2009 IEEE International Conference on Robotics and Automation
Kobe International Conference Center
Kobe, Japan, May 12-17, 2009

Temporal stabilization of discrete movement in variable environments: an attractor dynamics approach

Matthias Tuma, Ioannis Iossifidis and Gregor Schöner

Abstract— **The ability to generate discrete movement with distinct and stable time courses is important for interaction scenarios both between different robots and with human partners, for catching and interception tasks, and for timed action sequences. In dynamic environments, where trajectories are evolving on-line, this is not a trivial task. The dynamical systems approach to robotics provides a framework for robust incorporation of fluctuating sensor information, but control of movement time is usually restricted to rhythmic motion and realized through stable limit cycles. The present work uses a Hopf oscillator to produce discrete motion and formulates an on-line adaptation rule to stabilize total movement time against a wide range of disturbances. This is integrated into a dynamical systems framework for the sequencing of movement phases and for directional navigation, using 2D-planar motion as an example. The approach is demonstrated on a Khepera mobile unit in order to show its reliability even when depending on low-level sensor information.**

I. INTRODUCTION

Providing theoretical mechanisms for trajectory generation – for simple mobile units as well as robot arms with many degrees of freedom – remains an important task in robotics. This especially holds for autonomous agents expected to flexibly react to time-varying and partially unpredictable environments. Contrasting classical approaches that separate task planning, trajectory planning, and control, the behavior-based approach to robotics is aimed at linking action and perception at low levels of sensory information [1]. Its advantages include limited computational needs and some capacity to act in unknown surroundings. In contrast, temporal properties of movements are less easily controlled than in classical approaches. This is revealed when particular velocity profiles must be achieved, or when the total movement time is to be stabilized against perturbations to perform a movement "on time". Examples for tasks with external temporal boundary conditions include catching, hitting, or juggling of objects; sequential tasks, in which successive actions depend on the timely accomplishment of preceding ones; or coupling and coordination between several effectors or units. This is also true for interaction scenarios between robots and human operators: as humans exhibit strong temporal stabilization in their movements [2], they expect their robotic partners to behave likewise.

Within dynamical systems theory [3], stable limit cycles are known to produce oscillations in phase space with a fixed temporal structure. Such systems are related to both the dynamical systems approach to robotics [4][5], and to the concept of central pattern generators (CPGs, see [6] for a review).

The dynamical systems approach on the one hand uses on-line sensory information to dynamically change parameters of differential equations for a set of behavioral variables which define the robot's state. By integrating these time-variant dynamical systems and simultaneously making use of bifurcations, action sequences and complex trajectories can be produced [7]. In this manner, dynamical systems can be employed at the level of planning in addition to the level of control. Advantages include the possibility of mutual or external coupling; stability properties that may be analytically proven for a range of cases; the ability to incorporate sensory or feedback information on-line; the possibility to consider stochasticity; and a low computational footprint. In comparison to potential field methods, the mathematical grounding is stronger, and several problems like spurious minima and oscillations are avoided [8]. Also, the possibility of integrating multiple constraints and generating decisions through instabilities and multistability makes such systems much more flexible than nonlinear controllers.

CPGs on the other hand are neural circuits that generate rhythmic signals without depending on rhythmic input, and can mathematically be described as nonlinear dynamical systems with stable limit cycle (periodic) solutions. Originally found to be responsible for the generation of rhythmic motor acts in a variety of both invertebrate and vertebrate species, CPGs are increasingly used in robotics, especially to couple and control rhythmic movement of several effectors [6].

A. *Related work*

Neither in robotics nor in neuroscience has the timing of discrete motion been addressed with any depth. In particular, the stabilization of movement time against a broad range of perturbations remains, to the best of our knowledge, an open issue. More widely covered are the themes of learning of rhythmic patterns, combining dynamics for rhythmic and discrete motion, implementing juggling or catching tasks, as well as superposing movement primitives. Related work by Ijspeert et al. [9] decouples learned movement plans from temporal information to ensure correct pursuit of motion after disturbances, but depends on task-tailored stabilizing dynamics and does not attempt a temporal stabilization. A theoretical proposal by Righetti et al. [10] provides a frequency-learning mechanism for nonlinear oscillators in the presence of external signals, which has been applied to

