A new simulation tool for action-oriented perception systems

P. Arena, L. Fortuna, M. Frasca, G. Lo Turco, L. Patané and R. Russo Dipartimento di Ingegneria Elettrica, Elettronica e dei Sistemi, Universitá degli Studi di Catania

iparumento ul ingegneria Elettrica, Elettromea e del Sistemi, Universita degli Studi di Catama

Viale A. Doria 6, 95125 Catania, Italy.

parena@diees.unict.it

Abstract

In the last years, in the area of bio-inspired robotics, the research activity has been directed to hight level aspects that includes psychological theories and behavioral approaches.

In this paper a new simulation tool for perceptive system based on the sensing-perception-action loop is proposed. The framework has been designed to evaluate the performance of control strategies applied to the navigation of autonomous robots. The tool can be used to create a 3D environment in which the exploring capabilities of a robot executing a navigation task such as for example a food retrieval task, can be evaluated. The behavior of the system is monitored with the help of a 3D real-time visualizer supported by a graphic representations of the trajectory followed.

1 Introduction

Nowadays the research activity involved in the application of psychological and cognitive principals in fields like artificial intelligences and robotics is in continuous growing.

The bio-inspired approach represent the most recent attempt to comprehend the capacity of animals for autonomous generation of intelligent behavior in changing environments. These guidelines have been used in the formalization of the term Animats that indicates autonomous robots or simulations of animals [1]. Animats are precious instruments for the study of cognitive systems. With the term cognition we intend a lot of mental processes like attention, planning, reasoning, understanding language and others. Different levels of abstraction are commonly used to try to model cognitive phenomena, starting from the simplest stimulus-reaction approach, to arrive to face, at a high level, with the so-called action-perception cycle [2].

The perceptive system learns from the environment, subjected to the affordance of the senses. The new methodology for action-oriented perception will result in actions which will exceed a simple reactive behavior, to proceed towards deliberation [3]. In our case the perception task is focalized to solve the locomotion/navigation problem in a cluttered environment, fusing together different kinds of sensing capabilities to carry out a given task. Perception is treated as a dynamic phenomenon, and the relevant information drawn from the environment conditions are summarized in a pattern.

To sustain this fervent activity a new simulation tool has been designed. It can be used to validate and compare different navigation strategies that could be also successively implemented in hardware. Simulation offers a solution to the problem of increasing complexity in process sensory information and in control of numerous actuators by replacing the real robot with a computer model. The utilization of a simulation framework makes possible to evaluate at the same time a great number of variables and then to make an ease modification of the system and control parameters. The benefits of simulation include the reduced cost of design and testing, the ability to carry on controlled experiments, and the ability to store data [4]. Therefore in the first step of analysis, a simulation tool, producing a complete view of the system, is useful for a further application in the real world.

One of the field mostly indicated for the application of the biological principles is the navigation control of autonomous robots.

The simulation software that will be described in this work has been developed taking into account the depicted scenario. The aim is to create a platform in which innovative control strategies for the navigation of autonomous roving robots can be developed, evaluated and compared between them.

The task assigned to the robots in the first simulation carried out is the food retrieval, that is a standard benchmark for autonomous robot. The robot must show the ability to avoid obstacles and to collect targets.

A key point taken into account during the realization of the tool was the possibility to create and visualize a 3D environment in which the robot can be simulated. The objects and the robots enclosed into the simulation field have been designed to mirror a real environment. For example, the definition of the features of each element can be extended taking care of many practical details. Therefore it is possible to simulate the behavior of a robot in presence of several different sensors and to repeat subsequently the tests changing the set of sensors or modifying the environment conditions. In this way it is also possible to identify the best configuration for the robot.

In the next section a description of the software architecture is given, in section 3 the software interface is described in details, in section 4 the guidelines for a navigation control algorithm based on the paradigm of actionoriented perception are illustrated and in section 5 the simulation results are reported. Finally section 6 draws the conclusions and shows some possible future developments.

