# ROS-based Architecture for Autonomous Intelligent Campus Automobile (iCab)

**David Martín Gómez**
Doctor en Informática, Investigador Posdoctoral

**Pablo Marín Plaza**
Máster Universitario en Robótica y Automatización, Doctorando

**Ahmed Hussein**
Máster en Ingeniería Mecatrónica, Doctorando

**Arturo de la Escalera Hueso**
Doctor Ingeniero Industrial, Profesor Titular

**José María Armingol Moreno**
Doctor Ingeniero Industrial, Catedrático de Universidad

Laboratorio de Sistemas Inteligentes

Universidad Carlos III de Madrid

**Resumen**

Este artículo presenta una plataforma de investigación para fomentar los Sistemas Inteligentes de Transporte en entornos urbanos. El sistema se denomina iCab y es el vehículo inteligente y autónomo de la Universidad Carlos III de Madrid. El objetivo del presente artículo es la descripción de la línea de investigación base para lograr un vehículo autónomo completamente funcional y seguro que permita la movilidad de las personas en un entorno urbano. La plataforma autónoma que se está desarrollando en la Universidad Carlos III de Madrid, se basa en un carro de golf modelo EZ-GO, que se ha automatizado para operar en modo autónomo. La arquitectura de percepción del entorno y control de los actuadores se basa en el sistema operativo (ROS), que permite importantes funcionalidades para la navegación autónoma, como son: la fusión de datos de múltiples sensores para la percepción óptima del entorno o la marca de tiempo de los diferentes dispositivos en tiempo real, entre otras. El estudio experimental que se presenta en este artículo muestra las ventajas de una arquitectura basada en ROS, favoreciendo la utilización de los vehículos autónomos, por su portabilidad y viabilidad para crear redes de vehículos autónomos, es decir, la interacción y cooperación entre vehículos autónomos que faciliten la movilidad urbana.

**Abstract**

This paper presents a smart research platform to foster intelligent transportation systems in urban environments, called iCab (Intelligent Campus Automobile) autonomous vehicle. The aim of the paper is to describe the initial steps to achieve a functional autonomous vehicle. The platform is a golf cart vehicle, E-Z-GO model, which is modified to operate in autonomous mode. The software core is based on Robot Operating System (ROS) architecture, which allows the fusion of multiple sensors data and time stamp of different devices in one embedded computer on the board of the platform. The proposed system shows the advantages of ROS-based architecture data management, such as but they are not limited to, huge data handling from the surrounding environment, computer vision system perception and laser scanner data interpretation. The sensors data are integrated with the ROS-based architecture to develop cutting-edge applications, which cope with the autonomous navigation requirements and real-time data processing. The experimental study shows that the ROS-based architecture outperforms former works in autonomous vehicles, for its portability and feasibility to create a network of autonomous vehicles, that is, the autonomous interaction of more than one vehicle in closeness environments fostering the urban mobility.

## 1. Introduction

The data of the World Health Organization (WHO) show that 1.3 million of people around the world died in 2013 due to road traffic accidents [1]. The majority of these accidents were because of human error, which could be avoided and minimized through using autonomous vehicles instead.

The first two main demonstrations of the capabilities of autonomous vehicles took place in the United States through the Defense Advanced Research Projects Agency (DARPA) to develop autonomous cars. First one was in 2004 and 2005, the DARPA Grand Challenge was two races in the desert with no dynamic obstacles [2]. The second one was in 2007, the DARPA Urban Challenge was a race in an urban circuit among autonomous cars, simulating the dynamic traffic as in real urban environments [3]. Furthermore, the first functional autonomous vehicle is the Google Self-Driving Car, it is designed under the traffic laws after years of research and it resulted to a fully autonomous car [4]. Autonomous vehicles continue as an important topic in intelligent transportation systems, where a recent work shows that driver-less vehicles could become widely available in the next 5 to 10 years [5].

Nowadays the in-vehicle applications outperform the environment perception through computer vision and laser scanner. They overcome the most significant technical limitations, such as robustness to face the changes in the environmental conditions due to illumination variation, such as shadows, low lighting conditions, night vision among others. Accordingly, perception applications ensure the suitable robustness and safety in case of large variety of lighting conditions and complex perception tasks [6]. Additionally, the use of computer vision is well-established in recent researches about autonomous vehicles; for example the route from Mannheim to Pforzheim by Mercedes Benz S-Class car. The car navigated 103 km on the route autonomously, it was equipped with computer vision systems and radar sensors along with digital maps [7].

Moreover, further problems of the autonomous vehicles are autonomous navigation and path planning. Many researchers implemented several approaches towards solving the problem in indoors environments. The results showed the feasibility to generate an obstacle free path from one point to another and navigate through the generated path with the localization of the vehicle at each point on route [8, 9, 10]. On the other-hand, for the outdoors environments, researchers implemented several algorithms aiming to obtain autonomous outdoor vehicle. Robot Operating System (ROS) based architecture was used to generate the mapping and localization for the autonomous navigation and the results outperformed other algorithms [11, 12]. ROS-based systems provide an operating system-like services to operate robots with the fusion of multiple sensors data and time stamp of different devices [13].

