Reactive navigation through Multiscroll Systems: from theory to real-time implementation

Paolo Arena,* Sebastiano De Fiore,* Luigi Fortuna,* Mattia Frasca,*

Luca Patané,* Guido Vagliasindi *

July 10, 2007

*Dipartimento di Ingegneria Elettrica Elettronica e dei Sistemi, Universitá degli Studi di Catania, viale A. Doria 6, 95125 Catania, Italy. E-mail: [parena, lfortuna, mfrasca, lpatane, sdefiore, gvaglia]@diees.unict.it, Tel: +39-095-7382307, Fax: +39-095-330793.

Abstract

In this paper a new reactive layer for multi-sensory integration applied to robot navigation is proposed. The new robot navigation technique exploits the use of a chaotic system able to be controlled in real-time towards less complex orbits, like periodic orbits or equilibrium points, considered as *perceptive orbits*. These are subject to real-time modifications on the basis of environment changes acquired through a distributed sensory system. The strategy is inspired to the olfactory bulb neural activity observed in rabbits subject to external stimuli. The mathematical details of the approach are given including simulation results in a virtual environment. Furthermore the proposed strategy has been tested on an experimental environment consisting of an FPGA-based hardware driving an autonomous roving robot. The obtained results demonstrate the capability to perform a real-time navigation control.

Keywords

Chaotic systems, multi-sensory integration, navigation, FPGA.

Corresponding author

Luca Patané

Universitá degli Studi di Catania

Dipartimento di Ingegneria Elettrica Elettronica e dei Sistemi

Viale A. Doria, 6 - 95125 Catania

Tel: +39-095-7382307 Fax +39-095330793

Email: lpatane@diees.unict.it

1 Introduction

Machine perception researchers developed a new paradigm which considers perception no longer as a stand-alone process, but as an holistic and synergetic one, tightly connected to the motor and cognitive system [Arkin, 1998]. Perception is now considered as a process indivisible from action: behavioral needs provide the context for the perceptual process, which, in turn, works out the information required for motion control.

Motion control can be used to improve the perceptual process by placing the robot sensors in the most appropriate positions (active perception). This well refines the ability to select the most important information discarding what is less useful for the specific task. Furthermore, traditionally cognitive science divides real world in different categories (called "perceptual classes") used as internal representations or patterns indicating ways to face the environment. Drawing inspiration from perceptual mechanisms of biological systems, machine perception researchers are starting to develop new perception schemes for roving robots. For example, Verschure and co-workers developed a perceptual scheme (Distributed Adaptive Control, DAC5) as a neural model for classical and operant conditioning [Verschure et al., 2003a, Verschure et al., 2003b]. In DAC5 three tightly connected control layers are introduced: the *reactive layer*, the *adaptive layer* and the *contextual layer*. The reactive control layer implements a set of reflexes, based on low level sensorial unconditioned inputs. The adaptive control layer allows the system to associate more complex stimuli with the basic ones. The contextual layer constructs high-level representations by means of memory structures.

All forms of adaptive behavior require the processing of multiply sensory information and their transformation into series of goal-directed actions. In the most primitive animal species the entire process is regulated by external (environmental) and internal feedback through the animal body [Uexku, 1926, Fuster, 2003].

Cortical processes information coming from objects identified in the environment through spike trains from receptors by enrolling dedicated neural assemblies. These are nonlinear dynamical coupled systems whose collective dynamics constitutes the mental representation of the stimuli. Freeman and co-workers, in their long experimental studies on the dynamics of sensory processing in animals

[Freeman, 2003, Freeman, 2004], conceived a “dynamical theory of perception”. The hypothesis is that cerebral activity can be represented by a chaotic dynamics. They attained this result by different experiments on rabbits which inhaled in a pre-programmed way several smells. Through the electroencephalogram (EEG), Freeman evaluated the action potentials in the olfactory bulb and he noticed that the potential waves showed a complex behavior. So he came to the conclusion that an internal mental representation (cerebral pattern) of a stimulus is the result of a complex dynamics in the sensory cortex in cooperation with the limbic system that implements the supporting processes of intention and attention [Freeman, 1991]. More in details, according to Freeman [Freeman, 2004], the dynamics of the olfactory bulb is characterized by a high-dimensional chaotic attractor with multiple wings. The wings can be considered as potential memory traces formed by learning through the animal’s life history. In the absence of sensory stimuli, the system is in a high dimensional itinerant search mode, visiting various “wings”. In response to a given stimulus, the dynamics of the system is constrained to oscillations in one of the wings, which is identified with the stimulus. Once the input is removed, the system switches back to the high-dimensional, itinerant basal mode.

Analyzing the experimental data acquired, Freeman proposed a dynamical model of the olfactory system, called K-sets [Freeman, 1987, Skarda et al., 1987]. The model is able to show all the chaotic and oscillatory behaviours identified through the experiments. Freeman and Skarda [Skarda et al., 1987] discussed on the important role of chaos in the formation of perceptual meanings. Accordingly to their works, neural system activity persists in a chaotic state until sensors perturb this behaviour. The result of this process is that a new attractor emerges representing the meaning of the incoming stimuli. The role of chaos is fundamental to provide the flexibility and the robustness needed by the system during the migration through different perceptual states. A discrete implementation of Freeman’s K model (i.e. KA sets) was developed and applied to navigation control of autonomous agents [Harter et al., 2005a]. The controller parameters have been learned through an evolutionary approach [Harter, 2005b] and also by using unsupervised learning strategies [Harter et al., 2005a]. In this work we are not trying to reproduce bio-relevant model of neural assemblies, in fact our main objective is to propose a reactive control architecture for autonomous robots, taking care of the functional properties discovered by Freeman in the olfactory bulb. The idea

is to use a simple but chaotic dynamical system with suitable characteristics that can functionally simulate the creation of perceptual patterns. The patterns can be used to guide the robot actions and the control system can be easily extended to include a wide number of sensors. Furthermore the control architecture can be implemented at a hardware level in an FPGA-based board to be embedded on an autonomous roving robot [Arena et al., 2006a].

In this paper, taking inspiration from Freeman's experiments showing the presence of chaotic dynamics in neural system activities, and paying attention to the reactive layer in DAC5, we introduce a new control architecture implementing the perception-action loop. We applied the proposed architecture to the navigation control problem.

The perception stage is represented by a suitable chaotic attractor, controlled by incoming signals from sensors. In particular a multidimensional state feedback control strategy has been implemented. A peculiarity of the approach is that, whereas most of the chaos control techniques are focused on the control of the chaotic trajectories towards equilibrium points or native limit cycles, in this case the controlled system is able to converge also to orbits never shown by the uncontrolled system. This is due both to the particular system chosen and to the fact that a multireference control is required. For this reason, we called this technique "weak chaos control" (WCC). The crucial advantages of this approach are the compact representation of the perception system, the real time implementation in view of its application to robot navigation control and the possibility to take into consideration the physical structure of the robot within the environment where it moves. This characteristic is realised since the robot geometry is introduced within the phase space, where chaotic wanderings are represented. Moreover obstacles or target positions are mapped in the phase space directly reflecting their actual position, with respect to the robot, as in the real environment. We have therefore, in the phase space, a kind of mirrored real environment, as measured by the sensors. The emerging controlled orbit will influence the behaviour of the robot by means of suitable actions, gained through the implementation of a simple unsupervised learning phase, that has already been presented in [Arena et al., 2007]. Arena and co-workers developed a complete scheme, devoted not only to model the sensing-perception-action loop, but also to include memory of the past successful behaviours, in order to incrementally form a kind of contextual memory, helpful in case of time invariant environment. In that case the

perception strategy was implemented by using Turing patterns emerging in an array of cellular non-linear networks. In this paper, instead of using a lattice of cells, only a third order nonlinear system, showing a multiscroll attractor, is implemented to face with the perception stage. So in this paper we'll report the details regarding the perception stage. In particular the basic underlying concept, simulation results and experiments on a roving robot will be reported in the manuscript main body, leaving the theoretical aspects that justify the methodology, to the Appendix.

2 The Multiscroll system

In this section the chaotic circuit used as perceptual system is introduced. Since this system should be able to deal with a great number of sensorial stimuli and represent them, a chaotic system, able to generate multiscrolls [L*ü*, 2004], has been adopted. This can be viewed as a generalization of the Chua's double scroll attractor represented through saturated piecewise linear functions and of other circuits able to generate a chaotic attractor consisting of multiply scroll distributed in the phase space (i.e. *n*-scrolls attractor) [Manganaro et al., 1999]. It is able to generate one-dimensional (1-D) *n*-scrolls, two-dimensional (2-D) $n \times m$-grid scrolls or three-dimensional (3-D) $n \times m \times l$-grid scroll chaotic attractors by using saturated function series. In this paper a 2-D multiscroll system has been chosen. It is described by the following differential equations [L*ü*, 2004]:

$$\begin{cases} \dot{x} = y - \frac{d_2}{b} f_1(y; k_2; h_2; p_2, q_2) \\ \dot{y} = z \\ \dot{z} = -ax - by - cz + d_1 f_1(x; k_1; h_1; p_1, q_1) \\ \qquad + d_2 f_1(y; k_2; h_2; p_2, q_2) \end{cases} \tag{1}$$

where the following so-called saturated function series (PWL) $f_1(x; k_j; h_j; p_j, q_j)$ has been used:

$$f_1(x; k_j; h;_j p_j; q_j) = \sum_{i=-p_j}^{q_j} g_i(x; k_j; h_j) \tag{2}$$

where $k_j > 0$ is the slope of the saturated function, $h_j > 2$ is called *saturated delay time*, p_j and q_j are positive integers, and

$$g_i(x;k_j;h_j) = \begin{cases} 2k_j & \text{if } x > ih_j + 1, \\ k_j(x - ih_j) + k_j & \text{if } |x - ih_j| \leq 1, \\ 0 & \text{if } x < ih_j - 1 \end{cases}$$

$$g_{-i}(x;k_j;h_j) = \begin{cases} 0 & \text{if } x > -ih_j + 1, \\ k_j(x + ih_j) - k_j & \text{if } |x + ih_j| \leq 1, \\ -2k_j & \text{if } x < -ih_j - 1 \end{cases}$$

System (1) can generate a grid of $(p_1 + q_1 + 2) * (p_2 + q_2 + 2)$ *scroll attractors*. Parameters p_1 (p_2) and q_1 (q_2) control the number of *scroll attractors* in the positive and negative direction of the variable x (y), respectively. The parameters used in the following ($a = b = c = d_1 = d_2 = 0.7$, $k_1 = k_2 = 50$, $h_1 = h_2 = 100$, $p_1 = p_2 = 1$, $q_1 = q_2 = 2$) have been chosen according to the guidelines introduced in [Lü, 2004] to generate a 2-D 5×5 grid of scroll attractors. An example of the chaotic dynamics of system (1) is given in Fig. 1.

3 Control of the multiscroll system

In our approach the perceptual system is represented by the multiscroll attractor of equations (1), whereas sensorial stimuli can interact with the system through (constant or periodic) inputs that can modify the internal chaotic behavior. Since one of the main characteristics of perceptive systems is that sensorial stimuli strongly influence the spatial-temporal dynamics of the internal state, a suitable scheme to control the chaotic behavior of the multiscroll system on the basis of sensorial stimuli should be adopted.

Briefly, chaos control refers to a process wherein a tiny perturbation is applied to a chaotic system in order to realize a desirable behavior (e.g. chaotic, periodic and others). Several techniques have

been developed for the control of chaos [Boccaletti et al., 2000].

In view of our application, a continuous-time technique like the Pyragas's method is a suitable choice [Pyragas, 1992].

In this method [Pyragas, 1992, Pyragas, 1993] the following model is taken into account:

$$\frac{dy}{dt} = P(y,x) + F(t), \qquad \frac{dx}{dt} = Q(y,x) \tag{3}$$

where y is the output of the system (i.e. a subset of the state variables) and the vector x describes the remaining state variables of the system. $F(t)$ is the additive feedback perturbation which forces the chaotic system to follow the desired dynamics. Pyragas [Pyragas, 1992, Pyragas, 1993] introduced two different methods of permanent control in the form of feedback. In the first method, that is used here, $F(t)$ assumes the following form:

$$F(t) = K[\widehat{y}(t) - y(t)] \tag{4}$$

where $\widehat{y}$ represents the external input (i.e. the desired dynamics), and K represents a vector of experimental adjustable weights (adaptive control).

The method can be employed to stabilize the unstable orbits endowed in the chaotic attractor reducing the high order dynamics of the chaotic system.

3.1 Control scheme

In our case a strategy based on equations (4) has been applied. The desired dynamics is provided by a constant or periodic signal associated with the sensorial stimuli. Since more than one stimulus can be presented at the same time, the Pyragas method has been generalized to account for more than one external forcing.

Hence, the equations of the controlled multiscroll system can be written as follows:

$$\begin{cases} \dot{x} = y - \frac{d_2}{b} f_1(y;k_2;h_2;p_2,q_2) + \sum_i k_{x_i}(x_{r_i} - x) \\ \dot{y} = z + \sum_i k_{y_i}(y_{r_i} - y) \\ \dot{z} = -ax - by - cz + d_1 f_1(x;k_1;h_1;p_1,q_1) \\ \qquad + d_2 f_1(y;k_2;h_2;p_2,q_2) \end{cases} \tag{5}$$

where i is the number of external references acting on the system; x_{r_i}, y_{r_i} are the state variables of the reference circuits that will be described in details below and k_{x_i}, k_{y_i} represent the control gains. It can be noticed that the control acts only on the state variables x and y, and this action is sufficient for the proposed navigation control strategy. The complete control scheme is shown in Fig. 2.

