FEEDFORWARD CONTROL SYSTEM DEVELOPMENT AND IMPLEMENTATION FOR LONGITUDINAL AND LATERAL MOTION OF A FOUR-WHEELED ROBOT

Younes El koudia[1], Jarou Tarik[2], Abdouni Jawad[3], Sofia El Idrissi[4] and Elmahdi Nasri[5]

[1]Advanced Systems Engineering Laboratory, National School of Applied Sciences, Kenitra, Morocco

[2]Advanced Systems Engineering Laboratory, National School of Applied Sciences, Kenitra, Morocco

[3]Advanced Systems Engineering Laboratory, National School of Applied Sciences, Kenitra, Morocco

[4]Advanced Systems Engineering Laboratory, National School of Applied Sciences, Kenitra, Morocco

[5]Advanced Systems Engineering Laboratory, National School of Applied Sciences, Kenitra, Morocco

ABSTRACT

The work presented in this paper advances through a series of physics-based increasing fidelity models that are used to design robot controllers that respect the robot's capabilities. It also develops a reference simple controller that is applicable to a large subset of tracking conditions, most of which involve non-invasive or highly dynamic movements, defines path geometry in response to the control problem, and develops both a simple geometric control and a dynamic model predictive control approach. In this research, we propose the mathematical modeling of the longitudinal and lateral movements utilizing PID with a feed-forward controller for a nonlinear model with disturbance impact. A feed-forward controller is suggested in this work to get rid of the disturbance effect.

KEYWORDS

Robot, tracking, path geometry, geometric control, predictive control, feed-forward controller

1. INTRODUCTION

A PID controller modulation is used to allow a robot to move around a track while remaining in its lane by calculating the velocity and steering angle in proportion to the lateral distance between the robot and the reference trajectory, also known as the error of the cross track [1]. Although some of the reviewed speed and steering automatic controllers in the literature have been implemented separately on production robots, effective speed and steering control across a range of speeds is required for the autonomous operation of mobile robots. [2]. Due to significant engine dynamics nonlinearities, we present a design and implementation for a simplified adaptive cruise control (ACC) and lane keeping assistance (LKA) for an internal electric motor that is controlled by the throttle and steering [3].

The lane keeping assistance system is a control system that helps a motorist keep their vehicle in a clearly marked highway lane while traveling safely. This system operates when a vehicle swerves from a lane, and the LKA automatically corrects the steering without further driver input to bring the vehicle back into the lane [4]. On the other hand, a device known as the cruise control system regulates and

maintains the vehicle's speed at a predetermined point. The driver issues a signal of command. The cruise control system sends a control signal to the actuators that control the vehicle's throttle valve. This keeps the vehicle's speed constant and controls the fuel injection in the engine [5]. In this paper we are using these two concepts in a 4 wheeled robot, in a more streamlined model that combine the notions.

Industrial automation makes extensive use of independently driven mobile robots with four wheels [6]. We decided to treat the robot's kinematics for lateral movements like those of a bicycle, assuming that only the front wheel can be steered and that both the front and rear wheels are combined into unique wheels at the center of the front and rear axles. This was done to make the kinematic control study of this robot easier [7]. For longitudinal movements, we treated it like a two wheeled robot, assuming that there is neither lateral nor rolling slip [8] [9].

Most actuators have significant torque fluctuations at speeds below 13 m/s, a nonlinear phenomenon that causes significant variation in engine speed, crankshaft angular speed, and primarily the steering wheel rotation angle. Modern automobiles' automatic speed and steering control, also known as cruise control and lane keeping system, are typically recommended for use at speeds greater than 13 m/s and 16 m/s [10].

The road conditions are the focal disturbance effect that this paper looks at. The lack of a complete mathematical model that connects the two systems and the use of a single feedforward controller to reduce the nonlinear nature of the robot's dynamics and kinematics in both longitudinal and lateral movements, especially for the intended low speed range of 1-13 m/s, are the two main obstacles in designing an effective speed and steering controller. Because of these two factors, traditional control methods like the PID Controller are not easy to implement [11]. The fundamental idea behind feed-forward control is to measure significant disturbance variables and correct them before they disrupt the process in order to improve performance. Our system is most affected by road incline and steering effect, especially when the robot starts at zero speed, when this disturbance will be taken into account.

2. KINEMATIC MODELING

Generally, robots' motion can be modeled either by considering the geometric constraint that defines its motion, known as Kinematic Modeling or by considering all of the forces and moments acting on a it, known as Dynamic Modeling. In our case and as much we are working in low-speed range and the accelerations are not very significant, Kinematic Modeling is more than sufficient to capture the motion of our robot. In the simulation, the differentially driven mobile robot moves without lateral or longitudinal slippage as kinematic constraints.

Figure 1 represents the kinematic model for the robot used for this study.

2.1. Unicycle Representation

The robot's motion is constrained to move only forward to ensure the longitudinal movement for the robot, and because of this constraint that, wheels point in only one direction, it is called a nonholonomic constraint, which means that it restricts the rate of change of the robot's position. So, the robot can roll forward or backward and turn while rolling, but cannot move sideways directly, which means, without the wheels slipping.

Note $\{x_f, y_f\}$ a fixed coordinate system and $\{x, y\}$ a mobile frame linked to the robot.

Let $q^f = [x^f, y^f, \theta^f]^T$ be a point of the coordinate system $\{x_f, y_f\}$ and $q = [x, y, \theta]^T$ a landmark point for $\{x, y\}$.

Points q^f and q are related by the orthogonal matrix R(θ).

$$q^f = R(\theta)\, q$$

With:

$$R(\theta) = \begin{pmatrix} cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

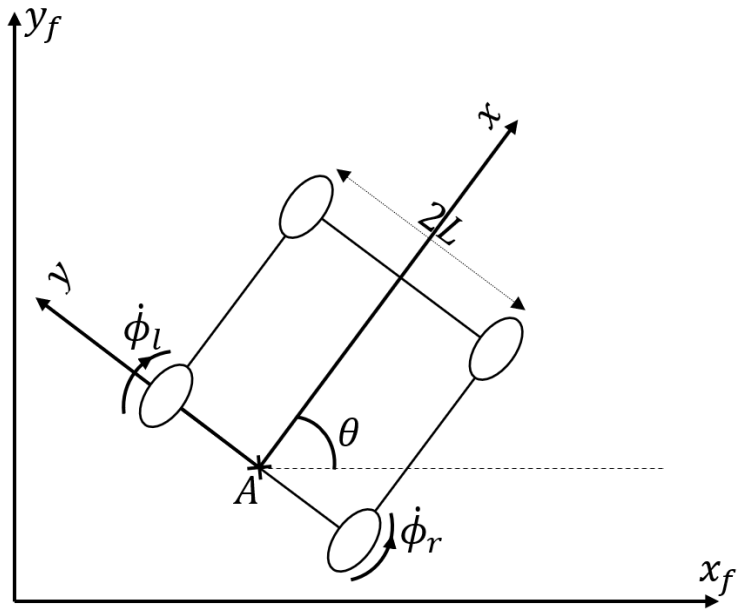

Figure 1 Kinematic Robot Model

- A : is the midpoint of the wheel axis.
- 2R : represents the diameter of the wheels.
- 2L : represents the robot width.
- $\dot{\phi}_r, \dot{\phi}_l$: represent the rotational velocity of the right and left wheels, respectively.
- θ : is the angle of orientation of the robot.