This paper presents the first steps in the implementation of a smart research platform to foster intelligent transportation systems in urban environments and describes the initial steps to achieve a functional autonomous vehicle. The project main objective is to implement and improve autonomous navigation and path planning approaches, based on image processing and laser scanner data interpretation. The implementation is performed over a smart ROS-based architecture, for real-time processing and communication of the software processes in an embedded computer. The computer is placed on the platform, golf cart vehicle, called iCab autonomous vehicle. This structure achieves the ease of data handling of the on-board sensors, in terms of camera and laser scanner, with the proposed ROS-based architecture to research on navigation applications. Also, the advantage of synchronizing low-level data by means of ROS-based systems is the use of reliable time stamp for the data acquisition from on-board iCab devices. Hence, the ROS-based systems allow the coordination of the drivers and middleware, which aim to simplify the complex task of global data acquisition and sensor synchronization.

Hence, the iCab applications can foster sensor fusion processes, which allows the improvement of the performance of each application in high-level stages. For this purpose, the proposed ROS-based architecture communicates the processes with each other in order to refine information and knowledge. It also provides higher level information to improve the decision making process, in other words, to avoid safely the collision with an obstacle or pedestrian in autonomous navigation. This proposed system enables the inter process communication in an independent and modular way, it also enables the on-board computer to run multiple and parallel algorithms in order to achieve both low-level objectives, such as sensor data acquisition and data preprocessing, and high-level objectives, such as pedestrian detection, obstacle avoidance, autonomous path planning and navigation. Last but not least, this architecture facilitates the scalability and adaptability for the changes of the on-board technology of the iCab, to accommodate novel sensors or higher requirements of the applications.

The remainder of this paper is organized into five sections. Section 2 introduces the experimental platform, emphasizing on the use of low-level on-board devices in the iCab, followed by Section 3, which presents the proposed ROS-based autonomous vehicle architecture. Section 4 explains the experimental results for different scenarios that will be used in autonomous navigation through urban environments. Finally, the conclusions and future work are summarized in Section 5.

## 2. Platform Description

The selected research platform is an electric golf cart vehicle, E-Z-GO model. It is modified to fulfil the project objectives, in terms of autonomous navigation and path planning. Moreover, in order to achieve multiple autonomous vehicles system, there are two identical golf carts, the first one see Figure 1.

*Fig. 1. Research platform: iCab 1*

Vehicle modifications are for the mechanical and electrical systems. Accordingly, the steering wheel is removed in order to install the motor-encoder system and to control the vehicle direction electronically, see Figure 2. Additionally, the throttle paddle is deactivated to control the traction electric motor of forward and backward motion through a power amplifier and governed by a PIC microcontroller. The rotor and stator parts of the motor are independent from each other, to facilitate the control of the power and torque in different environments, such as hard slope roads and rough terrains. The inputs of the microcontroller are the percentage of the maximum capacity for the stator, rotor and desired angle of the steering wheels.

*Fig. 2. iCab steering system: motor-encoder*

For environment perception, the vehicle is equipped with a laser rangefinder (SICK LMS 291). The device has over 180 degrees scanning range with 0.25 degrees angular resolution [14]. It is mounted on the front vehicle bumper at 30cm height above the ground. In order to avoid the detection of the steering wheels, the scanning range is limited to 100 degrees at 20Hz.

Additionally, the vehicle is equipped with a stereo vision binocular camera (Bumblebee 2). The camera has a maximum of 1032x776 pixels resolution at 20 frames per seconds [15]. It is mounted on the front windshield of the vehicle at 160 cm height above the ground and orientation of -45 degrees. The camera has three purposes, first to build a free road map in order to navigate in the environment, followed by visual odometry and finally pedestrian or obstacle detection.

These devices are connected to an on-board embedded computer. The computer has Intel Core i7 processor and is working under Ubuntu operating system. The display unit is a 7-inch TFT LCD touchscreen; it is installed on the vehicle front dashboard in order to view the system's interface software, display the current and desired locations in the map.

## 3. Proposed Autonomous Vehicle Architecture

### *3.1. Proposed Architecture*

In this work, the objective is to implement a complete architecture with various levels of complexity categorized in three layers; deliberative, sequencing and reactive skills [16]. Figure 3 shows the architecture structure, the advantages of this architecture are the ability to add more skills and modify the algorithms to obtain more efficient results during the development stage. For the architecture layers, the low-level has the simple reaction skills in the reactive layer, which controls the actuators and read the sensors data from the environment. It is followed by the sequencer in the hybrid layer, which incorporates a high level behaviour through logic sequence to the low-level layer to achieve the required behaviour. The highest level consists of the path planner in the deliberative layer, which generates the commands for the iCab to follow.