Each reference signal (x_{r_i}, y_{r_i}) can be a constant input or a periodic trajectory representing a native cycle. This can be generated using the multiscroll system (1) with particular parameters ($a = b = c = 1$). In this case the number of multiscroll systems needed to generate the reference cycles should be the same as the number of reference trajectories required. In a more simple way, these reference signals can be built using sinusoidal oscillators:

$$\begin{aligned} x_r(t) &= A_{x_r} sin(\omega_{xr} t - \varphi_{x_r}) + x_{off} \\ y_r(t) &= A_{y_r} sin(\omega_{yr} t - \varphi_{y_r}) + y_{off} \end{aligned} \tag{6}$$

where (x_{off}, y_{off}) is the center of the reference cycle, ω is the frequency (in this work $\omega_{xr} = \omega_{yr} = 1$), φ_{x_r} and φ_{y_r} are the phases, A_{x_r} and A_{y_r} define the amplitude of the reference signal.

A theoretical analysis on the emerging behaviours of the controlled multiscroll system is reported in Appendix A.

4 Multiscroll control for robot navigation control

In the following sections the proposed reactive control scheme will be applied to robot navigation control. Taking inspiration from Freeman's works, we adopt a chaos control approach to enslave the chaotic trajectories of our perceptual system towards different pseudo-periodic orbits. We chose to

use, as reference signals, periodic inputs, even if the method can be used also considering constant references. To this aim, taking into account a single reference periodic signal, the control gain range is defined in the following way:

$$k_x, k_y \geq k_{min}(x_r, y_r) \tag{7}$$

If the reference signal is a native orbit of the chaotic system, it can be shown that $k_{min} = 0.534$ (see Appendix A).

Below k_{min} the system presents a chaotic behavior, whereas if control gains are above k_{min} a limit cycle behavior occurs. In particular, as concerns the control by using a single reference dynamics, for low values of k_x and k_y (as shown in Fig. 3(a)) the control of the multiscroll attractor has a residual error, however for the purpose of navigation control this weak condition is still acceptable. For higher values of k_x and k_y (as shown in Fig. 3(b)), the steady error approaches zero.

One of the most interesting aspects of this technique, applied to the multiscroll system into consideration and useful for robot control purposes, is evident when there are more than one external reference. For example, let us consider the case in which there are two concurrently active inputs, and so there are two reference signals in the multiscroll phase plane. If the control gains of the two reference systems are not equal, the resulting controlled limit cycle (*emerged cycle*) will be placed, in the phase plane, near the reference cycle associated with the higher control gain. If the two reference dynamics have the same control gains, the resulting cycle will be placed exactly at halfway between them. These results are shown in Fig. 4. When stimuli are perceived, the system converges to a limit cycle that constitutes a representation of the concurrent activation of the sensorial stimuli. When stimuli are no longer active, the multiscroll returns to its default chaotic dynamics. An example of this process is shown in Fig. 5.

In the next sections a reactive system applied to the navigation control of a roving robot is proposed. For the theoretical aspects that justify the whole approach, please refer to the Appendix A.

5 Robot navigation

When a robot is placed in an unknown environment, it is subject to a huge amount of external stimuli. To explore the area avoiding obstacles, the robot, sensing the environment, can create an internal representation of the stimuli in relation to its body. The loop is closed by an action that is chosen to accomplish a given target behavior (e.g. exploration with obstacle avoidance). In the previous section we have defined a structure based on dynamical systems that can be involved in realizing a reactive navigation layer. It is based on a chaos control technique, used to enslave the dynamics of a multiscroll attractor to follow one or more reference trajectories. In this paper, every sensor equipped on the robot provides a reference cycle. This is addressed by associating for instance, the perception of an obstacle with a stimulus and associating it with a representation (pattern). Therefore the controlled multiscroll system is the perceptual system and the emerged orbit stands for the internal representation of the external environment. Finally, according to the characteristic of the emerged cycle (amplitude, frequency, center position) an action, in terms of speed and rotation angle, is associated. In this work, action is linked to perception (the emerged cycle) using a deterministic algorithm. However this association can be obtained through a bio-inspired adaptive structure and the classical Motor Map paradigm could represent a good candidate [Ritter et al., 1992]. Indeed, Motor Map artificial networks are suitable to control the robot navigation in an unknown environment because they are adaptive and unsupervised structures [Arena et al., 2005a].

Among the 5×5 possible reference cycles which can be used, in the application here reported, only a subset of them has been effectively used. Their distribution in the phase plane $x - y$ reflects the topological distribution of robot sensors. Fig. 6 shows a simulated robot equipped with four distance sensors and a target sensor. The corresponding reference cycles reported in the phase plane are related to the sensor positions. Distance sensors are directional and are associated each one to a single reference cycle, whereas target sensor is characterized by an omnidirectional field of view and for that reason it is associated to more than one (i.e four) reference cycles [Arena et al., 2005b]. The offsets assigned to each input cycle are defined to match the position of the scroll centers. Moreover in this work we have chosen to link the value of the control gains with the intensity of perceived

sensorial stimuli.

The technique, based on placing reference cycles in the phase plane in accordance with the distribution of sensors on the robot, is important to strictly connect the internal representation of the environment to the robot geometry. In our tests only distance and target sensors have been used, although other sensors could be included.

Distance sensors have a visibility range that represents the area where the robot is able to detect static and dynamic obstacles, whereas target sensor returns the target angular displacement with respect to the front axis of the robot, when the robot is inside the detection region of the target.

As concerns the action (in terms of absolute value of speed and heading) performed by the robot, it depends on the multiscroll behavior. In particular, when no stimulus is perceived (i.e. there are no active sensors) the system evolves chaotically and the robot continues to explore the environment moving with constant speed and without modifying its orientation. Moreover, another possible exploring strategy can be taken into consideration: the robot can exploit the chaotic wandering of its internal state variables to generate and follow a chaotic trajectory that can help the system during the environment exploration. When external stimuli are perceived, the controlled system converges to a cycle (i.e. a periodic pattern) that depends on the contribution of active sensors through the control gains k_{x_i} and k_{y_i}. The action that will be executed is chosen according to the characteristics of the cycle, in particular its position in the phase plane. A vector pointing to the center of the limit cycle of the controlled multiscroll attractor is defined; predefined actions are chosen on the basis of module and orientation of this vector. When the stimuli stop, the multiscroll returns to evolve in a chaotic way.

A different strategy has been adopted for the target. When a target is in the detection range of the robot, it is considered as an obstacle located in a position symmetric with respect to the motion direction. This is associated with a reference cycle which controls the multiscroll attractor with a low gain, so that avoiding obstacles has priority over reaching targets. In this way the generated reference cycle has the task to weakly suggest a rotation towards the target, since preserving the robot safety is considered more important than retrieving a target.

To summarize, the weak chaos control technique is employed to implement the reactive layer

of a sensing-perception-action scheme. Moreover the control framework can be extended adding another important layer to create a plastic association between perceptual states and actions that can be modulated, during an unsupervised learning phase, by using a reward-based structure.

6 Simulation results

To test the performance and the potential impact of the proposed architecture we have developed a software tool for mobile robot simulations and a hardware implementation in an embedded platform. The first evaluation stage was carried out via a 3D simulation environment [Arena et al., 2005b]. Fig. 7 shows the user interface of the simulator. This is characterized by four sub-windows used to modify the control parameters and to monitor the behavior of the controlled system. A robot model involved in a food retrieval task was simulated.

To evaluate the performance of the proposed control scheme, a comparison with a traditional navigation control method is reported. The navigation strategy chosen as benchmark is the Potential Field (PF) [Khatib, 1986]. A basic version using a quadratic potential has been implemented in the simulator for the same simulated robot used to test the weak chaos control approach [Beard et al., 2003]. Also in this case the robot can use only local information, acquired from its sensory system to react to the environment conditions (i.e. local PF). The parameters of the PF algorithm (e.g. robot speed, constraints for the movements) have been chosen in order to allow a comparison with the WCC technique.

The sensory system of the simulated roving robot consists of four distance sensors and a target sensor as depicted in Fig. 6. Several environmental configurations were considered; here we report the results for two different scenarios shown in Fig.8. Four targets were introduced in both the environments: the circle around each target represents the range in which the target can be sensed. Obstacles are represented by walls and by the black rectangles. The robot had to navigate in this environment reaching the targets, while avoiding obstacles. When a target is found, it is disabled, so that the robot navigates toward the other targets.

For each environment we performed a set of five simulations for the different control methods

taken into consideration. In particular we compared the navigation capabilities of the robot controlled through the local potential field method and through two versions of the WCC architecture. The difference between the two versions is limited to the behaviour of the robot during the exploration phase (i.e. when no stimuli are perceived). The former implements a very simple behaviour that consists into a forward movement with the speed set to its maximum value (i.e. WCC_f), whereas the latter considers the chaotic evolution of the multiscroll system to determine the action of the robot exploring the environment (i.e. WCC_c). Here when no stimuli are perceived, the perceptual core of the control system behaves chaotically and the robot action depends on the position of the centroid of the chaotic wandering shown by the system during the simulation step. Each simulation step corresponds to a single robot action and it is determined simulating the dynamical system for 2000 steps with an integration step equal to 0.1 [Arena et al., 2005b]. An example of the trajectories followed by the robot in the three cases is shown in Fig. 9.

For each simulation the robot is randomly placed in the environment and the three control methods are applied monitoring the robot behaviour for 10000 actions (i.e. epoches). To compare the performances of the algorithms we consider the cumulative number of targets found and the area explored by the robot [Harter et al., 2005a].

In Fig. 10 the cumulative number of targets found in the two environments, calculated in time windows of 1000 epochs, is shown. The performances of the three control methods are comparable in both the environments.

Another performance index taken into consideration is the area covered by the robot during each simulation. The results shown in Fig. 11 demonstrate that the WCC_c guarantees a high exploration capabilities with respect to the other control methods.

Movies of these simulations are available on the web [Arena et al., 2006b].

7 Hardware Tests

One of the key points of the approach is the feasibility of hardware implementation of the control architecture. A roving robot equipped with four distance sensors and controlled by an FPGA is used

to test the navigation control algorithm. In the experiment here proposed only distance sensors have been used and the interest has been devoted to the obstacle avoidance behavior. Considering their position and orientation (Fig. 12), sensor S_i is associated to the cycle Ref_i. Sensors S3 and S4 have been configured in order to have a limited activation range with respect to the others. In this way the robot could pass through narrow spaces.

7.1 Hardware implementation of the control scheme

Since the proposed method for robot navigation showed good results in a simulation environment the next step was the hardware implementation. The device devoted to control the whole system is an FPGA (Stratix II EP2S60 by Altera) which integrates a soft embedded processor, the NiosII. This solution was selected in order to combine in a single device both the hardware configurability of FPGA and the easy programmability of microcontrollers. NiosII is, in fact, a 32-bit RISC digital microprocessor whose clock frequency is 50 MHz. It is a configurable "soft-core" processor, so its functionalities in terms of number of peripherals and type or amount of memory, can be easily modified according to the specific task.

Simulation and control of the multiscroll system are made directly in VHDL (Very High Speed Integrated Circuits Hardware Description Language) implementing a fourth order Runge-Kutta algorithm (RK4)[Lapidus et al., 1971]. Other implementations were tested before the VHDL one. The first step was to simulate the chaotic system directly in the NiosII data path using a C code algorithm. The results were correct but the performances were not suitable for a real-time application on a roving robot. The high simulation time on NiosII (see Tab. 1) is due to the fact that its ALU (Arithmetic Logic Unit) does not support natively the floating point operations. This problem can be overcome using custom instructions. The "Soft-core" nature of the NiosII processor enables designers to integrate custom logic (e.g. written in VHDL language) into the ALU. Similar to native NiosII instructions, custom instruction logic can take values from up to two source registers and optionally write back a result to a destination register. Also in this case the results were correct but the execution time, although decreased of one order of magnitude, remained too high for a real-time application.

So, only the VHDL implementation is suitable as shown in Tab. 1.

Since the weak chaos control is implemented directly in hardware, the role of the NiosII processor is limited to drive the activity of the block that implements the RK4 algorithm, reading the sensor values, calculating the control gains and executing the navigation algorithm. Indeed, it simply provides the start-up input and reads the results necessary to identify the behavior of the multiscroll system at the end of the integration process. These information are used by the processor to calculate robot actions.

For the implementation of the dynamical system operations, with real numbers are necessary. This algebra is not supported by the standard VHDL library, so the first step was to develop a dedicated library. Since the fixed point operations are faster than the floating point ones, the fixed point arithmetic has been adopted. The total number of bits to code variables in VHDL code is 24. In particular, the most significant bit represents the sign, 13 bits were used to code the integer part and 10 bits to code the fractional part.

The top level entity *sim_full* is shown in Fig. 13. It is composed of two sub-entities:

- The *sim* block implements the simulation of the controlled multiscroll system adopting the RK4 algorithm with fixed integration step.
- The *sim_machine* implements a finite state machine (FSM) that controls the activity of the *sim* entity.

NiosII drives the *sim_full* entity through the input signals and provides the gains that are necessary to control the multiscroll system. Subsequently, it reads the results of the computation provided by the entity.

The *sim* entity (Fig. 14) consists of different components: the *rk4*, the *ram* and the *control* VHDL entities. The function of the *rk4* entity is to implement the Runge-Kutta algorithm and it needs the state vector (x, y, z) and the control gain in order to perform a simulation. The function of the other inputs (i.e. *Clk_rk4*, *Start_rk4* and *Reset_rk4*) is to coordinate the activity of the entity.
The *ram* block is a memory that stores all the results of the integration: *di* is the data input, *do* is

the data output, *dn* is the last sample produced and memorized, *we* is a write enable input and *addr* is necessary to address the memory. The other outputs of this block represent the maximum and the minimum of the *x* and *y* state variable. This is an important feature that permits to determine the characteristics of the emergent cycle and to allow the use of the *sim_full* entity for the following action selection step. The function of the *control* entity is to calculate the control signals, while the k_{xi} and k_{yi} inputs represent the control gains corresponding to the reference *i*. The reference signals are stored in a read only memory inside the *control* block.

Fig. 15 represents an example of the evolution of the controlled multiscroll system implemented in FPGA. The hardware results are congruent with the simulation ones, so the fixed point algebra and the *sim_full* VHDL entity are suitable for implementing the weak chaos control.

