Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 00 (2010) 1–9
Procedia Computer Science

International Conference on Computational Science, ICCS 2010

Car-driving assistance using organization measurement of reactive multi-agent system

Franck Gechter, Jean-Michel Contet, Pablo Gruer, Abderrafiaa Koukam

University of Technology of Belfort-Montbeliard (UTBM),
Systems and Transportation Laboratory (SET), Belfort, France
http://set.utbm.fr/

Abstract

This work presents an approach to the obstacle avoidance problem, applicable in the frame of driver assistance. A decision, expressed as a proposed acceleration vector for the vehicle, is elaborated from the evaluation of a set of indicators characterizing the global state of a system of reactive agents (RMAS). Those agents evolve in a virtual environment produced on the base of vehicle's perceptions of the material environment around it. Agent-to-agent and agent-to-environment interactions are defined in order to produce a distribution of agents over the virtual environment. This distribution, taken as the global state of the system, is analyzed by applying a set of indicators inspired from statistical physics, to calculate a new vehicle's acceleration vector. This work presents the details of the RMAS model and its interaction laws, together with the global state evaluation functions. The approach has been applied to experimentation with a laboratory vehicle. Some experimental results are presented here.

Keywords: Reactive Multi-Agent Systems, Obstacle Avoidance, Car-Driving Assistance

1. Introduction

Obstacle avoidance helps the driver to prevent accidents by detecting vehicles or other obstacles on the road ahead and by avoiding collision. Currently, cars with Adaptive Cruise Control, integrate obstacle avoidance function, with very basic functionalities: based on information obtained from radar sensors, those systems produce visual and acoustic warnings. Recent approaches tend to integrate obstacle avoidance with motion planning and path following, a couple of well formulated and closely tied problems, directed to the objective of autonomous navigation. They are classically solved by computing a trajectory avoiding known obstacles [4]. However, standard methods for motion planning can not be applied to dynamical and uncertain environment. Therefore, reactive navigation seems to be a suitable approach. Proposals found in literature can be classified into three categories:

- *Physics inspired methods*: in this kind of methods physics inspired model are applied to sensor information and are transformed into motion command law. Among these methods can be cited the potential field methods such as VFH (Vector Field Histogram) and VFF (Virtual Force Field) [5], the perfume and the fluid analogy [6].

- *Motion Command selection methods*: These approach compute a set of suitable motion commands to select one command based on navigation strategies [7]. The selection strategy can then be based on classical algorithms such as subsumption architecture or HTN (Hierarchical Task Networks).

- *High Level methods*: these methods compute some form of high-level information description from the sensory information to obtain a motion command. For instance, the nearness diagram navigation [8] belongs to this group of approaches.

Multi-agent systems are an efficient approach for problem solving and decision making applied to a wide range of applications. Among the classical models, the reactive approach is one of the most interesting thanks to their intrinsic properties and features such as simplicity, flexibility, reliability, self-organization/emergent phenomena, low cost agent design and adaptation skills,... It has been shown that Reactive Multi-Agent System (RMAS) approach is efficient for tackling complex problems such as pedestrian multi-level simulation [2], cooperation of situated agents/robots, data fusion and problem/game-solving.

This work aims to contribute to the obstacle avoidance problem within an autonomous navigation framework, but the proposed approach can also be integrated to driver assistance systems. The goal of this paper is to propose a decision making approach to obstacle avoidance, in which decision consists on a new acceleration vector for the vehicle to avoid the obstacle. In this approach, decision is a result of the evaluation of emergent organization patterns in a system of reactive agents. In this model interaction and environment have a preponderant role. As a matter of fact, the agents are immaterial and evolve in an environment which is an abstraction of material environment, constructed from vehicle's perceptions. As already stated, the decision is built by evaluating emergent dynamical properties, thanks to global state indicators inspired from statistical physics.

