Neural Fields for Real-Time Navigation of an Omnidirectional Robot

Mohamed Oubbati Günther Palm
Institute of Neural Information Processing, University of Ulm, 89069 Ulm, Germany

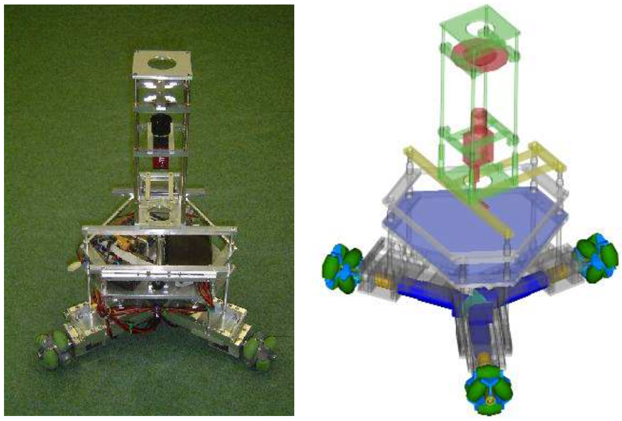
***Abstract*— In this paper, we implement a biologically inspired approach for the generation of real-time navigation of a real omnidirectional robot. The approach is based on a so-called neural fields, which are equivalent to continuous recurrent neural networks. Due to its dynamical properties, a neural field produces only one localized peak that indicates the optimum movement direction of the robot. Experimental results support the validity of the approach.**

***Index Terms*— Mobile Robots, Neural Fields, Behavior-based Control, Navigation.**

I. Introduction

The basic task the robot has to perform is to reach a goal under constraints, e.g moving towards a goal while avoiding obstacles. Approaches that have been developed for this problem can be divided into global and local methods. Global methods require the environment to be completely known and the terrain should be static, and they return a continuous free path. By contrast, local methods need only local information. It means that the path planning is done while the robot is moving, in response to environmental changes. Due to their low-computational costs, local methods are much more suitable for real application where the environmental state changes continually. The most popular local method is the potential field approach proposed by Khatib [1]. The idea is to consider that the robot moves under influences of an artificial potential field. The target applies an attractive force to the robot, while obstacles exert repulsive forces onto the robot. The sum of all forces determines the subsequent direction of the movement. While the potential field principle is particularly attractive because of its elegance and simplicity, substantial drawbacks have been identified, i.e. local minima (cyclic behavior), no passage between closely spaced obstacles, oscillations in narrow passages, etc [2].

Recently, the theory of dynamical systems has proven to be an elegant and easy to generate robot behavior [3][4][5]. The so-called Dynamic Approach invented by Schöner in 1995 [6] provides a framework to design differential equations for so-called behavior variables, which generates the robot's behavior. Usually, these variables directly parameterize the elementary behavior to be generated. However, there are cases for which the behavioral variable needs a more general form. For example, a behavioral variable can have multiple values or even no value at all. In those cases, it is necessary to express it by a continuous function. The neural field's model can represent such a behavioral variable. Originally, these fields were proposed by Amari [7] as models of the neurophysiology of cortical processes. They are equivalent to continuous recurrent neural networks, in which units are laterally coupled through an interaction kernel and receive external inputs. The concept of neural fields has proven to be a simple and an elegant approach to generate a behavior-based control for mobile robots [8][9]. In [10] we used neural fields to navigate the mobile robot to its goal in an unknown environment without any collisions with static or moving obstacles. Furthermore, their competitive dynamics were used to optimize the target path through intermediate home-bases. More recently, we investigated how neural fields can produce an elegant solution for the problem of moving multiple robots in formation [11]. The objective was to acquire a target, avoid obstacles, and keep a geometric configuration at the same time.



(a) (b)

Fig. 1. Omnidirectional robot. (a) hardware photo. (b) CAD model

In this paper, the neural field approach is implemented on a real omnidirectional robot (Figure 1). The objective is to acquire a target without any collisions with static or moving obstacles. We begin by describing the basic concept of neural fields. Then we will present our navigation model, supported with some experimental results.

