Autonomous Service Robotics

Michelle Viscaíno, Javier Romero, and Fernando Auat

Abstract Autonomous robotics emerged as a research and development field nearly forty years ago, but only fifteen years ago, after the DARPA (Defense Advanced Research Projects Agency of the United States Department of Defense) challenge, autonomous mobile systems started to be considered as a solution to the transportation and service problem. This chapter is focused on autonomous (i.e., robotic) vehicles used as human transportation service from two points of view: on one hand, the autonomous vehicle that leads to intelligent transportation systems; on the other hand, autonomous vehicles used for rehabilitation or for enhancing mobility capabilities of their users. Both perspectives of autonomous systems are linked by the use of rapid prototyping techniques, aimed at converting a previously commercial product into a robotic system with a specific transportation usage. This chapter shows, in particular, two cases: two electric commercial vehicles (one golf cart and one car) converted into an autonomous robot for transporting people in cities or for executing specific tasks in sites; and an assistive vehicle (an electric scooter) used by people with reduced mobility. The design of the different components needed to achieve such automation is shown in detail herein.

1 Introduction

Transportation systems worldwide are facing challenges from a multidisciplinary perspective. First, the strong commitment of developed countries to reduce the carbon print, such as the case of countries that belong to the European Union, who

M. Viscaíno · J. Romero · F. Auat (✉)
Departamento de Electrónica, Universidad Técnica Federico Santa María, Avda. España 1680, 2390123 Valparaíso, Chile
e-mail: fernando.auat@usm.cl

M. Viscaíno
e-mail: michelle.viscaino@sansano.usm.cl

J. Romero
e-mail: javier.romero@sansano.usm.cl

F. Auat et al. (eds.), *Rapid Roboting*, Intelligent Systems, Control
and Automation: Science and Engineering 82,
https://doi.org/10.1007/978-3-319-40003-7_7

proposed to ban fossil fuelled vehicles by 2035 [1]. Policies such as the previous one implies that not only governments have to adequate their regulations, but also society and its consumption habits and the economy, since, for many countries, fossil oil and its derivatives are one of their main commodities [1]. Second, for the case of ground transportation, specifically, for the transportation of people, vehicles manufacturing has been growing in the last 70 years, where market preferred its own transportation directives instead of using public systems [21]. The latter led to a worldwide vehicle population of approximately 5.5 billion cars, driving in 2019 (without considering cars that were discarded, public transportation, machinery, and other fuel-based transportation systems). From the previous statement, it is clear that the reduction of the carbon print in transportation systems, or the ambitious goal of banning fossil oil as fuel by 2035, will cause an impact in several dimensions of our way of living and our economic systems.

The constant traffic jams (commonly present in great cities) has motivated the development of driving tools to aid drivers in finding the shortest paths to reach work or home, such as Waze, Google Maps, and other internet-based solutions (for further details, the reader is encouraged to take a look at [3, 13, 18, 27]). Such tools do not only help drivers in specific routes they are taking, but also the traffic data and geo-located information collected while using such applications, have allowed an enormous amount of historical driving data that can be used to statistically model the traffic in any city and to prevent jams where possible, even if such tools increased the length of the route. Furthermore, such data also allows to build consumption maps. When using fossil fuel, a consumption map can be used for estimating the pollution in a city [2, 20], impacting the real state value and the location of refuel (gas) stations.

Unlike combustion engines, electrically powered motors can be found more and more often in transportation vehicles. They started in small and green applications (wheelchairs, golf cars, and mobile robots) and now the market is moving towards electric cars [12]. In this regard, electric vehicles still have to face one of the most important challenges when compared to combustion engines: the autonomy of the batteries. Currently, electric vehicles cannot compete in autonomy with combustion engines: a fossil fuelled vehicle has an average autonomy of 400 km, with four people—of average weight—inside; whereas an electric commercial car reaches around 150 km with a single person inside [14]. Although the latter has been changing in the last years, it is unlikely to think that combustion engines can be replaced today by electric ones and that our ways of living would be the same [17]. But electric cars do have several other capabilities that fuelled cars do not have: the fact that being electric makes it capable of being fully automated and integrated into the grid: the car might become an IoT (Internet of Things) device [11]. The latter is one of the main advantages of electric cars, since an actuator, that transports people, can be part of the Internet revolution. One of the main drawbacks, as was previously stated, is the autonomy, and from there, several questions arise: how the cars will be re-charged? How the energy/power will be managed (not only the motion consumes electric energy, but also all the auxiliaries, such as air conditioned, lights, radio, among others)? Where to place charging stations, which technology will be used (there are currently no regulations on this matter, as there are for fuelled cars [9])?

Electric vehicles also have an important advantage when compared to vehicles with combustion engines. The automation of electric vehicles is easier [6, 16]. In a car with a combustion engine, each element of the driving system (the acceleration and brake pedals, the wheel, among others) need their own actuator, which is usually electric-based [16]. In an electric vehicle, the automation of the acceleration can be directly solved through the vehicle's own electronics, as shown in [6, 16]. The brake is usually mechanic (including hydraulic ones) and as the wheels, require of an electric actuator. The entire automated system with sensors can be integrated into a single processing unit or even connected into the Cloud for autonomous navigation tasks. Such is the case of the commercial vehicles available today in several countries, as the ones provided by Tesla Company [5], or the vehicles developed by Google, Uber, and others, just to mention a few. It is to be noted that not all the electric vehicles available today are automated: their automation correspond to the regulations of the country where it is commercialized and the availability of smart cities or intelligent transportation facilities, as reported in [8, 11].

Autonomous electric vehicles can be used for urban transportation (the well known autonomous cars [8] or automated trains [26]); for rural applications, there are some approaches such as the service units developed by [15, 25, 28], where they perform a previously given task in mining or agriculture. Additionally, we can find autonomous vehicles assisting people with some impairment, such as the case of autonomous (e.g., robotized) wheelchairs [4, 7, 10]. In this chapter, we will focus on three specific cases: one service unit used for field tasks, one electric autonomous car, and one automated scooter. The three service units have been used for research purposes. The approach followed in this work answers the following question: how rapid prototyping techniques can help in the automation of commercial electric vehicles without interfering with the product delivered by the manufacturer? Therefore, our aim is to add value without taking value from the commercial product. To this end, we focus this chapter on the different stages of the automation of the three mentioned vehicles, where rapid prototyping played a crucial role.

2 The Autonomous Service Unit

As an example of using rapid prototyping techniques when automating a service unit, we present herein the commercial electric car Cushman Hauler Pro, which was used for terrain modeling, autonomous navigation, power consumption performance evaluation, among other tasks [22–24]. The car was equipped with an onboard computer Nvidia Jetson TX2 and a set of sensors to measure different data associated with the vehicle. The sensors installed were an RTK (Real Time Kinematic) device NavCom SF-3040, a Stereolabs ZED camera, and a voltage and current sensor system for the batteries. Figure 1 summarizes the architecture of the system.