The paper is structured as follow: Section 2 presents the reactive multi agent system (RMAS) model applied to decision making in dynamical obstacle avoidance, together with the approach to evaluate the emergent global state of the RMAS. Section 3 presents experimental results made on an experimental intelligent vehicle. Finally, we present conclusions and our view on future work.

2. A Reactive Multi-Agent Solution

This section gives a detailed description of the proposed RMAS driving assistant. After drawing the main principle of the system, each specific component will be described. The proposed model can be considered as an application of the methodology developed in [1]. This methodology puts the environment in the center of the problem-solving process as the place where the problem and its constraints are specified and presented to the perception of the agents. Then interactions are defined in order to take into account the problem's dynamics. Finally, these elements lead to an emergent structure considered to be a solution to the problem. In the context of this article, the emergent structure has to be interpreted as a command to be applied to the vehicle.

2.1. Main principle

The vehicle decision process is considered as a multi-agent system, the agents of which make collectively the best decision considering the perceived constraints. (To clarify the paper, these agents will be called Decision Agents (DA) in the following sections.) In order to compute the decision at each time step, DA need to perceive vehicle's world or an abstraction of it. As state in [1] agents environment can be considered to be the link between the real world and agents world. Thus, a DA environment is built by merging the information given by the vehicle sensors. Each obstacle detected by one or more sensors in real world is thus turned into an repulsive spot in DA environment. Repulsive spot characteristics are defined using sensors information (size, orientation, height,...). Driver command is also integrated in DA environment. It is represented as an attractive spot, the position of which depends on steering position and required velocity. The boundaries of DA environment are defined to be the union of all sensor ranges. Then, interaction are defined, using physical model, to make Decision Agents being repulsed by perceived obstacle and attracted by driver decision. The emerging organization of this system is finally interpreted and transformed as a decision taking into account global indicators based on topological analysis (mean position, mean agents speed,...) and level of constraint estimation (global energy, statistical physics...). Figure 1 shows an overview of the system.

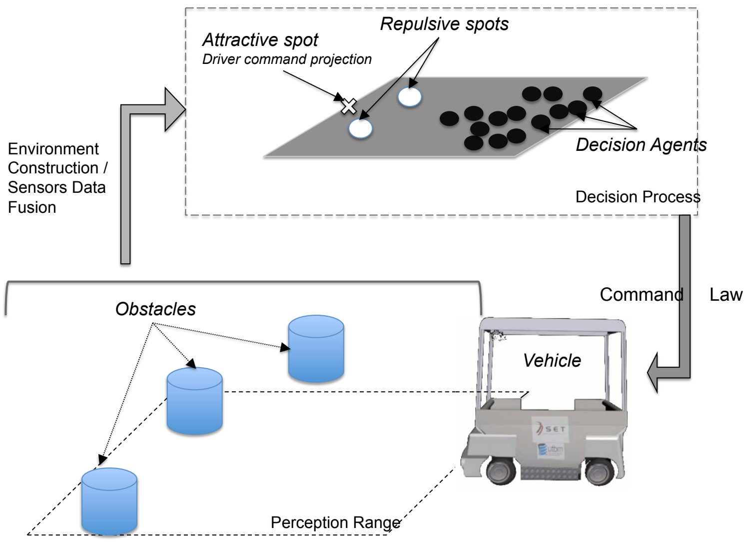

Figure 1: System overview

2.2. Environnement, Agents and Interactions

- **Environment**.
  As state before, agents environment is the corner stone of the approach. It links vehicle's world and the decision process mechanism. Generally, agents environments are described through their topology and their dynamics. In the approach presented in this paper, environment topology is defined as the merged boundaries of all available vehicle sensors. The dynamics is directly linked to obstacles dynamics. For each appearing real obstacle, a new repulsive spot is created. Spots characteristics depend on real obstacle properties: spot position is deduced by merging sensors information and repulsion coefficient, denoted R_o is determined taking into account real obstacle size. As explain in previous section, driver command is translated into an attractive point in agents environment the characteristics of which are a horizontal axis position depending on steering angle, vertical axis position linked to speed requirement and attractive coefficient, denoted A_{Dc}, computed taking into account the distance between the current command sent to the vehicle and driver command.