This work was supported by the German BMBF grant DESIRE (FKZ 01 IME 01G)

The authors are with the Institut für Neuroinformatik, Ruhr-Universität Bochum, 44780 Bochum, Germany {matthias.tuma | iossifidis | gregor.schoener} @neuroinformatik.rub.de

the learning of bipedal locomotion in a humanoid robot [11]. Degallier et al. [12] enabled a humanoid to perform a drumming task by superposing and switching between rhythmic and discrete movement patterns. Hersch and Billard [13] used VITE-like dynamical systems in redundant reference frames to produce human-like reaching acts. Studies on (dis)similarities between discrete and rhythmic movements in humans [14] should be seen as accompanying robotical research.

In earlier work, we proposed oscillating pattern dynamics to describe timing and coupling properties of discrete movement [15], developed a framework for the behavioral organization of sequential action [16], and fused both concepts to steer onset and termination points as well as movement times of discrete motor acts [17]. Other projects addressed obstacle avoidance using dynamical systems (see [4] for an overview). Santos [18] in turn attempted to include into these concepts a temporal stabilization mechanism against disturbances, which we however found to be subject to several limitations and structural inconsistencies (see section VI).

II. OVERALL SYSTEM

Building on this prior work, the present approach provides the capabilities i) to initiate and terminate discrete movement through a dynamical system for the stable generation of sequential actions; ii) to reach a possibly moving target while circumnavigating obstacles or dealing with other perturbations; iii) to do this while maintaining, as close as physically possible, an approximately constant total movement time; iv) to be implementable on a wide range of robotic systems for which directional dynamics and kinematics exist.

We formulate sets of dynamical systems on two layers of abstraction: on the level of behavioral sequencing, values for variables representing start, execution, and end phases of a movement are calculated depending on external signals. On the second level, dynamics for all behavioral variables that define the robot's state are integrated. In the simplest realization of 2D-planar movements, these can be the heading direction ϕ and the velocity v. The appearance of obstacles or a change of the target location influence the directional dynamics (e.g. that for ϕ) on-line: repellors and attractors are dynamically erected for safe navigation around obstacles and simultaneous acquisition of the target.

The dynamics for v makes use of stable limit cycle solutions of a Hopf oscillator. Generally utilized to create rhythmic motion [6], here one oscillator cycle is associated with one discrete reaching act by formulating a second order dynamics, i.e. one for the robot velocity. An adaptation rule for intrinsic properties of the limit cycle is formulated, so that the velocity profile is sped up or slowed down following perturbations, in order to reach the target in as close to the initially desired movement time as possible.

III. SEQUENCE GENERATING DYNAMICS

A dynamical system that produces discrete movement should be able to stabilize postural states before and after the main motion phase. The decision when to switch from

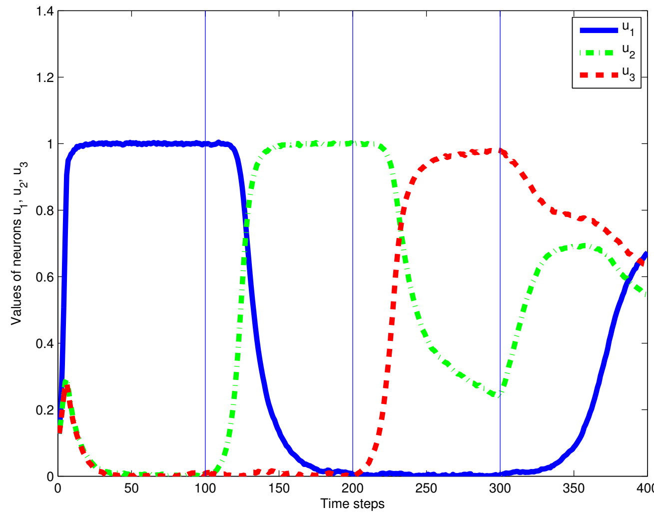

Fig. 1. Exemplary time course of three "neural" variables (u_1, u_2, u_3) governed by eq. (1). Their competitive advantages (μ_1, μ_2, μ_3) were controlled by quasi-booleans (b_1, b_2, b_3) according to eq. (2). Proportional differences between these b_i are amplified into decisions in which either one state is selected and all others are suppressed or in which a blend of multiple states may be activated at the same time. To demonstrate the state-controlling abilities and stability of the dynamics, the b_i were initially set to $(1, 0, 0)$ and changed every 100 timesteps. Their subsequent values were $(0.1, 0.9, 0.1)$, $(0.3, 0.3, 0.8)$, and $(0.6, 0.5, 0.55)$.

