Received 28 July 2023, accepted 10 August 2023, date of publication 15 August 2023, date of current version 25 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3305678

RESEARCH ARTICLE

Hardware and Software Design for Mobile Robot's Navigation Over On-Road and Off-Road Curvature Paths

N. S. MANIKANDAN[1] **AND GANESAN KALIYAPERUMAL**[2]
[1]TIFAC-CORE in Automotive Infotronics, Vellore Institute of Technology, Vellore, Tamil Nadu 632014, India
[2]School of Information Technology and Engineering, Vellore Institute of Technology, Vellore, Tamil Nadu 632014, India

Corresponding author: Ganesan Kaliyaperumal (kganesan@vit.ac.in)

This work was supported by the Technology Information Forecasting and assessment Council-Centre of Relevance and Excellence (TIFAC-CORE) in Automotive Infotronics Centre (Sponsored by the Department of Science and Technology, Government of India), Vellore Institute of Technology, Vellore.

ABSTRACT Navigating a mobile robot through a curve path is a difficult task in both on-road and off-road navigation. This research provides a hardware and software platform to address this issue. Two types of research have been conducted in this paper. First, a hardware platform consisting of a Linear Actuator-based steering mechanism is implemented to hold the steering at the desired angle. Second, the curvature information from Google Maps path data is collected and used to improve the existing Pure Pursuit algorithm, which is known as the curve-aware Pure Pursuit (C-PP) algorithm, which navigates a mobile robot in both on-road and off-road (with scattered small obstacles) environments. The mobile robot is navigated using C-PP and steering is controlled by a Linear Actuator. The results taken in real-time and simulation show that the proposed algorithm for on-road and off-road environments has lower longitudinal and lateral RMSE values compared to the existing algorithms.

INDEX TERMS Mobile robot, curvature path, pure pursuit, linear actuator.

I. INTRODUCTION

Autonomous mobile robots have been used to transport items in on-road [1] and off-road [2] environments. They have also been applied across agricultural land [3] to do weeding, harvesting, and other tasks. The mobile robots may encounter numerous challenges on their journey. One of the primary issues is that it must follow the curvature path. Curves on the road, off-road bends, turns in a building corridor, and turns in an agricultural land are the examples of curvature. It has been reported that when a mobile robot operating on an agricultural terrain without taking into account the path curvature, encountered a 0.1947-meter mean absolute error (MAE), showing that the mobile robot is diverting from the reference path [4]. Off-roads are occasionally on uneven ground with a small number of obstacles, causing the mobile robot to slip [5] at an uncertain angle.

Many researchers have created mobile robot steering systems as well as wheel mechanisms that can move on rough off-road terrain and in agriculture. Some are skid steering [6], [7], extended bicycle models [8], [9], all-wheel independent steering [10], and so on. Many researchers have designed wheel-legged robots in recent years [10], [11], [12]. The majority of researchers continue to use the DC motor-driven steering mechanism for on-road [13], off-road [14], and agricultural [15], [16] navigation, which is steered by the front wheel of the mobile robot using the basic bicycle (kinematic) [17] model. It regularly encounters slippage when travelling on uneven off-road terrain or approaching a curve. As a result, numerous researchers have worked on controlling slippage [18] and speed controls over curvature [19].

You and Tsiotras [14] proposed a high-speed rally-race mobile robot that was controlled by a DC motor steering system and it moved on uneven off-road surfaces. A series of high-speed cornering demos were conducted after they employed a trajectory learning technique to identify a basic

The associate editor coordinating the review of this manuscript and approving it for publication was P. Venkata Krishna.

trajectory that accurately reflected the key elements of high-speed cornering steering. They separate the high-speed cornering trajectory into three stages: entry, sliding, and exit. Then, for each step, they create a switching-mode control employing various control strategies to induce high-speed cornering.

Regarding on-road DC motor-controlled steering, Ali and Mailah [20] proposed a comprehensive navigation algorithm that allows the mobile robot to autonomously steer on the road in a variety of conditions. The mobile robot can locate itself inside the road environment and choose a collision-free course from a predefined start location to a goal point using a cutting-edge technology called a laser simulator (LS). LS and sensor fusion data from a laser range finder, camera, and odometry are used to calculate path planning and roundabout (curvature) detection. Using the DC motor-controlled steering system, Manikandan and Ganesan [21] have proposed a novel navigation cum decision-making algorithm to tackle the uncertain direction of wandering pedestrians on Indian roads. Their algorithm can detect an unmarked lane on the road, detect obstacles, and track the direction of movement of the obstacles and provide stability to the mobile robot. Based on the proposed technique, their decision-making algorithm was able to choose the right action to steer the mobile robot in the right direction.

Other steering mechanisms for slippage and curvature speed control have been considered. A path-tracking algorithm was carried out by Lenain et al. [8] for off-road mobile robots with a skid-steer configuration. Contrary to common belief, the proposed work describes a relationship between the front and back wheels, with the proposed control laws intending to move the front and back axle centres in the same direction. After that, the robot is split into two separate subsystems that manage two lateral deviations from a predetermined path. Making use of the same skid-steered mobile robot, Liu et al. [22] propose an algorithm for estimating the position, orientation, velocity, and wheel slip of skid-steered mobile robots traversing off-road terrain. A Multi-Innovation Unscented Kalman Filter (MI-UKF) is specifically designed to fuse data from various sensors. To improve the accuracy of motion estimates, historical innovations generated along the time sequence are blended into the normal UKF update procedure. An asymmetric Instantaneous Centre of Rotation (ICR) kinematic revealing wheel slip is used in the proposed estimator's localization method.

A hybrid aerial/terrestrial robotic system was presented by Premachandra et al. [23]. The proposed robot system was created by adding a ground movement mechanism to a quadcopter. It utilizes the quadcopter's flying mechanism to accomplish ground movement. Additionally, they tackled the problem of obstacle avoidance in their research using autonomous control. Thus, they discovered that the hybrid aerial/terrestrial robot system was capable of providing autonomous ground movement and flight control along with autonomous obstacle avoidance during the ground movement.

In general, many off-road and agricultural lands have uneven paths and are scattered with small obstacles in large or small numbers. If the mobile robot approaches an off-road curve on its path with small obstacles, there will be slippage or a change of direction. Existing algorithms, discussed in the literature are not well tested using small obstacles scattered along their path. From the literature survey, the author understood that there is a necessity for a standalone system that is designed with necessary hardware and software to negotiate the off-road and on-road curvature paths effectively. We would like address this key issue in our research study.

Some researchers [24] have created a mobile robot steering system with a linear actuator [25] in their actuator unit. In this research, the authors have introduced the Linear actuator as a steering control mechanism in order to keep the steering system in a set position to avoid the slippage and change of direction while moving in uneven and curvature paths. Additionally, a curvature-aware path-tracking algorithm was included to ensure that the mobile robot travels at a constant speed based on the estimated curvature speed limits as while encountering curves on its path. This research concentrated on the mobile robot's linear actuator steering mechanism, as well as how a mobile robot is maneuvered in on-road and off-road terrains using a path tracking algorithm for curvy paths. Our primary contributions include the following:

- The proposed hardware component is the Linear Actuator, which is utilized as a mobile robot steering mechanism, and it must steer and hold the front wheel at the desired angle.
- The path-tracking algorithm uses the proposed software component's Curve-aware Pure Pursuit (C-PP) strategy to improve its comprehension of an approaching curve. It uses an established curve-finding technique to identify curves in the Google Maps data. Identify the extracted curve's properties, including curve radius, start and end locations, and curve speed limit. A list of the path's approaching curves has been loaded by the C-PP, because it is aware of curvature. Based on the curve speed limits, the C-PP algorithm needs to slow down the mobile robot.
- In order to show the practicality of the proposed system, real-time testing is also carried out in on- and off-road paths. We have chosen an off-road path that is uneven and also scattered with many small obstacles.

The rest of this work is organized as follows: Section II discusses the hardware platform, the proposed curve-aware steering architecture, the curve finding technique, upgrading Pure Pursuit to curve-aware Pure Pursuit (C-PP), and the kinematic model. The experimental setups, as well as the outcomes of the proposed linear actuator steering and C-PP in a real-time environment and simulation software, are covered in Section III. The conclusion is covered in Section IV.

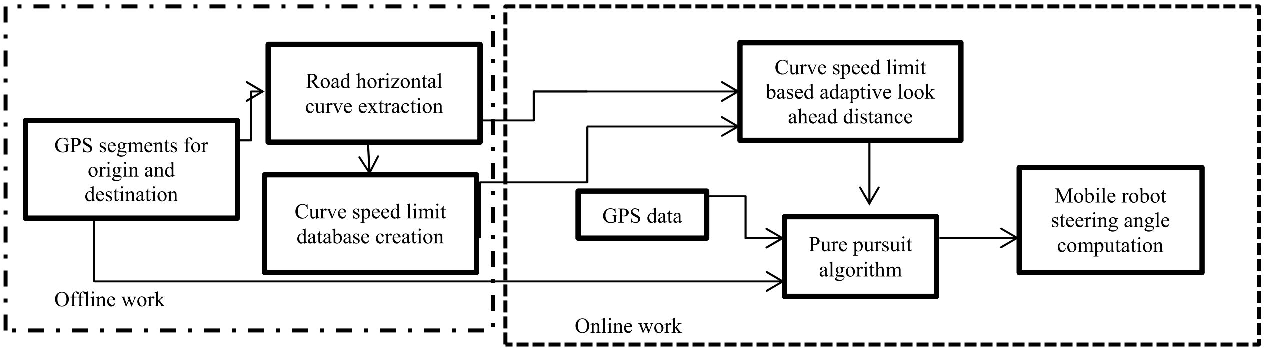

GPS segments for origin and destination
Road horizontal curve extraction
Curve speed limit database creation
Offline work
Curve speed limit based adaptive look ahead distance
GPS data
Pure pursuit algorithm
Mobile robot steering angle computation
Online work

FIGURE 1. Proposed model's block diagram.

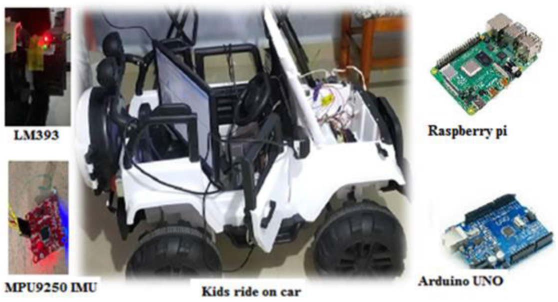


FIGURE 2. Proposed mobile robot with necessary hardware.

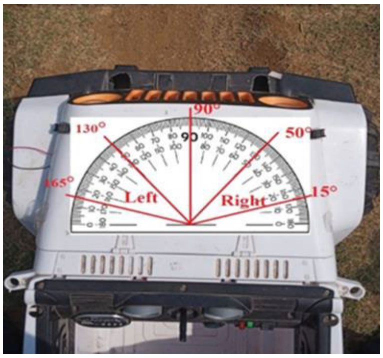


FIGURE 3. Mobile robot's steering angles.

II. PROPOSED MODEL

The software design of the proposed curve-aware path tracking algorithm is displayed in Figure 1. There are two types of processes in the system: online and offline. The end-user sets the mode before the device goes online and sets the place of origin and destination. The system collects the off-road/on-road Google Maps path waypoints of the shortest route of the selected origin and destination positions when the submit button is hit. The road curvature and its properties are extracted from the waypoint. Then it creates a database of speed limits for curves based on the radius of the curvature. This process is done offline. When working online, the speed limit for the curve and the extracted curve information are both input into the Pure Pursuit path tracking algorithm. The Pure Pursuit algorithm controls the mobile robot's speed and sets a dynamic look-ahead distance as it approaches the curve.

A. MOBILE ROBOT ARCHITECTURE

Figure 2 depicts the proposed architecture for mobile robot. DC motors in the front and rear have the following specifications: Voltage 15 V, Current 0.65 A, and Speed (N) 15,000 RPM. The front and rear DC motors are managed by the Arduino UNO controller. The DC motor requires a 7.2 V battery supply voltage and a current of 7 A. Instead of a motor drive, a 5V/10A two-channel relay is employed since it can supply more current than an engine drive can. Consequently, the engine is shielded from receiving excessively constant power. A different power source, provided by a Philips power bank with 20,000 mAh is available for the Raspberry Pi and Arduino UNO. The Raspberry Pi has integrated LM393 Hall Effect and MPU9250 IMU sensors. IMU and Hall Effect sensor data are inputs to the Raspberry Pi's embedded board, which outputs the steering angle.