- **Decision Agents**
  In this context, agents can be considered as small mass particles, the behavior of which is to reach the minimum of a potential field. This potential field is created by attractive/repulsive spots and by the decision agents themselves. Indeed, each decision agent induce a repulsion field so as to keep agents as far as possible from each other. Initially, agents are spread randomly in the virtual environment. There number is taken as the square root of the surface of the environment. This number is generally taken in reactive multi-agent applications. It is high enough to obtain significant emerging results and low enough to avoid agent jam. Agents parameters are the following:

Parameter	Description
m	mass of the decision agent
(p_x, p_y)	agent position in x and y
(v_x, v_y)	agent speed in x and y
(a_x, a_y)	agent acceleration in x and y
P_R	Perception Range

Table 1: Decision Agent Parameters

- **Interactions**
Interactions are inspired by physics. Three types of interaction are defined: (i) Interaction between Agents (ii) Interaction between Agents and Obstacles[1] (Repulsive spots) (iii) Interaction between Agents and Driver Command (Attractive spot)

  - *Agent-Agent Interaction*: Agent-Agent interaction can be considered as a simple Newtonnian repulsion force in $1/d^2$. Its expression is the following considering two decision agents i and j situated at positions A_i and A_j : $\vec{R_{ij}} = R_A.m_i.m_j \frac{\vec{A_iA_j}}{\left\|\vec{A_iA_j}\right\|^3}$ for each decision agent A_i such as $\left\|\vec{A_iA_j}\right\| < P_R$. In this equation, m_i and m_j are respectively the mass of agents i and j. In our case, since the population is homogeneous, the mass is the same for each. This force is applied to decision agent j due to the presence of decision agent i. Then, for one decision agent j all the forces induced by other decision agents can be summarized following this equation:

$$\begin{cases} R_{X_j} = \sum_i \left(R_A.m_i.m_j \frac{(x_j - x_i)}{((y_j-y_i)^2+(x_j-x_i)^2)^{\frac{3}{2}}} \right) \\ R_{Y_j} = \sum_i \left(R_A.m_i.m_j \frac{(y_j - y_i)}{((y_j-y_i)^2+(x_j-x_i)^2)^{\frac{3}{2}}} \right) \end{cases} \quad with\ i\ such\ as\ i \neq j\ and\ d_{ij} < P_R \tag{1}$$

  (d_{ij} is the distance between Agents i and j)

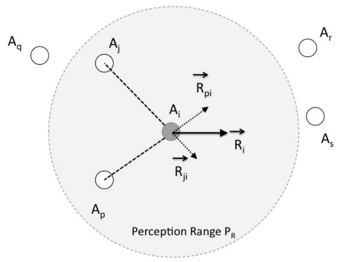

Figure 2: Agent-Agent repulsion behavior

  Figure 2 shows repulsion behavior induced by agents A_j (force $\vec{R_{ji}}$) and A_p (force $\vec{R_{pi}}$) on agent A_i. The global repulsive behavior is given by the force denoted $\vec{R_i}$. Agents A_j and A_p are in the perception range of agent A_i. Agents A_q, A_r and A_s that are out of the perception range have no influence on agent A_i.

