

# A Waypoint Tracking Controller for Autonomous Vehicles Using CARLA Simulator



Chinmay Srinivas and Sharanbassappa S. Patil

**Abstract** The main objective of this work is the development and testing of a waypoint tracking controller for autonomous vehicles. The waypoint tracking controller is developed in Python and tested using CARLA Simulator, an open-source simulator for autonomous driving research. The waypoint tracking controller is developed by implementing a PID controller for longitudinal control and a pure pursuit controller for lateral control. The PID gains are tuned to precisely track the reference speed profile. The pure pursuit controller is implemented to accurately follow the desired path. The developed controller is tested in CARLA Simulator and the simulation results indicate that the vehicle tracks 100% of the waypoints specified in the trajectory. The waypoint tracking controller also performs well in closely tracking the reference speed profile. The developed waypoint tracking controller is precise and suitable for application in autonomous vehicles for urban driving environments

**Keywords** Waypoint tracking controller · Autonomous vehicles · Autonomous driving simulation · CARLA simulator

## Nomenclature

$v$	Actual speed
$v_d$	Desired speed
$u$	Acceleration
KP	Proportional gain
KI	Integral gain

---

C. Srinivas (✉)  
PES University, Bengaluru, Karnataka 560085, India  
e-mail: [chinmay1567@gmail.com](mailto:chinmay1567@gmail.com)

S. S. Patil  
Department of Mechanical Engineering, PES University, Bengaluru, Karnataka 560085, India  
e-mail: [sspatil@pes.edu](mailto:sspatil@pes.edu)

KD	Derivative gain
TP	Throttle position
BP	Brake position
$\delta$	Steering angle
ld	Look-ahead distance

## 1 Introduction

Autonomous vehicles are the future of mobility and they are changing the way people think about travel. They can bring about a significant improvement in road safety and provide mobility for everyone while drastically reducing the cost of driving. A fully autonomous vehicle requires the automated driving system (ADS) to entirely perform the dynamic driving task (DDT) [1]. The driving task consists mainly of three sub-tasks namely perception, motion planning and vehicle control. Perception involves perceiving the driving surroundings including tracking and predicting a vehicle's motion, recognizing various elements such as road surface, road signs, vehicles and pedestrians. Motion planning contributes to defining the path or trajectory which can be tracked to reach the destination successfully. Finally, vehicle control is responsible for taking the appropriate steering, acceleration and braking decisions to control the vehicle's position, orientation and velocity.

At any point in time, the appropriate driving behaviour is decided by interpretation of the estimated vehicle pose and information on the current scene, obtained from the various on-board sensors such as GPS/IMU units, odometry, cameras and LIDARs. The driving behaviour which needs to be accomplished in the current situation is translated into a path or a trajectory, defined by a sequence of future waypoints that the vehicle should track. The generated waypoints can be tracked precisely by implementing a waypoint tracking controller. The waypoint tracking controller includes longitudinal and lateral control of the vehicle. In the developed waypoint tracking controller, a PID controller and a pure pursuit controller are implemented for longitudinal and lateral control, respectively.

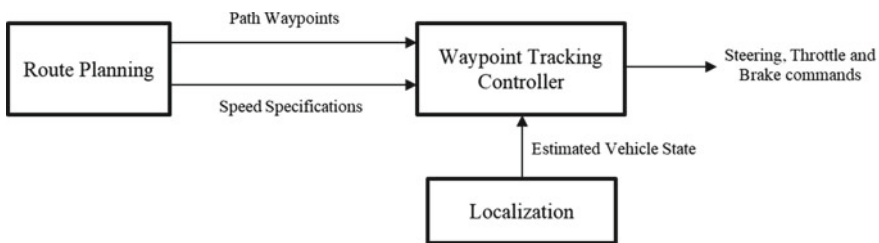
A robust control system allows for safe and precise execution of the planned movements. For specified path waypoints and reference speed specifications, the vehicle should be able to track the specified path with minimum error. Vehicle dynamics, feasibility and comfort must be considered by the controller to precisely track the waypoints and follow the required trajectory. The objective of this work is to develop and simulate a waypoint tracking controller for autonomous vehicles. The developed waypoint tracking controller is programmed in Python [2] and tested using the hyper-realistic CARLA Simulator [3], an open-source simulator for autonomous driving research.

## 2 Controller Design

Navigating an autonomous vehicle to a specified destination requires motion planning and vehicle control. Currently, many applications such as OpenDRIVE [4] and OpenStreetMaps [5] provide the tools for the effortless creation of maps for navigation. Route planning provides a path to a specific destination that should be tracked by the path tracking controller for successfully reaching that destination. A vehicle state estimator uses a GPS-based positioning system, inertial measurement unit (IMU) and wheel odometer to estimate the current position, orientation and velocity of the vehicle. The role of the controller then is to regulate these states of the vehicle by generating appropriate actuator signals based on the desired and currently estimated states (Fig. 1).

