical systems which lead to a flexible timed behavior stably adapted to changing online sensory information. Dynamical systems have various desirable properties which makes them interesting and powerful for trajectory generation. See [7], [8], [9], [10], [11], [12] for related work. First, the structural robustness of the solutions implies intrinsic robustness against small perturbations and noise and the possibility to fuse new inputs into the system without completely destroying its properties. Second, the low computation cost is well-suited for real time. Other properties are the smooth online modulation of the trajectories through changes in the parameters of the dynamical systems; the possibility to synchronize with external signals and to add sensory feedback pathways. The dynamics of the system globally encode a task (i.e. the whole attractor landscape) with the goal state as the point attractor. This is a "always online" property, i.e., once a task is encoded into a dynamical system (e.g. learning) it will be always active, and no discrete trials are needed. Once properly designed, the dynamical system can be robust enough against perturbations and able to smoothly recover from perturbations by means of coupling terms in the dynamics. Another particularity is that these systems produce coordinated multidimensional rhythms of motor activity, under the control of simple input signals. Such systems are deemed to strongly reduce the dimensionality of the control problem.

We build on previous work [13], [11], [14], where we proposed a dynamical system architecture that generated timed trajectories, including rhythmic and discrete movement, movement sequences and temporally coordinated movements. The model consists of a dynamical system composed of stable fixed points and a stable limit cycle (an Hopf oscillator). Trajectories are generated through the sequencing of these primitives, in which the limit cycle is activated over limited time intervals. This sequencing is controlled by a "neural" competitive dynamics. By controlling the timing of a limit cycle, the system performs well tasks with complex timing constraints. The online linkage to noisy sensorial information, was achieved through the coupling of these dynamical systems to time-varying sensory information [14], [13]. In [13], this architecture was implemented in a real vehicle and integrated with other dynamical architectures which do not explicitly parameterize timing requirements. In [11], we have generated temporally coordinated movements among two PUMA arms by coupling two such dynamical systems.

In this work, coordination is modeled through mutual

coupling of such dynamical systems. This coupling enables to achieve temporal coordination and synchronization of the different systems, providing an independency relatively to the specification of their individual parameters. Specifically, we address the following questions: Can the temporal coordination among different degrees-of-freedom (dofs) be applied to the robotics domain such that a tendency to synchronize among two vehicles is achieved? Can the applied dynamical systems approach provide a theoretically based way of tuning the movement parameters such that it is possible to account for relationships among these?

These questions are positively answered and shown in exemplary simulations in which two low-level vehicles must navigate in a simulated non-structured environment while being capable of reaching a target in an approximately constant time. For each robot, target position is internally acquired by a visual system mounted over the robot and robot velocity is controlled such that the vehicle has a fixed time to reach the target while continuously avoiding sensed obstacles in its path. The two robot movements are coupled in time such that if the two movements onsets are not perfectly simultaneous or if their time trajectories are evolving differently (one is going faster/slower than the other), leading to different movement times (time it takes to reach the target), this coupling coordinates the two movements such that they terminate approximately simultaneously.

Interesting properties of the system include: 1) the possibility to include feedback loops in order to do online trajectory modulation and take external perturbations into account, such that the environmental changes adjust the dynamics of trajectory generation; 2) online modulation of the trajectories with respect to the amplitude, frequency and the midpoint of the rhythmic patterns (discrete movements goal), while keeping the general features of the original movements, and 3) the coordination and synchronization among the robots, achieved through the coupling among the dynamics of each robot, that provides for a smooth and an adaptive behaviour of the complete system in face perturbations in the sensed environment. This type of control scheme has a wide range of applications in multi-dimensional control problems.

It is our belief that planning in terms of autonomous non-linear attractor landscapes promises more general movement behaviours than traditional approaches using time-indexed trajectory planning. Further, by removing the explicit time dependency one can avoid complicated 'clocking' and 'reset clock' mechanisms.

In the rest of the article, we will first present the dynamical systems approach and discuss the intrinsic properties. In the next section, we describe the two designed dynamical systems that act at different levels: one acting out at the level of heading direction and the other controlling the robot's velocities. In this section, it is also described the vision system. We then described the simulation environment for the application (Section 3). In section 4, we present two simulations and our results and discuss the properties of the system. We conclude by presenting the conclusions and

presenting future directions for the work (section 5).