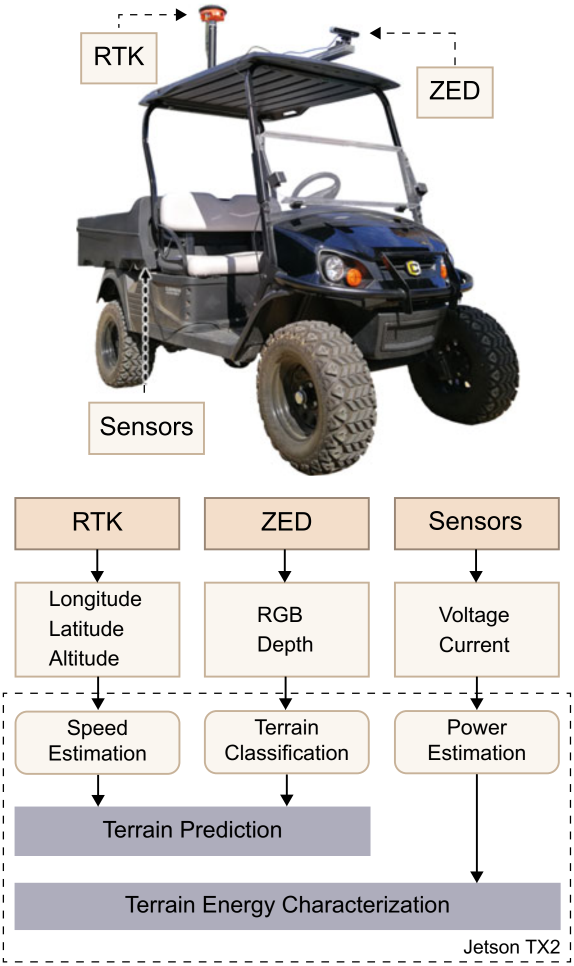


Fig. 1 System architecture. The vehicle is equipped with a ZED camera to obtain RGB-D data, an RTK for position and velocity estimation, and current and voltage sensors to estimate energy consumption. All the sensors connected to a Jetson TX2 are working on Ubuntu 16.04 under ROS operating system

Each sensor was used to measure data from the terrain and the vehicle itself which was then processed to estimate data on the vehicle-terrain interaction. All devices were implemented on ROS, or Robot Operating System, which allows to record all the information sampled with their respective time-stamps. Table 1 shows the main characteristics of each component of the architecture.

Table 1 Characteristics of the system and sensors

Characteristic	Description
Cushman hauler pro	
Motor	16.7 kW at peak torque
Electric system	72V DC
Max. speed	23.34 Km/h ± 0.80 Km/h
Curb weight	669 kg
ZED camera	
Range	From 0.5 to 20 m
Resolution	420 p, 720 p 1080 p and 2 k
Frame rate	Up to 100 fps
Communication	USB 3.0 port
Navcom SF-3040	
Accuracy RTK	Horizontal: 1 cm + 0.5 ppm
(<40 km)	Vertical: 2 cm + 1 ppm
Data rate	1 Hz, 5 Hz and 10 Hz
Communication	Serial port through USB
Voltage and current sensor	
Voltage range	From 15 to 80 V
Current range	From −300 to 300 A
Precision	12 bits ADC
Sampling rate	800 Hz average
Communication	Serial port through USB
Nvidia Jetson TX2	
GPU	NVIDIA Pascal^{TM}, 256 CUDA cores
CPU	HMP Dual Denver 2/2 MB L2 + Quad ARM\textregistered A57/2 MB L2
Memory	8 GB 128 bit LPDDR4 59.7 GB/s

2.1 Hardware

This section describes more in-depth the hardware involved in the automation of the service unit.

2.1.1 Jetson TX2

The Jetson TX2 is an embedded platform designed for artificial intelligence (AI) work in real-time. Two main parts comprise this developer kit, the carrier board and the processing module itself.

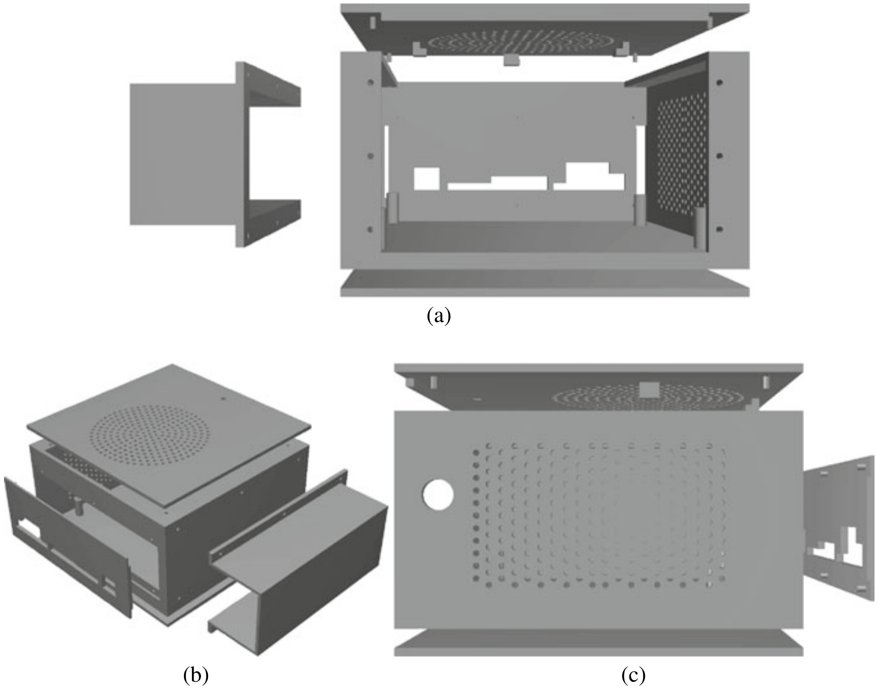


Fig. 2 3D model of the printed case. **a** Front view, **b** top back-side view and **c** side view of the case

The carrier board includes all the necessary connections and interfaces to allow the user to develop embedded software in this device. It can be mounted on any mini-ITX capable case and includes Ethernet, USB 2.0 and 3.0, HDMI, SD card, and Wi-Fi antenna ports on the backside. During automation of the service unit, we designed a 3D printed a custom case, Fig. 2 shows different views of the 3D model, while Fig. 3 shows a picture of the printed case.

The module contains all the processing power, which includes the mentioned characteristics in Table 1. The most important feature is the Nvidia Pascal GPU with CUDA support which allows this device to be a powerful tool for parallel processing and deep neural networks.

2.1.2 The RTK

The Real Time Kinematic or RTK used here is the NavCom StarFire SF-3040, which comprises a base and rover devices. The base was mounted on a static pole, while the rover was installed on the vehicle. Table 1 shows some of the most important features of this device pair. Figure 4 shows the installation of the RTK over on the vehicle.

