Multiagent System Model for Vehicle Platooning with Merge and Split Capabilities

Jean-Michel Contet, Franck Gechter, Pablo Gruer, Abder Koukam
Systems and Transportation Laboratory (SET),
University of Technology of Belfort-Montbéliard (UTBM), Belfort, France,
jean-michel.contet@utbm.fr, franck.gechter@utbm.fr, pablo.gruer@utbm.fr, abder.koukam@utbm.fr

Abstract

This paper presents a Multiagent based approach to the vehicle platooning problem. Our model is an alternative to centralized control solutions. In our approach, vehicles are autonomous entities in mutual interaction. The behaviour of each vehicle is determined from a physics inspired model that minimizes interactions: each vehicle relates only with the preceding one in the platoon. This allows for the emergence of all required collective behaviours: steady platoon motion over arbitrary trajectories, vehicle merging and splitting. Furthermore, a set of safety properties which are crucial to the viability of the application, such as stability of inter-vehicular distance, are demonstrated thanks to the intrinsic properties of the physics inspired model. Other aspects such as trajectory matching and merge/split behaviour have been validated by both simulation and experimentation with mini-robots.

Keywords: platoon, reactive multiagent, longitudinal and lateral control, merge and split, physics inspired behaviour model

1 Introduction

Since twenty years, many works have been done in research projects like PATH [1] to increase traffic safety and efficiency using automatic vehicles. These vehicles have a perception of the environment, and provide automated assistances (speed and distance control, obstacles detection, platooning, automatic car parking). Thanks to New Information and Communication Technologies (NICT) previous research can be put into practice in modern transportation. These public transportation systems provide new services such as car sharing or public open access to vehicle units. Collaborative driving is a research domain which aims to design automated vehicles that collaborate in order to navigate through traffic. Among this kind of systems, platoons are formed by a virtual train of semi-automated vehicles.

The main problem related to platoon systems consists in controlling the global platoon geometry: inter-vehicular distance and trajectory matching. Generally, the platoon control system splits up into longitudinal control and lateral control. Longitudinal control is one of the main aspects to be worked out. It consists in controlling braking and acceleration in order to stabilise the distance between the leader vehicle and the follower. This control takes as parameter the distance between the preceding and following vehicle or the time which separates these vehicles depending on the model used. *Sheikholeslam and Desoer* [2] proposed longitudinal control using linearization methods. *Ioannou and Xu* [3] controlled the brakes and acceleration by fixed gain PID control (Proportional, Integral, Derivative) with gain scheduling. *Hendrick et al* [4] used a control mode based on a non linear method with PID. *Lee and Kim* [5] proposed a longitudinal control by fuzzy logic. In related work, the following of robot by fuzzy logic is proposed in [6]. Lateral control consists in aligning the vehicle direction in relation to the preceding vehicle. *Daviet, Parent* [7] proposed lateral control by a PID controller. This control consists in keeping close to zero the angle between the preceding and the following vehicle. In order to model vehicle platoon systems with longitudinal and lateral control, *Gehrig and Stein* [8] inspired on physical particles submissive forces. *Soo-yeong Yi and Kil-to Chong* [9] represented immaterial fixing with an impedance control model. *Simon Halle* [10] used Multiagent System (MAS) in order to model immaterial vehicles fixing using constant values from [7].

In this work, we aim to introduce a multi-agent system that provides two functionalities. On one hand, longitudinal and lateral control and on the other hand, merge and split capabilities. Multiagent systems have become an attractive approach for problem-solving and have been used to a wide range of applications and simulations [11], such as localizing and tracking targets [12]. Among the classical models, the reactive approach is one of the most interesting due to its robustness, adaptability and simplicity. In this article, MAS are composed by reactive agents. In our model, as an alternative to centralized control solutions, vehicles are autonomous entities in mutual interaction. The behaviour of each vehicle is determined from a physics inspired model that minimises interactions:

each vehicle relates only with the preceding one in the platoon. This allows the emergence of all required collective behaviours: steady platoon motion over arbitrary trajectories, vehicle merging and splitting.

