



**HAL**  
open science

## From autonomous robotics toward autonomous cars

Assia Belbachir, Rémi Boutteau, Pierre Merriaux, Jean-Marc Blosseville,  
Xavier Savatier

### ► To cite this version:

Assia Belbachir, Rémi Boutteau, Pierre Merriaux, Jean-Marc Blosseville, Xavier Savatier. From autonomous robotics toward autonomous cars. 2013 IEEE Intelligent Vehicles Symposium (IV), Jun 2013, Gold Coast City, Australia. 10.1109/IVS.2013.6629656 . hal-01710403

**HAL Id: hal-01710403**

**<https://hal.science/hal-01710403>**

Submitted on 15 Feb 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# From Autonomous Robotics Toward Autonomous Car

Assia Belbachir<sup>1</sup>, Rémi Bouteau<sup>1</sup>, Pierre Merriaux<sup>1</sup>, Jean-Marc Blosseville<sup>2</sup> and Xavier Savatier<sup>1</sup>

**Abstract**—For decades, scientists have dreamed of building autonomous cars that can drive without a human driver. Progress in this kind of research recently received an increasing attention in car industries. There are many autonomous car models recently developed. However, they are still infancy since they still lack efficiency and reliability. To obtain efficient and reliable systems, the validation process plays an important role. Nowadays, the validation is strongly related to the number of kilometers of drive. Thus, simulation techniques are used before going into real world driving. We focused our work on developing a methodology to smoothly move from simulation into real world car driving. We defined a versatile architecture that simplifies the evaluation of different types of algorithms. Several evaluation systems are shown and discussed.

## I. INTRODUCTION

Artificial Intelligence technologies are emerging from research laboratory into industry, particularly in the area of autonomous systems. Since 1980, this technology has been implemented in several work. The development of new technologies, such as car driving systems have shown some viability in controlling their own motion in highways and urban streets. In 1990, an experiment was investigated on a Californian highway, demonstrating that autonomous driving in a secured environment is possible. Since then, several attempts show that this field of research became very active.

In this purpose, the Defense Advanced Research Projects Agency (DARPA [1]) organized two challenges, in which several cars should drive autonomously. DARPA evaluated capabilities in different kinds of environments, urban and rural ones. Several cars were developed for this challenges [2] [3] [4]. This challenge consists of a strong step for algorithms evaluation, where the winner was the car that reaches first its target location.

In this growing research, our aim is to evaluate methodologically the developed algorithms to reach the final step of autonomous car driving. Thus, we defined three evaluation phases: simulation, hardware in the loop and large scale. First, we used a simulator to define different types of scenarios. Second, we implemented several developed algorithms from simulation phase, into hardware in the loop. In this phase, we used a robot called: Wifibot. This robot is armed with heterogeneous sensors. Wifibot is able to perceive its environment, localize itself, execute and control its path. To evaluate the implemented algorithms, VICON

Motion Systems<sup>1</sup> are used. VICON play an important role for localization purposes. Third, we implemented the algorithms in a real time autonomous car.

All these steps, should be versatile for any required system. This is why, we defined an architecture that can test any type of system and can easily pass from simulation into real world scale.

### A. Paper organization

In section 2, we will discuss the existing work on self-driving and the different methods applied in evaluation. In section 3, we will explain the used algorithms for self-driving cars. In section 4, we will describe our methodology to evaluate the implemented algorithms. In section 5, we will demonstrate experimental work that was done in our laboratory. We close the paper by a conclusion and future works.

## II. STATE OF THE ART

In robotics system, several achievement are required. Before implementing the developed algorithms in the robot, simulation is an important step. Thus, several simulators are developed. We can divide simulators into four main classes as follow: The first type of simulator is related to robot dynamic, such as CarMaker [5]. Using a model, the simulator try, as close as possible, to behave as the vehicle. FlexSci Defence Solutions developed the VDyna<sup>2</sup> is another simulation framework that simulates wheeled armed vehicles in real time for virtual testing, mission planning etc.