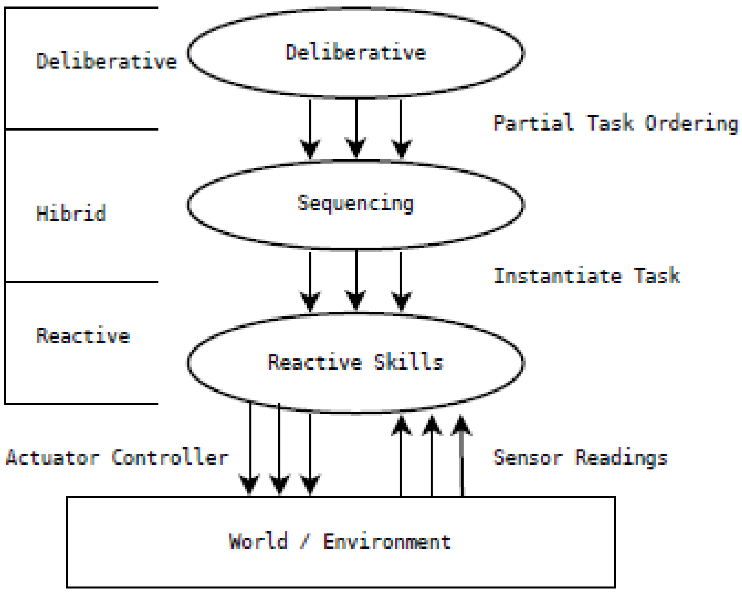


*Fig.3. Three-tier architecture*

- Reactive Skills: the initial control structure of the autonomous vehicle is implemented in this layer, in order to move the vehicle in the environment with basic commands such as “Move Forward”, “Move Backward”, “Turn Left”, “Turn Right” or “Stop”. The layer inputs are the outputs of the sequencing layer, which are conveyed one by one to generate the movement commands outputs via ROS-Services and send them to the controller.

- Sequencing: the layer inputs are the outputs of the deliberative layer, each input is considered as a specific task. The outputs are conveyed to the reaction skills layer with the desired actions for the vehicle movement. The complexity of this layer resides in the accuracy of generating simple skills after splitting the mid-level tasks. The behaviour is formed based on the accuracy level of these skills, in other words, low accuracy results in no movement of the vehicle, to avoid false actions.

- Deliberative: the logic in this layer manages the desired actions for the vehicle, in terms of localization, path planning, navigation and mapping. The layer inputs are from the user to define the desired destination on the map, then the layer generates the output tasks for the sequencing layer to split them in simple skills.

### *3.2 ROS Packages Description*

According to the former description of the architecture and with the consideration of the previous related work, the proposed architecture is implemented in ROS-based system. The Figure 4 shows the packages involved in the first steps of the project as the data sensor acquisition and the actuators.

The low-level layer is developed in C++ in a ROS package called “movement_manager”. This node is a server that receives the iCab status every 20ms (50Hz); in terms of encoders reading, battery voltage, heartbeat, PID configuration elements and state errors. These readings are published by “/movement_manager/status_info” topic, see Figure 5. It contains a custom message, which enables other nodes to subscribe to it and operate with the information. Additionally, the server waits other nodes to send the clientCall to perform a specific task. As input is written an incoming topic called “cmd_movement” which is other way to govern the actions of the iCab. The simple reactive skills layer contains the information to activate the actuators in moving forward, moving backward, turning left, turning right and stopping the vehicle.

