WILEY | Hindawi

Research Article

Mechatronic Modelling of Industrial AGVs: A Complex System Architecture

J. Enrique Sierra-García [ID][1] **and Matilde Santos** [ID][2]

[1]*Department of Electromechanical Engineering, University of Burgos, Burgos 09006, Spain*
[2]*Institute of Knowledge Technology, Complutense University of Madrid, Madrid 28040, Spain*

Correspondence should be addressed to J. Enrique Sierra-García; jesierra@ubu.es

Automatic guided vehicles (AGVs) are unmanned transport vehicles widely used in the industry to substitute manned industrial trucks and conveyors. They are now considered to play a key role in the development of the Industry 4.0 due to their temporal and spatial flexibility. However, in order to deal with the AGV as a potential mobile robot with high capacities and certain level of intelligence, it is necessary to develop control-oriented models of these complex and nonlinear systems. In this paper, the modelling of this vehicle as a whole is addressed. It can be considered composed of several interrelated subsystems: control, safety, driving, guiding and localization, power storage, and charging systems. The kinematics equations of a tricycle vehicle are obtained, and a controller is proposed. An extended hybrid automata formalism is used to define the behaviour of the safety and the control systems, as well as their interaction. In addition, the electrical equivalent circuit of the batteries, charger, and the motors is studied. The architecture of the holistic model is presented. Simulation results of the AGV in a workspace scenario validate the model and prove the efficiency of this approach.

1. Introduction

Automatic guided vehicles (AGVs) are unmanned transport vehicles widely used in the industry to substitute manned industrial trucks and conveyors [1]. They have been proved an efficient element in factory workspaces, helping to reduce logistic errors and operative costs.

Many researchers in manufacturing systems recognize the following Industry 4.0 (I4.0) design principles: interoperability, virtualization, decentralization, real-time capability, service orientation, and modularity. All these principles are connected in some way with the operative of the AGVs. For example, AGVs can easily readjust their program ensuring real-time reaction to unexpected changes in the logistic flows of the production plan. They can also operate in a decentralized way, without any central server commanding orders.

On the other hand, several I4.0 key enabling technologies have been identified in the literature to make these principles

possible: autonomous robots, additive manufacturing, simulation, system integration, cloud computing, Internet of things, cybersecurity, augmented reality, and big data. AGVs are included into the category of autonomous robots as one of these technologies. In addition, this AGV can be easily interconnected and integrated with other machines and devices in the plant by its input/output subsystem. Moreover, these AGVs generate a large amount of data which are transported by IOT protocols such as MQTT, stored and processed by cloud computing systems, and can be exploited by big data technologies to obtain KPIs and optimize the production.

For all these reasons, they are now considered to play a key role in the development of the Industry 4.0. Even more, the new concept of collaborative mobile robots seeks the convergence of AGVs and collaborative robots [2].

However, this synergy demands to deal with the AGV as a potential mobile robot with high capacities and certain level of intelligence. Thus, the necessity of control-oriented

models of these complex and strong nonlinear systems is clear. Even more, to take these robots to a further step, understanding their behavior and establishing safety requirements are a must. This will allow them to advance from assembly tasks to a genuine human-robot collaboration.

An AGV is a multivariable complex mechatronic system composed of several subsystems and with a strongly coupled dynamics. Each of the main elements of an AGV influences its dynamics. The power storage affects the control performance and the motion of the AGV. In turn, the control influences the power consumption and thus the power storage. An AGV usually works in a space shared with other robots, so these other vehicles or even people are perceived as obstacles by the safety subsystem that modifies the working conditions accordingly.

Even more, AGVs are normally used to automate logistics and production flows. These processes involve managing a fleet of robots traveling on the same scenario. Indeed, it is considered a multilayer problem [3], where several issues have to be addressed: the design of the layouts, the warehouse and the production plant, the storage strategies, the traffic scheduling, algorithms to share the charging stations, path planning, navigation, routes assignation, and so on. To take all of these simultaneous processes into account, a global model is necessary. This model will also allow to improve the efficiency of these vehicles and the production optimization.

In this paper, we address the modelling and simulation of an AGV as a whole. First, an architecture that relates all the subsystems that are part of the AGV is proposed. Then, each of them has been described and mathematically modelled: the control, safety, driving, guiding and localization, power storage, and charging systems. Besides, the kinematics and dynamics equations of the vehicle are developed. They have been implemented for a real industrial complex AGV, a hybrid tricycle-differential one. This holistic approach gives realistic results in the simulations of the vehicle, which allows us to validate the models.

The main contributions of this work can be summarized as follows:

(i) The development of a general and high level of the abstraction model of an AGV. It has been divided into subsystems whose models can be used to implement different types of AGVs.

(ii) The identification of all the interactions between the subsystems and its integration.

(iii) The development of the mathematical model of the dynamics of a hybrid differential-tricycle AGV (this hybrid differential-tricycle AGV has not been previously studied in the literature).

(iv) The mathematical description of the safety subsystem and the control subsystem based on the hybrid automata formalism. This approach is completely original.

(v) The implementation of this general model into a specific model of a real industrial AGV, the Unibot AGV of the company ASTI Mobile Robotics [4].

(vi) The integration of the kinematics, dynamics, and electrical equations in the simulation of this type of AGV.

The rest of the paper is organized as follows. Section 2 presents a brief state of the art on the AGV modelling. In Section 3, the architecture of the AGV is developed. The description and equations of the different subsystems which compose it are presented in Section 4. Simulation results on a working scenario are discussed in Section 5. The document ends with the conclusions and future works.

2. Related Works

The first paper published on the design and operation of AGV systems was the work by Maxwell and Muckstadt, in 1982 [5], where a methodological framework for specifying the operational characteristics of an automatic guided vehicle system was presented. Since then, the research on this topic has not stopped growing. Nevertheless, the works found in the literature mainly tackle the system focusing on individual aspects of the AGV: kinematics, control, charging, and so on, that are not independent. Indeed, the majority focus on its kinematics and control, or the localization algorithms, or the power storage, or the charging system, but they lack a holistic approach of the guided vehicle as a whole system.

There have been several attempts to model the AGV, for example, Veiga et al. compared vehicle models of a tugger-trolleys system with three linked towed. One model is a complete multibody model of the system. The second is a simplified version composed of vertical and handling models for motion in a plane, assuming no planar forces on the trolleys' front wheels and linking the vehicles with multibody joints and body, representing the coupling arms, developed in Modelica language. Results show good agreement between the models but the second one was much faster and provided more stable results [6].

A kinematical and dynamical analysis of a mobile robot belonging to the configuration classical tricycle is shown in [7]. For the kinematical analysis, authors consider a robot capable of locomotion on a surface by the action of wheels mounted on the robot. The dynamics of the nonholonomic mobile robot is obtained based on the Euler–Lagrange methodology. The paper is focused on the analysis of global asymptotic stability of a new position controller for this wheeled mobile robot.

In [8], authors state that the complexity of AGV dynamics and the difficulty of obtaining the actual vehicle dynamic parameters have led them to use two nonmodel-based control approaches. The robot is equipped with a front-centered steerable wheel and fixed parallel rear wheels. They present a comparison of control techniques for robust docking maneuvers of an AGV, specifically, fuzzy control and vector pursuit that do not require a mathematical model of the vehicle dynamics.

Recently, Montazerijouybari et al. proposed a mathematical model based on the kinematics of 2-DOF AGVs with differential driving wheels and caster wheels in order to

predict the behaviour of the robot when the vehicle is carrying heavy loads not well distributed on it.

A trajectory tracking control of an AGV based on sliding mode control with an improved reaching law is proposed in [9]. The AGV in this paper has two driving wheels and two universal wheels. They consider a fully functional AGV composed of the motion mechanism, the sensing system, and the control system. Authors do not consider the rest of the systems of the vehicle. The work designs a double closed-loop control method to control the vehicle to track the given trajectory. With the same control-oriented aim, in [10], the analysis of the kinematics and dynamics of a four-wheel omnidirectional automated guided vehicle is presented.

There are other AGV-related topics that are dealt with in this paper, such as localization and navigation. In many cases, the routes implemented by these vehicles are subject to constant changes and are adapted to the currently assigned task. Thus, the sensors that allow the AGV to follow the path are an important element of the guided vehicle. Although usually the vehicle sensors are not modelled, the most common ones have been studied. Just a couple of examples are as follows. An intelligent path recognition system for vision-guided AGVs, able to tackle noise in the images captured by using a CCD camera, is presented in [11]. The paper addresses some of the most challenging problems for a vision-based AGV running in a complex workspace and improves the accuracy and reliability of vision recognition of guide paths for AGVs by means of artificial intelligence methods. Authors in [12] proposed an interesting method to measure the wheel radius. This radius is not constant, and in the case of automated guided vehicles using odometry, this is a key information to calculate the position of the vehicle. Stetter et al. designed virtual sensors that estimate forces and torques acting on an AGV [13]. However, this AGVs consists of four sprung arms that each dispose of one drive motor. For this system, they obtain a simple description of the AGV dynamics and then develop a set of virtual sensors that enables estimation of the forces and torques. Indeed, authors say that they try to avoid the use of complex tire models. The study is meant for several diagnostic purposes such as fault detection or fault prevention.

Another AGV component, the charging system, has been also the object of study by some authors. For instance, a battery management simulation is developed for evaluating battery-related costs under various AGV operation modes and for designing battery management strategies [14]. Hou et al. developed a novel energy modelling method which allows to calculate and predict the AGV energy consumption [15]. However, in this work, we do not consider energy efficiency either the design of the battery, but the simulation of the electrical performance of the AGV's battery. The charging method is out of the scope of this paper.

Finally, there are several works about the control of a fleet of AGVs. A detailed state of art on the design and control of several AGVs is presented in [16]. In this paper, author discusses the literature concerning the usage of AGVs in manufacturing and other new areas of these vehicle application, namely, distribution, transshipment, and transportation systems. As a conclusion, the author states that the most important differences between the usage of AGVs in manufacturing areas and in other environments can be noticed in the number of AGVs used. Besides, the author claims the necessity of analytical and simulation models which simultaneously address multiple design and control problems. Different vehicle platooning control strategies are presented in [17, 18].

As shown above, most of the studies found in the literature are mainly focused on individual aspects of the AGV design problem: kinematics, control, guiding, energy, and so on. However, these components are not independent; indeed, these subsystems are highly coupled. Therefore, a holistic and complete vision of these automatic vehicles is missing, where the interactions among their various elements are evident in the models used so that the simulation is realistic.

3. Description of the Industrial AGV Architecture

There are different ways to split up an AGV into its different parts. We have considered the following interconnected subsystems that describe the whole behaviour of the vehicle: driving system, guiding and localization system, safety system, motors, power storage and charging, and control system. These elements interact with the working environment. Figure 1 shows the AGV architecture with all the mentioned systems and their relationships.

The control system can be considered the brain of the AGV. Depending on the localization, on the external inputs and the current state, it adjusts the charging system, generates target signals for the driving system, commands the outputs, and adjusts the safety system. The driving system collects the dynamic and kinematic of the AGV motion; it changes the position and orientation of the AGV and modifies the working environment. It also gives information about the current to the control system in order to close the control loop. The driving system is the component that demands more energy from the power system.

The safety system detects obstacles in the safety zone and warns the control system; it can even trigger an emergency stop. The charging system is responsible for recovering the energy of the power storage system. This subsystem is usually placed out of the vehicle when there is not enough space in the AGV. The power storage is in charge of providing energy to the electronic components and the motors.

The guiding and localization system (GLS) manages AGV position and attitude measures, which are used by the control system. Depending on the technologies used to implement the GLS, this information will have greater precision. There are technologies, such as simultaneous localization and mapping (SLAM), which provide information about the absolute coordinates (x, y, angle) of the AGV with high accuracy. Others only provide the deviation respect to a predefined trajectory, either marked on the floor or buried, such as magnetic antennas, optical sensors, and so on, and must be complemented with a localization system such as RFID tag sensors (low accuracy) or UWB devices (medium accuracy) to know the absolute position.

On the other hand, the input/output system allows to introduce information to the AGV from other external sources: machines, operators, and so on, and extract information. Finally, the workspace represents the external environment and everything that can produce any changes to the AGV behaviour.

In this paper, we are going to model a hybrid differential-tricycle AGV. This vehicle has been selected because nowadays is one of the most used AGVs in the industry. An example of this type of AGVs is the Unibot AGV of the company ASTI Mobile Robotics [4], represented in Figure 2.

The Unibot is mainly used to convey loads by pulling carts. This vehicle is robust, simple, flexible, and reliable and well-established in the market vehicles. Though with reduced dimensions, it has great load capacity. It is fully connected in real time.

This industrial tow AGV can be used in whatever industry where medium loads need to be towed. However, it achieves its full potential in enterprises where recurrent flows of pieces and/or products exist. They can be applied in any SMEs and large companies (for further information on enterprise categories see [19]) although it is commonly found in the automotive industry as it is one of the most automatized sectors. Moreover, it is easy to include them in already functioning enterprises as these automatic guided vehicles do not require to modify the infrastructure in order to implement the layout. Indeed, the layout can be usually easily removed and moved to adapt it to changes in the production plant. However, also in companies that are going to start the production, it is possible to design the plant taking into account the layouts of the AGVs to optimize the intralogistics.

In addition, it has many other functionalities such as magnetic guidance/navigation system (surface/buried), online charging system, safety laser for internal environment, safety PLC, and emergency stop button, among others.