2 Architecture of the simulation tool

The main logical blocks that have been taken into account during the simulator design are shown in fig. 1. The scheme follows the fundamental paradigm of sensingperception-action loop.

The objective of each robot is to accomplish a given task (e.g. obstacle avoidance and target finding). The robot is equipped with a sensory system used to capture the stimuli coming from the environment. The information acquired by the sensors are then processed to create an internal representation of the external environment. The perception of the environment is finalized to the formalization of an action in order to accomplish a given task. The internal state of the system has been modeled with a non linear dynamical system evolving in a chaotic attractor that can be stabilized into periodic patterns representing the percept. The whole control algorithm is finalized to select the best action from the robot point of view. Afterwards the action is taken, the environment will be influenced and the new condition will be processed during the next sensing step.



Figure 1. Basic scheme characterizing a perceptive system. The simulator design is based on three functional blocks: Sensing, Perception and Environment. The system autonomously chooses to perform an action on the basis of the external stimuli coming from the environment and on the given task that must be accomplished.

The blocks described in fig. 1 have been developed realizing standard interfaces that guarantee the expansion of the tool that represents the starting point of a continuously growing platform able to fuse together the multiple aspects characterizing a perceptive system. The main characteristic of the framework is modularity; in fact it is possible to implement new control algorithms only producing the corresponding function and linking it to the rest of the structure.

In the developed platform it is possible to create a three-dimensional environment in which different objects can be introduced, it is possible to create robots with different features and the user can choose between multiple graphical interfaces during the analysis of simulation results, in particular a 3D visualization of the environment, realized with the OpenGL library [5, 6]. This utility allows to view the scene from different points of view modifying position, orientation and inclination of the camera. Moreover, several output windows have been designed to monitor in details the state of the system, in particular the robot position, orientation, sensory input and so on.

In order to satisfy the previously discussed key points an object-oriented programming language, the MS VC++ 7.1 has been used. This choice allows us to organize the structure in several distinct classes which are also logically separated.

In fig. 2 the class hierarchy is shown. The software contains an I/O interface: the user can formulate the control law and define the environment in terms of objects, targets (i.e. sources) and robots equipped with sensors; the simulator gives a real-time output that includes a 3D visualizer, a trajectory tracking panel and several information about the state of the control algorithm.



Figure 2. Flow diagram of the interactions between the different classes constituting the simulation platform for perceptive systems. The Main class coordinates the exchange of information between the other classes, the simulation outputs are given by the ControlLaw class. The user can define the 3D environment through the four classes: ObjCreation, SrcCreation, RobotCreation, EnvCreation.

3 Platform description

In this section it is shown the software interface. A sequence of steps that would be executed to initialize a project are reported in the following.

The first step consists in defining the environment dimensions: length, height and width of the arena in which the robots will move. The successive operation is the creation of objects/obstacles. It is possible to define the shape of the obstacle (e.g parallelepiped, cylinder, cone, ...), its dimension (height, length, width or radius) and other properties that can be used for the visualization (e.g. the color). The tool has been designed to be a growing platform for the study of perceptive control strategies in particular for robot navigation, for this reason it is possible to include also other features to each element of the simulated environment in order to improve the capability of the controlled system. For example each obstacle can be characterized also in terms of mass if the robot is able to move or carry it, temperature, and so on. Finally the object can be deployed into the arena defining its position and orientation with the help of a graphic interface as shown in fig. 3. This represents the top view of the arena, different elements can be distinguished: black rectangles are obstacles previously defined and then already present in the scene; the white rectangle is the obstacle that we are creating, using the keyboard or the mouse it is possible to locate it into the room. Of course it is also possible to load and save every element from/to files.

Other important elements that can be introduced are the sources. When a source is created, it is represented by a point surrounded by a circle that indicates its visibility range.