The calculated steering angle data is provided to the controller for the Arduino UNO to be used for steering and forwarding commands. A mobile robot's steering angle must be separated into five sections and span from 15° to 165°. Absolute right angles are those between 15° and 45°, right angles between 45° and 75°, forward angles between 105° and 135°, and absolute left angles between 135° and 165°. Due to hardware control restrictions, as illustrated in Figure 3, the Mobile Robot is unable to transform between 15° and 0° (right) and 165° and 180° (left) angles (see Figure 3). Both the steering angle and forward motion are managed by the Arduino UNO. The Mobile Robot does not have any computation on its backside; hence the backward mobility facility has been reduced. Vehicles without payloads travel at 15.5 km/h. Depending on how much payload (10–15 kg) is being carried; the vehicle's payload speed can range from 12.5–9 km/h.

Using two LM393 Hall Effect sensors, the Mobile Robot's movement distance may be calculated. Given that the wheel's radius is 0.145 meters and its circumference is 0.9144 meters (3 feet), one Hall Effect Sensor can detect a rotation every

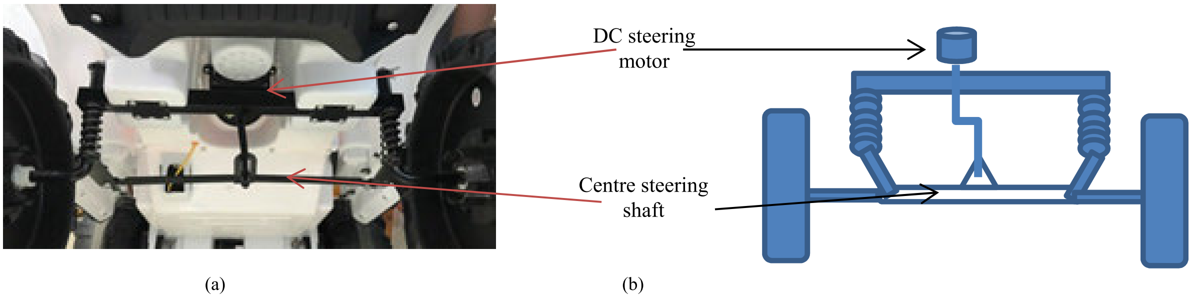


FIGURE 4. **(a) DC motor steering hardware (b) DC motor steering design outline.**

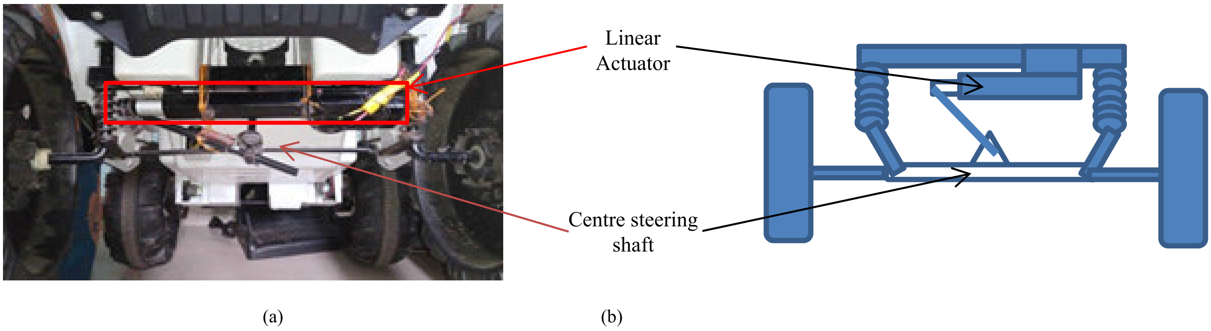


FIGURE 5. **(a) Linear actuator steering hardware (b) Linear actuator steering design outline.**

three feet, and two Hall Effect Sensors (pointing in opposite directions) can detect a rotation every 1.5 feet, allowing one to calculate the speed of the Mobile Robot. The Mobile Robot's MPU9250 IMU sensor provides information on its direction of movement. An accelerometer, gyroscope, and magnetometer are included in the sensor. The magnetometer has three axes: roll, yaw, and pitch. The Mobile Robot's direction is determined by its yaw value.

B. CURVE-AWARE STEERING DESIGN

Figure 4 depicts the actual mobile robot steering hardware. Figure 4 (b) clearly shows how the front DC motor controls the center steering shaft. The DC motor steering mechanism has been modified as a Linear Actuator controlled steering mechanism for travelling over curved paths, over uneven off-road, over tiny obstacles, and so on, as illustrated in Figure 5. In this redesign, a Linear Actuator is used and can hold the center steering shaft with a maximum load capacity of 110 N. The beginning point of the actuator rod has become entangled with the center portion of the center steering shaft (see Figure 5 (b)). If the actuator rod is released, the center steering shaft is automatically turned. The displacement of a 40 degrees steering angle is provided by a 10 mm actuator rod release.

Figure 6 depicts the relationship between steering wheel angle and linear actuator displacement. If the actuator is set to 0 mm, the steering angle is 15° (full right). Figure 7 (a) depicts the hardware for the full right turn of the front wheel; the actuator is 0 mm, as illustrated in the design sketch (see Figure 7 (b)). If the actuator reaches 30 mm, the steering angle is at 90° (center). Figure 8 (a) depicts the hardware

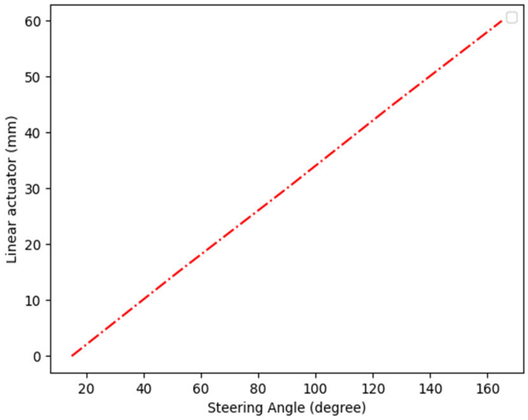


FIGURE 6. **Relationship between steering wheel angle and displacement of a linear actuator.**

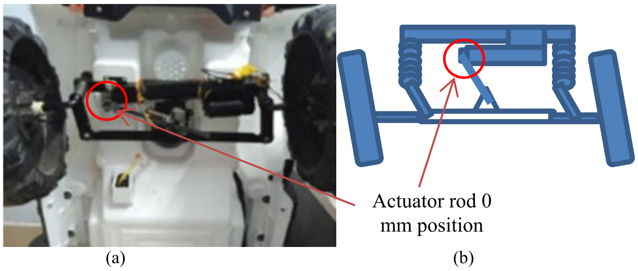


FIGURE 7. **(a) Front wheel right turn (b) Right turn steering displacement.**

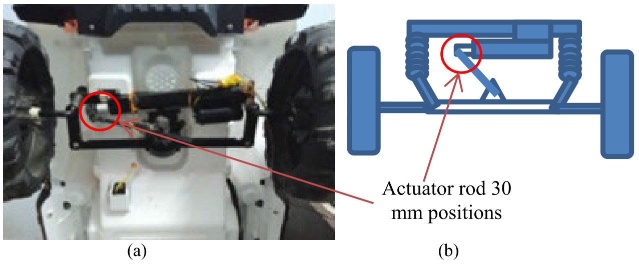


FIGURE 8. (a) Front wheel centre angle (b) Centre angle steering displacement.

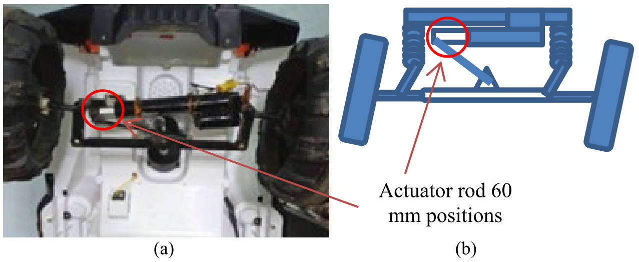


FIGURE 9. (a) Front wheel left turn (b) Left turn steering displacement.

center location of the front wheel; the actuator is 30 mm, as illustrated in the design sketch (see Figure 8 (b)). If the actuator is 60 mm in length, the steering angle is 165° (full left). Figure 9 (a) depicts the hardware for the full left turn of the front wheel, with the actuator at 60 mm as shown in the design outline (see Figure 9 (b)).

C. OFFLINE PROCESS

This section explains the offline mode implementation of the whole system. Once the origin and destination locations have been set, the system collects the GPS road segment data for the shortest route between the origin and destination location. Google Maps and OpenStreetMap will provide the direction API service for the chosen origin and destination locations; One can choose the shortest route among the selected places; we can even chose the most effective shortest path with least traffic congestion. From the GPS road segment data, one ought to extract the curve and then produce a curve speed limit database.

1) TRAJECTORY (PATH) CREATION

The trajectory must be established before navigating the mobile robot. In outdoor situations, particularly on and off roads, the open-street map and Google Maps are employed. If some off-road paths were not previously available in Google Maps and open-street maps, manually drawn lines were placed across a section of the map to construct this outdoor trajectory.

2) CURVE FINDING ALGORITHM AND CURVE SPEED LIMIT DATABASE CREATION

The Google Map-generated optimum route for the setup locations is provided as input to the current curve-finding technique [26]. The path in Figure 10(a) is made up of a number of waypoints, from W_1 to W_8, which have straight lines connecting them. The following formula uses the waypoint sequence to calculate the curve. Prior to finding out the path's radius of curvature, let's first establish the distance between two waypoints. Using the 'haversine' formula, one can determine the great-circle distance between two points.

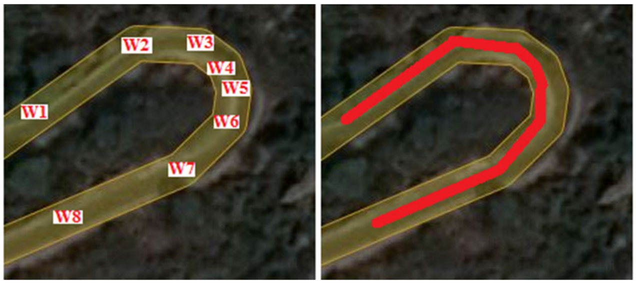


FIGURE 10. (a) Google maps waypoints (b) Identified curve.

$$ar = \sin^2\left(\frac{\Delta\varphi}{2}\right) + \cos\varphi_1 \times \cos\varphi_2 \times \sin^2\left(\frac{\Delta\lambda}{2}\right) \quad (1)$$

$$\mathrm{cr} = 2 \times \tan^{-1}\frac{\sqrt{ar}}{\sqrt{(1-ar)}} \quad (2)$$

$$\mathrm{dst} = \mathrm{Rad}\cdot\mathrm{cr} \quad (3)$$

where φ is latitude, λ is longitude, and Rad is the earth's radius (Rad = 6,371 km). Consider the distances from waypoints W_1 to W_2, W_2 to W_3, and W_1 to W_3 as l, m, and n, respectively. The radius is, therefore,

$$radius = \frac{(l \times m \times n)}{\sqrt{(l+m+n)\times(m+n-l)\times(n+l-m)\times(l+m-n)}} \quad (4)$$

Assume that the curve's largest radius is 200 meters. The radius of the three waypoints W_1, W_2, and W_3 in Figure 10(a), using Equation (4), is less than 200 meters because they appear to be on a curve. A straight line is defined as having a radius greater than 200 meters. The radius values of these three waypoints are preserved in the radius list and compared with those of another neighboring waypoint until the last waypoint on the path. The radius of six curves, Ri_1, Ri_2, Ri_3, Ri_4, Ri_5, and Ri_6, which are produced by the waypoints W_1 through W_8 in Figure 10 (a), were retained in the radius list. Finally, the waypoints' average curve radii (Ri_1 to Ri_6) are computed. Figure 10 (b) shows the path's identified curve from Figure 10(a).

The curve list contains the average curve radius in addition to the starting waypoint (W_1), ending waypoint (W_8), mid-waypoint (W_4), and average curve radius. Using the method outlined above, the path between the source and destination locations is examined, and the found curves are noted down along with their characteristics (start point, end point, mid-point, and average radius).