7.2 Experimental results

In this section some experiments on a real environment are shown. The experimental platform is constituted by the FPGA that implements the navigation algorithm, a mobile robot and a wireless communication system (Fig. 16). A computer can be used to make a backup of the data for a post process analysis.

The robot used for the experimental set-up is a Lynxmotion 4WD2 Robot. It is a classic four wheel drive rover controlled through a differential drive system. The robot is equipped with 4 DC 50:1 gear head motors and with an RF04 USB radio telemetry module for remote control operations and sensor acquisition. It is also equipped with four infrared distance sensors GP2D12 from Sharp, having a detection distance of up to 80 cm, mounted on the front part, as in the scheme of Fig. 17.

The wireless communication system is used to interface the FPGA with the robot in a bidirectional way, to read sensors and to drive motors. It is based on the ER400TRS radio transceiver that provides a serial interface with the host (FPGA in this work). The NiosII microprocessor is devoted to the execution of the deterministic navigation algorithm and to the supervision of the activity of the VHDL entity implementing the weak chaos control. Indeed, once the NiosII reads sensors it starts the *sim_full* VHDL entity. When the simulation ends, the NiosII reads the parameters that identify the

emerged pattern. Then it calculates the command to drive the roving robot (Fig. 17). The simulation process implemented in the *sim_full* entity lasts about 2.8 ms and the control algorithm running on NiosII about 80 ms.

An experiment in which the robot faces with a complex environment is shown in Fig. 18 together with the respective evolution of the controlled multiscroll system. Fig. 18 (a) and Fig. 18 (b) show a sensor output; high value means a low distance from an obstacle. The sample time is about 0.350 s. In Fig. 18.c the robot senses an obstacle on the right side; both sensors S1 and S4 are activated but in different way: the control gain associated with S4 is higher than the other gain. Therefore, the resulting cycle is placed between Ref1 and Ref4 but closer to Ref4 (Fig. 18 (d)). In this case the robot turns in the opposite direction, and the speed and rotation angle depend on the characteristics of the emerged cycle. Subsequently, the sensors do not see any obstacle (Fig. 18 (e)), so the multiscroll system shows a chaotic behavior (Fig. 18 (f)) and the robot continues to explore the environment, moving with constant speed and without modifying its orientation. In the last picture, Fig. 18 (g), only S1 sensor is slightly activated, so the emerged cycle is placed on the reference Ref1 although a small control gain is used (Fig. 18 (h)).

Another experiment, whose video is reported in [Arena et al., 2006c], shows the robot in an arena. It continuously explores the environment until it founds an exit.

As performed for the simulated robot, also with the real one we made a series of tests in two different environments (Fig. 19). Five experiments for each arena have been carried out placing the robot in a random initial position. An example of the trajectory followed in each environment by the robot controlled through the WCC_f method is shown in Fig. 19.

In each experiment the robot explores the environment for 7 minutes. The behaviour of the robot has been recorded through a video camera. The movies, available in [Arena et al., 2006c], have been used to extract the robot trajectory and to evaluate the explored area. The results obtained in the two arenas are shown in Fig. 20. In this case the WCC_f algorithm was selected to be experimentally shown. From the analysis of Fig. 20 the capability of the algorithm to densely explore the environment in a few minutes can be appreciated.

8 Conclusions

In this paper the problem of multi-sensory integration has been treated using a new technique called *weak chaos control*. This approach takes inspiration from the Freeman's theory of brain pattern formation, although it makes use of a more abstract model, and has been applied to a navigation control problem.

The phenomenon of encoding information stabilizing the unstable orbits endowed in a chaotic attractor has been investigated. The chaotic system, recently proposed in [L*ü*, 2004], was chosen for its simple model and for the possibility to extend the emerging multiscroll attractor to one, two and three dimensions varying also the number of scrolls.

The feedback from the environment has been introduced by using a continuous multi-reference chaos control technique based on the Pyragas' method. The mathematical analysis of the controlled system has been reported in the Appendix A, extending the discussion for the double scroll to a general *nxm* scrolls configuration. The analytical study has been exploited to develop a reactive navigation layer for a roving robot. The robot behavior during a food retrieval task has been evaluated in a 3D simulation environment.

Finally, an FPGA architecture has been chosen as a suitable device for a real-time implementation of the weak chaos control technique in a hardware board. The characteristics that make the FPGA an optimal solution for our purposes are the flexibility of a reprogramable hardware and the high computational power obtained with a parallel processing. Furthermore the FPGA contains a soft embedded processor that has been used to realize a high level controller that supervises and synchronizes the processes. The navigation control strategy has been evaluated by using a differential drive rover equipped with distance sensors. The results demonstrate the capability of the system that is able to explore different environments avoiding deadlock situations and finding the escape way from difficult environmental constraints.

The proposed reactive layer based on enslaving a chaotic attractor in periodic dynamics can be integrated in a complete architecture for the sensing-perception-action loop as already presented in [Arena et al., 2007], in which an adaptive and a contextual layer were developed to form a complete

action-oriented perception methodology, based on complex dynamical system control. In particular the adaptive layer is used to create a plastic association between perceptual states and actions that can be modulated, during an unsupervised learning phase, by using a reward-based structure. Furthermore a contextual layer including short term and long term memory can be added to improve the robot navigation skills.

Appendix A: Analysis of the controlled multiscroll system

The control of the multiscroll system has been performed by using the proposed method called weak chaos control. In this section, stability conditions are derived for the controlled Double Scroll, (1-D) n-scrolls and (2-D) $n \times m$-grid scrolls reported in [L*ü*, 2004]. The global stability of the controlled system is here discussed. Let us first sketch the steps followed in the demonstration of the results and then introduce the mathematical details.

Step 1) Since the chaotic system is characterized by a feedback nonlinear PWL function, it is possible to analyze its behavior in each linear (affine) subspace and therefore to find, for each one, the fixed point and its stability. It will be shown that both stability of equilibria and their physical position in the phase space depend on the control gain.

Step 2) The second step lies in finding suitable values of the control gain vector for which all the equilibria relative to the dynamic behavior of the system in each linear (indeed affine) subspace are both asymptotically stable and all of them except for the origin are virtual [Chua et al., 1987, Manganaro et al., 1999], i.e. they do not belong to the linear region the corresponding linear dynamic belongs to.

Step 3) Once these conditions are found, the origin results to be the only attractor of the system; in fact the other virtual equilibria act so as to attract the system trajectories towards the only stable attractor. Therefore in these conditions the controlled system is globally asymptotically stable. Other conditions allow the presence of multiple stable equilibria on different subspaces. The analysis will be given together with simulation results.

Controlled Double Scroll

The controlled Double Scroll system, considering a single reference signal, is described by:

$$\begin{cases} \dot{x} = y + k_x(x_r - x) \\ \dot{y} = z + k_y(y_r - y) \\ \dot{z} = -ax - by - cz + d_1 f_0(x;k_1) \end{cases} \tag{8}$$

where $a = b = c = d_1 = 0.7, k_1 = 10$, $k_x = k_y = k$ are the control gains and $f_0(x;k_1)$ is a simple PWL function defined by:

$$f_0(x;k_1) = \begin{cases} k_1 & \text{if } x > 1 \\ k_1 x & \text{if } |x| \leq 1 \\ -k_1 & \text{if } x < -1 \end{cases} \tag{9}$$

For the sake of simplicity only a couple of reference signals are considered. Defining a vector state as in eq. (11), the system (8) can be written in a compact form:

$$\dot{\mathbf{x}} = \mathbf{f(x)} \tag{10}$$

with

$$\mathbf{x} = [x, y, z]^T \tag{11}$$

As suggested in [L*ü*, 2004], eq. (8) has three different subspaces:

$$\begin{aligned} V_1 &= \{\mathbf{x} \in \Re^3 : x \geq 1\} \\ V_2 &= \{\mathbf{x} \in \Re^3 : |x| \leq 1\} \\ V_3 &= \{\mathbf{x} \in \Re^3 : x \leq -1\} \end{aligned} \tag{12}$$

Restricted to each of the three subspaces, system (8) is a linear third order dynamical system.

Into subspaces V_1 and V_3 the dynamical equations become:

$$\begin{cases} \dot{\overline{x}} = -k_x\overline{x} + \overline{y} + k_x\overline{x}_r \\ \dot{\overline{y}} = -k_y\overline{y} + \overline{z} + k_y\overline{y}_r \\ \dot{\overline{z}} = -a\overline{x} - b\overline{y} - c\overline{z} \end{cases} \tag{13}$$

with

$$[\overline{x},\overline{y},\overline{z}]^T = [x - \frac{d_1k_1}{a}, y, z]^T \text{ for } x \in V_1$$

$$[\overline{x}_r,\overline{y}_r] = [x_r - \frac{d_1k_1}{a}, y_r] \text{ for } x \in V_1$$

$$[\overline{x},\overline{y},\overline{z}]^T = [x + \frac{d_1k_1}{a}, y, z]^T \text{ for } x \in V_3$$

$$[\overline{x}_r,\overline{y}_r] = [x_r + \frac{d_1k_1}{a}, y_r] \text{ for } x \in V_3$$

The characteristic equation is:

$$p(\lambda) = |A - \lambda I| = \lambda^3 + \lambda^2 c_{V_1} + \lambda b_{V_1} + a_{V_1} \tag{14}$$

where:

$$\begin{aligned} c_{V_1} &= k_x + k_y + c \\ b_{V_1} &= k_xk_y + c(k_x + k_y) + b \\ a_{V_1} &= bk_x + ck_xk_y + a \end{aligned} \tag{15}$$

The eigenvalues are:

$$\begin{aligned}
\lambda_1 &= -\frac{c_{V_1}}{3} + \sqrt[3]{-\frac{q_{V_1}}{2} + \sqrt{\Delta_{V_1}}} + \sqrt[3]{-\frac{q_{V_1}}{2} - \sqrt{\Delta_{V_1}}} \\
\lambda_{2,3} &= -\frac{c_{V_1}}{3} - \frac{1}{2}\left(\sqrt[3]{-\frac{q_{V_1}}{2} + \sqrt{\Delta_{V_1}}} + \sqrt[3]{-\frac{q_{V_1}}{2} - \sqrt{\Delta_{V_1}}}\right) \\
&\pm \; j\frac{\sqrt{3}}{2}\left(\sqrt[3]{-\frac{q_{V_1}}{2} + \sqrt{\Delta_{V_1}}} - \sqrt[3]{-\frac{q_{V_1}}{2} - \sqrt{\Delta_{V_1}}}\right) \\
&= \alpha_{V_1} \pm i\beta_{V_1}
\end{aligned}$$

where

$$\begin{aligned}
\Delta_{V_1} &= \frac{a_{V_1}c_{V_1}^3}{27} - \frac{b_{V_1}^2 c_{V_1}^2}{108} - \frac{a_{V_1}b_{V_1}c_{V_1}}{6} + \frac{b_{V_1}^3}{27} + \frac{a_{V_1}^2}{4} \\
q_{V_1} &= \frac{2}{27}c_{V_1}^3 - \frac{1}{3}b_{V_1}c_{V_1} + a_{V_1}
\end{aligned}$$

In Fig. 21 the behavior of the eigenvalues for $k \in [0,30]$ is shown. The system is stable for $k \geq 0.105$ and the eigenvalues become purely real for $k \geq 2.9$.

For subspace V_2 the system dynamics is described by the following equations:

$$\begin{cases}
\dot{x} = -k_x x + y + k_x x_r \\
\dot{y} = -k_y y + z + k_y y_r \\
\dot{z} = -(a - d_1 k_1)x - by - cz
\end{cases} \tag{16}$$

The characteristic equation is:

$$p(\lambda) = |A - \lambda I| = \lambda^3 + \lambda^2 c_{V_2} + \lambda b_{V_2} + a_{V_2} \tag{17}$$

where:

$$\begin{aligned}
c_{V_2} &= k_x + k_y + c \\
b_{V_2} &= k_x k_y + c(k_x + k_y) + b \\
a_{V_2} &= bk_x + ck_x k_y + a - d_1 k_1 = a_{V_1} - d_1 k_1
\end{aligned} \tag{18}$$

The eigenvalues are:

$$\begin{aligned}
\lambda_1 &= -\frac{c_{V_2}}{3} + \sqrt[3]{-\frac{q_{V_2}}{2} + \sqrt{\Delta_{V_2}}} + \sqrt[3]{-\frac{q_{V_2}}{2} - \sqrt{\Delta_{V_2}}} \\
\lambda_{2,3} &= -\frac{c_{V_2}}{3} - \frac{1}{2}\left(\sqrt[3]{-\frac{q_{V_2}}{2} + \sqrt{\Delta_{V_2}}} + \sqrt[3]{-\frac{q_{V_2}}{2} - \sqrt{\Delta_{V_2}}}\right) \\
&\pm \; j\frac{\sqrt{3}}{2}\left(\sqrt[3]{-\frac{q_{V_2}}{2} + \sqrt{\Delta_{V_2}}} - \sqrt[3]{-\frac{q_{V_2}}{2} - \sqrt{\Delta_{V_2}}}\right) \\
&= \alpha_{V_2} \pm i\beta_{V_2}
\end{aligned}$$

where:

$$\begin{aligned}
\Delta_{V_2} &= \frac{a_{V_2}c_{V_2}^3}{27} - \frac{b_{V_2}^2 c_{V_2}^2}{108} - \frac{a_{V_2}b_{V_2}c_{V_2}}{6} + \frac{b_{V_2}^3}{27} + \frac{a_{V_2}^2}{4} \\
q_{V_2} &= \frac{2}{27}c_{V_2}^3 - \frac{1}{3}b_{V_2}c_{V_2} + a_{V_2}
\end{aligned}$$

In Fig. 22 the behavior of the eigenvalues for $k \in [0, 30]$ is shown. The system is stable for $k \geq 2.539$.

Equilibrium points analysis: the fixed points are obtained by solving the nonlinear algebraic equations:

$$\mathbf{f(x)=0} \tag{19}$$

Depending on the choice of the parameters k_x and k_y, some or all of the candidate solutions of eq. (19) can be real or virtual fixed points. A virtual fixed point is an equilibrium point that is outside of the domain subspace (i.e. V_1, V_2 or V_3 regions) in which the system (8) has been restricted. Virtual fixed points give as much information on the dynamics as the real equilibria ones.