A second class is very close to the precedent one (simulating all the component part of a vehicle in its 3D environment), however it focussed on the sensing parts. ProSIVIC<sup>TM</sup><sup>3</sup> is a simulator that belongs mainly to this class. Its main function is its ability to simulate physical sensor capabilities especially perception sensors.

The third simulator type is related to traffic simulation capabilities. SUMO<sup>4</sup> (Simulation of Urban Mobility) simulates vehicle behavior according to macro traffic characteristics. In most cases, this kind of simulator assumes that the vehicle is a point on the road (without any dynamics).

The last simulator type is dedicated to the driving simulation such as SCANeRTM [6]. This simulator try to collect information from the driver/human behavior while using a simulated vehicle.

<sup>1</sup> Institut de Recherche en Systèmes Electroniques Embarqués (IRSEEM), Technopôle du Madrillet, Avenue Galilée BP10024, 76801 Saint Etienne Du Rouvray, FRANCE. email: firstname.lastname at esigelec.fr

<sup>2</sup>Laboratoire de Mesure de la Mobilité Coopérative (LEMCO-IFSTTAR), 25 allée des Marronniers – Satory, 78000 Versailles, FRANCE. firstname.lastname at ifsttar.fr

<sup>1</sup><http://www.vicon.com/>

<sup>2</sup><http://www.flexsci.com>

<sup>3</sup><http://www.civitec.com/>

<sup>4</sup><http://sumo.sourceforge.net/>

The used algorithms for evaluation are not enough to prove 100% that the algorithms will be successful in real world experiments. There is no previous literature on how to pass from simulation into real world. Thus, in general to evaluate the developed algorithms, user studies are developed or competitions. DARPA [1] is a kind of algorithms evaluation in which the winner is the car that reaches first the ending point. Talos [4] is an example of a car driving that was designed to handle DARPA Urban Challenge requirements. This robot is able to perceive and navigate in a road network with segments defined by sparse waypoints. For this kind of navigation, Talos uses several sensors such as LIDAR (SICK), radars, Velodyne (3D laser scanner composed of 64 laser layers.), cameras and a GPS. For the same challenge another robot called Navlab11 [3] has been developed. This robot is a test bed vehicle fitted with several sensors and functions used for robot navigation, obstacle avoidance, road following and robot localization.

In our context, we want to have a system that is be able to control longitudinally and laterally a car such as the DARPA robot cars. Moreover, we want to offer a versatile architecture able to easily move from simulation into real world scale.

### III. IMPLEMENTED ALGORITHMS FOR AUTONOMOUS CAR DRIVING

In our work, we implemented different algorithms, one of them is related to the path planning. Knowing the actual position (Init<sub>*i*</sub> *x,y*) and the required position (End *x,y*), the tested path planning finds the next position (init<sub>*i+1*</sub> *x,y*) that reduces the distance between the final position and the actual one and takes into account the vehicle constraints (e.g. maximal wheel angle). At the end the vehicle obtains a list of roads that the vehicle can follow to reach its point. The shortest and fastest path is selected. The speed is not imposed at any point of the trajectory. A maximum longitudinal acceleration or deceleration and a maximum lateral acceleration is allowed. The speed value is recalculated continuously. Using the information from the map, we can determine any future speed limits, intersections required to stop the vehicle and sharp length curvature as the maximum lateral acceleration. If none of the above information is present, the speed limit on the road is used. PID type control is used for speed control.

To follow the trajectory, the vehicle computes its position in the map. Then, it computes the adequate speed to reach the next point. This step takes into account different criterion such as driver comfort, obstacle avoidance and curves following.

### IV. METHODOLOGY TO EVALUATE THE DEVELOPED ALGORITHMS

#### A. Our interest

Computer simulations are very important to train and test any robotic system. However, it is very hard to find an adequate model of the real world system (e.g. road shape, vehicle dynamic, weather climate). In our case, we have chosen a simulator that can integrate the vehicle dynamic models. In addition, the simulator will create different

weather changes and generate several roads. However, real world simulation have several constraints such as:

- 1) The physical laws that govern the system are not taken into account, such as mass, weight and friction.
- 2) The sensors perceive the environment with noise, where simulators are usually omitting the perception noise.
- 3) The vehicle or the environment model is quite far from the real world.