This is a complex system because it has two wheels as a differential robot and a tricycle configuration in the front part. Most of the models found in the literature work as differential robots.

4. Modelling of the AGV's Subsystems

4.1. Driving System. There are many types of AGVs regarding the kinematics, but four different models can be typically identified: tricycle, differential, quad, and omnidirectional. Figure 3 shows some standard configurations of the wheels for these AGVs. Depending on the number and type of the wheels, the kinematic constraints and the control variables of the vehicle will vary. The drive wheels provide traction but cannot change its direction. The steer wheels can change the direction but cannot provide traction. The steer and drive wheels provide both things, and they are also called motor-wheels.

All of these wheels are subjected to nonholonomic constraints to prevent wheel lateral slipping. This can disrupt the control of the vehicle and must be avoided. By contrast, the omnidirectional wheels only provide traction, but they
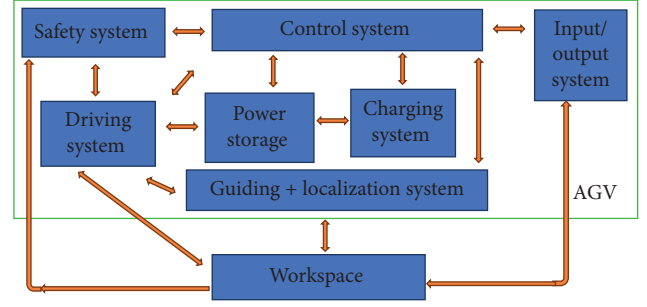

Figure 1: Subsystems AGV architecture.

are not affected by the nonholonomic constraints. In fact, the omni-wheels are equipped with rollers to enable the lateral slipping.

The Unibot AGV [4] is a mixture of tricycle and differential. The traction head is a differential robot; meanwhile, the body is linked with the traction unit by an axle, and it performs as a tricycle. The wheels of the body do not provide traction but support the weight of the vehicle and allow the body to follow the traction head.

As said before, the driving system includes the dynamics and the kinematics of the AGV motion. Therefore, in this section, both the kinematic and dynamic models of the hybrid tricycle-differential AGV are developed. To do it, the kinematic equations of the individual traction tricycle and differential units are combined in a novel way as the hybrid tricycle-differential AGV has not been previously studied in the literature. In addition, the dynamic model of the vehicle has been developed by the authors using the Newton–Euler fundamental equations of the translational and rotational dynamics.

Figure 4 represents the mechanical components of this AGV and the coordinate system (traction head, green square and AGV body, blue rectangle). The center of the wheels of the traction head in the inertial coordinate system is represented by (x_h, y_h), and the position of the center of the wheels of the body is (x_b, y_b), both in m. In this inertial frame, the attitude of the body is Φ_b (rad) and the attitude of the head is Φ_h (rad); the angle between the body and the head is γ (rad). L_b is the distance between (x_b, y_b) and (x_h, y_h) in m, and L_h is the distance between the wheels of the traction unit in m.

The equations of the traction head are given by the kinematic model of a differential robot [20]. That is,

$$\dot{x}_h = \frac{V_l + V_r}{2} \cos(\Phi_h),$$

$$\dot{y}_h = \frac{V_l + V_r}{2} \sin(\Phi_h), \qquad (1)$$

$$\dot{\Phi}_h = \frac{V_r - V_l}{L_h}.$$

On the other hand, the motion of the body follows the equations of a tricycle:

$$\dot{x}_b = v_h \cos(\gamma)\cos(\Phi_b),$$

$$\dot{y}_b = v_h \cos(\gamma)\sin(\Phi_b) \tag{2}$$

$$\dot{\Phi}_b = \frac{v_h}{L_b}\sin(\gamma),$$

where v_h is the longitudinal velocity of the steering drive (m/s), i.e., the traction head speed. These equations (1) and (2) are linked by the following expressions:

$$v_h = \sqrt{\dot{x}_h^2 + \dot{y}_h^2} = \frac{V_l + V_r}{2},$$

$$\gamma = \Phi_h - \Phi_b, \tag{3}$$

$$\gamma_{\min} \leq \gamma \leq \gamma_{\max},$$

where $\gamma_{\min}$ and $\gamma_{\max}$ are the mechanical limits of the angle of the traction head (rad). These mechanical limits are set to avoid damage in the wires between the body and the traction head.

Besides, in order to prevent sliding in the wheels, the following constraints must be met, equation (4) for the rear wheels and equation (5) for the front wheels:

$$\dot{x}_b \sin(\Phi_b) - \dot{x}_b \cos(\Phi_b) = 0, \tag{4}$$

$$\dot{x}_b \sin(\Phi_b + \gamma) - \dot{x}_b \cos(\Phi_b + \gamma) - \dot{\Phi}_b L_b \cos(\Phi_b) = 0, \tag{5}$$

with L_b is the distance between the center of the traction head and the rear axle.

By the translational and rotational Newton–Euler dynamic equations, the following expressions that represent the forces and momentums are obtained:

$$\begin{bmatrix} \dfrac{m_T \cdot R_h}{2} & \dfrac{m_T \cdot R_h}{2} \\[2ex] \dfrac{I_h \cdot R_h}{L_h} & -\dfrac{I_h \cdot R_h}{L_h} \end{bmatrix}\begin{bmatrix} \ddot{\theta}_r \\[1ex] \ddot{\theta}_l \end{bmatrix} = \begin{bmatrix} \dfrac{(M_{er} + M_{el})}{2R_h} - f_{rT} \\[2ex] \dfrac{(M_{er} - M_{el})L_h}{2R_h} - M_{rR} \end{bmatrix},$$

$$f_{rT} = 0.5 \cdot \delta_{\text{air}} \cdot S_{\text{AGV}} \cdot C_{\text{aero}} \cdot (v_h^2) \cdot \text{sign}(v_h)$$
$$+ g \cdot m_T \cdot C_{\text{roll}} \cdot \text{sign}(v_h),$$

$$M_{rR} = F_{vh} \cdot \dot{\Phi}_h + F_{sh} \cdot \text{sign}(\dot{\Phi}_h),$$

$$M_{er} = M_r - F_{vw} \cdot \dot{\theta}_r - F_{sw} \cdot \text{sign}(\dot{\theta}_r),$$

$$M_{el} = M_l - F_{vw} \cdot \dot{\theta}_l - F_{sw} \cdot \text{sign}(\dot{\theta}_l), \tag{6}$$

where m_T (kg) is the total mass of the system, that is, the mass of the AGV, m_{AGV} (kg), plus the mass of the load, m_L (kg); the radius of the wheels in the traction unit is R_h (m); I_h is the rotational inertia of the traction unit (kg m^2); θ_r and θ_l are the angular accelerations of the right and left wheel of the traction unit (rad/s^2), respectively; M_{er} and M_{el} are the
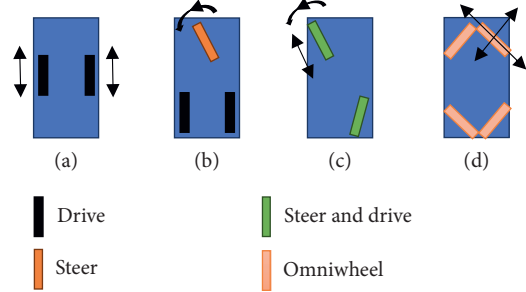

Figure 2: Unibot AGV.

Figure 3: Wheel configurations in AGVs: (a) differential; (b) tricycle; (c) quad; (d) omnidirectional.
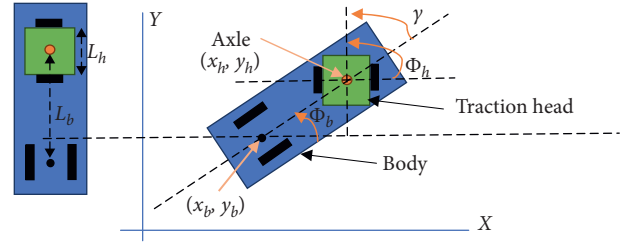

Figure 4: Unibot AGV driving system and the coordinate system.

effective torques in the right and left wheel of the traction unit (Nm); f_{rT} is the translational friction force (N); M_{rR} is the friction torque (Nm); δ_{air} is the density of the air (Kg/m^3); S_{AGV} is the front surface of the air (m^2); C_{aero} is the aerodynamic coefficient; C_{roll} is the rolling coefficient; g is the acceleration of the gravity; F_{vh} and F_{sh} are the viscous and static friction coefficients in the traction unit (Nm and Nms/rad); F_{vw} and F_{sw} are the viscous and static friction coefficients in the wheels (Nm and Nms/rad); M_{er} and M_{el} are the torques produced by the motors (Nm); and sign is the sign function. The effect of the inertia of the wheels has been neglected. It has been supposed that the AGV is working in an indoor environment, and the wind speed is zero.

Moreover, in the simulations, the slippage has been considered. The slippage happens when the traction force exceeds the surface friction force. The traction forces in the front wheels are given by M_{er}/R_h and M_{el}/R_h. Therefore, to avoid the slippage, the following constraints must be fulfilled:

$$\frac{M_{er}}{R_h} < \frac{g \cdot m_T \mu}{4},$$

$$\frac{M_{el}}{R_h} < \frac{g \cdot m_T \mu}{4}, \tag{7}$$

where μ is the friction coefficient. The number 4 that appears in the denominator means that the weight of the AGV is equally distributed into the four wheels.

4.2. Safety System. In this section, we develop the mathematical model of a general safety subsystem applying a new approach to do this, the hybrid automata formalism. The safety system is in charge of preventing collisions with humans and objects. The safety norms establish that the AGV must stop in the near presence of obstacles. This normally leads to introduce a processor in the safety system to take the control and trigger an emergency stop command when the control system is not operative.

Figure 5 shows the interactions between the driving, the safety, and the control systems. The safety system monitors the velocities of the wheels (v_l, v_r) by encoders. If the current longitudinal velocity is higher than the predefined current safety velocity, v_{safe}, during a configurable period of time, then the safety system fires an emergency stop procedure. This is represented in Figure 5 as the stop signal. Normally, this is carried out by acting directly on the brake of the vehicle or on the motors. The control system is also informed about this anomaly situation by the stop signal.

On the other hand, the control system notifies the safety system about what type of safety zones must be activated at each moment. These safety zones are polygonal areas where there cannot be obstacles. Each zone has a code$\in \mathcal{N}$ and a maximum velocity, $v_{safe} \in \mathcal{R}$ (m/s), which is calculated considering the deceleration rate of the AGV to ensure that it is able to stop before a collision. The control system knows the location of the AGV and, for instance, if the AGV is towing a trolley, the safety zones must be big enough to cover it. Moreover, if the AGV is going to turn left, the safety zones must be defined on this side of the AGV. If the safety system detects an obstacle inside the safety perimeter, it

automatically sends the code of the zone to the control system, with the signal code, and updates the safety velocity.

The AGV safety system is equipped with different types of sensors: mechanical, ultrasounds, infrared, or LIDAR, to detect the obstacles. In the Unibot AGV, the safety system has a PLC and a LIDAR sensor installed in the front of the vehicle. The safety zones can be mathematically defined as a set $Z_i \subseteq \mathcal{R}^4$ of pairs of points relative to the center of the LIDAR (x_{LID}, y_{LID}). In addition, each safety zone can be classified as warning zone or error zone. Therefore, by the union of all zones, it is possible to conform the set Zones$\subseteq Z_i \times \mathcal{N} \times \mathcal{R}$.

The LIDAR returns a set of distance measures, P_{DIST}, that are defined in polar coordinates as follows:

$$
\begin{aligned}
P_{DIST} &\subset (r, \theta) \in (\mathcal{R} < r_{max}) \times \theta_C|, \\
&\forall (r_1, \theta_1), (r_2, \theta_2)\theta_1 = \theta_2 \Rightarrow r_1 = r_2, \\
\theta_C &\subset \theta \in \mathcal{R}|\theta = (i - 1) \cdot res + \theta_{min}, \\
&i \in \mathcal{N} \le N_p.
\end{aligned}
\tag{8}
$$

The size of this set is $N_p = res \cdot (\theta_{max} - \theta_{min})$, where res is the angular resolution, and θ_{min} and θ_{max} are minimum and maximum angles detected by the LIDAR. This set may be transformed to Cartesian coordinates by the following function:

$$
\begin{aligned}
f_{xy}&: P_{DIST} \longrightarrow \mathcal{R}^2, \\
P_{XY} &= f_{xy}(r, \theta) = (r\cos(\theta), \rho\sin(\theta)).
\end{aligned}
\tag{9}
$$

(1) The subset of allowed zones, Zones$_{EN} \subseteq$Zones, is obtained according to the type of zone, type$_{zone}$, selected by the control system. The safety PLC checks if any of the points in P_{XY} belongs to the set of allowed zones by the function: $f_{zone}: P_{XY} \longrightarrow$ Zones$_{EN}$ (9).

$$
f_{zone}(x, y) = \begin{cases} (\varnothing, 0, v_{smax}), & \forall (z, c, v) \in \text{Zones}_{EN} | (x, y) \notin \text{Polygon}(z), \\ (z, c, v) \in \text{Zones}_{EN} | (x, y) \in \text{Polygon}(z), & \exists (z, c, v) \in \text{Zones}_{EN} | (x, y) \in \text{Polygon}(z), \end{cases}
\tag{10}
$$

where the function Polygon$(z): Z_i \longrightarrow (\mathcal{R}, \mathcal{R})^M$ gives the list of points within the perimeter of the safety zone. If there is not any obstacle in an allowed safety zone, the zone given by (10) is $(\varnothing, 0, v_{smax})$, being v_{smax} (m/s) the AGV maximum velocity. This v_{smax} is usually defined in the datasheet of the vehicle, and it normally goes from 0.8 m/s to 2 m/s.