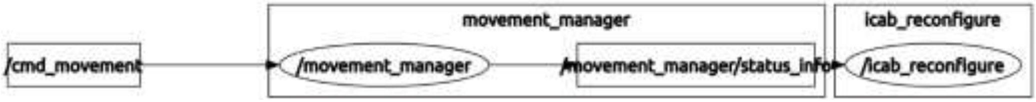

*Fig. 5. ROS low-level architecture: movement manager*

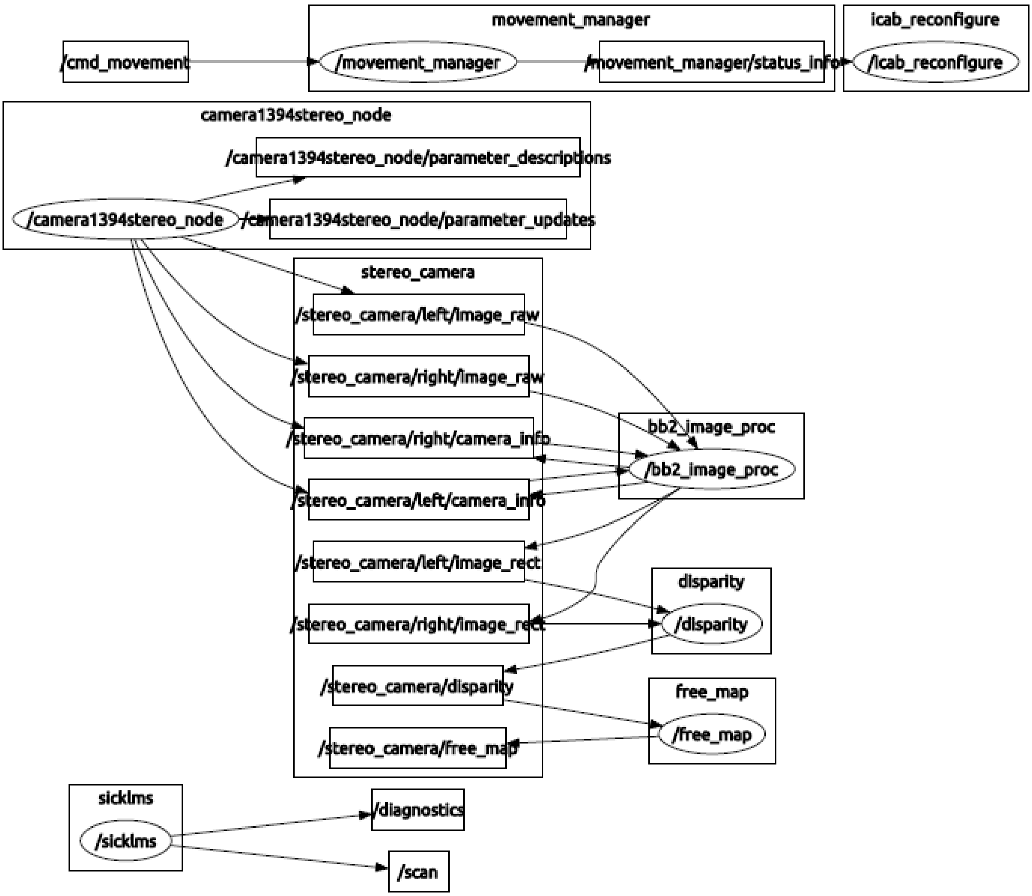


*Fig. 4. ROS low-level architecture*

For image data acquisition, there are three main packages, see Figure 6. First one is the "camera1394stereo_node" package, which receives interpolated data from both cameras (left and right), then it splits them into two different name-spaces; "stereo_camera/right" and "stereo_camera/left". The second package is the "bumblebee2", which receives the left and right images from the first package without any processing as "image_raw", then it rectifies both and publishes them in the same name-space as "image_rect". The last package is the take "disparity" package, which has the rectified images as inputs, then it generates the disparity map for the next step.

The system acquire information about the free space of the environment using an algorithm implemented by Musleh et al. in [17]. The node "free_map" receives the disparity map as an input, then it publishes the name-space known as road profile by "free_road" topic. This road profile is the result of