For the three subspaces of eq. (12) the candidate fixed points are respectively:

$$P_{V_1} = (X_{V_1}, Y_{V_1}) = \tag{20}$$

$$\left(\frac{k_x(x_r - \frac{d_1 k_1}{a})(b + ck_y) + ck_y y_r}{a + bk_x + ck_y k_x} + \frac{d_1 k_1}{a}, k_x(X_{V_1} - x_r + \frac{d_1 k_1}{a})\right)$$

$$P_{V_2} = (X_{V_2}, Y_{V_2}) = \tag{21}$$

$$\left(\frac{k_x x_r(b+ck_y)+ck_y y_r}{a+bk_x+ck_y k_x-k_1 d_1}, k_x(X_{V_2}-x_r)\right)$$

$$P_{V_3}=(X_{V_3},Y_{V_3})= \tag{22}$$

$$\left(\frac{k_x(x_r+\frac{d_1k_1}{a})(b+ck_y)+ck_y y_r}{a+bk_x+ck_y k_x}-\frac{d_1k_1}{a}, k_x(X_{V_1}-x_r-\frac{d_1k_1}{a})\right)$$

Case of zero input Evaluating equations (20-22) for $x_r=y_r=0$, it results that the equilibrium points for subspaces V_1 and V_3 are real points for $k<2.539$, they are virtual for larger control gain k. In particular, when the equilibrium points are virtual, they attract the trajectories starting from V_1 or V_3 towards V_2, where we have the only real fixed point (the origin). Fig. 23 shows the behavior of the fixed points for $x_r=y_r=0$ and $k\in[0,1000]$.

As a result, the whole controlled Double Scroll system in eq. (8) with $x_r=y_r=0$, is globally asymptotically stable for $k\geq 2.539$ and the origin is the unique real fixed point. For $0.105\leq k<2.539$ the origin is an unstable focus and the fixed point of the whole system belongs to the subspaces V_1 or V_3 dependently on the initial conditions (Tab. 2).

Case of constant input In this case the equilibria of V_1 (20) are real if:

$$X_{V_1}>1 \Longrightarrow Y_r>mX_r+q \tag{23}$$

where:

$$\begin{aligned} m &= -\frac{k_x(k_y c+b)}{k_y c} \\ q &= \frac{a_{V_1}-d_1k_1}{k_y c} \end{aligned} \tag{24}$$

The term q is zero for $k=2.539$.

The fixed point of V_2 (21) is real when:

$$\begin{aligned} X_{V_2} &< 1 \Longrightarrow Y_r < mX_r + q \\ X_{V_2} &> -1 \Longrightarrow Y_r > mX_r - q \end{aligned} \tag{25}$$

The equilibrium points of V_3 (22) are real if:

$$X_{V_3} < -1 \Longrightarrow Y_r < mX_r - q \tag{26}$$

Therefore there are three sets (Fig. 24):

$$\begin{aligned} A &= \{(x_r, y_r) \in \Re^2 : y_r > mx_r + q\} \\ B &= \{(x_r, y_r) \in \Re^2 : y_r \leq mx_r + q \wedge y_r \geq mx_r - q\} \\ C &= \{(x_r, y_r) \in \Re^2 : y_r < mx_r - q\} \end{aligned} \tag{27}$$

and so the behavior of the system (8) for constant input is:

$$\begin{aligned} (x_r, y_r) \in A &\Longrightarrow \text{ unique equilibrium point } \in V_1 \\ (x_r, y_r) \in B &\Longrightarrow \text{ unique equilibrium point } \in V_2 \\ (x_r, y_r) \in C &\Longrightarrow \text{ unique equilibrium point } \in V_3 \end{aligned}$$

with

$$\begin{aligned} A \cup B \cup C &= \Re^2 \\ A \cap C &= \emptyset \\ B &\neq \emptyset \end{aligned} \tag{28}$$

The conditions (28) are true for $q \geq 0$ and then for $k \geq 2.539$. When $q < 0$ the new conditions are:

$$\begin{aligned} D &= A \cap C \neq \emptyset \\ B &= \emptyset \end{aligned} \tag{29}$$

Therefore we can consider three distinct cases:

$$\begin{aligned} (x_r, y_r) &\in A \setminus D \Longrightarrow \text{ unique equilibrium point } \in V_1 \\ (x_r, y_r) &\in C \setminus D \Longrightarrow \text{ unique equilibrium point } \in V_3 \\ (x_r, y_r) &\in D \Longrightarrow \text{ two real equilibrium points } \in V_1, V_3 \end{aligned}$$

The results are the same as for the null input case. For $k < 2.539$ there are one or two real equilibrium points: the system converges to one of them dependently on the control inputs; for $k \geq 2.539$ the fixed point is unique. Therefore reference inputs work as biases for the system, moving the equilibrium points within the state space.

Fig. 25 shows the behavior of the system with initial conditions randomly distributed between $[-10, 10]$, control gain $k = 3$ (Fig. 25 (a)) and $k = 1$ (Fig. 25 (b)) and with $x_r = -2$, $y_r = 2$. In the same picture the position of the input in the sets *A*,*B*,*C* and *D* is shown.

Remark From the previous discussion it is important to notice that the behavior of the system depends on the value of the reference signals. A specific case that we want analyzing is the behavior of the controlled double scroll when the reference signals coincide with one of the scroll centers.

The position of the scroll centers, $(-\frac{d_1 k_1}{a}, 0)$ and $(\frac{d_1 k_1}{a}, 0)$, is included in sets *A* and *C* respectively when:

$$\begin{cases} y_r = mx_r + q \\ x_r = -\frac{d_1 k_1}{a} \\ y_r = 0 \end{cases} \qquad \Longrightarrow k_x = k_y \approx 0.534 \tag{30}$$

$$\begin{cases} y_r = mx_r - q \\ x_r = +\frac{d_1 k_1}{a} \\ y_r = 0 \end{cases} \qquad \Longrightarrow k_x = k_y \approx 0.534 \tag{31}$$

Therefore, the system converges to the centers for $k \geq 0.534$ for any initial condition.

Table 3 summarizes the results.

Case of periodic input The behavior of the controlled system with sinusoidal inputs (6) is evaluated by using extensive computer simulations.

The parameters used in eq. (6) are:

$$\begin{aligned} A_{x_r} &= A_{y_r} = A \in [0,25] \\ \omega_{xr} &= \omega_{yr} = \omega \in [0,40] \\ \varphi_{x_r} &= 0^\circ \\ \varphi_{y_r} &= 90^\circ \end{aligned}$$

The simulation results are reported in Fig. 26. The minimum value for the control gain k in order to obtain an input/output error less than 20% is plotted as a function of the sinusoidal input amplitude and frequency. Such error threshold is chosen since it is enough to determine the behavior of the controlled system and this is the information requested for the proposed application.

It is also worth noting that the behavior is similar to the previous exposed cases, although when the reference is far from the unstable orbits of the chaotic system the requested value of the control gain increases in an almost linear way.

Controlled 1-D n-Scrolls

The controlled 1D n-Scrolls,considering a single reference signal, is described by:

$$\begin{cases} \dot{x} = y + k_x(x_r - x) \\ \dot{y} = z + k_y(y_r - y) \\ \dot{z} = -ax - by - cz + d_1 f_1(x; k_1, h_1, p_1, q_1) \end{cases} \tag{32}$$

where $a = b = c = d_1 = 0.7, k_1 = 10$, $k_x = k_y$ and $f_1(x; k_1, h_1, p_1, q_1)$ is a PWL continuous function.

Considering

$$[x^{'}, y^{'}, z^{'}]^T = [x - ih_1, y, z]^T \tag{33}$$

$$[x_r^{'}, y_r^{'}] = [x_r - ih_1, y_r] \tag{34}$$

with $-p_1 \leq i \leq q_1$

and subspaces

$$\begin{aligned} E_{-p_1} &=]-\infty, -p_1 h_1 + \frac{h_1}{2}] \\ E_i &= [ih_1 - \frac{h_1}{2}, ih_1 + \frac{h_1}{2}] \text{ with } -p_1 + 1 \leq i \leq q_1 - 1 \\ E_{q_1} &= [q_1 h_1 - \frac{h_1}{2}, +\infty[\end{aligned} \tag{35}$$

with

$$\bigcup_{i=-p_1}^{q_1} E_i = \Re$$

The PWL function becomes:

$$f_1(x; k_1, h_1, p_1, q_1) = f_0(x^{'}; k_1) + 2ik_1 \tag{36}$$

The system (32), with $a = d_1$ e $h_1 = 2k_1$, becomes:

$$\begin{cases} \dot{x}' = y' + k_x(x_r' - x') \\ \dot{y}' = z' + k_y(y_r' - y') \\ \dot{z}' = -ax' - by' - cz' + d_1 f_0(x';k_1) \end{cases} \tag{37}$$

Therefore for each interval E_i of variable x, system (32) has the same dynamical behavior as the controlled double scroll (8), which can be verified by using the transformation (33) and (34).

Controlled 2D nxm-Scrolls

The controlled 2D nxm-Scrolls, considering a single reference signal, is described by:

$$\begin{cases} \dot{x} = y - \frac{d_2}{b} f_1(y;k_2,h_2,p_2,q_2) + k_x(x_r - x) \\ \dot{y} = z + k_y(y_r - y) \\ \dot{z} = -ax - by - cz + d_1 f_1(x;k_1;h_1;p_1,q_1) \\ \qquad + d_2 f_1(y;k_2;h_2;p_2,q_2) \end{cases} \tag{38}$$

Considering

$$[x',y',z']^T = [x - ih_1, y - jh_2, z]^T \tag{39}$$

$$[x_r', y_r'] = [x_r - ih_1, y_r - jh_2] \tag{40}$$

with $-p_1 \leq i \leq q_1$ e $-p_2 \leq j \leq q_2$

and subspaces

$$\Omega_{ij} = \{(x,y) : x \in E_i, y \in F_j\} \tag{41}$$

where E_i are defined in (35) and subspaces F_i are:

$$\begin{aligned} F_{-p_2} &=]-\infty, -p_2h_2 + \frac{h_2}{2}] \\ F_i &= [jh_2 - \frac{h_2}{2}, jh_2 + \frac{h_2}{2}] \text{ con } -p_2+1 \leq j \leq q_2-1 \\ F_{q_2} &= [q_2h_2 - \frac{h_2}{2}, +\infty[\end{aligned} \tag{42}$$

with

$$\bigcup_{j=-p_2}^{q_2} F_j = \Re \tag{43}$$

The PWL functions become:

$$f_1(x;k_1,h_1,p_1,q_1) = f_0(x';k_1) + 2ik_1 \tag{44}$$

$$f_1(y;k_2,h_2,p_2,q_2) = f_0(y';k_1) + 2jk_2 \tag{45}$$

When $a = d_1$, $h_1 = 2k_1$ and $h_2 = 2k_2$ for each region Ω_{ij}, system (38) has the same dynamical behaviors as the controlled double scroll system (8), which can be verified by using transformation (39) and (40).

Therefore, also the *nxm* grid configuration is brought back to the study to the first case, so we can apply all the theoretical and simulation results previously obtained.

Summary of the theoretical results

The theoretical analysis of the controlled multiscroll system has been carried out following a three steps demonstration as reported in the Appendix introduction. For sake of clarity we started analyzing the multiscroll chaotic system choosing the parameters in order to obtain a simple dynamical system behaving as a double scroll attractor. In this case the nonlinear function (PWL) that appears in the system equations, permits to study the effect of the weak chaos control on three subspaces in which the system can be linearized. The study of the linear subspaces leads to the behaviour of the whole

system using the concepts of real and virtual equilibrium points as discussed in [Chua et al., 1987, Manganaro et al., 1999]. A virtual equilibrium point is defined as an equilibrium point of a given linear subsystem that is placed out of its subspace. In this way the system trajectory tries to reach this virtual point but that behaviour leads to a transition towards a new subspace in which new equilibria govern the system dynamics. Indeed virtual equilibrium points are not equilibrium points of the subsystem because they are out of the system domain but they can be used to evaluate the system dynamical evolution, in fact they indicate the direction toward which the system trajectory is attracted. For each linear subspace, the eigenvalues have been studied as a function of the control gains. The mathematical analysis has been done for different input signals: in case of zero input and constant input. For each case the stability of the controlled system, through the eigenvalue analysis, was reported indicating the range of suitable control gains to be adopted. in the case of periodic inputs the system behaviour has been investigated through extensive computer simulations. Finally the results obtained for the controlled double scroll system, have been extended to an *nxm* grid of scrolls attractors with a suitable manipulation of the state variables. The theoretical analysis of the controlled multiscroll system, here reported, has been used to deeply understand the system behaviour and to suitably embed the control architecture on a roving robot, interfacing the system parameters with the robot sensors and actuators.

Acknowledgment

The authors acknowledge the support of the European Commission under the project FP6-2003-IST2-004690 SPARK "*Spatial-temporal patterns for action-oriented perception in roving robots*" and the PRIN 2005 project "*Nonlinear dynamic networks: analysis and applications*".

The authors would also like to thank Altera company for the support.

References

Arena P., Fortuna L., Frasca F., Pasqualino R., Patané L., 2004, “CNNs and Motor Maps for Bio-inspired Collision Avoidance in Roving Robots”, In *8th IEEE Int. Workshop on Cellular Neural Networks and their Applications (CNNA 2004), Budapest.*

Arena P., Crucitti P., Fortuna L., Frasca M., Lombardo D., Patané L., 2005a, “Perceptive Patterns for Mobile Robots via RD-CNN and Reinforcement Learning”, In *9th IEEE Int. Workshop on Cellular Neural Networks and their Applications, CNNA 2005, Taiwan*.

Arena P., Fortuna L., Frasca M., Lo Turco G., Patané L., Russo R., 2005b, “A new simulation tool for action oriented perception systems”, In *Proc. 10th IEEE International Conference on Emerging Technologies and Factory Automation, (ETFA 2005) September 19-22 Catania, Italy*.