  - *Agent-Obstacle Interaction*: Agent-Obstacle Interaction is nearly the same as the previously defined agent-agent interaction. Its expression is the following involving decision agent i and an obstacle denoted o : $\vec{R_{io}} = R_o.m_i.m_o \frac{\vec{OA_i}}{\left\|\vec{OA_i}\right\|^3}$ for each obstacle o such as $\left\|\vec{A_iO}\right\| < P_R$. In this equation, m_i and m_o are respectively the mass of the agents i and the mass of the obstacle o. This last is defined taking into account obstacle size. Then, for one decision agent i all the forces induced by obstacles can be summarized following this equation:

$$\begin{cases} R_{X_io} = \sum_o \left(R_o.m_i.m_o \frac{(x_i - x_o)}{((y_i-y_o)^2+(x_i-x_o)^2)^{\frac{3}{2}}} \right) \\ R_{Y_io} = \sum_o \left(R_o.m_i.m_o \frac{(y_i - y_o)}{((y_i-y_o)^2+(x_i-x_o)^2)^{\frac{3}{2}}} \right) \end{cases} \quad with\ o\ such\ as\ d_{io} < P_R \tag{2}$$

[1]Including environment boundaries

- *Agent-Driver Command Interaction*: Agent to attractive spot interaction is defined as a simple linear attraction force defined as follows: $\vec{A_{iD_c}} = A_{D_c}.m_i.\vec{A_iD_c}$.

- **Decision agent behavior computation**
  Position, speed and acceleration for each agent are computed considering agents environment to be continuous. Agents' dynamical characteristics are computed following the laws of the classical Newtonian physics. Each behavior applied to an agent corresponds to a force influencing its movement.
  By applying the fundamental law of dynamics, acceleration of each agent can be computed. Here, $\vec{\gamma}$ represents acceleration, m agent mass, and $\vec{F_b}$ the force resulting from behavior b: $\vec{\gamma} = \frac{1}{m}\sum_{behaviors}\vec{F_b}$. By substituting in the definition all the forces, collecting terms in the velocity vector $\vec{V}$, and integrating twice we obtain the following equation:

$$\vec{X_i}(t) = \vec{X_i}(t-1) + \left(\vec{V_i}(t-1)\delta t + \frac{(\delta t)^2}{2m}\left(\vec{R_{io}} + \vec{R_i} + \vec{A_{iD_c}}\right)\right) \; with \; \vec{X_i(t)} = \begin{pmatrix} x_i(t) \\ y_i(t) \end{pmatrix} \tag{3}$$

2.3. Evaluation of global properties

The emerging organization of this system is the result of agents-agents and agents-environment interactions. This organization takes into account both obstacles (position and dynamics) and driver command. From now on, organization characteristics must be transformed into a command law (steering and speed) to be sent to the vehicle. To that way, global indicators based on topological analysis (mean position, mean agents speed,...) and level of constraint estimation (statistical physics) are used.

- Mean position.

  The easiest way to compute a global indicator for an agents population is to find out the mean position of agents. This mean position allows to have information on the global configuration of the environment (detected obstacles, driver command,...)

- Mean speed.

  The agents mean speed computation allows to retrieve information on environment global configuration dynamics. If this configuration change only a little, the mean speed will stay nearly the same (equivalent to the mean speed of a Brownian process). By contrast, if the global configuration changes a lot, due to the the appearance of a new detected obstacle for instance, the mean speed will drop to another value.

- Statistical physics.

  A global estimation approach inspired by statistical physics and thermodynamics has been developed in [3]. This approach can be considered as a way to link the microscopic and a macroscopic points of view in agents system. The statistical physics based method gives an important role to partition function Z which is computed thanks to energetic measurement. From this mathematical function can be extracted indicators that represent the global evaluation of the system state based on local phenomena and constraints evaluation. According to the interaction model, the energy measurement can be detailed as follow :

  – **Kinetic energy**: . In the following equation, the agent i is represented by its mass m_i and its speed $\vec{V_i}$: $E_K = \frac{1}{2}m_i\vec{V_i}.\vec{V_i}$
  – **Potential energy**: it is computed, for agent i, using the classical expression of the energy U : $U = \delta W + \delta Q$ where δW represents the work done on the system and δQ the heat flow (here, $\delta Q = 0$ since no heat is dissipate). The work done on the system δW is expressed considering a conservative force (Cf. equation 4) with $\vec{du}$ a unit vector in the direction of agent speed.

