Proceedings of the
2007 IEEE Intelligent Vehicles Symposium
Istanbul, Turkey, June 13-15, 2007

A Platform for Designing Solutions for Automotive Autonomous Driving based on Evolutionary Robotics approach

F. Vicentini, M. Branciforte, R. Martorana, A. Rovetta

***Abstract*— We consider the autonomous driving of real vehicles as a challenging framework where Robotics-oriented approach might be effective in solve complex tasks. Here we discuss some of the autonomous navigation features in order to set up an environment to develop neural control systems for automotive autonomous driving. In this work we introduce the Evolutionary Robotics as a framework for automatically evolve and select neuro-controllers to test in a conceptual experiment reproducing the road scenario features. Preliminary experiments of Cruise Control and Adaptive Cruise Control were conducted in simulation, comparing the results of both our simulator and a commercial dedicated simulator providing a complete car library. The experiments help in evaluating the feasibility of novel approaches in the development of controllers for automotive vehicles.**

I. INTRODUCTION

In autonomous driving of vehicles, the accomplishment of tactical tasks, such as lane keeping/changing, speed setting, leading vehicles following, avoiding collisions, has been extensively studied as parts of a general solution in dynamically changing scenarios. Still, a complete automated driving is unavailable for cars as a large-scale equipment, with few exceptions like [1]. However, in such case of rough terrain navigation the required equipment happens to be quite demanding in terms of computational power and of devices. Nonetheless this kind of applications reveals the presence of next generation robot techniques in a pervasive and familiar field such as car driving. Hence, we observe a very promising convergence of the experience in mobile robotics research and the evolution of real vehicles towards autonomous driving. In this paper we consider the road scenarios and the basic navigation capabilities as a framework for discussing a Robotics-oriented approach in automotive autonomous driving. As a number of AI techniques have been developed to autonomously control vehicles or robots [2], we therefore discuss here the requirements, the feasibility and the preliminary results of the application of Evolutionary Robotics (ER, see [3]) in solving the navigation problem. Roughly speaking, ER is a methodological tool to automate the design of controllers. ER is based on the use of artificial evolution to find sets of parameters for artificial neural networks that guide agents to the accomplishment of their objective, avoiding dangers. One the most relevant features of ER is to allow the design of artificial neural network controllers for robots capable of facing circumstances never experienced during the training phase. The main purpose of our discussion is to look at the ER technique as a tool for addressing some of the most relevant problems in autonomous driving.

In addition to most of the related ER works ([4], [5] and related bibliographies) that deal with cognitive and social behaviors, other examples of real-time controls for navigation tasks have been successfully accomplished ([7], [8]). The common purpose of such techniques is to develop simple controllers, from both a computational and an architectural point of view. Although not extendable to all the possible sub-tasks of autonomous navigation of real vehicles, some interesting features of ER-based controllers are promisingly addressing the most relevant issues of autonomous driving.

As regards the related works, since the late nineties, a considerable part of autonomous driving is performed by vision-based systems (see one of the first successful experiments in [9]). Perception modules for image processing set the core information for lane or pedestrian detection in [10], [11], before planning the vehicle behavior. Some other solutions propose a mixed/extendable architecture of sensory level, after which an expert system perform an adaptive selection of the outputs to send to the operational control [12]. This latter work (exploited also in [13]) assumes the use of neural fields, i.e. a principle of neural network dynamics similar to that coded by the class of neural network proposed by the current work.

The paper is organized as follows: in session II we consider the autonomous navigation requirements addressed by the experimental framework. In particular we consider the problems of robustness against noise in signals and against environmental biases (unexperienced events). After setting a number of required features of a robust controller, we describe the subset of conditions to be preliminary tested and evaluated. Hence, in session III we introduce the preliminary work on autonomous vehicles, setting up the model and the experiments in order to face some environmental condition related to real vehicles. The validation of the tool and the approach are exploited through a standard procedural automotive simulator [14] able to reproduce real traffic conditions. In session IV some results are shown in order to highlight the feasibility of ER control in standard navigation manoeuvres, before setting the path towards any autonomous handling.