The procedure of creating robots is similar to the object creation, in fact it is necessary to define the robot dimensions (each robot is represented by a parallelepiped), position in the arena and other graphical features. A second step is devoted to the definition of the sensors equipped on the robot. Different types of sensors can be assigned to a robot; actually three category are implemented: target, distance and contact sensors. However contact sensor can be considered as a particular distance sensor with a very small detection range. Successively depending on the sensor type other parameters must be initialized, for example the distance sensor is characterized by an orientation angle respect to the reference system of the robot, a visibility angle, and a maximum visibility radius. Each sensor is then linked to the robot in the chosen position.

4 The perceptive control algorithm

The simulator has been developed in a modular way to test several control strategies; in this section a particular control algorithm, designed to guide the navigation of an autonomous roving robot, will be discussed. Drawing inspiration from Freeman's theories [7, 8], it has been realized a real-time control technique that emulates the perceptive processes of the brain in which particular cerebral patterns emerge depending on the perceived sensorial stimuli.

The creation of the internal representation can be modeled with a dynamical system that shows a chaotic behav-



Figure 3. View of the arena in which different elements constituting the environment can be allocated.

ior and when opportunely controlled converges to a limit cycle (i.e. a cerebral pattern).

The control technique here proposed is based on a multiscroll chaotic system [9] that can be controlled through reference signals towards regular periodic patterns.

To solve the robot navigation task, each sensor equipped on the robot is associated with a reference cycle, used to control the system realizing a multiscroll chaotic attractor.

The control has been realized with a simple feedback on the state variables x_1 and x_2 controlled to follow the reference cycles. The control gains are related to the amplitude of the sensory stimuli. The behavior of the controlled system is associated to an action, in terms of speed and rotation angle, that represent the robot response to the stimuli perceived from the environment. When none stimulus is perceived (there aren't active sensors) the system evolves in a chaotic behavior and the robot continues to explore the environment moving with constant speed and without modifying its orientation. However other alternative strategies may be applied during exploration for example by using the trajectory generated by the chaotic multiscroll system.

When external stimuli are perceived, the controlled system converges to a cycle (i.e. a periodic pattern) that depends on the contribute of active sensors. The action that will be executed is chosen according with the characteristics of the cycle, in particular its position in the phase plane. When the stimuli end the multiscroll evolves in chaotic way again. An example of the chaotic evolution followed by the perception of two different external stimuli is shown in fig. 4.

Moreover, an important issue consists in locating the reference cycles in the phase plane in accordance with the topological distribution of sensors on the robot (fig.5). Therefore in the proposed control technique, when a stimulus due to a distance or contact sensor is perceived, a reference cycle appears on the phase plane in the same



Figure 4. Evolution of the control system in the phase plane. (a),(b),(c) None sensorial stimulus is applied: the system presents a chaotic behavior. (d) A couple of stimuli are perceived: the system converges on a cycle which is the result of the trade-off between the two reference signals.

position in which the obstacle is detected. When a target sensor is active two different situations have been considered: only a target sensor is active; there are also other active sensors. In the first case the robot rotates of the angle detected from the target sensor (this angle is calculated respect to the frontal axis of the robot) and it moves directly toward the target (i.e. reactive behavior). Instead in the second case the target sensor is associated with a reference cycle that appears in the phase plane in a symmetrical position with respect to the motion direction. In other words, this technique consists in considering the target as a virtual obstacle. In this way the generated reference cycle produces a rotation towards the target with a weak gain because the obstacle avoidance is the principal task.

5 Simulation Results

In this section several simulation results are illustrated to demonstrate the potentialities both of the simulator and





of the control algorithm in it implemented.

The experiments have been accomplished in an arena of rectangular dimensions of 40x40ru (robot units), in which different environment configurations have been considered.

The robot has parallelepiped shape (1x1x0.6 ru) and it is equipped with six sensors: three distance, two distance/contact and one target sensors. A detailed description of each sensor is given in Table 1, a contact sensor is equivalent to a distance sensor with a reduced visibility range. The topological location of the sensors on the robot is described in fig. 6.