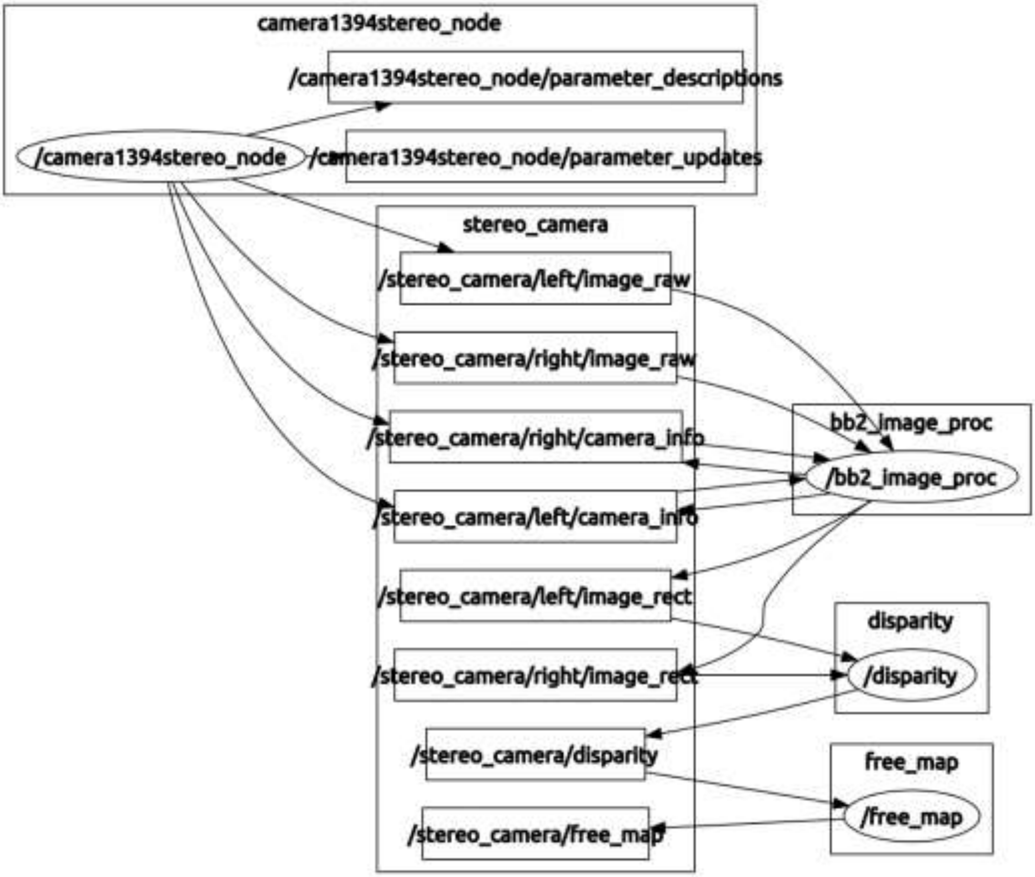


*Fig. 6. ROS low-level architecture: stereo processing*

the analysis of the u-v disparity for the environment and it split the image into free spaces and obstacles.

Last package is for the laser rangefinder, see Figure 7. The "sicktoolbox_wapper" package is used for the scanner, which receives the data of the laser rangefinder and publishes them as "LaserScan" messages through the "sicklms" node.

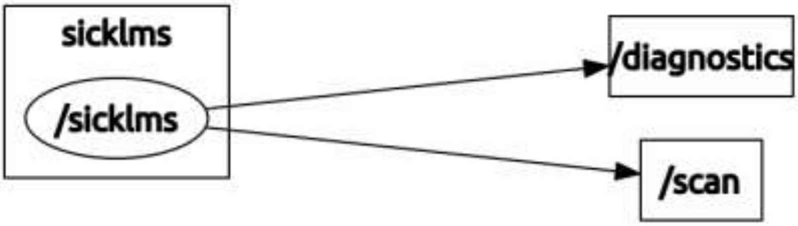


*Fig. 7. ROS low-level architecture: laser rangefinder*

The graphical user interface is for the communication and control of the iCab in this architecture, see Figure 8. This interface is implemented in ROS node called "icab_reconfigure", which sends the clientCall messages to the iCab server "movement_manager" node. The interface main layout is developed by Qt-Designer, which displays all the data acquired by the topic "/movement_manager/status_info" and allows the user to control the vehicle movements manually and stop in an emergency situation.

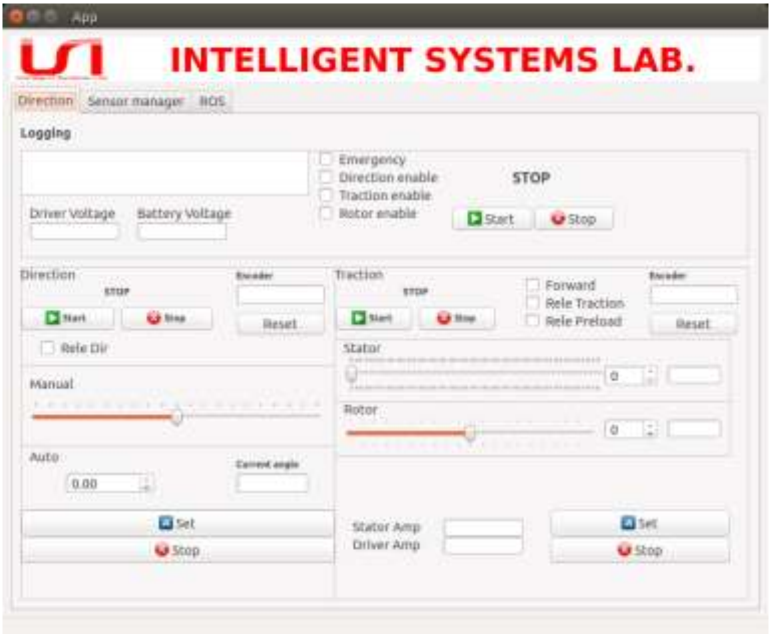


*Fig. 8. iCab graphical interface v0*

## 4. Exemplified ROS-based Architecture

The following section presents the exemplification of the proposed architecture, where the low-level structure, the data management of the perception devices and the time-stamp of the iCab platform are evaluated based on manoeuvres in urban environments. The urban scenarios have been evaluated in several experiments, however this section summarizes a representative scenario of each set of experiments.

The results have been obtained using the iCab platform where ROS-based architecture has been implemented in the on-board embedded computer. The algorithms can use the movement commands that control the forward and backward motion by adding or subtracting 5 % to the motor traction power to a maximum limit of 40 %. The steering wheel commands control the vehicle front wheels by adding or subtracting 5 degrees to the heading orientation angle, to a maximum limit of 25 degrees in either direction. These limits are selected as the safety measures during the initial steps.

