Hindawi
Complexity
Volume 2024, Article ID 9514486, 23 pages
https://doi.org/10.1155/2024/9514486

Research Article

Trajectory Tracking Nonlinear Hybrid Control of Automated Guided Vehicles

Antonio Sánchez-Rodríguez,[1] Eduardo Bayona,[2] J. Enrique Sierra-García,[2] and Matilde Santos[3]

[1]Computer Science Faculty, Universidad Nacional de Educación a Distancia, Madrid 28040, Spain
[2]Department of Digitalization, University of Burgos, Burgos 09006, Spain
[3]Institute of Knowledge Technology, Complutense University of Madrid, Madrid 28040, Spain

Correspondence should be addressed to Eduardo Bayona; ebayona@ubu.es and Matilde Santos; msantos@ucm.es

Received 2 January 2024; Revised 28 March 2024; Accepted 16 April 2024; Published 6 May 2024

Academic Editor: Ning Cai

Automated guided vehicles (AGVs), so necessary in industrial environments, require precise control of trajectory tracking to make accurate stops at logistics stations, such as loading stations, or to pick up or drop off trolleys, pallets, or racks. This paper proposes a hybrid control architecture for trajectory tracking of a hybrid tricycle-differential AGV. The control strategy combines conventional proportional integral derivative (PID) control with advanced nonlinear Lyapunov control (LPC). The LPC is used for trajectory tracking while the PID is used for speed control of the robot. The stability of the controller is demonstrated for any differentiable trajectory. When a PID optimized with genetic algorithms is compared with the proposed controller for several trajectories, the LPC outperforms it in all cases.

1. Introduction

Automated guided vehicles (AGVs) are used in industry to replace conveyors and manned industrial trucks. They provide more flexibility than other types of transportation for logistics and production applications. They are safer and more efficient than manned industrial vehicles, and they improve the quality of the processes where they are implemented [1]. The digitalization of processes and the industry 4.0 paradigm have driven its deployment in the industrial sector [2].

For these AGVs to develop their full potential, it is necessary to incorporate control systems that allow precise tracking of trajectories. On the one hand, they should not divert the defined routes so as not to cause safety problems, process failures and production stops [3]. On the other hand, they need to be very precise in following the trajectories to make stops at stations, for example at charging stations (so that there are no problems with the brushes or charging pads) or when they are going to pick up or drop off a trolley, a pallet or a rack [4].

Today, most of these vehicles use conventional PID controllers [5], although some research show that intelligent controllers are more and more used, being a fact that such implementations in industry do not occur at the same speed as in other fields [6]. The conventional controllers have proven to be very effective and versatile in different applications. However, it is well known that they have certain limitations for the control of nonlinear systems. In the case of AGVs, the trajectories described in logistics applications can be nonlinear, and even more, sometimes they must cope with non-linear effects. So, it is interesting to explore other advanced techniques for the design of controllers, such as intelligent algorithms or those based on Lyapunov theory.

In this work, a nonlinear controller based on Lyapunov (LPC) is proposed for AGV trajectory tracking. The controller is tested in simulation with a hybrid AGV that combines the tricycle and differential kinematic models. Those models represent a real AGV, the Mouse AGV developed by Asti Mobile Robotics. This AGV is quite common in the industry due to its versatility and performance although it is not usually considered in the literature because its model is quite more complex than a differential robot [7]. The new controller is compared with a trajectory tracking PID regulator optimized with genetic algorithms for different trajectories, and in all of them the LPC obtains better results than the conventional PID in terms of precision in following the trajectories. In both control approaches the wheels speed control is performed with the same conventional PID controllers.

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

(i) A hybrid control architecture for mobile robots has been proposed. This control strategy combines nonlinear Lyapunov control with conventional PID controllers.

(ii) Lyapunov's control law is based on the kinematic model of the vehicle and can be adapted for different types of mobile robots.

(iii) The stability of the proposed controller has been demonstrated. Following Lyapunov theory, it is proven that the controller is stable as long as the reference trajectory is differentiable.

(iv) The controller has been evaluated with 3 different trajectories: circle, ellipse, and lemniscate. These trajectories are common in the industrial applications of these vehicles [3, 8].

The rest of the paper is organized as follows. In Section 2 some related works are commented. The model of the AGV is described in Section 3. Section 4 presents the hybrid control architecture and proves its stability according to Lyapunov's theory. Simulation results are discussed in Sections 5 and 6. The last section is dedicated to the conclusions and future works.

2. Related Works

There are different studies focused on the control of an AGV in trajectory tracking and trajectory planning. The recent review by [9] presents different methods such as PID, sliding control, fuzzy logic combining with neural networks, model predictive control, backstepping techniques, among other, to compensate for lumped disturbances on the path tracking during the movement of an AGV. A systematic literature review about the position control of AGVs is presented in [10]. In this case the principal focus is on the control strategies used in the AGV position control problem, together with the mathematical model considered, the sensors and guidance system used, and the maximum payload of the vehicle and operation under different load conditions.

Studies on AGV trajectory tracking usually concentrate on controlling the AGV to minimize the guidance error for a defined trajectory. It is difficult to deal with the effects of nonlinearities present in the trajectories during AGV tracking of the reference path because they are implicit in the trajectories and affect the control of the guidance error. To address this problem, different techniques have been applied, some coming from the fields of artificial intelligence or machine learning, or from the field of nonlinear control such as sliding mode control or Lyapunov based control.

To mention some examples, in [11], a tracking controller for automatic guided vehicles (AGV) based on a kinematic model is proposed to follow a desired trajectory using Lyapunov method to provide robustness against load disturbances and sensor noise. Zhang et al. proposed a Lyapunov model-based predictive control method for the AGV path following problem [12]. The predictive path following the control method is restricted by the Lyapunov function, which guarantees the stability of the AGV motion control. The controller is constrained by the Lyapunov stability criteria to prove the local and global convergence of the control system. In [13], the authors propose the design of an adaptive controller of a mobile robot for trajectory tracking, performing the stability study by applying Lyapunov's theorem. In [14], the authors perform an analysis of the trajectory tracking control using a Lyapunov function that guarantees the global stability of the system based on the kinematic model. For this purpose, they use as a nonlinear control technique the backstepping, designing a trajectory tracking controller of the AGV with nonholonomic constraints; this study is performed for different trajectories. Another interesting article is the one by [15], where based on a kinematic model of an AGV wheeled parking robot, a sliding mode controller is designed with a Lyapunov direct method and an improved fast stationary power approach law is then applied. Simulation of linear and circular trajectories are used to demonstrate the effectiveness of this approach.

In the paper by [16], the dynamics and kinematics models of a four-wheel steering AGV were established, and the Lyapunov direct method was used to construct a trajectory tracking controller with global asymptotic stability. Besides, an adaptive particle swarm optimization (PSO) algorithm was applied to optimize the control parameters of the controller. The proposed control approach was compared with an adaptive model predictive control under simulated working condition of moving centroid giving better results regarding trajectory tracking accuracy.

Others research works use the sliding control approach as an effective method for nonlinear systems, with fast response and insensitivity to parameter changes and external disturbances. Jiang et al. address the AGV trajectory following control problem with nonholonomic constraints by proposing a sliding mode control approach [17]. This work analyses the AGV motion characteristics and designs a backstepping sliding mode control with a ranging law. This range law obtains the convergence rate of tracking errors. In addition, with the sliding mode controller, it aims to strictly guarantee the asymptotic stability of the tracking deviations. It is applied to an AGV which structure consists of two actuated wheels and two passive wheels, and the front wheels mainly support the body of the mobile AGV, whereas the rear wheels drive the vehicle. In [18], the authors design a robust controller for trajectory tracking of a two-wheeled mobile robot by applying a second-order sliding mode control methodology to ensure the continuity of torque input and robustness against disturbances. The designed sliding mode control smooths out the commutation term. In [19], the authors define the kinematic and dynamic models of the autonomous guided vehicle and propose a design of a closed-loop control to realize trajectory tracking. They design a sliding mode control based on an improved control law to control the guidance error of the AGV. Taghavifar and Mohammadzadeh [20], using a two-degrees-of-freedom bicycle model, propose an integral backstepping control that is hybridized with a terminal sliding mode control method to enhance the lateral path-tracking performance of the multilevel speed AGV. The Lyapunov stability theorem is employed to guarantee the global asymptotic stability of the closed-loop system and the convergence of tracking errors.

Model predictive control is another approach that has been used for autonomous guided vehicles. The paper by [21] defines a control scheme that consists of three parts: a referential path, a model predictive controller (MPC) controller, and a kinematic model of a two-wheel differential driving AGV. Simulation control based on MPC is proposed and the simulation control experiment is carried out in the robotic operative system (ROS). The MPC controller adopts a quadratic programming method to obtain the optimal linear and angular velocities as the output to control the AGV. The work by Liu et al. [22] proposes a real time nonlinear MPC control method that considers both tracking accuracy and lateral load transfer limitation for heavy duty AGVs. In this case, the AGV with a four-wheel-steering four-wheel-driving structure is simplified into the well-known bicycle mode. In [23], the authors develop an adaptive learning MPC (ALMPC) scheme for the AGV trajectory tracking problem with input constraints. This article considers a general AGV with two differential driving wheels subject to both the parametric uncertainty and external disturbances. A lemniscate trajectory is used to show the robustness of the method.

Some intelligent techniques have also been used to enhance other controllers or to implement some control elements. The paper by [24] designs a fuzzy logic controller for a 16-wheel heavy-duty AGV that is simplified as a two-wheel AGV model. Simulation of circular and straight trajectories prove a good tracking by the AGV. In [4], an intelligent hybrid control scheme that combines reinforcement learning-based control (RLC) with conventional PI regulators is proposed. The PI regulators are used to control the speed of each wheel while the input reference of these regulators is calculated by the RLC in order to reduce the path tracking guiding error and to maintain the longitudinal speed. In addition, the PID regulators have been tuned by genetic algorithms. Another example is found in [25], where four tuning methods: Ziegler Nichols, empirical, particle swarm optimization (PSO), and beetle antennae searching (BAS) are applied to adjust the parameters of the PID that controls the AGV platform. Results are compared in four paths including the circle, ellipse, spiral and 8-shaped paths. The paper by Binh et al. [26] proposes a solution for an AGV robot to effectively monitor a moving object using deep learning and follow it. It is based on a model of a four-wheeled self-propelled robot vehicle that is built and that can adaptively track the speed and direction of a human's movements. Similarly, in [27] an active tracking system and a control algorithm are implemented for a person-following real industrial AGV. The algorithm uses a LIDAR sensor to detect and track the target.

This brief review of some related works shows that most authors agree that AGVs must have capabilities provided by nonlinear control techniques to address the effects of nonlinearities present in trajectories or other intrinsic effects of the vehicle, such as wear or friction. Although there are some studies in the scientific literature on nonlinear control methods applied to trajectory tracking of AGVs, the main differences with the work here presented is that in this paper the stability is analyzed, it deals with a general model of a hybrid tricycle differential industrial vehicle, not just a differential robot as most of them, and that the control has been tested including some non-linearities such as the friction in the wheels and in the drive unit.

3. Modelling of the AGV

In this section the model equations that describe the physical behaviour of a hybrid differential-tricycle AGV are described. Briefly, the AGV is composed of a traction unit that functions as a differential mobile robot, connected to the body of the AGV through an axis on which it pivots. The AGV body kinematics follows the movement of a tricycle-type vehicle, but steering control is performed by the drive unit itself rather than by the front wheel of the tricycle. The mathematical model follows the equations described in detailed in [1]. Figure 1 shows a schematic of the AGV in the (X, Y) plane.