2.1.1. Kinematic Constraints

Two non-holonomic stresses derived from two assumptions characterize the robot's movement. A non-integrable constraint involving the time derivative of the robot's coordinates is known as a non-holonomic constraint [12]. A non-holonomic constraint is one in which a robot can instantly move forward or backward, but cannot move right or left without the wheels slipping. On the other hand, it is thought that the robot has holonomic behavior if each wheel can move forward and sideways.

2.1.2. Hypothesis

- Hypothesis 1: No Lateral Slip:

This constraint simply means that the robot can only move forward and backward, but not laterally. This means that the velocity of the robot associated with point A is zero along the lateral axis in the moving coordinate system, i.e. $\dot{y}_A = 0$

Using the rotation matrix $R(\theta)$, the expression of the robot velocity associated with point A in the fixed coordinate system is:

$$\begin{pmatrix} \dot{x}_A^f \\ \dot{y}_A^f \\ \dot{\theta}_A^f \end{pmatrix} = \begin{pmatrix} cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \dot{x}_A \\ 0 \\ \dot{\theta}_A \end{pmatrix}$$

Then:
$$\begin{cases} \dot{x}_A^f = \dot{x}_A \cdot \cos(\theta) \\ \dot{y}_A^f = \dot{x}_A \cdot \sin(\theta) \end{cases}$$

Thus,
$$-\dot{x}_A^f \cdot \sin(\theta) + \dot{y}_A^f \cdot \cos(\theta) = 0$$

- Hypothesis 2: No Rolling Slip:

As depicted in Figure 2 below, non-slip rolling stress indicates that each wheel maintains a point of contact with the ground.

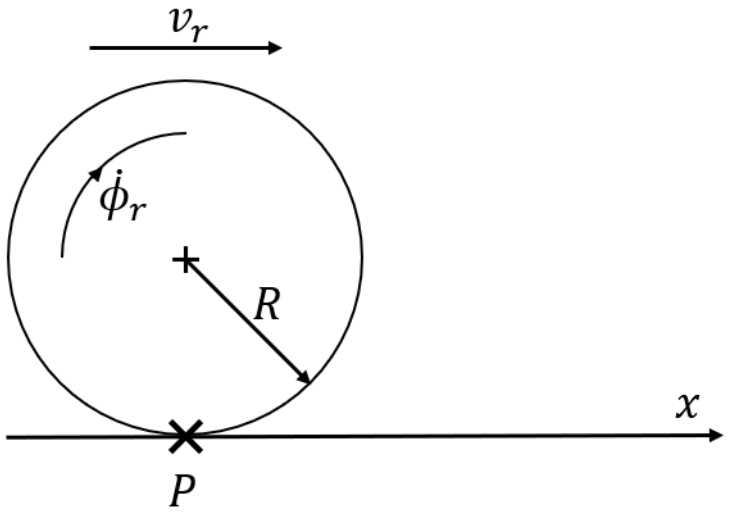

Figure 2 The right wheel

Thus, the linear velocity of each wheel of the robot at the point of contact P is given by:

$$v_{pr} = R\dot{\phi}_r$$

$$v_{pl} = R\dot{\phi}_l$$

Where v_{pl} is the linear velocity of the left wheel, and v_{pr} is the linear velocity of the right wheel.

As a function of the coordinates of point A, the expressions of generalized positions and generalized velocities in the fixed coordinate system are as follows:

Right wheel: $\begin{cases} x_{pr}^f = x_A + L\sin(\theta) \\ y_{pr}^f = y_A - L\cos(\theta) \end{cases} \Rightarrow \begin{cases} \dot{x}_{pr}^f = \dot{x}_A + L\dot{\theta}\cos(\theta) \\ \dot{y}_{pr}^f = \dot{y}_A + L\dot{\theta}\sin(\theta) \end{cases}$

Left wheel: $\begin{cases} x_{pl}^f = x_A + L\sin(\theta) \\ y_{pl}^f = y_A - L\cos(\theta) \end{cases} \Rightarrow \begin{cases} \dot{x}_{pl}^f = \dot{x}_A - L\dot{\theta}\cos(\theta) \\ \dot{y}_{pl}^f = \dot{y}_A - L\dot{\theta}\sin(\theta) \end{cases}$

Using the rotation matrix R(θ) and applying it to the right wheel we have:

$$\begin{pmatrix} \dot{x}_{pr}^f \\ \dot{y}_{pr}^f \\ \dot{\theta}^f \end{pmatrix} = \begin{pmatrix} cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \dot{x}_{pr} \\ \dot{y}_{pr} \\ \dot{\theta} \end{pmatrix}$$

With $\dot{y}_{pr} = 0$ means that the velocity at point P of the right wheel is zero (because no lateral slip). Thus

$$\begin{pmatrix} \dot{x}_{pr}^f \\ \dot{y}_{pr}^f \\ \dot{\theta}^f \end{pmatrix} = \begin{pmatrix} \dot{x}_{pr}cos(\theta) \\ \dot{x}_{pr}sin(\theta) \\ \dot{\theta} \end{pmatrix}$$

We have: $$v_{pr} = \dot{x}_{pr} = R\dot{\phi}_r$$

Thus:
$$\begin{cases} \dot{x}_{pr}^f \cos(\theta) = \dot{x}_{pr}\cos^2(\theta) & (a) \\ \dot{y}_{pr}^f sin(\theta) = \dot{x}_{pr}\sin^2(\theta) & (b) \end{cases}$$

By summing (a) and (b), we can form the equation system of two wheels:

$$\begin{cases} \dot{x}_{pr}^f \cos(\theta) + \dot{y}_{pr}^f sin(\theta) = R\dot{\phi}_r \\ \dot{x}_{pl}^f \cos(\theta) + \dot{y}_{pl}^f sin(\theta) = R\dot{\phi}_l \end{cases}$$

Hypothesis 1 and 2 and the previous equations produce the following constraints:

$$\begin{cases} -\dot{x}_A^f \cdot \sin(\theta) + \dot{y}_A^f \cdot \cos(\theta) = 0 \\ \dot{x}_{pr}^f \cos(\theta) + \dot{y}_{pr}^f sin(\theta) = R\dot{\phi}_r \\ \dot{x}_{pl}^f \cos(\theta) + \dot{y}_{pl}^f sin(\theta) = R\dot{\phi}_l \end{cases} \quad (1)$$

Then we can write:

$$A(q)\dot{q} = 0$$

$A(q)$ is the matrix of non-holonomic constraints given by:

$$A(q) = \begin{pmatrix} -\sin(\theta) & \cos(\theta) & 0 & 0 & 0 \\ \cos(\theta) & \sin(\theta) & L & -R & 0 \\ \cos(\theta) & \sin(\theta) & -L & 0 & -R \end{pmatrix}$$

$\dot{q}$ represents the derivative of the generalized coordinate q, given by $\dot{q} = \left[\dot{x}_A, \dot{y}_A, \dot{\theta}, \dot{\phi}_r, \dot{\phi}_l\right]^T$

Then we acquire that the expression of the linear velocities of the right and left wheels at the resource P is written in the accompanying structure:

$$\begin{cases} v_{pr} = v_A + L\dot{\theta} \\ v_{pl} = v_A - L\dot{\theta} \end{cases}$$

With v_A the velocity of the point A, v_{pr} is the velocity of the right wheel at point P and v_{pl} is the velocity of the left wheel at point P.