Following the references of articles specified by the Indian Road Congress (IRC) [26], the curve speed limit database was created. According to the IRC paper, Table 1 compares the intended speed limit to the curve's radius. The super-elevation of the curve dictates the range of the radius. Depending on the radius of the curve, Column 2 indicates the IRC speed limit for moving vehicles. The defined speed limits for the mobile robot are set in column 3 because they are not set due to the slow movement of the mobile robot. It must be warned five meters in advance if the Mobile Robot will be approaching the curve before it reaches the curve's starting point.

The Indian Road Congress (IRC) articles [26] were referenced in the compilation of the curve speed limit database. According to the IRC document, Table 1 shows the intended speed limit in relation to the radius of the curve. The super-elevation of the curve determines the radius range. Based on the radius of the curve, Column 2 indicates the IRC speed limit for moving vehicles. As a result, the established speed limit for the mobile robot is set in column 3. This is because the mobile robot goes slowly. The Mobile Robot must be warned five meters in advance if it will be approaching the curve before it reaches the curve's starting point.

TABLE 1. Restricted curve speed database.

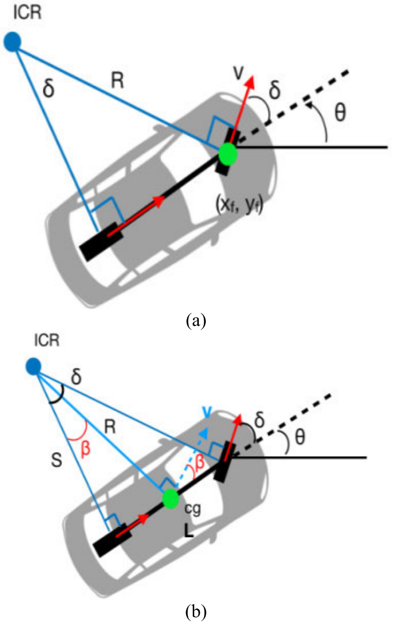
S.No	Curve radius in meter	IRC declared speed for vehicles against curve radius	Mobile robot speed set up	Mobile robot speed update location
1	Curve range in 50 to 100 meters	20 km/h	2 km/h	Five meters prior to beginning of the curve
2	Curve range in 70 to 150 meters	25 km/h	5 km/h	Five meters prior to beginning of the curve
3	Curve range in 100 to 200 meters	30 km/h	7 km/h	Five meters prior to beginning of the curve

D. ONLINE PROCESS

1) SIMPLIFIED VEHICLE MODEL (KINEMATIC)

The vehicle kinematics model supported by the Ackerman steering principle will convert the front-wheel steering vehicle into a two-wheel model assuming that the parking technique belongs to the low-speed movement process, disregarding the lateral slippery of the wheel, and also taking into account that the body is considered a rigid body. The geometric relationship is shown in Figure 11 (a). In Figure 11 (a), x and y are the coordinates of the middle of the rear shaft, θ is the heading angle of the vehicle, δ is the steering angle of the front wheel, and L is the distance of the body. When determining the kinematics constraint equation, utilise the centre of the rear axle as the reference point if the friction between the tyre and the ground is only rolling friction.

$$\dot{x} = v\, cos\theta \tag{5}$$



(a)

(b)

FIGURE 11. (a) Kinematic model (b) With slippery angle β.

$$\dot{y} = v \sin\theta \tag{6}$$

$$\dot{\theta} = v\, \tan\delta / L \tag{7}$$

where $R = L/\tan(\delta)$ so the steering angle δ can be calculated as

$$\delta = \arctan(\frac{2Lsin\alpha}{l_d}) \tag{8}$$

If the intended point is at the center of gravity (cg), the slip angle β is in Figure 11(b) below. The following equations can be found using the illustration in the image:

$$\dot{x} = v\cos(\beta + \theta) \tag{9}$$

$$\dot{y} = v\sin(\beta + \theta) \tag{10}$$

One need to compute R to get $\dot{\theta}$. As shown in Figure 11(b) above, then first compute S.

$$S = L/\tan(\delta) \tag{11}$$

Then one can use S and β angle to compute R.

$$R = S/\cos(\beta) = L/(\tan(\delta) \cdot \cos(\beta)) \tag{12}$$

R is found, then get $\dot{\theta}$ as follows:

$$\dot{\theta} = v/R = v \cdot \tan(\delta) \cdot \cos(\beta)/L \tag{13}$$

2) PURE PURSUIT ALGORITHM

The geometric path-tracking controller is pure pursuit. Any tracking controller that simply uses the geometry of the vehicle's kinematics and the reference path to track a reference path is known as a geometrical path tracking controller. The look-ahead point used by pure pursuit controllers is always a predetermined distance ahead of the vehicle on the reference path. The steering angle must be calculated in order for the

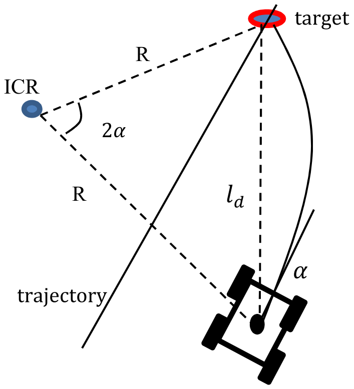


FIGURE 12. **Pure pursuit algorithm.**

vehicle to move to that location. Due to the reference point on the vehicle, this method uses the middle of the rear axle. The red point at the top of Figure 12 is selected as the target point, and l_d is used to indicate how far the rear axle is from the target point.

The target is to manoeuvre the vehicle at a precise angle to arrive there. As a result, Figure 12's geometric relationship is as follows: the angle between the vehicle's body heading and the look-ahead line is denoted by the symbol α. As a result, the vehicle has a hard body and moves around the circle. The radius of this circle is given as R, and the instantaneous centre of rotation (ICR) is shown as follows:

$$\frac{l_d}{\sin(2\alpha)} = \frac{R}{\sin(\frac{\pi}{2} - \alpha)} \tag{14}$$

$$\frac{l_d}{2\sin(\alpha)\cos(\alpha)} = \frac{R}{\cos(\alpha)} \tag{15}$$

$$\frac{l_d}{\sin(\alpha)} = 2R \tag{16}$$

3) CURVE-AWARE SPEED LIMITATION AND LOOK-AHEAD DISTANCE SETTINGS

The system navigates the mobile robot over the curve using curve-aware pure pursuit algorithm which is coded in Algorithm 1.

Algorithm 1 Curve-Aware Pure Pursuit Algorithm

```
Input: GPS waypoints, curve information, vehicle location,
curve speed limit, vehicle speed
Output: Controls vehicle speed and steering angle
    Function speedLimt_LookAhead(GPS_segments path,
            curve_Start, VCurrent_loc, $v_c$, $v_{current}$)
      if( vCurrent_loc near to curve_Start) then
        if ($v_{current} \geq v_c$) then
          Apply $a_{neg}$ until $v_{current} == v_c$
          if($v_{current} == v_c$) then
             $L = L_{min} + v$
             Pure_pursuit(GPS_segments path, $v_{current}$ , L)
end Function
```

where v_c, $v_{current}$ are curve speed and vehicle current velocity. Here the a_{neg} is vehicle deceleration speed, which is meatured in unit m/s^2. For the adaptive pure pursuit algorithm, the look-ahead distance L is calculated based on Equation (17)

$$L = L_{min} + v \left\{ \begin{array}{l} if\ SI \leq 1.01 v = v_c * 0.1 \\ if\ SI > 1.05\ v = v_c * 0.09 \\ if\ SI > 1.25\ v = v_c * 0.05 \end{array} \right\} \tag{17}$$

where L_{min} is defined to be 2 meters as a minimum look ahead distance.

4) RMSE METRICS

The RMSE (Root Mean Square Error) is used as a metric to measure the errors.

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(g_i - p_i)^2}{n}} \tag{18}$$

where n denotes the number of events, g_i denotes the ground truth, and p_i denotes the predicted value.

5) PERFORMANCE METRICS [21]

a: NAVIGATION TIME

This value represents the time it takes the navigation algorithm and steering control algorithm to steer the mobile robot towards the desired destination. It offers details about the processing speed and the navigational time slice.

b: TRAJECTORY LENGTH

This number gives the information about each algorithm's effectiveness, with higher steering and path-forwarding precision. It computes the distance the mobile robot had travelled to reach its destination in terms of meters. The two LM393 Hall Effect sensors were used to calculate the distance traveled in feet. The feet values were later converted in to meters.

c: SUCCESS RATE

This result, which is provided in percentage, demonstrates the overall effectiveness of the algorithm in terms of avoiding collisions and achieving the goal.

E. EXPERIMENTAL SETUP AND RESULTS

1) REAL-TIME EXPERIMENTAL TESTBED

This experiment was tested on-road (latitude: 12.985369, longitude: 79.169898), which is 50 meters in length with road curvatures in it, and off-road (latitude: 12.969936, longitude: 79.169475), which is 20 meters in length with off-road curvatures in and small obstacles in it. Figure 13 depicts the off-road (see Figure 13 (a)) terrain, which is littered with small impediments such as M-sand (see Figure 13 (b)), building waste (see Figure 13 (c)), and blue metal (see Figure 13 (d)).

Tables 2 and 3 and Figures 14 through 19 show the results and analyses of this experiment. The mobile robot was put through its tests with 24 different experiments on building waste, blue metal, and M-Stand. Mobile robot's steering is controlled by a DC steering motor with a 90-degree and

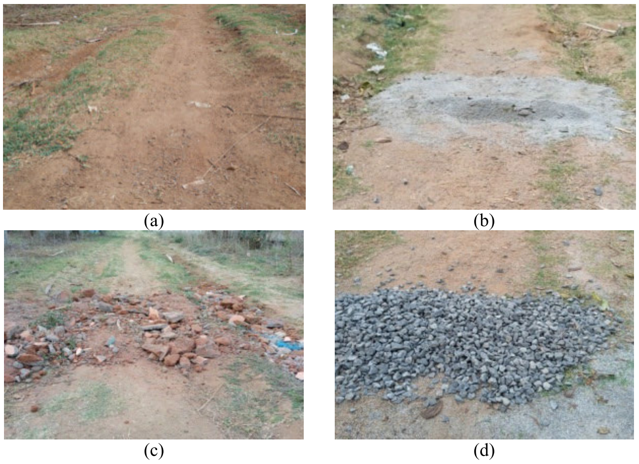


FIGURE 13. (a) Off-road (b) Off-road with M-Sand (c) Off-road with building waste (d) Off-road with blue metal.

165-degree setting at 2 km/h and 5km/h speed respectively, and mobile robot's steering is controlled by a linear actuator with a 90-degree and 165-degree setting at 2 km/h and 5 km/h speed respectively. These are two of the 24 types of studies we have done in the present research.

a: TEST-BED -1 (OVER BUILDING WASTE)

Table 2 shows a real-time comparison of a DC steering motor vs. a Linear Actuator steering control over small obstacles. Column 1 depicts a steering angle set up with a small obstacle, column 2 depicts DC motor steering controls with 2km/h and 5km/h moving speeds over the small obstacles, and column 3 depicts Linear Actuator steering controls with 2km/h and 5km/h moving speeds over the small obstacles. This is the steering angle value that was collected at the second (time) of the experiment in every column of the Figure in Table 2. Figures 14 through 19 depict the mobile robot's steering variation over several seconds when travelling over small obstacles at speeds of 2km/h and 5km/h.

The Mobile Robot's MPU9250 IMU sensor provides the information about the direction of movement of the robot. The metric RMSE error defined in section D.4 is used to evaluate the direction of movement. While calculating the RMSE error, n stands for the number of seconds (testbed time), g_i refers to setup direction (90 degrees or 165 degrees), and p_i is the actual direction of the mobile robot while moving during testbed. Table 3 illustrates the Root Mean Square Error (RMSE) for steering angle over small obstacles at two different movement speeds using two distinct steering hardware (a DC motor and a linear actuator). The results are analyzed in the following manner:

The off-road was scattered with small obstacles associated with building waste. Here the steering mechanism of a mobile robot is controlled by a DC motor. The steering control was tested with a 90 degrees (forward) and 165 degrees (left turn) setting.