A. Rovetta and F. Vicentini are with Poliecnico di Milano, Mech. Dept., Robotics Lab alberto.rovetta@polimi.it and federico.vicentini@polimi.it

M. Branciforte and R. Martorana are with STMicroelectronics S.r.l., Advanced System Technology (AST), Stradale Primosole 50, Fab M6, 95121 Catania - Italy marco.branciforte@st.com and rosario.martorana@st.com

II. Framework for experiments on autonomous navigation

In autonomous driving the notion of navigation may be assumed as the capability of choosing direction and velocity (a, *kinematics*), controlling locomotion parameters on board of the real vehicle in real time mode (b, *dynamics*), detecting the environment features (c, *mapping*) and taking some tactical decision (d, *knowledge*). The collision avoidance, for instance, is a compulsory skill and achieved by all (a)-(d) capabilities. In partially autonomous driving, however or in assisted driving such as ADAS (Adaptive Driving Assistance Systems), some functions are neglected, such as the direction steering that is not up to the autonomous controller. In the following text the vehicle is referred as *agent* because the proposed approach is originally developed for robots. Unlike [1] or many other works with robots moving on rough terrain, we consider the road scenario as the main framework. The constraints are therefore not related to the landscape or terrain features, but are due to the conditions and rules of the traffic. As in related works (see seminal [15] and related issues), we set the experiments in a non-urban scenario for discussing some capabilities that an agent must have to perform an adapted and robust navigation.

From a modeling point of view, the environment is affected by a large uncertainty due to the variability of the mechanics represented. The road surface, for instance, may present different friction characteristics that affect the adherence of the contact between the agent and the ground. Hence, the physical phenomena are modelled with a certain amount of inherent approximation and a number of such model parameters are subject to large variance. The agent itself is subjected to a relative degree of approximation due the complexity of the real agent to be modeled. As long as the embodiment problem in autonomous robotics is critically relevant ([16], [17]) it may become disruptive whenever any solution developed by simulation is coded into a real controller. In an automotive framework this problem must be addressed in the selection of the simulator and in the portability of the developed solutions to the standard buses and controllers. The simulation itself is affected by the accuracy of the model especially in the required computational power and in relative impact of the modeled details (surface contact nonlinearity, for instance) over the feasibility of the solution. Under these general assumptions, we make use of the ER techniques considering simplified models of the agents populating the simulated environment. Moreover, we introduce the uncertainty of the environment and of the agent as ranges of values that may be assumed by a number of modeled mechanics. The simulated inputs, for instance, are only partially reproducing the real physics of the sensors applied, but a preliminary sampling analysis is used to affect the simulated signal by a random noise whose parameters are suitably verified in post processing, or validation phases. In this way the evolved controllers experience any disturbed condition during the development phase, and are therefore able to robustly face the real conditions. Similarly, the environmental features, like road conditions, are experienced by the agent during the simulation in a wide range of variability. Finally, as in the approach of [18], the other agents populating the environment (like cars in the same road) are part of a dynamically changing scenario, perceived by the agents only by its input channels. The right or acceptable behavior is therefore dependent on the particular set of interaction of the agent with everything else. For this reason, even if the mechanics of interactions are suitably described by known models, the combinatorial nature of local interaction agent-environment makes the coding of behavioral rules very hard to establish. Setting the ER simulated process, these environmental and procedural conditions are experienced by many neuro-controllers. In such a framework, in fact, the evolution selects throughout the generations the fittest controller that can guarantee the higher robustness in facing all the proposed (to the agent by the simulator) combinations of conditions. In such a manner, the capabilities of evolved neuro-controller are *emergent* and not predefined. In the current preliminary work, the ER technique is used in a batch mode, i.e. the selection of the fittest controller is simulated off-line, then the resulting solution is re-evaluated in order to check the validity of the controller with regard to the constraints.

As regards the hardware features, the agents are usually equipped with many sensors that generate a large amount of data to use during the control process. However, we here consider the case of only few noisy measures available for getting information from the environment coming from RADARs. The simulation environment developed represents a reconfigurable extra urban scenario where several vehicles are moving independently. The navigation behavior is based only upon the information from the onboard sensors and the local perception of the agent. No supervisor or hierarchical infrastructure can steer the agent behavior because the controller provide autonomously all the information for the behavior. Since collective navigation is likely to be the most common scenario in road environment, the capability of recognizing the behavior of the other agents and dynamically adapt the driving activity is fundamental for solving complex tasks.

