# Perspectives of Using Membrane Computing in the Control of Mobile Robots

Cătălin Buiu, Ana Brânduşa Pavel, Cristian Ioan Vasile, and Ioan Dumitrache

Politehnica University of Bucharest,
Splaiul Intepedenţei, No. 313, 060042, Bucharest, Romania
{cbuiu,apavel,cvasile,idumitrache}@ics.pub.ro

**Abstract.** This paper presents an original approach to the control of mobile robots using a natural computing based solution, which falls beyond the traditional use of Artificial Intelligence and bio-inspired methods in robot control. The idea is to use the modeling power of P systems, which are based on the structure and functioning of living cells. Results obtained so far are presented in a synthetic way and directions for further developments are given.

**Keywords:** numerical P systems, enzymatic numerical P systems, membrane controllers, mobile robots, distributed and parallel systems

## 1 Introduction

AI-inspired methods have a long tradition in the control of mobile autonomous robots, and a number of important concepts, tools and techniques of AI have been applied successfully to various robotic applications. One can mention here planning, learning and other aspects. Recently, computational intelligence field has provided a variety of methods that are being applied to the problem of controlling mobile robots, such as fuzzy logic, neural networks, evolutionary computing.

Of more recent interest is the paradigm of natural computing which refers to that category of methods which are directly inspired by the structure and functioning of living systems, such as evolutionary computing and swarm intelligence.

Membrane computing, introduced by Gh. Paun [1], has raised a lot of interest during the last 10 years. Several results, mainly at theoretical level, have been generated and some practical applications of P systems (or membrane systems) were also presented. For a review, consult [2].

A P system (PS) is a computational model based upon the structure of an eukaryotic cell and it mimics the interaction and evolution of chemicals inside of biomembranes [2]. The architecture (or membrane structure) of a PS is a hierarchical arrangement of membranes, in which the outermost membrane is called the skin membrane and separates the system from its environment. This skin membrane is analogous to the plasma membrane of a living cell. The membranes define regions as the biomembranes define working compartments. A membrane is called elementary if it has no other membrane inside. For a sample membrane structure, refer to [3] and [2].

Next section will present the main arguments for using membrane systems in robotics applications, while section 3 will briefly enumerate the main results obtained so far, both at practical level, and at theoretical level. This paper ends with some conclusions and ideas for possible developments.

## 2 Membrane Computing and Robotics. An Overview of Concepts

The basic idea of membrane systems, to separate computing processes in different compartments (membranes) which are able to inter-communicate, sounds appealing to researchers in autonomous robotics. In 2009, there was proposed the first idea of using membrane systems as basic modules for cognitive architectures, in general, and for robot control architectures, in particular. The idea is to treat the control architecture as a basic membrane (skin membrane, in the terms of membrane systems), while different modules may be treated/implemented as specific membranes. For example, the execution level of a general control architecture may be implemented as a membrane containing several sub-membranes responsible for various desired behaviors of the robot.

Numerical P systems (NPS) are a class of computing models inspired by the cell structure in which numerical variables evolve inside the compartments by means of programs; a program (or rule) is composed of a production function and a repartition protocol. The variables have a given initial value and the production function is usually a polynomial. The value of the production function for the current values of the variables is distributed among variables in certain compartments according to a repartition protocol; NPS model was first introduced in [3] with possible applications in economics. Using NPS as a possible tool for modeling robot behaviors is a novel approach which will be further discussed.

In [4], several examples of robot behaviors (obstacle avoidance, wall following, following a leader) modeled with NPS, are presented. The main advantages of using NPS as a modeling tool are the numerical and the naturally parallel and distributed nature of the model. Membranes of a NPS can be distributed over a grid or over a network of microcontrollers in a robot. The computation done in each membrane region (the execution of the membranes programs) can also be done in parallel. This is very important, because given a membrane system, which is an abstract implementation of a desired behavior, it can be executed in a distributed and parallel way without having to worry about the design and implementation issues regarding parallelization and distribution. Therefore, NPS can be used as a modeling tool for parallel and distributed control systems.

The performance of a controller can be measured by the mean execution time of a cycle [4]. After designing a membrane controller, the membrane system can be simulated using a numerical P systems simulator such as SNUPS [5], [6] or SimP [7]. One benefit of using membrane controllers is that the computational performance can be increased by improving the simulator's performance (parallelization, distribution and other optimizations) and not by modifying the membrane controllers themselves. The membrane simulator can be considered as a virtual machine like Java or Python virtual machines. It can be seen as a "membrane computer" that runs "membrane programs" (in this case, the membrane controllers). The simulator is the middleware between the hardware and the membrane controllers. Therefore, by optimizing its performance, the performance of all the defined membrane controllers increases.

Design and implementation of robot controllers require deterministic computational models. In order to be deterministic, a NPS model must have only one rule per membrane and, most of the time, this restriction makes the model rigid and difficult to use. Therefore, an extension of the NPS model, Enzymatic Numerical P Systems (ENPS), was proposed in the context of modeling robot behaviors. ENPS model allows the parallel execution of more rules (programs) per membrane while keeping the deterministic behavior. ENPS use some special variables, inspired by the behavior of biological enzymes which, associated to rules (in analogy to chemical reactions), can decide whether a rule is active or not at a given computational step. A rule is active if the associated enzyme has a greater value than the minimum of the variables involved in the rule or if the rule has no associated enzyme. Details about ENPS model together with formal definition and examples can be found in [8], [9] and [10]. Theoretical results about the universality of the model are presented in [11].