II. Neural Field Theory

The field equation of a one-dimensional neural field is given by

$$\tau \dot{u}(\varphi,t) = -u(\varphi,t) + S(\varphi,t) + h + \int_{-\infty}^{+\infty} w(\varphi,\acute{\varphi}) f(u(\acute{\varphi},t) d\acute{\varphi} \quad (1)$$

where $u(\varphi, t)$ is the field excitation at time t ($t \geq 0$) at the position $\varphi \in R$. The temporal derivative of the excitation is defined by

$$\dot{u}(\varphi, t) = \frac{\partial u(\varphi, t)}{\partial t} \tag{2}$$

The constant h defines the pre-activation of the field, and $f(u)$ is the local activation function. Usually, f is chosen as a step-function:

$$f(u) = \begin{cases} 1, & u \geq 0 \\ 0 & u < 0 \end{cases} \tag{3}$$

The stimulus $S(\varphi, t) \in R$ represents the input of the field which is dependent on the field position and varies with time. A nonlinear interaction between the excitation $u(\varphi)$ at the position φ and its neighboring positions is achieved by the convolution of an interaction kernel $w(\varphi, \acute{\varphi})$.

Depending on the parameter h and the form of S, f and w, the activation dynamics (1) can have different types of solutions [7]:

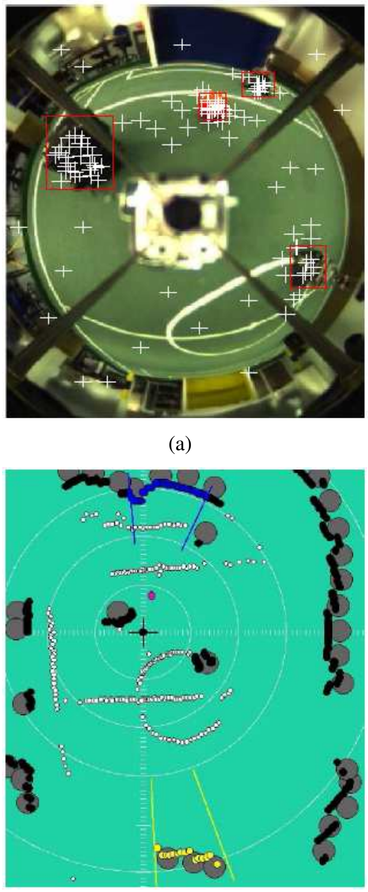
1. $\emptyset$-solution, if $u(\varphi) \leq 0, \forall \varphi$.
2. ∞-solution, if $u(\varphi) > 0, \forall \varphi$.
3. a-solution, if there localized excitation from a place a_1 to a place a_2. This solution is also called a *single-peak* or *mono-modal* solution.

The correct choice of the parameters of the field equation enables the existence of one a-solution. In this solution, when an input of a stimulus $S(\varphi, t)$ is very large compared with the within-field cooperative interaction, a single-peak will be stabilized by interaction, even if the stimulus is removed. In a robot behavior-based control, the position of the single-peak on the field is used as a behavior variable, for example the heading angle of the robot.

III. Robot System

A. Hardware Platform

In this work, the neural field approach is implemented on an omnidirectional mobile robot built recently at the university of Stuttgart (Figure 1). It is equipped with 3 omni-wheels, each of them driven by a 90W DC motor. Gearboxes with reduction ratios of 14:1 are used to reduce the high angular speeds of the motors (7000 rpm) and to amplify the wheel's mechanical torques, and 500 ppr digital incremental encoders are used to measure the actual wheels speed. Motors are controlled by 3-channel microprocessor-based interface. The robot is equipped with a laptop to manage different sensors and tasks. The communication between the sensors and the laptop can be done through USB, RS232, or IEEE1394 (FireWire). For environment sensing, the robot platform is equipped with an omnidirectional vision system, based on a hyperbolic mirror and a standard IEEE1394 (FireWire) camera. The Omnidirectional vision provides the robot a very large field of view, which has some useful properties. For instance, it can facilitate the tracking of robots, the ball, and a set of environmental features used for self-localization. To extract information, the captured image is segmented using the calibrated colors of relevant objects, such the ball, field lines, and obstacles. An example of an image with recognized objects is shown in Figure 2.



(a)

(b)

Fig. 2. Information extraction from the camera a) Image captured from the omni-camera b) Recognition of relevant objects: lines(white), ball(red), obstacles (black), and goals (blue and yellow).

B. Self-Localization

To estimate the robot pose relative to its environment, we use a probabilistic localization algorithm called Monte Carlo localization (MCL) [12]. MCL can solve the localization problem in a highly robust and efficient way, even with temporary partial or total occlusion of relevant sensor features.