A waypoint tracking controller obtains a set of path waypoints and reference speed specifications along with the estimated state of the vehicle as inputs and generates the vehicle control signals as outputs. The waypoint tracking controller is further comprised of two controllers, a longitudinal controller and a lateral controller. The longitudinal controller regulates the vehicle's longitudinal motion via throttle or braking whereas the lateral controller regulates the vehicle's lateral motion via steering.

Some of the basic controllers such as proportional integral and derivative (PID) controller [6] or a combination of feedback and feedforward controller [7] can be employed for longitudinal control. For lateral control, the most popular controllers are the geometric ones such as pure pursuit [8] and Stanley [9] controllers. Also, advanced applied control strategies like model predictive control (MPC) [10] can be incorporated for joint longitudinal and lateral control. In the developed waypoint tracking controller, a PID controller and a pure pursuit controller are employed for longitudinal control and lateral control, respectively.



**Fig. 1** General navigation architecture of an autonomous vehicle

### 2.1 Waypoints and Reference Speed Profile

The RaceTrack map in CARLA Simulator is chosen for the deployment of the waypoint tracking controller. A starting point and destination are selected in this track and the roadway between these two points is taken as the path that must be tracked. This path is defined by equally spaced waypoints that are stored in a sorted list. In this track, a total of 1724 waypoints are defined which are equally spaced at 1 m apart along the path. Each of the waypoints includes their positions defined by the x and y coordinates, as well as the speed which the vehicle should attain. Hence, the waypoints become the reference signal for the waypoint tracking controller and navigating through all the waypoints effectively completes the full path.

The set of waypoints is updated as the vehicle progresses on the path by finding the nearest waypoint. The controller directly uses the next waypoint in the path for implementing vehicle control. At higher speeds, the controller performs better with a continuous path. Since the waypoints are discrete, interpolating between each waypoint will reduce the spacing between each waypoint and provide a finer resolution path. The linear interpolation method is adopted to generate a smoother path and reference speed profile (Fig. 2).

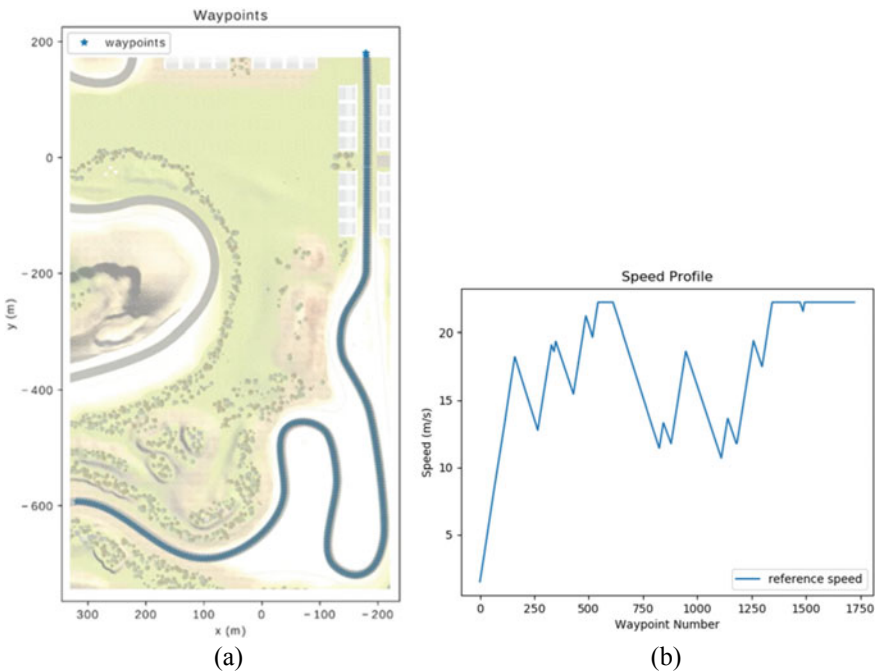


Fig. 2 Track used for simulation: a path waypoints b reference speed profile

## 2.2 Longitudinal Control

The longitudinal controller regulates the vehicle’s longitudinal motion by applying throttle or brake. A well-known example of longitudinal control is cruise control. A cruise control system performs the function of maintaining a fixed reference speed as well as accelerating or decelerating to a new desired speed using throttle and brake commands. A typical longitudinal controller can be split into two levels, an upper level controller and a lower level controller. Based on the speed error, i.e., the difference between the desired speed and the vehicle’s actual speed, the upper level controller determines the acceleration needed at each time step to match the desired speed. The lower level controller generates the throttle or braking actuation to track the desired acceleration (Fig. 3).

The developed waypoint tracking controller uses a PID controller for longitudinal control. The desired speed is fed to this PID controller as a reference and it outputs the throttle and brake commands for actuation. In urban environments, assuming that the desired speed is reasonably low and stable, the lower level controller is completely overridden (Fig. 4).