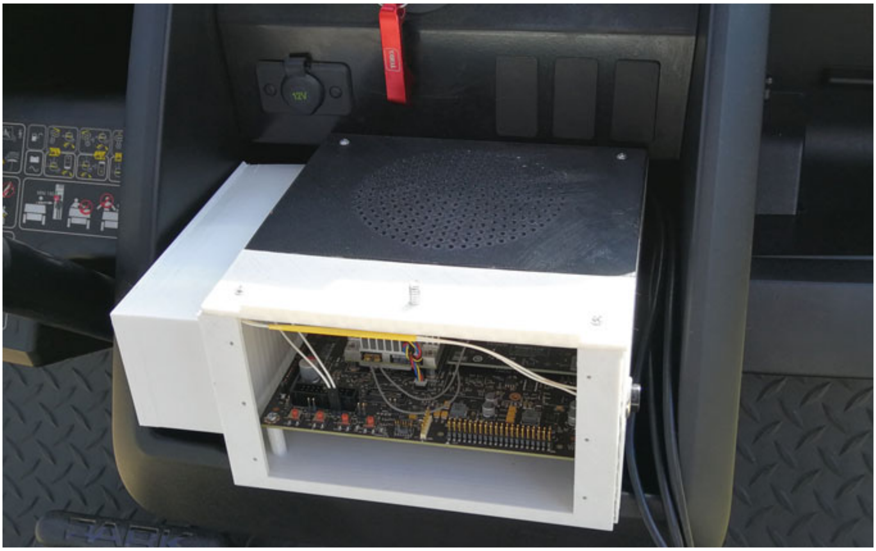


Fig. 3 Jetson TX2 mounted in the front panel of the electric vehicle on a custom 3D printed case. The case also includes a mounting hole for a tripod head to install a camera in case the vehicle used is more compact and space for a battery on the side

This device combination allows for sub-metric measurements of the position of the vehicle. We use the localization for three purposes, the first for vehicle localization, the second for velocity estimation of the vehicle, and lastly to geolocalize each image of the terrain taken and classified.

2.1.3 Camera

The camera used corresponds to the ZED camera by Stereolabs, a stereo camera designed for use outdoors. The ZED is a 2 k resolution stereo camera designed for depth sensing and motion tracking. Using binocular vision, the camera allows to measure objects in a range of 0.5–20 m as fast as 100 frames per second, depending on the resolution used, indoors, and outdoors. Figure 4 shows the installation of the camera on top of the vehicle.

Using the included Software Development Kit or SDK, it is possible to obtain and save depth information on each of the frames taken in real time. The SDK is compatible with both the Jetson series of embedded systems of Nvidia and with ROS. The SDK for this camera also includes more capabilities, which were not used in this work, for example, spatial mapping and visual odometry, these features may allow to improve and expand this work in the future.

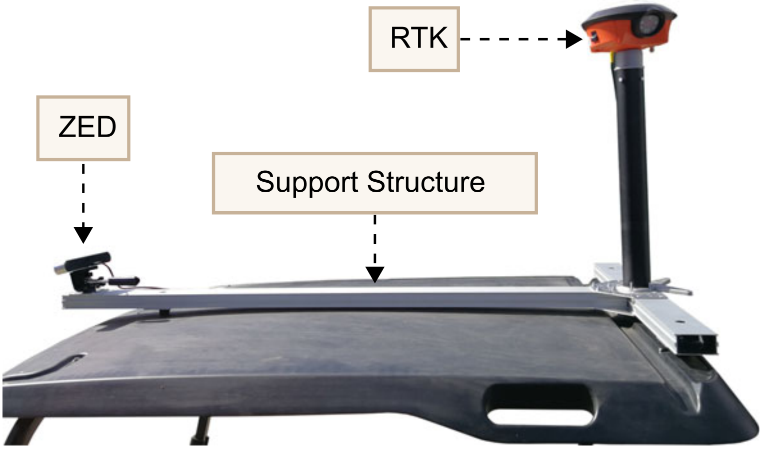


Fig. 4 Top view of the service unit's ceiling with the RTK and the ZED camera installed on it, using a supporting structure

2.2 Software

This section gives a description of the software environment used for the automation of the service unit. The focus of this part is on the acquisition and recording of the data in each experimental test.

The software architecture is divided in three parts: data logging, preprocessing, and processing. Figure 5 shows a diagram that represents each of these divisions. The following subsections will describe each of the parts, how they work, and their function.

2.2.1 Data Logging

Data logging of the sensors corresponds to the first part of the system. Schematic 1 in Fig. 5 shows this process. As shown in the figure, each of the sensors is connected to the main computer which runs ROS.

Each sensor has a Node, or ROS process, associated with it, that allows to save the data acquired with the sensor in the computer. The main advantage of using this method is that ROS allows to save each sample with a time-stamp, which also allows the user to synchronize or later re-run the experiments as required.

The nodes used are:

- nmea_navsat_driver: this node connects through the serial port with the RTK and publishes over a topic called \ `fix` which contains longitude, latitude and altitude measured by the RTK system and logged with the time-stamp for each sample.

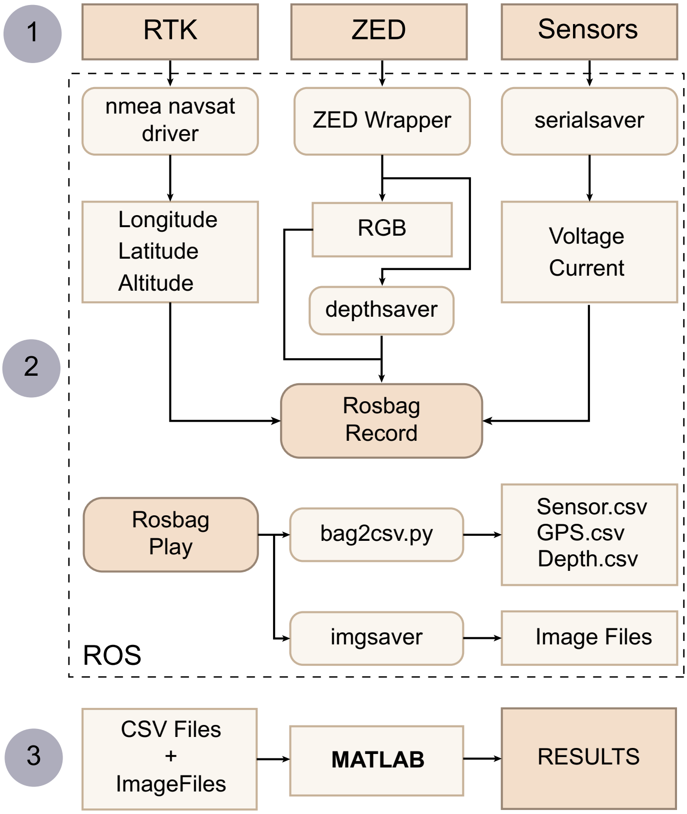


Fig. 5 Software architecture used in this work (1) corresponds to data logging, (2) to Preprocessing of data, and (3) Processing of the data for results. Rounded rectangles correspond to nodes or ROS packages, light gray rectangles to data. Pointed line square surrounding schematics (1) and (2) correspond to the main computer running ROS. Dark gray sharped rectangles with outline correspond to the different hardware used, MATLAB square is indicated with the same color because it was run in a different computer hardware than the rest of the system

- ZED_wrapper: this is a package that lets the user utilize all of the ZED SDK capabilities through ROS. It publishes all the necessary data in different topics (e.g., `zed\rgb\image_rect_color` publishes the rectified image captured by the left camera)
- serialsaver: this node captures the information sent by the voltage and current sensor through serial port, parses it and publishes in a topic called `\serial \sensorser` which contains voltage and current measurements.
- depthsaver: this is a secondary node that subscribes to the ZED depth topic and takes only a segment of the depth information given. After extracting the required segment, which corresponds to 600 samples within a rectangle of 1200 × 200

pixels centered on the image, a message is published in the topic `\depth\array` in the form of a vector containing all these depth samples.