C. Software Architecture

The Software architecture has a modular design structured into three parallel working layers (Figure 3):

1 *Sensor layer*, where low level features or raw data are gathered.

2 *World Model layer*, which stores locally gathered data as well as communicated data by other robots. It provides also so-called Data Processor modules, for example monte carlo based localization and Kalman filter based ball tracking.

3 *Control layer*, responsible for the execution of actions such as driving or shooting a ball.

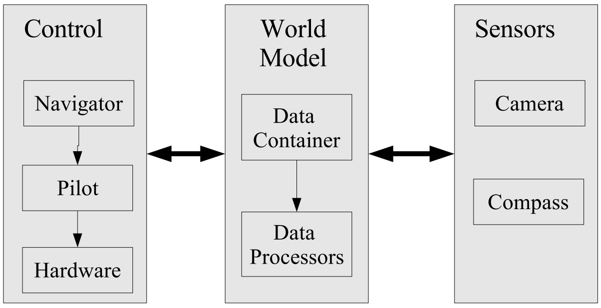


Fig. 3. Robot Software Architecture.

IV. Control Design

When the neural field (1) is in a-solution and an input of a stimulus $S(i,t)$ at time t is very large compared with the within-field cooperative interaction, this input will dominate the solution. As a result, a single-peak will be stabilized by interaction and remains there, even if the stimulus is removed. The idea is to use the position of the peak to encode the optimum movement direction of the robot. In the experiments, the neural field has to encode angles from $-\pi$ to $+\pi$ divided to N discrete directions.

A. Field Activation

For numerical reasons, the one-dimensional neural field (1) is discretized:

$$\tau u(i,t+\Delta t) = (\tau - \Delta t)u(i,t) + \Delta t\left[\sum_{j=1}^{N} w(i,j) f[u(j,t)] + S(i,t) + h\right] \quad (4)$$

The interaction kernel is chosen as:

$$w(i,j) = k_w e^{-\sigma_w (i-j)^2} - H \quad (5)$$

where the parameter σ_w fixes the range of excitation, and k_w its amplitude. The global inhibition H allows only one localized peak on the field. The interaction kernel $w(.)$ is chosen as a Mexican hat function so that excitatory connections dominate for proximate units, and inhibitory connections dominate at greater distances. Its parameters were chosen depending on the robot and the obstacles shapes.
After the stabilization of the field, the most activated neuron decodes the direction to be executed by the robot:

$$\varphi_F = arg_{\max}\{u(i) | i \in [1, N]\} \quad (6)$$

B. Field Stimulus

Before selecting an appropriate direction, the neural field needs some necessary information (stimulus). The stimulus is determined according to two stimulus-functions. These functions describe

- the direction towards the target $S_T(i,t)$. This stimulus is designed excitatory, showing an attraction towards the target direction. It is chosen as

$$S_T(i,t) = C_{T1} - C_{T2}|i - i_T(t)| \quad (7)$$

where C_{T1}, C_{T2} are constants positive, and $i_T(t)$ is the field position, equivalent to the main target direction at time t.
- directions to obstacles $\{S_{Ol}(i,t) : l \in [1, N_{Obst}]\}$, where N_{Obst} is the number of obstacles detected by the robot sensors. This stimulus must be inhibitory, since obstacles collision must be avoided. It is chosen as a Mexican Hat function centered at the direction of an obstacle.

$$S_{Ol}(i,t) = C_O e^{-\sigma_O (i-i_l)^2} \quad (8)$$

where C_O and σ_O are positive constants. σ_O defines the range of inhibition of an obstacle. In practical situations, this parameter is tuned according to the radius of the robot and the obstacles. i_l reflects the direction of the obstacle l at time t. However, the stimulus considers only obstacles in distances d_{Ol}, which are below a threshold d_{Th}.

The contributions of different stimuli determine the state of the field. Thus, the global stimulus of the field at time t is determined by

$$S(i,t) = S_T(i,t) - \sum_{l=1}^{N_{Obst}} g(d_{Ol}) S_{Ol}(i,t) \quad (9)$$

where g is a step function:

$$g = \begin{cases} 1, & d_{Ol} < d_{Th} \\ 0 & d_{Ol} \geq d_{Th} \end{cases} \quad (10)$$

Using the step function g, only obstacles , which their distances d_{Ol} to the robot are below a threshold d_{Th}, are considered.