This paper is organized as follows. First part presents the main aspects of agent behavioural models, in an independent fashion, relatively to the intended control model. Second part is dedicated to the impedance-control model. Then, section four deals with merging and splitting. Finally, experimental results are presented and discussed.

2 The Multiagent Model

2.1 Roles and interactions

In order to define the multi agent model, we first identify different roles. A role is a set of abstract behaviours, each characterized by a set of interactions. The vehicle platoon application includes two main roles: vehicle and platoon. Vehicle role specializes towards two sub-roles: head vehicle and follower vehicle. The head vehicle role interacts with the environment (road). The follower role interacts with another vehicle and with the environment. The nature of interactions can be diverse: explicit communication or simple mutual perception. In our case it depends on the physics-inspired model presented in a later session.

2.2 Agent's behaviour

We use the language of statecharts as the formalism for the specification of agent's behaviour. Statecharts are widely used in behavioural descriptions, because of their expressive power and well founded semantics. This has been the reason why we adopted the statecharts in the frame of our activity on Multiagent system's formal specification and verification [13].

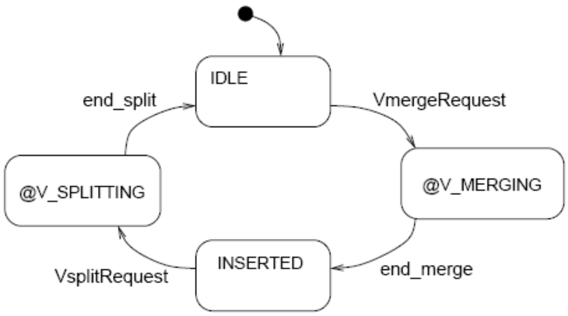

Figure 1 : Vehicle agent behaviour

Figures 1 and 2 describe respectively the *Vehicle* and *Platoon* role behaviours at a high level of abstraction. State names that begin by an ampersand character include a sub-statechart, which describes the behaviour of a sub-role. The follower vehicle (*Vehicle* for short) role behaviour is formed by a cycle of four sub-roles representing vehicle at rest, merging with the platoon, inserting in the platoon and splitting from it. The *Platoon* role behaviour represents three global situations: the platoon is steady, a vehicle is merging with the platoon and a vehicle is splitting from it.

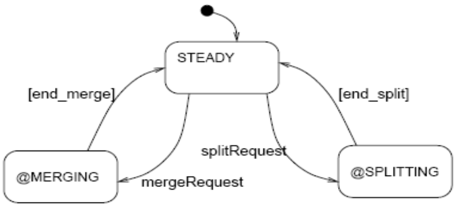

Figure 2 : Platoon agent behaviour

According to this hierarchy, Figures 3 and 4 describe the *MERGING* and *SPLITTING* sub-role behaviours of *Vehicle* and *Platoon* roles. The *MERGING* sub-role for the Vehicle (*Vehicle: MERGING*) waits for the detection of the last vehicle in the platoon, which will be his followed vehicle. Then, the merging process begins.

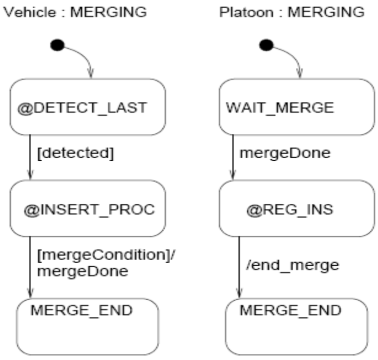

Figure 3 : The MERGING sub-roles

The merging process consists in first detecting the condition to insert, then launching the merging procedure. The mergeCondition is verified when the following distance is the required one to merge a new vehicle without danger. When the mergeCondition is satisfied, the *Platoon:MERGING* process is made aware and refreshes the values of a set of attributes, to take into account the merging at the platoon level.