Putting:

$$\begin{cases} v = v_A \\ \dot{\theta} = \omega \end{cases} \quad \text{Et} \quad \begin{cases} v_{pr} = v_r \\ v_{pl} = v_l \end{cases}$$

As a function of the rotational velocities of the left wheel $\dot{\phi}_l$ and the right wheel $\dot{\phi}_r$, we can determine the expressions for the mobile robot's linear velocity v and angular velocity $\dot{\phi}_r$.

$$v = \frac{v_r + v_l}{2} = R\frac{(\dot{\phi}_r + \dot{\phi}_l)}{2}$$

$$\omega = \frac{v_r - v_l}{2L} = R\frac{\dot{\phi}_r - \dot{\phi}_l}{2L}$$

In the moving coordinate system, the coordinates of point A are:

$$\begin{cases} \dot{x}_A^r = v \\ \dot{y}_A^r = 0 \\ \dot{\theta}_A^r = \omega \end{cases} \quad (2)$$

These equations represent the kinematic model of the unicycle robot.

2.2. Bicycle Model

We combined the front and rear wheels axles into unique wheels at the center, reducing them into a pair of single wheels, that are assumed to be side-free, and only the front wheels are steerable, we can develop the kinematic classic bicycle model, that used to be surprisingly well at capturing vehicle motion in normal conditions, as a suitable control-oriented model for representing vehicles, because of its simplicity and adherence to the nonholonomic constraints.

The kinematic bicycle model reduces the left and right wheels to a pair of single wheels in the center of the front and rear axles, as shown in the Figure 3.

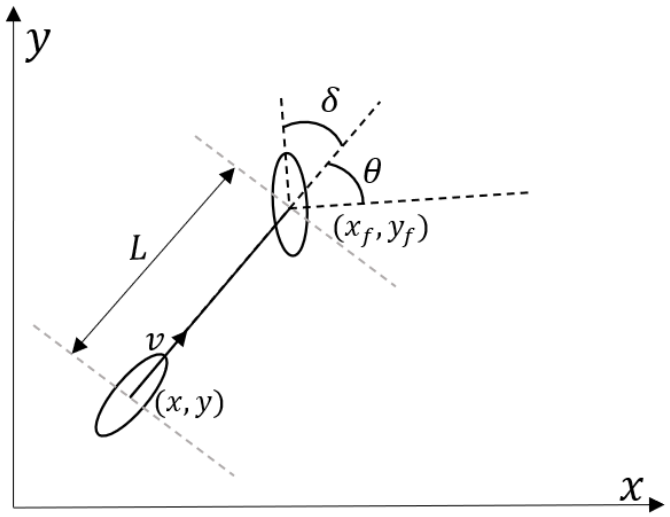

Figure 3 Bicycle model

The wheels are assumed to be side-free and only the front wheels are steerable [7]. By limiting the model to motion in a plane, the non-holonomic stress equations for the front and rear wheels are as follows:

$$\dot{x}_f \sin(\theta + \delta) - \dot{y}_f \cos(\theta + \delta) = 0$$

$$\dot{x} \sin(\theta) - \dot{y} \cos(\theta) = 0$$

To analyze the kinematics of the bicycle model, we selected a reference point on the robot at the center of the front axle (x_f, y_f), where θ is the orientation of the robot in the overall framework and δ is the steering angle in the body frame. We always assume that the robot operates in a 2D plane, which is represented by the overall coordinate of the rear wheel (x, y). We'll refer to the bicycle's length as L, which is measured between the axes of the two wheels, (x_f, y_f) can be expressed as follows,

$$\frac{d}{dt}\left(\dot{x}_f \sin(\theta + \delta) - \dot{y}_f \cos(\theta + \delta)\right) = 0$$

$$\frac{d(x + L\cos(\theta))}{\mathrm{d}t}\sin(\theta + \delta) - \frac{d(y + L\sin(\theta))}{\mathrm{d}t}\cos(\theta + \delta) = 0$$

$$\dot{x} \sin(\theta + \delta) - \dot{y} \cos(\theta + \delta) - L\dot{\theta}(\sin^2(\theta)\cos(\delta) + \cos^2(\theta)\cos(\delta)) = 0$$

$$\dot{x} \sin(\theta + \delta) - \dot{y} \cos(\theta + \delta) - L\dot{\theta}(\cos(\delta)) = 0$$

By the elimination of (x_f, y_f), the non-holonomy constraint on the rear wheel, is satisfied by $\dot{x}\cos(\theta)$ and $\dot{y}\sin(\theta)$ and any multiple scalars of these. This scalar corresponds to the longitudinal velocity v, such that,

$$\dot{x} = v\cos(\theta)$$

$$\dot{y} = v\sin(\theta)$$

Applying this to the stress on the front wheel gives a solution for $\dot{\theta}$,

$$\dot{\theta} = \frac{\dot{x} \sin(\theta + \delta) - \dot{y} \cos(\theta + \delta)}{L\cos(\delta)}$$

$$\dot{\theta} = \frac{v\cos(\theta)\,(\sin(\theta)\cos(\delta) + \cos(\theta)\sin(\delta))}{L\cos(\delta)} - \frac{v\sin(\theta)\,(\cos(\theta)\cos(\delta) - \sin(\theta)\sin(\delta))}{L\cos(\delta)}$$

$$\dot{\theta} = \frac{v(\tan(\delta))}{L}$$

The instantaneous radius of curvature R of the robot determined from v and leads to the previous introduction $\dot{\theta}$,

$$R = \frac{V}{\dot{\theta}}$$

$$\frac{v(\tan(\delta))}{L} = \frac{v}{R}$$

$$\tan(\delta) = \frac{L}{R}$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\delta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \\ \frac{\tan(\delta)}{L} \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \dot{\delta} \quad (3)$$

Where v and $\dot{\delta}$ are respectively the longitudinal velocity and the angular velocity of the steering wheel.

2.3. Dynamic Modeling

In this first part any depreciation is neglected [9]. As a model with one degree of freedom we are only interested in the translational movement of the robot.

The potential energy of the robot:

$$E_p = E_{P_P} + E_{Pe} = mgx + \frac{1}{2}kx^2$$

The kinematic energy of the robot:

$$E_c = \frac{1}{2}m\dot{x}^2$$

Applies the Lagrange formalism, we find by applying the Laplace transform:

$$ms^2x + mg + kx = 0$$

Then the transfer equation is of the form:

$$G[s] = x(s) = \frac{m_g/_k}{\frac{m}{k}s^2 + 1} \quad (4)$$

3. CONTROL

The control model for this four-wheeled robots is a framework that is used to understand and simulate locomotion the robot. The control model is based on the principles of kinematics and dynamics, which are used to describe movement in two-dimensional space.

3.1. Inverse Kinematic Model (IKM)

The inverse kinematic model allows to switch from operational velocities v and $\dot{\theta}$ to the velocities of each wheel and the following equations are accepted:

$$\dot{\phi}_d = \frac{v + L\omega}{R} \quad (5)$$

$$\dot{\phi}_g = \frac{v - L\omega}{R} \quad (6)$$

From these relationships we can build our inverse kinematic model in Simulink.