## II. THE DYNAMICAL SYSTEMS TRAJECTORY GENERATOR

In this section we describe the dynamical system architecture that generates timed trajectories. First, we describe the dynamical systems composed of stable fixed points and a stable limit cycle (an Hopf oscillator). Second, the "neural" dynamics that control the sequential activation of these dynamic primitives is described. The solutions of these dynamical systems are temporally coordinated through the coupling of these architectures. Finally, we discuss some relevant properties of the overall system that enables to achieve generation and temporal coordination of complex movements.

### A. Fixed points and limit cycle solutions generator

A dynamical system architecture generates timed trajectories for a relevant robotic variable as described in [13]. Specifically, timed trajectories are modeled as time courses of behavioral variables  $(m, n)$  which are stable solutions of dynamical systems. Although only the variable,  $m$ , will be used to set the robotic variable, a second auxiliary variable,  $n$ , is needed to enable the system to undergo periodic motion.

We set two spatially fixed coordinates systems each centered on the initial robot position: one for the  $x$  and the other for the  $y$  spatial coordinates of robot movement. A dynamical system which generates both stable oscillations (limit cycle solutions) and two stationary states [11], [13], is defined for each of these fixed coordinate systems as follows:

$$\begin{pmatrix} \dot{m}_i \\ \dot{n}_i \end{pmatrix} = 5 |u_{\text{init},i}| \begin{pmatrix} m_i \\ n_i \end{pmatrix} + |u_{\text{hopf},i}| f_{\text{hopf},i} \\ + 5 |u_{\text{final},i}| \begin{pmatrix} m_i - A_{\text{ic}} \\ n_i \end{pmatrix} + \text{gwn}, \quad (1)$$

where the index  $i = x, y$  refers to dynamics of  $x$  and  $y$  spatial coordinates of robot movement. A neural dynamics controls the switching between the three regimes through three "neurons"  $u_{j,i}$  ( $j = \text{init}, \text{hopf}, \text{final}$ ). The "init" and "final" contributions generate stable stationary solutions at  $m_i = 0$  for "init" and  $A_{\text{ic}}$  for "final" with  $n_i = 0$  for both. These states are characterized by a time scale of  $\tau = 1/5 = 0.2$ .

Herein, an approach is defined to achieve temporal coordination among the two robots, by coupling these two architectures in a way that generates phase-locking (synchronization) in the oscillation regime. This was achieved by modifying the "Hopf" contribution that generates the limit cycle solution as follows:

$$\begin{aligned} f_{\text{hopf},i} &= \begin{pmatrix} \alpha & -\omega \\ \omega & \alpha \end{pmatrix} \begin{pmatrix} (m_i - \frac{A_{\text{ic}}}{2}) \\ n_i \end{pmatrix} \\ &- \gamma_i \left( \left( m_i - \frac{A_{\text{ic}}}{2} \right)^2 + n_i^2 \right) \begin{pmatrix} (m_i - \frac{A_{\text{ic}}}{2}) \\ n_i \end{pmatrix} \\ &+ c |u_{\text{hopf},j}| \begin{pmatrix} \cos \theta_{ij} & -\sin \theta_{ij} \\ \sin \theta_{ij} & \cos \theta_{ij} \end{pmatrix} \begin{pmatrix} m_j \\ n_j \end{pmatrix} \quad (2) \end{aligned}$$

where index  $j$  refers to index  $i$  time courses of the coupled dynamical system (the other robot),  $\gamma_i = \frac{4\alpha}{A_{\text{ic}}^2}$  defines

amplitude of Hopf contribution and  $\theta_{ij}$  is the desired relative phase among oscillators  $i$  and  $j$  ( $-\theta_{ij}$  among oscillators  $j$  and  $i$ ). For instance,  $(m_x, n_x)$  of robot 1 is coupled with  $(m_x, n_x)$  of robot 2. The coupling term is multiplied with the neuronal activation of the other system's Hopf state so that coupling is effective only when both components are in the oscillation regime. Because we want both coupled dynamical systems to be in-phase we set  $\theta_{ij} = 0$  degrees.