The mobile robot was operated and tested over this building waste obstacle at two distinct speeds (2 and 5 km/h). As shown in Table 2, Column 2, Figures 14 and 15 (see DC 2 km/h and DC 5km/h), we notice that the mobile robot traveled at unpredictable direction. As a result, it has the largest RMSE Angle error in the 17-92 degree range. In the above setup, compared to a mobile robot with Linear Actuator steering control, the Linear Actuator steered mobile robot successfully moved over the small building waste obstacle while holding the steering at the desired angle. With the two different moving speeds, the RMSE Angle error for this Linear Actuator controlled steering mechanism is less, ranging from 0.5 to 2.5

b: TEST-BED -2 (OVER BLUE METALS)

The blue metals scattered off-road as the mobile robot moved over them. In a 90-degree arrangement, DC-controlled steering exhibits a number of angle differences, but the blue metal obstacles were successfully passed by Linear Actuator-controlled steering with two distinct speeds and no angle discrepancies (see Figure 16). In the other case, the mobile robot crossed the blue metal obstacles in a 165-degree turn (left turning direction). The DC-controlled steering made a 165-degree (see Figure 17) turns in an unpredictable direction. Hence, the overall (90 and 165 degrees) angle RMSE error rate of DC-controlled steering ranges from 2.5 to 77.8 degrees, but the Linear Actuator steering control had a proper hold in the steering and achieved a lower Angle RMSE error rate ranging from 0.6 to 2.5 degrees.

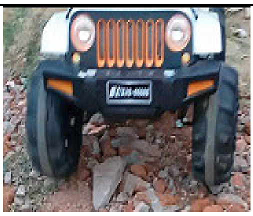
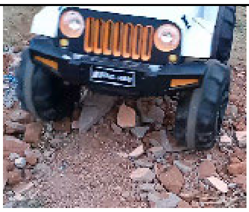
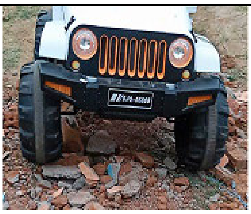
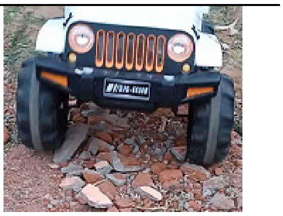
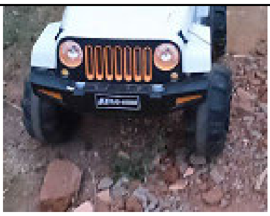
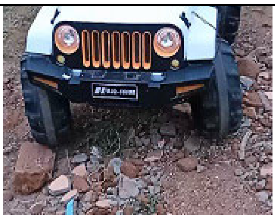
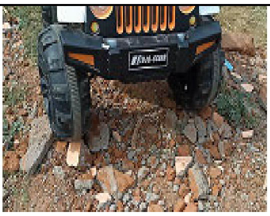
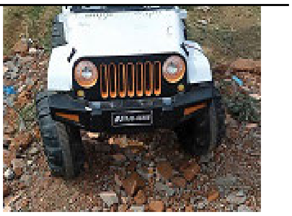
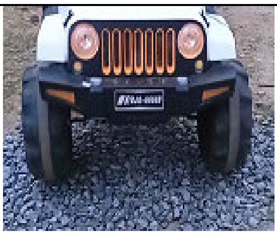
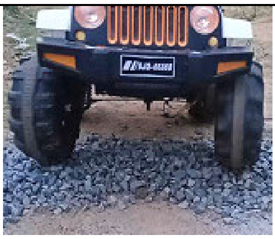
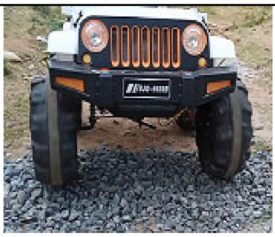
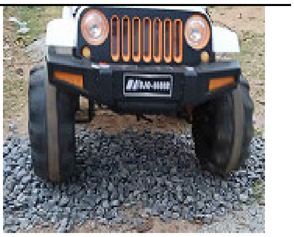
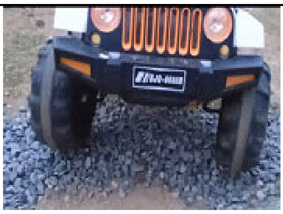
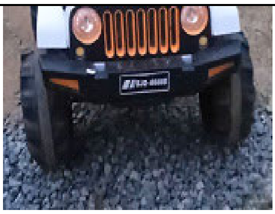
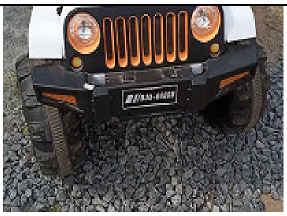
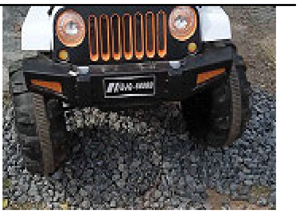
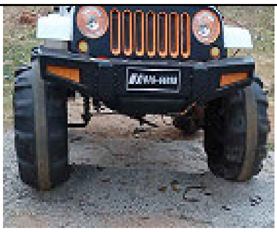
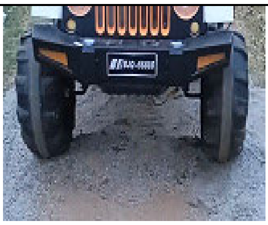
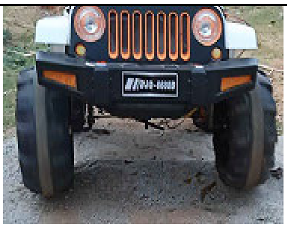
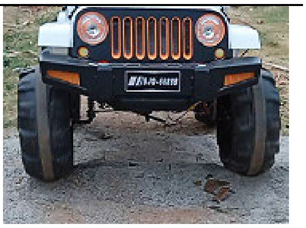
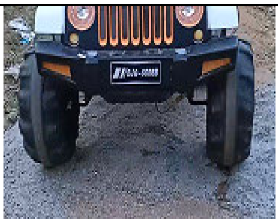
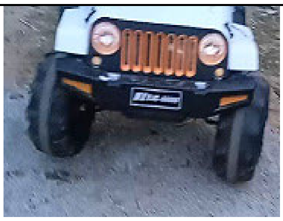
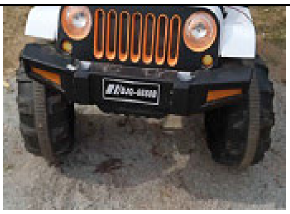
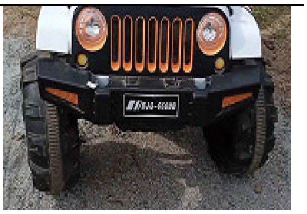
c: TEST-BED -3 (OVER M-SAND)

When the mobile robot passes through the M-Sand that has scattered across the off-road in a 90-degree (forward moving direction) arrangement, both the DC-controlled steering and the Linear Actuator-controlled steering with two different speeds effectively overcame the M-Sand obstacles without any angle discrepancies (see Figure 18). The DC-controlled steering made an uncertain turn (see Figure 19) when crossing the M-Sand obstacles in a 165-degree turn (left turning direction). Whereas the Linear Actuator steering control had a proper hold on the steering and moved along a set (165 degrees) direction. The overall RMSE angle error for DC-controlled steering is high, ranging from 0.5 to 110.2 degrees. But the Linear Actuator-controlled steering system achieved less steering Angle RMSE error rates ranging from 0.5 to 0.8 degrees.

d: TEST-BED -4 (OVER UNEVEN OFF-ROAD AND ON-ROAD)

The proposed curve-aware pure pursuit (CPP) algorithm, which is listed in Table 4, was used to navigate the mobile robot over on-road and off-road scenarios. The test results of a mobile robot (red line) over an off-road reference path (white line) that is navigated by the proposed C-PP were listed in columns 1 (DC steering) and 2 (Linear Actuator). The mobile robot's test results over the chosen roads are listed in columns 3 (DC steering) and 4 (Linear Actuator). Row 1 in Table 4 shows the real-time test image of the mobile robot, which is moving in a different direction compared

TABLE 2. DC steering motor vs. Linear Actuator steering.

	DC Steering Motor		Linear Actuator	
	2 km/h	5 km/h	2 km/h	5 km/h
Building waste with a Centre angle (90°)	 60°	 43°	 95°	 87°
Building waste with Left turn (165°)	 56°	 164°	 165°	 165°
Blue metal with a Centre angle (90°)	 90°	 50°	 94°	 93°
Blue metal with Left turn (165°)	 84°	 90°	 165°	 165°
M-Sand with Centre angle (90°)	 90°	 89°	 90°	 90°
M-Sand with Left turn (165°)	 86°	 43°	 165°	 165°

to the reference path. Row 2 shows the mobile robot's trajectory over Google Maps, as described in Section E.1. The mobile robot's two LM393 Hall Effect sensors and GPS are used to measure the distance travelled in feet. Row 3 shows the RMSE error of the completed trajectory. The RMSE error defined in Section D.4 is used to evaluate the mobile robot's moving trajectory vs. the ground truth path, which is described in Section C.1. The proposed C-PP algorithm recognizes path curvature and reduces mobile robot speed based on curvature speed limitations. Due to a lack of adequate steering wheel holding, the mobile robot using a DC motor steering mechanism has traveled in a more uncertain

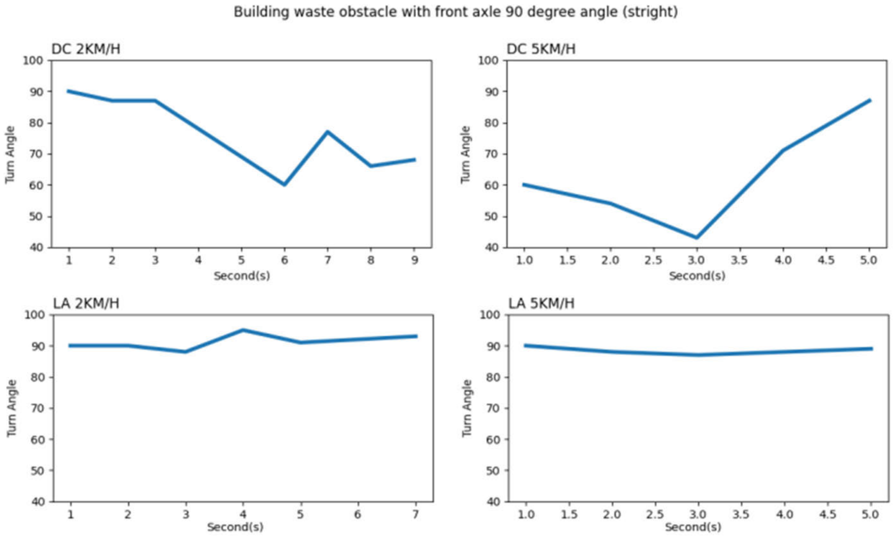


FIGURE 14. Over building waste with a 90-degree steering angle.

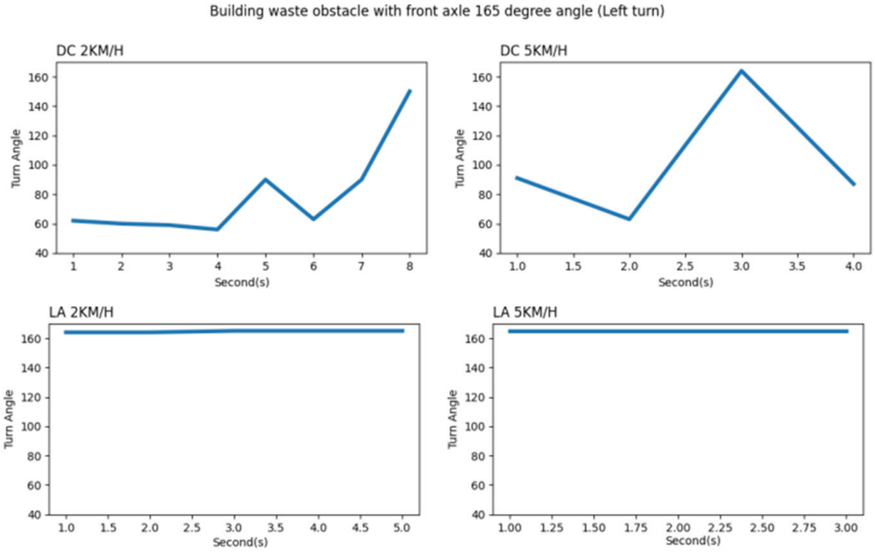


FIGURE 15. Over building waste with a 165-degree steering angle.

direction over off-road, resulting in a larger RMSE lateral error (0.000005) and longitudinal error (0.000011). The values 0.000005 and 0.000011 mean the error differences in 5 and 11 meters, respectively. In comparison, Linear Actuator steering holds the steering wheel in a direction navigated by the CPP algorithm, resulting in lower RMSE lateral (0.000001) and longitudinal errors (0.000001).