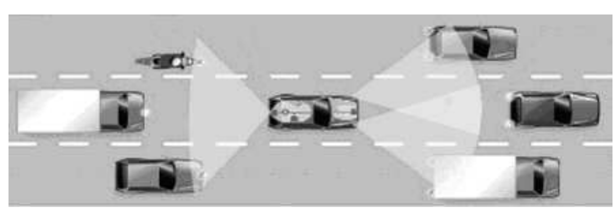

Fig. 1. Highway traffic scenario. The presence of other vehicles, the traffic constraints and the perception signals are translated in the experiment.

In such a framework, the evolved neuro-controller must manage different reactions to the detected situations. In the current preliminary experiments the main purpose is to reproduce basic navigation: a number of simplifications are taken in order to prepare the experimental framework for the attainment of the requirements previously stated.

III. Basic experiments

The choice of ER techniques is based upon the possibility to set an high level description of the desirable accomplishment of the task without detailing any hand coded rule for agent's behavior. The principle of the task is coded in a fitness function that describes the fundamental components of the desired behavior. This function is used to measure the degree of success of any controller that is evaluated. In this way the agents are left to evolve a fit strategy for accomplishing the task without a predefined sequence of rules. The tasks in the experiments are mainly focused on assisted driving applications (i.e. ADAS), in particular the Adaptive Cruise Control (ACC). In this approach, the neural controller is devoted to high-level functions (*dynamics* actions), i.e. setting the driving variables according to the elaboration of the current situation.

A. The agent

The agent (equipped with the ACC neuro-controller) is a vehicle equipped with a RADAR sensor to detect the other vehicles. The model of vehicle chosen for the experiments is a standard library car included in veDyna simulator. The control variables for the model are the engine torque and brake force. The RADAR has a distance range $\mathscr{R} = [30, 150]m$ and an azimuth angle of 15°. A typical ACC controller output is a required acceleration ([19], [20]). The model is provided with a transform function of the required acceleration into the engine torque and brake force. The vehicle model introduced is used to test the ACC controller on either real car or veDyna simulation. Recall that the transform function is parametrized on a particular real car. As introduced in II about modeling issues, and as being clearer in subsection III-B about the computational demands of the evolutionary process, the simplified model (with transform function of the vehicles dynamics) is first used for the development of the ACC controller and then validated with veDyna simulator, using the same controller's output (acceleration) to compare the results (vehicle acceleration and speed). As shown in figure 2 the acceleration of the vehicle in simplified model is well approximated by the veDyna complete vehicle. High order dynamics discrepancies, which depend on not modeled gear shifts, are beyond the purpose of the testing of the feasibility of the ER control in basic driving. However, the velocity response is almost equal for both simulators-

B. The controller and the evolutionary algorithm

A genetic algorithm is employed [21] to set the parameters of the controller (see Equation 1) and to select, generation after generation, the best performing one according to the fitness function score (see Equation 2). Every individual in each generation represents a controller.Then the agent is simulated during a limited lifespan in order to test its behavior due to the individual controller. Generations of individuals following the first one are produced by a combination of genetic operators, i.e. selection with elitism,

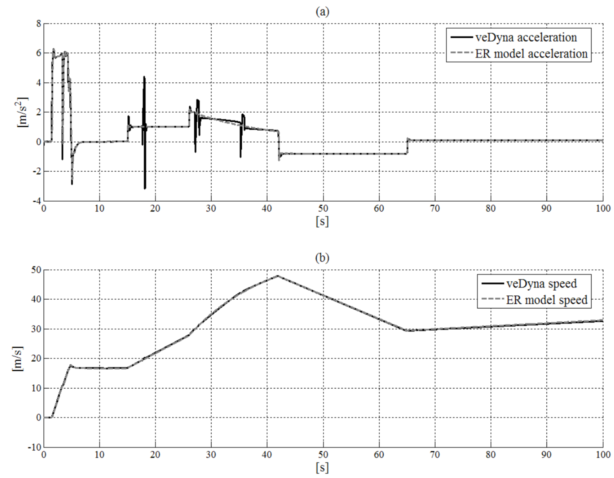

Fig. 2. Comparison between the response of model from veDyna and simplified stimulator given the same input. The veDnyna acceleration is well approximated by our model, the high order dynamics discrepancies are due to gear shifts (a). The velocities are almost equal (b).