- Rosbag: this is a ROS package that allows to save all the topics required by the user in a file with file-type `.bag`. This file contains all the information published in the topic and the corresponding time-stamps in a way that it is easy for the user to repeat the experiments performed or extract information for off-line processing. Record is the command that allows the user to save the information.

The result of this first stage of the process is a `.bag` file. Second stage, Preprocessing the data, is in charge of this data extraction.

2.2.2 Preprocessing

This stage is the shortest and its purpose is to prepare the files containing all the data to be read and processed in the last stage. Schematic 2 in Fig. 5 shows this process.

The nodes used are:

- Rosbag: this package is used again in this part, though the Play command is used. This command allows the user to reproduce a `.bag` file simulating the original experiment. It is also possible to configure the sampling rate and other parameters on this replaying.
- bag2csv.py: this is a small script that allows the user to extract each topic in separate `.csv` files. The topics that can be extracted in this manner must contain text or numbers, topics with other types of data (e.g., images) cannot be saved in this manner.
- imgsaver: this node allows the user to save each image frame from the `.bag` file into an image file with `.png` file-type. It also saves a `.csv` file with the time-stamps for each image.

After this second stage of the process, the results are four `.csv` files corresponding to voltage and current sensor data, localization data, and depth data. It also generates image files for each frame captured by the camera during the experiment.

2.2.3 Processing

The last stage corresponds to Processing. In this stage, the data obtained from the previous one is, as the name says, processed. We use the results from each experiment for further analysis and study, namely estimating models for each terrain measured and Energy consumption prediction with these models. Schematic 3 in Fig. 5 shows this process.

As can be seen in the figure mentioned, this stage comprises entering the data extracted in the previous stage to MATLAB and obtaining the final results. Inside MATLAB, we use several processes to clean, filter, synchronize, and classify all the data. Figure 6 shows a diagram that resumes the processes executed and their order.

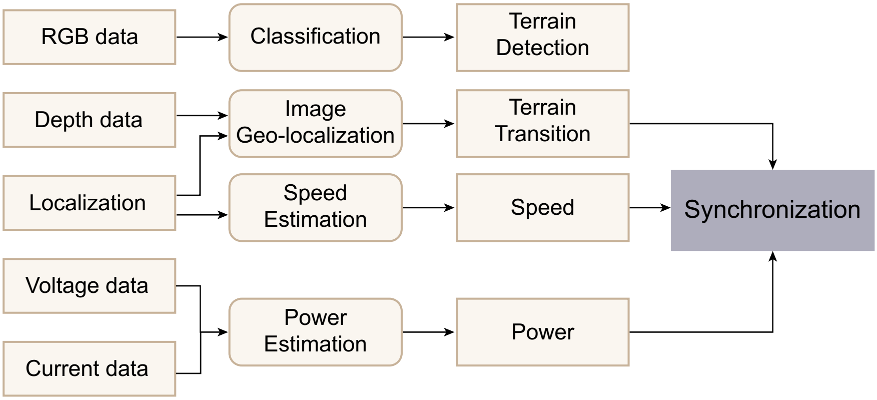


Fig. 6 Diagram of the processes performed inside MATLAB for each experiment

3 The Autonomous Car

An electric car is a vehicle propelled by one or more electric motors, which are powered by electric energy from batteries housed inside the vehicle.

This type of vehicle has the advantage that its electric motor provides instant torque, is more efficient than internal combustion engines, does not pollute, and is much easier to build and maintain than an internal combustion engine.

Although the first electric automobile appeared in the 1880s, due to recent developments in batteries, (which have made it possible to increase the amount of stored energy, lower costs and increase the life span of the car), is that electric cars are gaining importance again, slowly succeeding in displacing internal combustion vehicles.

The 100% electric vehicle Renault Twizy, is a small car designed for urban areas, which started in 2009 as a concept for the Frankfurt Motor Show, and it is marketed since 2012 mainly in European countries, where it has managed to sell more than 18,000 units accumulated until 2017. Figure 7 shows a picture of the electric car, whereas Table 2 summarizes the main mechanical and electrical features of the Twizzy.

3.1 The Brake

The Twizy vehicle has a hydraulic braking system, with a single-circuit configuration. In this type of configuration, there is only a single brake fluid circuit, where the pressure from the brake pedal is transmitted to the other pistons which actuate the brake presses. Some vehicles, in order to increase the the robustness of the system against possible fluid leaks in the brake circuit (which may render the brake system useless), contain independent brake circuits, which consist of two hydraulic circuits

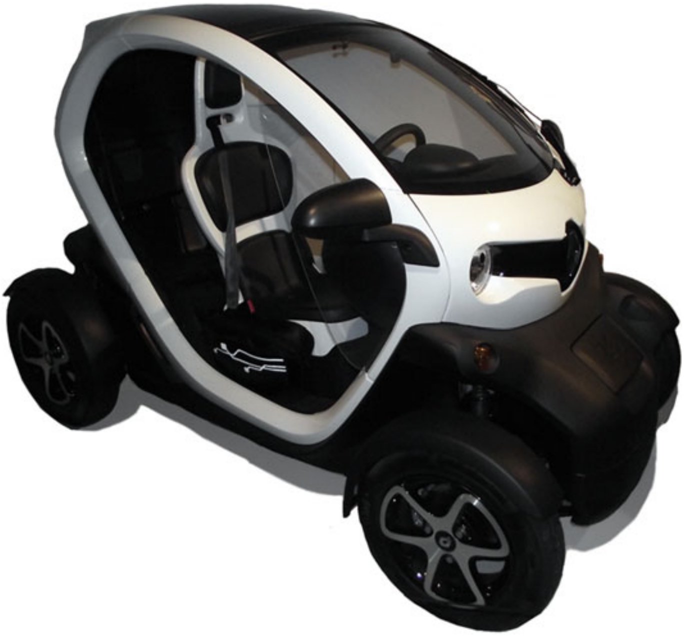


Fig. 7 Picture of the electric car Twizzy, by Renault

Table 2 Characteristics of the Twizzy

Characteristic	Description
Twizzy, by Renault	
Motor	Electric, synchronous
Power	20 HP
Torque	57 Nm
Transmission	Reduction, with one march ahead and one reverse
Max. speed	80 Km/h
Autonomy of the battery	100 Km
Battery	Ion-lithium 6.1 kWh
Break	Regenerative
Acceleration	Electronic, resistive type
Charging	220 V/10 A

actuating separately the front and rear brakes. If there is a leak in one of the circuits, the other continues to function.