Arena P., De Fiore S., Fortuna L., Frasca M., Patané L., Vagliasindi G., “Weak Chaos Control for Action-Oriented Perception: Real Time Implementation via FPGA”, In *Proc. International conference on Biomedical Robotics and Biomechatronics (Biorob 2006), Pisa, Italy, February 20-22, 2006.*

Arena P., De Fiore S., Frasca M., Patané L., 2006b, Web page available on-line: http://www.scg.dees.unict.it/activities/biorobotics/perception.htm

Arena P., Frasca M., Patané L., 2006c, Web page available on-line: http://www.spark.diees.unict.it/WCC.html

Arena P., Crucitti P., Fortuna L., Frasca M., Lombardo D., Patané L., 2007, “Turing patterns in RD-CNNs for the emergence of perceptual states in roving robots”, *International Journal of Bifurcation and Chaos*, 18(1), pp. 107-127.

Arkin R.C., 1998, *Behaviour Based Robotics*, MIT Press.

Beard, R. and McClain, T., 2003, “Motion Planning Using Potential Fields,” BYU.

Boccaletti S., Grebogi C., Lai Y. C., Mancini H., Maza D., 2000, “The Control of Chaos: Theory and Applications”, *Physics Reports*, 329:103–197.

Chua L. O., Desoer C. A., Kuk E. S., 1987, *Linear and nonlinear circuits*, McGraw-Hill, New York, NJ.

Freeman W. J., 1987, "Simulation of chaotic EEG patterns with a dynamic model of the olfactory system," *Biol. Cybern.*, vol. 56, pp. 139-150.

Freeman W. J., 1991, "The physiology of perception," *Sci. Am.* 264:78–85.

Freeman W.J., 1999, "Characteristics of the Synchronization of Brain Activity Imposed by Finite Conduction Velocities of Axons", *International Journal of Bifurcation and Chaos*, 10(10).

Freeman W. J., 2003, "A Neurobiological Theory of Meaning in Perception. Part I: Information and Meaning in Nonconvergent and Nonlocal Brain Dynamincs", *International Journal of Bifurcation and Chaos*, 13(9).

Freeman W. J., 2004, "How and Why Brains Create Meaning from Sensory Information", *International Journal of Bifurcation and Chaos*, 14(2).

Fuster, J.M., 2003, *Cortex and Mind: Unifying Cognition*, Oxford University Press.

Khatib, 0., 1986, "Real-time Obstacle Avoidance for Manipulators and Mobile Robots," *Intemational Journal of Robotics Research*, vol. 5, no. 1, pp 90–98.

Kozma R., Freeman W. J., 2000, "Chaotic Resonance - Methods and Applications for Robust Classification of Noisy and Variable Patterns", *International Journal of Bifurcation and Chaos*, 11(6).

Harter D., Kozma R., 2005, "Chaotic Neurodynamics for autonomous agents", *IEEE Trans. on Neural Networks*, 16(3), pp. 565–579.

Harter D., 2005, "Evolving neurodynamics controllers for autonomous robots", in *International Joint Conference on Neural Networks*, pp. 137–142.

Lapidus L., Seinfeld J., 1971, *Numerical Solution of Ordinary Differential Equations*, New York: Academic Press.

L*ü* J., Chen G., Yu X., Leung H., 2004, "Design and Analysis of Multiscroll Chaotic Attractors from Saturated Function Series",*IEEE Trans. Circuits Syst., I: Regular Paper*, 51.

Manganaro G., Arena P., Fortuna L., 1999, *Cellular Neural Networks: Chaos, Complexity, and VLSI Processing*, Springer-Verlag, Berlin.

Pyragas K., 1992, “Continuos Control of Chaos by Self-Controlling Feedback”, *Phisical Letters A*, 170:421–428.

Pyragas K., 1993, “Predictable Chaos in Slightly Pertirbed Unpredictable Chaotic Systems”, *Physics Letters A*, 181:203–210.

Ritter H., Martinetz T., Schulten K., 1992, “Neural Computation and Self-Organizing Maps - An Introduction”, *Addison-Wesley, New York.*

Skarda C.A., Freeman W.J., 1987, “How brains make chaos in order to make sense of the world,” *Behav. Brain Sci.*, vol. 10, pp. 161-195.

Uexku, J.V., 1926, *Theoretical Biology*, Harcourt, Brace.

Verschure P. F. M. J., Voegtlin T., Douglas R. J., 2003a, “Environmentally mediated synergy between perception and behaviour in mobile robots”, *Nature*, 425:620–624.

Verschure P. F. M. J., Althaus P., 2003b, “A real-world rational agent: unifying old and new AI”, *Cognitive Science* 27:561–590.

Tables

Table 1: Performace of the weak chaos control implemented on FPGA

	Time[a]
WCC implemented on the Nios	10 s
WCC implemented on the Nios using custom instructions	1 s
WCC implemented in VHDL	2.8 ms

[a]The time is referred to the execution of 2000 samples and the value of the integration step is 0.1.

	V_1	V_2	V_3	P_{V_1}/P_{V_3}
$0 \leq k < 0.105$	saddle	unstable focus	saddle	real
$0.105 \leq k < 2.539$	stable focus	unstable focus	stable focus	real
$2.539 \leq k < 2.953$	stable focus	stable focus	stable focus	virtual
$k \geq 2.953$	stable node	stable focus	stable node	virtual

Table 2: Stability of a controlled double scroll system with a zero input. In the last column the real or virtual condition of P_{V_1} and P_{V_3} are reported.

$0 \leq k < 0.105$	chaotic behavior
$0.105 \leq k < 0.534$	two stable equilibria
$0.534 \leq k < 2.539$	two stable equilibria,
	unique when the reference coincide with a scroll center
$k \geq 2.539$	unique equilibrium point

Table 3: Behavior of the double scroll with constant input.

Figure Captions

Figure 1. Projection of the 5x5 grid of *scroll attractors* in the plane *x*-*y*.

Figure 2. Block diagram of the control scheme when three distinct reference signals (i.e. sensorial stimuli) are perceived by the multiscroll system.

Figure 3. Behavior of the multiscroll system controlled by a reference cycle. (a) Low control gains: $k_x = k_y = 0.7$; (b) High control gains: $k_x = k_y = 1.8$.

Figure 4. Limit cycle obtained when the multiscroll system is controlled by two sensorial stimuli. (a) Control gains: $k_{x_1} = k_{x_2} = k_{y_1} = k_{y_2} = 0.8$; (b) Control gains: $k_{x_1} = k_{y_1} = 2$, $k_{x_2} = k_{y_2} = 0.6$.

Figure 5. An example of the evolution of the multiscroll system when controlled by reference dynamics associated with sensors. (a) When there is no input sensorial stimulus, the system has chaotic behavior; (b) when a single stimulus is perceived, the system converges to the reference cycle; (c) the system persists in the limit cycle behavior until the sensorial stimulus is no more perceived; (d) when the stimulus ends, the system behaves chaotically.

Figure 6. Scheme of a simulated roving robot equipped with four distance sensors and a target sensor. In the phase plane $x - y$, the reference cycles associated to each sensor are reported. Target sensor, due to its omnidirectional field of view, is associated to four reference cycles. Cycles can be generated by system (1) with parameters: $a = b = c = d_1 = d_2 = 1$, $k_1 = k_2 = 50$, $h_1 = h_2 = 100$, $p_1 = p_2 = 1$, $q_1 = q_2 = 2$ and changing the offset. Equivalently, equations (6) can be used.

Figure 7. Interface of the 3D simulator used to compare robot navigation control algorithms: frame (I) is the menu that permits to set the robot and the environment characteristics; frame (II) permits

to choose the simulation parameters; frame (III) represents the behavior of the controlled system; frames (IV) and (V) show a 3D view and a 2D top view of the environment and the robot.

Figure 8. Environments used to evaluate the performance of the proposed architecture in comparison with the Potential Field. The dimension of both the arenas is 50x50 robot units, the randomly placed objects correspond to walls and obstacles and the circles indicate the visibility range of the targets.

Figure 9. Trajectories followed by the robot controlled through: (a) (d) local potential field; (b)(e) WCC with forward exploration behaviour, (c)(f) WCC with chaotic exploration behaviour.

Figure 10. Cumulative number of targets found for the three control algorithms in the (a) four-rooms environment (Fig.8 (a)) and (b) environment filled with obstacles (Fig.8(b)). The simulation time is 10000 epochs and the mean number of targets, mediated over 5 simulations, calculated with time windows of 1000 epochs, is indicated. The bars show the minimum and maximum number of targets found.

Figure 11. Area explored using the three control algorithms in the (a) four-rooms environment (Fig.8 (a)) and (b) environment filled with obstacles (Fig.8(b)). The arena which dimension is 50x50 robot units, has been divided into locations of 2x2 robot units. The simulation time is 10000 epochs and the mean value of area explored, mediated over 5 simulations, calculated with time windows of 1000 epochs, is indicated. The bars show the minimum and maximum value.

Figure 12. Reference cycles representing the sensors of the robot.

Figure 13. VHDL Entity *sim_full*.

Figure 14. VHDL Entity *sim*.

Figure 15. An example of the evolution of the multiscroll system when controlled by reference signals associated with sensors. (a) When there are no input sensorial stimuli, the system shows a chaotic behavior; (b) when a single stimulus is perceived, the system converges to the reference cycle; (c),(d) when more than one stimulus are perceived, the resulting cycle is placed between the references and the position depends on the gain values associated to the references.

Figure 16. The experimental platform is composed by the FPGA Altera board, a roving robot, a computer and a wireless communication system.

Figure 17. Description of the framework used during the robot navigation experiments.

Figure 18. Screen shoots of the robot during the exploration of an unknown environment and the evolution of the controlled multiscroll system. (a)(b) Sensory signals acquired during the experiment with a sample time of about 0.350 s; the acquired infrared sensor output reported on the y-axis goes from 0 to 70 that correspond to a distance from 80 cm to 25 cm. (c)(d) More than one sensor is concurrently active. (e)(f) No obstacle is considered relevant for the robot movement. (g)(h) The right sensor (S3) detects an obstacle.

Figure 19. Environments used to evaluate the performance of the proposed architecture controlling a roving robot. The dimensions of both arenas are 10x10 robot units. An example of the trajectory followed by the robot controlled through the WCC_f algorithm is shown.

Figure 20. Area explored for the environments shown in Fig. 19 (a) and (b). The arena whose dimension is 10x10 robot units, has been divided into locations of 1x1 robot units. The experiment time is 420s and the mean value of area explored, mediated over 5 experiments, calculated with time windows of 30s, is indicated. The bars show the minimum and maximum value.

Figure 21. The behavior of the subspaces V_1's and V_3's eigenvalues for $k_x = k_y = k \in [0,30]$, the system is stable for $k \geq 0.105$.

Figure 22. The behavior of the subspace V_2's eigenvalues for $k_x = k_y = k \in [0,30]$, the system is stable for $k \geq 2.539$.

Figure 23. Equilibrium points for the subspaces V_1, V_2 and V_3 for $k_x = k_y = k \in [0,1000]$, for $k \geq 2.539$ the unique real fixed point is the origin.

Figure 24. The sets A, B and C for $k_x = k_y = 3$. They are the sets of the input such that equilibrium points are real for subspaces V_1, V_2 and V_3 respectively.

Figure 25. Position of input $x_r = -2$, $y_r = 2$ in the sets *A*,*B*,*C*,*D* and response of the system for $k_x = k_y = 3$ (a)(c) and $k_x = k_y = 1$ (b)(d).

Figure 26. Value of the control gain needed to obtain an input/output error less than 20% at different values of frequency and amplitude of the periodic reference signal. The offset is $(x_{off}, y_{off}) = (10,0)$. The behavior for the other offsets is similar.

Figures

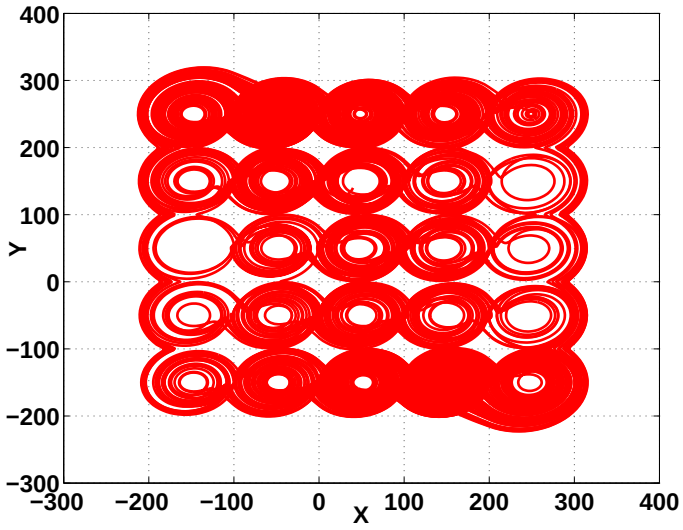


Figure 1: Projection of the 5x5 grid of *scroll attractors* in the plane *x*-*y*.