If more than one zone is obtained by (10), the safety velocity, v_{safe}, is set as the minimum safe velocity. This can be expressed as follows:

$$
v_{safe} = \text{MIN}(v|(z, c, v) = f_{zone}(x, y), \quad \forall (x, y) \in P_{XY}).
\tag{11}
$$

The model of the safety controller can be described by the hybrid automata formalism. A hybrid automata is a formal model that combines discrete control graphs, usually called finite state automata, with continuously evolving variables. An extension to the hybrid automata H defined in [21] is a tuple $H = (Q, X, U, Y, \text{Init}, D, \text{Dom}, E, G, R)$, where

(i) **Q** is a set of discrete variables, and Q is countable. Also it is also known as the set of discrete states of the automata.

(ii) **X** is a set of continuous variables, that is, the set of continuous state variables of the automata.
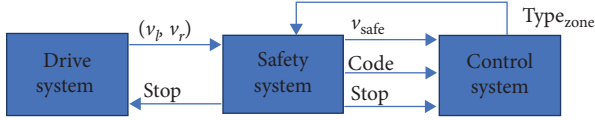
(iii) **U** is a set of input variables.

FIGURE 5: Interactions between driving system, safety system, and the control system.

(iv) **Y** is a set of output variables.
(v) **Init**$\subseteq Q \times X \times U$ is a set of initial states.
(vi) **D**: $Q \times X \times U \longrightarrow TX$ is a vector field of the differential equations.
(vii) **Do m**: $Q \longrightarrow P(X)$ assigns a domain to each $q \in Q$.
(viii) **E** $\subset Q \times Q$ is a collection of discrete transitions.
(ix) **G**: $E \longrightarrow P(X \times U, X \times U)$ assigns a guard condition to each $e = (q, q') \in E$.
(x) **R**: $E \times X \times Y \longrightarrow P(X \times Y)$ assigns a reset relation to each $e = (q, q') \in E$ and $x \in X$.

Let $\mathbf{Q} = \{q_1, q_2, q_3, q_4\}$ be the set of discrete variables; $\mathbf{X} = \{t\}$ is the set of continuous variables; $\mathbf{U} = \{v_l, v_r, \text{type}_{\text{zone}}, P_{\text{DIST}}\}$ is the set of input variables;

$\mathbf{Y} = \{\text{stop} \in \{0, 1\}, v_{\text{safe}}, \text{code}\}$ is the set of output variables, and **Init** $= \{q_1, t = 0, \text{stop} = 0\} \in Q \times X \times U$ is the initial. When the variable stop $= 1$, the emergency stop is triggered. As expected, the safety controller receives as inputs the velocity of the wheels, the type of allowed zones, and the points measured by the LIDAR; its output is the stop signal, the current safe velocity, and the code of the active safety zone.

The differential equations are given by

$$\mathbf{D}(q_1, x) = \mathbf{D}(q_3, x) = 0, \mathbf{D}(q_2, x) = \mathbf{D}(q_4, x) = 1. \quad (12)$$

The domains are $\mathbf{Do\,m}(q_1) = \mathbf{Do\,m}(q_3) = 0$, $\mathbf{Do\,m}(q_2) = \{t \in \mathcal{R} < t_2\}$, and $\mathbf{Do\,m}(q_4) = \{t \in \mathcal{R} < t_4\}$. In addition, the discrete transitions are given by

$$\mathbf{E} = \{(q_1, q_2), (q_2, q_1), (q_2, q_3), (q_3, q_4), (q_4, q_3), (q_4, q_1)\}. \quad (13)$$

The guard conditions are the following:

$$\mathbf{G}(q_1, q_2) = \{(t, v_l, v_r, \text{type}_{\text{zone}}, P_{\text{DIST}}) \in X \times U \mid \frac{(v_l + v_r)}{2} > v_{\text{safe}}(\text{type}_{\text{zone}}, P_{\text{DIST}})\},$$

$$\mathbf{G}(q_2, q_1) = \{(t, v_l, v_r, \text{type}_{\text{zone}}, P_{\text{DIST}}) \in X \times U \mid \frac{(v_l + v_r)}{2} \leq v_{\text{safe}}(\text{type}_{\text{zone}}, P_{\text{DIST}})\},$$

$$\mathbf{G}(q_2, q_3) = \{(t, v_l, v_r, \text{type}_{\text{zone}}, P_{\text{DIST}}) \in X \times U \mid t \geq t_2\},$$

$$\mathbf{G}(q_3, q_4) = \{(t, v_l, v_r, \text{type}_{\text{zone}}, P_{\text{DIST}}) \in X \times U \mid \frac{(v_l + v_r)}{2} \leq v_{\text{safe}}(\text{type}_{\text{zone}}, P_{\text{DIST}})\}, \quad (14)$$

$$\mathbf{G}(q_4, q_3) = \{(t, v_l, v_r, \text{type}_{\text{zone}}, P_{\text{DIST}}) \in X \times U \mid \frac{(v_l + v_r)}{2} > v_{\text{safe}}(\text{type}_{\text{zone}}, P_{\text{DIST}})\},$$

$$\mathbf{G}(q_4, q_1) = \{(t, v_l, v_r, \text{type}_{\text{zone}}, P_{\text{DIST}}) \in X \times U \mid t \geq t_4\},$$

where v_{safe} is calculated as a function of $\text{type}_{\text{zone}}$ and P_{DIST} by equations (9)–(11) and t_2 and t_4 are parameters of the safety controller which can be adjusted by the user.

The reset relations can be expressed as follows:

$$\mathbf{R}(q_1, q_2, x, \text{stop}) = \{0, \text{stop} = 0\},$$
$$\mathbf{R}(q_2, q_1, x, \text{stop}) = \{t, \text{stop} = 0\}, \quad (15)$$

$$\mathbf{R}(q_2, q_3, x, \text{stop}) = \{t, \text{stop} = 1\},$$
$$\mathbf{R}(q_3, q_4, x, \text{stop}) = \{0, \text{stop} = 1\}, \quad (16)$$

$$\mathbf{R}(q_4, q_3, x, \text{stop}) = \{t, \text{stop} = 1\},$$
$$\mathbf{R}(q_4, q_1, x, \text{stop}) = \{t, \text{stop} = 0\}. \quad (17)$$

The outputs: v_{safe} and code, are calculated in the same way for all discrete states, by equations (10) and (11); thus, they have not been included in (15)–(17).

To summarize, if an obstacle is detected in a warning zone, the maximum allowed speed is reduced. If the obstacle is in an error zone, the maximum speed is automatically set to zero. In both cases, the control system receives a warning, and it must follow the commands of

the safety system. If the safety system detects that the AGV velocity is higher than the allowed speed for a period of time, the safety chain is broken and the vehicle is stopped.

4.3. Guiding and Localization System

The guiding and localization functionalities of the AGV can be implemented by only one device, such as a navigation system based on laser reflectors or the SLAM navigation systems. Both technologies are commonly complemented with the use of an accurate odometry of the vehicle [22]. To combine this information that come from different sensors, a Kalman filter or another similar mathematical tool can be used. However, there are situations, such as slippage produced by an inadequate load distribution or a slippery surface, where the estimation of the speed using only the encoders of the wheels is not possible. In these cases, an inertial measurement unit (IMU) embedded in the vehicle can help to estimate the AGV velocity. However, this solution increases the cost of the AGV, and thus many mobile robots are not equipped with the IMU neither implement a Kalman filter. This is the case of the AGV studied here. As it will be explained below, in this AGV, guiding and localization are decoupled. Guiding (understood as deviation from the expected trajectory without localization) is provided by the magnetic sensor, and the localization is provided by a RFID reader.

The guiding system allows us to know the deviation between the AGV and the path defined in the working environment. There are different types of guiding sensors: optical ones (to follow a painted line), magnetic ones (to follow a magnetic tape), inductive ones (to follow a buried wire), and so on. In the AGV used in the study, the guiding is based on a magnetic sensor.

The localization system detects points of reference placed in the working environment, with a known specific location on the plane. These reference points may be sensed by an optical device (QR codes), a magnetic device (magnetic spots), an electromagnetic device (RFID tags), and so on. This way the AGV can know its own location through references. The accuracy of the localization system is half the distance between two points. In the standard Unibot robot, the localization is RFID-based.

The traction unit of the AGV equipped with the guiding sensor and the magnetic tape on the floor is shown in Figure 6. The guiding sensor gives the error signal err_{gui}, that is, the deviation respect to the planned trajectory. If the sensing point is at the right of the magnetic tape, the error is considered positive; otherwise, it is negative.

4.4. Control System.

The main contribution of this section is the formal description of a general control subsystem of the AGV, as well as the identification of the interactions with the other subsystems. Figure 7 shows the interaction between the subsystems driving and safety, the guiding and location systems, as well as the internal structure of the control system (orange dashed rectangle). The control system has three internal blocks: the control unit, the velocity control, and a block called MIN that calculates the minimum of its inputs.

The control unit has as inputs the localization variable from the GLS and the stop signal from the safety system. Its outputs are the reference of the cruise velocity, v_{ref}, and the type of safety zone, $\text{type}_{\text{zone}}$, associated to the section of the route where the AGV is currently located.

The inputs of the MIN block are v_{ref} and v_{safe} (from the safety system). It selects the minimum of them and calculates the effective velocity that the AGV should have, v_{foll}.

Finally, the velocity control module is in charge of generating the control signals for the motors (u_l, u_r) in order to track the reference longitudinal velocity v_{foll} and to minimize the error signal, err_{gui}. To do it, it receives the velocity of the wheels (v_l, v_r) from the driving system and the error signal err_{gui} from the GLS. Even more, it is notified if the stop signal is triggered, and it also receives the references for the wheel speeds $(\text{man}, u_{\text{ml}}, u_{\text{mr}})$ directly from the user if the manual controller is on. It interacts with the workspace through a set of input/output signals $(\text{IO}_{\text{signals}})$.

The velocity control is carried out by the function f_{vcon}, which receives as inputs v_{foll} (m/s) from the MIN block, err_{gui} (m) from the GLS, (v_l, v_r) (m/s) from the driving system, $(u_{\text{ml}}, u_{\text{mr}})$ (m/s) from an external control device, and the digital signals $(\text{stop}, \text{man}) \in \{0, 1\}^2$. The stop signal, as it was explained, comes from the safety system. The man signal is generated in an external control device. It is 1 when the manual external controller device is on and 0 otherwise. The output of the f_{vcon} function is the pair of control signals for the motors $(u_l, u_r) \in \{u \in \mathcal{R} \mid -t1n \leq quh \leq_1\}^2$. Thus, this function $f_{\text{vcon}}(v_{\text{foll}}, \text{err}_{\text{gui}}, v_l, v_r, u_{\text{ml}}, u_{\text{mr}}, \text{stop}, \text{man}): \mathcal{R}^6 \times \{0, 1\}^2 \longrightarrow \{u \in \mathcal{R} \mid -t1n \leq quh \leq_1\}^2$ can be defined as follows:
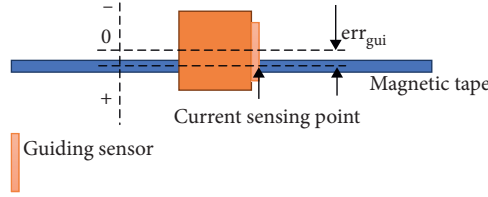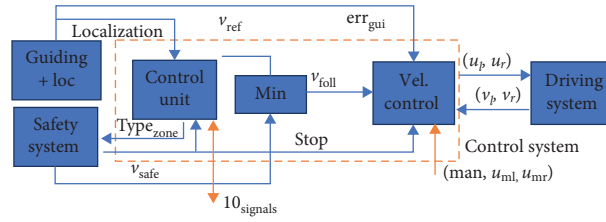
Figure 6: Guiding sensor of the AGV.

Figure 7: The control system and the interaction with other subsystems.

$$\dot{\Phi}_{href} = \text{PID}\left(\text{err}_{gui}\right) = K_{\Phi P} \cdot \text{err}_{gui} + K_{\Phi D} \cdot \dot{\text{err}}_{gui} + K_{\Phi I} \int \text{err}_{gui}, \tag{18}$$

$$v_{lref} = v_{foll} - \frac{\left(\dot{\Phi}_{href} \cdot L_h\right)}{2},$$
$$v_{rref} = v_{foll} + \frac{\left(\dot{\Phi}_{href} \cdot L_h\right)}{2}, \tag{19}$$

$$(u_l, u_r) = f_{vcon}(\ldots) = \begin{cases} (0, 0), & \text{if stop} = 1, \\ (u_{ml}, u_{mr}), & \text{if stop} = 0 \wedge \text{man} = 1, \\ (\text{PI}(v_{lref}, v_l), \text{PI}(v_{rref}, v_r)), & \text{if stop} = 0 \wedge \text{man} = 0. \end{cases} \tag{20}$$

As it may be observed in (19), when stop signal is active, the motor targets are set to 0. However, when the manual mode is active, the external velocity control signals (u_{ml}, u_{mr}) are changed to (u_l, u_r). On the other hand, if automatic mode is on (last statement in (20)), the control output is generated by a couple of PID controllers, in particular a PI controller per wheel. The targets for the left and right wheels are v_{lref} and v_{rref}, respectively. These targets are calculated by equation (19) using as inputs the longitudinal velocity reference, v_{foll}, and the angular velocity reference, $\dot{\Phi}_{href}$. The angular reference is computed by a PID to compensate the guiding error, err_{gui}.