When testing mobile robots on the road, the DC steering motor initially steered the mobile robot in the right direction on a smooth road, but when the mobile robot moved over a small obstacle, the steering direction changed, and the mobile robot eventually moved over the edge of the road and fell into the roadside dig. But linear actuator-based steering had extremely little RMSE lateral and longitudinal error and steered in the appropriate direction. In this DC-base steering mechanism, the C-PP has more work maneuvering the steering during the curvature because it is not holding position at a right angle.

2) SIMULATION SETUP AND RESULTS

Only the CPP has undergone real-time testing. The Mobile Robot can be directed along the path using a variety of existing path-tracking algorithms. The proposed CPP as well as the current path-tracking technique were both put to the test using simulation. A Python-based program called Python-Robotics [27] was used for simulation.

This program both builds the road geometry and simulates Mobile Robot motion along the path. The proposed CPP is compared with four known path-tracking algorithms while they navigate the simulated Mobile Robot. Model Predictive Control (MPC), Linear Quadratic Regulator (LQR), Stanley, and Pure Pursuit are some of the path tracking algorithms now

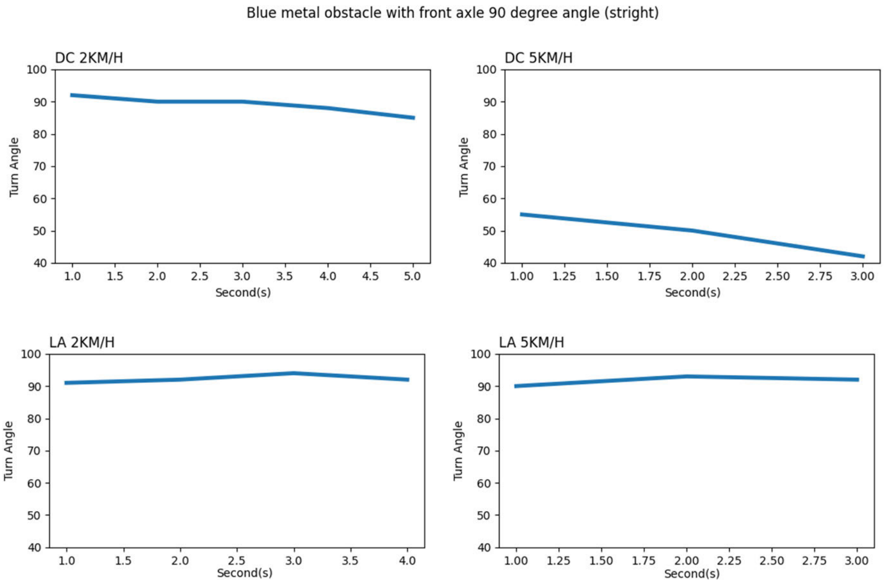


FIGURE 16. Over blue metal with a 90-degree steering angle.

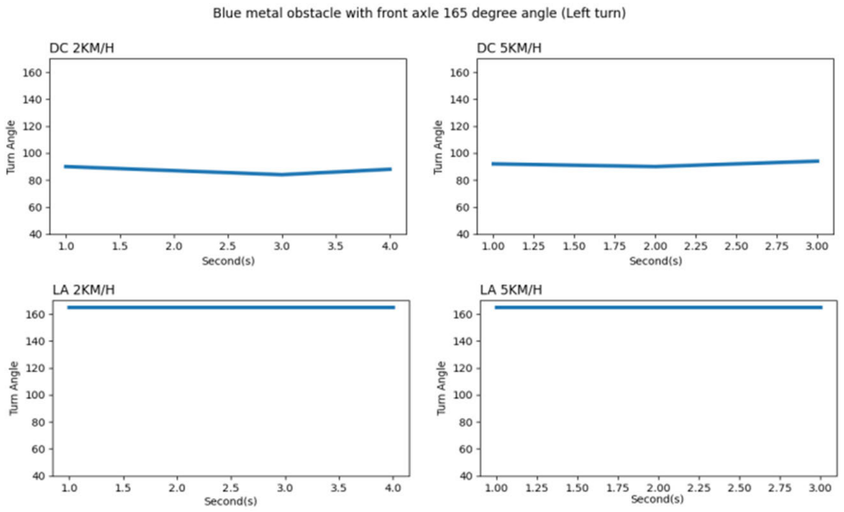


FIGURE 17. Over blue metal with a 165-degree steering angle.

in use. On the basis of speed vs. time, moving trajectory, and the RMSE (Root Mean Square Error) lateral and longitudinal locations, the results are compared. The RMSE defined in Section D.4 is used in this simulation of mobile robot navigation. Here the g_i is the simulated reference path, p_i is the path tracking algorithm's navigated path, and n is the simulation time. Both on-road and off-road navigation are choices in the simulation test bed.

a: OFF-ROAD TEST-BED

The path geometry for the off-road navigation simulation was designed with five curves and six straight lines, as shown in Figures 20 (DC) and 21 (LA).

The speed of the simulated mobile robot has been set to 25 km/h. The existing path-tracking algorithms (MPC, LQR, Stanely, and Pure Pursuit) as well as the proposed C-PP have been tested in this off-road scenario. Tables 5 and 6 provide the RMSE errors, while Figures 22 (DC) and 23 (LA) show the speed vs. time results.

While the C-PP with curve aware speed restriction drives the simulated mobile robot along the path at the curve speed limit, as shown in Figures 22 and 23 (red line), The existing path tracking algorithms propel the mobile robot down the path at a constant speed of 25 km/h.

Other algorithms finished the path in less than 55 seconds; however the proposed C-PP took 60 to 70 seconds. The simulated mobile robot was moved in the curvature path by the C-PP at a slower speed of 5 km/h, it caused the mobile robot's speed to gradually decrease before it approached the curve alert point's curvature. The simulated mobile robot

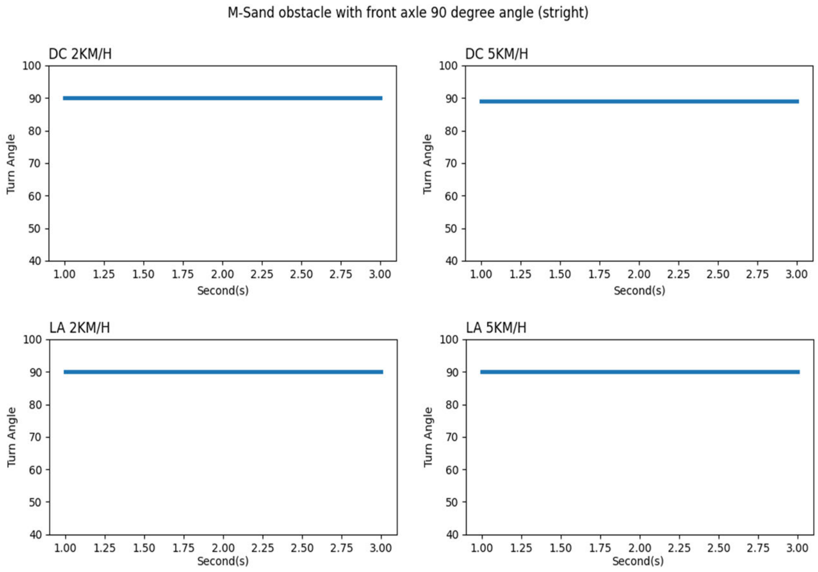


FIGURE 18. Over M-Stand with a 90-degree steering angle.

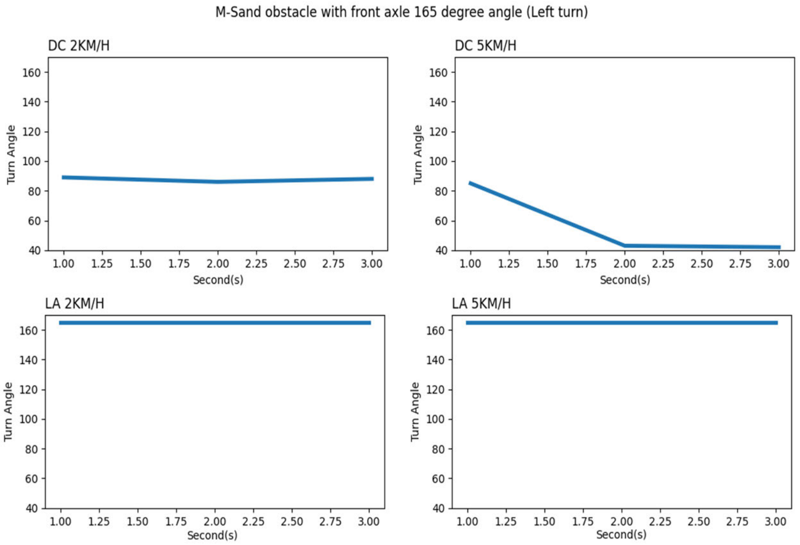


FIGURE 19. Over M-Stand with a 165-degree steering angle.

approaches the reference path using the C-PP strategy in both steering (DC and LA) mechanisms, which also has a lower RMSE (see Tables 5 and 6) lateral (using DC it is 0.52 meter, and using LA it is 0.19 meter) and longitudinal (using DC it is 0.56 meter, and using LA it is 0.21 meter) error than other algorithms. But the mobile robot navigated by C-PP using the LA steering mechanism encountered less RMSE error than using DC steering.

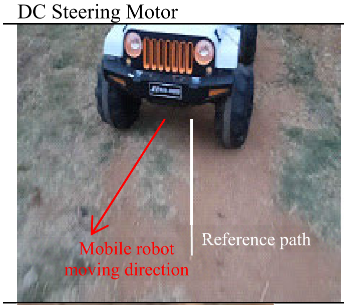
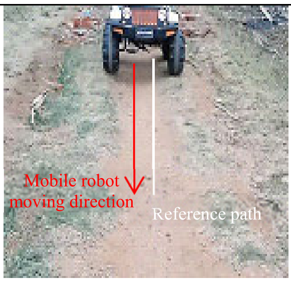
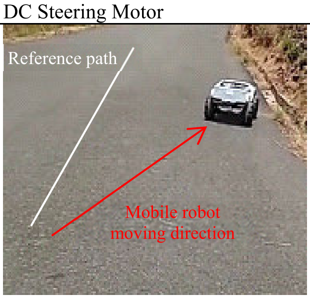
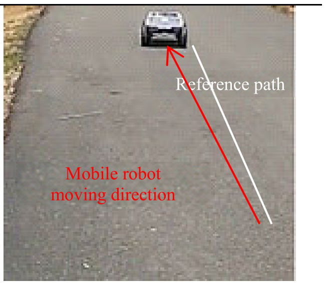
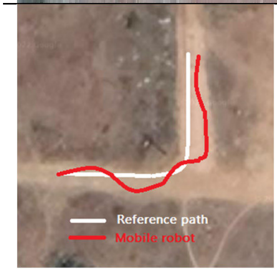
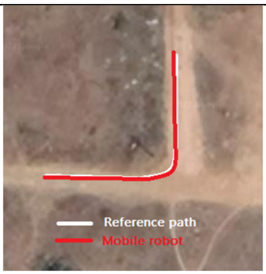
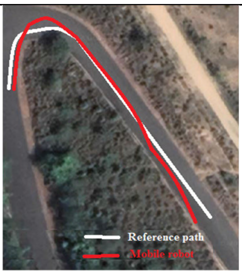
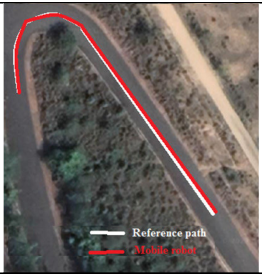
The slip angle error (β) explained in Section D.1 is combined with the steering angle (δ). Section E.1and Table 3 describes the real-time analysis of slip angle error (β) for DC motor-based steering (DC) and Linear Actuator-based steering (LA), where DC motor-based steering has a higher overall slip angle error (β) (building waste, blue metal, and m-stand) RMSE error rate ranging from 0.91 to 110 degrees (see table 3) than Linear Actuator based steering ranging from 0.5 to 2.5 degrees (see table 3). From the RMSE error range, the average RMSE degree error value is given as input to the slip angle error (β) parameter. Figures 20 and 21 display the reference path and the traces of each algorithm's trajectory tracking.