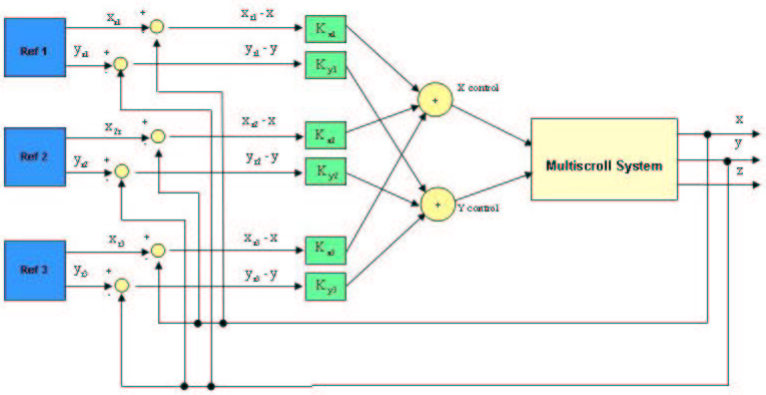
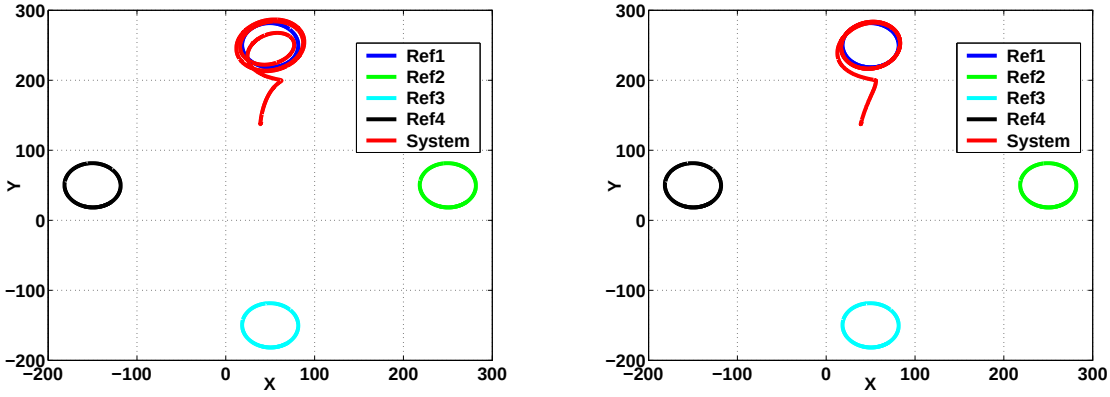
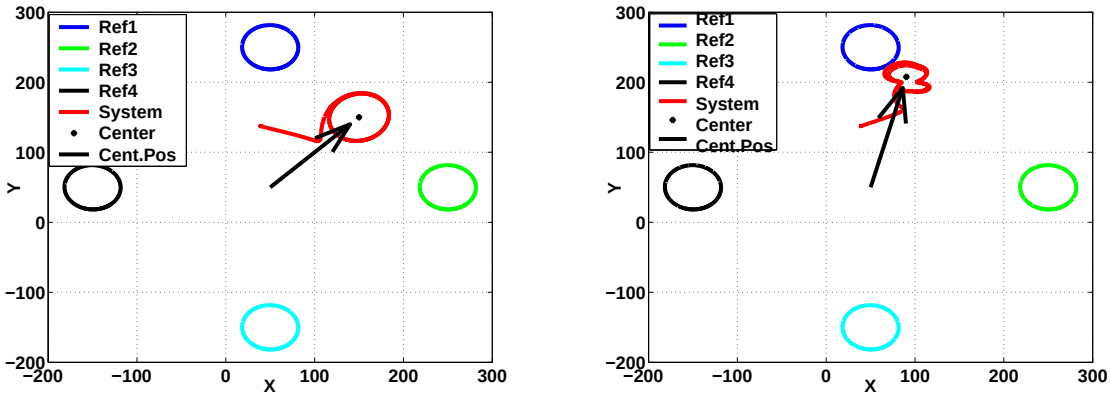


Figure 2: Block diagram of the control scheme when three distinct reference signals (i.e. sensorial stimuli) are perceived by the multiscroll system.



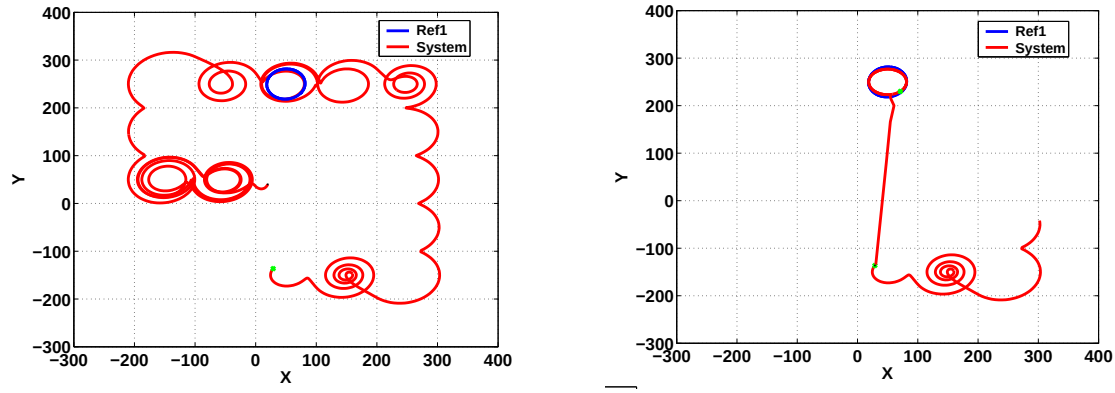
(a) (b)

Figure 3: Behavior of the multiscroll system controlled by a reference cycle. (a) Low control gains: $k_x = k_y = 0.7$; (b) High control gains: $k_x = k_y = 1.8$.

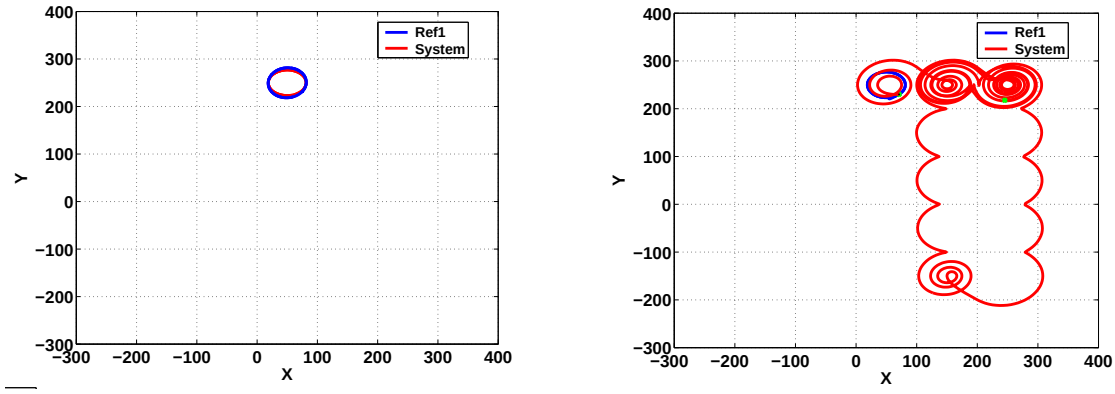


(a) (b)

Figure 4: Limit cycle obtained when the multiscroll system is controlled by two sensorial stimuli. (a) Control gains: $k_{x_1} = k_{x_2} = k_{y_1} = k_{y_2} = 0.8$; (b) Control gains: $k_{x_1} = k_{y_1} = 2$, $k_{x_2} = k_{y_2} = 0.6$.



(a) (b)



(c) (d)

Figure 5: An example of the evolution of the multiscroll system when controlled by reference dynamics associated with sensors. (a) When there is no input sensorial stimulus, the system has chaotic behavior; (b) when a single stimulus is perceived, the system converges to the reference cycle; (c) the system persists in the limit cycle behavior until the sensorial stimulus is no more perceived; (d) when the stimulus ends, the system behaves chaotically.

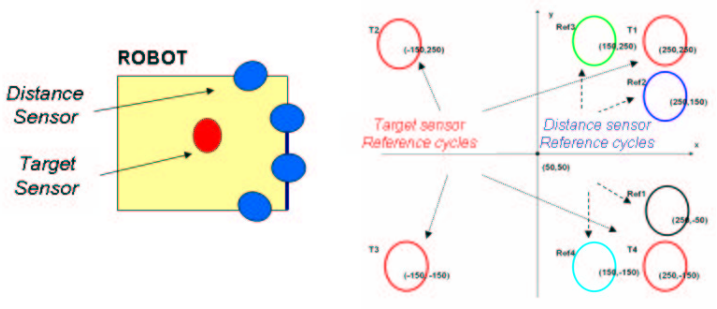


Figure 6: Scheme of a simulated roving robot equipped with four distance sensors and a target sensor. In the phase plane $x-y$, the reference cycles associated to each sensor are reported. Target sensor, due to its omnidirectional field of view, is associated to four reference cycles. Cycles can be generated by system (1) with parameters: $a = b = c = d_1 = d_2 = 1$, $k_1 = k_2 = 50$, $h_1 = h_2 = 100$, $p_1 = p_2 = 1$, $q_1 = q_2 = 2$ and changing the offset. Equivalently, equations (6) can be used.

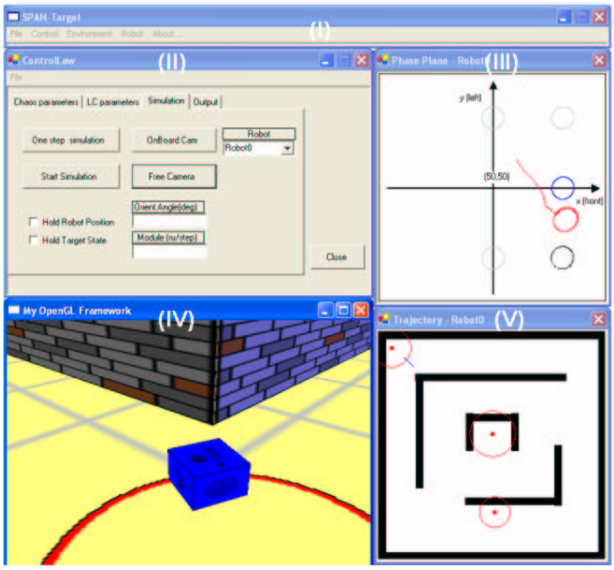
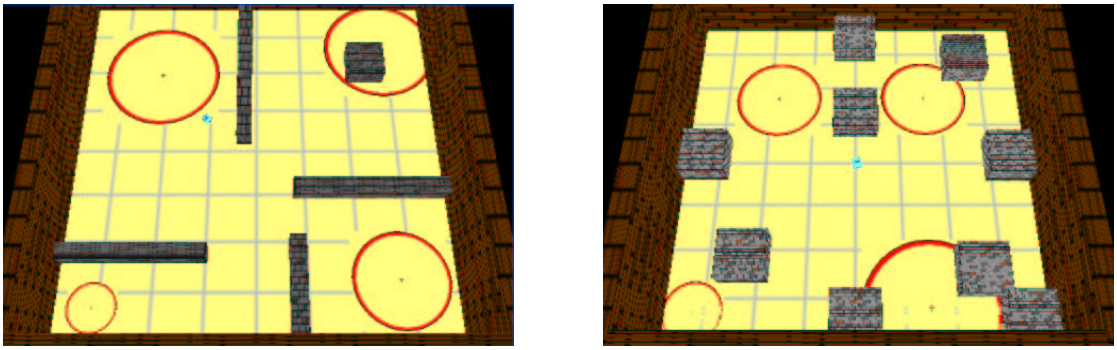
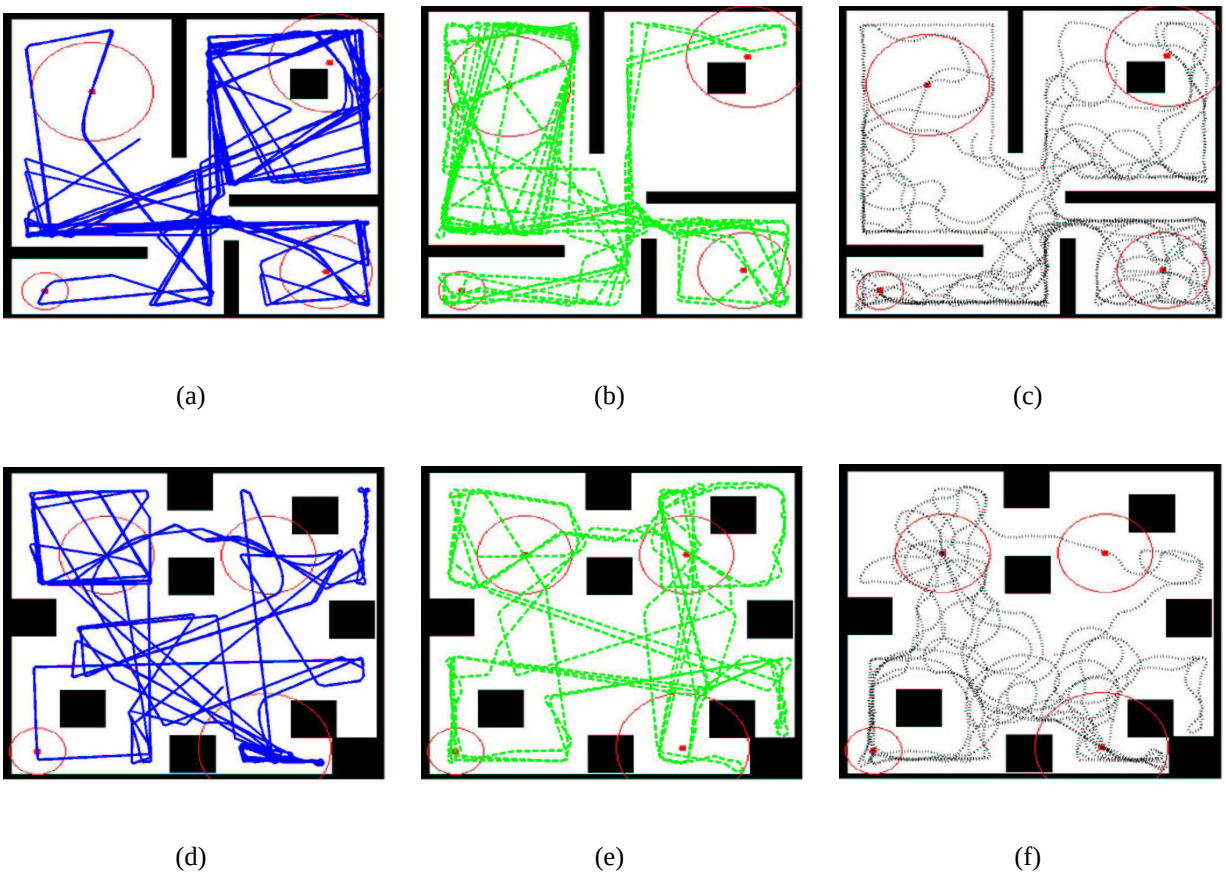


Figure 7: Interface of the 3D simulator used to compare robot navigation control algorithms: frame (I) is the menu that permits to set the robot and the environment characteristics; frame (II) permits to choose the simulation parameters; frame (III) represents the behavior of the controlled system; frames (IV) and (V) show a 3D view and a 2D top view of the environment and the robot.