As explained before, the control unit is in charge of assigning the type of safety zone, type_{zone}, and a cruise velocity reference, v_{ref}. A lookup table is generated by the user with localization as input and this information.

Likewise with the safety controller, the control unit can be modelled as a hybrid automata. It has three macrostates: automatic, manual, and safety stop. Therefore, the set of discrete states is $\mathbf{Q_{CS}} = \{\text{auto}, \text{manual}, \text{safestop}\}$; the set of continuous variables is $\mathbf{X_{CS}}$, that is, defined by the user according to the actions to be carried out by the AGV. $\mathbf{U_{CS}} = \{\text{Localization}, \text{man}, \text{inp}_{prog}\}$ is the set of input variables and $\mathbf{Y_{CS}} = \{\text{type}_{zone}, v_{ref}, \text{out}_{prog}\}$, where the inputs inp_{prog} and outputs out_{prog} are configured by the user in the lookup table. The initial state is **Init CS** $= \{\text{auto}, \text{type}_{zone} = 0, v_{ref} = 0\} \in Q_{CS} \times X_{CS} \times U_{CS}$.

The set of discrete transitions is given by

$$\mathbf{E_{CS}} = \{(\text{auto}, \text{manual}), (\text{auto}, \text{safestop}), (\text{manual}, \text{auto}), (\text{manual}, \text{safestop}), (\text{safestop}, \text{auto}), (\text{safestop}, \text{manual})\}. \tag{21}$$

The set of guard conditions are given by

$$\mathbf{G_{CS}}(\text{auto}, \text{manual}) = G(\text{safestop}, \text{manual}) = \{\text{man} = 1 \wedge \text{stop} = 0\},$$
$$\mathbf{G_{CS}}(\text{manual}, \text{auto}) = G(\text{safestop}, \text{auto}) = \{\text{man} = 0 \wedge \text{stop} = 0\}, \tag{22}$$
$$\mathbf{G_{CS}}(\text{auto}, \text{safestop}) = G(\text{manual}, \text{safestop}) = \{\text{stop} = 1\}.$$

The hybrid automata is defined for each macrostate: H_{auto}, H_{man}, and H_{stop} by a tuple $H = (Q, X, U, Y, \text{Init}, D, \text{Dom}, E, G, R)$, as defined before. In this work, we are going to focus on H_{auto} because it has more interactions with the rest of subsystems. When in the automatic macrostate, the set of discrete states is $\mathbf{Q_{CSauto}} = \{w\text{con}_i, \text{march}_i\}\, i \in \mathcal{N} < N_{\text{loc}}$, where N_{loc} is the number of known localizations. In other words, the number of microstates is $2 \cdot N_{\text{loc}}$. The parameter N_{loc} is configured by the user for each application. The discrete variables $w\text{con}_i$ mean "waiting for conditions," and the variables march_i mean "the AGV is in motion." Therefore, for each localization defined in the lookup table, the AGV will be either waiting for conditions or in motion once the conditions are fulfilled.

The set of continuous variables in this macrostate, $\mathbf{X_{CSauto}}$, and its corresponding vector field of differential equations, $\mathbf{D_{CSauto}}$, are defined by the user according to the actions to be carried out by the AGV. The user can associate

to each localization$_i$ in the lookup table a set of inputs $\text{inp}_{i\text{prog}}$, a set of outputs $\text{out}_{i\text{prog}}$, a set of conditions Cond_i, a cruise velocity reference $v_{i\text{prog}}$, and a type of safety zone $tsf_{i\text{prog}}$. Thus, the set of inputs in this macrostate is $\mathbf{U_{CSauto}} = \{\text{Localization}\} \cup \text{inp}_{\text{prog}}$, and the set of outputs is $\mathbf{Y_{CSauto}} = \{\text{type}_{\text{zone}}, v_{\text{ref}}, \} \cup \text{out}_{\text{prog}}$.

The set of discrete transitions in the automatic mode is as follows:

$$\mathbf{E_{CSauto}} = \{(w\text{con}_i, \text{march}_i)\}\, i \in \mathcal{N}$$
$$< N_{\text{loc}} \cup \{(\text{march}_i, \text{cond}_j)\}\, (i, j) \in \mathfrak{R}_{\text{prog}}, \tag{23}$$

where $\mathfrak{R}_{\text{prog}}$ is a user-defined relationship. This relationship defines the sequence of localization values of an action and so it is necessary to specify a pair of microstates $(\text{march}_i, \text{cond}_i)$. As shown in (23), the jump between two microstates can only be done if its sequence appears in $\mathfrak{R}_{\text{prog}}$.

From the relationship $\mathfrak{R}_{\text{prog}}$, the set of guards is automatically generated as follows:

$$\mathbf{G_{CSauto}}(w\text{con}_i, \text{march}_i) = \{\text{Cond}_i\}, \quad i \in \mathcal{N} < N_{\text{loc}}, \tag{24}$$

$$\mathbf{G_{CSauto}}(\text{march}_i, w\text{con}_j) = \{\text{localization} = j \wedge (i, j) \in \mathfrak{R}_{\text{prog}}\}, \quad (i, j) \in (\mathcal{N}^2 < N_{\text{prog}}, i \in N_{\text{loc}}), \tag{25}$$

where $\{\text{Cond}_i\}\, i \in \mathcal{N} < N_{\text{loc}}$ is a set of conditions. It is possible to observe in (23) and (24) how the AGV in the microstate $w\text{con}_i$ waits for the meeting of condition Cond_i. When they are fulfilled, it switches to the microstate march_i.

And vice versa, when it is at march_i state and it reaches a localization defined in $\mathfrak{R}_{\text{prog}}$, it changes to $w\text{con}_j$.

Finally the set of reset relations is given by

$$\mathbf{R_{CSauto}}(w\text{con}_i, \text{march}_i) = \left\{\text{type}_{\text{zone}} = tsf_{i\text{prog}}, v_{\text{ref}} = v_{i\text{prog}}, \text{out}_{\text{prog}} = \text{out}_{i_{\text{prog}}}\right\}, \quad i \in \mathcal{N} < N_{\text{loc}}. \tag{26}$$

$$\mathbf{R_{CSauto}}(\text{march}_i, w\text{con}_j) = \left\{\text{type}_{\text{zone}} = tsf_{j\text{prog}}, v_{\text{ref}} = 0, \text{out}_{\text{prog}} = \text{out}_{j_{\text{prog}}}\right\}, \quad (i, j) \in \mathfrak{R}_{\text{prog}}. \tag{27}$$

As expressed in (26), when the AGV reaches the microstate march$_i$, it sets v_{ref} to the value $v_{i prog}$ stored in the lookup table associated to localization$_i$.

4.5. Electrical System: Charging, Power Storage, and Motors. The novelty of this section is the integration of the electrical models of the battery, the charger, the motors, and the drives. The most frequently power storage technologies used in AGVs are the lead-based and lithium batteries. More recently, supercaps, electric capacitors with an unusually high power density, are being used for very high charging currents demand. However, still lead batteries are the most widely used because they are much cheaper than the other options. There are several mathematical models of these batteries, each one considering different aspects. The electrical model is adapted from [23, 24] to include the charger and the motors, as shown in Figure 8.

The electrical model of the motors (Figure 8) is adapted from [25–27]:

$$V_{\text{mot}_i} = I_{\text{mot}_i} \cdot R_m + L_m \cdot \dot{I}_{\text{mot}_i} + K_m \cdot w_{\text{mot}_i}, \quad i \in \{L, R\},$$

$$J \cdot \dot{w}_{\text{mot}_i} + K_r \cdot w_{\text{mot}_i} + M_{X_i} = K_m \cdot I_{\text{mot}_i}, \quad i \in \{L, R\},$$

$$M_i = K_m \cdot I_{\text{mot}_i}, \quad i \in \{L, R\},$$

$$(28)$$

$$V_{\text{mot}_i} = u_i \cdot V_{\text{bat}}, \quad i \in \{L, R\}, \tag{29}$$

$$\text{abs}\left(V_{\text{mot}_i} \cdot I_{\text{mot}_i}\right) = \eta \cdot V_{\text{bat}} \cdot I_{M_i}, \quad i \in \{L, R\}. \tag{30}$$

Regarding motor i, V_{mot_i} (V) and I_{mot_i} (A) are the voltage and the current; R_m is the motor winding resistor (Ω); L_m is the motor inductance (H); K_m is the electromotive constant (kgm^{2}s^{-2}A^{-1}); w_{mot_i} is the rotor angular velocity (rad/s); J is the moment of inertia of the rotor (kgm^2); K_r is the coefficient of rotation resistance; M_{X_i} is the load moment (kg m^2/s^2); η is the performance coefficient; I_{M_i} is the input current (A); and abs is the absolute value function; the control signals of the motors u_L and u_R are calculated by equation (33), and they are the inputs of the motor drivers.

The main equations of the equivalent electrical circuit of the AGV and the charger are as follows:

$$I_{\text{bat}} + I_c \cdot u_c = \frac{V_{\text{bat}}}{R_c} \cdot u_c + I_{\text{stop}} + I_{M_L} + I_{M_R},$$

$$I_{R2} = I_p + I_{\text{bat}},$$

$$I_{R2} = I_{R1} + I_{c1}, \quad I_{C1} = -C_1 \cdot \dot{V}_C, \tag{31}$$

$$Em_{\text{bat}} = V_c + R_2 \cdot I_{R2} + R_0 \cdot I_{\text{bat}} + V_{\text{bat}},$$

$$\dot{Em}_{\text{bat}} = -\frac{I_{\text{bat}}}{(3600 \cdot C_n)},$$

where I_{bat} (A) and V_{bat} (V) are the current and the voltage in the terminals of the battery; I_{stop} (A) is the current consumed by the AGV when it is stopped; I_c (A) is the current given by the charging station; R_c (Ω) is the equivalent resistance of the charger; I_p (A) is the parasitic current in the battery when is charging; Em_{bat} (V) is the voltage in the open circuit of the battery; $R_1(\Omega)$, $R_2(\Omega)$, and C_1(F) are internal parameters of the battery; and C_n (Ah) is the capacity of the battery. The control signal u_c is active, $u_c = 1$, when the AGV is in a charging station. For more details of these variables and parameters, see [23].

4.6. Workspace. The workspace, although is not embedded in the AGV as the rest of the subsystems, plays an important role in the performance of the automatic vehicle. It represents the environment where the AGV moves. It collects information about the ground, the obstacles, and other possible AGVs and provides information to the safety system, the guiding and localization system, the driving system, and the control system.

Figure 9 shows the interaction among these subsystems. The workspace gives the landmarks and references to the GLS to calculate the guiding error err$_{gui}$ and localization. It also detects obstacles that will be handled by the safety system. In addition, it interacts with the driving systems by the friction coefficients and the ramps of the ground. In turn, the driving system moves the AGV, changing its position coordinates (x, y). Finally, the workspace sends IO signals to the control system that may come from other machines [28], AGVs, and human operators, and it also receives commands from the control system.

The workspace can be simulated as a lookup table, where the inputs of the table are the AGV position coordinates (x, y), and the output is a set that provides information about the obstacles, the landmarks and references, the friction coefficients, the ramps, and the IO signals in that spatial location.

5. Simulation Results and Discussion

Several simulations have been carried out to validate the analytical modelling approach [29] and analyze the AGV model. The model has been implemented with Matlab/Simulink software package. In order to reduce the discretization error, a variable simulation step size has been used, with maximum step size set to 25 ms. The control sample time Tc has been also fixed to 25 ms. The AGV parameters used in the simulations are listed in Table 1. The load mass m_L is set to 500 kg.

The simulation scenario is represented in Figure 10. It is a magnetic tape loop (green line) with a charging station and a traffic light. The charging station supplies the energy to the vehicle, so the AGV does not need to leave the circuit to reload. This component, part of the workspace, has been included in the electrical equations of Section 4.5 due to its electrical nature.

A typical element that interacts with the AGV through the IO System is a traffic controller (shown in Figure 10 as a traffic light). When the AGV arrives at an intersection, it stops; then, it requests crossing priority to the traffic
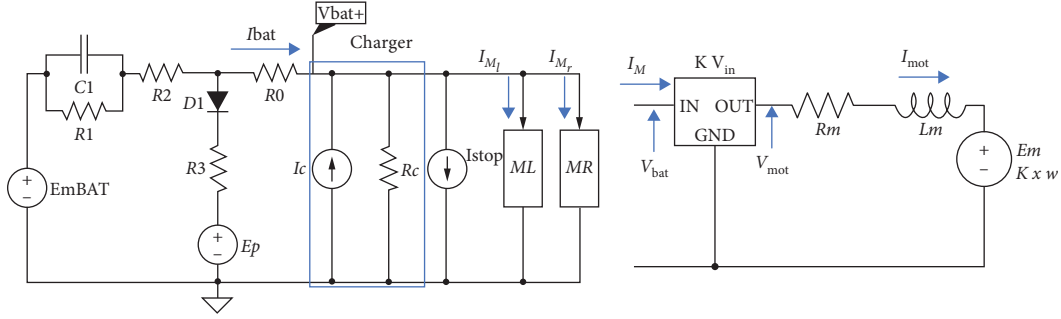
FIGURE 8: Equivalent circuit of an AGV and charger (a) and of one motor (b).
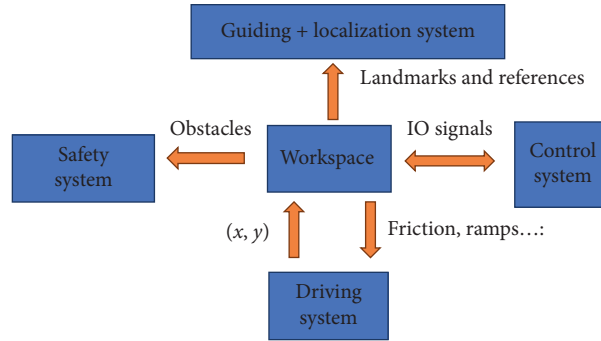

FIGURE 9: Interaction between the workspace and the AGV subsystems.

controller, and when the intersection is free, the traffic controller informs the AGV and the vehicle continues moving.