recombination and mutation. The agent is equipped with controller that is made up of a feed-forward multilayer network (see figure 3). The network neurons are ruled by the following state equation, first introduced by [22]:

$$\tau_i \dot{x}_i = -x_i + \sum_{j=1}^{N} \omega_{ij} \sigma(x_j, \beta_j, g_j) + g_i I_i \qquad (1)$$

where $\sigma(x, \beta, g) = \frac{1}{1+e^{-g(x+\beta)}}$ and N is the number of neurons. This formulation is an extension of Hopfield networks [23]. The cell potential (x_i) of the 7^{th} neuron, mapped into [0,1] by the sigmoid function (σ), is then linearly scaled into $[-3m/s^2, 2m/s^2]$ [24] in order to set the requested vehicle acceleration. The following parameters are genetically

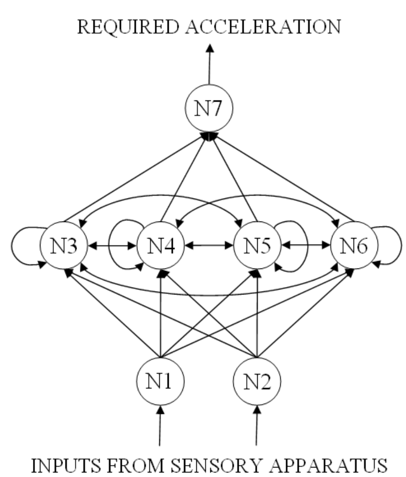

Fig. 3. The network topology.

encoded: (i) the strength of synaptic connections ω_{ij}; (ii) the decay constant τ_i of the inter-neurons N_3, N_4, N_5 and N_6; (iii) the bias term β_j and (iv) the gain term g_j for the neurons in the input and hidden layers. All the neurons of each layer share the same bias term and gain term for the firing rate function. The neuron N_7 has not-evolved $\beta_7 = 0.0$ and $g_7 = 1.0$. The decay constant τ_i of the sensory neurons and of the output neurons are set equal to dt (see below). Cell potentials are set to 0 any time the network is initialized or reset, and circuits are integrated using the forward Euler method with an integration step-size of $dt = 0.01$s.

C. The task

The chosen basic navigation task in the framework of assisted driving systems (ADAS) is the test of an Adaptive Cruise Control (ACC). A simulated vehicle for ACC is equipped with a RADAR to measure the current distance from vehicles ahead, as well as the speed sensor is considered. The task is established by the fitness function as a principle to get the desired behavior, that is to keep the reference speed whenever the distance from the proceeding car is computed to be safe, otherwise to slow down the car until the safe distance is reached and maintained. The main task is analyzed and modeled splitting the desired behavior into 2 different subtasks: *cruise speed* and *safe distance* control (see session IV), that are coded into the global fitness function. As a result it was possible to obtain a single neural controller which satisfies both the subtasks. This controller has 2 input (see figure 3): the curent velocity and the safe distance errors. The inputs are acquired by a model of real noisy sensors, in order to let the neural network to develop a filtering and pattern recognition capability.

D. The environment

As introduced in II, the simulation environment suitable of evolving a robust neural controller using an evolutionary optimization must be a able to reproduce real traffic situations in combinatorial way. The developed simulator is based on a single *host vehicle* provided of ACC controller and a number of *n target vehicles*. The performance of each individual, which represents a single NN controller, has to be evaluated with different conditions of traffic and road geometry (*evaluations* for each individual in IV description). The simulator randomly generates the number of *target vehicles*, their speed profiles and number of lane change. The handling manoeuvres are fixed and not up to the controller in the current experiment. A generic road geometry with 9 parameters is used to generate the test track for each evaluation (see figure 4). All possible variables (i.e. target vehicle speed profile, curve radius, etc.) are chosen complying real situations. In order to limit the number of input nodes of the NN controller, the range covered by the RADARs is split in 6 areas (3 for Short Range and 3 for Long Range RADAR) and for each area only one target vehicle is considered with its distance, angle of view and relative velocity (see figure 5) in manage eventual overlapping in the sensing. The measures of distance, view angle and relative velocity are biased with a Gaussian distributed noise. Moreover two sources of error are take in account: false alarms and fault detections. For each area of the RADAR range, 3 measures are recorded (distance, angle and relative velocity). In our preliminary experiments we use the distance of *target vehicles* on the central area of long range RADAR out of the 18 records can be used as input to the NN controller.