Pure Pursuit, Stanley, LQR, and MPC steered the simulated mobile robot out of the reference path in a DC motor-based steering scenario using a constant speed of 25 km/h and slip angle error (β). In this scenario, C-PP guided the simulated mobile robot closer to the reference path as it became aware of

TABLE 3. RMSE angle error (in degree).

S.No	Type	Building waste with Left turn (90°)	Building waste with a Centre angle (165°)	Blue metal with a Centre angle (90°)	Blue metal with Left turn (165°)	M-Sand with Centre angle (90°)	M-Sand with Left turn (165°)
1	DC with 2km/h in central angle	17.42	91.26	2.57	77.78	0.91	77.34
2	DC with 5km/h in central angle	30.9	74.1	41.35	73.02	1.03	110.17
3	LA with 2km/h in central angle	2.48	0.63	2.5	0.71	0.71	0.51
4	LA with 5km/h in central angle	1.9	0.55	2.08	0.62	0.82	0.63

TABLE 4. Curve-aware Pure Pursuit algorithm tested over off-road and on-road.

DC Steering Motor	Linear Actuator	DC Steering Motor	Linear Actuator
			
			
RMSE lateral error: 0.000005 RMSE longitudinal error: 0.000011	RMSE lateral error: 0.000001 RMSE longitudinal error: 0.000001	RMSE lateral error: 0.000002 RMSE longitudinal error: 0.000006	RMSE lateral error: 0.000001 RMSE longitudinal error: 0.000002

curvature and lowered speed in curvature, resulting in lower RMSE longitudinal and lateral errors, as shown in Table 5. Due to the constant speed of the simulated mobile robot with slip angle error (β), the Stanley was unable to travel nearer to the reference path, leading to a substantial RMSE error.

The LQR and MPC are navigated slightly nearer to the reference path in this case; the C-PP was navigated much nearer (see Figure 20 Zooms 1, 2, and 3) than other path tracking algorithms. Linear actuator-driven steering, as shown in Figure 21, has reduced the RMSE lateral and longitudinal errors (see Table 6) for all the chosen algorithms as well as C-PP due to its lowered slip angle error (β).

However, the Stanley algorithm has directed the simulated mobile robot somewhat outside of the reference path and has a slightly higher RMSE error than the other algorithms. The C-PP drove the simulated mobile robot substantially closer to the reference path in this scenario (see Figure 21 Zooms 1, 2, and 3) and has the lowest RMSE longitudinal and lateral error of all (see Table 6). The LQR RMSE error is closer to C-PP. However, in terms of computing complexity, the LQR exceeds the C-PP.

b: ON-ROAD TEST-BED

Figures 24 and 25 show the desired path geometry, which consists of two straight lines and one curve, for the on-road testing simulation. A speed of 25 km/h has been specified for the simulated mobile robot. The proposed C-PP and the existing path-tracking algorithm have both been evaluated in this on-road scenario. Figures 26 and 27 show the results for speed vs. time, and Tables 7 and 8 list the RMSE errors. As seen in Figures 26 and 27 (red line), the simulated mobile robot is navigated by the C-PP with a curve-aware speed limit at the curvature, whereas existing path tracking algorithms maintain the speed of the simulated mobile robot at 25 km/h throughout the path. While other algorithms finished the path in 25 seconds, the proposed C-PP needed 30 seconds. Before reaching the alerting point for the simulated mobile robot, the C-PP drove it at a speed of 20 km/h. As the C-PP followed

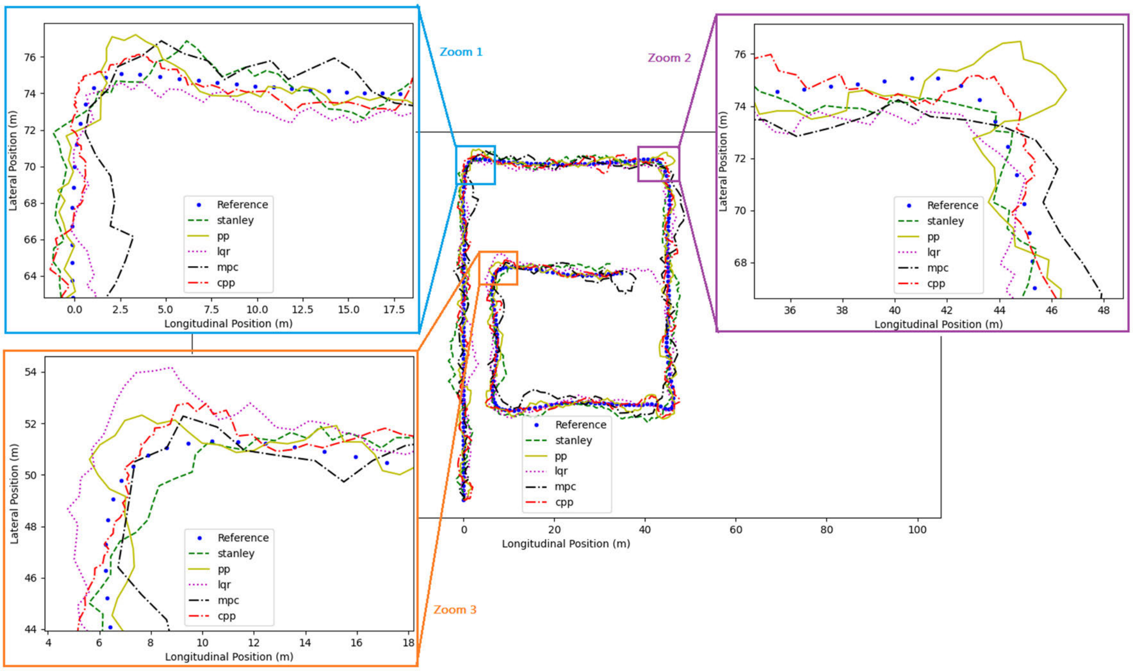


FIGURE 20. Trace for tracking an off-road path (DC steering).

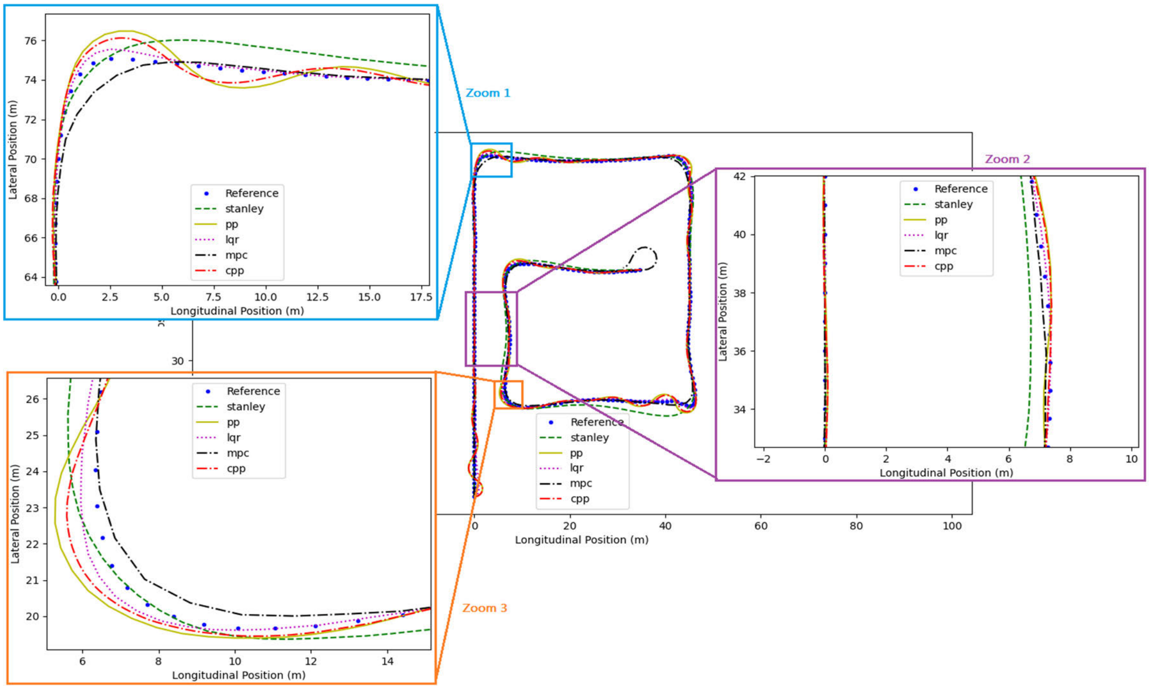


FIGURE 21. Trace for tracking an off-road path (LA steering)

the curvature path at a slower speed of 10 km/h, the speed of the simulated mobile robot steadily decreased. The C-PP approach has a smaller RMSE (Tables 7 and 8) error than the existing algorithms and consequently moves the simulated mobile robot a lot closer to the reference path.

The slip angle error (β) which is explained in Section D.1, is added to the steering angle (δ). where the DC motor steering has a higher slip angle error (β) than Linear Actuator-based steering. But compared to the off-road scenario, the on-road scenario has a lesser slip angle error (β).

TABLE 5. DC off-road.

S.no	Existing algorithms and proposed	lateral RMSE error (meter)	Longitudinal RMSE error (meter)
1	Pure Pursuit (PP)	2.68	0.83
2	Stanley	2.84	4.04
3	LQR	1.48	0.61
4	MPC	2.99	1.82
5	C-PP	0.52	0.56

TABLE 6. LA off-road.

S.no	Existing algorithms and proposed	lateral RMSE error (meter)	Longitudinal RMSE error (meter)
1	Pure Pursuit (PP)	0.29	0.40
2	Stanley	1.82	3.52
3	LQR	0.25	0.30
4	MPC	1.18	1.82
5	C-PP	0.19	0.21

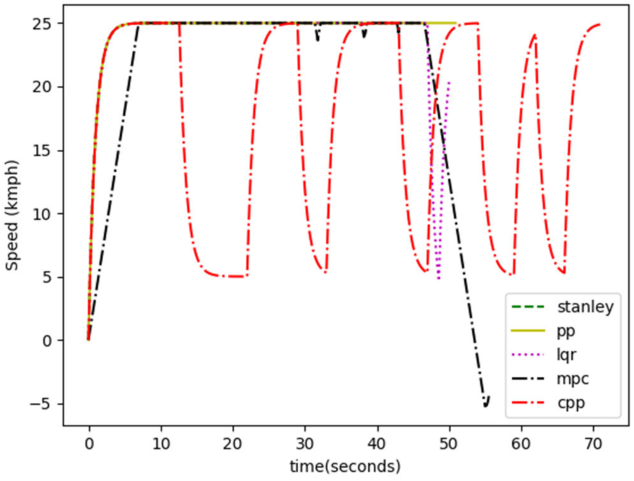


FIGURE 22. Time vs. speed result for off-road path (DC steering).

Figures 24 and 25 show the reference path as well as the traces for each algorithm's trajectory tracking. Here, pure pursuit and Stanley have higher RMSE lateral and longitudinal errors in DC and LA steering scenarios. The C-PP has a smaller RMSE error, but the LQR was nearing the C-PP. The C-PP has lower computational complexity than the LQR.

III. DISCUSSION

The metrics defined in sections D.4 and D.5 are used to compare the proposed algorithm with the existing algorithms, and the results are shown in Tables 9, 10, and 11. In terms of speed of mobile robot in curvature, navigation time, trajectory length, and success rate, the proposed algorithm is compared with the existing algorithms in both on-road and off-road environments and the results are shown in tables 9 and 10, respectively.