such a postural state to movement, or vice versa, should be met according to external information (e.g. reaching a critical time-to-contact, arrival at the target, disappearence of the target). The corresponding decision dynamics must therefore be continuously updated, but also be stable against temporary fluctuations of the respective signals.

The following competitive dynamics are formulated for each of three "neural" variables $u_i \in \{u_{Init}, u_{Hopf}, u_{Final}\}$ representing the three phases before, during, and after the movement, respectively. The system is based on a degenerate pitchfork bifurcation with an additional competitive term, that stabilizes states in which only one neuron u_i has values close to one while the other two have values close to zero [19]:

$$\tau \dot{u}_i = \mu_i u_i - |\mu_i|\, u_i^3 - \nu \sum_{a \neq i} u_a^2 u_i \qquad (1)$$

The parameter ν controls the strength of competition, and τ determines the time scale of the dynamics. If one competitive weight μ_i is larger than the other two, the corresponding neuron u_i is most likely to win the competition, i.e. to switch to the "on"-state while suppressing the other two neurons. For sufficiently small differences between the μ_i, multiple outcomes are possible, so that the system is multistable. For the dynamics in eq. (1), the values of $\nu = 2.1$ and $1.5 \leq \mu_i \leq 3.5$ give a reasonable trade-off between stability and flexibility.

The competitive advantages μ_i can be used to "switch" between different stable states of the system by binding them to external signals, e.g. a time-to-contact or the target proximity. Such a link to external information can have any mathematically suitable form, as long as the range of possible values is mapped into the correct interval, here between 1.5 and 3.5. A simple form uses quasi-logical variables $b_i \in [0, 1]$ that steer the switching dynamics through:

$$\mu_i = 1.5 + 2b_i, \quad \text{with } 0 \le b_i \le 1 \tag{2}$$

Fig. 1 shows an exemplary run of such a system to demonstrate its state-controlling and stability properties. See [17] for an elaborate framework for movement (re)initialization and termination based on coupling the μ_i to sensory input.

IV. DYNAMICS OF THE MOVEMENT STATE

In the dynamical systems approach to robotics, the robot's state is described by a set of behavioral variables that suit both the physical design of the robot and its task. The present approach uses two separate dynamics to define the robot's state: one for the robot velocity, and another for the remaining set of behavioral variables. In theory, this remaining set can have any suitable form, as long as it can provide for directional navigation towards the target and away from obstacles or joint limits. We shall, in the following, consider the simplest case of 2D-planar movement with a heading direction ϕ (in angular space, and in a fixed reference frame) as second behavioral variable.

A. Directional dynamics

The dynamics for ϕ contains one term that attracts towards the direction ψ_{Tar} in which the target lies, and another term which repels from the directions in which obstacles are perceived:

$$\tau\dot{\phi} = \sum_{i=1}^{N} f_{Obs,i}(\phi, \psi_{Obs,i}) + f_{Tar}(\phi, \psi_{Tar}) \tag{3}$$

Here, N is the total number of obstacles, sensed at angles $\psi_{Obs,i}$, and f_{Tar} is a linear attracting force of strength λ_{Tar}:

$$f_{Tar}(\phi) = -\lambda_{Tar}\,(\phi - \psi_{Tar}) \tag{4}$$

The obstacle terms $f_{Obs,i}$ are gaussian-modulated linear repelling functions:

$$f_{Obs,i} = \lambda_i\,(\phi - \psi_{Obs,i}) \exp\left[-\frac{(\phi - \psi_{Obs,i})^2}{2\sigma_i^2}\right] \tag{5}$$

The repellor strengths λ_i are set to decay exponentially with the distance between the obstacles and the sensors. In angular space, the range of the repellor is delimited by the width σ_i. All contributing terms are wrapped onto the circle. See [5] for a detailed example of a heading direction dynamics.