The dynamic equations are based on the Lagrange-Euler approach (1) and (2) [1]. The effective torque on the right and left wheels of the traction unit (1) and the system equations that describe the angular accelerations are given by

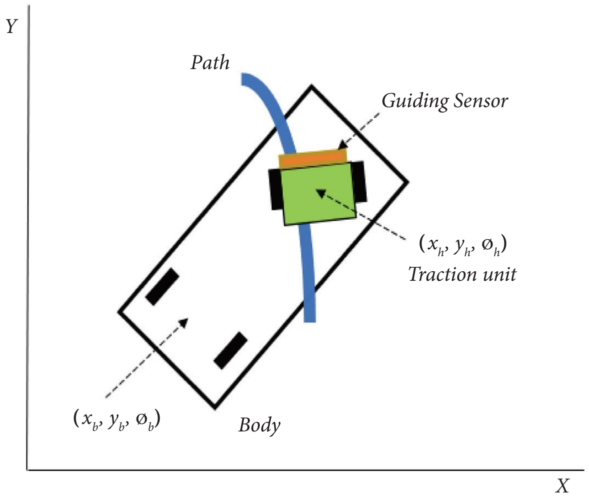


Figure 1: Traction unit and AGV body [1].

$$\begin{aligned} M_{\mathrm{eR}} &= M_R - F_{\mathrm{SW}_R} \cdot \operatorname{sign} \cdot \left(\dot{\theta}_R\right), \\ M_{\mathrm{eL}} &= M_L - F_{\mathrm{SW}_L} \cdot \operatorname{sign} \cdot \left(\dot{\theta}_L\right), \end{aligned} \tag{1}$$

$$\begin{bmatrix} m_T \cdot \dfrac{R_h}{2} & m_T \cdot \dfrac{R_h}{2} \\ \dfrac{I_h R_h}{L_h} & -\dfrac{I_h R_h}{L_h} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_R \\ \ddot{\theta}_L \end{bmatrix} = \begin{bmatrix} \dfrac{\left(M_{\mathrm{eR}} + M_{\mathrm{eL}}\right)}{2R_h} - f_{\mathrm{rT}} \\ \dfrac{\left(M_{\mathrm{eR}} - M_{\mathrm{eL}}\right) L_h}{2R_h} - f_{\mathrm{rR}} \end{bmatrix}, \tag{2}$$

where the translational and rotational frictions are described by (3) and (4), respectively.

$$f_{rT} = 0.5 \cdot \delta_{\mathrm{air}} \cdot S_{\mathrm{AGV}} \cdot C_{\mathrm{aero}} \cdot \left(V_h^2\right) \cdot \operatorname{sign}\left(V_h\right) + 9.8 \cdot m_T \cdot C_{\mathrm{roll}} \cdot \operatorname{sign}\left(V_h\right), \tag{3}$$

$$f_{\mathrm{rR}} = F_{\mathrm{vh}} \cdot \dot{\varnothing}_h + F_{\mathrm{sh}} \cdot \operatorname{sign}\left(\dot{\varnothing}_h\right). \tag{4}$$

On the other hand, the kinematic equations are obtained as the combination of a differential and a tricycle vehicle. Equations (5)–(8) allows to obtain the position and orientation of the vehicle.

$$\begin{aligned} V_L &= R_h \cdot \dot{\theta}_L, \\ V_R &= R_h \cdot \dot{\theta}_R, \end{aligned} \tag{5}$$

$$\begin{aligned} \dot{X}_h &= \frac{V_L + V_R}{2} \cos\left(\varnothing_h\right), \\ \dot{Y}_h &= \frac{V_L + V_R}{2} \sin\left(\varnothing_h\right), \\ \dot{\varnothing}_h &= \frac{V_R - V_L}{L_h}, \end{aligned} \tag{6}$$

$$\begin{aligned} \dot{X}_b &= V_h \cos(\gamma) \cos\left(\varnothing_b\right), \\ \dot{Y}_b &= V_h \cos(\gamma) \sin\left(\varnothing_b\right), \\ \dot{\varnothing}_b &= \frac{V_h}{L_b} \sin(\gamma), \end{aligned} \tag{7}$$

$$V_h = \sqrt{\dot{X}_h^2 + \dot{Y}_h^2} = \frac{V_L + V_R}{2}. \tag{8}$$

The vehicle is subjected to kinematic constraints to avoid the winding of the cables and the slippage (91011):

$$\gamma = \varnothing_h - \varnothing_b, \gamma_{\min} \le \gamma \le \gamma_{\max}, \tag{9}$$

$$\dot{X}_b \sin\left(\varnothing_b\right) - \dot{Y}_b \cos\left(\varnothing_b\right) = 0, \tag{10}$$

$$\dot{X}_b \sin\left(\varnothing_b + \gamma\right) - \dot{Y}_b \cos\left(\varnothing_b + \gamma\right) - \dot{\varnothing}_b L_b \cos(\gamma) = 0, \tag{11}$$

where the main variables of the previous equations with the corresponding units are the following:

(i) $(x_h, y_h, \varnothing_h)$: position (m) and orientation (rad) of the AGV drive unit.

(ii) $(x_b, y_b, \varnothing_b)$: position (m) and orientation (rad) of the AGV's body.

(iii) γ: steering angle (rad). That is the orientation of the traction unit respect the body.

(iv) V_h: longitudinal speed of the traction unit (m/s).

(v) $\ddot{\theta}_r$: angular acceleration of the right wheel of the drive unit (rad/s^2).

(vi) $\ddot{\theta}_l$: angular acceleration of the left wheel of the drive unit (rad/s^2).

(vii) M_{eR}: effective torque on the right wheel of the traction unit (Nm).

(viii) M_{eL}: effective torque on the left wheel of the traction unit (Nm).

(ix) f_{rT}: translational frictional force (N).

(x) f_{rR}: rotational frictional force (N).

(xi) M_R: motor torque in the right wheel (Nm).

(xii) M_L: motor torque in the left wheel (Nm).

4. Hybrid Control Architecture

As seen in equation (2), which models the dynamic of the system, the relationship between the torque $(M_{\mathrm{eR}}, M_{\mathrm{eL}})$ and the acceleration $(\ddot{\theta}_R, \ddot{\theta}_L)$, and thus with the speed $(\dot{\theta}_R, \dot{\theta}_L)$, is mainly linear. There are only small nonlinearities related to the friction. Therefore, the control of the wheel speed by adjusting the torque can be efficiently solved by PID regulators, whose ability to control linear systems has been widely demonstrated.

On the other hand, in equations (6) and (7) it is possible to observe that the sinusoidal functions introduce nonlinear behavior in the kinematic model of the vehicle. In addition, the trajectories are other sources of nonlinearities. These nonlinearities appear in the trajectory tracking control, that in this case is implemented so the references for the angular and the longitudinal speed (w, u) are generated to minimize the tracking error. Therefore, the trajectory tracking control is solved by the LPC strategy which has proven efficient in nonlinear control.

Therefore, the idea of this proposed control architecture consists of combining the following controllers:

(1) Two conventional linear PI controllers, to control the speed of each wheel, right and left, of the traction unit.

(2) A controller based on the Lyapunov stability method to deal with trajectory tracking control.

Figure 2 shows the architecture of this hybrid control system where the LPC and the PI controllers can be identified.

The variables involved in the speed control (PID) are:

(i) M_R, motor torque of the right wheel (output of the right PI speed controller)

(ii) M_L, motor torque of the left wheel (output of the left PI speed controller)

(iii) v_R, longitudinal speed of the right wheel (output of the AGV and input of the right PI speed controller)

(iv) v_L, longitudinal speed of the left wheel (output of the AGV and input of the left PI speed controller)

(v) $v_{R_{\mathrm{REF}}}$, reference speed of the right wheel (input of right PI speed controller), it comes from the LPC passing first through the inverse kinematics of the AGV model.

(vi) $v_{L_{\mathrm{REF}}}$, reference speed of the left wheel (input of right PI left controller), it comes from the LPC passing first through the inverse kinematics of the AGV model.

The variables involved in the trajectory control (LPC) are:

(i) u, longitudinal speed reference generated by the LPC.

(ii) w, angular speed reference generated by the LPC.

(iii) $(X_e, Y_e, \varnothing_e)$, position error (m) and orientation error (rad) with respect to the trajectory to be tracked. The LPC tries to make these values zero.

Finally, the variables that play a role in the trajectory planner are:

(i) v_{refd}, reference longitudinal velocity, specified by the user; it is the one considered to sample the points in the trajectory.

(ii) $(X_d, Y_d, \varnothing_d)$, reference position (m) and reference orientation (rad) generated by the trajectory planner. These desired position and orientation are used to calculate the error $(X_e, Y_e, \varnothing_e)$, that is the difference between these desired values and the current position and orientation $(X_r, Y_r, \varnothing_r)$.

The functional description of this control strategy is as follows. The PI controllers of the left and right wheels have the function of adjusting the input torque of the motors, M_R, and M_L, respectively, so the AGV can follow the speed reference of each wheel $(v_{R_{\mathrm{REF}}}, v_{L_{\mathrm{REF}}})$. These speed references of each wheel are obtained by considering the inverse kinematics of the drive unit of the AGV.

The inverse kinematics module receives both, the reference angular velocity w and the reference longitudinal velocity u obtained by the LPC controller, transforming them into reference velocities for each wheel $(v_{R_{\mathrm{REF}}}, v_{L_{\mathrm{REF}}})$.

The LPC controller receives as input the position and orientation error, the difference between the position and orientation of the trajectory generated by the trajectory planner and the output position and orientation of the AGV

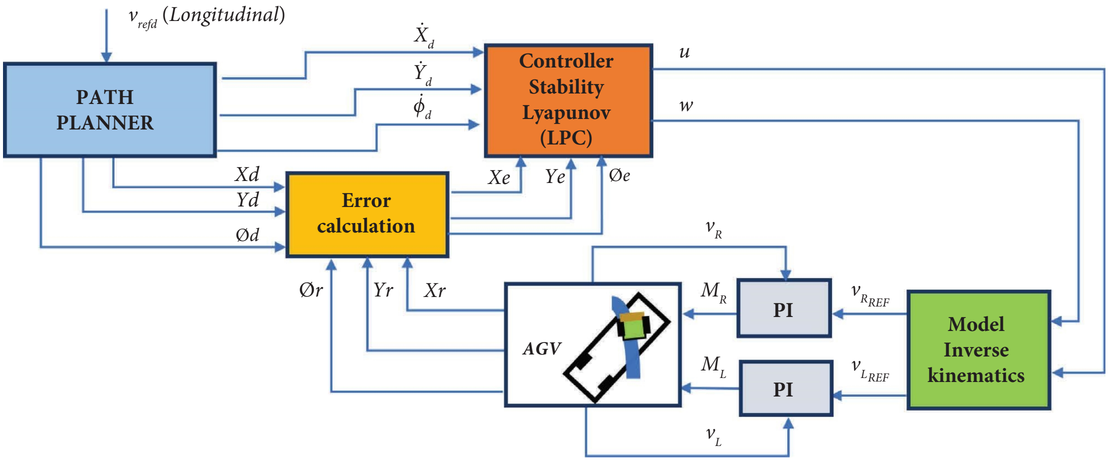


Figure 2: Hybrid control architecture for AGV trajectory tracking with Lyapunov-based controller.

model. The reference longitudinal velocity v_{refd} (of the system as a whole) configured in the trajectory planner module is used to sample the trajectory to generate the position and orientation that has to be tracked. In this way, the reference longitudinal velocity is included in the control strategy.

With this control strategy, both the reference longitudinal velocity u and the reference angular velocity w are generated by the LPC control ensuring the stability, so that the AGV can follow the path predefined by the path planner correctly and, at the same time, solve both the v_{refd} velocity profile tracking problem and the tracking control to minimize the error in the path, making both adjustments simultaneously and in real time.

It is important to remark that the speed profile tracking and trajectory tracking settings must be done simultaneously because they are interrelated and coupled in the AGV, and with this architecture the intention is to achieve that goal.

It is also noteworthy to note that v_{refd} and u are different signals. The input of the path planner is v_{refd}, the reference longitudinal velocity. This speed is specified by the user as it is intended that the robot follows a determined speed profile. This speed is used by the path planner to sample the desired points of the trajectory and this way to obtain the points $(X_d, Y_d, \varnothing_d)$. On the other hand, the outputs of the LPC, u and w, are generated to reduce the tracking error to zero. If the AGV perfectly follows the trajectory, the output u of the LPC would perfectly match v_{refd}. However, there is always a small tracking error. In other words, the output u of the LPC asymptotically tends to follow v_{refd}.

This control architecture can be formalized by equations.

$$\text{he}(t_i) = \begin{bmatrix} X_e(t_i) \\ Y_e(t_i) \\ \varnothing_e(t_i) \end{bmatrix} = \begin{bmatrix} X_d(t_i) - X_r(t_{i-1}) \\ Y_d(t_i) - Y_r(t_{i-1}) \\ \varnothing_d(t_i) - \varnothing_r(t_{i-1}) \end{bmatrix}, \dot{h}d(t_i) = \begin{bmatrix} \dot{X}_d(t_i) \\ \dot{Y}_d(t_i) \\ \dot{\varnothing}_d(t_i) \end{bmatrix}, \tag{12}$$

$$[w(t_i), u(t_i)] = f_{\text{LPC}}\left(\text{he}(t_i), \dot{h}d(t_i)\right), \tag{13}$$

$$\begin{aligned} v_{L_{\text{REF}}}(t_i) &= u(t_i) - \frac{w(t_i)\cdot L_h}{2}, \\ v_{R_{\text{REF}}}(t_i) &= \mathrm{u}(t_i) + \frac{w(t_i)\cdot L_h}{2}, \end{aligned} \tag{14}$$

$$\begin{aligned} \text{err}_{v_L}(t_i) &= v_{L_{\text{REF}}}(t_i) - v_L(t_{i-1}), \\ \text{err}_{v_R}(t_i) &= v_{R_{\text{REF}}}(t_i) - v_R(t_{i-1}), \end{aligned} \tag{15}$$

$$M_L(t_i) = K_{Vp} * \text{err}_{\text{vL}}(t_i) + K_{\text{VI}} \int \text{err}_{\text{vL}}(t_i)dt, \tag{16}$$

$$M_R(t_i) = K_{\text{Vp}} * \text{err}_{\text{vR}}(t_i) + K_{\text{VI}} \int \text{err}_{\text{vR}}(t_i)dt. \tag{17}$$

The inverse kinematics model is calculated by equation (14). These expressions are obtained from the kinematic model of the differential vehicle (5). Substituting the current speed by its reference in the equations of the angular and longitudinal speed, the equation of the inverse kinematics (14) is derived and then it is possible to obtain equation (18). This explains how reference angular velocity and the reference longitudinal velocity are obtained from the inverse kinematics model.