The trajectory proposed in Figure 10 is Bernoulli's lemniscate. This trajectory is interesting because the curvature radius is variable and even more, it has an intersection point as it happens in many real layouts. It can be described as the set of P points where the product of the distance to two focus, $F1$ and $F2$, located at a distance $2a$ between them, is a^2.

$$\left(x^2 + y^2\right)^2 - 2a^2\left(x^2 - y^2\right) = 0. \qquad (32)$$

The intersection between the straight line projected by the guiding sensor and the lemniscate is used to calculate the guiding error. To calculate the crossing points, we substitute the well-known straight-line equation $y = mx + b$ in (32) and obtain a fourth-degree polynomial (33), with the constants k_0 to k_4 defined in the following:

$$k_4 x^4 + k_3 x^4 + k_2 x^4 + k_1 x + k_0 = 0, \qquad (33)$$

$$k_4 = m^4 + 2m^2 + 1, \qquad (34)$$

$$k_3 = 4m^3 b + 4mb, \qquad (35)$$

$$k_2 = b^2\left(2 + 6m^2\right) - 2a^2\left(1 + m^2\right), \qquad (36)$$

$$k_1 = 4m\left(b^3 + a^2 b\right), \qquad (37)$$

$$k_0 = b^4 + 2a^2 b^2. \qquad (38)$$

The solution of this polynomial gives the x coordinate of the intersection points. As it is fourth-degree polynomial, the analytical solution is not evident so it is solved by numerical methods.

During the simulation, it has been supposed that there are not ramps in the environment, all the localizations have the same friction coefficient, and friction coefficients are considered constant due to the large system time constants.

For the sake of clarity, the inputs and outputs of the simulation blocks used during the experiments are shown in Table 2.

5.1. Kinematic Behaviour Simulation.

Figure 11 shows the trajectory followed by the traction unit (Figure 11(a)) and the body of the AGV (Figure 11(b)) when reference is a lemniscate curve with a sinusoidal velocity profile, with $v_{\text{ref}} = 0.4$ m/s of amplitude and 0.4 m/s of mean value. In this experiment, the a parameter of the lemniscate is set to 5, and the initial position of the AGV is (3.144, 2.5) with an angle of 0° in the traction unit and the body.

This figure represents 200 steps of the AGV motion on the XY plane, between the initial and the final position. It is possible to observe how the positions described by the traction unit follow a lemniscate-shaped trajectory. This type of representation is very useful to design the layout where the AGV is going to travel in a factory or workhouse. It allows to foresee, among other cases, whether the AGV will be blocked by an obstacle, get stuck in a narrow corridor, or if it will reach correctly a station.

TABLE 1: Parameters of the wind turbine model.

| Parameter	Description	Value/units
L_h	Distance between wheels of the AGV	30 cm
L_b	Distance between rear wheels and traction unit	100 cm
R_h	Radius of front wheels	6 cm
R_b	Radius of rear wheels	9 cm
m_{AGV}	Mass of the AGV	100 kg
I_h	Inertia of traction unit	0,11 kg m^2
C_{roll}	Rolling coefficient	0.01
C_{aero}	Aerodynamic coefficient	0.35
F_{sh}	Static friction coefficient of traction unit	0.1 N
F_{vh}	Viscous friction coefficient of traction unit	0.01 Ns/rad
F_{sw}	Static friction coefficient of traction wheels	$2.94e - 2$
F_{vw}	Viscous friction coefficient of traction wheels	$5e - 4$ Ns/rad
g	Acceleration of the gravity	9.8 m/s^2
η	Performance coefficient of the motor driver	0.9
I_{stop}	Current when AGV is stopped	1.5 A
R_C	Resistance of the charger	10 KΩ
$[R_0, \ R_1, \ R_2]$	Resistances of the electrical model	$[0.05, 0.1, 0.05] \, \Omega$
C_1	Capacitance of the battery	1500 F
I_p	Parasitic intensity	50 mA
R_m	Resistance of the motor	1.2 ohm
L_m	Inductance of the motor	$5.8e - 4$H
K_m	Electromotive constant of the motor	0.66 kgm^{2}s^{-2}A$^-$1
C_n	Capacity of the battery	150 Ah
$[K_{vp}, K_{vI}]$	PID constants for velocity control of wheels	[2, 0.1]
$[K_{\Phi p}, K_{\Phi D}, K_{\Phi I}]$	PID constants for angular velocity control of the traction unit	[9.8, 1, 0.1]
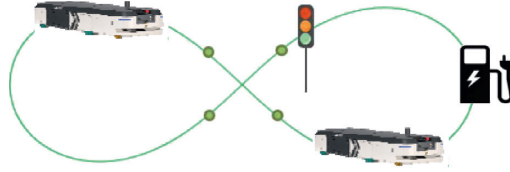
FIGURE 10: Simulation scenario.

The trajectories followed by the traction unit (red line) and the body (orange line) of the AGV are represented in Figure 12, with the reference (blue line). The trajectory on the plane is shown on the left, and the evolution of each coordinate is represented on the right. The coordinates (x_H, y_H) represent the center of the traction unit, and (x_B, y_B) are the coordinates of the middle of the rear axle.

The traction unit follows reasonably well the lemniscate-shaped magnetic tape; thus, the reference and the traction unit lines are overlapped in Figure 12(a). Nevertheless, the body of the AGV follows a lemniscate curve closer than the reference. This difference decreases with the width of the lemniscate, parameter a in (30). This effect is more visible in the edges of the lemniscate, and it is not shown at the intersection.

Looking closely at Figure 12(b), local maximums and minimums appear at $x \approx 7$ m, $x \approx -7$ m, $y \approx 2.5$ m, and $y \approx -2.5$ m, the extremes of the lemniscate, as expected. It is also interesting to note how when the AGV is in the first and third quadrant, from $t = 0$ to $t \approx 10$ s and from t around 60 s to 100 s, x_H seems to be delayed respect to x_B. However, when it is in the second and fourth quadrants, the opposite happens.

Figure 13(a) represents the same trajectory as in Figure 12 in order to help to understand the angles shown in Figure 13(b). On the right side of this Figure 13, the angle of the traction unit (blue line) and of the body (red line) of the AGV is shown.

The angle of the body of the AGV decreases up to 225° around $t = 28$ s when the straight part of the lemniscate starts, and it keeps constant until that straight part of the lemniscate end about to $t = 40$ s. After this point, the AGV turns counterclockwise, which increases the angle up to approximately $t = 70$ s when the straight part of the lemniscate starts again. On the other hand, the angle of the traction unit is related to the radius of curvature. At the top and bottom edges of the lemniscate, this radius slightly varies and the angle of the traction unit keeps practically constant at −25° from $t = 10$ s up to $t = 25$ s and at 25° from $t = 50$ s up to $t = 65$ s. This angle is positive when the AGV turns clockwise and negative otherwise. In the straight part of the lemniscate, as expected, the angle of the traction unit is 0°.

So, these results validate the model as the AGV moves as expected.

TABLE 2: Inputs and outputs of the simulation blocks.

Subsystem	Inputs	Outputs
Control system	v_{safe}, err_{gui}, stop, localization	u_l, u_r, $type_{zone}$
Safe system	$type_{zone}$, x_b, y_b, Φ_b, obstacles	v_{safe}, stop
Electrical system	u_l, u_r	M_l, M_r
Driving system	M_l, M_r	x_b, y_b, Φ_b, x_h, y_h, Φ_h, $\dot{\theta}_l$, $\dot{\theta}_r$
Guiding and localization system	x_h, y_h, Φ_h, layout	err_{gui}, localization
Workspace	x_b, y_b	Layout, obstacles

(a)

(b)
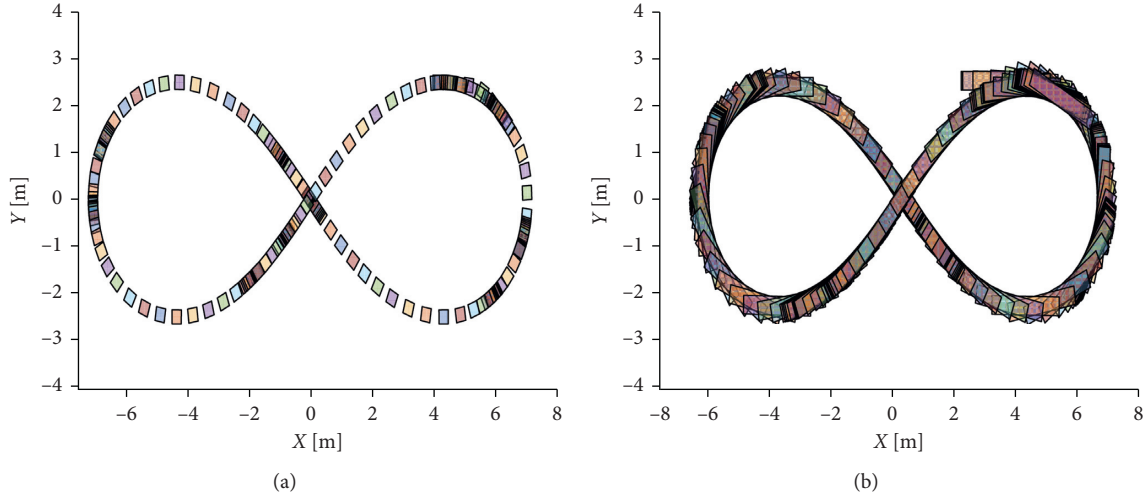
FIGURE 11: Movement of the traction unit (a) and movement of the body of the AGV (b).

(a)

(b)
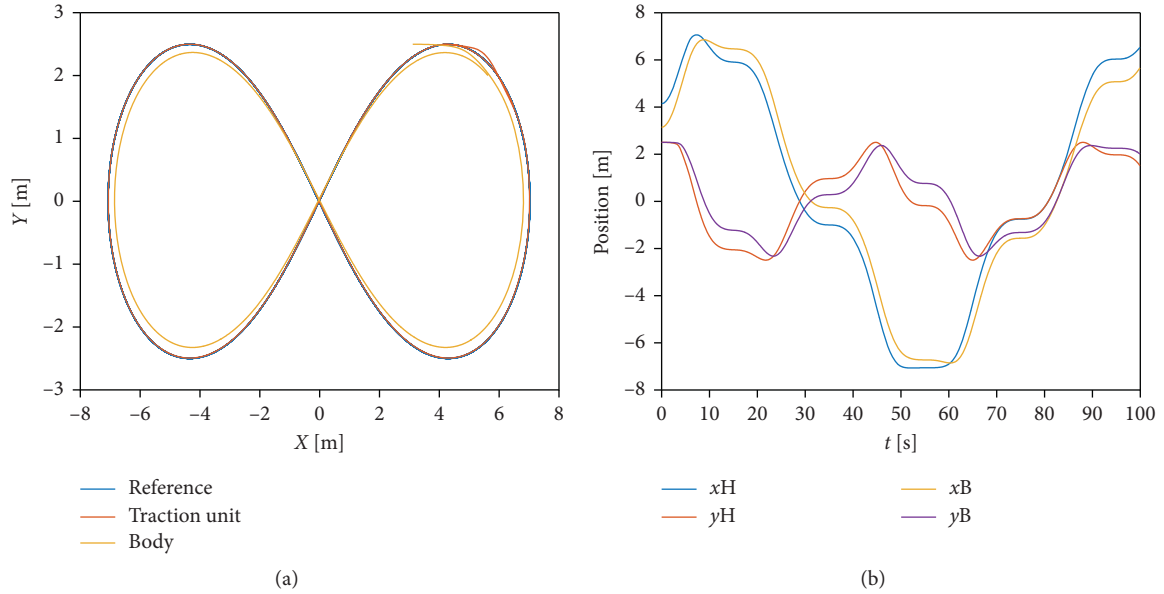
FIGURE 12: Trajectory of the AGV (a) and x-y coordinates of the traction unit and the body (b)