$$E_p = \delta W = \vec{F}_{total}.\vec{du} = \vec{F}_{Obstacles}.\vec{du} + \vec{F}_{Agents}.\vec{du} + \vec{F}_{Driver}.\vec{du} \tag{4}$$

Agent's energy corresponds to the sum of the kinetic and the potential energy : $E = E_k + E_p$. From now, the free energy could be computed :

$$\begin{cases} A(T, V, N_i) = -ln(Z) \\ Z = e^{-\beta E} \end{cases} \tag{5}$$

with T the temperature, V the volume and N_i number of element. From now on, thermodynamic potentials A is defined. Thus, the system indicators can be computed [9]. The indicator A study allows to deduce some properties about system evolution. For instance, given a system in an initial non equilibrium state, if it evolves to an equilibrium state, the negative of the difference in the Helmholtz energy is equal to the maximum amount of work and the function A is minimized at equilibrium.

2.4. Decision vector computation

Decision vector computation takes into account the indicators described above. Mean position will give the global decision vector according to the relative position of the vehicle in decision agent environment. This decision vector consists in an orientation (angle) and a speed (value) to be applied to the vehicle. Decision vector angle is then modified according to mean speed value. So as to make this angle modification, a combination mean vector $\vec{M}$ between mean position vector $\vec{M}_p$ and mean speed vector $\vec{M}_s$ is made. Then, combination mean vector $\vec{M}$ norm is set to the former mean speed position vector norm following this equation (cf. Figure 3):

$$\vec{M} = \left(\vec{M}_p + \frac{\vec{M}_s}{\sqrt{nb_A}}\right) \frac{\left\|\vec{M}_p\right\|}{\left\|\left(\vec{M}_p + \frac{\vec{M}_s}{\sqrt{nb_A}}\right)\right\|} \tag{6}$$

with nb_A the number of decision agents.

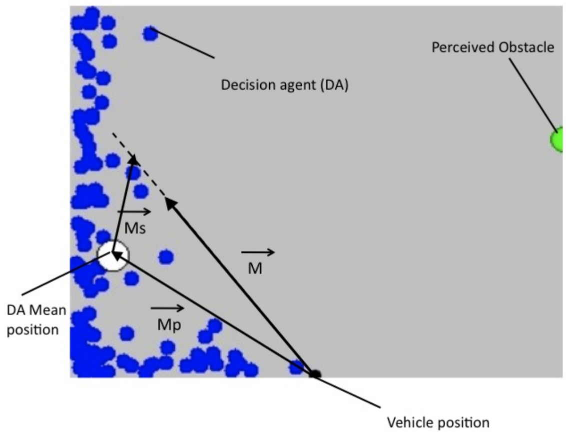

Figure 3: Decision Vector computation

Finally, decision vector $\vec{D}$ is computed according to free energy value following this equation:

$$\vec{D} = (1 + \kappa.A).\vec{M} \tag{7}$$

with κ a ponderation factor. This vector is then applied as a command law to the vehicle.

3. Experimentations

The model described in the previous section has been implemented thanks to the MadKit multiagent platform[2]. Experimentation are used to validate some model characteristics.

3.1. Vehicle architecture

The validation of the proposed reactive multi-agent system has been realized on real vehicle. The System and Transportation Laboratory has two electric vehicles (cf. figure 4 on the left). These vehicles have been automated and can be controlled by an onboard system.