This ‘‘Hopf’’ contribution provides a stable periodic solution (limit cycle attractor) with cycle time  $T = \frac{2\pi}{\omega} = 20\text{s}$ . We use it because it can be completely solved analytically, providing complete control over its stable states. This analytical specification is an innovative aspect of our work. Relaxation to the limit cycle solution occurs at a time scale of  $1/(2\alpha) = 0.2$  time units.

The dynamics of (1) are augmented by a Gaussian white noise term, gwn, that guarantees escape from unstable states and assures robustness to the system.

### B. Neural dynamics

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

$$\alpha_u \dot{u}_{j,i} = \mu_{j,i} u_{j,i} - |\mu_{j,i}| u_{j,i}^3 - 2.1 \sum_{a \neq j} u_{a,i}^2 u_{j,i} + \text{gwn}. \quad (3)$$

where ‘‘neurons’’ can go ‘‘on’’ ( $=1$ ) or ‘‘off’’ ( $=0$ ). This dynamics enforces competition among task constraints depending on the neural ‘‘competitive advantages’’ parameters,  $\mu_i$ . 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  $\mu_i$  values multiple outcomes are possible (the system is multistable).

In order to control switching, the  $\mu_i$  parameters are explicitly designed as functions of user commands, sensory events, or internal states and control the sequential activation of the different neurons (see [11], for a general framework for sequence generation based on these ideas). We vary the  $\mu$ -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). Hence, we assure that one neuron is always ‘‘on’’. Herein, the time,  $t$ , and target location, fully control the neural dynamics through the quasi-boolean parameters. A sequence of neural switches is generated by translating sensory conditions and logical constraints into values for these parameters (see [13] for a description).

The time scale of the neuronal dynamics is set to a relaxation time of  $\tau_u = 0.02$ , ten times faster than the relaxation time of the  $(m, n)$  dynamical variables. By using different time scales one can design the several dynamical systems separately [13].

Temporally discrete movement is autonomously generated through a sequence of neural switches such that an oscillatory state exists during an appropriate time interval of about a half-cycle. This approximately half-cycle is movement time (MT), here  $MT = 10\text{s}$ .

### C. Intrinsic properties of the overall dynamics

The fact that timed movement is generated from attractor solutions of nonlinear dynamical systems leads to a number of desirable properties for trajectory generation. The system is able to make decisions such that it flexibly responds to the demands of any given situation while keeping timing stable. Intrinsic stability properties are inherent to the Hopf oscillator, which has a structurally stable limit cycle. Thus, the generated trajectories are robust to the presence of noise and stable to perturbations. This property is specially useful for adding feedback pathways because sensory information is forgotten as soon as it disappears from the environment. This structural robustness of solutions further guarantees the stability and controllability of the overall system if the time scale separation principle is obeyed. These intrinsic properties, including bifurcation and hysteresis, enable planning decisions to be made and carried out in a flexible, yet stable way, even if unreliable sensory information is used to steer action. These properties are explained in more detail in [13].

An advantage of this approach is 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. Smooth trajectory online modulation of the trajectories with respect to the goal, amplitude and frequency is now possible, while keeping the general features of the original movements. Trajectories are thus modulated according to the environmental changes, such that action is steered by online modulation of the parameters. A simple modulation of the parameters can generate an infinite variation of stable trajectories.

Moreover, we showed that it was easy to couple two dynamical systems to generate coordinated multidimensional trajectories. The extension to a more enlarged number of dynamical systems is feasible and brings no added complications. The coordination and synchronization among the generated trajectories, achieved through the coupling of their dynamical systems, provides for a smooth and an adaptive behavior of the complete system in face of perturbations in the sensed environment. The coupling of nonlinear oscillators offers multiple interesting properties which enable smooth integration of their parameters and makes them interesting and powerful for trajectory generation.

In the next section, we show the application of this dynamical architecture to the generation of timed trajectories for two vision-guided vehicles.

## III. TIMED TRAJECTORIES GENERATION FOR TWO VISION-GUIDED VEHICLES

The dynamical systems formulated in order to solve this robotic problem are divided onto two integrated architectures which act out at different levels. The dynamics of heading direction act out at the level of the turning rate. The dynamics of driving speed express time constraints.