One of the important features, present in most electric vehicles (including the Twizzy), is regenerative braking. Since an electric vehicle uses an electric motor to convert the energy into kinetic energy, the process of regenerative braking uses the opposite process. The electric motor absorbs the vehicle's kinetic energy and converts it into electrical energy. This energy can then be stored again in the vehicle's own batteries increasing the vehicle's autonomy. However, there are some limitations to the use of regenerative braking. One of these is given by the system's capacity to absorb the energy it generates. Since the batteries have a charging power limit, if this limit is exceeded, the batteries could be damaged. This is why part of the energy it regenerates can be lost. Thus, some vehicles include some type of resistor that burns the excess energy, or, as in the case of the Twizzy electric vehicle, regenerative braking is partially disabled, also changing the behavior of the car.

Figure 8 shows the original pedals from the Twizzy. When automating the vehicle one of the constraints was to implement a system such that the brake pedal could be either actuated or pressed by a driver, we designed the adaptation shown in Fig. 9. Such adaptation, printed using a 3D printer, was positioned on the top of the pedal and allowed for the two functions previously mentioned. To govern the brake, we installed a linear actuator, whose response time—for the actuator at full length—was less than 0.5 s, which was required by the Chilean traffic regulations.

3.2 The Traction Velocity

As mentioned above, the Renault Twizy electric vehicle has an electric accelerator, which can be operated directly on the vehicle, without having to couple an electromechanical system as in the case of the brake pedal seen above. Figure 10 shows a lateral, top, and connected view of the acceleration pedal into the vehicle's bus. The connection is a set of resistances arrangement which were previously identified. To be able to govern acceleration, we developed another electronic interface, using an Arduino Mega platform. Such interface interprets our velocity commands and converts them into voltage levels which are later input into the vehicle's bus. Figure 11 shows a scheme of the interface designed for controlling the acceleration pedal.

Since we are able to command acceleration via the computer, in order to close the loop and have wheel velocity readings, we installed an encoder at the rear wheel. Such encoder provides of dead-reckoning estimates of the vehicle and allows us to implement different control strategies to test the performance of the automated system, as we did in [19]. Figure 12 shows a picture of the encoder placed at the right rear wheel.

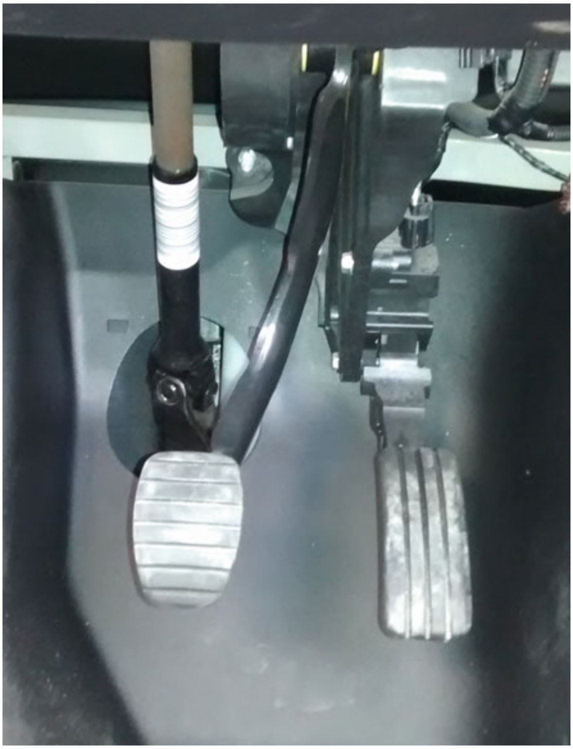


Fig. 8 Original brake and acceleration pedals from the Twizzy

3.2.1 The Heading

In this particular case, we designed a similar gear as the one shown later in this work, in Sect. 4.3.1, where we printed two gears and adapted one of them to a servomotor. The other one was installed on the vehicle's wheel. The interface to govern the heading was the same used to govern the brake, shown in Fig. 11.

4 The Autonomous Assistive Vehicle

For implementation of the proposed system and real-world experimentation, a modified robotic scooter is used, shown in Fig. 13. The scooter is equipped only with exteroceptive sensors, and does not possess proprioceptive sensors for measuring velocity, pose, or any internal condition.

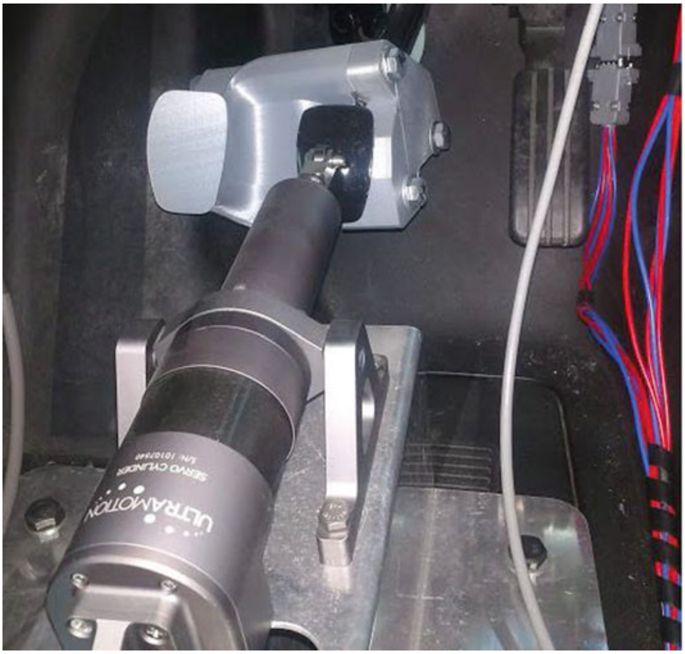


Fig. 9 Adaptation of the brake, using 3D printing

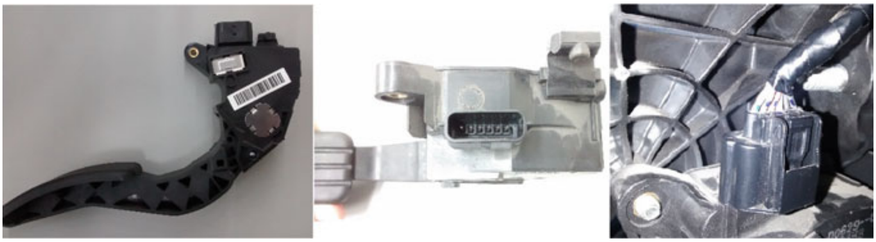


Fig. 10 Different snap-shots of the acceleration pedal and its electronic connection to the vehicle bus

4.1 Motors and Encoders

The scooter is equipped with a bipolar stepper motor to control linear speed when in motion. A calibration stage allows a rough mapping of the motor's position and the robot's stationary linear speed. The driver for this motor connects via USB to the processing unit (in this case, a portable computer). Additionally, an unipolar DC motor is mounted onto the scooter's front steering axis via two gears. This allows the system to automatically control the vehicle's steering angle from the computer. This motor counts with an encoder that, after calibration, allows a rough measurement of the steering angle in real time. The driver also connects to the computer via USB.