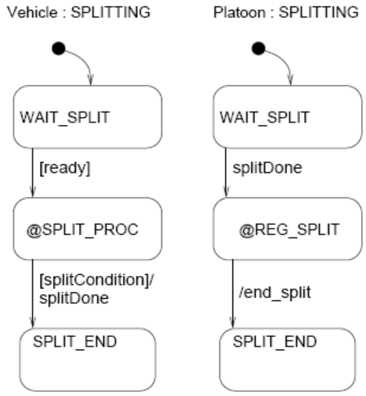

Figure 4 : The SPLITTING sub-roles

The SPLITTING sub-roles (figure 4) consists in first be ready to split, then launching the splitting procedure. The splitCondition is verified when the distance is convenient in order to split a vehicle out of the train. When the splitCondition is satisfied, the *Platoon:SPLITTING* process is made aware and refreshes the values of a set of attributes, to take into account the splitting at the platoon level.

The functions performed in the V_MERGING, V_SPLITTING and INSERTED states of figure 1 are based on algorithms defined from an interaction model. The following section presents the one that we adopt.

3 Interaction Between a Vehicle and the Preceding one

3.1 Interaction model

The connection between two vehicles is made by an impedance-control model like in [9]. The virtual link connection between each vehicle is a spring damper with stiffness *k*, damping *h* and spring's non stretched length l_0. These forces are composed to exerted force by the spring F_s, the shock absorber force F_a and the friction force of the surface F_f. The interaction with the environment accounts for friction parameters λ. Figure 5 shows a 3-vehicle part of the considering platoon. Each vehicle *i* is represented by its position $\vec{X_i} = [x_i, y_i]$. The mass of the vehicle is denoted as *m*. The distance between vehicles is:

$$d = \|\vec{X}_{n+1} - \vec{X}_n\|$$

Spring force $F_{s:}$

$$\vec{F}_s = -k(\|\vec{X}_{n+1} - \vec{X}_n\| - l_0)\vec{u_{n+1\;n}} \tag{1}$$

Shock absorber force $F_{a:}$

$$\vec{F}_a = -h(\dot{\vec{X}}_{n+1} - \dot{\vec{X}}_n) \tag{2}$$

Friction force of the surface $F_{f:}$

$$\vec{F}_f = -\lambda \dot{\vec{X}} \tag{3}$$

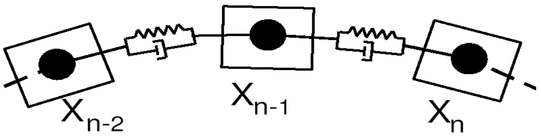

Figure 5 : Simplification of forces applied to the vehicle.

From the Newton's law of motion, in case of equilibrium of forces, relatively to the preceding vehicle:

$$m * \vec{\gamma} = \sum_{force} \vec{F_i} \tag{4}$$

From this equation, acceleration γ can be computed. By discrete integration, we can then determine the speed and the vehicle orientation angle[1], which are taken as reference signals for vehicle motion.

3.2 Interest of the model

The physics inspired model is used to specify a MAS. Each vehicle is represented by a reactive agent. The behaviour of a reactive vehicle agent is calculated from agent-environment and agent-agent interactions and perceptions. The physics inspired model presented in the previous section is similar to the one presented in [9]. The difference with previous work consists in the fact that we take into account only a local point of view. For each vehicle agent, perception is limited to an estimation of relative position of preceding-vehicle. Action is decided using only this perception. Since global stability is achieved from the local actions of each agent, it becomes important to establish a proof of the emergent global stability. We can verify that the new model is stable by following an approach based on an energy variation.