B. Velocity dynamics

The velocity v is set to the value of a state variable a at each timestep. Together with an auxiliary variable b, this state variable a evolves according to a 2D dynamical system:

$$\begin{aligned} \tau \begin{pmatrix} \dot{a} \\ \dot{b} \end{pmatrix} = & - c_1 \cdot u_{Init}^2 \begin{pmatrix} a \\ b \end{pmatrix} \\ & + u_{Hopf}^2 \cdot f_{Hopf}(a - R_h, b) \\ & - c_2 \cdot u_{Final}^2 \begin{pmatrix} a^2 - a \cdot \alpha_{tc} \\ b \end{pmatrix} \end{aligned} \tag{6}$$

where c_1 and c_2 are scaling parameters, α_{tc} is the bifurcation parameter of a transcritical bifurcation, and f_{Hopf} is a Hopf oscillator of radius R_h (possibly time-dependent, see below). As only one of the neural variables u_i is in an "on"-state at any given time, only one of the three terms in eq. (6) is different from zero. Associated with u_{Init} is an attractor for (a, b) at (0,0), which keeps the robot at rest before movement onset. After u_{Final} is activated, the velocity is stabilized at zero again for any negative α_{tc}. For positive values of α_{tc}, the velocity v approaches $\alpha_{tc} > 0$ itself. This is useful in cases in which, despite the stabilization mechanism described below, the target can not be reached in the initially planned time (e.g. due to hardware velocity limits).

During the main movement phase (the "on"-state of the neuron u_{Hopf}), a Hopf oscillator f_{Hopf} governs the dynamics:

$$\begin{aligned} f_{Hopf}(a - R_h, b) \; = \; & \begin{pmatrix} \lambda & -\omega \\ \omega & \lambda \end{pmatrix} \begin{pmatrix} a - R_h \\ b \end{pmatrix} \\ & -\gamma \left[(a - R_h)^2 + b^2\right] \begin{pmatrix} a - R_h \\ b \end{pmatrix} \end{aligned} \tag{7}$$

The angular frequency ω defines the cycle time $T = 2\pi/\omega$ and hence the total movement time. The parameters $\lambda > 0$ and $\gamma > 0$ mutually set the oscillator radius R_h:

$$R_h = \sqrt{\frac{\lambda}{\gamma}} \tag{8}$$

In phase space, the Hopf cycle is shifted along the a-axis by the cycle radius R_h, so that the variable a smoothly rises from zero to $2R_h$ and back during one oscillator cycle.

V. TEMPORAL STABILIZATION

If we disregard physical effects (like friction, motor discretization, etc.), the robot's velocity profile will equal the time course of the variable a, defined by eqs. (6) and (7). In the dynamics for (a, b), a fixed cycle radius R_h is appropriate if no disturbances (e.g. obstacles, target displacement, etc.) occur. If, however, the task setup changes during the robot's movement, an adaptation rule for the cycle radius R_h will automatically adjust the system in eq. (7) in order to stabilize total movement time.

A. Undisturbed case

Integrating the dynamics in eq. (7) for a fixed cycle radius R_h gives the distance s covered by the robot in time t:

$$s(t) = \int_0^t R_h(1 - \cos\omega\tau)d\tau = R_h(t - \frac{1}{\omega}\sin\omega t) \tag{9}$$

After one full cycle, the distance $2\pi R_h/\omega$ is reached. Hence, the radius of the oscillator should initially be set to

$$R_h = \frac{\omega D(t = 0)}{2\pi} \tag{10}$$

with $D(t = 0)$ being the distance between the robot's initial position (x_0, y_0) and the target coordinates (x_{tar}, y_{tar}). Together, the dynamics in eq. (3) keep the robot oriented towards the target, and eq. (7) drives the robot exactly over the required distance in a sinusoidal velocity profile.