C. Dynamics of Speed

In a free obstacle situation, the robot moves with its maximum speed $V_{\max}$, and slows down when it approches a target. This velocity dynamics can be chosen as:

$$V_T(t) = V_{\max}(1 - e^{-\sigma_v d_T}) \quad (11)$$

where σ_v is a positive constant tuned in a relation with the acceleration capability of the robot. d_T represents the distance between the robot and the target at time t.
Close to obstacles, the robot needs also to be slowed down. In case of many obstacles, the nearest obstacle on the robot direction is considered. This dynamics can be chosen as:

$$V_O(t) = C_O(1 - g(d_{no}) e^{-\sigma_{Ov}(\varphi_F - i_{nO})^2}) \quad (12)$$

where C_O and σ_{Ov} are positive constant, i_{nO} is the nearest obstacle direction to the robot movement direction and d_{nO} is its distance relative to the robot.
The final dynamics of the velocity is the contribution of (11) and (12). It is also considered when no appropriate direction can be selected, for example when the robot is completely surrounded by obstacles. In this case the robot must stop until the environmental situation changes. Thus, the robot velocity that satisfies the above design creteria is the following:

$$V(t) = \begin{cases} V_T(t) V_O(t), & \varphi_F > 0 \\ 0 & \varphi_F \leq 0 \end{cases} \quad (13)$$

V. Experimental Results

The expriments were performed on a soccer field, since the localization software is made for a RoboCup soccer game. On the neural field we chose the number of discrete directions $N = 60$ neurons, which means that each direction N decodes a step of 6^o. The neural field is expected to align the robot's heading with the direction of the target, and to bring it away from the nearby obstacles. We will illustrate this with some experiments showing target acquisition with collision avoidance for multiple static and moving obstacles.

The first experiment (Figure 4) shows the test of acquisition a target while avoiding static obstacles. Figure 4 (a) and Figure 4 (b) show the activation and the stimulus in their temporal course, respectively. In the first 20 time steps, both the stimulus and the activation are unimodal, since no obstacles are detected. The peak location provides therefore the direction of the target, and the robot moves straight towards it. In the following time steps, the stimulus contains both the target and obstacle entries (obsacle entries are depicted as a "cavity" in Figure 4 (b)). Therefore, it becomes bimodal. By contrast, the field activation remains unimodal, since the field's parameters are adjusted such as the field is in the a-solution state, i.e. stabilization of a single-peak even the stimulus is removed. Due to the design of the stimulus, the peak moves smoothly to an optimum local position on the field, which corresponds to a new robot behavior "target acquisition with obstacle avoidance". The field provides now the appropriate heading direction (Figure 4 (c)), which permits at this stage the avoidance of obstacle 1. Furthermore, according to (13), the robot moves with its maximum linear velocity when no obstacles are detected, and slowed down when it is near the obstacle or approaches the target (Figure 4 (d)). Due to the stimulus contribution of all obstacles, and the correspodant activity of the field, the robot was obliged to pass also to the right of obstacle 2 and obstacle 3, until having a free path to the target. The global path of the robot is illustrated on Figure 4 (e). Figure 5 shows some photos from its video sequences. In the second experiment (Figure 6), the distance between obstacles 2 and 3 is now large enough enabling the robot to pass through. At the begining of this experiment the robot has the same behavior as in experiment 1 until it passes obstacle 2. At this moment the robot could show a flexibility and moved effectively between obstacles 2 and 3 in order to reach the target. Figure 7 shows some photos from video sequences taken from the second experiment. The third experiment (Figure 8) was carried out to verify the navigation approach ability in tackling moving obstacles. In this context, the problem is how to reach a target in the presence of dynamically moving obstacles. When an obstacle is moving, collision avoidance is harder because the robot has to detect not only its position, but also its direction. As in the prior experiments, during the first 20 time steps, the field activation (Figure 8 (a)) is unimodal, since the stimulus (Figure 8 (b)) contains only the target entries. In the following time steps, the robot deviates from the straight line to avoid the static obstacle 1.