3.3 Proof of stabilization

The proof of vehicle stability is made by considering energy, which is composed of kinetic and potential energies. We take vehicle *n* as reference frame to express the energy knowing that the head vehicle index is 0. So $\vec{X}$ represents the difference between the distance separating vehicles *n-1* and *n* and the spring's non stretched length. Also $\dot{\vec{X}}$ expresses the speed difference between vehicles *n-1* and *n* (figure 5).

$$E = E_{pot} + E_{kin} = \frac{1}{2} * m * (\dot{\vec{X}}^t . \dot{\vec{X}}) + \frac{1}{2} * k * (\vec{X}^t . \vec{X}). \tag{5}$$

however,

$$\frac{dE}{dt} = \frac{\partial E}{\partial \vec{X}} * \frac{\partial \vec{X}}{\partial t} + \frac{\partial E}{\partial \dot{\vec{X}}} * \frac{\partial \dot{\vec{X}}}{\partial t} \tag{6}$$

so, since $\frac{\partial E}{\partial \vec{X}} = k * \vec{X}$ and $\frac{\partial E}{\partial \dot{\vec{X}}} = m * \dot{\vec{X}}$, we obtain

$$m * \ddot{\vec{X}} = -k * \vec{X} - h * \dot{\vec{X}} \tag{8}$$

thus,

$$\frac{dE}{dt} = k * \vec{X}^t * \dot{\vec{X}} + \dot{\vec{X}} * (-k * \vec{X} - \xi * \dot{\vec{X}}) \tag{9}$$

finally,

$$\frac{dE}{dt} = -h * \dot{\vec{X}}^t . \dot{\vec{X}} \tag{10}$$

With h the coefficient of shock absorber, h is a positive number. So, if we take the limit,

---

[1] The choice of a command law takes into account the characteristics of a test vehicle used in our laboratory

$$\lim_{t\to\infty} E = 0 \qquad\qquad E > 0 \tag{11}$$

Consequently,

$$\lim_{t\to\infty} \vec{X} = \vec{0} \qquad\qquad \lim_{t\to\infty} \dot{\vec{X}} = \vec{0} \tag{12}$$

Thus, the derivative of energy is always negative (10) because energy will decrease until becoming null and energy E are the sum of two positive number (11). Consequently, ad infinitum, we will have the kinetic energy reaching zero like the potential energy. Thus relative speed between each couple of agent will become null (12) because of the kinetic energy. And the distance between vehicle will be constant since the potential energy is zero and X become null. In consequence agents will be followed at same speed and distance between agents will converge to the spring's non stretched length.

3.4 Parameters of the model

This model uses five variables, mass *m*, coefficient of viscous friction λ, stiffness *k*, damping *h*, and spring's non stretched length l_0. The mass of agent vehicle is set by the mass of real vehicle. However, other parameters must be set too. In order to find the values of parameters or a boundary, we used the law expressed in (4). We can deduce a differential equation of second degree.

$$m\ddot{\vec{X}}(t) = -h\dot{\vec{X}}(t) - \lambda\dot{\vec{X}}(t) - k\vec{X}(t) + kl_0 - h\dot{\vec{X}} \tag{13}$$

This equation can be solved by putting it on a point form. And we can discuss from the level damping.

$$\Delta = \omega_0^2 * (\varepsilon^2 - 1) \tag{14}$$

with

$$\varepsilon = \frac{h+\lambda}{m*2*\omega_0} \qquad\qquad \omega_0 = \sqrt{\frac{k}{m}} \tag{15}$$

In order to have an absorbed system, pseudo periodical damping must be negative ($\Delta < 0$) thus ε must be under value of one ($\varepsilon < 1$). So we can deduce an over estimate of variables.

$$\frac{h+\lambda}{2+m*\sqrt{m*k}} < 1 \tag{16}$$

The last parameter is spring's non stretched length l_0, its value is the safety distance. This distance depends of the safety stop. According to the schedule equation, we obtain:

$$l_0 = \frac{-V_0^2}{2*\gamma} \tag{17}$$

4 Vehicle Merging and Splitting

The model presented in this article include merge and split capabilities. It has been implemented on the base of a MAS architecture. Naturally, vehicle agents include a set of attributes. Attribute values can be changed as a consequence of inter-agent communication, i.e. by a message from another agent vehicle. When a vehicle receives message, it forwards this message to the preceding or following vehicle and refreshes its own attributes value. Particularly, this mechanism is used to increment the mass weighting attribute value in order to increase the spring force during the merge or split phase.