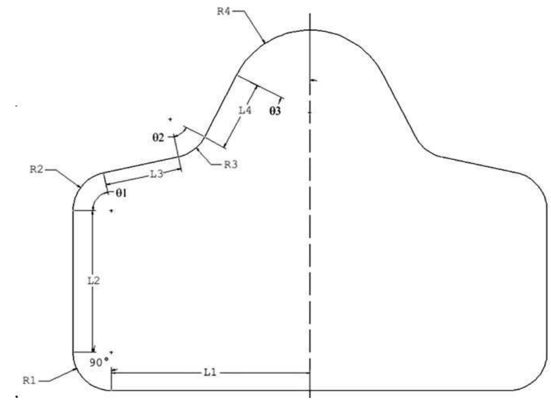

Fig. 4. Testing track where the vehicles move. With the 9 selected parameters, it possible to simulate almost every real extra urban situation.

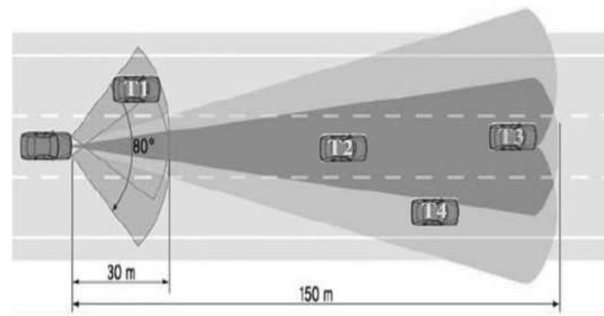

Fig. 5. Host vehicle is equipped with long range and short range RADAR. Six different areas are shown, for each of them just one vehicle can be detected (i.e. car *T3* is hidden by *T2*).

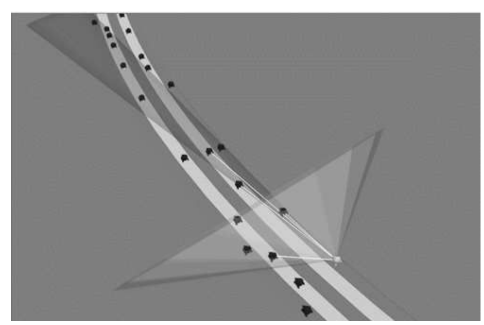

Fig. 6. 3D image from the environment simulator. It shows the radar equipped vehicle and several other vehicles moving independently.

E. The fitness function

According to the described features of main task (ACC), the complete fitness function used for evolutionary selection is made of 3 terms:

$$F = P\left\{\sum_{t=i}^{T} f_1(t_i) f_2(t_i)\right\} \quad (2)$$

where time steps are indexed by t , T is the final step of the trial. The F components are:

- $P = 0$ if any collision occurs during the trial. Otherwise $P = 1$;
- $f_1(t_i)$ rewards the *host* vehicle moving at speed near to the reference speed:

$$f_1(t_i) = 1 - tanh\left\{k \left| 1 - \frac{v(t_i)}{v_{ACC}} \right| \right\} \quad (3)$$

  where $v(t_i)$ is the vehicle speed, v_{ACC} is the reference speed and k is a severity parameter;
- $f_2(t_i)$ punishes the reduction of safe distance measured by the *host* vehicle:

$$f_2(t_i) = \begin{cases} 1 - tanh\{(d_s(t_i) - d(t_i))k\} & \text{if } d_s \geq d \\ 1 & \text{else} \end{cases} \quad (4)$$

  where $d_s(t_i)$ is the safe distance related to the current speed, $d(t_i)$ is the distance from the *target vehicle* and k is a severity parameter.