### A. Attractor dynamics of heading direction

The robot action of turning is generated by letting the robot's heading direction,  $\phi_h$ , measured relative to some allocentric reference frame, vary by making  $\phi_h$  the behavioral variable of a dynamical system (for a full discussion see [7]). This behavioral variable is governed by a nonlinear vector field in which task constraints contribute independently by modeling desired behaviors (*target acquisition*) as attractors and undesired behaviors (*obstacle avoidance*) as repellers of the overall behavioral dynamics. Integration of the *target acquisition*,  $F_{\text{tar}}(\phi_h)$  and *obstacle avoidance*,  $F_{\text{obs}}(\phi_h)$  contributions is achieved by adding each of them to the vector field that governs heading direction dynamics

$$\frac{d\phi_h}{dt} = F_{\text{obs}}(\phi_h) + F_{\text{tar}}(\phi_h) + F_{\text{stoch}}(\phi_h). \quad (4)$$

We add a stochastic component force,  $F_{\text{stoch}}$ , 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 [15], [13].

### B. Coupling to sensorial information

Ball position is acquired by simulating a camera mounted on the top of the robot and facing in the direction of the driving speed. The goal is to robustly detect a red ball in an unstructured, complex environment.

The most common algorithms for visual object tracking in robot applications are typically based on the detection of a particular cue, most commonly edges, color and texture [16], [17].

In our application, we have to deal with the following main computer-vision problems: (1) a clutter environment, including non-uniform light conditions and different objects with the same color pattern (distractors); (2) irregular object motion due to perspective-induced motion irregularities; (3) image noise and (4) a real-time performance application with high processing time. Some of these problems may not be a problem in a simulated environment, but they will be as soon as we move on to a real application.

Although conventional single-cue algorithms fail to catch variations like changes of orientation and in shape, if flexibility and/or simplicity, speed and robustness are required, as in our case, they are a good option. Specifically, we have chosen a color based real-time tracker, Continuously Adaptive Mean Shift (CAMSHIFT) algorithm [18], that handles the described computer-vision application problems during its operation.

This algorithm tracks the  $x'$ ,  $y'$  image coordinates and area of the color blob representing the white and black ball. We have assumed that ball size is known and can be measured in the image. A perspective projection model transforms the  $x'$ ,  $y'$  image coordinates onto the  $x$  world coordinates, as follows:

$$x' = \frac{a_{11}x + a_{12}y + a_{13}}{a_{31}x + a_{32}y + 1} \quad (5)$$



(a) Presence of a distractor element

(b) Variations in lighting conditions.

Fig. 1. Application of the CAMSHIFT algorithm to a real, clutter environment, where some computer vision problems in visual object tracking are addressed.

Herein, we illustrate two real applications of this algorithm to a real, clutter environment. Fig. 1(a) shows the result of this algorithm in the presence of a distractor element. In Fig. 1(b) the incident illumination as been increased by a factor of 1.5. In both situations, the algorithm is able to track the ball.

To simulate sensor noise (which can be substantial if such optical measures are extracted from image sequences), we added either white or colored noise to the image coordinates. Here we show simulations that used colored noise,  $\zeta$ , generated from

$$\dot{\zeta} = -\frac{1}{\tau_{\text{corr}}} \zeta + \sqrt{Q} \text{gwn} \quad (6)$$

where gwn is gaussian white noise with zero mean and unit variance, so that  $Q = 5$  is the effective variance. The correlation time,  $\tau_{\text{corr}}$ , was chosen as 0.2 sec.

### C. Velocity

Robot velocity is controlled by a dynamics similar to that described in [15], such that the planning variable is in or near a resulting attractor of the dynamical system most of the time. This dynamics assures that velocity depends whether or not obstacles are detected for the current heading direction value. In case an obstacle has been detected, velocity is set as  $V_{\text{obs}}$ , which is computed as a function of the current distance to the obstacle [15]. In case no obstacle has been detected, velocity is set as  $V_{\text{timing}}$ :

$$V_{\text{timing}} = \sqrt{\dot{m}_x + \dot{m}_y}, \quad (7)$$

where  $m_x, m_y$  are given by eq. 1. The path velocity,  $V_{\text{timing}}$ , of the vehicle is thus controlled through the described dynamical system architecture that generates timed trajectories.