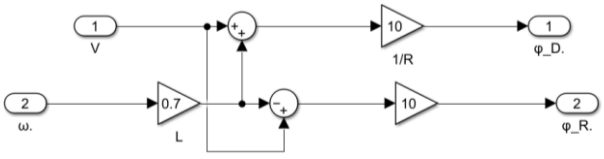

Figure 4 Inverse Kinematic Model

3.2. Actuator Modeling

We use DC motors, which convert electrical energy from direct current into mechanical energy, to drive the robot's wheels as actuators. DC motors are used to drive the wheels of our robot which are considered to be the actuators. The equations of a DC motor are the equation of:

The electrical circuit with the counter-electromotive force

$$L\frac{\mathrm{di}}{\mathrm{d}t} + r\mathrm{i} = u - v_b$$

The counter EMF is related to the speed

$$v_b = k_b\omega_m$$

The torque produced by the engine

$$\tau_m = k\mathrm{i}$$

Newton's second law for the motor shaft

$$J\ddot{\theta} + f\omega = \tau_m - \tau$$

The gear ratio

$$\omega_m = N\omega_R$$

The voltage u is used as the input of the motor, i is the armature current, the strength and inductance of the armature winding are respectively (r, L). We consider v_b the electromotive force, ω_m is the angular velocity of the robot and τ_m is the torque of the motor.

The torque constant and the electromotive force constant are respectively (k, k_b). J is the inertia of the engine and f is the damping coefficient. N is the reduction ratio

Table 1 shows the DC motor parameters and specifications that meet our robot design specifications

Table 1 DC motor parameters and specifications

Parameter	Value
$r\ (\Omega)$	0.8
$L\ (H)$	0.011
$J\ (K_g.m^2)$	0.2
$k\ (m.N/A)$	0.4
$k_b\ (V.s/rad)$	0.4
N	31.4

Below, in Figure 11, the Simulink model of the DC motor mathematical model.

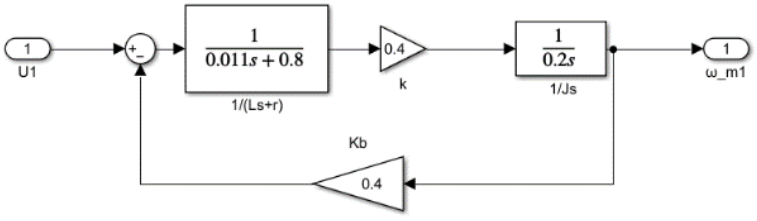

Figure 5 Modeling of motor actuators

3.3. Direct Kinematic Model (DKM)

The direct kinematic model allows us to know the Cartesian velocities (v_x, v_y, v_z) of the tool and the rotational velocity vector $(\omega_x, \omega_y, \omega_z)$ of the tool coordinate system as a function of the positions and angular velocities of the axes.

From the kinematic modeling we found that:

$$v = \dot{x}_r = \frac{v_d + v_g}{2} = R\frac{\dot{\phi}_d + \dot{\phi}_g}{2}$$

$$\omega = \dot{\theta} = \frac{v_d - v_g}{2L} = R\frac{\dot{\phi}_d - \dot{\phi}_g}{2L}$$

From these equations we can build our direct kinematic model (DKM) under Simulink/Matlab

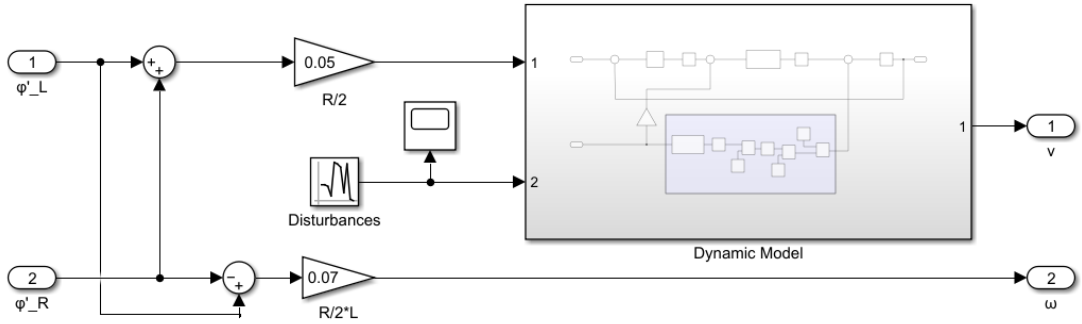

Figure 6 Direct kinematic model

3.4. Representation of Dynamic Disturbances

The basic concept of feed-forward control is to measure important disturbance variables and take corrective action before they disrupt the process in order to improve the performance result.

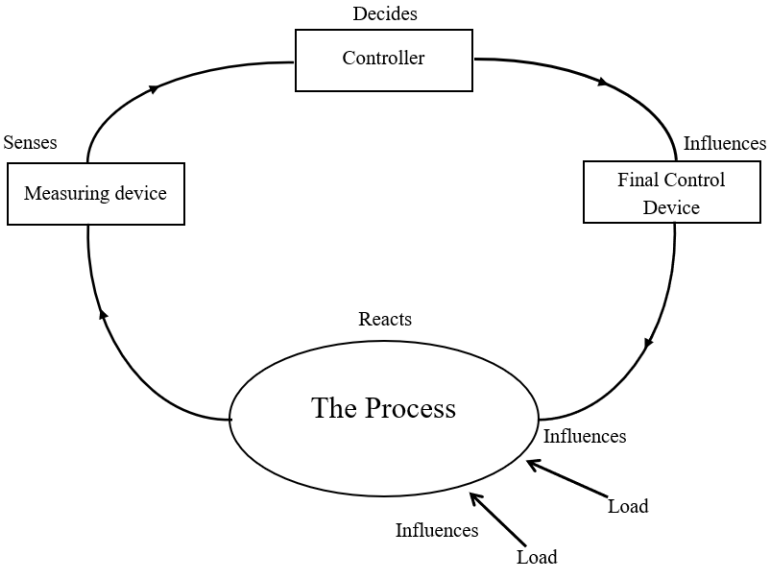

Figure 7 Advance control

Feedforward control has generally been considered to be an advanced control method, and if we break it down into different stages, we obtain a thorough and comprehensive analysis of the planning and control system, a regular review of the system for input variables and interrelationships for a consolidated result, a collect data on input variables and synchronize them with the developed system, regularly analyze variations in actual input data compared to planned inputs and assess their effect on

the expected result and a based on the analysis, take corrective actions to align planned and actual trajectories.

The main disturbance that acts on our system is the inclination of the road and the effect of the steering, especially when the robot starts at zero velocity, this disturbance is taken into account. Figure 5 shows an anticipatory control system, in which disturbances are measured and compensatory control actions are taken by the anticipatory controller.