Table 1. Parameters that characterize the					
sensors equipped on the robot: sensor					
type, visibility radius and cone, orientation					
angle and the label related to fig. 6					

Sensor	Vis. Radius	Vis. Cone	Orient.	Label
Target	Variable	360 deg	//	(T)
Distance	2.5 ru	60 deg	0 deg	(A)
Contact	0.6 ru	90 deg	315 deg	(B)
Contact	0.6 ru	90 deg	45 deg	(C)
Distance	1.0 ru	50 deg	270 deg	(D)
Distance	1.0 ru	50 deg	90 deg	(E)

The first simulation that will be discussed has been realized to test the exploratory abilities of the autonomous robot completely controlled by the perception system in a food retrieval task. A top view of the created environment is depicted in fig. 7 where four targets with different visibility radius are located in a complex environment filled with obstacles.

Figure 8 shows the complete user interface of the simulator, in which can be distinguished four different sections. Two windows are dedicated to the specification of the control algorithm: the ControlLaw panel is used to modify the parameters of the control system, the Phase Plane panel shows the x-y evolution of the multiscroll system. In particular in the sub-section Output of the the ControlLaw panel, it is possible to monitor the status of every robot





(c)

Figure 6. (a) Positions of sensors equipped on the robot. The robot is seen from the top and the front side is on the right. (b) Reference cycles associated to the distance/contact sensors present on the robot. (c) Configuration and visibility cone of the distance/contact sensors.

introduced into the environment. It is possible to watch the current position, its orientation and the last chosen action. Moreover in the sub-section Simulation it is possible instead to modify the parameters for the 3D visualization. The user can choose between a free camera which is static and can be positioned everywhere in the environment and an on-board camera that is located on a desired robot and changes its position and orientation with the robot.

The other two sections of the user interface permit to see in real-time the results of simulation and to interact with the environment: the Trajectory panel gives a top view of the environment and is used to monitoring the robot trajectory, while the OpenGL panel guarantees a complete knowledge of the environment with a three dimensional view. Therefore all the information necessary to evaluate the robot behavior and then the control algorithm performance, are visible to the user and can be also saved for a successively accurate analysis.

The robot, thanks to the sensory information processed by the multiscroll system, shows the ability to avoid obstacles and to reach targets. When a target is reached, it is



Figure 7. A top view of the simulation environment used to test the capability of the robot controlled by the perceptive control algorithm based on dynamical systems.



Figure 8. The user interface is characterized by four sub-windows used to modify the control parameters and to monitor the behavior of the controlled system.

turned off until another target will be found, for this reason the robot is constraint to explore new areas. In fig. 9 (a) the trajectory followed by the robot is shown, in short time the robot has been able to reach all the targets as required by the assigned task. Moreover extending the simulation time neither collision or deadlock problems have been found. In fig. 9 (b) the explored area covered by the robot is shown. As can be noticed the environment has been completely explored and then it is possible to affirm that the robot is able to find targets located in arbitrary positions.

Similar results have been obtained in a completely different scenario, where the arena has been divided into four rooms each containing a target. Also in this case the robot has been able to retrieve all the targets exploring the whole area (see fig. 10).

Furthermore the three dimensional reconstruction of



Figure 9. Trajectories covered by the robot. (a) After a short simulation time the robot has been able to reach all the targets introduced into the environment. (b) The whole arena has been explored at the end of simulation.



Figure 10. (a) Simulation environment, (b) at the end of simulation the whole environment has been explored. The robot guided by the control algorithm maintains a safety distance from obstacles and walls.

the environment is extremely useful to monitor the behavior of multiple robots that operate in competition or cooperation into the area. In fig.11 three different views of the same scene are reported. The environment selected is the same of the last simulation but in this case two robots are simulated at the same time. The first snapshot of fig.11 has been captured by a camera equipped on one of the robots. This capability is fundamental to implement a visual sensor, in fact other sensors like a virtual camera can be added to the sensor library. The information acquired by the camera can be processed by the perceptive system to make a data fusion with the other sensory information in order to accomplish more complex tasks like landmark navigation, place recognition, path planning and others.