Three-representative outdoor scenarios have been selected in order to show the performance of the ROS-based architecture managing the iCab movement. During all the scenarios, the iCab uses the laser rangefinder and the stereo camera in three-basic reactive tasks, with all perception devices publishing and processing data in real-time. The exemplification of the architecture in these three cases demonstrates that perception data, the low-level algorithms, and the movement commands are synchronized and time-stamp aim is achieved by using proposed ROS-based architecture. The performance of each instantiated scenario is illustrated by plotting the throttle values, heading angle and the distance to the object versus time.

The first basic reactive task is to follow a wall on one of the roads sides, see Figure 9(a). The iCab follows the left wall maintaining a parallel position, the left wall is selected for its uniformity. The starting point is at 5 meters from the wall, where the iCab motor traction commands are activated at 20 % of the rotor maximum power, which corresponds to 17.6 cm/s, to follow the wall for 85 seconds using laser scanner data in real-time. So, this first graph illustrates the performance of the ROS-based architecture exemplified in an iCab basic-task: the red curve is the distance from the vehicle to the left wall, whilst the blue curve is the steering wheel command to maintain the motion following the wall. The graph shows that the steering wheel command and laser scanner data are used both in real-time by ROS-based architecture to accomplish a basic low-level reactive task. The steering wheel commands are perfectly synchronized with laser data to maintain the distance to the wall.

The second basic reactive task is a straight forward movement with a stop reactive command when an obstacle (pedestrian in this case) appears in front of the iCab, see Figure 9(b). The exemplification in this case is the perception-action control loop based on laser scanner data and stop command in real-time, where perception-action synchronization is embedded in the ROS-based architecture. The iCab is moving and laser scanner detects an obstacle (pedestrian) trying to cross the street in front of it. This exemplification of the reactive command activation is crucial as low-level basic task in real-time for autonomous driving within university campus vicinity with many pedestrians. The basic reactive task stop the iCab whenever a laser scanner measurement from all array data is minor or equal to a minimum distance of 3.5 m.

In this experiment, the pedestrian crossed in front of the vehicle twice to force two stop reactive commands. The graph illustrates this performance. The blue curve is the throttle percentage power; whilst the red curve is the distance to pedestrian, where one representative distance has been selected from all measurements from the laser array to be plotted, this distance value belongs

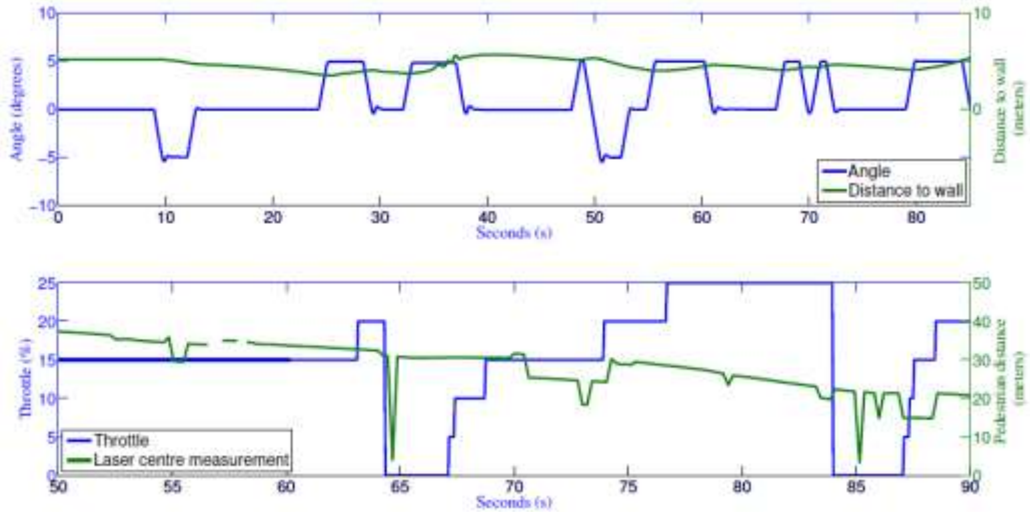


*Fig. 9. Basic reactive tasks using laser scanner data, motor traction commands and the steering wheel command in real time: (a) follow a wall, (b) stop with obstacle (pedestrian)*

to the laser centre measurement (that is, 0 degrees of the iCab header angle). The graph shows that laser scanner data in front of the vehicle, and how the stop command is activated in real-time before the pedestrian is in front of the vehicle and the throttle power is set to zero. This reactive behaviour by ROS-based architecture has been achieved successfully and demonstrate the perception-action loop in real-time of the iCab vehicle in urban environments.