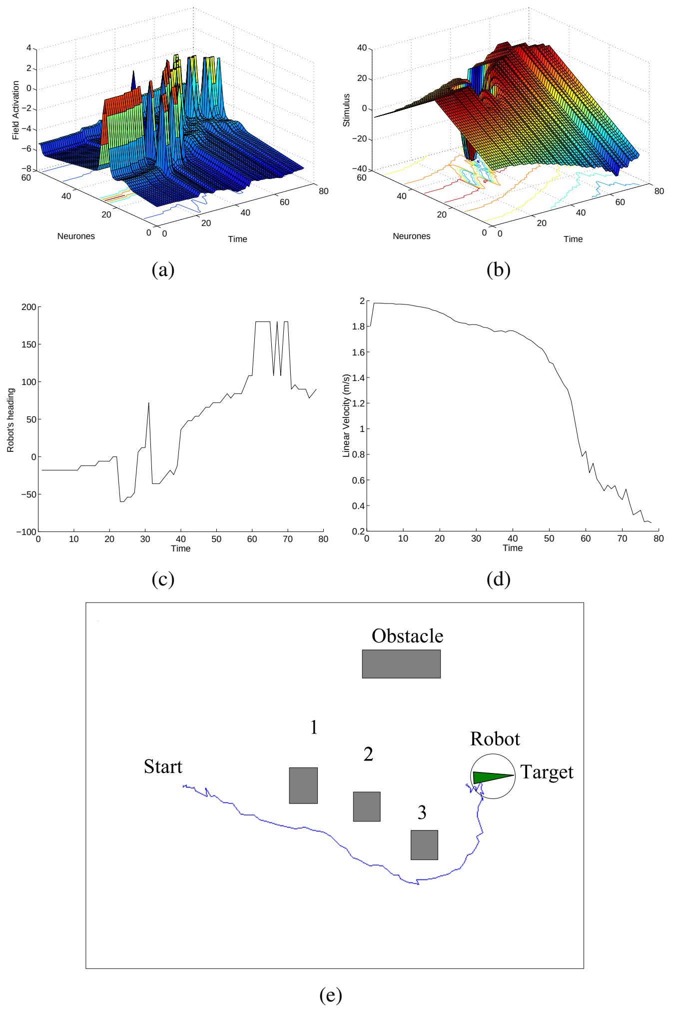


Fig. 4. Target acquisition with Obstacle avoidance: Experiment 1.

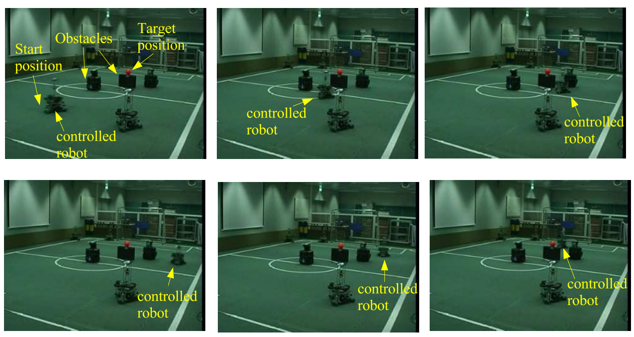


Fig. 5. Photos from the video sequence of the Experiment 1.

At time step 51, the stimulus (Figure 8 (b)) contains now entries from the moving obstacle 2. When approaching it, the robot percieves that it is a moving obstacle and knows its direction. It maintaines a lower speed (Figure 8 (d)) and changes its direction (Figure 8 (c)) to avoid it. After passing obstacle 2, the peak moved to the position, which encodes the target direction. The final path is illustrated on Figure 8 (e). In this experiment the robot was able to reach the target after reacting appropriately to the unexpected static and moving obstacles.