(a) (b)

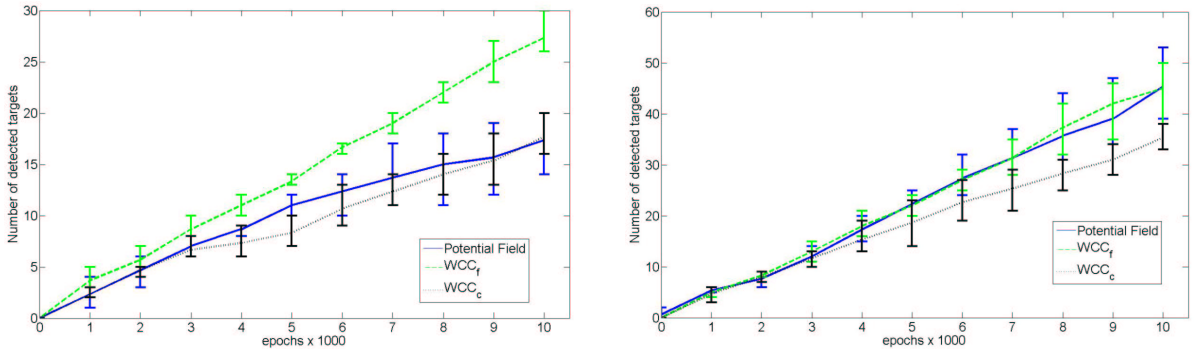
Figure 8: Environments used to evaluate the performance of the proposed architecture in comparison with the Potential Field. The dimension of both the arenas is 50x50 robot units, the randomly placed objects correspond to walls and obstacles and the circles indicate the visibility range of the targets.



(a) (b) (c)

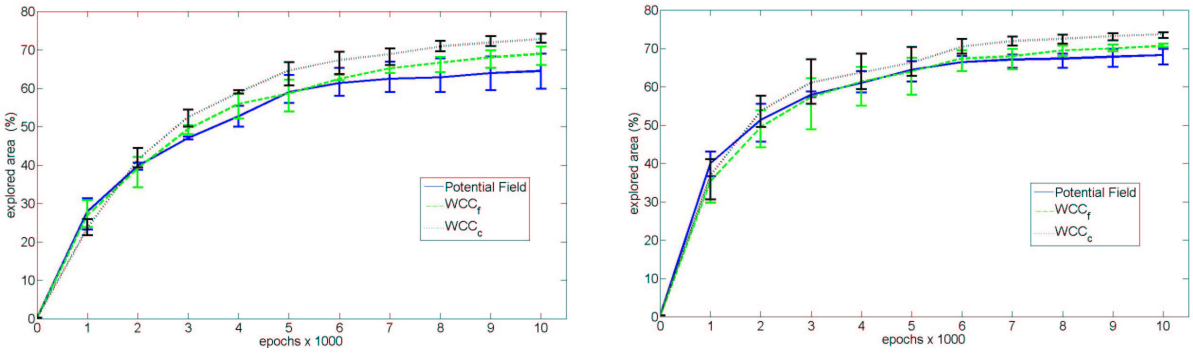
(d) (e) (f)

Figure 9: Trajectories followed by the robot controlled through: (a) (d) local potential field; (b)(e) WCC with forward exploration behaviour, (c)(f) WCC with chaotic exploration behaviour.



(a) (b)

Figure 10: Cumulative number of targets found for the three control algorithms in the (a) four-rooms environment (Fig.8 (a)) and (b) environment filled with obstacles (Fig.8(b)). The simulation time is 10000 epochs and the mean number of targets, mediated over 5 simulations, calculated with time windows of 1000 epochs, is indicated. The bars show the minimum and maximum number of targets found.



(a) (b)

Figure 11: Area explored using the three control algorithms in the (a) four-rooms environment (Fig.8 (a)) and (b) environment filled with obstacles (Fig.8(b)). The arena which dimension is 50x50 robot units, has been divided into locations of 2x2 robot units. The simulation time is 10000 epochs and the mean value of area explored, mediated over 5 simulations, calculated with time windows of 1000 epochs, is indicated. The bars show the minimum and maximum value.

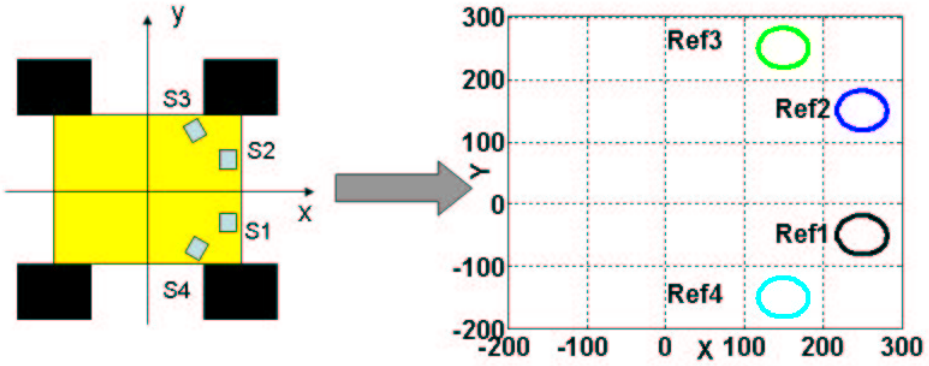


Figure 12: Reference cycles representing the sensors of the robot.

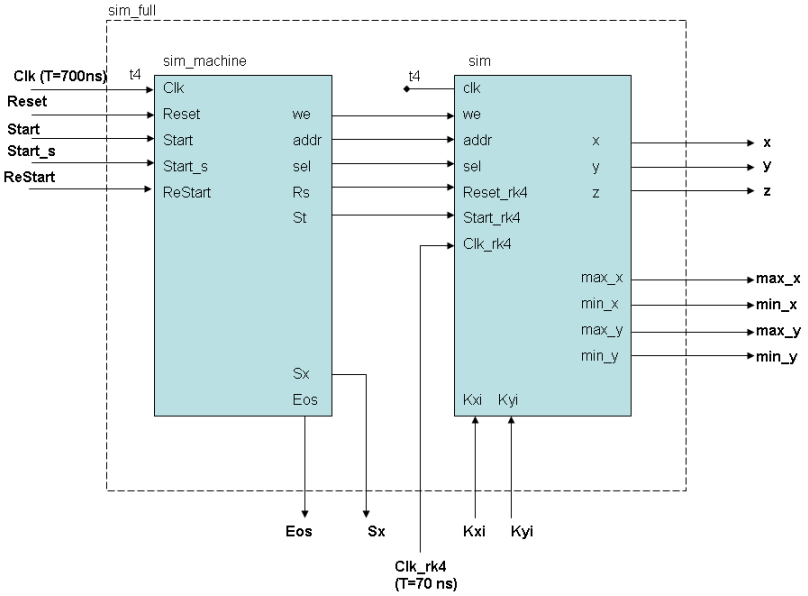


Figure 13: VHDL Entity *sim_full*.

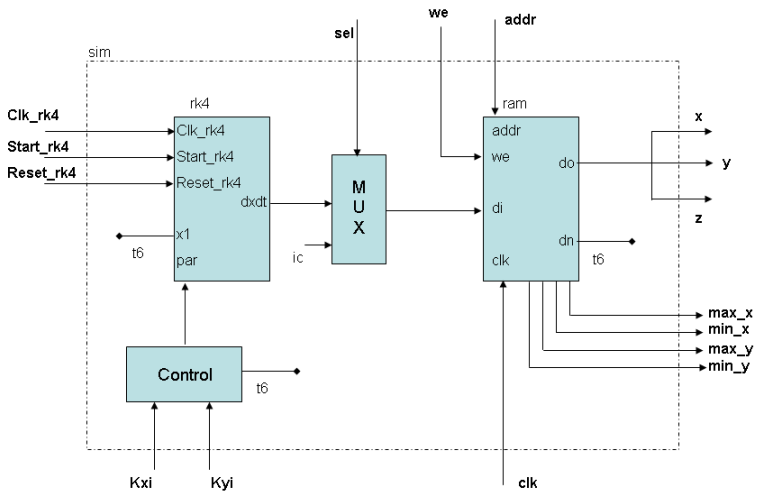
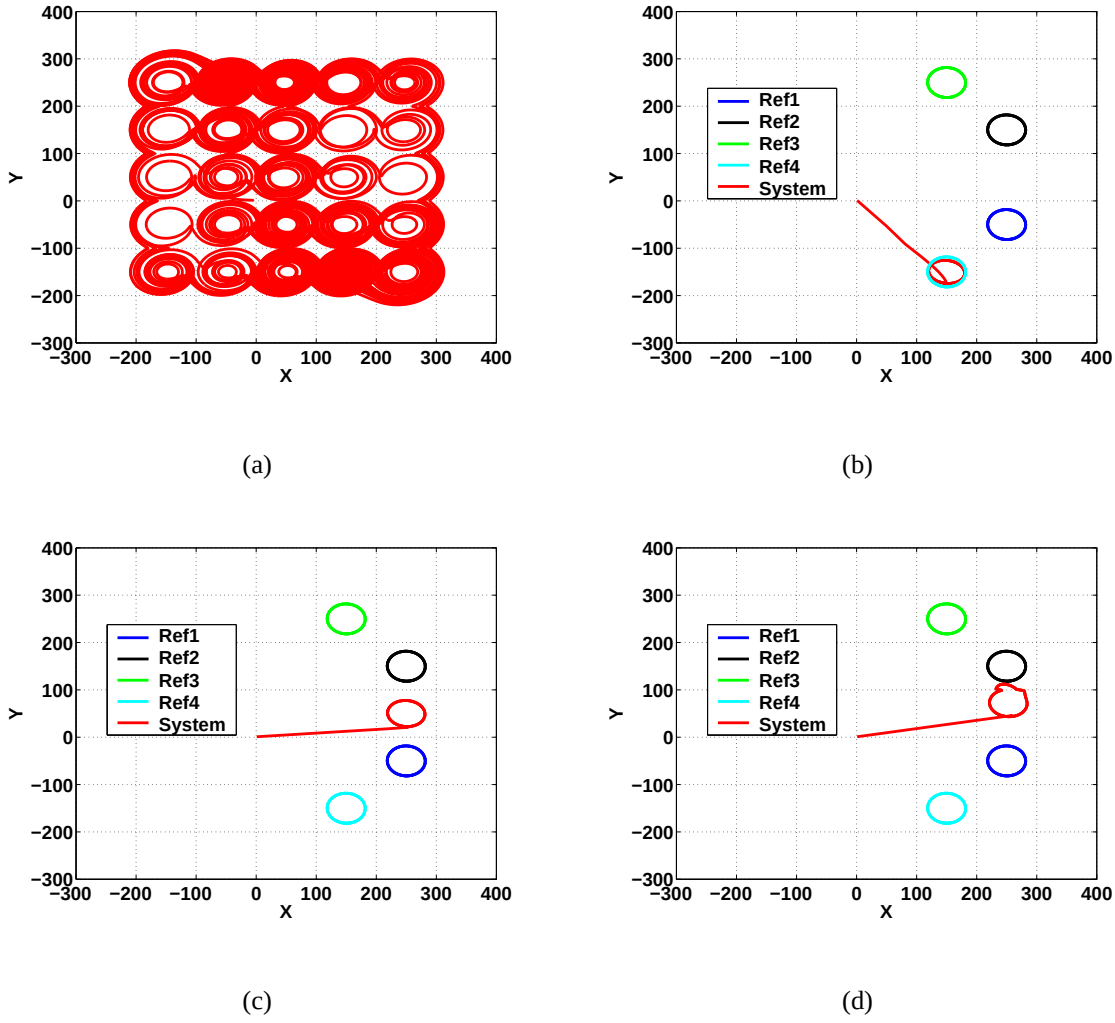


Figure 14: VHDL Entity *sim*.



(a) (b)

(c) (d)

Figure 15: An example of the evolution of the multiscroll system when controlled by reference signals associated with sensors. (a) When there are no input sensorial stimuli, the system shows a chaotic behavior; (b) when a single stimulus is perceived, the system converges to the reference cycle; (c),(d) when more than one stimulus are perceived, the resulting cycle is placed between the references and the position depends on the gain values associated to the references.

Figure 16: The experimental platform is composed by the FPGA Altera board, a roving robot, a computer and a wireless communication system.

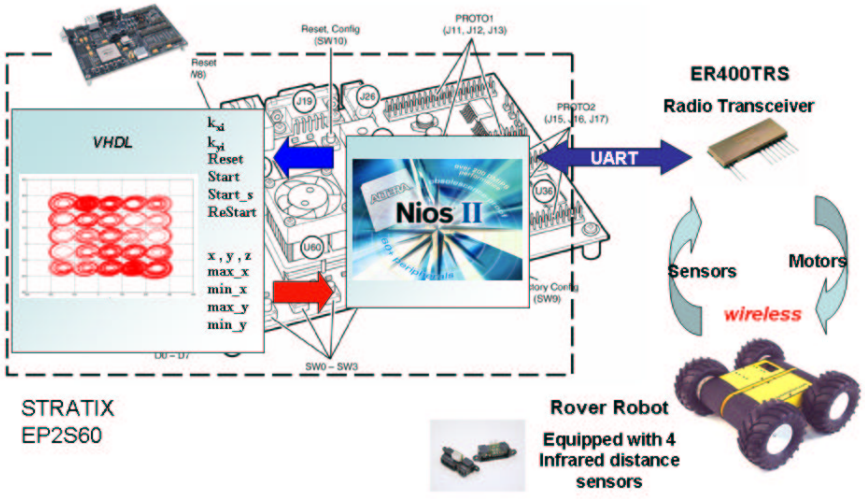
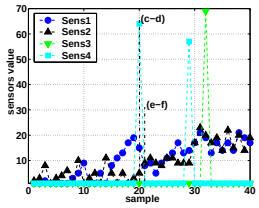
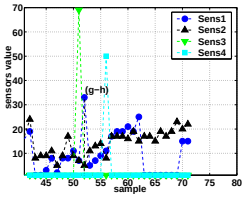


Figure 17: Description of the framework used during the robot navigation experiments.