Following, we use data from stereo camera in order to test again the performance of the proposed architecture in the low-level perception loop. The implemented algorithm obtains the disparity map and free map of the road in real-time by using both stereo images. The free map is generated by applying a specific threshold to the v-disparity of the disparity map. Figure 10 displays in top-right the disparity map, and in the top-left the free space (binary image), where the right side of the road appears as free space because of the unfeasibility to distinguish it from the actual road in v-disparity. The both stereo images are shown also in bottom-left and bottom right area. The perception algorithm is processed and published in real-time, that is, stereo images and processing data are available to other future processes inside proposed ROS-based architecture.

In order to compare the free_map in two different cases, in the next exemplification, the iCab is approaching to an outdoor exit door at the campus university and iCab can only navigate by free space, see Figure 11. The processed data of the stereo camera can be observed where the difference between the road and the wall is perfectly classified and integrates into

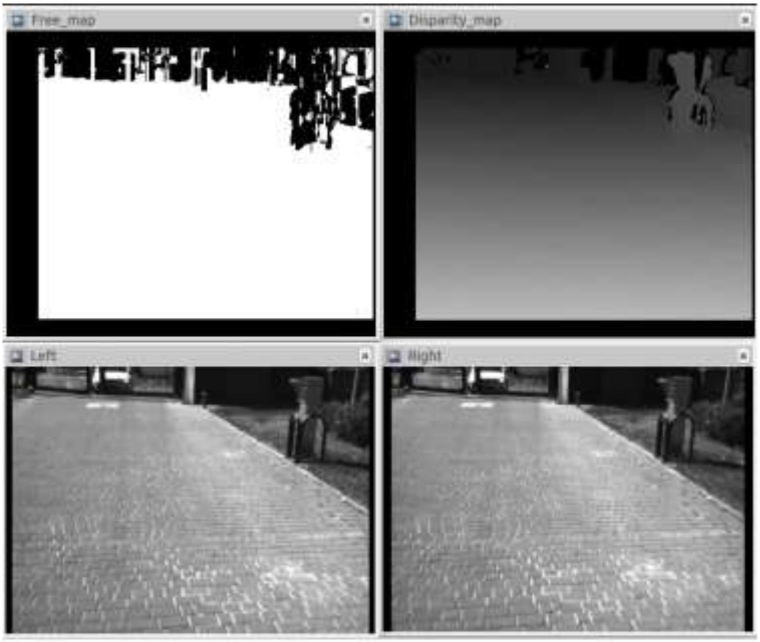


*Fig. 10. Stereo images, disparity map and free map ROS-based low-level perception processes in real-time without obstacles to accomplish autonomous navigation*

ROS-based architecture. That is, white area corresponds to the free space to iCab displacement; whilst black areas are objects higher than the road plane.

## 5. Conclusion and Future Works

This paper presents the design, development and exemplification of a ROS-based architecture for the iCab autonomous vehicle in urban environments. The aim of the architecture is to provide the iCab platform the capabilities to be used as a functional intelligent transportation vehicle. The perception-action low-level processes are accomplished by proposed architecture using laser rangefinder, the stereo camera and motor commands, where the exemplification of real-time data acquisition, time stamp and perception processing has been demonstrated. That is, the exemplification of the architecture shows the high performance of the system to obtain the necessary data from different scenarios to accomplish basic reactive tasks.

The future aspects of research include the integration of ROS-based high-level reasoning to accomplish path-planning, navigation and trajectory planning tasks for autonomous movement. At which the vehicle navigates a

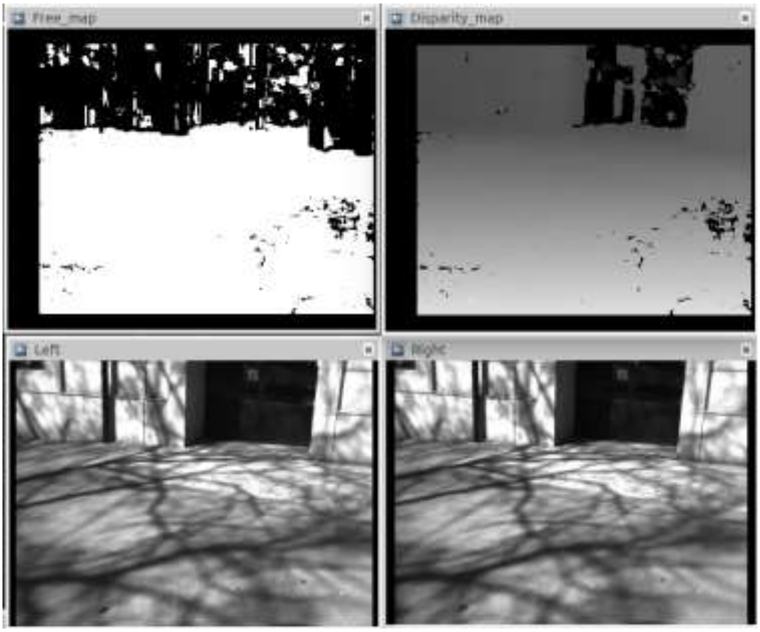


*Fig. 11. ROS-based low-level perception processes in real-time with obstacles information to perform an approximation manoeuvre for taking people into iCab*

given environment avoiding static obstacles and manoeuvring dynamic ones. Moreover the iCab platform can be extended to deal with more than one vehicle and create Multiple Vehicle Communication System (MVCS), at which the coordination and cooperation between the vehicles is necessary to achieve a network of autonomous transportation systems in urban environments.