$$\begin{aligned} u(t_i) &= \frac{\mathrm{v}_{\mathrm{L}_{\mathrm{REF}}}(t_i) + \mathrm{v}_{\mathrm{R}_{\mathrm{REF}}}(t_i)}{2}, \\ w(t_i) &= \frac{\mathrm{v}_{\mathrm{R}_{\mathrm{REF}}}(t_i) - \mathrm{v}_{\mathrm{L}_{\mathrm{REF}}}(t_i)}{L_h}. \end{aligned} \tag{18}$$

The tuning constants of the PI speed controllers were adjusted by genetic algorithms (GA). The equation (19) was used as fitness function. The configuration of the GA was as follows. The size of the population was set to 50 individuals, the crossover fraction was set to 0.8, the mutation rate was 0.2, and the elite size was 5%. Individuals were randomly initialized.

$$f_{\mathrm{cv}} = \frac{1}{T_{\mathrm{sim}}} \sum_i \left| u(t_i) - V_h(t_i) \right|. \tag{19}$$

4.1. Mathematical Description of the LPC. The control objective is formulated directly in the trajectory space. The goal is that the vehicle must follow the desired motion in the plane of the trajectory to be followed. This motion is specified by the trajectory $[xd(t), yd(t)]^T$, which is assumed to be differentiable. This objective can be formalized by

$$\lim_{t \longrightarrow \infty} \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} xd(t) \\ yd(t) \end{bmatrix}. \tag{20}$$

In this industrial robot, the kinematic model is defined as a function of longitudinal velocity u and angular velocity w, with the point located at the center of the axle of traction unit $(x_h, y_h, \varnothing_h)$, as represented in Figure 3. It can be expressed by.

$$\begin{bmatrix} \dot{x}_h \\ \dot{y}_h \\ \dot{\varnothing}_h \end{bmatrix} = \begin{bmatrix} \cos\varnothing_h & 0 \\ \sin\varnothing_h & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ w \end{bmatrix}. \tag{21}$$

The matrix that describes the kinematic model of equation (21) is not invertible. However, this is a requirement to apply the state feedback technique to design the control low. Thus, to make it invertible, the orientation of the differential unit is not considered. In this way, one row of the matrix of the kinematic model is removed, obtaining a square and thus invertible matrix.

A point $(x_{\mathrm{hp}}, y_{\mathrm{hp}})$ located in the perpendicular axis to the axle of the wheels at a distance called "a" is defined (see Figure 3). This point of the traction unit will follow the path. The Jacobian matrix of the kinematic model at the point displaced from the wheel axis a distance "a" will be used. For this purpose, it is considered that there is no lateral displacement.

This point $(x_{\mathrm{hp}}, y_{\mathrm{hp}})$ is given by

$$\begin{bmatrix} x_{\mathrm{hp}} \\ y_{\mathrm{hp}} \end{bmatrix} = \begin{bmatrix} x_h + a * \cos\varnothing_h \\ y_h + a * \sin\varnothing_h \end{bmatrix}. \tag{22}$$

The derivative of this point is.

$$\dot{x}_{\mathrm{hp}} = \dot{x}_h - a * \dot{\varnothing}_h * \sin\varnothing_h = u * \cos\varnothing_h - a * w * \sin\varnothing_h, \tag{23}$$

$$\dot{y}_{\mathrm{hp}} = \dot{y}_h + a * \dot{\varnothing}_h * \cos\varnothing_h = u * \sin\varnothing_h + a * w * \cos\varnothing_h. \tag{24}$$

These equations can be expressed in matrix form

$$\begin{bmatrix} \dot{x}_{\mathrm{hp}} \\ \dot{y}_{\mathrm{hp}} \end{bmatrix} = \begin{bmatrix} \cos\varnothing_h & -a * \sin\varnothing_h \\ \sin\varnothing_h & a * \cos\varnothing_h \end{bmatrix} \begin{bmatrix} u \\ w \end{bmatrix} = J \begin{bmatrix} u \\ w \end{bmatrix}. \tag{25}$$

As said, the matrix J must be invertible; therefore, its determinant must be different from zero. To ensure this, the distance a must be also different from zero. In this work the value $a = 0.0\ 8\,\mathrm{cm}$ has been set, without losing generality.

By applying the state feedback, the control law is defined as follows:

$$u_{\mathrm{Ref}} = \begin{bmatrix} u \\ w \end{bmatrix} = J^{-1} * \left(\begin{bmatrix} \dot{x}_d \\ \dot{y}_d \end{bmatrix} + K \begin{bmatrix} x_d - x_{\mathrm{hp}} \\ y_d - y_{\mathrm{hp}} \end{bmatrix} \right) = J^{-1} * (\dot{h}d + K \cdot \mathrm{he}), \tag{26}$$

where

(i) $\mathrm{hd} = \begin{bmatrix} x_d \\ y_d \end{bmatrix}$ is the vector of the desired or reference position.

(ii) $h = \begin{bmatrix} x_{\mathrm{hp}} \\ y_{\mathrm{hp}} \end{bmatrix}$ is the position vector of the vehicle.

(iii) $\mathrm{he} = [\mathrm{hd} - h] = \begin{bmatrix} x_d - x_{\mathrm{hp}} \\ y_d - y_{\mathrm{hp}} \end{bmatrix}$ is the position error vector.

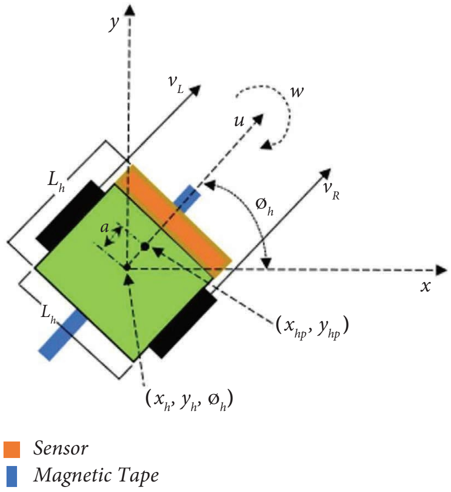


Figure 3: Traction unit of the AGV and displaced point.

(iv) $\dot{h} = \begin{bmatrix} \dot{x}_{\text{hp}} \\ \dot{y}_{\text{hp}} \end{bmatrix}$ is the velocity vector of the mobile robot.

(v) $\dot{h}d = \begin{bmatrix} \dot{x}_d \\ \dot{y}_d \end{bmatrix}$ is the velocity vector of the reference.

(vi) K is a positive defined matrix which determines the gain of the controller.

(vii) $u_{\text{Ref}} = \begin{bmatrix} u \\ w \end{bmatrix}$ is the reference vector of the control law. The components are the longitudinal reference speed u and the rotational reference speed w that are the outputs of the LPC.

Those equations illustrate the design of J for a differential robot unit. But this control law can be applied to different types of robots just updating matrix J. For instance, if the traction unit were a tricycle, equations (23) and (24) must be updated and therefore, so must the J matrix.

In this case the point $(x_{\text{hp}}, y_{\text{hp}})$ is given by equations as follows:

$$\dot{x}_{\text{hp}} = \dot{x}_h - a * \dot{\varnothing}_h * \sin\varnothing_h = u\cos(\gamma) * \cos\varnothing_h - a * \frac{u}{L_b}\sin(\gamma) * \sin\varnothing_{h,} \tag{27}$$

$$\dot{y}_{\text{hp}} = \dot{y}_h + a * \dot{\varnothing}_h * \cos\varnothing_h = u\cos(\gamma) * \sin\varnothing_h + a * \frac{u}{L_b}\sin(\gamma) * \cos\varnothing_{h,} \tag{28}$$

where γ is the steering angle of the steering wheel. Now let us define the auxiliar control signals: $u_1 = u\cos(\gamma)$ and $u_2 = u\sin(\gamma)$. Then equations (27) and (28) can be also expressed in a matrix form (29), which is equivalent to (25).

$$\begin{bmatrix} \dot{x}_{\text{hp}} \\ \dot{y}_{\text{hp}} \end{bmatrix} = \begin{bmatrix} \cos\varnothing_h & -\frac{a}{L_b} * \sin\varnothing_h \\ \sin\varnothing_h & \frac{a}{L_b} * \cos\varnothing_h \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = J \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}. \tag{29}$$

Thus, the same control law (26) used for the differential traction unit is also applicable to the tricycle case, but it is necessary to consider that the J matrix has to be updated. It is noteworthy to remark that in this case the control law generates the signals u_1 and u_2.

$$u_{\text{Ref}} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = J^{-1} * \left(\begin{bmatrix} \dot{x}_d \\ \dot{y}_d \end{bmatrix} + K \begin{bmatrix} x_d - x_{\text{hp}} \\ y_d - y_{\text{hp}} \end{bmatrix} \right) = J^{-1} * (\dot{h}d + K \cdot \text{he}). \tag{30}$$

From signals u_1 and u_2, the control signals for the tricycle u and γ are obtained by

$$u = \sqrt{\frac{\left(u_1^2 + u_2^2\right)}{2}}, \tag{31}$$

$$\gamma = a \tan\left(\frac{u_2}{u_1}\right). \tag{32}$$

This process can be applied for other robots with different kinematic configurations to obtain the equivalent J matrix and control law.

4.2. Study of the Control Law Stability. To carry out the stability analysis, the AGV is assumed to perfectly follow the reference, so $he = [0\,0]^T$. This way it is possible to express the relationship between the velocity vector $\dot{h}$ of the AGV and the reference vector u_{Ref} that is the output from the Lyapunov controller by

$$\dot{h} = J * u_{\text{Ref}}. \tag{33}$$

This expression can be considered correct if a perfect tracking of the AGV velocity $\dot{h}$ is assumed, that is, without errors. This way the control law cancels the J matrix.

A Lyapunov candidate function given by (34) is proposed to demonstrate the stability of the control law. This function is a positive definite function $V \geq 0$.

$$V = \frac{1}{2}[h_e]^T [h_e], \tag{34}$$

The derivative of this function is

$$\dot{V} = \frac{1}{2}[\dot{h}_e]^T [h_e] + \frac{1}{2}[h_e]^T [\dot{h}_e] = [h_e]^T [\dot{h}_d - \dot{h}] = [h_e]^T [\dot{h}_d - J * u_{\text{Ref}}]. \tag{35}$$

Substituting (26) in (35), equation (36) is obtained.

$$\dot{V} = [h_e]^T \left[\dot{h}_d - J * J^{-1} (\dot{h}d + K * \text{he})\right] = -[h_e]^T [K * \text{he}]. \tag{36}$$

As K is positive defined, $\dot{V}$ is negative defined, and this implies that the system with this control law is locally asymptotically stable (LAS).

When $t \longrightarrow \infty$, then $V \longrightarrow \infty$, therefore the system also is globally asymptotically stable (GAS). It is noteworthy to remind once again that the trajectories must be derivable, which is usually the case of real trajectory of these industrial vehicles.

As equations (25) and (29) are equivalent as well as (26) and (30), the study of the stability according to Lyapunov is the same for differential or tricycle traction units.

5. Simulation Setup

The architecture of Figure 2 with the Lyapunov stability controller (LPC) has been validated in simulation with the model of a hybrid AGV with the parameters listed in Table 1. The results have been obtained using Matlab software. Each simulation runs for 40 s.

The values of the parameters used in the simulation that correspond to a real AGV, the commercial Mouse AGV of Asti Mobile Robotics, are shown in Table 1 [1].

The application of the equations derived in the previous sections can be implemented by an algorithm which pseudocode is shown in Algorithm 1. Given a set of known motor torques $\{(M_L(t_i), M_R(t_i))\}$ in $t_i \in \{t_0, t_1, \ldots, T_{\text{sim}}\}$, this algorithm can be run in open loop, obtaining the evolution of the state variables: $(x_h, y_h, \varnothing_h)$, $(x_b, y_b, \varnothing_b)$, (θ_R, θ_L), $(\dot{\theta}_R, \dot{\theta}_L)$. For closed-loop execution, as it is part of the path-following control strategy, the function *Controller()* must be substituted by the expression of the function used to control the system in practice. In our case, the implementation of the controllers is explained in Section 4.

5.1. Trajectories Evaluated. To validate the proposed hybrid control strategy, three different trajectories have been used: a circle (Figure 4), an ellipse (Figure 5) and a lemniscate (Figure 6). These are characteristic curves that can be found in the workspace where AGV performs logistics operations.

Circle:

$$x(t) = a_{\text{cir}} \cos(t), \tag{37}$$

$$y(t) = a_{\text{cir}} \sin(t), \tag{38}$$

For this circular trajectory, a_{cir} is the radius. The kinematic constraints of the AGV are considered, so trajectory tracking becomes more difficult as the radius becomes smaller. If the curves are very sharp, the AGV could leave out the circular trajectory. After several simulation tests with the circular trajectory this parameter is set to $a_{\text{cir}} = 0.7$ m.

Ellipse:

$$x(t) = a_{\text{eli}} \cos(t), \tag{39}$$

$$y(t) = b_{\text{eli}} \sin(t). \tag{40}$$

In the ellipse trajectory, a_{eli} and b_{eli} are the semi-axes of the ellipse. Again, due to the kinematic constraints of the AGV, the trajectory tracking becomes more difficult as the semi-axes become smaller. If they are too small, the AGV could abandon the trajectory. After several simulation tests, the selected values are $(a_{\text{eli}}, b_{\text{eli}}) = (1.4, 0.7)$ m.