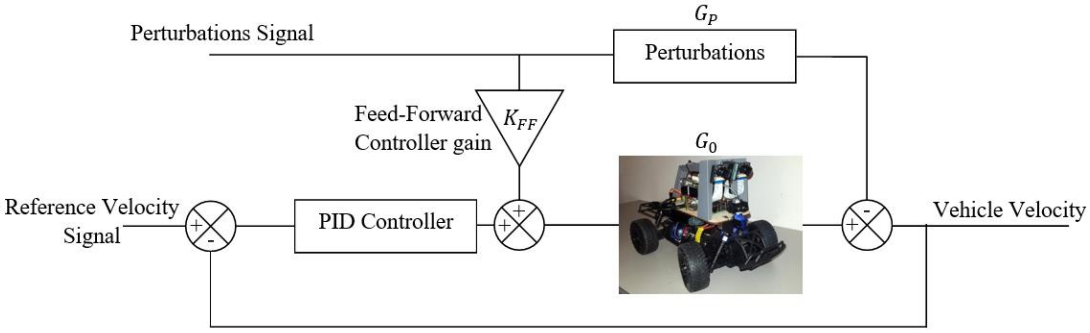

Figure 8 Block scheme for anticipation control

The controller adjusts the motor torque gain to increase or decrease the motor drive force F_d in response to the command signal, which is the reference velocity, and the speed sensor's feedback signal. The longitudinal elements of the vehicle as administered by Newton's low is

$$F_d = m\frac{\mathrm{d}v}{\mathrm{d}t} + F_a + F_g$$

$m\frac{\mathrm{d}v}{\mathrm{d}t}$ is the inertia force, F_a is the aerodynamic resistance, F_d the driving force of the engine and Fg is the resistance to rise (or the force of descent).

$$F_g = mg\,sin\,\theta$$

$$F_a = c_a(v - v_w)^2$$

v_w is the velocity of the wind thrust, M is the mass of the carriage robot, θ is the slope of the road and c_a is the coefficient of aerodynamic resistance.

Simplifying the model is the first step in planning the direct control of this system. Setting all initial conditions to zero is being considered.

$$\dot{v} = \frac{1}{m}(F_d - C_a v^2)$$

$$TL(\ddot{v}) = TL\left(\frac{1}{m}\cdot\left(\dot{F}_d - 2c_a v\dot{v}\right)\right)$$

$$\mathrm{X}s^3 = \frac{1}{m}\cdot(sF_d - 2c_a\cdot X^2 s^3)$$

$$\mathrm{X} = \frac{c_a}{m}\left(-1 + \sqrt{1 - \frac{mF_d}{sc_a}}\right)$$

Measure important disturbance variables and take corrective action before they disrupt the process to improve performance is the fundamental idea of advance control (FF). This disturbance will be taken into consideration, particularly when the robot starts at zero velocity.

The simulation section at the end of this article shows a feed-forward control of the direct-acting control system. In this section, disturbances are measured and compensatory command actions are taken by the direct-acting controller. The following is how deviations from controlled variables can be calculated:

$$\Delta v = G_p \cdot G_{FF} + G_0 = 0$$

$$G_{FF} = -G_p^{-1} \cdot G_0$$

With $$G_P = \frac{c_a}{m}\left(-1 + \sqrt{1 - \frac{mF_d}{sc_a}}\right) \quad (7)$$

Et $$G_0 = \frac{mg/_k}{\frac{m}{k}s^2+1} \quad (8)$$

So, we find:

$$G_{FF} = \left(\frac{m}{c_a\left(1-\sqrt{1-\frac{mF_d}{sc_a}}\right)}\right) \cdot \left(\frac{mg/_k}{\frac{m}{k}s^2+1}\right) \quad (9)$$

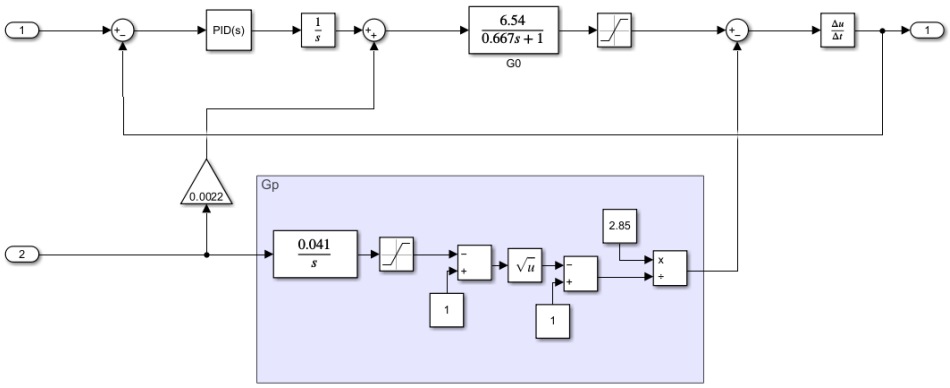

Figure 9 Dynamic robot model

3.6. Kinematic Model in the Robot Posture and the Lateral Control for Robot Orientation

We have modeled the equations $\dot{x}$, $\dot{y}$ et $\dot{\theta}$, note that γ the angle that makes the frame with the x-axis, and θ the steering angle for the robot. So we can find the generalized coordinate that are characterized by the longitudinal and lateral velocities coordinate axis, rotational and both wheels velocities and steering.

$$\dot{x} = vcos(\theta), \dot{y} = vsin(\theta), \dot{\theta} = \frac{v}{L}tan(\gamma)$$
$$\dot{\gamma} = \omega_m \quad (10)$$

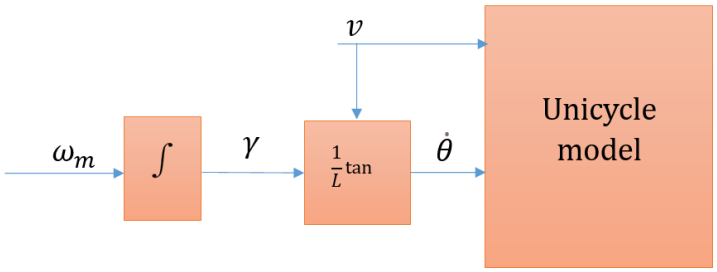

Figure 10 Block diagram corresponding to the unicycle model

In Simulink which allows us to have the velocities $\dot{x}$, $\dot{y}$ from v and θ.

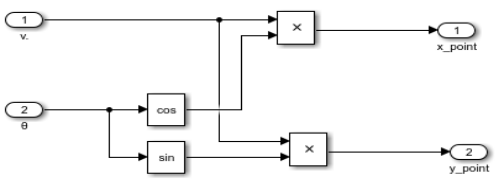

Figure 11 Kinematic model in robot posture

With assembling all pieces and putting them in place, now we can determine the complete model of our robot.

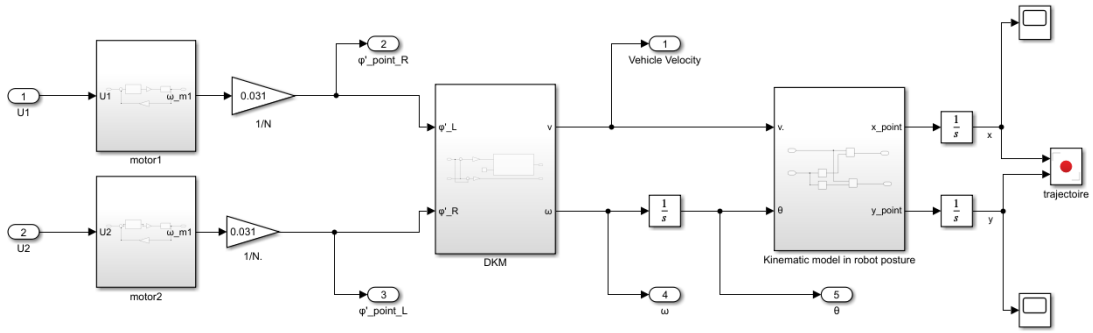

Figure 12 Unicycle Model

4. LONGITUDINAL AND LATERAL CONTROL OF THE ROBOT MOVEMENT

PID controller is a way to regulate an application to smoothly follow a value or a specific path. Although a full explanation of an application can be extensively complex, it summarized the math behind it in a super elegant and concise way and made the idea can easily extended to many real-world problems. In this chapter, we will follow the key structure of a PID controller.