In the following, we briefly explain the dynamic architecture behavior of each robot. At  $t = 0$  s the robot is resting at its initial fixed position,  $x_{\text{Rinit}}, y_{\text{Rinit}}$ . The robot rotates in the spot in order to orient towards or look for the target direction. At time  $t_{\text{init}}$ , timed forward movement is initiated. The periodic motion's amplitude,  $A_{\text{mc}}$ , is updated during periodic movement each time step as follows

$$A_{\text{mc}} = (x_{\text{target}} - x_{\text{Rinit}}) - ((x_{\text{R}} - x_{\text{Rinit}}) - m_x), \quad (8)$$

where  $x_{\text{target}}$  is  $x$  target position,  $x_{\text{R}}$  is  $x$  robot position and  $m_x$  is the dynamical variable.

The periodic solution is deactivated again when the vehicle position comes into the vicinity of  $A_{mc}$ , and the final postural state (which equals  $A_{mc}$ ) is turned on instead. The same behavior applies for the dynamical systems defined for the  $y$  spatial coordinate.

#### IV. EXPERIMENTAL RESULTS

The dynamic architecture was simulated in Matlab/Simulink (product of the MATHWORKS company) and in webots[19]. This simulator is based on ODE, an open source physics engine for simulating 3D rigid body dynamics. Each vehicle has seven infrared sensors equidistantly mounted on a ring on the robot's periphery, used to measure distance to surfaces at the height of the ring. The model of the robots are as close to the real robots as the simulation enable us to be. Thus, we simulate the exact kinematic equations, mass distributions, infra-red sensor and the visual system. The dynamics of heading direction, timing, competitive neural, path velocity and dead-reckoning equations are numerically integrated using the Euler method with fixed time step. The cycle time is 70 ms and MT is 10s.

The initial heading direction is 90 degrees. Forward movement initiation is triggered by an initial time set by the user,  $t_{init} = 3s$ , and not from sensed sensorial information. Sensed obstacles do not block vision. In case the target is not currently in the limited viewing angle of the camera but has been previously seen, we algorithmically update the previous target location based on dead-reckoning information.

The rotation speeds of both wheels are computed from the angular velocity,  $w$ , and the path velocity,  $v$  of the robot. The former is obtained from the dynamics of heading direction. The later, as obtained from the velocity dynamics is specified either by obstacle avoidance contribution or by  $V_{timing}$  (eq. 7). By simple kinematics, these velocities are translated into the rotation speeds of both wheels and sent to the velocity servos of the two motors.

In order to verify if temporal coordination among the two robot movements is achieved we have performed several simulations. Herein, due to space constraints, we illustrate two exemplary simulations.

During its path towards the target, robot 2 is faced with an obstacle which it must circumnavigate. This obstacle does not interfere with the robot 1 movement towards the target. Fig. 2(a) illustrates the robot motions and time stamps of these trajectories. The ball is depicted by a light circle. Small crosses around ball position indicate ball position as acquired by the vision systems. The robots path are indicated by lines formed by crosses. The interval between two consecutive crosses indicates the robot's path velocity since the time acquisition interval is constant: the smaller the velocity the closer the points. When the obstacle is no longer detected for the current heading direction, at  $t = 9.1s$ , robot 2 is strongly accelerated in order to compensate for the object circumnavigation.

Robot velocities are depicted in Fig. 3.  $v$  represents forward velocity of the robot.  $v_{timing}$  and  $v_{obs}$  represent velocity imposed by the discussed dynamical architecture and

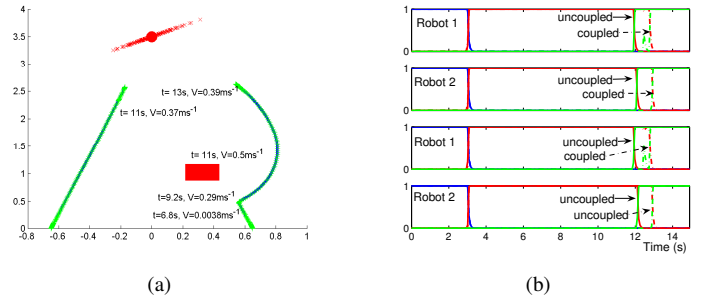


Fig. 2. A simulation run where the robot meets the red ball. a) Robots timed trajectories. b) Top and bottom panels illustrate the neural variables of  $x$  and  $y$  coordinate dynamical systems of both robots.

velocity imposed in case an obstacle is detected, respectively.