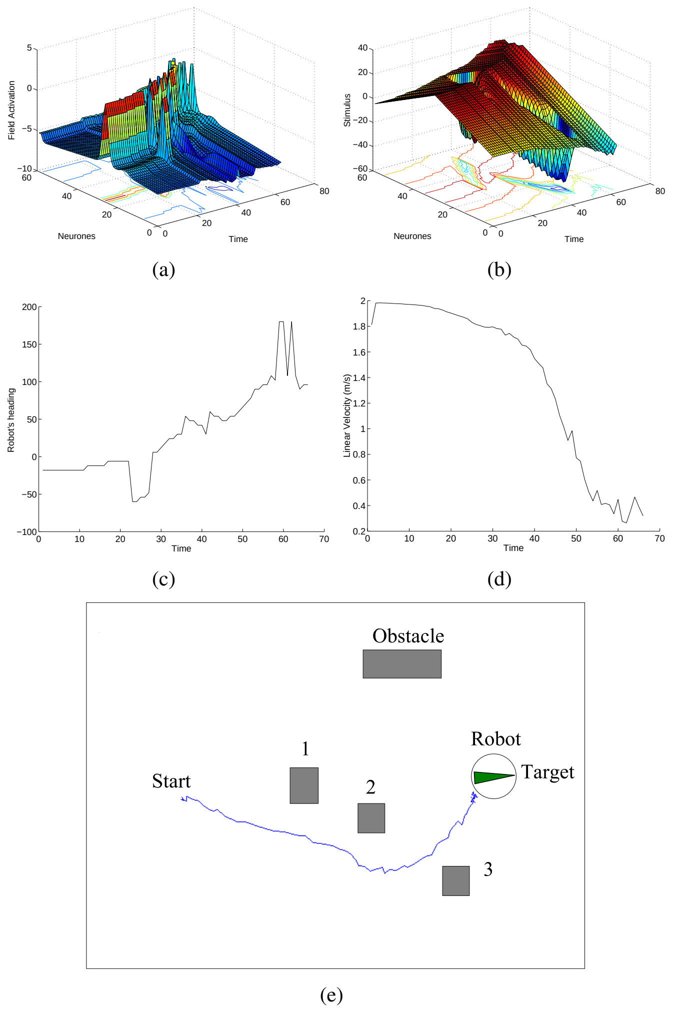


Fig. 6. Target acquisition with Obstacle avoidance: Experiment 2.

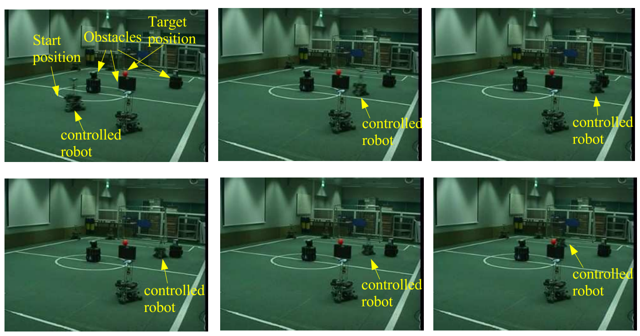


Fig. 7. Photos from the video sequence of the Experiment 2.

VI. Discussion and Conclusion

The most popular method to solve the problem of target acquisition with obstacle avoidance is the potential field approach proposed by Khatib [1]. While the potential field principle is particularly attractive because of its elegance and simplicity, substantial drawbacks have been identified, i.e. local minima (cyclic behavior), no passage between closely spaced obstacles, oscillations in narrow passages, etc [2].