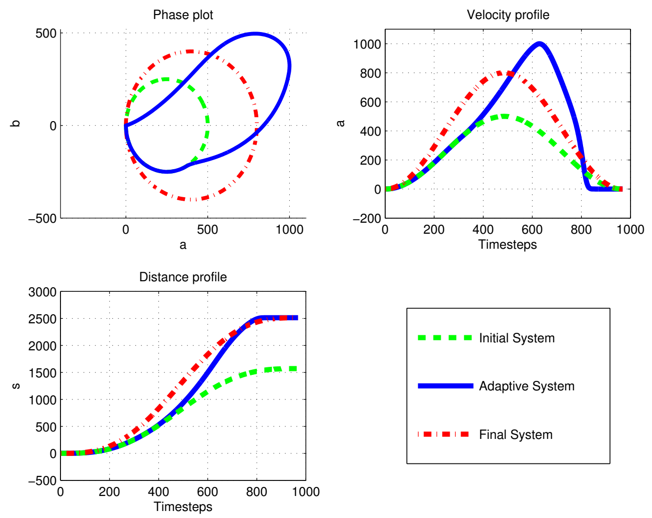

Fig. 2. Three different realizations of integrating eq. (7). The subplots show the resulting paths in phase space, the time course of the variable a (velocity profile), and the time course of its integral s (distance profile). The "variable" system begins with the same oscillator radius R_h as the "initial" system. Starting at one third of the cycle, the total distance to be covered is gradually increased up to the total distance reachable with the "final" system, and R_h is adapted according to eq. (12). All units are arbitrary.

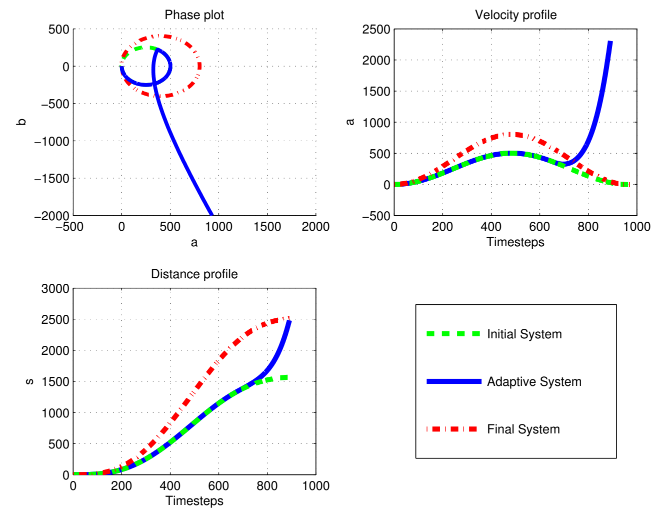

Fig. 3. Three different realizations of integrating eq. (7). The subplots show the resulting paths in phase space, the time course of the variable a (velocity profile), and the time course of its integral s (distance profile). The "variable" system begins with the same oscillator radius R_h as the "initial" system. Starting at two thirds of the cycle, the total distance to be covered is instantaneously increased up to the total distance reachable with the "final" system, and R_h is adapted according to eq. (12). All units are arbitrary.

B. Disturbed case

There are many possible influences that may disturb the planned time course of robot motion, including physical stalling, movement of the target, or obstacles along the path. Regardless of whether the disturbance requires speeding up or slowing down, the target should still be reached in as close to the initially planned movement time as possible. To this end, an adaptation rule for the Hopf cycle radius R_h is formulated. For the undisturbed case, using $\omega = 2\pi/T$ and eqs. (9), (10) yields the relation

$$\begin{aligned} D(t=0) &= D(t) + \int_0^t v(\tau)d\tau \\ &= \frac{D(t)}{\left(1 - \frac{t}{T} + \frac{\sin(2\pi \cdot t/T)}{2\pi}\right)} \end{aligned} \tag{11}$$

between the initial distance, $D(t=0)$, the distance remaining, $D(t)$, and the currently elapsed fraction of one oscillator cycle, t/T. With eq. (10), the last identity can be interpreted as an adaptation rule for the Hopf cycle:

$$R_h(t) = \frac{\omega}{2\pi} \frac{D(t)}{\left(1 - \frac{t}{T} + \frac{\sin(2\pi \cdot t/T)}{2\pi}\right)} \tag{12}$$

This online updating rule takes into account not only the current distance to the target, but adapts the limit cycle so as to accelerate or decelerate the motion sufficiently so that remaining distance is traversed within the remaining time. If no disturbances occur, the adaptation rule in eq. (12) will not alter R_h.