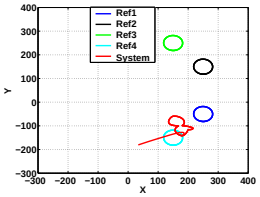


(a)



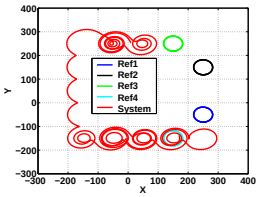
(b)

(c)



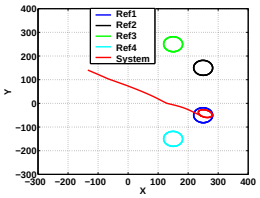
(d)

(e)



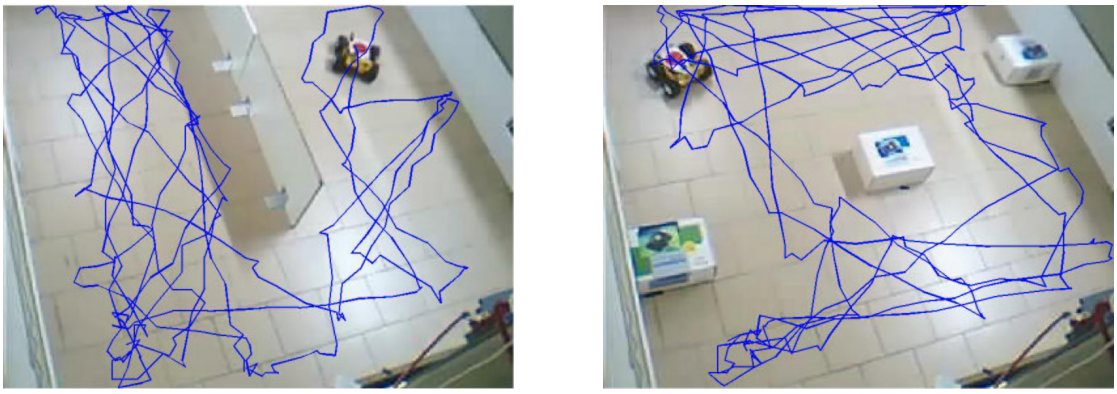
(f)

(g)



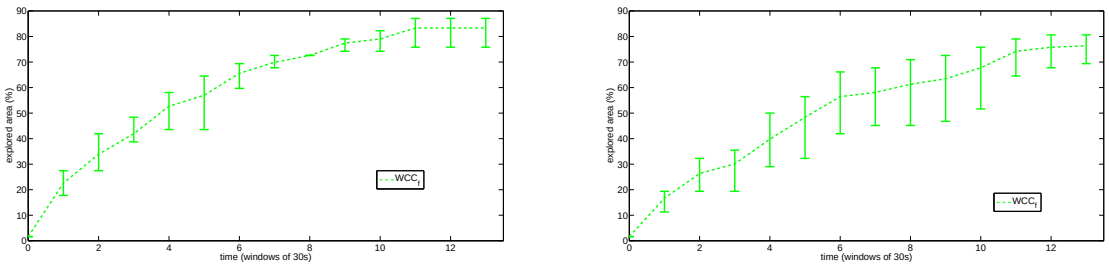
(h)

Figure 18: Screen shoots of the robot during the exploration of an unknown environment and the evolution of the controlled multiscroll system. (a)(b) Sensory signals acquired during the experiment with a sample time of about 0.350 s; the acquired infrared sensor output reported on the y-axis goes from 0 to 70 that correspond to a distance from 80 cm to 25 cm. (c)(d) More than one sensor is concurrently active. (e)(f) No obstacle is considered relevant for the robot movement. (g)(h) The right sensor (S3) detects an obstacle.



(a) (b)

Figure 19: Environments used to evaluate the performance of the proposed architecture controlling a roving robot. The dimensions of both arenas are 10x10 robot units. An example of the trajectory followed by the robot controlled through the WCC_f algorithm is shown.



(a) (b)

Figure 20: Area explored for the environments shown in Fig. 19 (a) and (b). The arena whose dimension is 10x10 robot units, has been divided into locations of 1x1 robot units. The experiment time is 420*s* and the mean value of area explored, mediated over 5 experiments, calculated with time windows of 30*s*, is indicated. The bars show the minimum and maximum value.

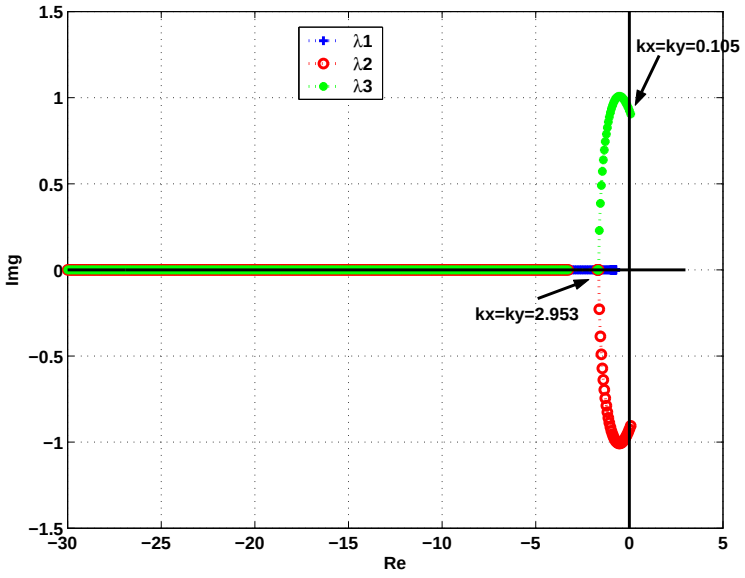


Figure 21: The behavior of the subspaces V_1's and V_3's eigenvalues for $k_x = k_y = k \in [0, 30]$, the system is stable for $k \geq 0.105$.

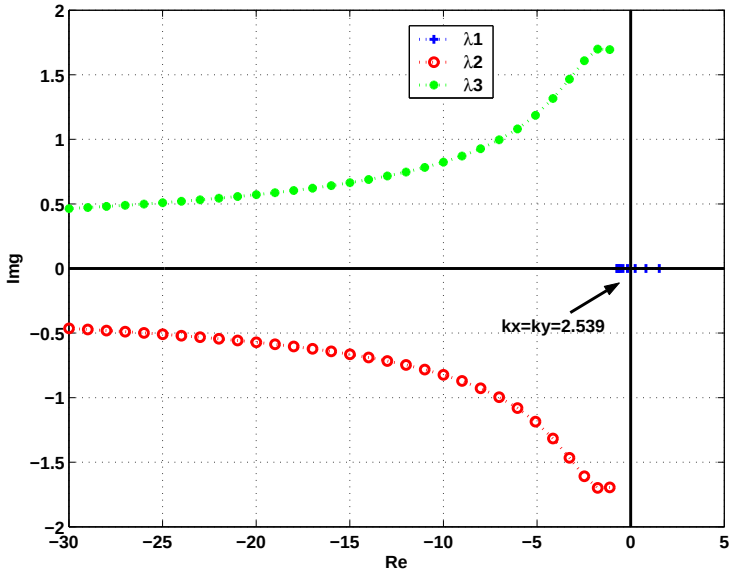


Figure 22: The behavior of the subspace V_2's eigenvalues for $k_x = k_y = k \in [0, 30]$, the system is stable for $k \geq 2.539$.

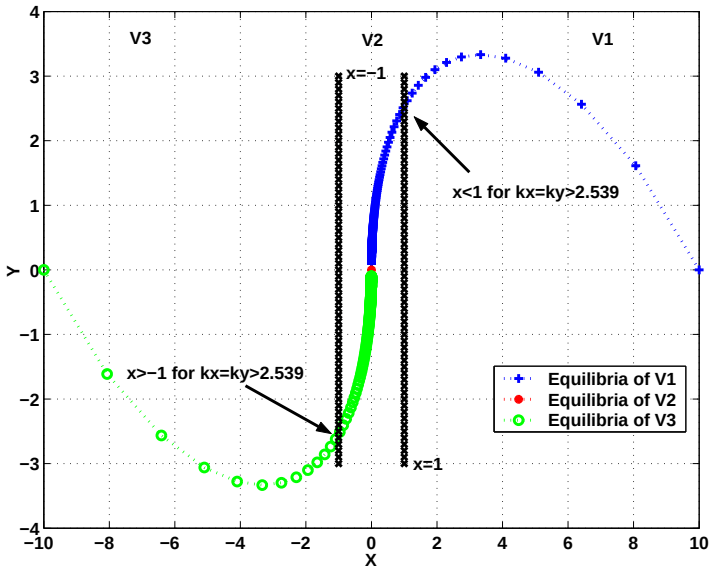


Figure 23: Equilibrium points for the subspaces V_1, V_2 and V_3 for $k_x = k_y = k \in [0, 1000]$, for $k \geq 2.539$ the unique real fixed point is the origin.

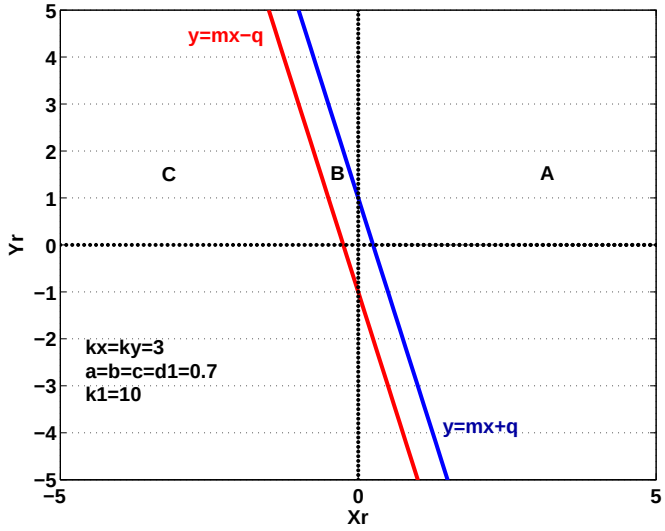


Figure 24: The sets A, B and C for $k_x = k_y = 3$. They are the sets of the input such that equilibrium points are real for subspaces V_1, V_2 and V_3 respectively.

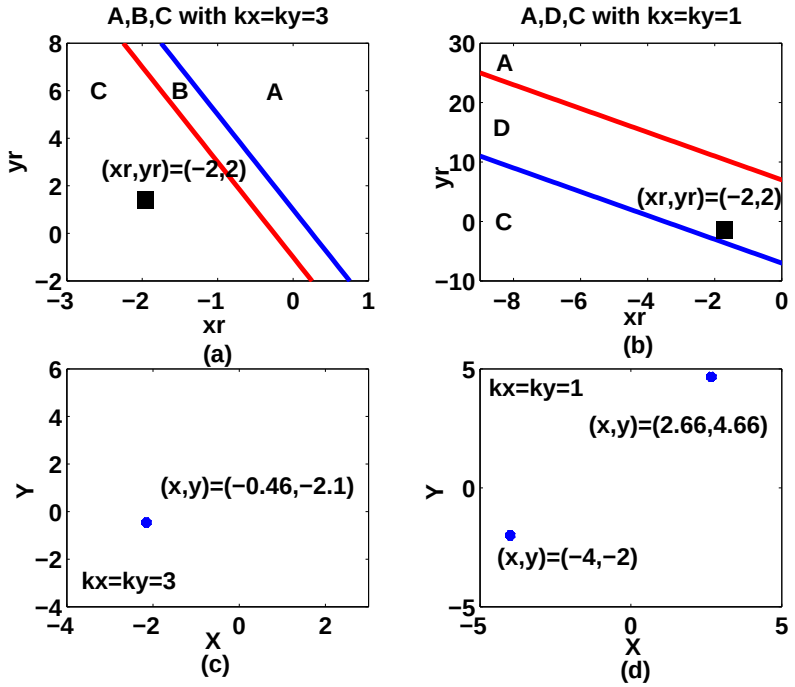


Figure 25: Position of input $x_r = -2$, $y_r = 2$ in the sets A,B,C,D and response of the system for $k_x = k_y = 3$ (a)(c) and $k_x = k_y = 1$ (b)(d).

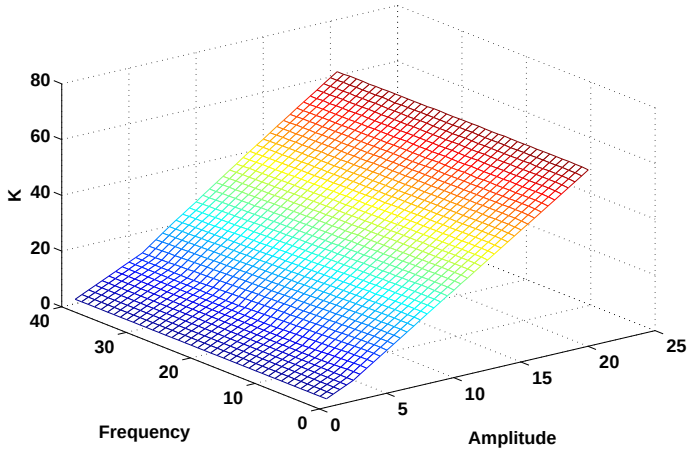


Figure 26: Value of the control gain needed to obtain an input/output error less than 20% at different values of frequency and amplitude of the periodic reference signal. The offset is $(x_{off}, y_{off}) = (10, 0)$. The behavior for the other offsets is similar.