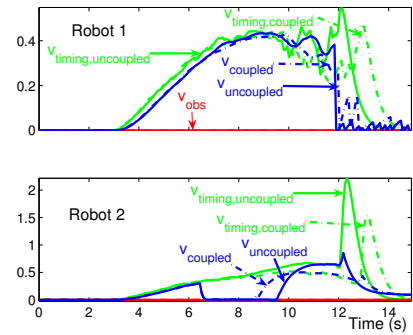


Fig. 3. Velocity variables for robot 1 and 2.

The proposed dynamic architecture without coupling ( $c = 0$  in eq. 2) is similar to work presented in [13], where results have shown that robot velocity is controlled such that the target is reached in an approximately constant time ( $MT = 10s$ ) independently of the environment configuration and of the distance to the target.

The introduction of a coupling of this form tends to synchronize movement in the two robots. Thus, when  $x$  and/or  $y$  movement of robot 2 is affected by the environment configuration such that its periodic motion amplitude is increased, robot 1 movement is coordinated through coupling such that movements of both robots terminate simultaneously. This results in delayed simultaneous switch, around  $t = 12.8s$ , among Hopf and final contributions for  $x$  and  $y$  dynamical systems of both robots (see Fig. 2(b)). Note that synchronization only exists when both dynamical systems exhibit periodic motion.

Coupling two such dynamical systems removes the need to compute exactly identical movement times for two robot movements that must be temporally coordinated. Even if there is a discrepancy in the movement time programmed by the parameter,  $\omega$ , of the Hopf dynamics (which corresponds to larger MTs due to complex environment configurations), coupling generates identical effective movement times.

One interesting aspect is that since the velocities applied to the robots are different depending if there is coupling or



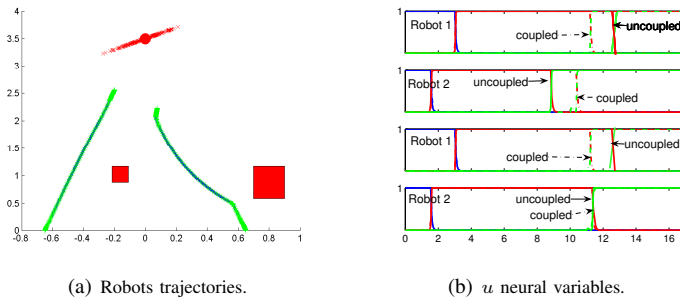


Fig. 4. In this simulation, movement on-sets are set differently for each robot. Object circumnavigation leads to different movement times for each robot as well.

not, this results in slightly different qualitative paths followed by the robot.

Fig. 4(a) illustrates the robot motions and time stamps of these trajectories towards the target, when both robots are faced with obstacles which they must circumnavigate. These circumnavigations lead to different movement times for both robots. Further, movements on-sets are set differently: robot 1 starts its movement at  $t_{init} = 3s$  and robot 2 starts its movement at  $t_{init} = 1.5s$ . The coupling coordinates the two movements such that they terminate approximately simultaneously (see fig. 4(b)).

## V. CONCLUSION/OUTLOOK

In this article, an attractor based dynamics autonomously generated temporally discrete and coordinated movements. The task was to temporally coordinate the timed movements of two low-level vehicles, which must navigate in a simulated non-structured environment while being capable of reaching a target within a certain time independently of the environment configuration. Movement termination was entirely sensor driven and autonomous sequence generation was stably adapted to changing unreliable simulated visual sensory information. We applied autonomous differential equations to formulate two integrated dynamical architectures which act out at the heading direction and driving speed levels of each robot. Each robot velocity is controlled by a dynamical systems architecture based on previous work [13], which generates timed trajectories. Temporal coordination of the two robots is enabled through the coupling among these architectures.

Results enable to positively answer to the two questions addressed in the introduction. The former asked if synchronization among two vehicles can be achieved when we apply temporal coordination among dofs. Results illustrate the dynamic architecture robustness and show that such a coupling tends to synchronize movement in the two robots, a tendency captured in terms of relative timing of robots movements. The later question asked if the applied approach provides a theoretically based way of tuning the movement parameters such that it is possible to account for relationships among these. Results show that the coupled dynamics enable synchronization of the robots providing an independence relatively to the specification of their individual movement

parameters, such as movement time, movement extent, etc. This synchronization reduces computational requirements for determining identical movement parameters across robots. From the view point of engineering applications, the inherent advantages are huge, since the control system is released from the task of recalculating the movement parameters of the different components.