In terms of steering type, purpose, curvature detection method, small obstacles on its path, and slip angle error rate, The proposed algorithm is compared with existing methods in Table 11. According to Ali and Mailah [20], and Manikandan and Ganesan [21], a camera mounted on the mobile robot with DC motor-driven steering was utilized to extract road features [21], and the mobile robot successfully travelled along the on-road location (described in Section E.1), as illustrated in Figures 28 (a) and (b). The mobile robot, on the other hand, struggled to move through the intersection (latitude: 12.969705, longitude: 79.158631) shown in Figures 28 (c) and (d), because the region is larger and the road borders are hard to locate. If the mobile robot was larger,

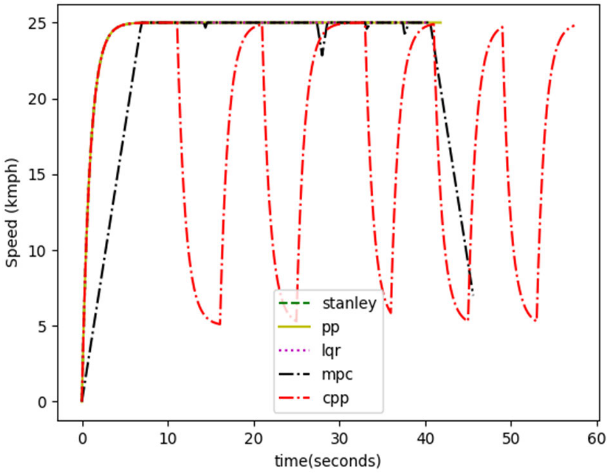


FIGURE 23. Time vs. speed result for off-road path (LA steering).

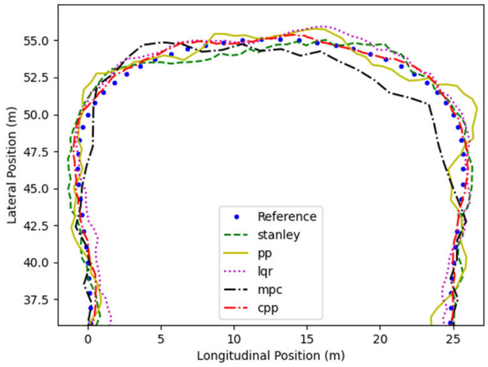


FIGURE 24. Trace for tracking an on-road path (DC steering).

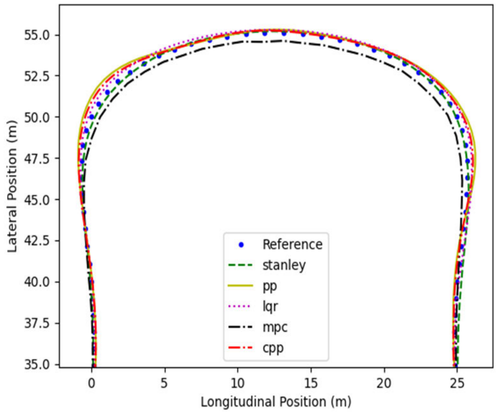


FIGURE 25. Trace for tracking an on-road path (LA steering).

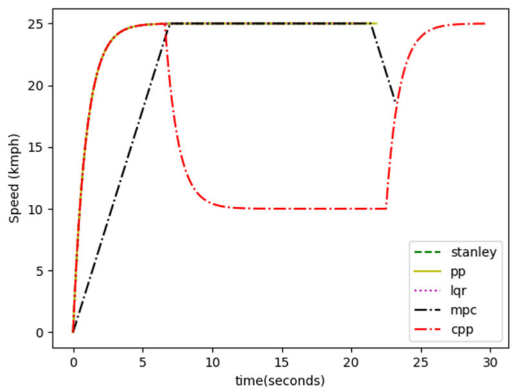


FIGURE 26. Time vs. speed result for on-road path (DC steering).

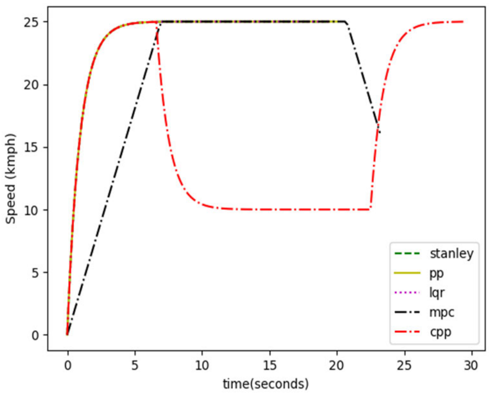


FIGURE 27. Time vs. speed result for on-road path (LA steering).

TABLE 7. DC on-road.

S.no	Existing algorithms and proposed	lateral RMSE error (meter)	Longitudinal RMSE error (meter)
1	Pure Pursuit (PP)	1.51	1.44
2	Stanley	1.10	2.28
3	LQR	0.33	0.39
4	MPC	0.51	1.50
5	C-PP	0.25	0.32

TABLE 8. LA on-road.

S.no	Existing algorithms and proposed	lateral RMSE error (meter)	Longitudinal RMSE error (meter)
1	Pure Pursuit (PP)	0.24	0.38
2	Stanley	0.81	1.15
3	LQR	0.20	0.37
4	MPC	0.40	2.58
5	C-PP	0.18	0.31

this problem might be avoided. The proposed algorithm eliminates road-wide intersection issues, as the curve-aware Pure Pursuit algorithm was navigating the mobile robot and it was also aware of the curvature speed limit. The proposed Linear actuator-controlled steering system with a curve-aware Pure Pursuit algorithm was compared with the existing Ali and Mailah [20] and Manikandan and Ganesan [21] algorithms in terms of performance metrics and RMSE metrics, as shown in Tables 9 and 11, respectively. In table 9, the existing algorithm navigated the mobile robot at maximum speed (10 km/h) in the curvature path, moved off the road, and fell in a dig before reaching the target. So it achieved an 85% of success rate, it consume higher navigation time, and was covering more distance when compared to the actual trajectory length (50 meters defined in Section E.1). But the proposed algorithm navigates the mobile robot with the curve-aware speed limit, which is described in Table 1, based on the radius of curvature (5 km/h), and the Linear actuator-controlled steering was holding the steering at the right angle. Hence, it has successfully reached the target. In table 11, we have shown the slip angle error. For the existing algorithm [20 & 21] tested on road, we notice 0.5–20 degrees of RMSE angle error and one meter of lateral and four meters of longitudinal error only.

TABLE 9. On-road testing.

S.no	Algorithms	Speed of mobile robot in curvature	Navigation time (in seconds)	Trajectory length (in meters)	Success rate (%)
1	Ali & Mailah [20], and Manikandan & Ganesan [21]	10 km/h	154	61	85
2	The proposed algorithm	5 km/h	117	52	100

TABLE 10. Off-road testing.

S.no	Algorithms	Speed of mobile robot in curvature	Navigation time (in seconds)	Trajectory length (in meters)	Success rate (%)
1	You & Tsiotras [14] and Manikandan & Ganesan [21]	7 km/h	89	48	65
2	The proposed algorithm	2 km/h	44	26	100

In the case of off-road environment, the existing algorithm of You and Tsiotras [14], and Manikandan and Ganesan [21], the mobile robot with DC motor-controlled steering system was successfully tested at high speed on the specified Lane detected [21] off-road path (Figure 29 (a)), which is described in Section E.1. When the mobile robot's speed is increased from 5 km/h to 15 km/h [14], it deviates from the off-road course (Figure 29 (b)) as it approaches a curve. Accroding to You & Tsiotras paper, the slippage angle error ranges from 10 to 20 degrees. But, According to the author, the slip

TABLE 11. Comparison between the proposed algorithms with existing ones.

S.no	Existing Algorithms	Steering type	Purpose	Curve detection algorithm	Small obstacles on the path	Slip angle error rate (in degrees)
1	Ref [20]	DC motor steering	On-road navigation	Roundabout (curvature) detection is determined using Laser Simulator (LS) and sensor fusion measurements from a laser range finder, camera, and odometry.	In ref [20] & ref [21] there were no small obstacles on road.	Slip angle error not analyzed in ref [20] & ref[21]. From author's on-road test. RMSE Angle error is 0.5 – 20 degrees. & RMSE trajectory error is lateral error: 1 meter longitudinal error: 4 meter
2	Ref [21]	DC motor steering	On-road navigation	No curve detection method.		
3	Ref [14]	DC motor steering	Off-road navigation	A series of high-speed cornering instances are followed by a trajectory learning technique to find a simple trajectory that encapsulates the key aspects of high-speed cornering steering.	In ref [14] the track is a 3.3 m wide flat clay surface with two straight lines connected by 180 degree with constant radius turns	10 – 20 degrees in ref [14]. From author's off-road test. RMSE Angle error is 1 – 109 degrees. & RMSE trajectory error is lateral error: 7 meter longitudinal error: 13 meter
4	Ref[8]	Skid steering	Off-road navigation	Not specified curve detection but proposed control laws aiming to follow the same path for the front and rear axle centers.	In ref [8] robot was navigated on the flat ground	30 – 40 degrees in ref [8].
5	Ref [22]	Skid steering	Off-road navigation	Using Multi-Innovation Unscented Kalman Filter (MI-UKF) approaches.	In ref [21] robot was navigated on the flat ground	0.5 – 40 degrees in ref [21]
6	Ref[23]	No separate steering mechanism	hybrid aerial/ terrestrial robot	No curve detection method.	Tested on floor and flight mode to avoid human holding obstacle (from ref[22])	Slip angle error not analyzed in ref[22]
7	Proposed algorithm	Linear Actuator steering	On-road, and off-road navigation	Using a curve-finding technique, the proposed Linear Actuator steering with the proposed C-PP slows down the mobile robot before it reaches the curve's starting point. Curves are extracted from the Google Maps path using curve finding technique. The mobile robot travels at different speeds.	Small obstacles scattered in uneven off-road and on road environment.	0.55 to 1.9 degrees

angle error of the existing algorithm [14], [21] which was tested on an off-road condition reported 1–109 degrees of RMSE angle error, 7 meters of lateral error, and 13 meters of longitudinal error.

The proposed algorithm was compared with the existing algorithms [8], [21] over an off-road testbed in terms of performance metrics, as shown in Table 10. The off-road location is defined in Section E.1, which is an uneven off-road with scattered small obstacles and 20 meters in length. Due to the presence of scattered small obstacles on the road, the mobile robot's existing navigation algorithm, which was controlled by a DC motor-driven steering mechanism, was unable to reach the destination (65% success rate), required more navigation time (89 seconds), and took a longer trajectory (48 meters), as shown in Table 10. Due to the DC-motor-controlled steering mechanism's inability to hold the steering in the right direction, the mobile robot traveled in a zigzag fashion.

But the proposed algorithm has properly held the steering in the right direction, navigated the mobile robot at curve aware speed over off-road curvatures, reached the target successfully at reasonable navigation time, and traveled closer to the ground truth trajectory.

Other skid-steering mechanisms proposed by Lenain et al. [8] and Liu et al. [22] have not been tested in real-time, although they employed different types of curvature-detecting approaches and, according to their paper, encountered slippage angle errors of up to 40 degrees.

In Premachandra et al.'s [23] work, a hybrid aerial/terrestrial robot system was used. The testbed was a flat surface with

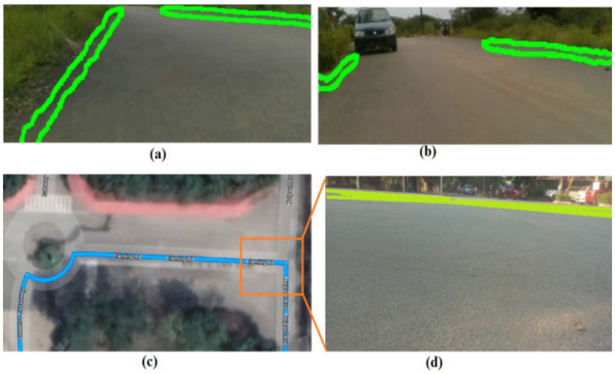


FIGURE 28. Vision-based on-road navigation (a) identification of road edges (b) obstacle (car) detection on road (c) road junction top view (d) mobile robot vision in on-road junction.

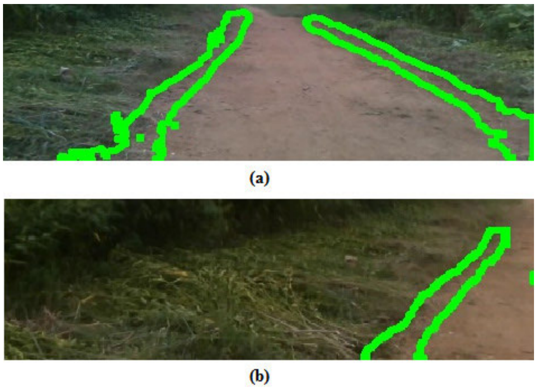


FIGURE 29. Vision-based off-road navigation(a) identification of off-road path edge (b) mobile robot moved off form curvature path.

the human beings holding the obstacles. The robot was set to the flight mode whenever it encountered the obstacles on its path. And then it continued its travel on the ground. There is no specification about the curvature and slip angle error as the system was operated in flight mode. Their robot is not tested in off-road with scattered small obstacles.