After the simulation step, the system should be implemented into specific real world tests (two steps are developed. The first one called hardware in the loop and the second called large scale real world). In most cases, it is challenging and time consuming to move from simulation into real world evaluation. For facility, we encapsulated our algorithms using a developing tool.

The hardware in the loop step of testing has several advantages as shown bellow:

- 1) Gain in time. The use of a small scale environment and robot, allow us to test different types of scenarios. We can also fuse simulation and real world.
- 2) Reduce the development cost. The used sensors and the environment have less cost than the ones used in car driving. In the case the algorithm fails its task, the material damage is not as important as in an autonomous car.
- 3) Measurable environment. The area of the test is still under control, such as the environment light, which we can control the light. The position of the vehicle can also be known with a quite precision using VICON motion system. VICON eases us to evaluate the developed algorithms for location.

In large scale real world evaluation step, we have implemented our algorithms under autonomous car.

#### B. Simulation tools

In our experiments we are using Pro-SiVIC™, RTMaps and OpenDrive. Pro-SiVIC™ simulates the car behavior and generates different climate changes. RTMaps allows to develop and encapsulate our implemented algorithms. OpenDrive creates different types of road.

1) **Pro-SiVIC™**: Create scenarios for testing and disturbing the sensors such as rain, etc. Pro-SiVIC™ is developed in order to be independent of application type. *mg Engine* is the graphical 3D engine used. To reduce the computing board process, *mg Engine* uses a tree of binary positioning (BSP) [7]. To ensure its portability under numerous operating systems, the application is developed in C++ under LGPL with OpenGL and SDL libraries. In general several functionalities can be developed such as:

a. *Simulated sensors*: Several sensors can be simulated such as camera, inertial platform, odometer and telemeter.

- *Camera (module *sivicCamera*):* It simulates different sets of camera configured by using the Pro-SiVIC™ parameters or by using the parameters related to OpenGL.
- *Inertial Navigation System (module *sivicInertial*):* this module simulates the inertial sensor.
- *Odometer (module *sivicOdometer*):* It provides the distance covered by a vehicle.
- *Telemetric scanner (module *sivicTelemeter*):* This module simulates a laser scanner. Depending on the type of the telemeter, several methods can be implemented such as ray tracing.

*b. Vehicle model:* Three axes are defined : Roll, pitch and head. A generic model is able to reproduce the movement of the vehicle taking into account shock absorbers, viscosity and tie adherence [7]. In Pro-SiVIC™ other car models can be implemented and used from external libraries.

2) *RTMaps:* Multi-sensor systems are important in several domain such as robotics, security, virtual reality. RTMaps is a technology is a Real Time Multi-sensor Advanced Prototyping Software. It encapsulates the developed algorithms and interface them such as input and output. Each developed component acquires data asynchronously i.e. “on the flow”, each data sample being captured at its own genuine pace. Precise “Time Stamps” are assigned to every data which are then processed and/or recorded in so-called Synchronized Time-stamped Data Bases (STDB). The recorded data playback offers post-processing capabilities, such as off line algorithm development and flexible data logger design.

Data fusion algorithms can be developed to the real time capabilities of the software, the data time stamps and the Software Development Kit (SDK). Its intuitive graphical interface, RTMaps Studio, associated to a reliable and robust technology, engine and component library, makes RTMaps one of the most efficient tools.

3) *OpenDrive:* OpenDrive file provides a common base to describe track-based road networks. The data stored in an OpenDrive file offers a description of roads in terms of: line, circle, cycloid segments; it offers signal positions, lane speed limits, intersection description etc. . The format is organized in nodes which can be extended with user-defined data. By this, a high degree of specialization for individual applications is feasible while maintaining the commonality required for the exchange of data between different applications.

### C. Hardware in the loop prototyping (*hardware in the loop*)

For this evaluation step we are providing an environment with several sensors and wifibot.