## REFERENCES

- [1] M. Erdmann and T. Lozano-Perez, "On multiple moving objects," *Algorithmica*, vol. 2, pp. 477–521, 1987.
- [2] J. Reif and M. Sharir, "Motion planning in the presence of moving obstacles," in *Proceedings of the 25th IEEE Symposium on the Foundation of Computer Science*, Portland, OR (USA), October 1985, pp. 144–153.
- [3] K. Fujimura and H. Samet, "A hierarchical strategy for path planning among moving obstacles," *IEEE Transaction on Robotics and Automation*, vol. 5, no. 1, pp. 61–69, February 1989.
- [4] R. Gladius, A. Komoda, and S. Gielen, "Population coding in a neural net for trajectory formation," *Network: Computation in Neural Systems*, vol. 5, pp. 549–563, 28 July 1994.
- [5] T. Fraichard, "Trajectory planning in a dynamic workspace: A "state-time space" approach," *Advanced Robotics*, vol. 13, no. 1, pp. 75–94, 1999.
- [6] D. K. M. Bühler and Kindlmann, "Planning and control of a juggling robot," *International Journal of Robotics Research*, vol. 13, no. 2, pp. 101–118, 1994.
- [7] G. Schöner and M. Dose, "A dynamical systems approach to task-level system integration used to plan and control autonomous vehicle motion," *Robotics and Autonomous Systems*, vol. 10, pp. 253–267, 1992.
- [8] J. Tani, Y. Ito, and Y. Sugita, "Self-organization of distributedly represented multiple behavior schemata in a mirror system: reviews of robot experiments using rnnpb," *Neural Networks*, vol. 17, pp. 1273–1289, 2004. [Online]. Available: <http://www.bdc.brain.riken.go.jp/~tani/publications.htm>
- [9] S. Schaal, S. Kotosaka, and D. Sternad, "Nonlinear dynamical systems as movement primitives," in *International Conference on Humanoid Robotics*. Cambridge, MA, Sept 6-7, 2001. Springer, 2001, pp. 117–124.
- [10] Y. Fukuoka, H. Kimura, and A. Cohen, "Adaptive dynamic walking of a quadruped robot on irregular terrain based on biological concepts," *The International Journal of Robotics Research*, vol. 3–4, pp. 187–202, 2003.
- [11] G. Schöner and C. Santos, "Control of movement time and sequential action through attractor dynamics: A simulation study demonstrating object interception and coordination," in *9th Intelligent Symp. On Intelligent Robotic Systems*, Toulouse, France, 18-20, July 2001.
- [12] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Learning control policies for movement imitation and movement recognition," in *Neural Information Processing System (NIPS2001)*, 2001.
- [13] C. Santos, "Generating timed trajectories for an autonomous vehicle: A non-linear dynamical systems approach," in *IEEE Int. Conf. On Robotics and Automation*. New Orleans: IEEE, April 26-1 May 2004, pp. 3741–3746.
- [14] G. Schöner, "Dynamic theory of action-perception patterns: The time-before-contact-paradigm," *Human Mov. Science*, vol. 3, pp. 415–439, 1994.
- [15] E. Bicho, P. Mallet, and G. Schöner, "Target representation on an autonomous vehicle with low-level sensors," *The Int. Journal of Robotics Research*, vol. 19, no. 5, pp. 424–447, May 2000.
- [16] E. M. M. Pressigout, "Real-time planar structure tracking for visual servoing: A contour and texture approach." *Int. Conf. on Intelligent Robots and Systems, IROS'05*, August 2005.
- [17] G. Taylor, R. Nudds, and A. Thomas, "Flying and swimming animals cruise at a strouhal number tuned for high power efficiency," *Nature*, vol. 425, pp. 707–711, 2003.
- [18] G. Bradski, "Computer vision face tracking as a component of a perceptual user interface." Princeton: Applications of Computer Vision, October 1998, pp. 214–219.
- [19] O. Michel, "Webots: Professional mobile robot simulation," *International Journal of Advanced Robotic Systems*, vol. 1, no. 1, pp. 39–42, 2004.