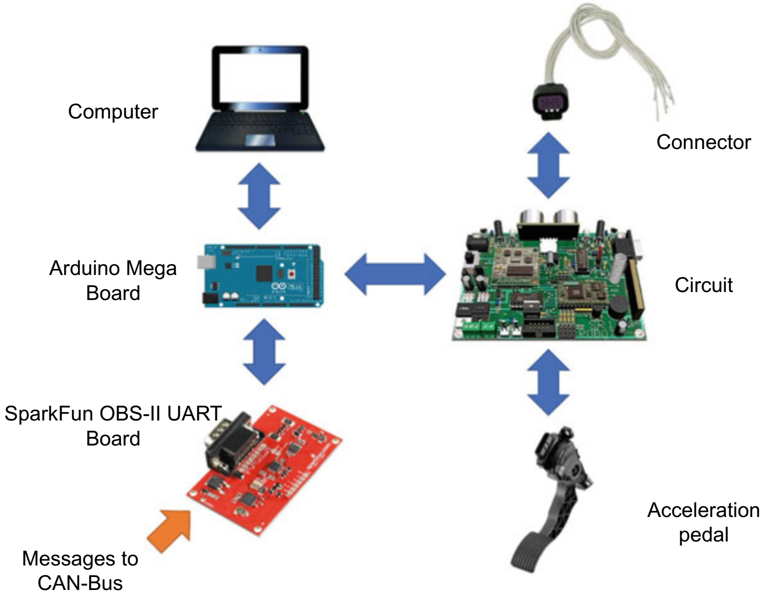


Fig. 11 Interface based on Arduino Mega to connect the computer into the vehicle's CAN bus, to thus control the acceleration via programming

4.2 Sensors

The only exteroceptive sensor mounted on the scooter is an Hokuyo laser range finder,This allows measurement in 240 deg, up to 4 m from the laser.

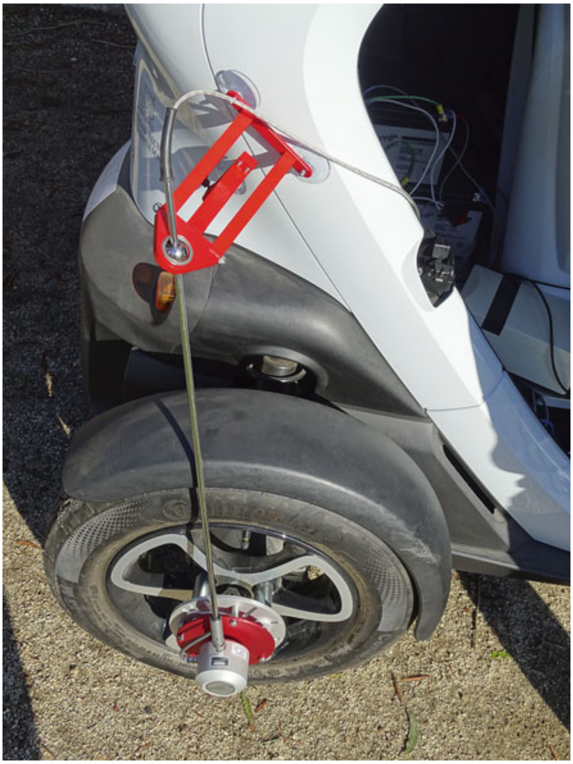
4.3 Communication System

In order to communicate the main program (in charge of controlling the vehicle, monitoring all activity, running mapping and path planning algorithms, among others) with the sensors and hardware onboard the scooter, shared memory is used. All motor drivers are written in C++, and via shared memory communicate with the main program. The same approach is used to read and buffer all laser scans, which can be read by the main program at any time. The basic architecture is shown in Fig. 14.

4.3.1 Mechatronics Design and Assembling

Using the capabilities of rapid prototyping, the mechatronization of the scooter was rapidly achieved in terms of functionality and costs. For example, Fig. 15 shows

Fig. 12 Installation of an encoder at the rear wheel for positioning purposes



several snapshots of the gears developed to control the vehicle's heading. One gear was designed to be placed at the scooter's clamp, whereas the other gear was located at the bottom engine, in charge of controlling the degrees of turning of the system. Although the accuracy of the movement is mainly given by the bottom engine, the teeth of the gears had a slight backslash to avoid jamming. As can be seen, both gears were especially designed for the needs of automation of the scooter.

A few things are worth mentioning about the mechatronized scooter:

- The system is controlled by an Arduino microcontroller connected to a laptop.
- One back wheel, as shown in Fig. 15 (bottom, left) has a potentiometer, acting as an encoder, for wheel velocity estimation.
- The front chassis has the electronics: a LiDAR, the microcontroller, the batteries, among others.
- The velocity of the vehicle is controlled via a servo-motor, through a PWM signal from the microcontroller.
- The gears were designed and printed in our 3D printers, whereas the rest of the chassis was made of aluminum.
- The entire system was adapted to the scooter without changing the manufacturer's specifications.

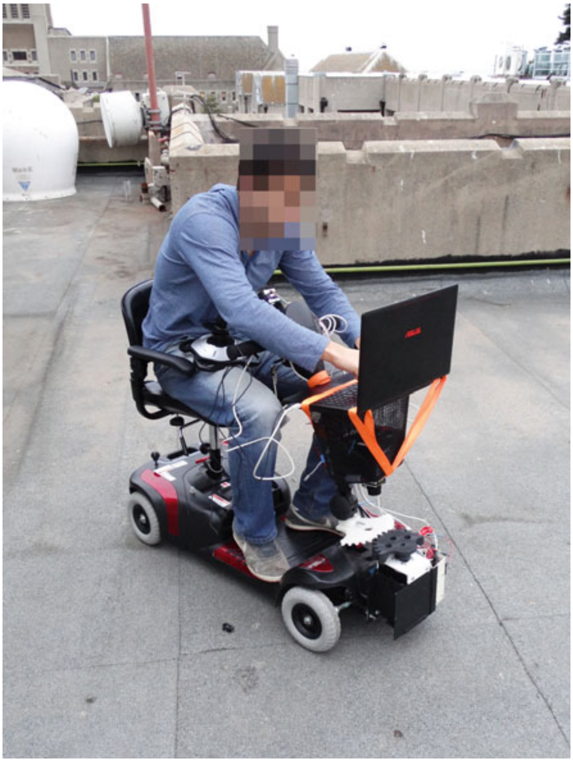


Fig. 13 Picture of the automatized scooter

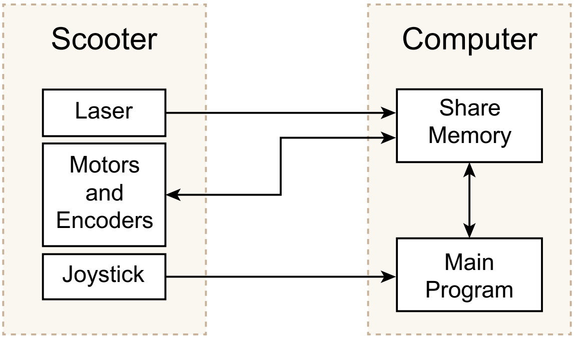


Fig. 14 Layout of the communication system

4.4 Human–System Interface

Though a robot's capacity to predict a user's intention has been addressed as an important challenge, the user's capacity to correctly interpret the robot's actions, understood as manifestations of intentional states, is equally important for satisfactory and sustainable human–robot interaction. A GUI (graphical user interface) was designed to provide feedback to the user, in terms of the machine's interpretation