6 Conclusions

In this work a simulation tool for the study of actionoriented perceptive system has been presented.

The software has been developed to help the process of application of psychological and biological principles characteristic of the real world into an artificial environment simulated in order to evaluate the performance of the proposed strategies.

The architecture has been designed following a project that takes into account the concept of modularity as a fundamental purpose. Taking inspiration by functional flow diagrams describing the paradigm of action-oriented perception, several distinct classes have been developed by using a standard interface. In this way the platform is predisposed to the definition of new elements or the substitution of one block with another class for example when a different control law may be tested.

Therefore once chosen the control algorithm to validate, the user creates the environment that includes different types of elements: obstacles, sources and robots. The creation of a robot includes also the definition of the sensory system equipped on it.

The OpenGL library has been adopted to realize a three-dimensional representation of the environment and to guarantee a real-time visualization of the simulated robot.

The control algorithm discussed in this paper is based on the Freeman's theory that consider the perception mechanism as the formation of spatio-temporal patterns emerging from complex chaotic behavior. These principles have been modeled by using the methodologies of chaos control applied to a dynamical system characterized by a multiscroll chaotic attractor.

The simulation results obtained in different scenarios have been illustrated in details with particular attention to the capability of the tool. The simulation output is visible in different panel each one used to monitor a specific aspects: the action selected by the control system, the sensory inputs perceived by the robot, the trajectory followed, the internal representation of the environment (e.g. the state trajectories of the multiscroll) and a real-time 3D visualizer of the dynamical environment.

The possibility to introduce new types of sensors has been also discussed. In particular a visual sensor can be easily implemented acquiring the output of a camera equipped on the robot.

Moreover the tool can be also used to simulate multiple robots that can cooperate to achieve a given task or show a competitive behavior sharing limited resources.

It is obvious that at the moment it is possible to find many improvements that can be executed on the platform, because we are showing a new application, born with the objective to be improved continuously.

Acknowledgments

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".

References

- J. Dean, "Animats and what they can tell us," *Trends in Cognitive Sciences*, vol. 2, pp. 60–67, February 1998.
- [2] R. C. Arkin, Behaviour Based Robotics, MIT Press, 1997.
- [3] H. A. Mallot, "Behavior-Oriented Approaches to Cognition: Theoretical Perspectives", *Theory in Bio-sciences*, 116, 196-220, 1997.
- [4] J. Reichler and F. Delcomyn, "Biorobotic approaches to the study of motor systems," *Int. Journal of Robotics Research*, vol. 19(1), pp. 41–57, January 2000.
- [5] M. Segal K. Akeley, "The OpenGL Graphics System", Version 1.2.1 - Oct. 14, 1998.
- [6] Redbook, The Official Guide to Learning OpenGL, Second Edition.
- [7] W.J. Freeman, "How and Why Brains Create Meaning from Sensory Information", *International Jour*nal of Bifurcation and Chaos, Vol. 14, No. 2, 2004.
- [8] W.J. Freeman, "Characteristics of the Synchronization of Brain Activity Imposed by Finite Conduction Velocities of Axons", *International Journal of Bifurcation and Chaos*, Vol. 10, No. 10, 1999.
- [9] J. L, G. Chen, X. Yu, H. Leung, "Design and Analysis of Multiscroll Chaotic Attractors from Saturated Function Series", IEEE Trans. Circuits Syst., I: Regular Paper, Vol. 51, 2004.



(a)



(b)



(c)

Figure 11. Different points of view acquired by an on-board camera (a) and a free camera (b)-(c) used to monitor the two robots simulated into the environment.