The fitness function F in Equation (2) is used to obtain the NN controller for the main task which includes the 2 subtasks. In the experimental set up, a NN controller for each subtasks was obtained. The equation (3) represents the fitness function used for the cruise speed control subtask. Instead the equation (4) slightly modified represents the fitness function for the safe distance control subtask.

IV. Experimental results

The first experiments regards the *Cruise Speed control*. The goal is to evolve a NN controller able to provide an acceleration required to reach and maintain a predefined cruise speed. The input of NN is the difference between the current vehicle speed and the cruise speed, the output is the required acceleration. The NN is composed of 1 input neuron, 1 output neuron and 4 hidden neurons and it is obtained after 200 generations, a population of 200 individuals, 20 evaluations for individual taking 100s of lifespan each. The results, depicted in figure 7 shows that NN is able to solve its task. Recall that NN controller is tested with the ER simulation environment and with veDyna simulator, the vehicle behaviour is the same in both cases.

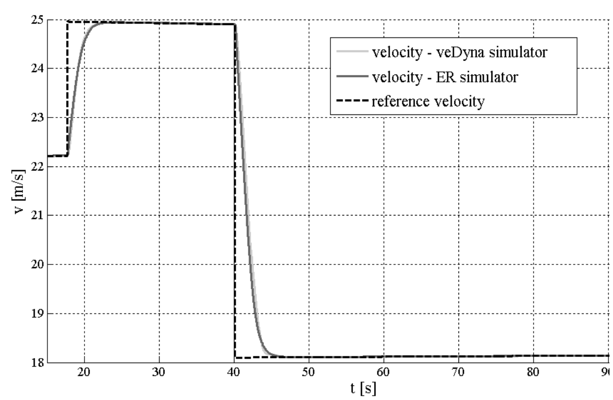

Fig. 7. The result of CruiseControl subtask is shown. *Velocity - veDyna simulator* is the host velocity coming from the accurate automotive simulator software veDyna; *Velocity - ER simulator* is the host velocity from the simulator used to obtain the NN control; *reference velocity* is the velocity that the host vehicle has to maintain.

The second experiment regards the *Safe Distance control*. The goal is to evolve a NN controller able to provide an acceleration required to reach and maintain a safe distance from the following vehicle considering a straight road and a single target vehicle. The input of NN is the difference between the current distance from the vehicle ahead and the safe distance, the output is the required acceleration. The NN structure and the evolution parameters are the same of the previous ones. Two important aspects have to be underlined. The NN controller is able to filter the noisy measurement from the RADAR and the host vehicle reaches the safe distance from a target which move at constant speed (see figure 8a). If the proceeding vehicle changes its speed the safe distance changes, in this case the NN controls the car to maintain the variable safe distance (see figure 8b).

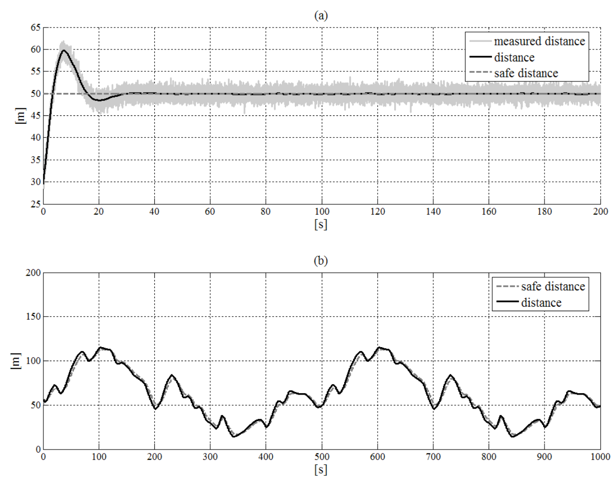

Fig. 8. The result of Safe Distance subtask is shown. *Measured distance* is the distance from the following vehicle measured by a noisy radar; *safe distance* is the distance that the host vehicle has to maintain; *distance* is the distance from the following vehicle. The NN control works with noisy measured distance (a). If the following vehicle changes its speed, the safe distance change consequently (b).