Lemniscate:

TABLE 1: Parameters of the AGV.

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

```
$t = t_0$
$\theta_R \longleftarrow \theta_{R_0}, \theta_L \longleftarrow \theta_{L_0}, \dot{\theta}_R \longleftarrow \dot{\theta}_{R_0}, \dot{\theta}_L \longleftarrow \dot{\theta}_{L_0}$
$X_h \longleftarrow X_{h_0}, Y_h \longleftarrow Y_{h_0}, \varnothing_h \longleftarrow \varnothing_{h_0}, X_b \longleftarrow X_{b_0}, Y_b \longleftarrow Y_{b_0}, \varnothing_b \longleftarrow \varnothing_{b_0}$
$A \longleftarrow \begin{bmatrix} m_T \bullet R_h/2 & m_T \bullet R_h/2 \\ I_h R_h/L_h & -I_h R_h/L_h \end{bmatrix}$
While $\{\mathrm{t} < \mathrm{T_{sim}}\}\{$
  $M_{\mathrm{eR}} \longleftarrow M_R - F_{\mathrm{SW}_R} \bullet \mathrm{sign} \bullet (\dot{\theta}_R),\ M_{\mathrm{eL}} \longleftarrow M_L - F_{\mathrm{SW}_L} \bullet \mathrm{sign} \bullet (\dot{\theta}_L)$
  $[M_{\mathrm{eR}}, M_{\mathrm{eL}}] \longleftarrow \mathrm{Controller}\,(x_h, y_h, \varnothing_h, \mathrm{trajectory})$
  $V_L \longleftarrow R_h \bullet \dot{\theta}_L,\ V_R \longleftarrow R_h \bullet \dot{\theta}_R$
  $V_h \longleftarrow (V_L + V_R)/2$
  $f_{rT} \longleftarrow 0.5 \bullet \delta_{\mathrm{air}} \bullet S_{\mathrm{AGV}} \bullet C_{\mathrm{aero}} \bullet (V_h^2) \bullet \mathrm{sign}\,(V_h) + 9.8 \bullet m_T \bullet C_{\mathrm{roll}} \bullet \mathrm{sign}\,(V_h)$
  $f_{\mathrm{rR}} \longleftarrow F_{\mathrm{vh}} \bullet \dot{\varnothing}_h + F_{\mathrm{sh}} \bullet \mathrm{sign}\,(\dot{\varnothing}_h)$
  $B \longleftarrow \begin{bmatrix} (M_{\mathrm{eR}} + M_{\mathrm{eL}})/2R_h - f_{\mathrm{rT}} \\ (M_{\mathrm{eR}} + M_{\mathrm{eL}})L_h/2R_h - f_{\mathrm{rR}} \end{bmatrix}$
  $[\ddot{\theta}_R \ddot{\theta}_L] \longleftarrow A^{-1} \times B$
  $\dot{X}_h \longleftarrow V_L + V_R/2 \cos(\varnothing_h),\ \dot{Y}_h \longleftarrow V_L + V_R/2 \sin(\varnothing_h),\ \dot{\varnothing}_h \longleftarrow V_R + V_L/L_h$
  $\dot{X}_b \longleftarrow V_h \cos(\gamma)\cos(\varnothing_b),\ \dot{Y}_b \longleftarrow V_h \cos(\gamma)\sin(\varnothing_b),\ \varnothing_b \longleftarrow V_h/L_b \sin(\gamma)$
  $\theta_R \longleftarrow \int \dot{\theta}_R \mathrm{d}t,\ \theta_L \longleftarrow \int \dot{\theta}_L \mathrm{d}t$
  $\dot{\theta}_R \longleftarrow \int \ddot{\theta}_R \mathrm{d}t,\ \dot{\theta}_L \longleftarrow \int \ddot{\theta}_L \mathrm{d}t$
  $X_h \longleftarrow \int \dot{X}_h \mathrm{d}t, Y_h \longleftarrow \int \dot{Y}_h \mathrm{d}t, \varnothing_h \longleftarrow \int \dot{\varnothing}_h \mathrm{d}t$
  $X_b \longleftarrow \int \dot{X}_b \mathrm{d}t, Y_b \longleftarrow \int \dot{Y}_b \mathrm{d}t, \varnothing_b \longleftarrow \int \dot{\varnothing}_b \mathrm{d}t$
  If $\varnothing_h > \varnothing_b + \gamma_{\mathrm{max}}$ then
    $\varnothing_h \longleftarrow \varnothing_b + \gamma_{\mathrm{max}}$
  endIf
  If $\varnothing_h < \varnothing_b + \gamma_{\mathrm{min}}$ then
    $\varnothing_h \longleftarrow \varnothing_b + \gamma_{\mathrm{min}}$
  endIf
  $\gamma \longleftarrow \varnothing_h - \varnothing_b$
  If $\dot{X}_b \sin(\varnothing_b) < > \dot{X}_b \cos(\varnothing_b)$ then
    stopSimulation
  endIf
  If $(\dot{X}_b \sin(\varnothing_b + \gamma) - \dot{X}_b \cos(\varnothing_b + \gamma)) < > \dot{\varnothing}_b L_b \cos(\varnothing_b)$ then
    stopSimulation
  endIf
  $t = t + \Delta T$
}endWhile
```

ALGORITHM 1: AGV model simulation algorithm.

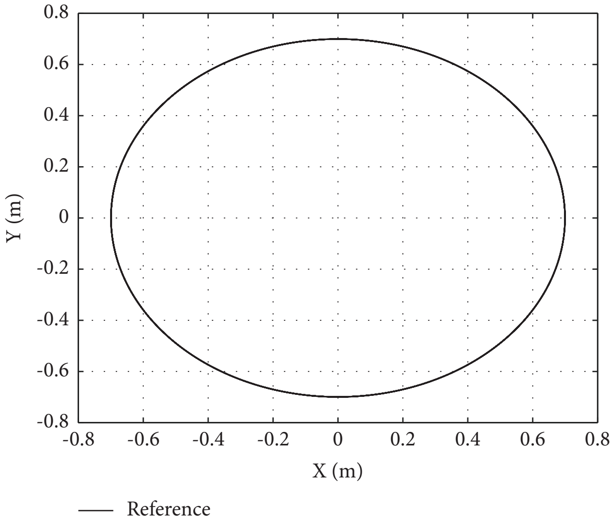


Figure 4: Circle trajectory.

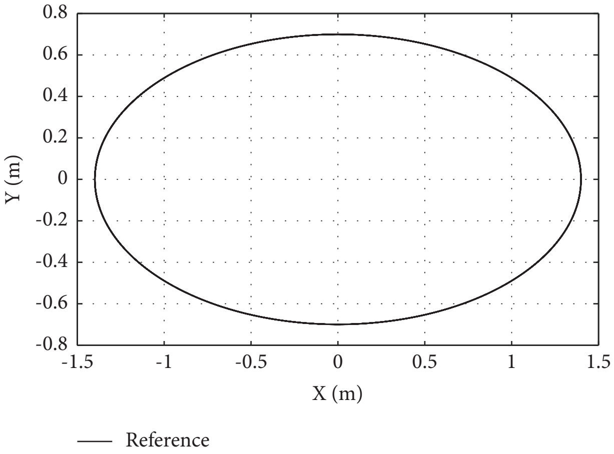


Figure 5: Elliptical trajectory.

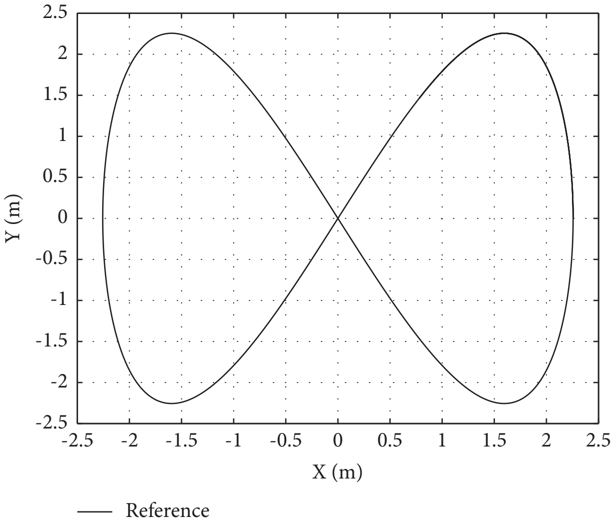


Figure 6: Lemniscate trajectory.

Table 2: Tuning parameters of the PID for the different trajectories.

Trajectory	Kp	Kd	Ki
Circle	8.4402	2.221	1.4104
Ellipse	12.4902	2.2661	1.4421
Lemniscate	25.4700	2.2300	0.1201

$$x(t) = \sqrt{2} \cdot a_{\text{lem}} \cdot \frac{\cos(t)}{1 + \sin(t)^2}, \tag{41}$$

$$y(t) = \sqrt{2} \cdot a_{\text{lem}} \cdot \frac{\cos(t) \cdot \sin(t)}{1 + \sin(t)^2}. \tag{42}$$

In the lemniscate trajectory, a_{lem} is the width of the curve. After several simulation tests, the value of a_{lem} is set to 1.6 m.

5.2. Trajectory Tracking Conventional Regulator Used for Comparison Purposes. As already explained, in these commercial vehicles the guidance error controller is usually a PID. Therefore, a comparison of the proposed Lyapunov stability controller with a PID controller given by (43) is performed.

$$u(t) = K_p e(t) + K_i \int_0^t e(t)\mathrm{d}t + k_d \frac{\mathrm{d}e(t)}{\mathrm{d}t}. \tag{43}$$

The PID parameters $[K_p, K_i, K_d]$ are tuned with the following procedure. A simulation scenario is created with Simulink software, considering the complete dynamic model of the AGV. The guidance error will be obtained for three different trajectories, namely circle, ellipse, and lemniscate. For each trajectory, the gains of the PID have been optimized with genetic algorithms.

Equation (44) was used as fitness function of the GA. T_{sim} denotes the total simulation time when the AGV correctly completes the simulation without losing its track. e_{MAX} represents the maximum assumable error, in this case this value was set to 40 cm. This fitness function is piecewise defined. The first row is used to ensure the AGV completes the simulation without leaving out the route. The second row is used to reduce the tracking error to zero.

$$cf_{\text{gui}} = \begin{cases} 1 + \dfrac{(T_{\text{sim}} - t_i)}{T_{\text{sim}}} & |e(t_i)| > e_{\text{MAX}}, \\ \dfrac{1}{T_{\text{sim}} \cdot e_{\text{MAX}}} \displaystyle\sum_i |e(t_i)| & |e(t_i)| < e_{\text{MAX}} \wedge t_i = T_{\text{sim.}} \end{cases} \tag{44}$$

The configuration of the GA is as follows. The size of the population was set to 50 individuals, the crossover fraction was 0.8, the mutation rate was 0.2, and the elite size was 5%. Initial individuals were randomly initialized. The PID tuning parameters obtained during this optimization are shown in Table 2.

It is important to correctly tune the gains of the PID, if the gains are set with the values of Table 2, the AGV does not looses the track. However, poorly tuned gains can make the AGV to lose the track as shown in Figures 7–9. It is assumed that the AGV loses the trajectory when the tracking error is larger than 40 cm. When this event raises the simulation is stopped.

To summarize, the here proposed control strategy uses a LPC for trajectory tracking control and a PI regulator for speed control, that is, LPC + PI. It has been compared to a conventional PID controller for trajectory tracking that has the same PI for speed control, that is, PID + PI. In both cases the gains of the PI speed controller have the same values.

5.3. Comparison Metrics. To make a comparison of the different controllers in a quantitative way some metrics that measure the performance when tracking the different reference trajectories are defined. The metrics applied for the evaluation of the control strategies are expressed by equations.

$$\text{MAE} = \frac{1}{T_{\text{sim}}} \sum_i |e(t_i)| * T_s(t_i), \tag{45}$$

$$\text{RMSE} = \sqrt{\frac{1}{T_{\text{sim}}} \sum_i e(t_i)^2 * T_s(t_i)}, \tag{46}$$

$$\text{ME} = \frac{1}{T_{\text{sim}}} \sum_i e(t_i) T_s(t_i), \tag{47}$$

$$\text{STD} = \sqrt{\frac{1}{T_{\text{sim}}} \sum_i |e(t_i) - \text{ME}|^2 * T_s(t_i)}, \tag{48}$$

$$\text{MAX} = \underset{t_i \in T_{\text{sim}}}{\text{MAX}} \left[|e(t_i)| \right]. \tag{49}$$

6. Discussion of Simulation Results of Trajectory Tracking Controllers

The two AGV tracking trajectory controllers, PID and LPC, have been simulated for the three defined paths: circle, ellipse, and lemniscate. The AGV reference longitudinal velocity v_{refd}, is the absolute value of a sinusoidal signal with amplitude 1 m/s. The simulation lasts 40 s, although it ends if the AGV completes or leaves out the trajectory.

Figures 10–12 show the results obtained for the trajectory described by the AGV drive unit and the guidance error of the two controllers, LPC and PID. The color coding is as follows. In the trajectory tracking plots, the reference trajectory is the black line, the one followed with the LPC controller is the blue line and the one followed by the AGV with the PID controller is in red. In the tracking error plots on the x-axis and y-axis, the Lyapunov controller error is represented in blue and the PID error in red.