4.1. Longitudinal Control

The goal of the PID controller is to reduce the amount of yaw error that occurs between the robot's heading and the centerline of the lane. As a cruise control system, longitudinal control is designed to keep the robot in the desired speed during flat, uphill, and downhill driving. sensing the robot speed and adjust its commands to match the desired speed set by the autonomous motion planning system. The numerical model of PID regulator given by the mixed form of the PID transfer function written in the form shown here.

$$u(t) = k_p \mathrm{e}(t) + k_I \int_0^t \mathrm{e}(t)\,\mathrm{d}t + k_D \dot{\mathrm{e}}(t)$$

4.2. Lateral control

Lateral controller is intended to keep the robot on the lane. If the vehicle veers out this lane, the controller must return the vehicle to its original position, and the controller chosen for this geometric trajectory tracking is pure tracking

A controller that only uses the reference trajectory and the geometry of the robot kinematics to follow a reference trajectory is known as a geometric path tracking controller. The anticipation point, which is a fixed distance on the reference trajectory in front of the robot, is used by the pure tracking controller as shown below. We must determine the steering angle that the robot must use to move in the direction of the anticipation point.

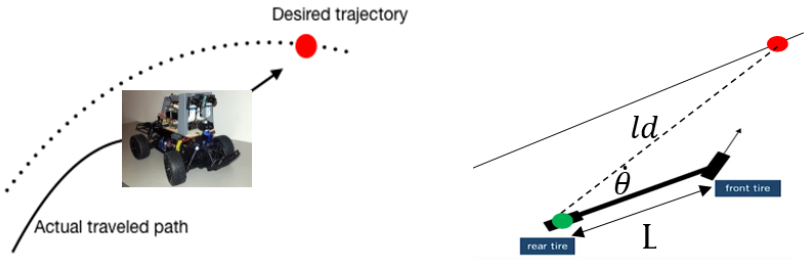

Figure 13 Pure Pursuit geometry

The concept that a reference point can be placed a predetermined distance ahead of the robot on the path is at the heart of the pure pursuit method. Additionally, the steering commands needed to intersect with

this point at a constant, calculable steering angle. The point continues to move forward, lowering the steering angle and gently guiding the robot toward the path as the robot turns to follow this curve.

In this method, the center of the rear wheel is used as a reference point on the robot. The target point is selected as the red point in the figure above. And the distance between the rear axle and the target point is denoted ld. Our goal is to get the robot to a correct angle and then get to that point. The figure of geometric relationship is therefore as follows, the angle between the heading of the robot body and the line of sight is designated by $\dot{\theta}$. Because the robot is a rigid body and moves around the circle. The instant center of rotation *(ICR)* of this circle is shown as follows and the radius is denoted by R.

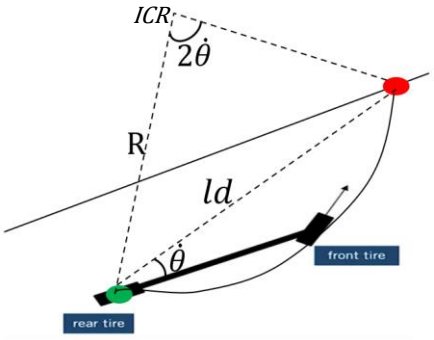

Figure 14 The angle between the carriage body heading and the line of sight

From the law of sines, we have

$$\frac{l_d}{\sin(2\dot{\theta})} = \frac{R}{\sin\left(\frac{\pi}{2} - \dot{\theta}\right)}$$

$$\frac{l_d}{2sin(\dot{\theta})\,cos(\dot{\theta})} = \frac{R}{cos(\dot{\theta})}$$

$$\frac{l_d}{sin(\dot{\theta})} = 2R$$

Thus

$$R = \frac{L}{tan(\delta)} \tag{11}$$

The steering angle δ can therefore be calculated as follows:

$$\delta = Arctan\left(\frac{2Lsin(\dot{\theta})}{l_d}\right) \tag{12}$$

The pure tracking controller is a simple control. It ignores dynamic forces acting on the robot and assumes that the no-slip condition holds at the wheels. But it's important to note that the fixed value of ld leads to a curvature controller that does not take into account the robot speed. This means that the selected steering angle would be the same regardless of whether the robot is going with 10m/s or 100m/s, leading to very different lateral accelerations, and one tuned for low speeds, would be dangerously aggressive at high speeds.

We make one more chance to this pure pursuit controller to get around this restriction. The look-ahead distance ld can be altered in response to the robot's speed. By defining it to rise in proportion to the forward speed of the robot. We arrive at the complete pure pursuit controller by removing this adjustment from the steering angle command equation. The controller determines the steering angle that will create an arc toward the look-ahead reference point and adjusts it so that the robot moves at a faster rate the further away it is from the reference point.

$$\delta = Arctan\left(\frac{2Lsin(\dot{\theta})}{K_{ff} \cdot v}\right) \tag{13}$$

The control law can be rewritten by scaling the anticipation distance in accordance with the robot's longitudinal velocity, making the adjustment simpler. This method of increasing the monitoring distance is common. Additionally, the monitoring distance is typically saturated to a minimum and a maximum. In our instance, these thresholds have been set at 3m and 1m, respectively. The experiments are consequently carried out on the lane change path.

4.3. Setting of gains 'Ziegler Nichols Method'

The following is a breakdown of how each of the PID terms gains and govern the response of the regulator which is denoted by $u(t)$ and affects the steering angle of the robot:

- Proportional component:

When used by itself to calculate the steering angle, the proportional term produces a steering angle that is proportional to the Cross Track Error. On the other hand, it fluctuates around the reference trajectory as a result. The robot's rate of oscillation (or overshoot) around the reference trajectory is determined by the proportional coefficient (k_P).

- Derivative component:

The proportional component's overshoot is minimized by the derivative component by employing a rate of change of error. The robot's overshoot, or oscillation amplitude, distance from the reference trajectory can be optimized using this derivative coefficient (k_D) term.

- Integral component:

Systematic bias causes errors in the steering angle over time, which could eventually, but not immediately, cause the robot to leave the track. This issue is corrected by the integral component. The integral coefficient (Ki), which has a significant impact on the performance as a whole, should be carefully optimized in small steps because this component affects the error over time.

Manually adjusting the k_P, k_I, and k_D numbers would require many trials and error before the robot could be made to go smoothly around the track. Another alternative is to automatically modify the parameters using an algorithm, such as twiddle. While this may be the optimal approach, it takes some time and work to construct the algorithm. By understanding the interdependencies of k_P, k_I, and k_D, we were able to select a tuning technique that was simpler to utilize than the prior two. With only two parameters chosen, we used the Ziegler-Nichols approach and the equations in Table 2 below to determine the k_P, k_I, and k_D terms.