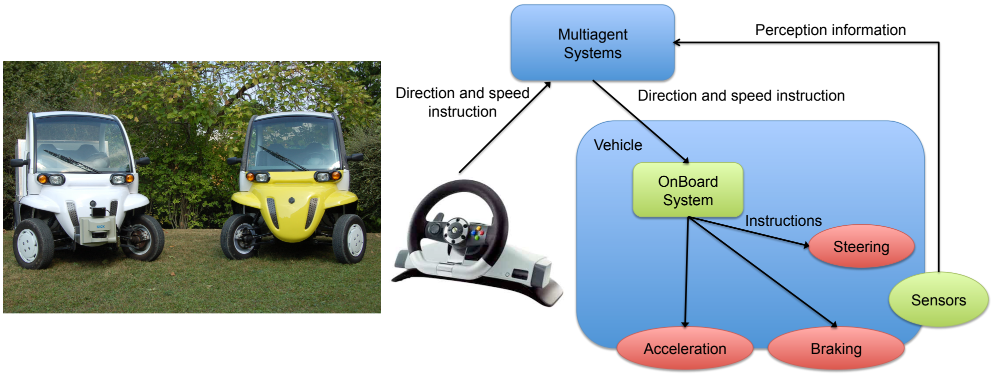

Figure 4: SeT laboratory electrical vehicle (left), vehicle architecture (right)

Figure 4 shows the vehicle system onboard. An onboard computer receives direction and speed instructions from the driver (numerical steering wheel) and sends direction and speed intructions to the vehicle control. The reactive multi-agent system has been implemented on this onboard computer. The next subsection proposed to validate the Car-driving assistance in two cases: simple or several obstacles on the road.

3.2. Simple obstacle

The first experiment is to evaluate the path taken by the vehicle, in the case of the appearance of a single obstacle (figure 5 on the left). The figure illustrates the configuration adopted for testing the simple obstacle avoidance, i.e a single object on the floor. The driver in this experiment give a straight line direction instructions and a constant speed (8 km/h). In this case, if instruction is respected, the vehicle run directly into obstacle.

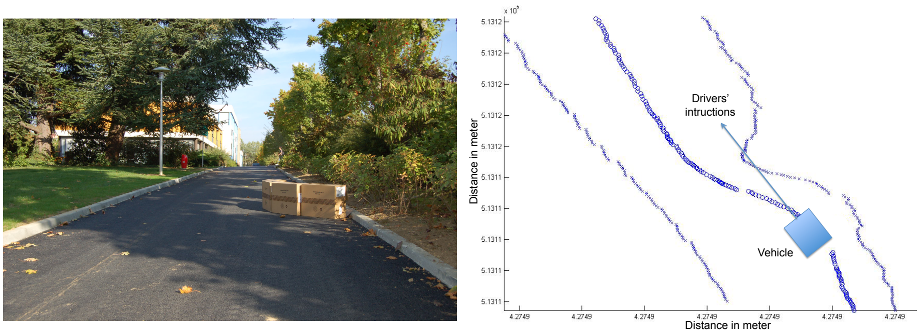

Figure 5: Experimentation configuration (left) path results from the obstacle avoidance (right)

[2]Madkit5, http://www.madkit.org

Figure 5 (on the right) shows vehicle trajectory during the experimentation. The driver brings about a collision and the reactive multi-agent system on the vehicle onboard system allows to avoid obstacle on the road. This case studies the onboard system pre-fill driver's direction intructions.

3.3. Multiple obstacles

The goal of this experiment is to point out the evolution of vehicle speed depending on the constraints involved by perceived obstacles. Figure 6 (on the left) shows the obstacle configuration. In this case study, the driver gives a straight line direction instructions and a constant speed (8 m/s).

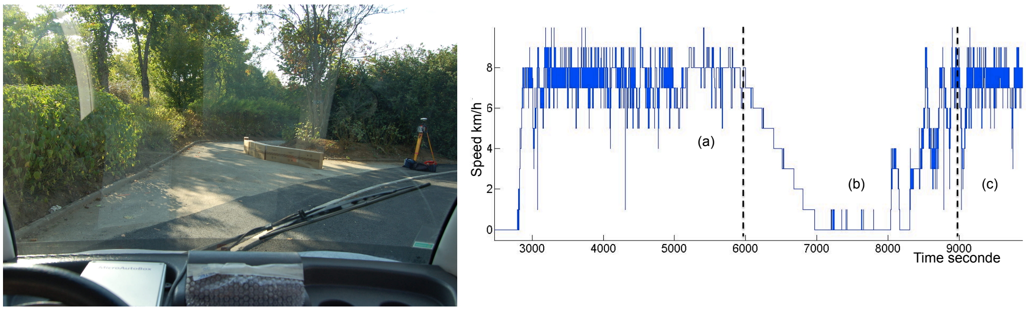

Figure 6: Experimentation configuration (left) vehicle speed variation (right)