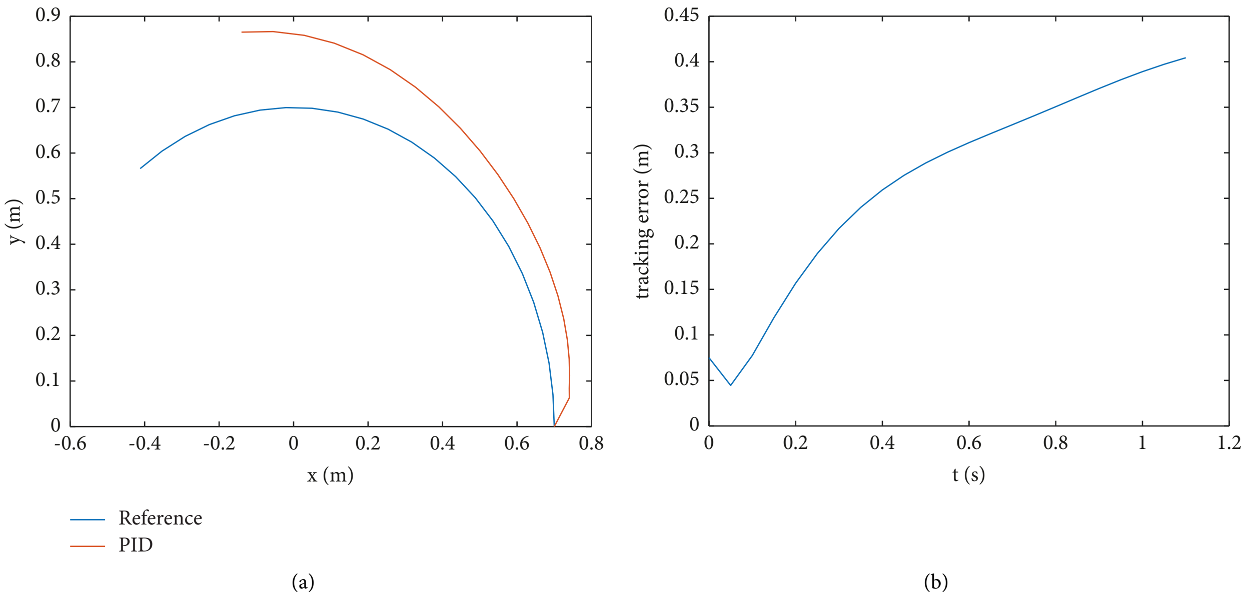


Figure 7: Tracking of circle with poorly tuned PID (a) and its tracking error (b).

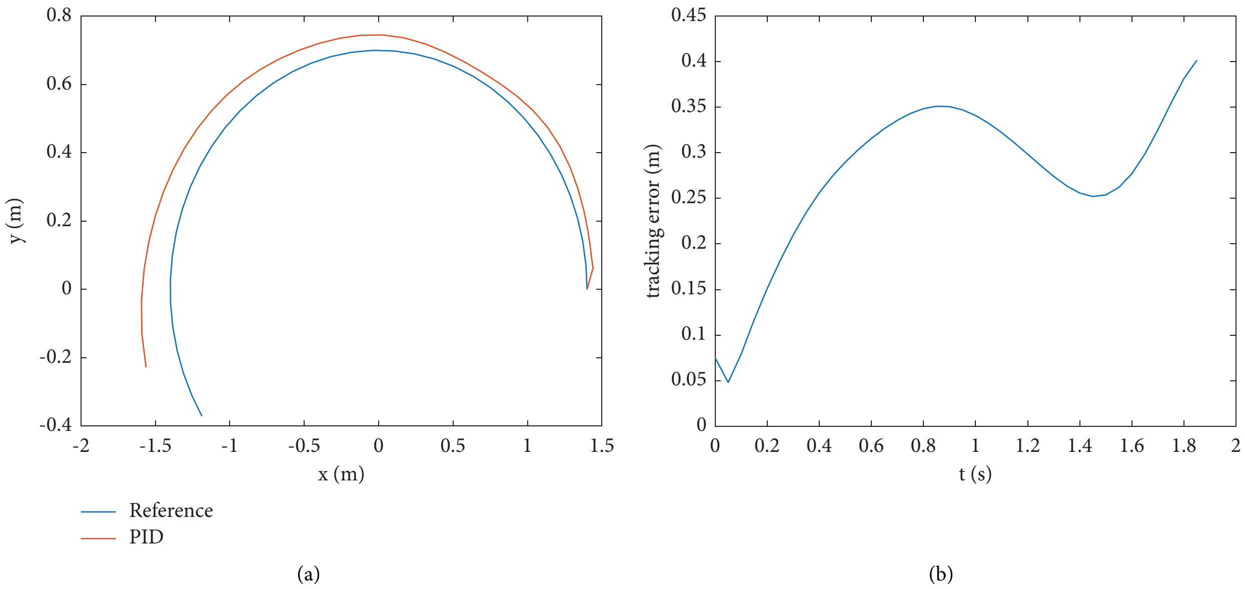
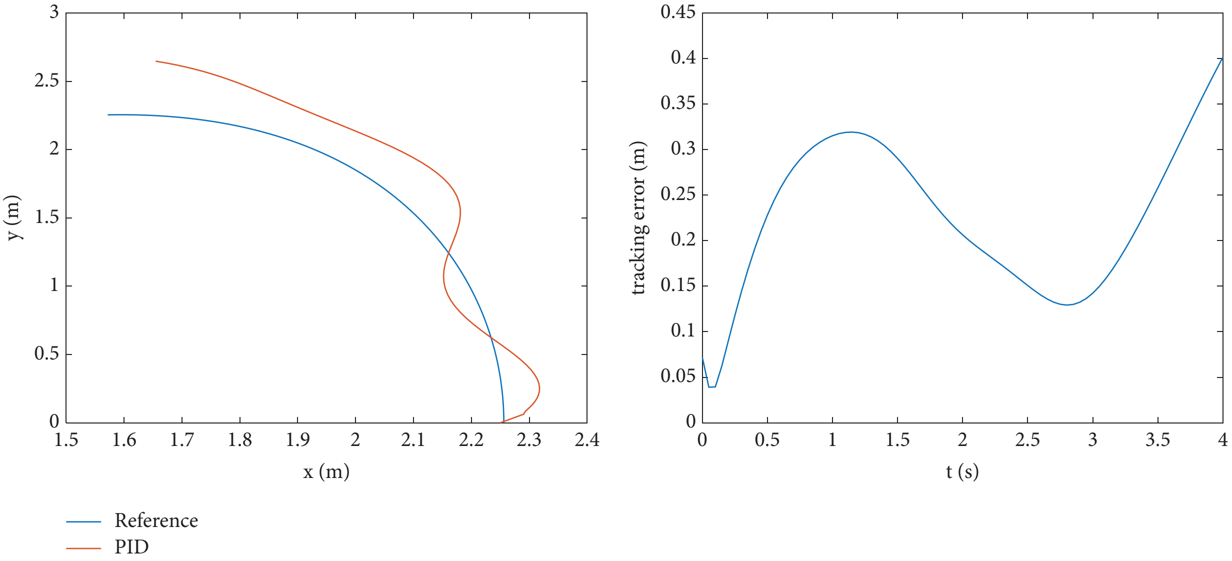


Figure 8: Tracking of ellipse with poorly tuned PID (a) and its tracking error (b).

Figures 13–15 show the yaw rate, the lateral acceleration and the longitudinal velocity obtained when the LPC and the PID are applied. The color coding is as follows. The results obtained when the LPC is applied are shown in blue, and in red for the PID. In the figure that illustrates the longitudinal velocity, the reference for the velocity $v_{\mathrm{ref}\,d}$ is depicted in black.

For the circular trajectory, we consider that the AGV vehicle is initially located at the point (0.7, 0) and moves counterclockwise. Figure 10 shows the simulation results for this trajectory. It can be seen (Figure 10(a)) how the LPC trajectory and the reference trajectory are very close. With the PID controller, a greater distance to the reference is observed. In the x and y axis tracking error graphs it can be seen that the errors of the LPC controller are closer to zero and are much smaller than with the PID (Figure 10(c)).

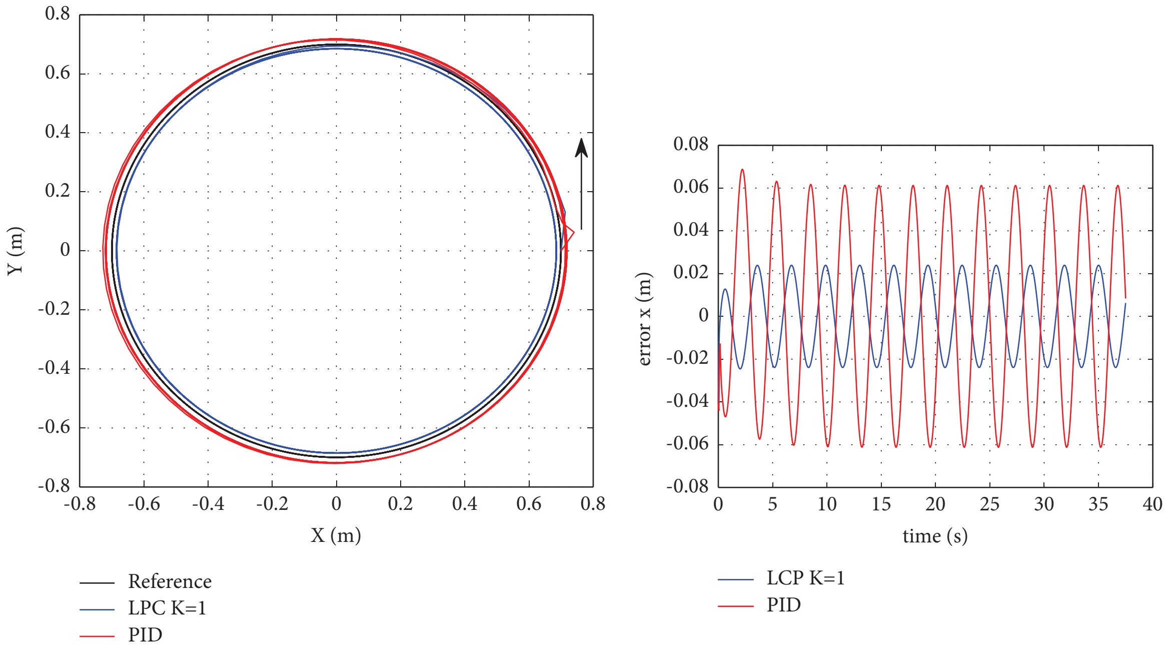
In Figure 13 it is possible to see how the yaw rate is larger at the starting of the simulation. This may be caused because the AGV needs to be aligned with the trajectory. After this, the yaw rate tends to stabilize around a constant value. This result is logic as the curvature in the circle is also constant. On the other hand, as the turning radius is constant the lateral acceleration has a similar shape to the speed. Finally, it Figure 10(c), it is possible to observe how the controller correctly follows the speed profile sinusoidal in absolute value.



(a)

(b)

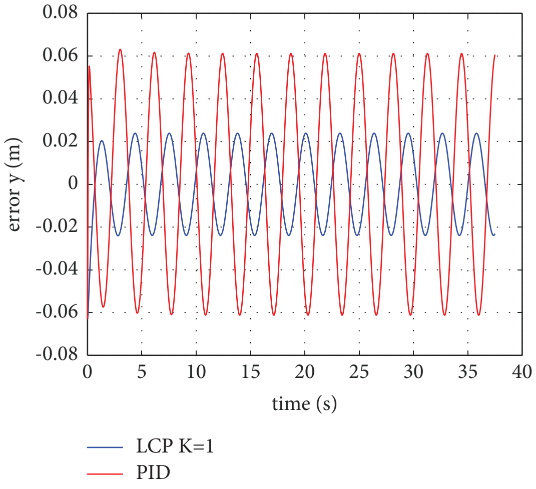
Figure 9: Tracking of lemniscate with poorly tuned PID (a) and its tracking error (b).



(a)

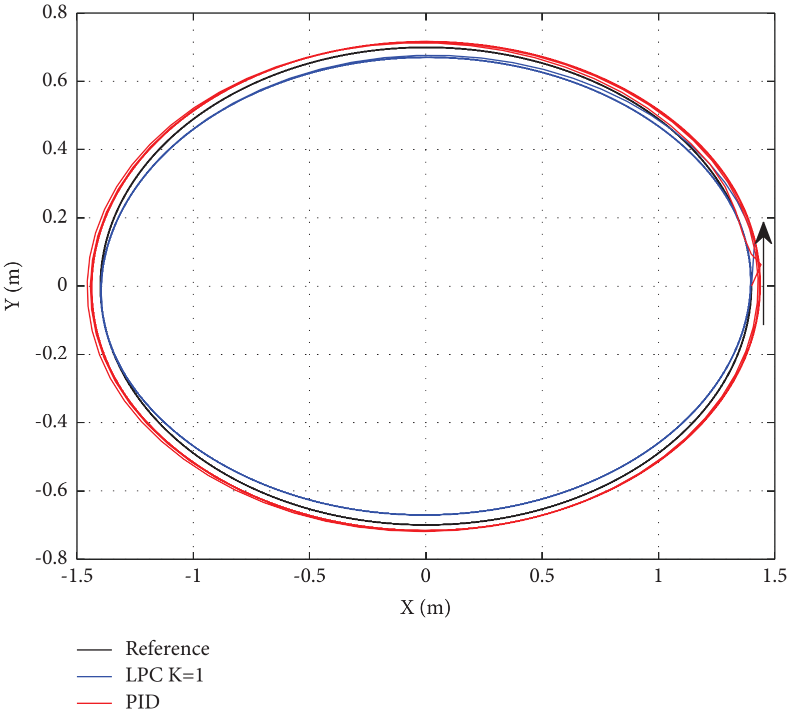
(b)

Figure 10: Continued.