The integral and derivative actions are first canceled. The proportional action is increased until the closed loop's output signal oscillates continuously. The maximum gain, also known as the critical gain, is then recorded as k_U, the signal's oscillation period is noted T_U. The following table is used to select the controller's k_P, T_I, and T_D parameters.

Table 2 Ziegler Nichols' method

Controller type	$\boldsymbol{k_P}$	$\boldsymbol{T_I}$	$\boldsymbol{T_D}$	$\boldsymbol{k_I}$	$\boldsymbol{k_D}$
P	$0.5k_U$				
PI	$0.45k_U$	$T_U/1.2$		$0.54k_U/T_U$	
PD	$0.8k_U$		$T_U/8$		$k_UT_U/10$
PID	$0.6k_U$	$T_U/2$	$T_U/8$	$1.2k_U/T_U$	$3k_UT_U/40$

According to the modeling equations we have

$$k_U = {mg}/{k} = 6.54 \text{ and } T_U = T = \frac{2\pi}{\omega} = 5.13\text{s} \text{ while } \omega = \sqrt{\frac{k}{m}}$$

5. SIMULATIONS AND INTERPRETATIONS

To demonstrate how the simulated system uses a PID controller with feed-forward to cancel the influence of the road disturbance, numerous experiments have been conducted to assess the system's realism. Figure 16 depicts the system signal with and without the PID controller. the signal would have sped up below the reference speed, but with the PID controller, it would have remained fairly close to the reference speed with a minor amount of error. Figure 17 shows the actual system signal during uphill and downhill driving while using a PID controller without feed-forward and with a feed-forward controller with a reference speed of 10 m/sec. Table 3 shows the controller parameters and specifications that meet our design specifications.

Table 3 Controller Parameters

Parameter	Value
k_P	3,94
k_I	1,52
k_D	2,51
k_{ff}	0.0022

5.1. Absence and Presence of Controllers

Figure 17 illustrates what happens when there are no controllers, 10 m/s as the reference velocity produced by a pulse generator, and 0.5 rad as the steering angle. The motors lose control, and the robot does not follow the reference. The findings, however, show us the significant rule of our controllers by proving the stability of the system, and in the presence of the controllers we can observe that the velocity and intended angle act as a reference for all circumstances.

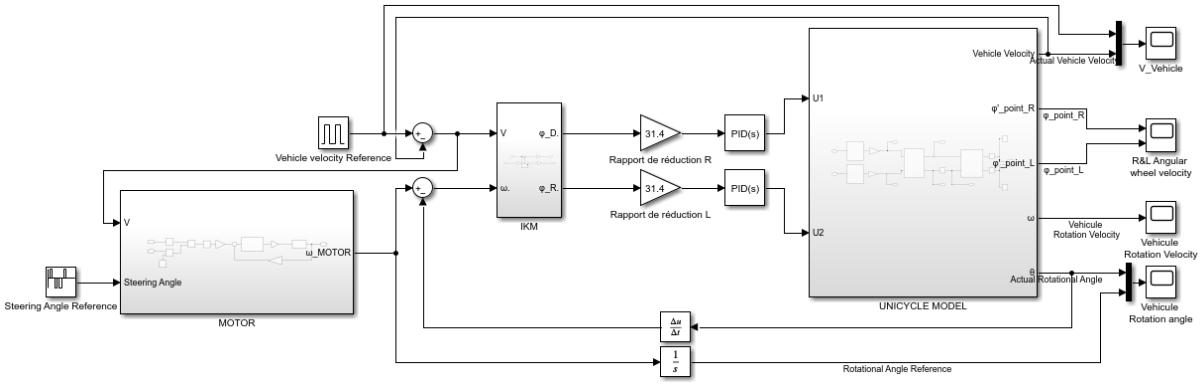

Figure 15 Presence of controllers

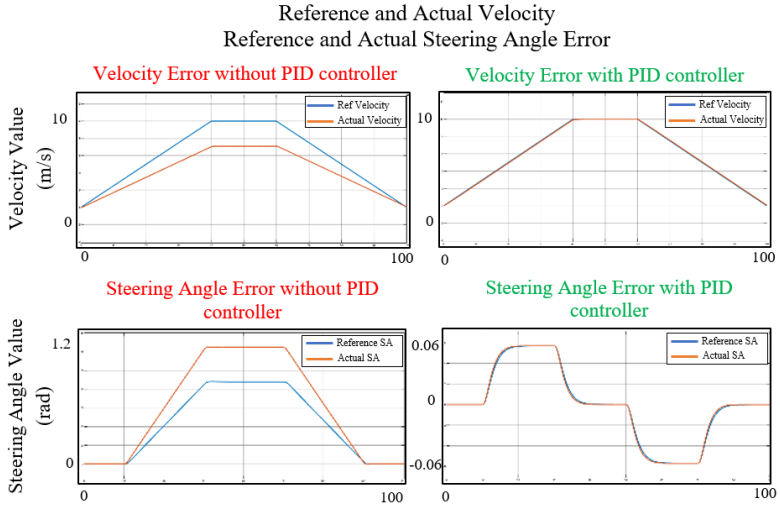

Figure 16 The real system with and without PID controller

5.2. Absence and presence of feedforward

The simulation in Figure 16 and 17 shows how the PID controller with feed-forward cancels the effect of the disturbance and shows the real system signal and trajectory during the robot displacement. The system without using a feed-forward system also shows how this disturbance can affect the robot's trajectory. To get the best results for the robot's ACC and LKA systems as a whole, numerous experiments have been carried out.