5.2. Dynamic Behaviour Simulation. The model of the dynamics of the AGV has been also implemented. Figure 14(a) shows the angular velocity of the traction unit (red line) and its reference (blue line). Figure 14(b) represents the reference of v_{ref} (blue line) and the longitudinal velocity (red line). When the AGV starts moving, the angular speed reference, that is, generated by the controller, oscillates due to error of the velocity. Once the reference speed is reached, at $t = 8$ s, the oscillations are much smaller and the mean value is around 0. In any case, the angular velocity reference is well tracked, with a slight oscillation (smaller than 0.5 rad/s). The sinusoidal profile of v_{foll} (Figure 14(b) is well followed by the
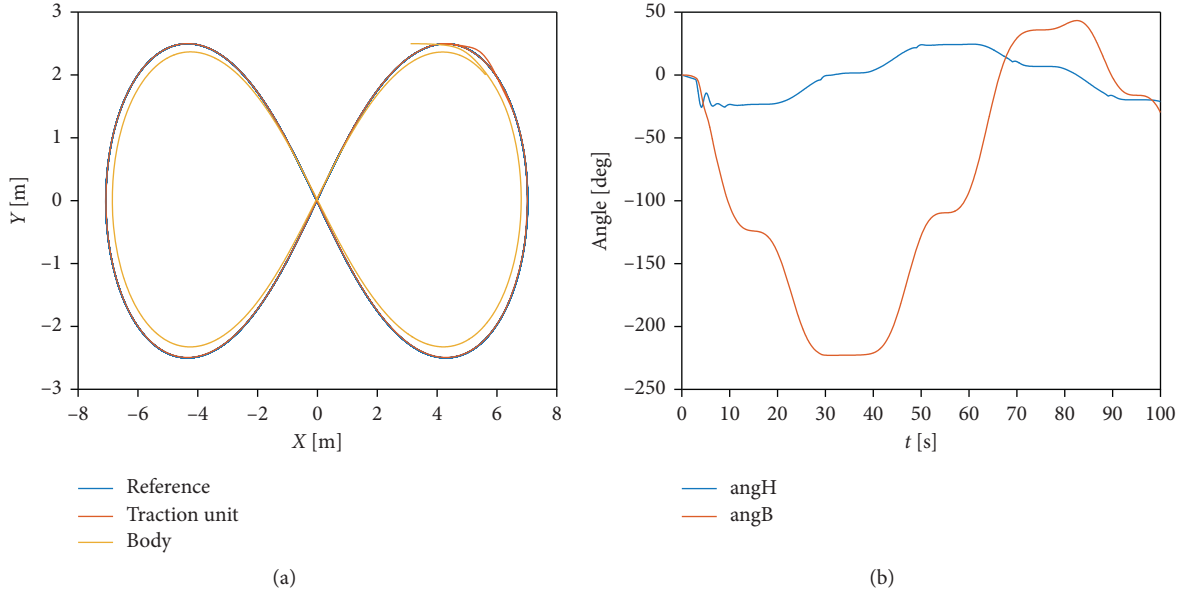
(a)

(b)

FIGURE 13: Trajectory of the AGV (a) and angle of the AGV (b).

controller. The biggest error appears at the beginning of the simulation as the AGV is stopped, and it starts accelerating to reach the reference speed.

In order to reach these angular and longitudinal velocities, the corresponding wheel speeds are calculated. Figure 15(a) shows these references, and Figure 15(b) shows the effective wheel velocities. The blue lines represent the right wheel and the red lines the left one. As it is possible to see, the controller is able to track the wheel speed references. The largest error appears at the beginning of the simulation as it is necessary to overcome system inertia. Once the reference value is achieved, the biggest difference between the velocities of the wheels appears at the inflection points of the angle of the traction unit. This happens approximately at $t = 25$ s, 45 s, 65 s, and 85 s. At these points, if the angular velocity of the traction unit is positive, the speed of the right wheel is bigger than the left one, and it is smaller in the opposite case.

In order to follow the wheel speed references (Figure 15(a)), the control has to generate the right power values. Figure 16(a) shows the control signals which are the inputs of the electrical model equation (27); in Figure 16(b), the torque of the wheels is represented. As explained before, this torque is the result of subtracting the friction forces from the torque provided by the motors. The blue lines represent the right wheel and the red lines the left one. The range of the control signals is $[-1, 1]$. These control signals are inputs in equation (29), and they are dimensionless; they represent the percentage of battery voltage that is applied in the motors.

At the beginning of the simulation, both control signals are saturated to 1, as the initial state of the AGV is stopped. These values are maintained until the wheel speed reference is reached; from this point, the control signals show a sinusoidal behaviour similar to the reference but slightly noisier. Negative peaks appear at t ≈ 30 s, 50 s, 70 s, and 90 s. These values match with the peaks of Figure 14(a). They

correct the angular velocity of the traction unit to maintain the angle when the AGV gets close to the top and bottom edges of the lemniscate trajectory. Due to this, the curvature radius is practically constant (Figure 13).

As expected, the shape of the effective torque in the wheels is similar to the control signals but with different amplitude. At the beginning, when the control signals are saturated to 1, the value of the torque is maximum. Then, the effective torque decreases although the control signals are constant due to the effect of the viscous friction forces which increase with the velocity.

5.3. Electrical Behaviour Simulation. In this section, the same experiment is carried out but the analysis is focused on the electrical model. At the initial state, the voltage in the capacitor is $V_c = -4$V, and Em_{bat} is set to 24 V, that are the common values after the charging. The initial current in the motors is 0 A.

Figure 17 shows the current in the motors on the left and the effective torque on the right. The blue lines represent the right motor and the red lines the left one. As expected from equation (26), the absolute value of the effective torque is practically proportional to the current. The difference comes from the friction forces. In the simulations, the current of the motor is considered always positive no matter the sign of the control signal. The sign of the torque in each motor is equal to the sign of its control signal.

In addition to the current of the motors, it is also interesting to study the current and the voltage in the battery terminals. This helps us to scale up the layout and the production cycles, as well as to foresee the battery life. Figure 18 shows the current and voltage in the battery terminals (blue and red lines), the voltage in open circuit of the battery (yellow line), and the voltage in the equivalent capacitor (purple line). As expected, the battery current
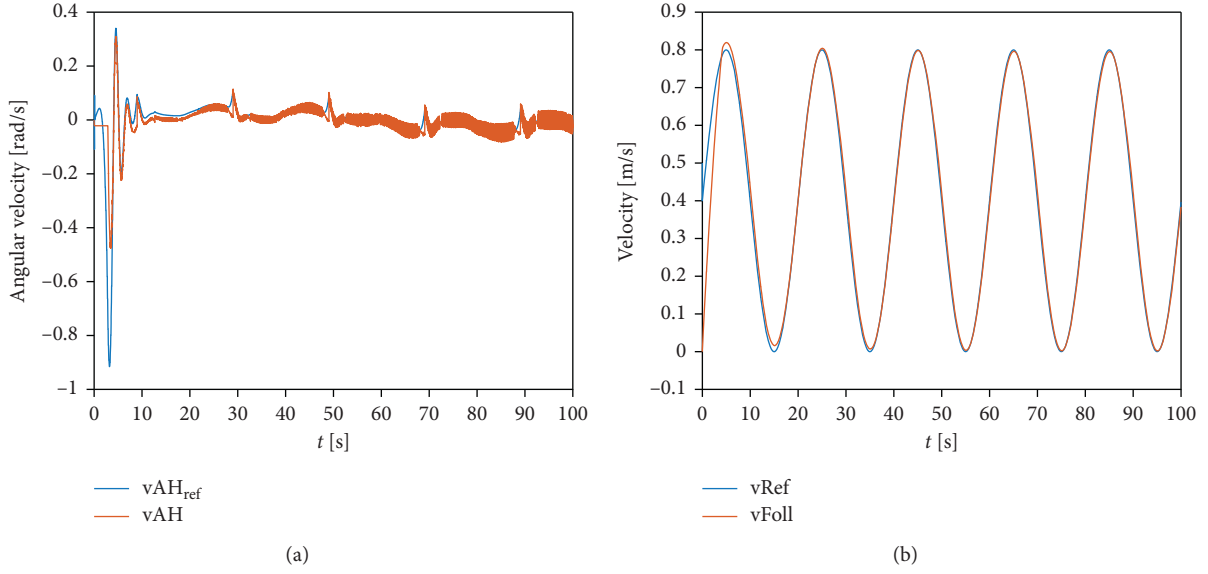
(a)

(b)

FIGURE 14: Angular velocity of the traction unit (a) and longitudinal velocity of wheels (b).
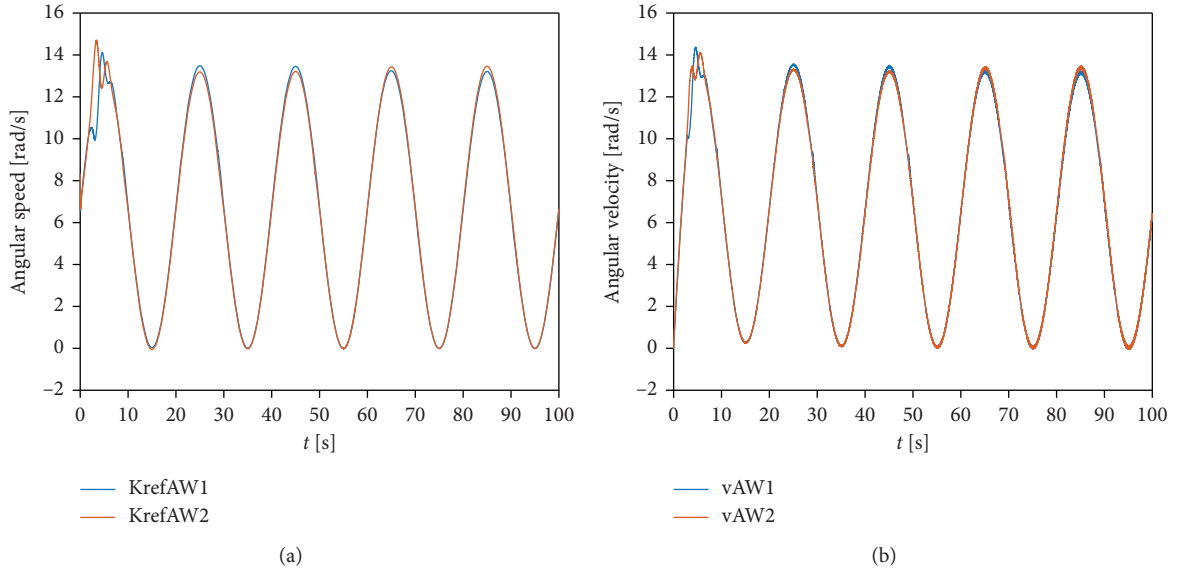

(a)

(b)

FIGURE 15: Wheel angular speed references (a) and effective wheel angular velocities (b).

follows a sinusoidal signal. Although sometimes the current in the motors is 0 A, the current in the battery is bigger due to the consumption of the other electronic devices of the AGV. This consumption is considered constant and modelled by I_{stop}. The negative peaks of the torque can be seen in the battery current as positive peaks at the bottom part of the signal.

The battery voltage measured in terminals varies with the current; when the battery current increases, the voltage decreases due to the effect of the internal resistance, producing an observable sinusoidal curve. When the system is working, the absolute value of the voltage in the capacitor decreases. This also produces a decreasing trend in the battery voltage. On the other hand, the Em_{bat} slightly decreases as well but this is not noticeable due to the high

capacity of the battery, i.e., the bigger the capacity, the slower the decreasing trend.

Another interesting effect is that when the battery voltage decreases, the voltage applied to the motor for the same control value also decreases. And if the motor voltage decreases, its speed also decreases. This forces to increase the control signal to get the same motor velocity. In turn, the increment of the control makes the consumed current to increase, as it is possible to see in Figure 18. This effect is less significant when the capacitor is discharged since the decreasing trend of the battery voltage is much smaller.