4.1 Merging

The Merge phase is described by the statechart presented in figure 3. The merging phase agrees results in a new vehicle being added at the end of train. The merging vehicle is initially parked and waits for the train. When the merge vehicle detects the last platoon vehicle, it follows it by applying the impedance control model. If the merging vehicle is close enough to the last train vehicle, it sends a message to the preceding vehicle with new weighting and index.

4.2 Splitting

Split phase is described by the statechart of figure 4. Any vehicle can split from train. When a vehicle wants to split, it waits for car parking. When vehicle is close enough to the car parking, it sends new weighting and index to the following and preceding vehicle. Then, it splits from the vehicle train.

5 Experimentations

Experiments have been made with the same parameter values in order to draw a parallel between computer simulation and robot application.

m	λ	k	h	l_0
500 kg	200 Nm^2kg^{-2}	400 Nm^{-1}	10 kgs^{-1}	50 mm

Table 6 : parameter values

These parameters have been chosen in order to respect (16) and (17).

5.1 Computer simulation with the Madkit platform

The model described in the previous section has been implemented thanks to the multi-agent platform[2] proposed by *J. Ferber* and *O. Guknecht* [14]. Computer simulation is used to validate some model characteristics. The simulation runs with a platoon of 4 following vehicles. The first vehicle follows a preset trajectory: a square with rounded angles.

---

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

Regular trajectory error: the simulation of home stretch has shown that the error in following is below the millimetre. Figure 6 illustrates a case of rotational motion, showing the leader and follower trajectories. It shows an increase in trajectory error from any vehicle to its follower. This is due to the impedance control model, in which each vehicle anticipates the position of the following one. It can be completed in order to deal with this problem. This error can be scaled up towards real vehicle length: 1.2 m compared to the 2 m length of a vehicle.

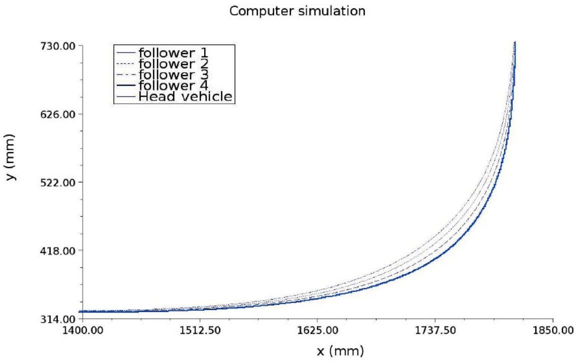

Figure 7 : Computer simulation

Obstacle avoidance: figure 7 illustrates a case of obstacle avoidance. The leading platoon vehicle avoids an obstacle on the road: we can see that all following vehicles also avoid this obstacle and preserve the platoon structure.

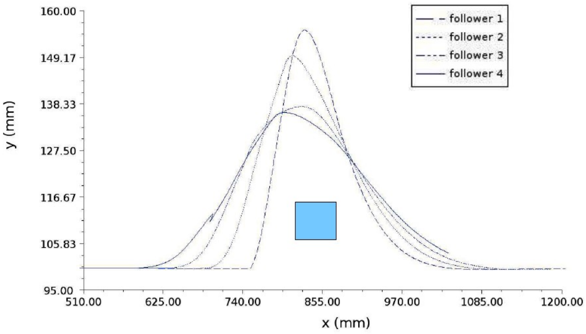

Figure 8 : Obstacle avoidance

Vehicle merging: the merging process was simulated in order to visualise the duration of transitory phase of distance stabilization, as illustrated by figure 8.