In eq. (8), the two parameters λ and γ jointly define the cycle radius through their ratio, while their absolute values influence the relaxation behavior of the system in eq. (7). This additional degree of freedom should be carefully adjusted in order to provide reliable relaxation into the current oscillator state. In the implementations below, we use a fixed value for λ of 0.1.

VI. IMPLEMENTATION

The approach laid out above was tested in simulation and implemented on a mobile robotic vehicle (K-Team Khepera unit) generating movements in the plane. Its performance was tested in a target acquisition task in cluttered environments.

Although we also implemented a temporal stabilization mechanism proposed by Santos [18], problems in this approach prevented a side-by-side comparison. Specifically, Santos' use of independent dynamics and movement abortion conditions for each cartesian coordinate axis severely constrained the scope of possible task setups. Also, Santos' stabilization method rescaled the velocity profile in the case of delays in an ad hoc manner that does not theoretically warrant invariant movement time. A comparison would thus have merely demonstrated that a fixed but arbitrary degree of qualitative adaptation is not well suited across varying target configurations.

A. Simulations

Figs. 2 and 3 show simulations of the temporal stabilization mechanism. In both figures, three different realizations of integrating the dynamics in eq. (7) are displayed each in phase space, as a velocity profile, and as a distance profile. The first realization is a complete Hopf cycle with an oscillator radius of $R_h = 250$ units ("initial system"), the second a complete Hopf cycle with an oscillator radius of $R_h = 400$ units ("final system"), and the third is the result

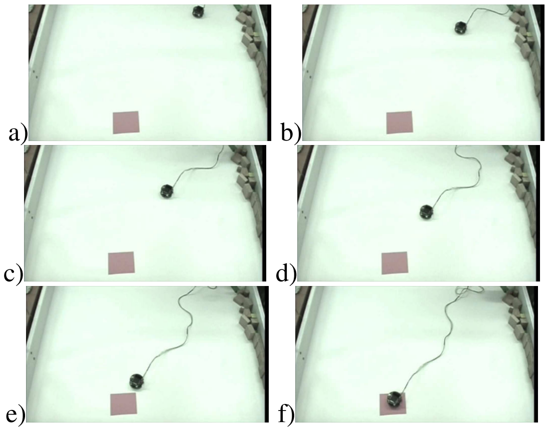

Fig. 4. Sequence depicting robot movement to a target area without the occurence of disturbances. The robot's velocity is set according to eq. (6), and the distance profile follows eq. (9).

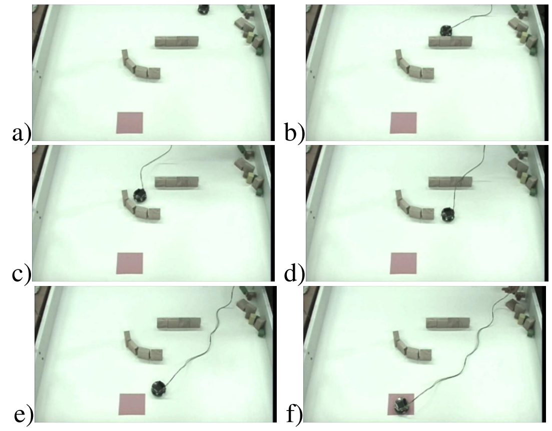

Fig. 5. Sequence depicting robot movement in a parcours unknown at starting time. The oscillator radius is adapted according to eq. (12), and the movement time approaches that of the undisturbed case shown in Fig. 4.

of integrating eq. (7) with a time-variable oscillator radius calculated from the adaptation rule in eq. (12).

This "adaptive system" starts with an initial radius of $R_h = 250$. During the oscillator cycle, the fictional target is assumed to move away from the robot, up to a distance that would have required an initial oscillator radius of $R_h = 400$. In Fig. 2, this enlargement of the distance to be covered is gradual and starts at about one third of the cycle. In Fig. 3, the enlargement is instantaneous at about two thirds of the cycle, thus requiring a more sudden increase of velocity to cover the remaining distance in the remaining time. In both cases, the adaptive system produces a velocity curve well-suited for covering the larger distance in the same time.