**Acknowledgments**

This work was supported by the Spanish Government through the CICYT projects (TRA2013-48314-C3-1-R) and Comunidad de Madrid through SEGVAUTOTRIES (S2013/MIT-2713). The authors wish to express their gratitude to Francisco Javier Sánchez from the Electrical Engineering Department at Universidad Carlos III de Madrid for all his knowledge in electronics.

**References**

[1] W. H. Organization et al., WHO global status report on road safety 2013: supporting a decade of action. World Health Organization, 2013.

[2] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, et al., “Stanley: The robot that won the darpa grand challenge,” Journal of field Robotics, vol. 23, no. 9, pp. 661–692, 2006.

[3] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer, et al., “Autonomous driving in urban environments: Boss and the urban challenge,” Journal of Field Robotics, vol. 25, no. 8, pp. 425–466, 2008.

[4] M. Harris. (2015) Google’s self-driving car pals revealed. IEEE Spectrum. [Online]. Available: http://spectrum.ieee.org/cars-that-think/transportation/self-driving/googles-selfdriving-car-pals-revealed

[5] M. Mitchell, “Autonomous vehicles: No drivers required,” Nature, vol. 518, pp. 20–23, 2015.

[6] D. Martín, F. García, B. Musleh, D. Olmeda, G. Peláez, P. Marín, A. Ponz, C. Rodríguez, A. Al-Kaff, A. De La Escalera, et al., “Ivvi 2.0: An intelligent vehicle based on computational perception,” Expert Systems with Applications, vol. 41, no. 17, pp. 7927–7944, 2014.

[7] J. Ziegler, P. Bender, M. Schreiber, H. Lategahn, T. Strauss, C. Stiller, T. Dang, U. Franke, N. Appenrodt, C. Keller, et al., “Making bertha drive? an autonomous journey on a historic route,” Intelligent Transportation Systems Magazine, IEEE, vol. 6, no. 2, pp. 8–20, 2014.

[8] S. Shen, N. Michael, and V. Kumar, “Autonomous multi-floor indoor navigation with a computationally constrained mav,” International Conference on Robotics and Automation (ICRA’2011). IEEE., pp. 20–25, 2011.

[9] I. Wieser, A. V. Ruiz, M. Frassl, M. Angermann, J. Mueller, and M. Lichtenstern, “Autonomous robotic slam-based indoor navigation for high resolution sampling with complete coverage,” Position, Location and Navigation Symposium-PLANS’2014. IEEE., pp. 945–951, 2014.

[10] M. A. Hossain and I. Ferdous, “Autonomous robot path planning in dynamic environment using a new optimization technique inspired by bacterial foraging technique,” Robotics and Autonomous Systems, Elsevier., vol. 64, pp. 137–141, 2015.

[11] A. Llamazares, E. Molinos, M. Ocana, and F. Herranz, “Integrating absynthe autonomous navigation system into ROS,” International Conference on Robotics and Automation (ICRA’2014). IEEE., 2014.

[12] S. Zaman, W. Slany, and G. Steinbauer, "ROS-based mapping, localization and autonomous navigation using a pioneer 3-dx robot and their relevant issues," Saudi International Electronics, Communications and Photonics Conference (SIECPC'2011). IEEE., pp. 1–5, 2011.

[13] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source robot operating system," in ICRA workshop on open source software, vol. 3, no. 3.2, 2009, p. 5.

[14] S. S. Intelligence, "LMS200/211/221/291 laser measurement systems," Technical Description, 2006.

[15] P. Grey, "Bumblebee: stereo vision camera systems," Technical Description, 2012.

[16] R. P. Bonasso, R. J. Firby, E. Gat, D. Kortenkamp, D. P. Miller, and M. G. Slack, "Experiences with an architecture for intelligent, reactive agents," Journal of Experimental and Theoretical Artificial Intelligence, vol. 9, no. 2-3, pp. 237–256, 1997.

[17] B. Musleh, D. Martin, A. de la Escalera, and J. M. Armingol, "Visual ego motion estimation in urban environments based on uv disparity," in Intelligent Vehicles Symposium (IV), 2012 IEEE. IEEE, 2012, pp. 444–449.