The third experiment regards a NN controller which is able to satisfy the main task, *ACC control*, re-evaluated in case of a straight road and a single target vehicle. The NN has to control the vehicle to maintain the cruise speed if there is not car in the RADAR range, otherwise the host vehicle has to maintain the safe distance. The NN has two input, one neuron for each (velocity and the safe distance error), 1 output neuron (acceleration request) and 4 hidden neurons. It is obtained after 200 generations, a population of 250 individuals, 20 evaluations taking 200s of lifespan each. The NN control meets the requirements of the task (see figure 9).

V. Conclusions

In this work we addressed the problem of designing a driving assistance system for automotive from a Robotics point of view. We therefore selected a subset of navigation features to study the possibility of application of a typically robot-oriented technique such as ER in automotive field. This

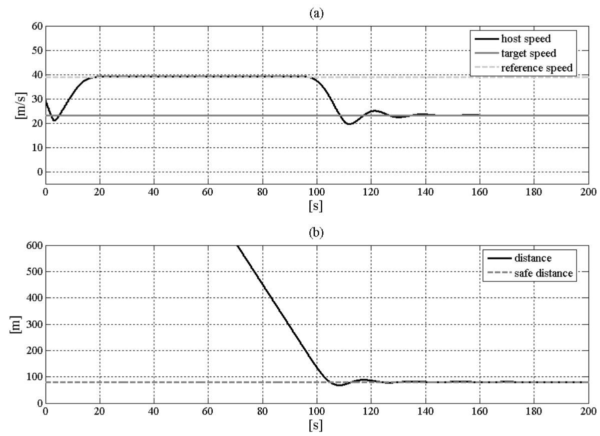

Fig. 9. The result of ACC task is shown. The host vehicle reaches and maintain the reference speed till about 80s (a). After this time the target vehicle gets in the radar range and the host vehicle reaches and maintains the safe distance (b).

is particularly relevant whenever considering the adoption of non hand-coding methodologies in designing the controllers for the vehicles.

We designed an environment scenario, which includes road geometry and traffic generation, as a framework for development of neuro-controllers in order to address that subset of requirements. We then designed three experiments that conceptually reproduces some basic task features.

Finally, we observed the capability of neuro-controllers, evolved with model that necessarily does not reproduce all the physical features, to achieve the assigned tasks in presence of unknown situation and noisy measures. The results are compared with those obtained with real a car-dedicated simulator, equipped with the same neuro-controller. In this way the capabilities of the neuro-controller in extending the navigational properties has been proved to suit a completely modeled vehicle. Hence, the simplified model used during the evolutionary phase does not hinder the possibility for the neural network to control a more complex agent in the re-evaluation phase.

As a future work we expect to enlarge the capabilities developed by ER considering more complex situations (i.e. tactical maneuvers like lane change, etc.) exploiting the features of our developed environment of simulation. We expect that the results coming from the preliminary observed capabilities of the controller, might improve the robustness of the application in order to set up some basic tactical experiments with real vehicles.

REFERENCES

[1] Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Hoffmann, G., Lau, K., Oakley, C., Palatucci, M., Pratt, V., Stang, P., Strohband, S., Dupont, C., Jendrossek, L.E., Koelen, C., Markey, C., Rummel, C., van Niekerk, J., Jensen, E., Alessandrini, P., Bradski, G., Davies, B., Ettinger, S., Kaehler, A., Nefian, A., Mahoney, P.: Winning the darpa grand challenge. Journal of Field Robotics (2006)

[2] Kolodko, J., Vlacic, L.: Cooperative autonomous driving at the intelligent control systems laboratory. Intelligent Systems, IEEE **18** (2003) 8–11

[3] Nolfi, S., Floreano, D.: Evolutionary Robotics: The Biology, Intelligence and Technology of Self-Organizing Machines. MIT Press, Cambridge, MA (2004)

[4] Baldassarre, G., Nolfi, S., Parisi, D.: Evolving mobile robots able to display collective behaviour. Artificial Life **9** (2003) 255–267