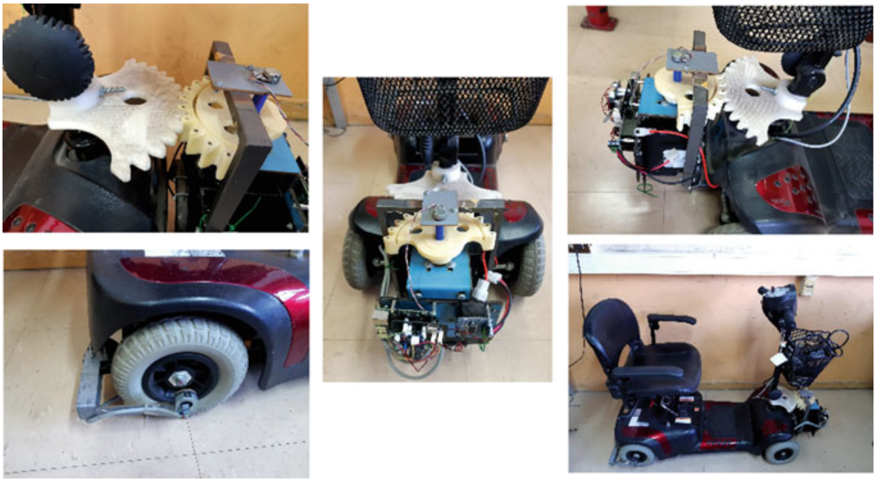


Fig. 15 Different snap-shots of the mechatronized scooter

of a given scenario. This feedback comes in the form of a map of the environment with visual representations of the vehicle's future state, the current path, possible destinations, currently preferred areas of navigation, among others. It also displays messages in the form of text, but this can be improved via automatic voice generation. In addition, the user may select automatic destinations from the interface for autonomous or semi-autonomous modes. Despite the importance of an adequate interface in terms of increasing a user's trust in the system, as well as reducing frustration due to poor human–robot communication, this was not the main focus of this research. Additionally, the user can control the vehicle through a joystick in manual mode and interact with the system in semi-autonomous modes.

4.4.1 General System Architecture

The general system architecture is shown in Fig. 16. The robot uses a laser range sensor to scan the environment. This data is processed by the localization and mapping module to estimate the robot's location and generate a map for path planning and displaying information to the user via a GUI. Users receive and use feedback from the environment and the GUI to generate navigation intentions using a joystick, which translates into control commands via the Controller Module. If manual mode is selected, the joystick-generated control command drives the assistive robot directly. If autonomous navigation is selected (a goal must be set) the Path Planner Module determines a safe trajectory for the vehicle, which the Controller Module translates into control commands for each sampling time. In this mode, the joystick plays no role. Finally, if collaborative control is selected, the joystick not only influences control directly, but serves as input for the Path Planner Module (assisted autonomous

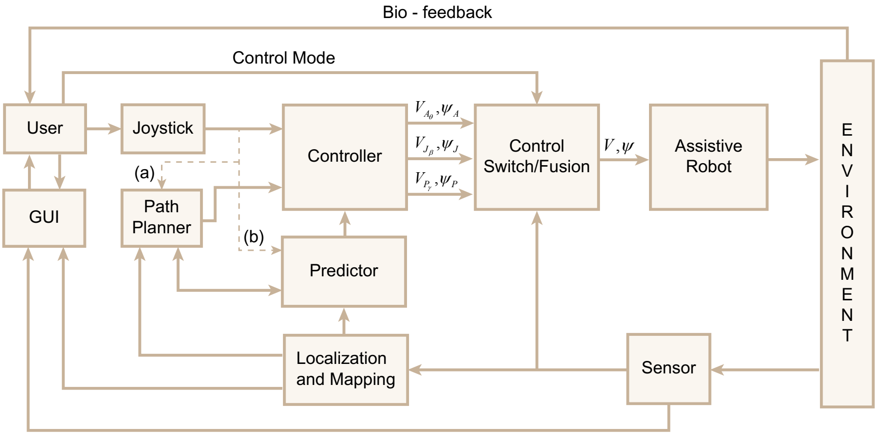


Fig. 16 General control scheme implemented on the assistive vehicle to allow autonomous navigation at different modes

mode) or to the Prediction Module (assisted manual mode). These modules interpret the joystick movements so as to aid the user during navigation. Up to three different control commands (linear speed and steering angle) are generated independently, corresponding to manual control (what the user wants), autonomous control (what the machine wants), and the predictor module (what the machine thinks the user wants). These three are switched and combined in accordance with the selected control mode and sent to the motor control unit to act on the vehicle.

4.4.2 Control Modes

Four control modes are available to the user, offering different levels of autonomy. This can be switched at any time using the GUI available onboard the vehicle.

1. Manual mode: The user has complete control of the vehicle via joystick. No collision detection systems are available, thus allowing the user to navigate and push objects if necessary.
2. Assisted manual mode: The user controls the vehicle via joystick, and the predictor module uses the user's control command, the estimated pose, and map information to estimate the future trajectory. If the path is deemed safe, the mode is indistinguishable from manual mode. If a future collision is detected, the map information is used to generate a safe path in the direction of the user's command.
3. Assisted autonomous mode: This mode requires the user to set a destination on the map offered by the GUI. The robot will navigate autonomously with the information provided by the path planner and localization and mapping modules. In particular, Dijkstra's Algorithm is used to calculate the optimal path to the goal. The user may use the joystick to take control of the vehicle at any time,

while simultaneously re-defining transition weights on the Dijkstra nodes, thus conditioning the autonomous behavior.

4. Autonomous mode: The user sets a destination on the map and the robot navigates autonomously. The user has no means of intervening in the navigation, other than stopping the vehicle if necessary. This mode is implemented primarily for testing and comparison purposes.

5 Concluding Remarks

In this chapter, we have shown three cases where rapid prototyping was used to develop autonomous robots used for research purposes, but the aim of providing a service, i.e., to assist a person or a process.

The autonomous service unit consisted of an electric golf car. The rapid prototyping techniques were especially focused on solving the sensor placement on the chassis of the vehicle and to develop a case to contain all the electronic devices. The main outcomes were that we were able to locate the GNSS antenna, the stereo camera, and other sensors in a quick yet efficient manner. The system was tested in the field and the main results were published in [22–24].

The autonomous car is an iconic case study: the electric vehicle Twizzy by Renault was purchased by the Advanced Center for Electrical and Electronic Engineering, Federico Santa Maria Technical University, Chile in 2017, and it became the first electric vehicle fully automated in the country. The rapid prototyping techniques allowed us to design the piece for the brake pedal: the assembled piece should allow one person to still press the brake since Chilean regulations do not allow yet autonomous cars on the streets. Therefore, a person should always be behind the wheel. The designed piece was installed onto the brake pedal without affecting the pedal's functioning. Additionally, the linear actuator used to automate the brake was selected according to the Chilean response time regulations: the brake should be pressed at full length in less than 0.5 s. Additionally, we used rapid prototyping techniques to install the motor that controls the heating system, without modifying the vehicle as it is delivered from the manufacturer. The scientific results achieved with the automated Twizzy can be seen here [19].