For Fig. 3, note that the integrated eq. (7) is only one part of the whole dynamics in eq. (6), in which an additional attractor to the origin "re-sets" (a, b) and stops the robot's motion. In practice, two parameters of this final attractor (onset criterion b_{Final} and attractor strength c_2) have to mutually balance between two opposing properties: first, smoothness of trajectories even in cases of high velocities (e.g., as in Fig. 3), and second, the degree to which the overall trajectory in phase space given by eq. (6) represents the system that is given by eqs. (7) and (12) only. Apart from this trade-off and additional physical effects like friction and motor discretization, the courses of the simulated dynamics remain valid for the following hardware implementations, both for gradual and sudden changes in the task setup.

B. Khepera mobile unit

We use a two-wheeled K-Team Khepera robot to demonstrate the approach. But, as laid out in section IV, the dynamics proposed are to a large degree independent from the physical implementation, and the only modifications necessary are in the directional dynamics.

The implementation on a Khepera robot demonstrates that the approach, in spite of its mathematical sophistication, is suited for low-level robotic units with uncalibrated sensors and a fairly simple control system, as is typical of the dynamical systems approach. Here, obstacle sensing is provided by the Khepera's built-in infrared sensors with a maximum range of approx. 8 cm. The odometry is based on the Khepera's wheel encoder values. For simplicity, targets were directly represented through coordinates rather than by visual extraction. The dynamics were integrated on an external PC, and velocities for each wheel communicated to the unit. All dynamics were additionaly superposed with gaussian white noise to provide realistic conditions and ensure escape from meta-stable states.

For the sequence generating dynamics in eq. (1), the competitive advantages μ_i were chosen to depend on a set of logical conditions b_i as in eq. (2). After an initial phase allowed for orienting towards the target, b_{Hopf} was activated to begin the movement phase. Once the robot was as close to the target as 6% of the original total distance, b_{Final} was activated.

Fig. 4 shows a Khepera robot while approaching a target without constraints. Initially, the oscillator radius R_h is set according to eq. (10), i.e. the distance to the target coordinates. The resulting trajectory is a straight line towards the target, with a velocity profile similar to that of the initial and final system in Fig. 2. In Fig. 5, the path towards the target is obstructed by obstacles, which the system has no prior knowledge about and only senses as it drives close by. They are circumnavigated by the heading direction dynamics in eq. (3). At the same time, the total distance needed to be driven in the time of one oscillator cycle rises. The stabilizing mechanism in eq. (12) thus gradually increases the radius R_h of the oscillator and produces a velocity profile similar to that of the variable system in Fig. 2.

Movement times for the experiment runs are, averaged over several trials, shown in Tb. I. The setup in Fig. 5 is listed as "Medium Disturbance", while another course, requiring more extensive detours, is shown under "High Disturbance". The total distance driven gives an overview over the demands of both setups. Although an influence of the disturbances on the movement time is visible, it is marginal when compared to the relative increase in total distance driven and due both to the system's relaxation in phase space and physical effects.

TABLE I

AVERAGE MOVEMENT TIMES (MT) IN DIFFERENT SETUPS.

Setup	Total Distance driven (cm)	MT (s)	Increase in Distance (Factor)	Increase in Time (Factor)
Undisturbed	72.4	12.3		
Medium Disturb.	96.0	12.7	1.33	1.03
High Disturb.	109.7	12.9	1.52	1.05

C. Robustness

For obstacle avoidance and generation of movement sequences, a dynamics of the heading direction and a competitive dynamics were employed in a modular fashion. These building blocks maintain their original properties of robustness with regard to their intrinsic parameters (e.g. ν, λ_i) [8][19]. Concerning the velocity dynamics with its adaptation rule, the oscillator will not relax into its foreseen state fast enough if the value of λ is too high. Below a corresponding threshold region, we found the system to be stable both against changes in λ and for a variety of representative changes in the task setup, including sudden displacement of targets (e.g., as in Fig. 3). While a systematic theoretic examination of robustness does not seem feasible, a more comprehensive empirical approach such as in [13] can be the aim of future investigations.