By adding enzyme-like variables to the NPS models, the modeling power of NPS increases. The enzymatic mechanism controls the execution flow of a NPS with multiple rules per membrane. The possibility of selecting and executing more production functions per membrane makes NPS a more flexible modeling tool. ENPS robot controllers (as those described in [8], [9]) have a less complex structure than the NPS ones, therefore less computations must be performed, increasing the performance of the system.

## 3 Current Theoretical and Experimental Results

Both NPS and ENPS models could be used for modeling autonomous mobile robot behaviors [4], [8] and [9]. The numerical nature, the distributed and parallel structure and the computing power, make membrane controllers suitable candidates for the control of complex systems.

A framework for testing membrane controllers on real and simulated robots has been developed. The framework integrates xml files which store robot behaviors in a platform-independent way, a simulator for numerical P systems [7] and Webots, a professional mobile robot simulator [12]. The proposed membrane controllers have been tested both on real and simulated robots (e-puck and KheperaIII).

Membrane controllers modeled using ENPS have less complicated structures than the ones modeled with NPS. For instance, the ENPS controller for obstacle avoidance proposed in [8] is a membrane system with 9 membranes, while the NPS model for obstacle avoidance proposed in [4] has 37 membranes. Although the ENPS model for obstacle avoidance has more rules than the NPS model, not all of them are active (fewer rules are executed during a computational step in the ENPS model than in the NPS one).
A NPS structure for odometric localization has been modeled, but its structure is far way complicated than the ENPS one proposed in [9]. In this case, the NPS controller was modeled using 24 membranes, while the ENPS model has only 5 membranes. In this case, enzymes control the program flow and are used as stop conditions and synchronization mechanism. Therefore, the model of the controller is clearly simplified, easier to implement and more efficient (regarding the computational process) than the one modeled by classical NPS. The naturally parallelized membrane representation and the numerical nature of the membrane components represent advantages for both NPS and ENPS in designing and modeling robot behaviors.

Based on the theoretical and practical results, some important advantages of using ENPS to classical NPS are mentioned in [8] and [9]. The main advantages of ENPS towards NPS are that enzyme-like variables can control the program flow by deciding which rules to be executed in a computational step, they can control the synchronization between parallel computations, they can be used to filter noise from the sensors or to detect the termination of the program.

## 4 Conclusions

This paper presents the currently existing results in modeling robot controllers by means of P systems. Numerical P systems and its extension, enzymatic numerical P systems, have been used to achieve cognitive robot behaviors.

Taking into account the most important properties of numerical P systems: their numerical nature, parallelization, distribution and the tree-like structure of the membranes, a future aim of this research direction is to design hardware membrane controllers and prove their efficiency (for example implementing the membrane controllers on a FPGA which can be connected to the robot).

## Acknowledgments

This paper is suported by the Sectorial Operational Programme Human Resources Development, financed from the European Social Fund and by the Romanian Government under the contract number SOP HRD/107/1.5/S/82514.

## References

1. Păun, Gh.: Computing with membranes. *Journal of Computer and System Sciences*, vol. 61, 108–143. Elsevier (2000)
2. Păun, Gh., Rozenberg, G., Salomaa, A. (eds.): The Oxford Handbook of Membrane Computing. Oxford University Press (2010)
3. Păun, Gh., Păun, R.: Membrane computing and economics: Numerical P systems. Fundamenta Informaticae, no. 1, vol. 73, 213–227 (2006).
4. Buiu, C., Vasile, C., Arsene, O.: Development of membrane controllers for mobile robots. *Information Sciences*, doi:10.1016/j.ins.2011.10.007 (2011)
5. Arsene, O., Buiu, C., Popescu, N.: SNUPS – A simulator for numerical membrane computing. *Intern. J. of Innovative Computing, Information and Control*, vol. 7, no. 6, 3509–3522 (2011)
6. Buiu, C., Arsene, O., Cipu, C., Pătraşcu M.: A software tool for modeling and simulation of numerical P systems. *BioSystems*, vol. 103, no. 3, 442–447 (2011)
7. Pavel A.B.: Membrane controllers for cognitive robots. Master Thesis, Department of Automatic Control and System Engineering, Politehnica University of Bucharest (2011)

8. Pavel A.B., Buiu C.: Using enzymatic numerical P systems for modeling mobile robot controllers. *Natural Computing*, in press (2011)
9. Pavel A.B., Vasile, C., Buiu, C.: Robot controllers implemented with enzymatic numerical P systems. submitted (2011)
10. Pavel, A.B., Arsene, O., Buiu, C.: Enzymatic numerical P systems – a new class of membrane computing systems. *The IEEE Fifth Intern. Conf. on Bio-Inspired Computing. Theory and applications* (BIC-TA 2010), 1331–1336, Liverpool, U.K. (2010)
11. Vasile, C., Pavel, A.B., Dumitrache, I., Păun, G: On the Power of Enzymatic Numerical P Systems. submitted (2011)
12. Cyberbotics, professional mobile robot simulation http://www.cyberbotics.com/