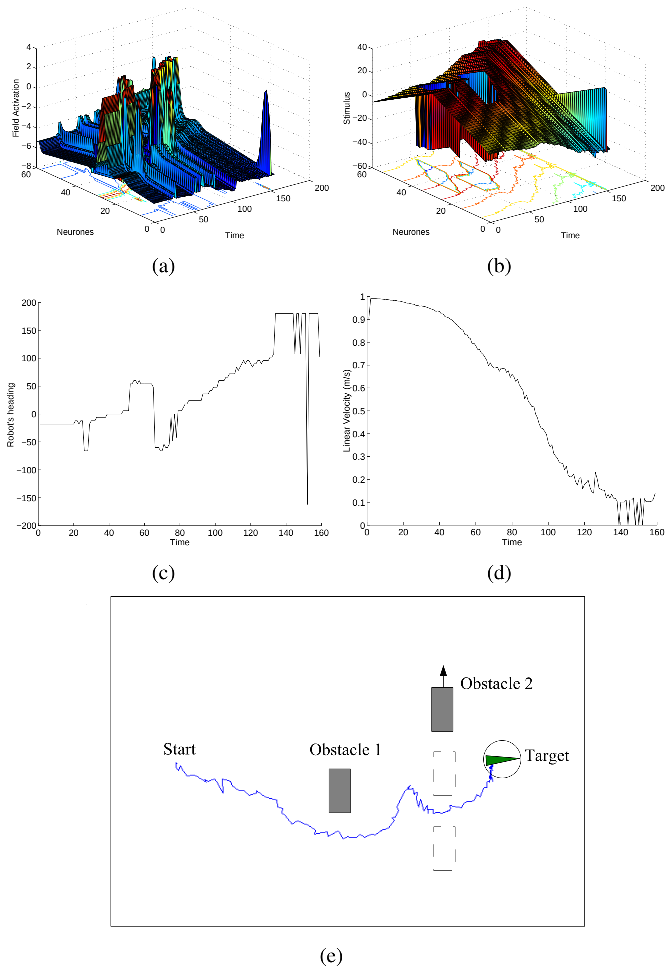


Fig. 8. Target acquisition with Moving Obstacle avoidance

With this paper, we tried to solve these problems using a biologically inspired approach based on a so-called neural fields, which are equivalent to continuous recurrent neural networks. The navigation model is developed in order to produce peak-solutions of the field activation, which encode the appropriate robot direction in response to a change in the environment. When two potential targets (directions) are presented to the field, two type of responses are possible. When the stimulus consists of two narrowly spaced peaks, a single peak of activation localized over an averaged direction is generated. When the stimulus contains two peaks which are more separate, the field dynamic becomes bi-stable and generates *only* a single peak, positioned either at one or at the other stimulated location. The possibility to generate only one single peak presents the power of this approach. This property could, for instance, solve some drawbacks of the potential field method: local minima (cyclic behavior) and no passage between closely spaced obstacles. Moving through narrow passages is tested only on simulation, and it has shown promising results. Furthermore, the tunning of the field parameters is simple and doesn't depend on the environement. First, the field should be forced to the a-solution, in order to produce a single peak. Then, the sitmulus should be tuned according to the problem at hand. In [10] we used neural fields to navigate the mobile robot to its goal in an unknown environment with optimizing the target path through intermediate home-bases. More recently in [11], we investigated how neural fields can produce an elegant solution for the problem of moving multiple robots in formation. The objective was to acquire a target, avoid obstacles and keep a geometric configuration at the same time. In this paper we want to test this approach on a real omnidirectional robot. The results show that this approach provides robustness against uncertainty in the perception system, and demonstrates the robot ability to adapt its behavior to unexpected situations and navigate among static and moving obstacles. However, a real implementation for a multiple robots system is still needed to test this approach in more complex scenarios, and to ensure the effectiveness of the simulation results achieved in [11].

REFERENCES

[1] O. Khatib, Real-time obstacle avoidance for manipulators and mobile robots, *in International Journal of Robotic Research*, 1986.

[2] Y. Koren and J. Borenstein, Potential field methods and their inherent limitations for mobile robot navigation, *in Proceedings of the IEEE Conference on Robotics and Automation*, pages 1398-1404, California, April 1991.

[3] S. Monteiro and E. Bicho, A Dynamical Systems Approach to Behavior-Based Formation Control, *in Proceedings of the 2002 IEEE International Conference on Robotics and Automation*,Washington, May 2002.

[4] Steinhage A. and Schöner G., The Dynamic Approach to Autonomous Robot Navigation, *in IEEE International Symposium On Industrial Electronics*, 1997.

[5] Goldenstein S. and Metaxis D. M. and and Large E. W, Nonlinear Dynamic Systems for Autonomous Agent Navigation, *in Proceedings of the Seventeenth National Conference on Artificial Intelligence*, American Association for Artificial Intelligence, 2000.

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

[7] S. Amari, Dynamics of pattern formation in lateral-inhibition type neural fields, Biol. Cybern., vol. 27, pp. 77-87, 1977.

[8] H. Edelbrunner, U. Handmann, C. Igel, I. Leefken, and W. von Seelen, Application and optimization of neural field dynamics for driver assistance, *in IEEE 4th International Conference on Intelligent Transportation Systems*, pages 309-314. IEEE Press, 2001.

[9] P. Dahm, C. Bruckhoff, and F. Joublin, A neural field approach to robot motion control, *in Proceedings of the 1998 IEEE International Conference On Systems, Man, and Cybernetics*, pages 3460-3465, 1998.

[10] M. Oubbati, M. Schanz Michael, and P. Levi, Neural Fields for Behavior-based Control of Mobile Robots, *in 8th International IFAC Symposium on Robot Control (SYROCO 2006)*, Bologna, Italy, September 2006.

[11] M. Oubbati, and Günther Palm, Neural Fields for Controlling Formation of Multiple Robots, *in 7th IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA2007)*, Florida, USA, June 20-23, 2007.

[12] S. Thrun, D. Fox,W. Burgard, and F. Dellaert, Robust monte carlo localization for mobile robots, *Artificial Intelligence*, 2000.