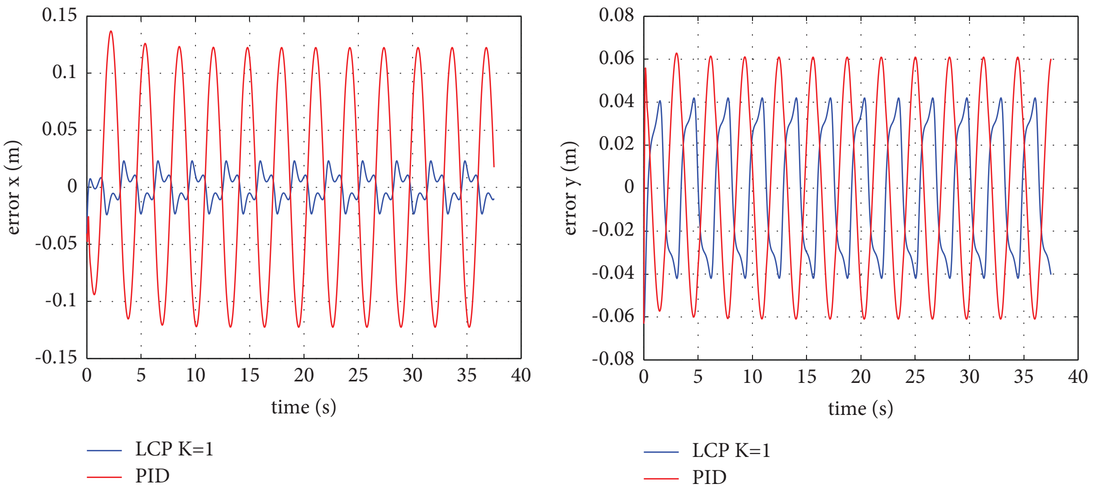
(c)

Figure 10: Circular trajectory tracking (a) and tracking errors, *x* (b) and *y* (c) axes.



(a)

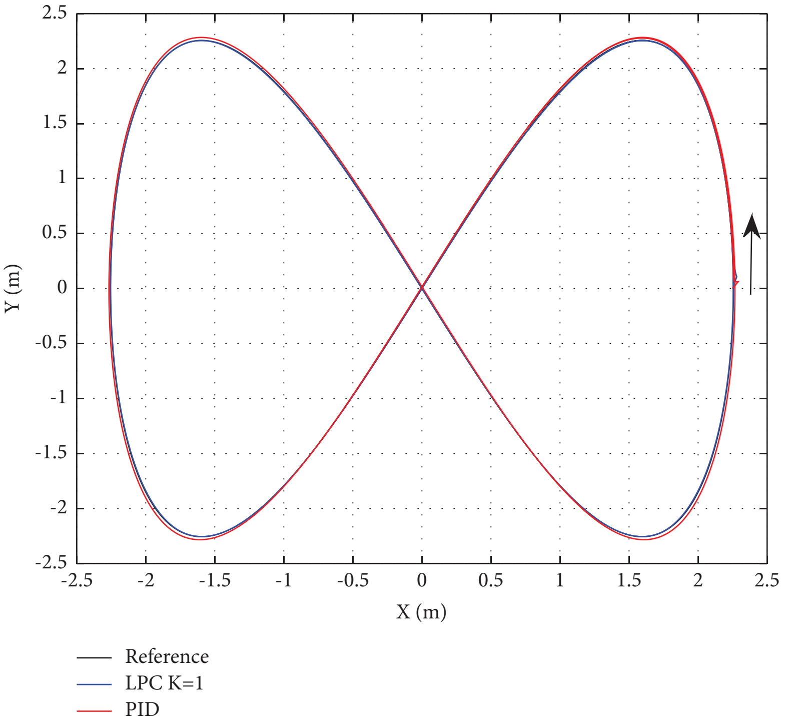
Figure 11: Continued.



(b)

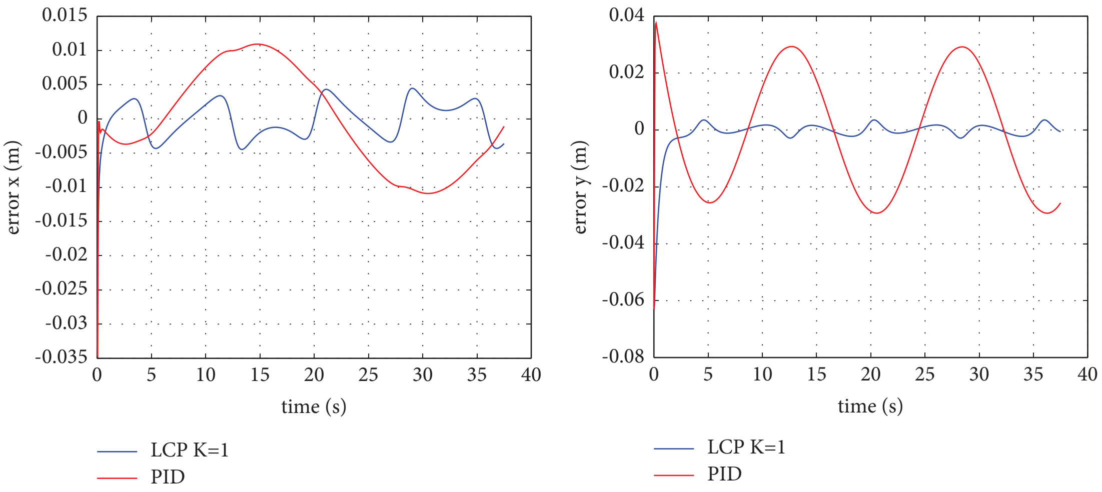
(c)

Figure 11: Elliptical trajectory tracking (a) and tracking errors, x (b) and y (c) axes.



(a)

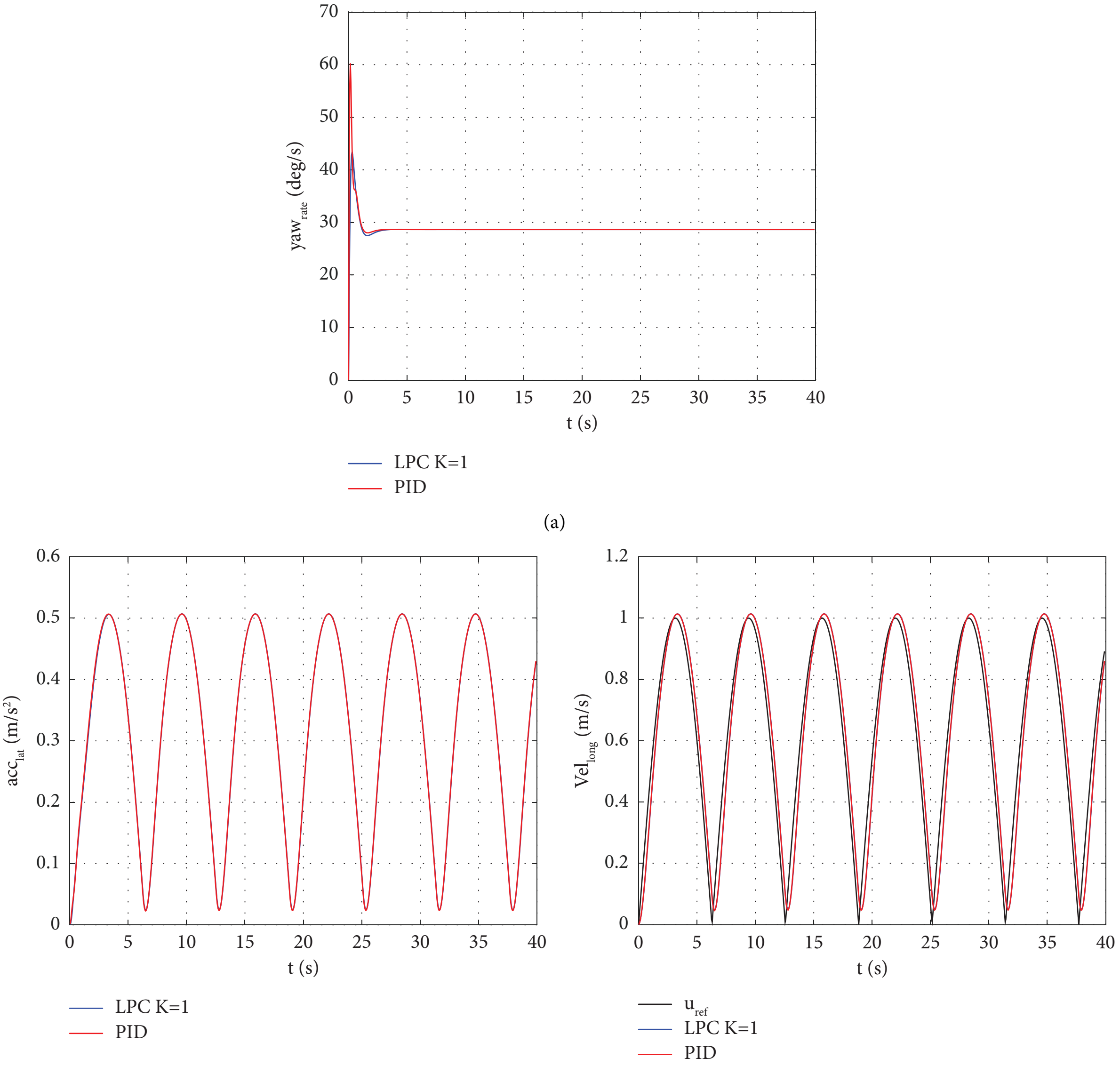
Figure 12: Continued.



(b)

(c)

Figure 12: Lemniscate trajectory tracking (a) and tracking errors, x (b) and y (c) axes.

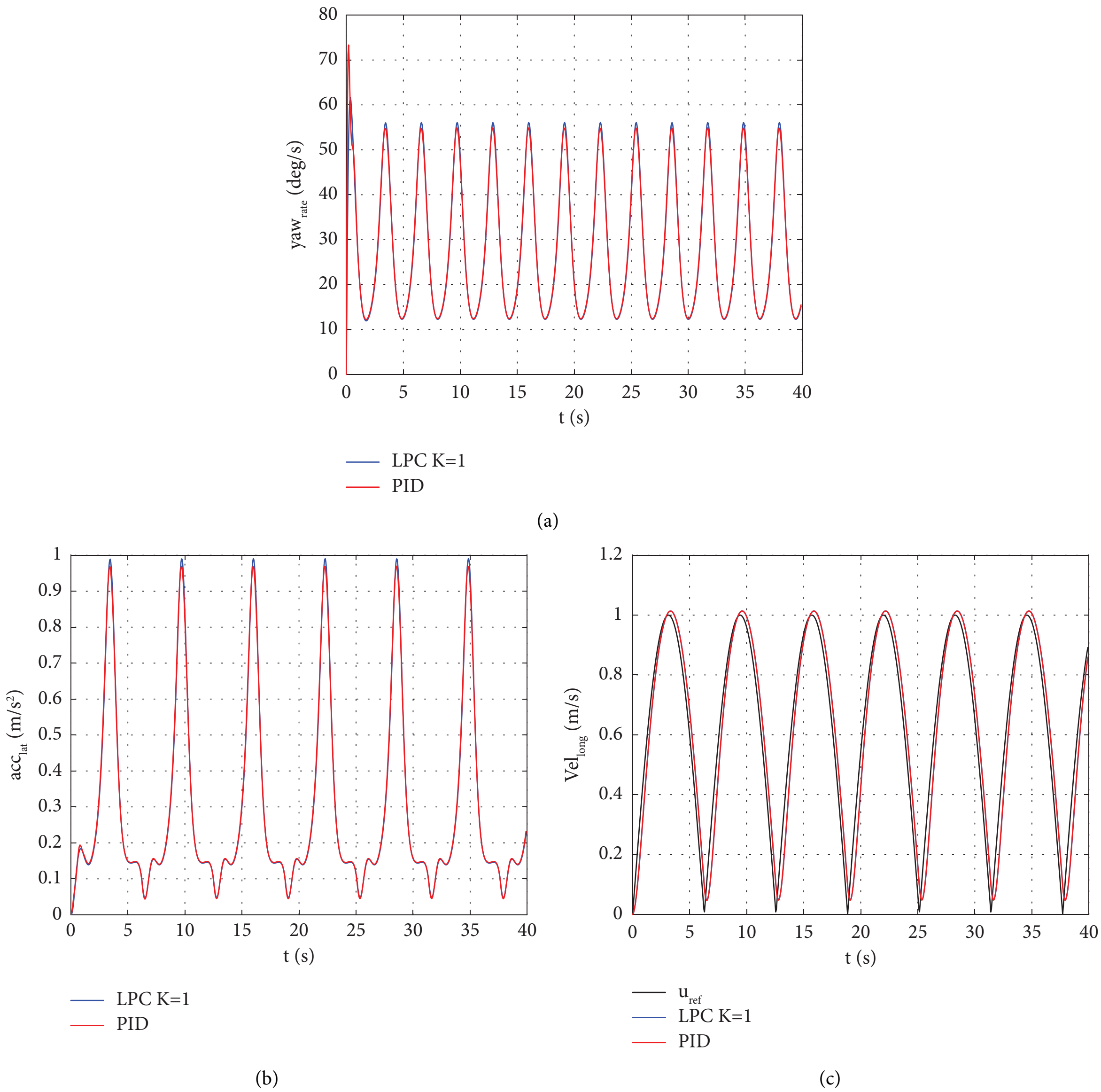


(a)

(b)

(c)

Figure 13: Circular trajectory, yaw rate (a), lateral acceleration (b), and longitudinal velocity (c).