1) *Environment description:* Laboratory of Autonomous Navigation (LNA) is an experimental room of around 15x10x5 meters. It contains 20 T40S-VICON cameras covering the whole room. VICON system is composed of different cameras, the controlling hardware module, the software to analyze the data, and the host computer to run the software. Using markers in the robot, and after calibration, VICON



Fig. 1. Laboratory of Autonomous Navigation

system is able to localize accurately the robot (error < 1mm). Figure 1 is a view of the experimental laboratory (LNA).

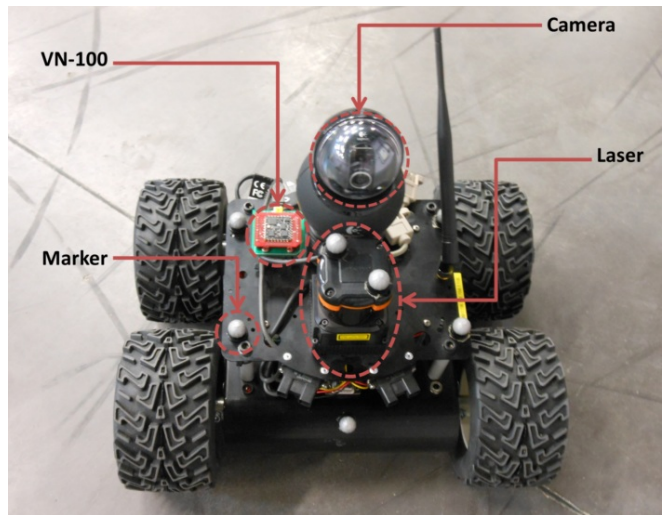


Fig. 2. Illustration of Wifibot with different sensors

2) *Robot description:* Wifibot is the name of our robot. It contains several sensors for localization and mapping. Different algorithms are implemented to extract information from the perception. Wifibot contains inertial measurement unit (VN-100), a camera and a laser.

The VN-100 SMD (Surface-Mount Device) is a miniature, surface mount, high-performance Inertial Measurement Unit (IMU) and Attitude Heading Reference System (AHRS). It combines an assortment of inertial sensors, including a 3-axis accelerometer, 3-axis gyroscope, 3-axis magnetometer, and a barometric pressure sensor. The camera is Logitech QuickCam Orbit AF Webcam (960-000111) with 2.0 Megapixel. With 30 meters and 270 degrees scanning range the laser UTM-30LX has an angular resolution of 0.25 degrees. Markers allow VICON Motion system to localize the robot in the environment.

#### D. Large scale prototyping (large real world)

Robert is the name of the robot that drives the car without inputs from human operator and it is also the name of the associated platform. Robert is implemented in a Renault Grand Espace (see figure 3). The car integrates a combination of different sensors: 4 cameras, GPS, Lidar and Ixsea landins (Inertial measurement unit). These cameras are mounted to provide a full 360 degree field of view. The first camera is mounted in front of the car for pedestrian/cars/roadsides detection. The other three cameras are in the left, right and back side of the roof box. All these sensors are linked to a perception computer. Another computer is used to run all lateral and longitudinal control algorithms.



Fig. 3. Equipped car for autonomous driving

Satory test tracks was used as a test network. On this network, several types of roads are represented: curves, straight lines, etc.

#### E. Versatile architecture

We defined an architecture that drive the vehicle into autonomous behavior. This architecture (Figure 4) is divided into four main parts as follow:

- Database: This part contains different security criterion, such as inter-distance of security between one vehicle and an obstacle. These criterion are defined by the user. A predetermined trajectory can be defined in the database, which can be executed by the vehicle. A Map of the environment can be utilized.
- Perception: each vehicle can contain different type of sensors. This part is responsible to extract the required information. We implemented the required algorithms to extract information from camera, inertial measurement unit and lidar.
- Path planning: This part is divided into: path finding and map matching. The Path finding computes the path of the vehicle, dealing with both the required speed and the future location of the vehicle. The Map Matching is responsible of matching the actual position of the vehicle in the map.
- Control: three types of control can be occurred. Longitudinal control depends only on the vehicle direction. Lateral control depends on the speed control. Lateral