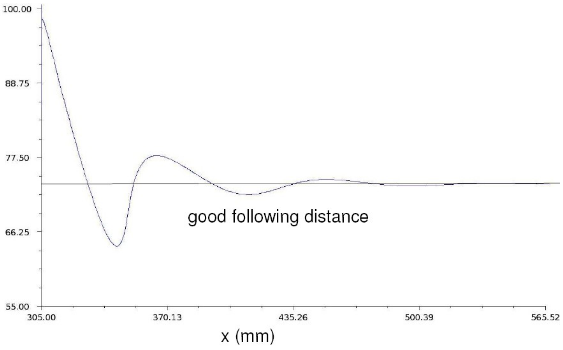

Figure 9 : Safety length

The graph shows the expected pseudo-periodical damping. We can see that the damping time is below 7 seconds and the maximal deviation from the safety length is 10 % of the non stretched spring length.

5.2 Real experimentation with the robot soccer platform

Experiments have been made to test the controller model under real-world conditions. These real experiments were done with a robot-soccer platform. As for the real vehicle, some small 2-wheel drive Mirosot6[3] soccer robots have been used (cf. figure 9). These robots move on a playground, the size of which is 2.20 m by 1.80 m. Robots are controlled by a standard PC computer that sends data to each robot through a RF interface. The perception is performed by a CCD camera placed above the playground (cf. figure 7). A standard PC computer has been used to execute the SMA software.

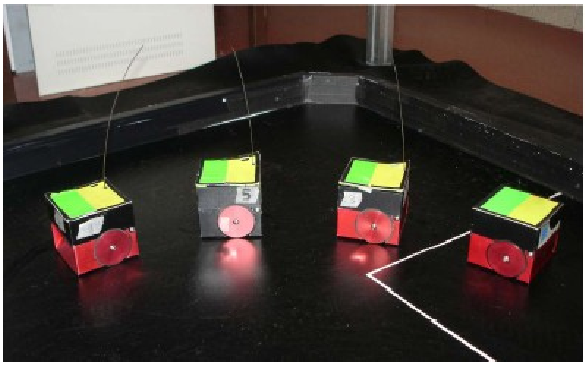

Figure 10 : robot soccer platform

This experiment allows us to check the model flexibility and adaptability. For instance, we can over estimate the errors by half a vehicle length (i.e. 35 mm). A platoon system with two vehicles was simulated. As in the previous experiments, the first vehicle follows a preset trajectory. Figure 10 shows platoon motion at high speed (more than 50 km/h, if scaled up to real vehicle size) and the effect of centrifugal force on the following vehicle, in the curved sections of the trajectory.

---

[3] http://www.merlinrobotics.co.uk/merlinrobotics/

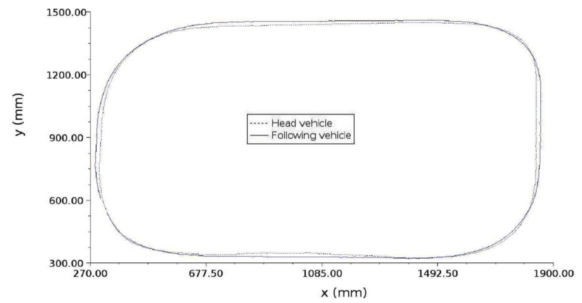

Figure 11 : Real simulation

A platoon system with two following vehicles was simulated. The errors can be scaled up from robot size to real vehicle size. The resulting value is 2.5 m compare to the 2 m length of the vehicle.