VII. CONCLUSION

We presented a framework for the generation and temporal stabilization of discrete movements. Consistently formulated within the dynamical systems approach to robotics, the proposed method has the capabilities i) to initiate and terminate discrete movement through a dynamical system for the stable generation of sequential actions; ii) to reach a possibly moving target while circumnavigating obstacles or dealing with other disturbances; iii) to do this while maintaining, as close as physically possible, an approximately constant total movement time; iv) to be implementable on a wide range of robotic systems for which directional dynamics and kinematics can be formulated. We have demonstrated the approach on a Khepera mobile unit to show its reliability even when depending on low-level sensor information. The use of a wheeled vehicle moving in a 2D plane with obstacles should however be seen as one particular realization of a more abstract problem: stabilizing movement time of discrete movements in the presence of perturbations.

Other tasks for the future include exploiting the coupling capabilities of the Hopf oscillator when coordinating multiple movements and further exploring possible similarities and necessary differences between systems generating discrete versus rhythmic movement. Currently, we are working on an extended implementation for a redundant robot arm with seven degrees of freedom.

VIII. ACKNOWLEDGMENTS

The authors are grateful for being supported by DESIRE, German BMBF grant FKZ 01 IME 01G.

REFERENCES

[1] R. C. Arkin, *Behavior-Based Robotics*. MIT Press, 1998.

[2] G. Schöner, "Timing, clocks, and dynamical systems," *Brain and Cognition*, vol. 48, pp. 31–51, 2002.

[3] S. Strogatz, *Nonlinear Dynamics and Chaos*. Westview Press, 2001.

[4] G. Schöner, M. Dose, and C. Engels, "Dynamics of behavior: Theory and applications for autonomous robot architectures," *Robotics and Autonomous Systems*, vol. 16, pp. 213–245, 1995.

[5] I. Iossifidis and G. Schöner, "Dynamical systems approach for the autonomous avoidance of obstacles and joint-limits for an redundant robot arm," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2006.

[6] A. J. Ijspeert, "Central pattern generators for locomotion control in animals and robots: A review," *Neural Networks*, vol. 21, no. 4, pp. 642–653, 2008.

[7] R. Menzner, A. Steinhage, and W. Erlhagen, "Generating interactive robot behavior: a mathematical approach," in *From Animals to Animats: Proc. of the Int. Conf. on Simulation of Adaptive Behavior*, 2000.

[8] B. Fajen, W. Warren, S. Temizer, and L. Kaebling, "A dynamical model of visually-guided steering, obstacle avoidance, and route selection," *International Journal of Computer Vision*, vol. 54, no. 1-2, pp. 13–34, 2003.

[9] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2002.

[10] L. Righetti, J. Buchli, and A. J. Ijspeert, "Dynamic hebbian learning in adaptive frequency oscillators," *Physica D: Nonlinear Phenomena*, vol. 216, no. 2, pp. 269–281, 2006.

[11] L. Righetti and A. J. Ijspeert, "Programmable central pattern generators: an application to biped locomotion control," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2006.

[12] S. Degallier, C. Santos, L. Righetti, and A. J. Ijspeert, "Movement generation using dynamical systems: a humanoid robot performing a drumming task," in *Proc. of the IEEE-RAS Int. Conf. on Humanoid Robots (HUMANOIDS)*, 2006.

[13] M. Hersch and A. Billard, "Reaching with multi-referential dynamical systems," *Autonomous Robots*, vol. 25, pp. 71–83, 2008.

[14] N. Hogan and D. Sternad, "On rhythmic and discrete movement: Reflections, definitions and implications for motor control," *Experimental Brain Research*, vol. 181, pp. 13–30, 2007.

[15] G. Schöner, "A dynamic theory of coordination of discrete movement," *Biological Cybernetics*, vol. 63, pp. 257–270, 1990.

[16] A. Steinhage and G. Schöner, "Dynamical systems for the behavioral organization of autonomous robot navigation," in *Sensor Fusion and Decentralized Control in Robotic Systems: Proceedings of SPIE*, 1998.

[17] G. Schöner and C. 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.

[18] C. 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.

[19] 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.