The assistive vehicle was also automated using rapid prototyping techniques. The system was specially designed to enhance the mobility capabilities of the users. The main challenge was the designing of the gear system to control the clamp. The electronics were placed at the front of the vehicle and the powering was external to the vehicle's own batteries. The main outcomes of this system were published in [10]. In particular, an extensive human–machine study was conducted to validate the usability of the four navigation strategies programmed in the vehicle.

It is worth noticing that all the vehicles or systems presented herein are electric-based. Unlike combustion engines, electric motors allow for a quick electronic or electric modification or assembling, there are almost no vibrations (less dynamic complexities) and the vehicle can be later programmed. From an electric basis of the vehicle, the electronics are designed to read its sensors and control its actuators. Later, a control strategy is implemented on a programmed board (such as a microcontroller or even a computer). Finally, a navigation strategy is developed. Such a strategy might be integrated into the cloud, making the vehicle one more device of IoT (Internet of Things), which is the current trend as was detailed in the introduction. When working with combustion engines, the challenge is even bigger (reaching an electronic basis for developing autonomous behavior requires further development and often changes in the chassis). But current carbon prints regulations are making such engines obsolete.

Acknowledgements This work was partially supported by Basal Project FB0008, CONICYT-PCHA/Doctorado Nacional/2018-21181420.

References

1. Carey N, Steitz C (2021) EU proposes effective ban for new fossil-fuel cars from 2035. Accessed 14 July 2021. https://www.reuters.com/business/retail-consumer/eu-proposes-effective-ban-new-fossil-fuel-car-sales-2035-2021-07-14/
2. Arango I, Lopez C, Ceren A (2021) Improving the autonomy of a mid-drive motor electric bicycle based on system efficiency maps and its performance. World Electric Veh J 12(2):59
3. Ashfaq R, Saleem M (2019) Use of global navigation satellite system (gnss) software defined receiver (sdr) for determining the 3d real time position variation in dense urban areas by averaging method. In: 2019 sixth international conference on aerospace science and engineering (ICASE), IEEE, pp 1–9
4. Bastos-Filho TF, Cheein FA, Müller SMT, Celeste WC, de la Cruz C, Cavalieri DC, Sarcinelli-Filho M, Amaral PFS, Perez E, Soria CM et al (2013) Towards a new modality-independent interface for a robotic wheelchair. IEEE Trans Neural Syst Rehabil Eng 22(3):567–584
5. Betz S (2021) The top 21 self-driving car companies paving the way for an autonomous future. Accessed 22nd July 2021. https://builtin.com/transportation-tech/self-driving-car-companies
6. Buechel M, Frtunikj J, Becker K, Sommer S, Buckl C, Armbruster M, Marek A, Zirkler A, Klein C, Knoll A (2015) An automated electric vehicle prototype showing new trends in automotive architectures. In: 2015 IEEE 18th international conference on intelligent transportation systems, IEEE, pp 1274–1279
7. De la Cruz C, Bastos TF, Cheein FAA, Carelli R (2010) Slam-based robotic wheelchair navigation system designed for confined spaces. In: 2010 IEEE international symposium on industrial electronics, IEEE, pp 2331–2336
8. Eppenberger N, Richter MA (2021) The opportunity of shared autonomous vehicles to improve spatial equity in accessibility and socio-economic developments in european urban areas. Eur Trans Res Rev 13(1):1–21
9. European Commission (2021) Electric vehicle charging - standards for recharging points for e-buses. Accessed 22nd July 2021. https://ec.europa.eu/info/law/better-regulation/have-your-say/initiatives/12906-Electric-vehicle-charging-standards-for-recharging-points-for-e-buses_en
10. González E, Cheein FAA (2018) Preliminary results on reducing the workload of assistive vehicle users: a collaborative driving approach. Int J Soc Robot 10(5):555–568
11. Guevara L, Auat Cheein F (2020) The role of 5g technologies: challenges in smart cities and intelligent transportation systems. Sustainability 12(16):6469
12. EA (2021) Global EV Outlook 2021, IEA, Paris. Accessed 22nd July 2021. https://www.iea.org/reports/global-ev-outlook-2021

13. Irie K, Tomono M (2012) Localization and road boundary recognition in urban environments using digital street maps. In: 2012 IEEE international conference on robotics and automation, IEEE, pp 4493–4499
14. Isorna Llerena F, López González E, Caparrós Mancera J, Segura Manzano F, Andújar J (2021) Hydrogen vs. battery-based propulsion systems in unipersonal vehicles–developing solutions to improve the sustainability of urban mobility. Sustainability 13(10):5721
15. Jeon CW, Kim HJ, Yun C, Han X, Kim JH (2021) Design and validation testing of a complete paddy field-coverage path planner for a fully autonomous tillage tractor. Biosyst Eng 208:79–97
16. Jeong S, Jang YJ, Kum D, Lee MS (2018) Charging automation for electric vehicles: is a smaller battery good for the wireless charging electric vehicles? IEEE Trans Autom Sci Eng 16(1):486–497
17. Malmgren I (2016) Quantifying the societal benefits of electric vehicles. World Electric Vehicle J 8(4):996–1007
18. Muramatsu S, Tomizawa T, Kudoh S, Suehiro T (2017) Mobile robot navigation utilizing the web based aerial images without prior teaching run. J Robot Mechatron 29(4):697–705
19. Prado AJ, Michałek MM, Cheein FA (2018) Machine-learning based approaches for self-tuning trajectory tracking controllers under terrain changes in repetitive tasks. Eng Appl Artifi Intell 67:63–80
20. Raboaca MS, Bizon N, Grosu OV (2021) Optimal energy management strategies for the electric vehicles compiling bibliometric maps. Int J Energy Res 45(7):10129–10172
21. Rfidtires (2021) How many cars are there in the world today? Accessed 22nd July 2021. https://www.rfidtires.com/how-many-cars-world.html
22. Romero Schmidt J, Eguren J, Auat Cheein F (2019) Profiling the instantaneous power consumption of electric machinery in agricultural environments: an algebraic approach. Sustainability 11(7):2146
23. Schmidt JR, Cheein FA (2019) Assessment of power consumption of electric machinery in agricultural tasks for enhancing the route planning problem. Comput Electron Agric 163:104868
24. Schmidt JR, Cheein FA (2019) Prognosis of the energy and instantaneous power consumption in electric vehicles enhanced by visual terrain classification. Comput & Electr Eng 78:120–131
25. Shojaei K (2021) Intelligent coordinated control of an autonomous tractor-trailer and a combine harvester. Eur J Control 59:82–98
26. Xu Z, Qu D, Hong J, Song X (2021) Research on decision-making method for autonomous driving behavior of connected and automated vehicle. Complex Syst Complexity Sci 18(3):88–94
27. Yadav R, Kishor G, Kashyap K, et al (2020) Comparative analysis of route planning algorithms on road networks. In: 2020 5th international conference on communication and electronics systems (ICCES), IEEE, pp 401–406
28. Zangina U, Buyamin S, Aman MN, Abidin MSZ, Mahmud MSA (2021) Autonomous mobility of a fleet of vehicles for precision pesticide application. Comput Electron Agric 186:106217