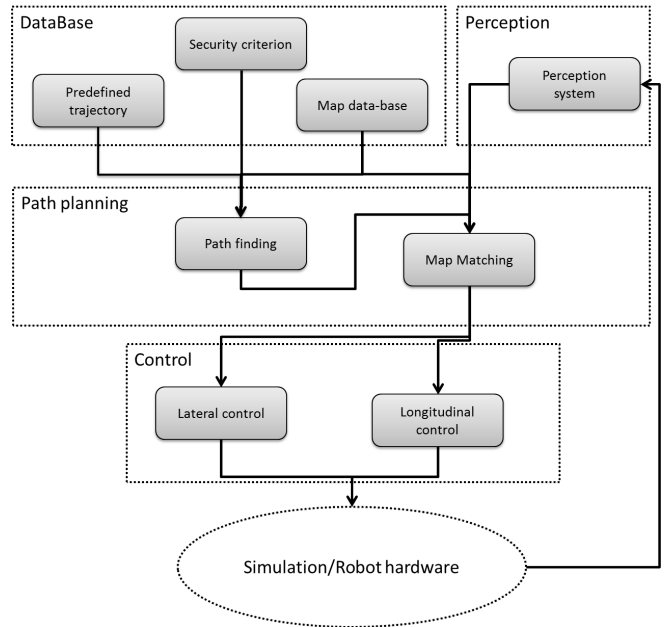


Fig. 4. Illustration of versatile architecture

and longitudinal control represent the autonomous vehicle control. All these types of control can be chosen by the user before or during the driving. This manipulation depends on what kind of evaluation do we want.

We can easily link this architecture into our simulator and into a real driving vehicle.

#### V. EXPERIMENTAL RESULTS

We evaluated our algorithm in different project. The first experiment concerns the evaluation of different type of sensors (camera, GPS-RTK combined with IMU, odometer.), where we generate a repeatable scenario. The second experiment evaluates the control part of the vehicle. A predefined path is embedded in the vehicle and the evaluation is a comparison between the predefined and the executed path.

##### A. Sensors evaluation

This project concerns the evaluation of different types of sensors. To evaluate the algorithms of perception, we followed three steps using our tools.

###### Step 1:

In the simulation part, we were able to implement and evaluate the camera model using several roads, climate changes. Several criterion of evaluation [8] has been defined to be able to evaluate the camera.

###### Step 2:

In this step we used different scenarios with our wifibot. The instance of the versatile architecture is shown as follow:

- Map matching: 1) calculation of the real vehicle position and orientation from the odometer, GPS-RTK and IMU outputs; 2) calculation of the vehicle position and orientation in the referential map;
- Lateral control: 1) computation of the distance ( $\epsilon$ ) to reach the nearest trajectory waypoint ( $\tau_i$ ); 2) calculation

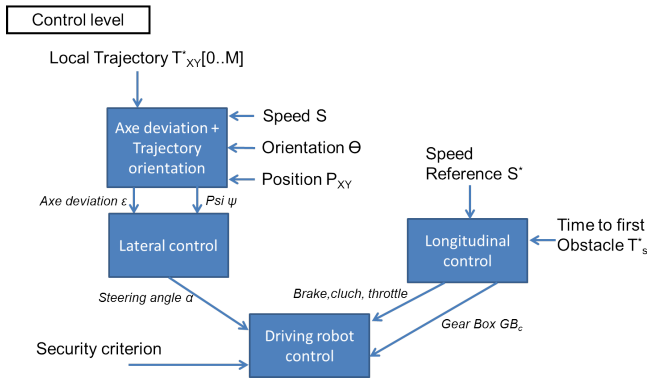


Fig. 5. Low level control algorithms

of the deviation angle  $\alpha$  between the trajectory tangent at  $(\tau_i)$  and the vehicle orientation; 3) computation of the steering angle  $(\psi)$  as a combination between  $\epsilon$  and  $\psi$ ;

- Longitudinal control: the robot allows a maximal acceleration and braking (1g). The speed value is either predefined or calculated taking into account the network characteristics (see path planning below). Robert used a PID controller for the speed control.