In comparison, the proposed C-PP with the Linear Actuator steering control successfully navigates the mobile robot in off-road and on-road terrains; the mobile robot navigated by the C-PP moves at variable speed due to curvature awareness. In the over-all range of 0.5 to 1.9 degrees, the proposed navigation algorithm has less slippage angle error over on road and off-road testbeds. Using IMU and Hall Effect sensors, the proposed C-PP algorithm and Linear Actuator steering mechanism navigate the mobile robot.

It has restrictions when it has been violently dashed by any object and has deviated from its reference path. In the future, the issue will be researched and resolved. In addition, the mobile robot will be equipped with additional sensors for obstacle detection.

IV. CONCLUSION AND FUTURE WORK

Curves can be found on off-road and on-road pathways. It is difficult to maneuver the mobile robot through the curved path because of its speed setting. When the mobile robot travels along uneven off-road terrain which is filled with scattered small obstacles and also along on-road terrain, a linear actuator-based steering solution is proposed and the same was implemented to keep the steering angle in the desired direction. A mobile robot fitted with a linear actuator-based steering control system was successfully tested in a path that was filled with small obstacles like building waste, M-Stand, and blue metals. We observed a lower overall RMSE slip angle error of 0.5–1.9 degrees in this scenario. We have noticed that our mobile robot can be navigated successfully using the proposed curve-aware pure pursuit (C-PP) algorithm on all terrains by extracting the curves present in our path using Google Maps. The proposed linear actuator-controlled steering system with a curve-aware pure pursuit (C-PP) algorithm successfully guided the mobile robot in real-time with lower RMSE lateral (0.000001) and longitudinal errors (0.000001) in off-road and lower RMSE lateral (0.000001) and longitudinal errors (0.000002) in on-road environments. According to the simulation, the proposed curve-aware pure pursuit (C-PP) algorithm navigated the mobile robot with lower RMSE lateral (0.19 meter) and longitudinal (0.21 meter) errors in simulated off-road and lower RMSE lateral (0.18 meter) and longitudinal (0.31 meter) errors in simulated on-road. When compared with the existing algorithms, the proposed algorithm consume a shorter navigation time (44 seconds), a shorter traveled trajectory length (24 meters), and a 100% success rate to reach the target location in off-road test-bed. The same 100% success rate, with lesser navigation time (117 seconds) and lesser traveled trajectory length (52 meters) was observed in the on-road test-bed. Future studies will look into the method's limitations, such as sudden course changes and overtaking obstacles.

ACKNOWLEDGMENT

This work forms part of the research work carried at Technology Information Forecasting and assessment Council-Centre of Relevance and Excellence (TIFAC-CORE) in Automotive Infotronics Centre (sponsored by Department of Science and Technology, Government of India), Vellore Institute of Technology, Vellore. The authors would like to acknowledge the support rendered by the centre in completing this work successfully.

REFERENCES

[1] H. Christensen, D. Paz, H. Zhang, D. Meyer, H. Xiang, Y. Han, Y. Liu, A. Liang, Z. Zhong, and S. Tang, "Autonomous vehicles for micro-mobility," *Auton. Intell. Syst.*, vol. 1, no. 1, Dec. 2021, Art. no. 11, doi: 10.1007/s43684-021-00010-2.

[2] M. Fnadi, F. Plumet, and F. Benamar, "Nonlinear tire cornering stiffness observer for a double steering off-road mobile robot," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 7529–7534, doi: 10.1109/icra.2019.8794047.

[3] Z. Al-Mashhadani and B. Chandrasekaran, "Survey of agricultural robot applications and implementation," in *Proc. 11th IEEE Annu. Inf. Technol., Electron. Mobile Commun. Conf. (IEMCON)*, Nov. 2020, pp. 76–81, doi: 10.1109/iemcon51383.2020.9284910.

[4] D. Aghi, S. Cerrato, V. Mazzia, and M. Chiaberge, "Deep semantic segmentation at the edge for autonomous navigation in vineyard rows," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2021, pp. 3421–3428.

[5] G. Kahn, P. Abbeel, and S. Levine, ''BADGR: An autonomous self-supervised learning-based navigation system,'' *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 1312–1319, Apr. 2021, doi: 10.1109/lra.2021.3057023.

[6] B. Fernandez, P. J. Herrera, and J. A. Cerrada, ''A simplified optimal path following controller for an agricultural skid-steering robot,'' *IEEE Access*, vol. 7, pp. 95932–95940, 2019, doi: 10.1109/access.2019.2929022.

[7] P. Kurtser, O. Ringdahl, N. Rotstein, R. Berenstein, and Y. Edan, ''In-field grape cluster size assessment for vine yield estimation using a mobile robot and a consumer level RGB-D camera,'' *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 2031–2038, Apr. 2020, doi: 10.1109/lra.2020.2970654.

[8] R. Lenain, A. Nizard, M. Deremetz, B. Thuilot, V. Papot, and C. Cariou, ''Path tracking of a bi-steerable mobile robot: An adaptive off-road multi-control law strategy,'' in *Proc. 15th Int. Conf. Informat. Control, Autom. Robot.*, 2018, pp. 173–180, doi: 10.5220/0006865801730180.

[9] J. Ni, J. Hu, and C. Xiang, ''An AWID and AWIS X-by-wire UGV: Design and hierarchical chassis dynamics control,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 2, pp. 654–666, Feb. 2019, doi: 10.1109/tits.2018.2824346.

[10] J. Li, J. Wang, H. Peng, Y. Hu, and H. Su, ''Fuzzy-torque approximation-enhanced sliding mode control for lateral stability of mobile robot,'' *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 52, no. 4, pp. 2491–2500, Apr. 2022, doi: 10.1109/tsmc.2021.3050616.

[11] J. Li, H. Qin, J. Wang, and J. Li, ''OpenStreetMap-based autonomous navigation for the four wheel-legged robot via 3D-LiDAR and CCD camera,'' *IEEE Trans. Ind. Electron.*, vol. 69, no. 3, pp. 2708–2717, Mar. 2022, doi: 10.1109/tie.2021.3070508.

[12] J. Zhao, T. Han, S. Wang, C. Liu, J. Fang, and S. Liu, ''Design and research of all-terrain wheel-legged robot,'' *Sensors*, vol. 21, no. 16, p. 5367, Aug. 2021, doi: 10.3390/s21165367.

[13] Y. Onozuka, R. Matsumi, and M. Shino, ''Autonomous mobile robot navigation independent of road boundary using driving recommendation map,'' in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2021, pp. 4501–4508, doi: 10.1109/IROS51168.2021.9636635.

[14] C. You and P. Tsiotras, ''High-speed cornering for autonomous off-road rally racing,'' *IEEE Trans. Control Syst. Technol.*, vol. 29, no. 2, pp. 485–501, Mar. 2021, doi: 10.1109/tcst.2019.2950354.

[15] Y. Gu, Z. Li, Z. Zhang, J. Li, and L. Chen, ''Path tracking control of field information-collecting robot based on improved convolutional neural network algorithm,'' *Sensors*, vol. 20, no. 3, p. 797, Jan. 2020, doi: 10.3390/s20030797.

[16] F. Rovira-Más, V. Saiz-Rubio, and A. Cuenca-Cuenca, ''Augmented perception for agricultural robots navigation,'' *IEEE Sensors J.*, vol. 21, no. 10, pp. 11712–11727, May 2021, doi: 10.1109/jsen.2020.3016081.

[17] Y. Ding. (Nov. 22, 2021). *Simple Understanding of Kinematic Bicycle Model*. [Online]. Available: https://dingyan89.medium.com/simple-understanding-of-kinematic-bicycle-model-81cac6420357

[18] I. Gwayi and M. S. Tsoeu, ''Rollover prevention and path following of autonomous vehicle using nonlinear model predictive control,'' in *Proc. Open Innov. Conf. (OI)*, Oct. 2018, pp. 13–18, doi: 10.1109/oi.2018.8535896.

[19] F. Gauthier-Clerc, A. Hill, J. Laneurit, R. Lenain, and É. Lucet, ''Online velocity fluctuation of off-road wheeled mobile robots: A reinforcement learning approach,'' in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 2421–2427, doi: 10.1109/icra48506.2021.9560816.

[20] M. A. H. Ali and M. Mailah, ''Path planning and control of mobile robot in road environments using sensor fusion and active force control,'' *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 2176–2195, Mar. 2019, doi: 10.1109/tvt.2019.2893878.

[21] N. S. Manikandan and G. Kaliyaperumal, ''Collision avoidance approaches for autonomous mobile robots to tackle the problem of pedestrians roaming on campus road,'' *Pattern Recognit. Lett.*, vol. 160, pp. 112–121, Aug. 2022, doi: 10.1016/j.patrec.2022.06.005.

[22] F. Liu, X. Li, S. Yuan, and W. Lan, ''Slip-aware motion estimation for off-road mobile robots via multi-innovation unscented Kalman filter,'' *IEEE Access*, vol. 8, pp. 43482–43496, 2020, doi: 10.1109/access.2020.2977889.

[23] C. Premachandra, M. Otsuka, R. Gohara, T. Ninomiya, and K. Kato, ''A study on development of a hybrid aerial terrestrial robot system for avoiding ground obstacles by flight,'' *IEEE/CAA J. Autom. Sinica*, vol. 6, no. 1, pp. 327–336, Jan. 2019, doi: 10.1109/jas.2018.7511258.

[24] Q. Xu, H. Li, Q. Wang, and C. Wang, ''Wheel deflection control of agricultural vehicles with four-wheel independent omnidirectional steering,'' *Actuators*, vol. 10, no. 12, p. 334, Dec. 2021, doi: 10.3390/act10120334.

[25] P. I. Philatenkov, E. V. Morozova, and T. S. Morozova, ''Linear actuator modeling,'' in *Proc. IEEE Conf. Russian Young Researchers Electr. Electron. Eng. (EIConRus)*, Jan. 2020, pp. 807–809, doi: 10.1109/eiconrus49466.2020.9039171.

[26] S. Manikandan, G. Kaliyaperumal, S. Hakak, and T. R. Gadekallu, ''Curve-aware model predictive control (C-MPC) trajectory tracking for automated guided vehicle (AGV) over on-road, in-door, and agricultural-land,'' *Sustainability*, vol. 14, no. 19, p. 12021, Sep. 2022, doi: 10.3390/su141912021.

[27] AtsushiSakai, GitHub. (Nov. 25, 2019). *AtsushiSakai/PythonRobotics*. [Online]. Available: https://github.com/AtsushiSakai/PythonRobotics

N. S. MANIKANDAN received the M.Sc. degree in computer science from Pondicherry University, Pondicherry, India, in 2002, and the M.Tech. degree in computer science and engineering from the Vellore Institute of Technology, Vellore, India, in 2006, where he is currently pursuing the Ph.D. degree in self-driving vehicle with the TIFAC-CORE Automotive Infotronics Laboratory. His research interests include digital image processing, deep learning, reinforcement learning, embedded system design, wireless sensor networks, mobile ad-hoc networks, vehicular ad-hoc networks, path planning, and path tracking.

GANESAN KALIYAPERUMAL received the M.Sc. and Ph.D. degrees from Bharathidasan University, India, in 1986 and 1993, respectively. He received Postdoctoral Fellowship from the Queen's University of Belfast, Belfast, U.K., in 1997. He was the Head of the Department of Computer Science, the Dean of the School of Information Technology and Engineering, the Director of the TIFAC-CORE in Automotive Infotronics Centre, and the Chairperson of the VIT Business School, Vellore Institute of Technology (VIT), Vellore, where he is currently a Senior Professor with the School of Information Technology and Engineering. His research interests include digital signal, image and video processing, cryptography, embedded system design, wireless systems, chaos based security systems in transform domains, decision support systems, soft computing techniques, the Internet of Things, machine learning, artificial intelligence, and big data.