Figure 19 shows the total accumulated current in Ah (Figure 19(a)) and the percentage (%) of the battery (Figure 19(b)). As expected, the sinusoidal current in the battery terminal produces a sinusoidal total current. It is a
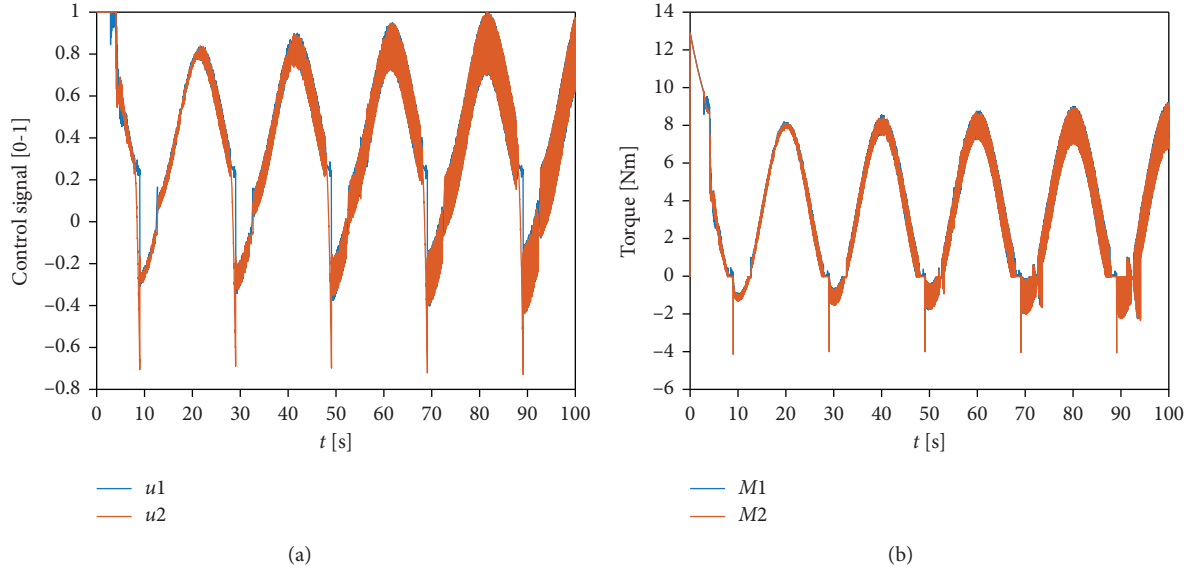
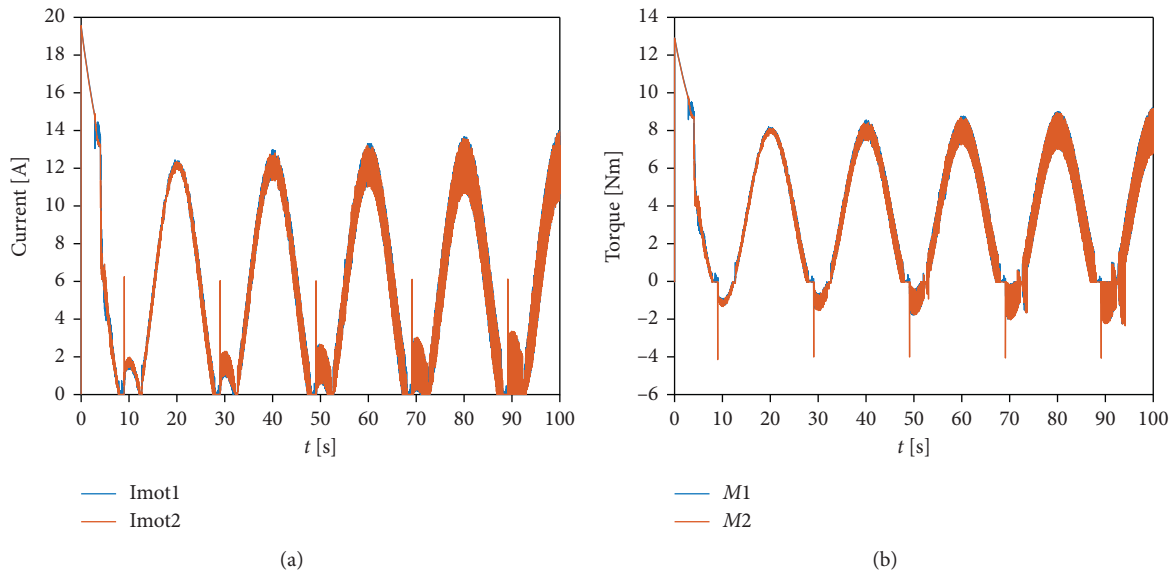FIGURE 16: Control signals for each wheel (a) and torque of the wheels (b).

FIGURE 17: Motor current (a) and effective torque in wheels (b).

usual practice to calculate the percentage of the battery using the total consumed current. In this case, the capacity of the battery is 150 Ah; thus, a total current of 0.3 Ah is equivalent to 0.2% battery, as shown in Figure 19(b). This provides an autonomy of approximately 13.8 h. This is an approximation since it has been seen, the battery voltage also affects the current.

5.4. Influence of the Mass of the Load.
The effect of the load on the AGV has also to be analyzed. To do it, different experiments varying the load mass have been carried out with two different speed profiles: a sinusoidal one, as before, and a constant speed of 0.8 m/s. All these experiments

are configured as in the previous sections but the mass. In the previous experiments, the load mass m_L was set to 500 kg.

Figure 20 represents the variation of the average battery power with the mass (Figure 20(a)) and the variation of the average current (Figure 20(b)). The blue lines represent the values for a sinusoidal speed profile and the red lines for a constant velocity. As expected, the bigger the load mass, the larger the power and current. This effect is more marked when the speed is constant. As the AGV travels more time at a bigger speed, it needs more energy.

Figure 21 shows the variation of the MSE guiding error (Figure 21(a)) and the variation of the standard deviation of the total torque in the wheels (Figure 21(b)). The color code
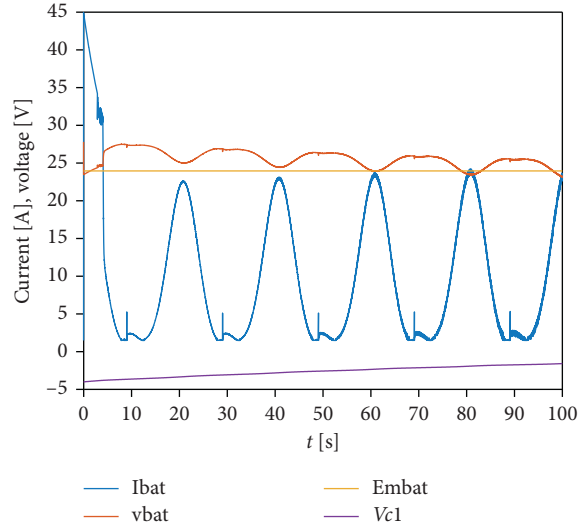
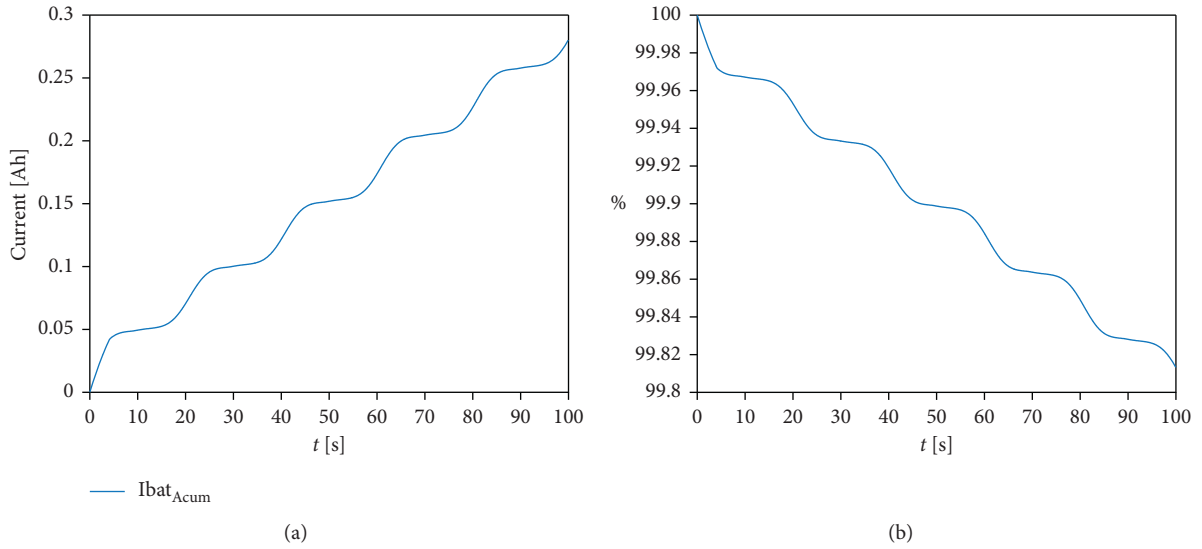FIGURE 18: Current and voltage in the battery.

(a)

(b)

FIGURE 19: Total consumed current (a) and percentage of battery (b).

is the same as in Figure 20. These indicators have been calculated using the following equations:

$$\mathrm{MSE}\,[\mathrm{cm}] = 100\sqrt{\frac{1}{T_{\mathrm{sim}}}\sum_i \left(\mathrm{err}_{\mathrm{gui}_i}\right)^2 T_{s_i}}, \qquad (39)$$

$$\overline{M_e}\,[\mathrm{Nm}] = \frac{1}{T_{\mathrm{sim}}}\sum_i \left(\left|M_{eL_i}\right| + \left|M_{eR_i}\right|\right) T_{s_i}, \qquad (40)$$

$$\mathrm{SD}_{M_e}\,[\mathrm{Nm}] = \sqrt{\frac{1}{T_{\mathrm{sim}}}\sum_i \left[\left(\left|M_{eL_i}\right| + \left|M_{eR_i}\right|\right) - \overline{M_e}\right]^2 T_{s_i}}, \qquad (41)$$

where T_{s_i} is the sampling time at i and T_{sim} is the total simulation time. This period T_{s_i} is necessary since a variable simulation step has been used. This way the MSE gives the

mean squared value of the guiding error in cm. The standard deviation of the total torque can give us an idea of the degradation of the electromechanical pieces, as larger total torque means bigger degradation.

In Figure 21, it is possible to observe how the MSE is not affected by the mass until certain mass threshold is surpassed. This value is around 300 kg for the sinusoidal speed profile and 150 kg for the constant speed. This may be explained as the average velocity with the sinusoidal profile is approximately half than the obtained with the constant speed. However, the effect of the mass on the standard deviation of the torque is worse. The sinusoidal speed profile produces bigger deviations, and moreover, the influence of the mass is bigger.

5.5. *Influence of the AGV Velocity.* Finally, the effect of using two AGV velocity profiles, sinusoidal and constant, is
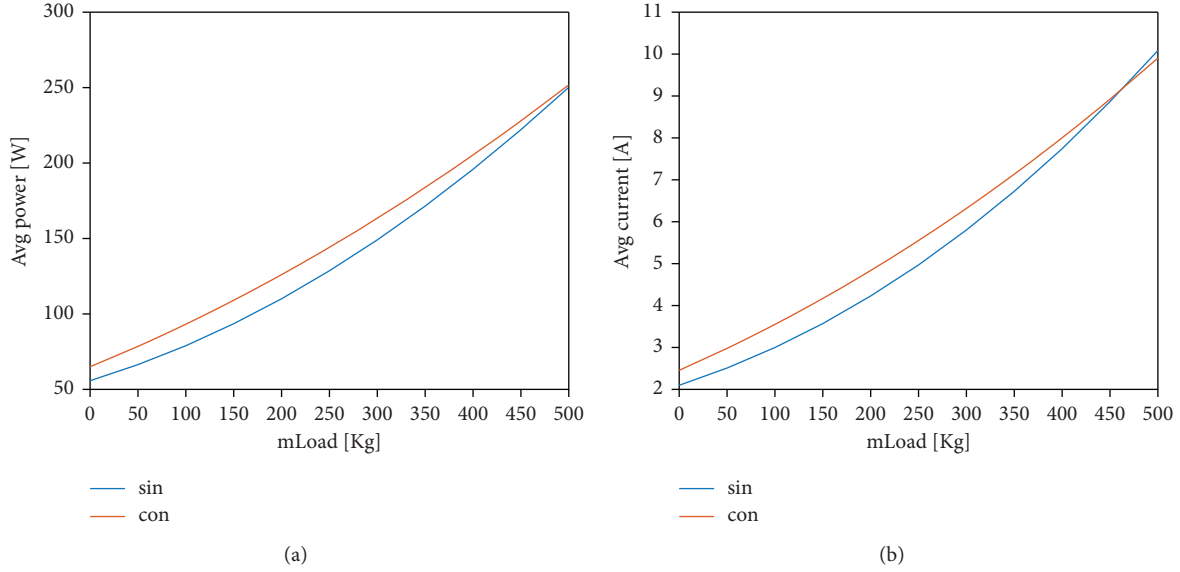
FIGURE 20: Variation of the average power (a) and average current (b) with load mass.
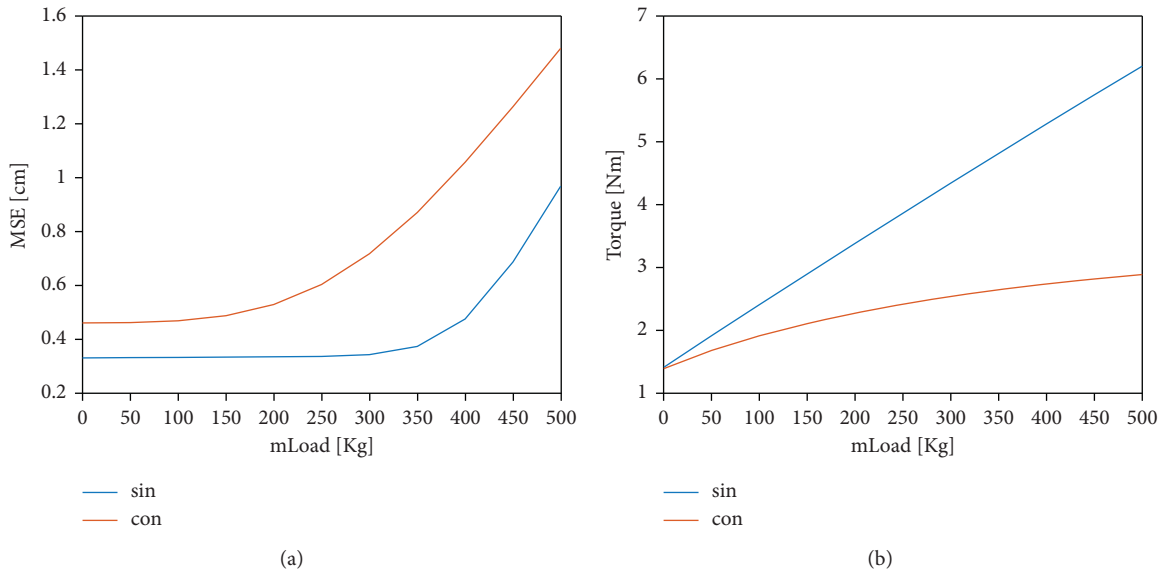

FIGURE 21: Variation of the MSE (a) and the standard deviation of the torque (b) with the load mass.

studied. The mean amplitude of the sinusoidal signal is half the value fixed for the constant one. The load mass is 500 kg. The rest of parameters of the experiment have been configured as in previous sections.