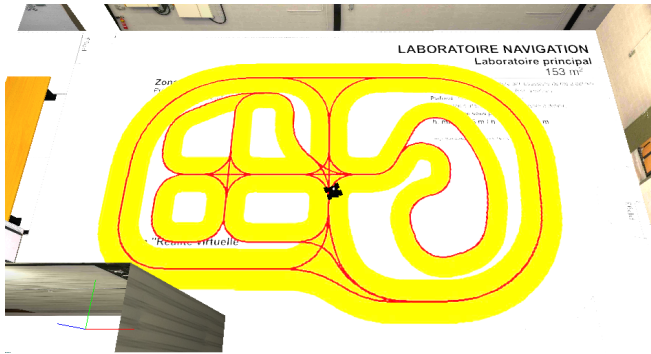


Fig. 6. Illustration of the virtual implemented road in LNA

- Short range path control: This component aims at re-considering the vehicle trajectory at short term. It could activate some lower level actions in case of emergency (e.g. emergency braking) or any reason to adapt the trajectory parameters ordered by higher levels. Robert can also host algorithms of this kind to be assessed.

Figure 5 shows the control layer that is used in the step 2 and 3.

In this step we can generate different types of road (see figure 6) and evaluate the existent sensors.

### Step 3:

The vehicle (See figure 3) whose perception capabilities should be assessed is moved automatically along a horse-ring track. At a certain point of the track, the vehicle is supposed to cross an obstacle.

The mock-up that figures the obstacle is shown in figure 7. This mock-up is made dynamic, moving transversally, crossing the vehicle trajectory as a pedestrian could make it



Fig. 7. Obstacle avoidance scenario

(from one side to the other side of the vehicle trajectory). The obstacle movement is triggered when the vehicle crosses an optical barrier located at a detection distance from the obstacle moving line. As well, the obstacle is moving on the opposite direction when the vehicle crosses a second optical line. The obstacle mock-up and its moving system are made of soft foam in order to avoid damages on the tested vehicle in case of collision.

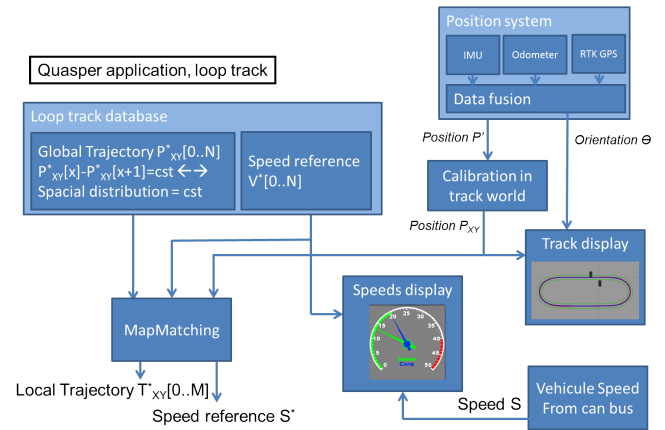


Fig. 8. Illustration of the path following

A successful sequence includes to start a controlled detection phase when the vehicle reaches the first optical barrier. At this point the obstacle is moved in the perception cone of the on-board sensors. In a second phase, the obstacle being detected, the vehicle is supposed to brake. In a third phase, the vehicle moves again on its horse-ring trajectory. Figure 8 shows a more detailed implemented architecture in the car.

During the whole operation, the robot is used to operate in a full automatic mode. This automation allows to make a great number of trials ensuring statistical significance to the assessment.

### B. Path planning and control evaluation

The components that are added in this experiment concern the path planning. From an initial and an ending point, the vehicle is able to plan its path. Figure 9 shows all the developed components. The used algorithm is the discussed algorithm of section III.