Figure 6 on the right shows vehicle speed evolution along the previous trajectory. The first part (denoted (a)) of this curve corresponds to a vehicle speed without any detected obstacle. In this part, vehicle reaches its maximum speed. As soon as detected obstacles are put into decision agents environment (part denoted (b)), vehicle speed starts to decrease due to the statistical physics correction factor (cf. equation 7). Indeed, when there are many obstacles, Statistical physics potential is increasing. This involves an amplification of decision vector trend. When, vehicle has passed the obstacle line (part denoted (c)), speed increases until its maximum value.

4. Conclusion

This paper presents a car-driving assistance based on a reactive multi-agent system. This decision process relies on two main elements: the decision-agent environment on the one side and the global state estimator of the agent population on the other. Decision agent environment merges the information that stems from the sensors and turn it into forces aimed at influencing decision agents behaviors. It can thus be considered as the input layer of the designed decision process. By contrast, the global state estimator is the output of the system. It allows to build up decision considering valuable estimation of the global emerging phenomenon.
Experimentation exhibited that the decision system is reliable and has auto-adaptation capabilities. These allow pre-fill driver's intruction in order to avoid obstacle and stop an imminent collision.
In order to continue these research, we are now working on several key points. First, development are made to integrate data from several sensors in order to be able to deal several points of view (i.e. taking into account information on sidewalk given by a specifically oriented sensor). Finally, we also plan to modify this application to be able to use it as autonomous driving.

5. Bibliography

[1] O. Simonin and F. Gechter, *An environment-based principle to design reactive multi-agent systems for problem solving*, Environments for Multiagent Systems II, extended and selected paper from E4MAS'05 workshop, Springer-Verlag Berlin Heidelberg, LNAI 3830 pp. 32-49, 2006

[2] N. Gaud, S. Galland, F. Gechter, V. Hilaire, A. Koukam, *Holonic multilevel simulation of complex systems: Application to real-time pedestrians simulation in virtual urban environment*, Simulation Modelling Practice and Theory 16(10): 1659-1676 (2008)

[3] J.M. Contet, F. Gechter, P. Gruer and Abder Koukam, *Evaluation of global system state thanks to local phenomenona*, proceedings of the 18th European Conference on Artificial Intelligence ECAI 2008: pp. 865-866

[4] J. C. Latombe, *Robot Motion Planning*. Norwell, MA: Kluwer, 1991.
[5] J. Borenstein and Y. Koren, *The Vector Field Histogram – Fast Obstacle-Avoidance for Mobile Robots*, IEEE Journal of Robotics and Automation, Vol. 7, No. 3., June 1991, pp. 278-288.
[6] A. Masoud, S. Masoud, and M. Bayoumi, *Robot navigation using a pressure generated mechanical stress field, the biharmonical potential approach*, in Proc. IEEE Int. Conf. Robotics and Automation, San Diego, CA, 1994, pp. 124-129.
[7] R. Simmons, *The curvature-velocity method for local obstacle avoidance*, in Proc. IEEE Int. Conf. Robotics and Automation, Minneapolis, MN, 1996, pp. 3375-3382.
[8] J. Minguez, L. Montano., *Nearness Diagram Navigation (ND): Collision Avoidance in Troublesome Scenarios*, IEEE Transactions on Robotics and Automation Vol 20 N 1, pp 45-59, february 2004.
[9] R. Balian *From Microphysics to Macrophysics*, Springer 2007.