Figure 22 represents the variation of the average battery power with the velocity (Figure 22(a)) and the variation of the average current (Figure 22(b)). The color code is the same as in the previous experiments. Again, as expected, bigger velocity produces larger power and bigger current. However, in this case, the sensitivity, defined as the derivative of the measured variable respect to the parameter that is modified by the sweep, is smaller.

The sensitivity is denoted by $\beta_{\text{par}}^{\text{var}}$, where var is the measured variable and par is the parameter that is modified

in the experiment. In particular, $\beta_{m_L}^{\text{Pow}}$ is 0.4 W/kg, $\beta_{v_{\text{foll}}}^{\text{Pow}}$ is 0.18 W·s/cm, $\beta_{m_L}^{I_{\text{bat}}}$ is around 0.016 W/kg, and $\beta_{v_{\text{foll}}}^{I_{\text{bat}}}$ is 0.007 W·s/cm, approximately.

Figure 23 shows the variation of the MSE (Figure 23(a)) and of the standard deviation of the total torque in the wheels (Figure 23(b)). The color code is the same as in Figure 22. It is remarkable how the error is related to the cruise velocity; lower speeds produce smaller errors. This is so in general, but there exist an inferior limit related to the accuracy of the encoders that measure the velocity. The velocity is also related to the torque, as shown in Figure 23(b), where larger velocities demand bigger effective torque. This is not a general rule since the torque is directly related to the acceleration; in order to increase the
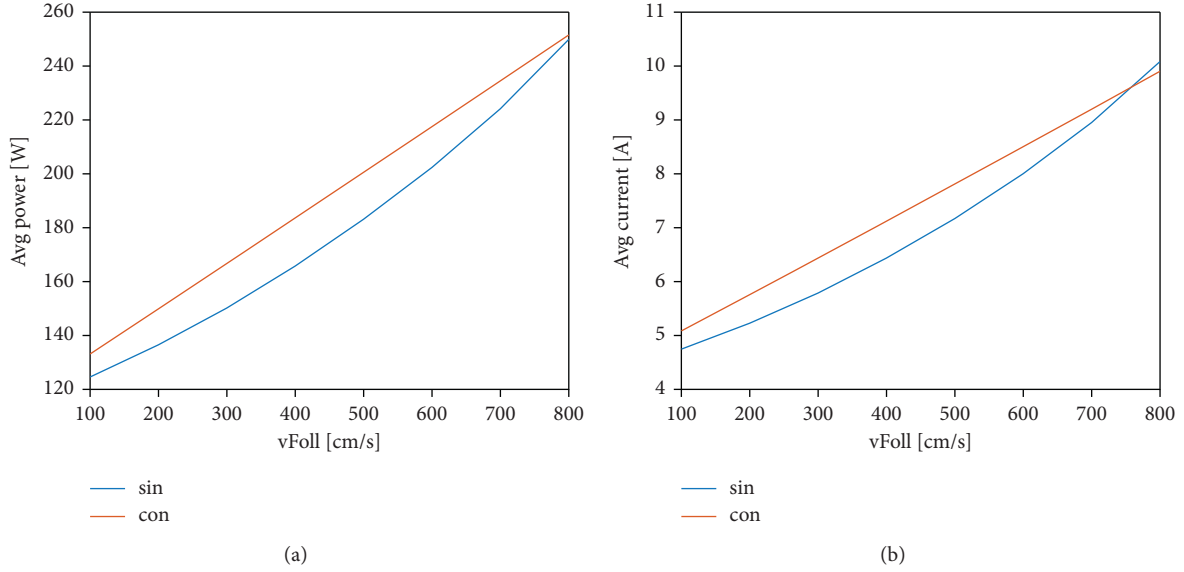
(a)

(b)

FIGURE 22: Variation of the average power (a) and average current (b) with the cruise velocity.
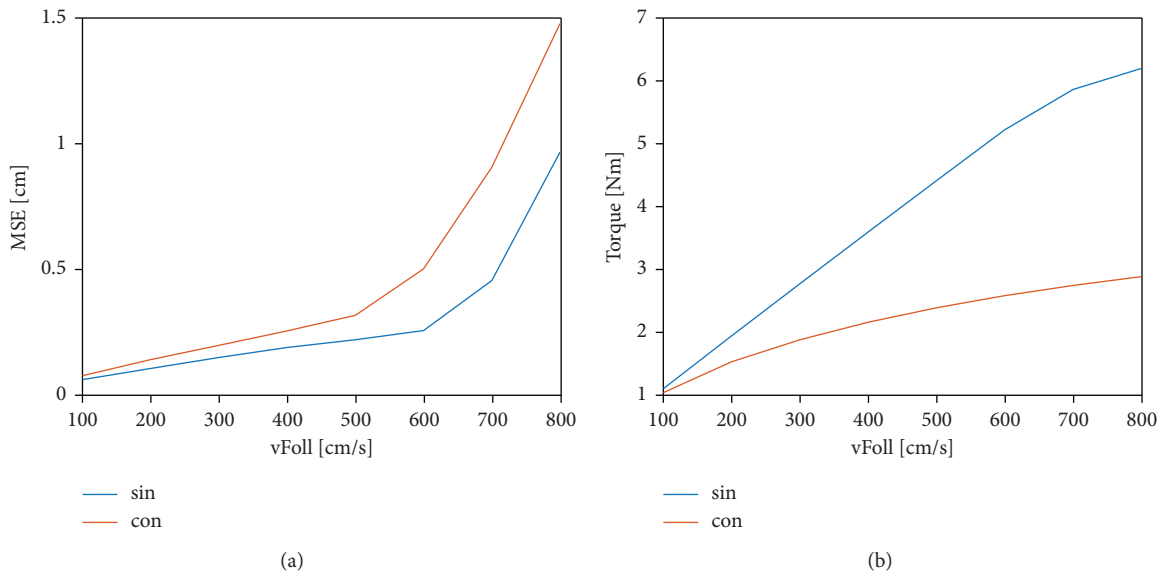

(a)

(b)

FIGURE 23: Variation of the MSE (a) and the standard deviation of the torque (b) with the cruise velocity.

acceleration of the AGV, it is necessary to increase the effective torque of the wheels. However, a bigger velocity does not necessarily entail a bigger torque since a higher velocity can be reached with the same acceleration and torque but spending more time. However, in this case, the AGV cannot leave the magnetic circuit; thus, bigger accelerations are necessary to maintain the cruise velocity and that increases the effective torque.

6. Conclusions and Future Works

Automatic guided vehicles are complex systems with highly coupled internal variables. Their use has lately grown exponentially due to the Industry 4.0 paradigm, as these are industrial devices and play a key role in process flow and in the logistic sector. However, some of their safety requirements depend on the dynamics and control of these vehicles. Therefore, a holistic modelling approach of the AGV is necessary in order to better understand its performance and to design efficient controllers.

In this paper, a complete model of a hybrid AGV has been developed. The vehicle is a general one, widely used, that is a mixture of tricycle and differential robots. The different subsystems of the AGV have been described and mathematically modelled, particularly, the control system, the safety system, the driving system, the guiding and location system, the charging, the power storage, and the motors. Even more, the interactions among them have been identified and represented in the proposed architecture of the AGV.

Simulations results, varying some of the vehicle parameters, have allowed it to validate the models as the global performance of the AGV is as expected.

Among other possible future works, we may highlight the study of the influence of the deformation of the wheels and the design of neural-controllers to improve the performance provided by the current PIDs. Besides, rollover conditions depending on the load could be estimated to ensure the safety of these systems.

Data Availability

The equations and parameters used in this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

Acknowledgments

References

[1] J. E. S. Garcia, F. Espinosa, and C. Santos, "Multi-AGV transport of a load: state of art and centralized proposal," *Revista Iberoamericana de Automática e Informática industrial*, vol. 2020, 2020.

[2] CoLLaboratE, 2020, https://collaborate-project.eu/.

[3] J. Gonzalez, X. Álvarez, A. M. González et al., "Fleet planning of AGVs," 2016.

[4] ASTI Mobile Robotics, 2019, https://asti.es/es/agvs#easybots.

[5] W. L. Maxwell and J. A. Muckstadt, "Design of automatic guided vehicle systems," *IIE Transactions*, vol. 14, no. 2, pp. 114–124, 1982.

[6] J. Veiga, J. Sousa, J. Machado, J. Mendonça, T. Machado, and P. Silva, "Modeling of dynamic behavior of AGV systems," 2019.

[7] M. Bonilla, F. Reyes, and M. Mendoza, "Modelling and simulation of a wheeled mobile robot in configuration classical tricycle," 2005.

[8] J. Villagra and D. Herrero-Pérez, "A comparison of control techniques for robust docking maneuvers of an AGV," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 4, pp. 1116–1123, 2011.

[9] Y. Han, Y. Cheng, and G. Xu, "Trajectory tracking control of AGV based on sliding mode control with the improved reaching law," *IEEE Access*, vol. 7, pp. 20748–20755, 2019.

[10] D. A. Aligia, G. A. Magallán, and C. H. De Angelo, "Control de Tracción para un Vehículo Eléctrico basado en Observadores no Lineales," *Revista Iberoamericana de Automática e Informática industrial*, vol. 15, no. 1, pp. 112–123, 2017.

[11] X. Wu, C. Sun, T. Zou, H. Xiao, L. Zhai, and J. Zhai, "Intelligent path recognition against image noises for vision guidance of automated guided vehicles in a complex workspace," *Applied Sciences*, vol. 9, no. 19, p. 4108, 2019.

[12] M. Smieszek, M. Dobrzanska, and P. Dobrzanski, "Measurement of wheel radius in an automated guided vehicle," *Applied Sciences*, vol. 10, no. 16, p. 5490, 2020.

[13] R. Stetter, M. Witczak, and M. Pazera, "Virtual diagnostic sensors design for an automated guided vehicle," *Applied Sciences*, vol. 8, no. 5, p. 702, 2018.

[14] T. Kawakami and S. Takata, "Battery life cycle management for automatic guided vehicle systems," *Design for Innovative Value towards a Sustainable Society*, vol. 8, pp. 403–408, 2012.

[15] L. Hou, L. Zhang, and J. Kim, "Energy modeling and power measurement for mobile robots," *Energies*, vol. 12, no. 1, p. 27, 2019.

[16] I. F. A. Vis, "Survey of research in the design and control of automated guided vehicle systems," *European Journal of Operational Research*, vol. 170, no. 3, pp. 677–709, 2006.

[17] A. Bayuwindra, J. Ploeg, E. Lefeber, and H. Nijmeijer, "Combined longitudinal and lateral control of car-like vehicle platooning with extended look-ahead," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 3, pp. 790–803, 2019.

[18] S. Baldi, D. Liu, V. Jain, and W. Yu, "Establishing platoons of bidirectional cooperative vehicles with engine limits and uncertain dynamics," *IEEE Transactions on Intelligent Transportation Systems*, vol. 2, 2020.

[19] P. Ghafoorpoor Yazdi, A. Azizi, and M. Hashemipour, "A hybrid methodology for validation of optimization solutions effects on manufacturing sustainability with time study and simulation approach for SMEs," *Sustainability*, vol. 11, no. 5, p. 1454, 2019.

[20] Oriolo, G. Control of Nonholonomic Systems, 2020, https://www.dis.uniroma1.it/~oriolo/cns/cns_slides.pdf.

[21] T. A. Henzinger, "The theory of hybrid automata," *Verification of Digital and Hybrid Systems*, vol. 23, pp. 265–292, 2000.

[22] S. Ito, M. Soga, S. Hiratsuka, H. Matsubara, and M. Ogawa, "Quality index of supervised data for convolutional neural network-based localization," *Applied Sciences*, vol. 9, no. 10, p. 1983, 2019.

[23] R. A. Jackey, "A simple, effective lead-acid battery modeling process for electrical system component selection," *SAE Transactions*, vol. 9, pp. 219–227, 2007.

[24] F. J. Gómez, L. J. Yebra, A. Giménez, and J. L. Torres-Moreno, "Modelado de baterías para aplicación en vehículos urbanos eléctricos ligeros," *Revista Iberoamericana de Automática e Informática industrial*, vol. 16, no. 4, pp. 459–466, 2019.

[25] F. Dušek, D. Honc, and P. Rozsíval, "Mathematical model of differentially steered mobile robot," 2011.

[26] E. Amaya-Cruz, O. Gutierrez-Frias, A. Luviano-Juarez, and H. Sossa-Azuela, "Design and construction of a robotic platform for 3D reconstruction through an embedded processing system," *IEEE Latin America Transactions*, vol. 16, no. 1, pp. 19–24, 2018.

[27] A. Beltrán, J. Rumbo, H. Azcaray, K. Santiago, M. Calixto, and E. Sarmiento, "Simulation and control of the velocity and electromagnetic torque of a three-phase induction motor: an electric vehicles approach," *Revista Iberoamericana de Automática e Informática industrial*, vol. 16, no. 3, pp. 308–320, 2019.

[28] F. Espinosa Zapata, J. L. Lázaro Galilea, and J. Olivares Bueno, "Proyecto ALCOR: contribuciones a la Optimización del Guiado Remoto de Robots en Espacios Inteligentes," *Revista Iberoamericana de Automática e Informática industrial*, vol. 15, no. 4, pp. 416–426, 2018.

[29] J. E. Sierra and M. Santos, "Modelling engineering systems using analytical and neural techniques: Hybridization," *Neurocomputing*, vol. 271, pp. 70–83, 2018.