We tested the algorithm on simulation, hardware in the loop and large scale real world. The developed algorithm

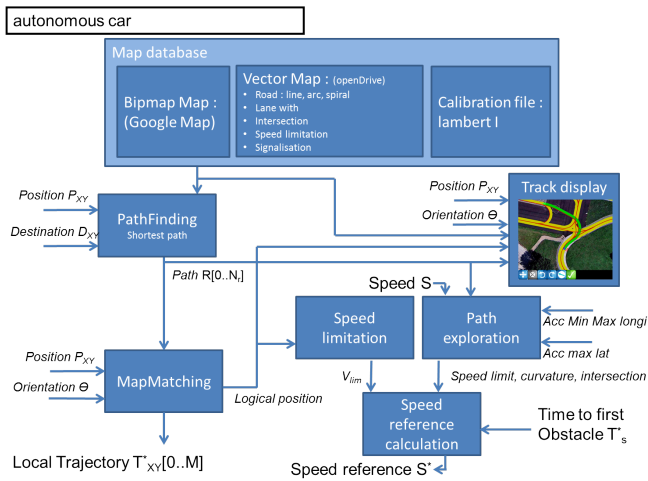


Fig. 9. Path planning architecture

gives an interesting results. The car was able to reach any given initial and ending point.

## VI. CONCLUSION AND DISCUSSION

The most important result in this work shows a methodology to easily move from autonomous robotics into autonomous car driving. We were able to evaluate different types of algorithms using three steps: simulation, hardware in the loop and large scale real world. To ease the evaluation of developed algorithms we defined a versatile architecture. This architecture is composed of different layers: database, path planning, control and perception. Different experiments were evaluated one to test different perception algorithm and the other to evaluate the path planning and control algorithms. Moreover, with our architecture it was really easy to move from one method into another.

## ACKNOWLEDGMENT

The authors would like to acknowledge Samer Khodor Al Kork for his help. Thanks for all the team for their daily support.

## REFERENCES

- [1] C. Mark, E. Magnus, H. J. P. and M. R. M, "Autonomous driving in urban environments: approaches, lessons and challenges." *Philos Transact A Math Phys Eng Sci*, vol. 368, no. 1928, pp. 1471–2962, 2010. [Online]. Available: <http://www.biomedsearch.com/nih/Autonomous-driving-in-urban-environments/20819826.html>
- [2] C. Urmson, C. Baker, J. Dolan, P. Rybski, B. Salesky, W. R. L. Whittaker, D. Ferguson, and M. Darms, "Autonomous driving in traffic: Boss and the urban challenge," *AI Magazine*, vol. 30, no. 1, pp. 17–29, June 2009.
- [3] C. Thorpe, R. Aufrere, J. Carlson, D. Duggins, T. Fong, J. Gowdy, J. Kozar, R. MacLachlan, C. McCabe, C. Mertz, A. Suppe, C.-C. Wang, and T. Yata, "Safe robot driving," in *Proceedings of the International Conference on Machine Automation (ICMA 2002)*, September 2002.
- [4] J. Leonard, J. How, S. Teller, M. Berger, S. Campbell, G. Fiore, L. Fletcher, E. Frazzoli, A. S. Huang, S. Karaman, O. Koch, Y. Kuwata, D. Moore, E. Olson, S. Peters, J. Teo, R. Truax, M. Walter, D. Barrett, A. Epstein, K. Maheloni, K. Moyer, T. Jones, R. Buckley, M. Antone, R. Galejs, S. Krishnamurthy, and J. Williams, "A perception driven autonomous urban robot," *International Journal of Field Robotics*, vol. 25, no. 10, pp. 727–774, 2008.

- [5] B. Schick, B. Kremer, J. Henning, and M. z. Heiden, "Simulation methods to evaluate and verify functions, quality and safety of advanced driver assistance systems," in *IPG Technology Conference*, 2008.
- [6] F. Saidi and G. M., "Transport delay characterization of scanner driving simulator," in *Driving Simulation Conference*, 2010.
- [7] M. S. Glaser, "Modélisation et contrôle d'un véhicule en trajectoire limite : application au développement d'un système d'aide à la conduite," Ph.D. dissertation, Ecole Doctorale Sitevry (Université Evry-Val-D'Esonne), 12 mars 2004.
- [8] A. Belbachir, J. christophe Smal, and J. marc Blosseville, "A robotic platform to evaluate autonomous driving systems," *15th International IEEE Annual Conference on Intelligent Transportation Systems*, September 16-19, 2012, Anchorage, Alaska, USA.