[5] Blynel, J., Floreano, D.: Levels of dynamics and adaptive behavior in evolutionary neural controllers. In: Proceedings of the 7^{th} international conference on simulation of adaptive behavior (SAB02) - From animals to animats, MIT Press Cambridge, MA, USA (2002) 272–281

[6] Tuci, E., Ampatzis, C., Vicentini, F., Dorigo, M.: Operational aspects of the evolved signalling behaviour in a group of cooperating and communicating robots. In et al., P.V., ed.: Symbol Grounding and Beyond: Proceedings of the Third International Workshop on the Emergence and Evolution of Linguistic Communication, Springer (2006) 113–127

[7] Quinn, M., Smith, L., Mayley, G., Husbands, P.: Evolving controllers for a homogeneous system of physical robots: Structured cooperation with minimal sensors. Phil. Trans. of the Royal Soc. of London, Series A **361** (2003) 2321–2344

[8] Vicentini, F., Tuci, E.: Scalability in evolved neurocontrollers that guide a swarm of robots in a navigation task. In Şahin, E., Spears, W., Winfield, A.F., eds.: Proceedings of the 9^{th} International Conference on the Simulation of Adaptive Behavior (SAB06). LNAI, Springer Verlag, Berlin, Germany (2007)

[9] Broggi, A., Bertozzi, M., Fascioli, A., Conte, G.: Automatic Vehicle Guidance: the Experience of ARGO Autonomous Vehicle. World Scientific Co. Publisher (1999)

[10] Bucher, T., Curio, C., Edelbrunner, J., Igel, C., Kastrup, D., Leefken, I., Lorenz, G., Steinhage, A., von Seelen, W.: Image processing and behavior planning for intelligent vehicles. IEEE Trans. Industrial Electronics **50** (2003) 62–75

[11] Ramstrom, O., Christensen, H.: A method for following of unmarked roads. In: IEEE Intelligent Vehicles. (2005) 650–655

[12] Edelbrunner, H., Handmann, U., Igel, C., Leefken, I., von Seelen, W.: Application and optimization of neural field dynamics for driver assistance. In: IEEE Proceedings Intelligent Transportation Systems. (2001) 309–314

[13] Pellecchia, A., Igel, C., Edelbrunner, J., Schoner, G.: Making driver modeling attractive. Intelligent Systems, IEEE [see also IEEE Intelligent Systems and Their Applications] **20** (2005) 8–12

[14] TESIS, ed.: veDyna 3.9 - User Manual. TESIS DYNAware (2005)

[15] Agogino, A., Goebel, K., Alag, S.: Intelligent sensor validation and sensor fusion for reliability and safety enhancement in vehicle control. Technical Report UCB-ITS-PRR-95-40, California Partners for Advanced Transit and Highways (PATH) (1995)

[16] Jakobi, N.: Half-baked, ad-hoc and noisy: Minimal simulations for evolutionary robotics. In Husbands, P., Harvey, I., eds.: Proceedings of the 6th European Conf. on Artificial Life, MIT Press, Cambridge, MA (1997) 348–357

[17] Watson, R.A., Ficici, S.G., Pollack, J.B.: Embodied evolution: Distributing an evolutionary algorithm in a population of robots. Robotics and Autonomous Systems **39** (2002) 1–18

[18] Beer, R.D.: A dynamical systems perspective on agent-environment interaction. Artificial Intelligence **1** (1995) 173–215

[19] Vibhor L. Bageshwar, W.L.G., Rajamani, R.: Model predictive control of transitional maneuvers for adaptive cruise control vehicles. IEEE Transaction on Vehicular Technology **53** (2004) 1573–1585

[20] Wang, J., Rajamani, R.: Should adaptive cruise-control systems be designed to maintain a constant time gap between vehicles? IEEE Transaction on Vehicular Technology **53** (2004) 1480–1490

[21] Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading, MA (1989)

[22] Beer, R.D.: On the dynamics of small continuous-time recurrent neural networks. Adaptive Behavior **4** (1995) 469–509

[23] Hopfield, J.J., Tank, D.W.: Computing with neural circuits: a model. Science **233** (1986) 625–633

[24] AA.VV.: ISO 15622 - Transport Information and Control Systems - Adaptive Cruise Control sysmtems - Preformance requirements and test procedures. International Standard (2002)