(a)

(b)

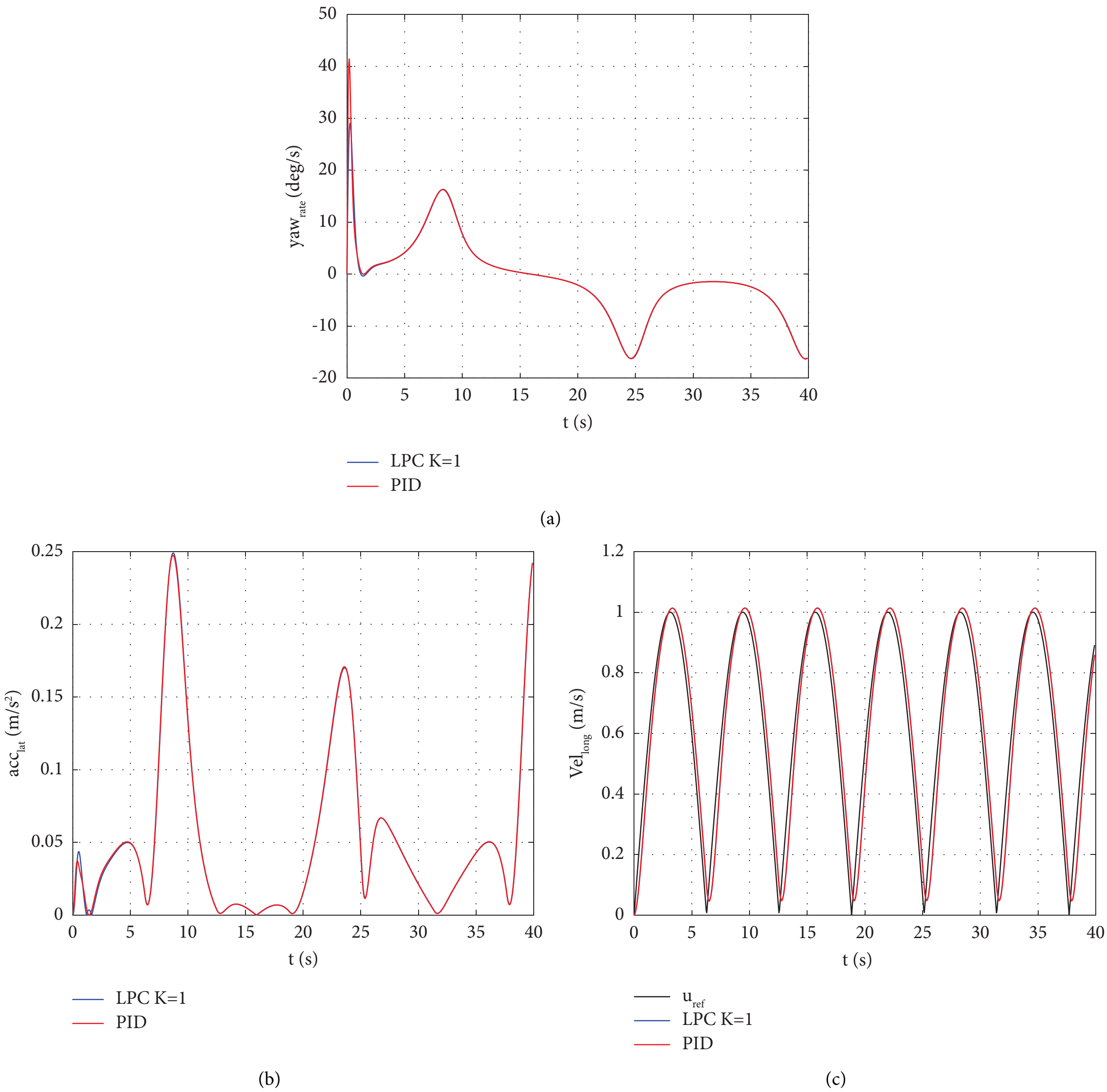
(c)

Figure 14: Elliptical trajectory, yaw rate (a), lateral acceleration (b), and longitudinal velocity (c).

For the elliptical trajectory, the AGV starts at the right side of the ellipse, at point (1.4, 0), and moves counterclockwise. In Figure 11, the simulation result for tracking the elliptical trajectory is shown. The LPC controller tracks the reference trajectory better than the PID, especially at the ends of the ellipse, where they overlap. The PID maintains a nearly constant and larger error than with the LPC over the entire ellipse. The error of the LPC controller remains close to zero, presenting smaller error in amplitude than the PID in both the x-axis and y-axis, especially in the x-axis where the error amplitude with the LPC controller is much smaller.

The large value of yaw rate at the start of the simulation is also observable in the case of the ellipse in Figure 14. However, in this case the yaw rate does not tend to a constant value as the curvature changes along the trajectory. Like in the error signal, the values show a repetitive pattern as the trajectory is repeated several times. The lateral acceleration changes with the longitudinal speed, when the longitudinal speed becomes zero, the lateral acceleration also does it. The changes in the turning radius also affect this value, when the AGV moves in the zones of the ellipse where the radius decreases the acceleration grows. Finally, as in the case of the circular trajectory, the speed profile is correctly followed.

For the lemniscate trajectory, the AGV starts at the right end, at point (2.25, 0), and moves counterclockwise. Figure 12(a), shows the simulation results of this trajectory tracking with the PID controller and the LPC. The trajectories are very similar, indeed both controllers track the reference trajectory quite well, although the LPC controller is slightly better as shown in the guiding errors. In addition, the PID controller presents a disturbance at the beginning of the AGV motion, which causes the error to present large peaks (Figure 12(b)).



(a)

(b)

(c)

Figure 15: Lemniscate trajectory, yaw rate (a), lateral acceleration (b), and longitudinal velocity (c).

Yaw rate in Figure 15 reaches smaller values than in previous trajectories, this is due to the in the middle of the lemniscate the curvature is very low, and the AGV behaves like in a straight line. Another interesting result is that in this case, the yaw rate takes positive and negative values. This is explained because in the first half of the lemniscate the AGV travels counterclockwise, and the second half clockwise. Regarding the lateral acceleration, as expected the zones of the trajectory with a low curvature, mainly the center of the lemniscate, experiment the lowest lateral acceleration. Conversely in the zones with larger curvature the acceleration tends to be larger. Again, the controller follows perfectly the sinusoidal speed profile.

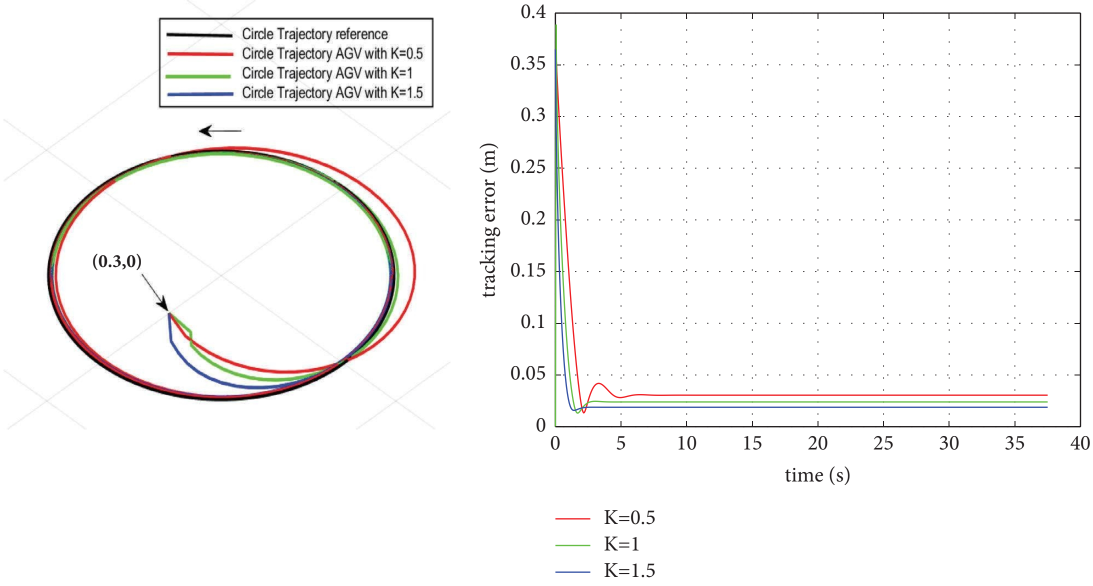
The values of the metrics are shown in Table 3, where the columns present the mean absolute error (MAE), root mean squared error (RMSE), the standard deviation, the maximum value of the guidance error, and the yaw angle rate (YAR) obtained when the LPC or the PID are applied to the reference trajectories.

Analyzing the results of Table 3, the error values are smaller when the LPC controller is used in the tracking of each of the reference trajectories. Lyapunov controller gives good results in curved trajectories and with varying radius. If we look at the results of the table by type of trajectory, the lemniscate is the one with the smallest error for both, the PID and LPC controllers.

If we make a comparison between LPC and PID for each type of trajectory, the LPC controller gives better results. For the ellipse, the MAE with the LPC shows an improvement by 65.22% with respect to the PID, the RMSE improvement is 65.38%, 69.15% improvement of the STD and regarding the MAX metric, the LPC improves by 65.52%.

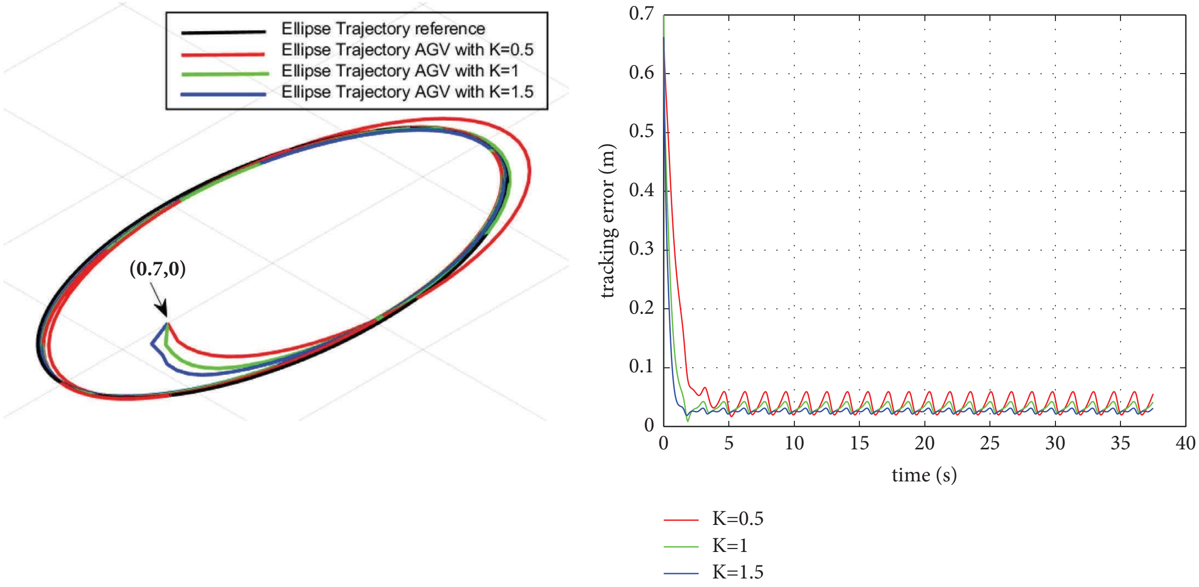
Table 3: Comparison of the controller performance for different trajectories.