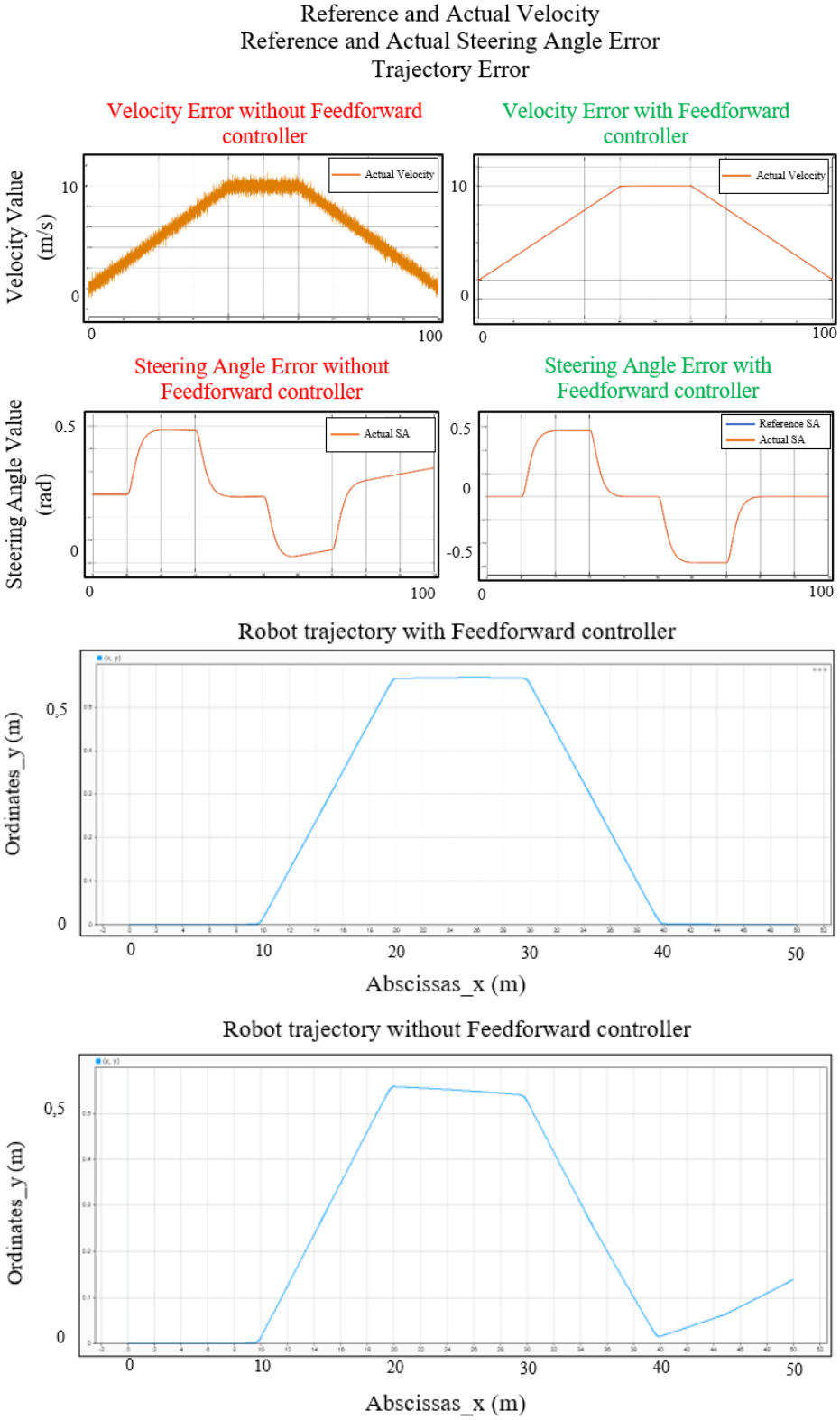

Figure 17 The real system with and without Feedforward controller

6. CONCLUSION AND FUTURE WORK

In conclusion, using track algorithms for the ego-vehicle, this study successfully produced the mathematical unified model for the simplified ACC and LKA systems. The system-satisfying parameters for the PID and Feed-forward controllers are estimated, and the impact of the incline road disturbance is simulated. However, all responses were contrasted in order to quantify the outcomes. However, in subsequent research, more advanced control systems will be used to test the model because it performed better than previous control strategies.

ACKNOWLEDGEMENTS

The authors would like to thank everyone, just everyone!

REFERENCES

[1] Abed, M.E., Ammar, H.H., Shalaby, R.: Steering control for autonomous vehicles using PID control with gradient descent tuning and behavioral cloning. IEEE (2020)

[2] Martinez-Garcia M, Kalawsky RS, Gordon T, Smith T, Meng Q, Flemisch F. Communication and interaction with semiautonomous ground vehicles by force control steering. IEEE Trans Cybern 2020..

[3] Perrier, M.J.R.; Louw, T.L.; Carsten, O. User-centred design evaluation of symbols for adaptive cruise control (ACC) and lane-keeping assistance (LKA). Cogn. Technol. Work 2021

[4] M. K. Diab, A. N. Abbas, H. H. Ammar and R. Shalaby, "Experimental Lane keeping assist for an autonomous vehicle based on optimal PID controller", 2nd Novel Intelligent and Leading Emerging Sciences Conference, 2020.

[5] Abdulnabi, A.R.: 'PID controller design for cruise control system using particle swarm optimization', Iraqi J. Comput. Inf., 2017.

[6] J. Liao, Z. Chen and B. Yao, "Model-based coordinated control of four-wheel independently driven skid steer mobile robot with wheel-ground interaction and wheel dynamics", IEEE Trans. Ind. Informat., 2019.

[7] MatuteJ.A. et al. Experimental validation of a kinematic bicycle model predictive control with lateral acceleration consideration IFAC-PapersOnLine (2019).

[8] J. Meng, A. Liu, Y. Yang, Z. Wu and Q. Xu, "Two-wheeled robot platform based on PID control", Proc. 5th Int. Conf. Inf. Sci. Control Eng. (ICISCE), Jul. 2018.

[9] S.-K. Kim and C. K. Ahn, "Self-tuning position-tracking controller for two-wheeled mobile balancing robots", IEEE Trans. Circuits Syst. II Exp. Briefs, Jun. 2019.

[10] Matute-Peaspan, J.A.; Zubizarreta-Pico, A.; Diaz-Briceno, S.E. A Vehicle Simulation Model and Automated Driving Features Validation for Low-Speed High Automation Applications. IEEE Trans. Intell. Transp. Syst. 2020.

[11] K. Sailan and K. D. Kuhnert, "Modeling and design of cruise control system with feedforward for all terrian vehicles", Comput. Sci. Inf. Technol., Nov. 2013.

[12] Huang, J., et al.: Adaptive output feedback tracking control of a nonholonomic mobile robot. Automatica 50(3), 821– 831 (2014)

Authors

Younes EL KOUDIA a PhD student in the Electrical Department at the National School of Applied sciences in Kenitra, Morocco. Degree in Mechanical and automated systems Engineering from the National School of Applied sciences Fez, Morocco 2021. In the same year, he started his PhD within in the Engineering of the Advanced Systems Laboratory researching on Self-Driving Cars and can be contacted by email: younes.elkoudia@uit.ac.ma

Tarik Jarou is University Professor at the National School of the Sciences Applied of the Ibn Tofail University, Kenitra, Morocco. He received Doctorat dregree (2008) in Electric Engineering from the Engineer School Mohammedia of the Mohamed V University , Rabat, Morocco. He is a member of the Advanced Systems Engineering Laboratory and Ex-member of LEECMS (Laboratory of Electric Engineering, Computing and Mathematical Sciences) of Ibn Tofail University, Kenitra, Morrocco. His main research area include the modelling, the control electronic and embedded systems for smart electric and cyberphysical system and their application fields in the automotive and aeronautical industry. He can be contacted at email: jaroutarik@hotmail.com.

Jawad ABDOUNI is a PhD student at the Advance Systems Engineering Laboratory of the National School of Applied Sciences of Ibn Tofail University. He obtained his engineering degree in electromechanics from the Ecole Nationale Supérieure des Mines in Rabat in 2017. His field of research mainly focused on trajectory planning algorithms in autonomous navigation systems. He can be contacted at email: j.abdouni@enim.ac.ma.

Sofia El Idrissi is a PhD student in the Advance Systems Engineering Laboratory at the National School of Applied Sciences within Ibn Tofail University. She received her engineering degree in embedded systems from the National School of Applied Sciences in 2019. Her field of research is mainly focused on autonomous vehicle's motion planning and control systems in an urban environment. She can be contacted at email: sofia.elidrissi@uit.ac.ma

M. Nasri Elmehdi received his engineering degree in electical engineering from the national school of applied sciences Agadir Morocco, in 2019. He is now Ph.D. student in the Advanced systems Engineering Laboratory, National School of Applied Sciences,Ibn Tofail University /Kenitra, Morocco. His main research interest includes control of Lithium batteries, Battery Management system, power control in electrical vehicles. He can be contacted at email: elmahdi.nasri1@uit.ac.ma