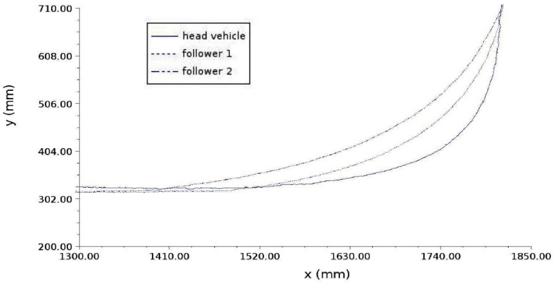

Figure 12 : Real simulation

6 Conclusion

The model proposed in this paper consists in a MAS-oriented approach to vehicle platoons, where vehicles agents interact using laws inspired by physics. This model is based on simple agents with neither cognitive abilities nor representations of the collective goal. One of the interesting points is the emergent structure obtained. The use of physics inspired forces enables an easier tuning of the behavioural parameters and the demonstration of stability. These experiments allowed to demonstrate the essential characteristics of this kind of resolution method: *Flexibility/Adaptability* (clinging and avoidance capabilities) and *Reliability* (low error rate in curve and damping time). With the real simulation, we note that the system admits the speed constraints and curve radius of town roads. We are now working on the use of formal models in order to prove some application properties and to ensure a zero default embedded software design for real prototypes. Moreover, we are also advancing into further research on other interaction models.

7 References

[1] J.K. Hedrick, M. Tomizuka and P. Varaiya, Control issues in automated highway systems, *IEEE Control Systems Magazine*, 14(6), pp 21- 32 (1994).

[2] S. Sheikholeslam and C.A. Desoer, Longitudinal control of a platoon of vehicles with no communication of lead vehicle information: A system level study, *IEEE Transactions on Vehicular Technology*, 42(4), pp 546–554 (1993).

[3] P. Ioannou and Z. Xu, Throttle and brake control systems for automatic vehicle following, *IVHS Journal*, 1(4), pp 345 (1994).

[4] J. Moskwa and J. Karl Hedrick, Nonlinear algorithms for automotive engine control, *IEEE Control Systems Magazine*, 10(3), pp 88-93 (1990).

[5] H. Lee and M. Tomizuka, Adaptive vehicle traction force control for intelligent vehicle highway systems (ivhss), *IEEE Transactions on Industrial Electronics*, 50(1), pp 37-47 (2003).

[6] M. Lee, M. Jung and J. Kim, “Evolutionary programming-based fuzzy logic path planner and follower for mobile robots”, *Proceedings of the IEEE Conference on Evolutionary Computation, ICEC*, (1) pp 139-144 (2000).

[7] P. Daviet and M. Parent, “Longitudinal and lateral servoing of vehicles in a platoon”, *IEEE Intelligent Vehicles Symposium, Proceedings*, Tokyo, Jpn, pp 41-46 (1996).

[8] S.K. Gehrig and F.J. Stein, “Elastic bands to enhance vehicle following”, *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, Oakland, CA, pp 597-602 (2001).

[9] S. Yi and K. Chong, Impedance control for a vehicle platoon system, *Mechatronics (UK)*, 15(5), pp 627-38 (2005).

[10] S. Halle and B. Chaib-draa, A collaborative driving system based on multiagent modelling and simulations, *Transp. Res. C, Emerg. Technol. (UK)*, 13(4), pp 320-45 (2005).

[11] H.V.D. Parunak, ”go to the ant”: Engineering principles from natural multi-agent systems, *Annals of Operations Research*, (75), pp 69 (1997).

[12] F. Gechter, V. Chevrier and F. Charpillet, “Localizing and tracking targets with a reactive multi-agent system”, *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2004*, New York, NY, United States, 3, pp 1490-1491 (2004).

[13] P. Gruer, V. Hilaire, A. Koukam and P. Rovarini, Heterogeneous formal specification based on object-z and statecharts : semantics and verification, *Journal of Systems and Software*, 70(1-2), pp 95–105 (2004).

[14] O. Gutknecht and J. Ferber, Madkit, “A generic multi-agent platform”, *Proceedings of the International Conference on Autonomous Agents*, Barcelona, Spain, pp 78 -79 (2000).