Trajectory	MAE		RMSE		STD		MAX	
	LPC	PID	LPC	PID	LPC	PID	LPC	PID
Circle	0.2	0.57	0.279	0.79	0.35	1.02	1.17	2.45
Ellipse	0.4	1.15	0.54	1.56	0.62	2.01	2.22	6.44
Lemniscate	0.04	0.175	0.0542	0.24	0.058	0.28	1.83	2.085



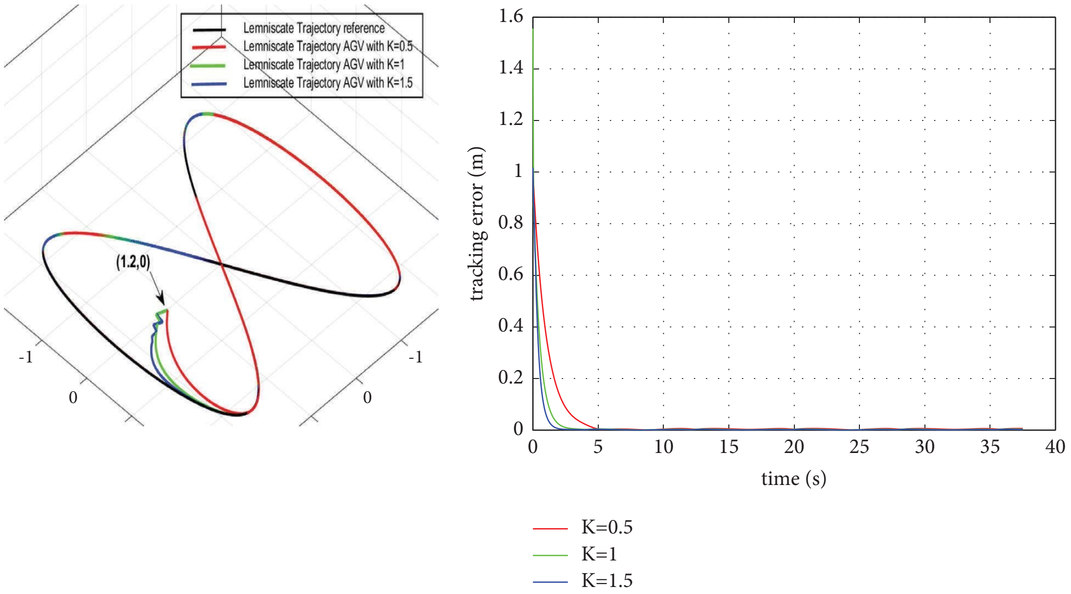
(a) (b)

Figure 16: Circular trajectory (a) for different values of gain K and its corresponding guiding error (b).



(a) (b)

Figure 17: Ellipse trajectory (a) for each K and its corresponding guiding error (b).



(a)

(b)

Figure 18: Lemniscate trajectory (a) for each K and its corresponding guidance error (b).

Summarizing, the Lyapunov LPC stability controller outperforms the PID for the guidance control of the tested trajectories.

6.1. Sensibility Analysis: Influence of the Gain of the LPC. An analysis of the effects of varying the gain K of the Lyapunov controller has been carried out. The paths followed by the AGV for different LPC gains when tracking each of the proposed trajectories are studied. The simulation period is 40 s. Three values of K have been considered: less than 1, 1, and bigger than 1.

In the following figures of the trajectories and the errors, the AGV trajectory tracking response with the LPC controller is represented in blue for $K = 1.5$, in green for $K = 1$ and in red for $K = 0.5$. The reference trajectory is shown in black.

Figure 16 shows the effect of the variation of the Lyapunov control gain K when it follows the circular trajectory. The AGV starts at (0.3, 0), outside the circle, and moves counterclockwise.

It can be observed in Figure 16 that for $K = 1.5$ (blue line), the variations at the beginning of the AGV movement are larger but it quickly follows the circular reference (black line). Reducing the LPC gain $K = 1$ (green line) makes the path to join the reference shorter. With a LPC gain smaller than 1, i.e., $K = 0.5$ (red line), the response is even quicker, but the errors are bigger. In fact, there is a stationary error with all the values of K tested (Figure 16, right), being the LPC with $K = 1.5$ the more stable and with less stationary error.

Figure 17 shows the effects of varying the gain K of the LPC when the AGV follows the elliptical trajectory. The AGV starts at (0.7, 0), outside the path, and moves counterclockwise.

Again, the larger the gain of the LPC the slower it is to join the reference. But observing the error (Figure 17(a)), the LPC with the smallest gain $K = 0.5$ requires more time to reach the reference and it presents the more unstable response, with oscillations. The green line ($K = 1$) shows a peak at the beginning and then it also presents an oscillatory stationary error, although the amplitude of these oscillations is smaller than with $K = 0.5$. Finally, the solution with $K = 1.5$ (blue line) is the more stable although it still presents small error oscillations. The stationary error is bigger than with the circular trajectory.

Figure 18 shows the response of the AGV following the lemniscate trajectory with the three values of K. The vehicle starts at (1.2, 0), outside the path, and moves counterclockwise.

As in the previous trajectories, the larger the K the shorter the time it requires to reach the reference trajectory (Figure 18(a)). Again, there are small perturbations at the beginning of the movement. In this case, the LPC makes the AGV follow the trajectory with a very small error, that presents smooth oscillations, but it can be considered stable.

In conclusion, it seems that Lyapunov controller K values bigger than 1 gives better response as the stationary error is smaller than with smaller values of K, and the response is also faster.

7. Conclusions and Future Works

In this article, a control architecture is proposed that combines a Lyapunov controller for tracking the trajectory of an AGV and PID controllers. This hybrid conventional and advanced control scheme has proven very efficient in industrial and engineering applications.

The objective of proposing an LPC system is to guarantee the stability of the controller and provide it with robustness against non-linearities, such as those related to the trajectories and friction experienced by the wheels or the traction unit of the AGV, which brings it closer to reality.

The main advantages of the controller proposed in this paper can be summarized as follows:

(i) The stability of the trajectory tracking is mathematically demonstrated, something that is lacked in most of the papers found in the literature.

(ii) The independence of the tuning of the speed control and the trajectory control that is something that facilitates the design of these controllers. Specifically, the speed control is tuned adjusting the gains of the PI speed controllers. On the other hand, the tuning of trajectory tracking control is done by adjusting the gain K in the control law.

(iii) The control law can be applied to different mobile robots adjusting the J matrix.

To evaluate the controller, a hybrid AGV that combines the tricycle and differential kinematic models is used, with parameters that correspond to a commercial vehicle. The LPC has been compared with a PID controller optimized with genetic algorithms when following three trajectories that can be found in the work environment of these industrial vehicles. The LPC outperforms the PID in all cases in terms of trajectory tracking.

Furthermore, the sensitivity of the LPC has been analyzed for different values of its gain, which allows it to be better adjusted.

Among other possible future works, it is possible to highlight the application of optimization techniques to tune the gain of the LPC controller. The influence of the changes in friction in the performance of the controller is something that deserves to be analyzed. Finally, the validation of the controller with an industrial AGV prototype could be worth it.

Data Availability

The findings of this study have been generated by the equations and parameters cited within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

Open Access funding enabled and organized by CRUE-UNIRIS Gold.

References

[1] J. E. Sierra-García and M. Santos, "Mechatronic modelling of industrial AGVs: a complex system architecture," *Complexity*, vol. 2020, pp. 1–21, 2020.

[2] F. Espinosa, C. Santos, and J. E. Sierra-García, "Transporte multi-AGV de una carga: estado del arte y propuesta centralizada," *Revista Iberoamericana de Automática e Informática industrial*, vol. 18, no. 1, pp. 82–91, 2020.

[3] S. Vakaruk, J. E. Sierra-García, A. Mozo, and A. Pastor, "Forecasting automated guided vehicle malfunctioning with deep learning in a 5G-based industry 4.0 scenario," *IEEE Communications Magazine*, vol. 59, no. 11, pp. 102–108, 2021.

[4] J. E. Sierra-Garcia and M. Santos, "Combining reinforcement learning and conventional control to improve automatic guided vehicles tracking of complex trajectories," *Expert Systems*, vol. 41, no. 2, 2022.

[5] M. De Ryck, M. Versteyhe, and F. Debrouwere, "Automated guided vehicle systems, state-of-the-art control algorithms and techniques," *Journal of Manufacturing Systems*, vol. 54, pp. 152–173, 2020.

[6] D. Preuveneers and E. Ilie-Zudor, "The intelligent industry of the future: a survey on emerging trends, research challenges and opportunities in Industry 4.0," *Journal of Ambient Intelligence and Smart Environments*, vol. 9, no. 3, pp. 287–298, 2017.

[7] R. Sánchez, J. E. Sierra-García, and M. Santos, "Modelado de un AGV híbrido triciclo-diferencial," *Revista Iberoamericana de Automática e Informática industrial*, vol. 19, no. 1, pp. 84–95, 2021.

[8] W. Nakimuli, J. Garcia-Reinoso, J. E. Sierra-Garcia, P. Serrano, and I. Q. Fernández, "Deployment and evaluation of an industry 4.0 use case over 5G," *IEEE Communications Magazine*, vol. 59, no. 7, pp. 14–20, 2021.

[9] T. M. Do and H. N. Nguyen, "Path tracking control approach and processing lumped disturbances for AGV in recent years: an overview," *Advances in Information and Communication Technology*, in *Proceedings of the International Conference on Advances in Information and Communication Technology*, pp. 180–198, Switzerland, December 2023.

[10] W. P. N. D. Reis, G. E. Couto, and O. M. Junior, "Automated guided vehicles position control: a systematic literature review," *Journal of Intelligent Manufacturing*, vol. 34, no. 4, pp. 1483–1545, 2023.

[11] T. L. Bui, P. T. Doan, H. K. Kim, and S. B. Kim, "Trajectory tracking controller design for AGV using laser sensor based positioning system," in *Proceedings of the 2013 9th Asian Control Conference (ASCC)*, pp. 1–5, IEEE, Istanbul, Turkey, June 2013.

[12] S. Zhang, Q. Xia, S. Cheng, M. Chen, and G. Xiao, "Research on lyapunov-based predictive path following control of AGV based on time constraint," *International Journal of Control, Automation and Systems*, vol. 20, no. 12, pp. 4005–4014, 2022.

[13] Y. Koubaa, M. Boukattaya, and T. Dammak, "Adaptive control of nonholonomic wheeled mobile robot with unknown parameters," in *Proceedings of the 2015 7th International Conference on Modelling, Identification and Control (ICMIC)*, pp. 1–5, IEEE, Sousse, Tunisia, December 2015.

[14] J. Shang, J. Zhang, and C. Li, "Trajectory tracking control of AGV based on time-varying state feedback," *EURASIP Journal on Wireless Communications and Networking*, vol. 2021, no. 1, pp. 162–212, 2021.

[15] X. Zhou and L. He, "Sliding mode control for trajectory tracking of AGV based on improved fast stationary power approach law," *Journal of Physics: Conference Series*, vol. 2417, no. 1, p. 012032, 2022.

[16] W. Liu, Y. Wan, Q. Zhang, Y. Yu, P. Liu, and Z. Shi, "Trajectory tracking control of four-wheel steering automatic guided vehicle under the working condition of moving centroid," *Proceedings of the Institution of Mechanical*

Engineers-Part D: Journal of Automobile Engineering, vol. 237, no. 4, pp. 691–705, 2023.

[17] B. Jiang, J. Li, and S. Yang, "An improved sliding mode approach for trajectory following control of nonholonomic mobile AGV," *Scientific Reports*, vol. 12, no. 1, 2022.

[18] H. Aithal and S. Janardhanan, "Trajectory tracking of two wheeled mobile robot using higher order sliding mode control," in *Proceedings of the 2013 International Conference on Control, Computing, Communication and Materials (ICCCCM)*, pp. 1–4, Allahabad, India, August 2013.

[19] 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.

[20] H. Taghavifar and A. Mohammadzadeh, "Adaptive robust terminal sliding mode control with integral backstepping synthesized method for autonomous ground vehicle control," *Vehicles*, vol. 5, no. 3, pp. 1013–1029, 2023.

[21] H. Xu, Z. Yu, X. Lu, S. Wang, S. Li, and S. Wang, "Model predictive control-based path tracking control for automatic guided vehicles," in *Proceedings of the 2020 4th CAA International Conference on Vehicular Control and Intelligence (CVCI)*, pp. 627–632, IEEE, Hangzhou, China, December 2020.

[22] X. Liu, G. Wang, and K. Chen, "Nonlinear model predictive tracking control with c/gmres method for heavy-duty agvs," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 12, pp. 12567–12580, 2021.

[23] K. Zhang, Q. Sun, and Y. Shi, "Trajectory tracking control of autonomous ground vehicles using adaptive learning MPC," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 12, pp. 5554–5564, 2021.

[24] X. Zhou, L. Jin, and Y. Liu, "Modeling and simulation research of heavy-duty AGV tracking control system based on magnetic navigation," *Journal of Physics: Conference Series*, vol. 1746, no. 1, p. 012021, 2021.

[25] A. J. Moshayedi, J. Li, N. Sina et al., "Simulation and validation of optimized pid controller in agv (automated guided vehicles) model using pso and bas algorithms," *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1–22, 2022.

[26] N. T. Binh, B. N. Dung, L. X. Chieu et al., "Deep learning-based object tracking and following for AGV robot," *Intelligent Systems and Networks*, in *Proceedings of the International Conference on Intelligent Systems and Networks*, pp. 204–214, Singapore, March 2023.

[27] J. E. Sierra-García, V. Fernández-Rodríguez, M. Santos, and E. Quevedo, "Development and experimental validation of control algorithm for person-following autonomous robots," *Electronics*, vol. 12, no. 9, p. 2077, 2023.