The PID controller includes three gain terms. First, a proportional gain term  $K_P$  multiplies the acceleration of the vehicle with a magnitude proportional to the speed error, ensuring that the vehicle accelerates in the right direction. Second, an integral gain term  $K_I$  considers the previously accumulated speed errors to make certain the steady-state errors are removed in the output. Finally, the derivative gain term  $K_D$  reduces the settling time by damping the overshoot caused by the integration term based on the rate of change of speed errors. The acceleration of the vehicle given by the PID controller output is,

$$u = K_P(v_d - v) + K_I \int_0^t (v_d - v)dt + K_D \frac{d(v_d - v)}{dt} \tag{1}$$

Proper selection of the PID gains is critical for precise tracking of the desired speed.

A good PID controller response should have minimal overshoot and steady-state error with a low rise time and settling time. The comparison between the response of a P, PD and PID controller is shown in Fig. 5. The P controller has a higher overshoot

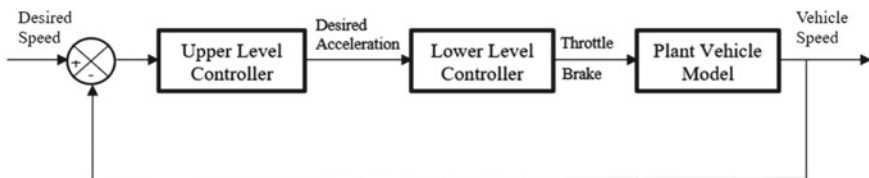


Fig. 3 Longitudinal control system

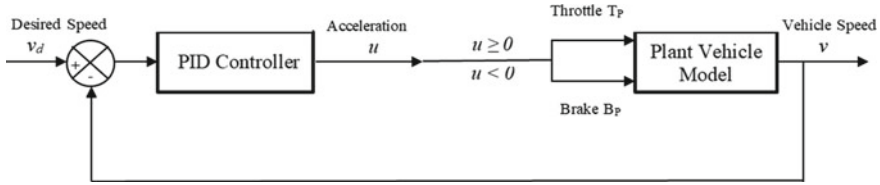


Fig. 4 Longitudinal controller implemented in the waypoint tracking controller

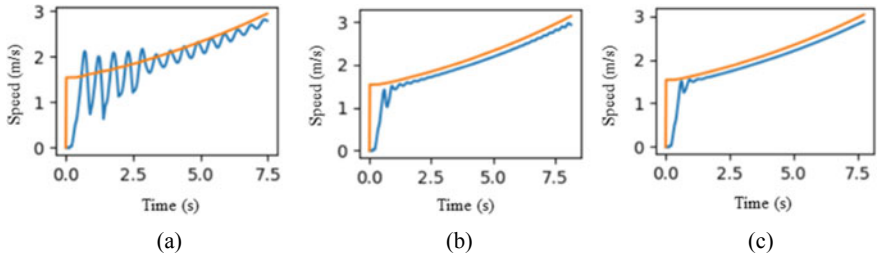


Fig. 5 Longitudinal controller response: a P controller b PD controller c PID controller

and a longer settling time than PD and PID controllers. But the PD controller has a slightly larger steady-state error than the PID controller. Hence, the PID controller performs better in precisely tracking the desired speed.

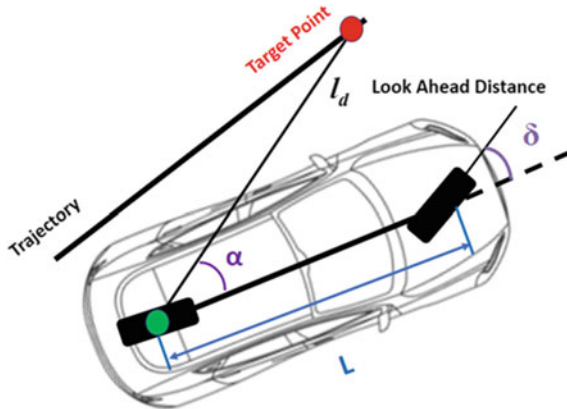
Finally in the longitudinal controller, the PID controller output, i.e. the obtained acceleration value is converted into throttle and brake commands. To generate the throttle and brake actuation signals, the throttle and brake outputs are clamped between 0 and 1. The positive acceleration outputs correspond to throttle and negative acceleration outputs correspond to brake. The throttle and brake outputs of the longitudinal controller can be written as,

$$\text{If } u \geq 0 : T_P = u, B_P = 0 \tag{2}$$

$$\text{If } u < 0 : T_P = 0, B_P = -u \tag{3}$$

### 2.3 Lateral Control

The lateral controller regulates the vehicle’s lateral motion via steering. A type of lateral controller is the geometric path tracking controller. Assuming the no-slip condition is true at the wheels, the geometric path tracking controller ignores dynamic forces on vehicles. This controller depends on a kinematic bicycle model to create and regulate the steering commands for path tracking. However, this approach cannot



**Fig. 6** Pure pursuit method for geometric path tracking

be applied for aggressive driving and manoeuvres with high lateral acceleration as its accuracy declines once the no-slip assumption doesn't hold true. Hence this controller is more suitable for driving in urban environments.

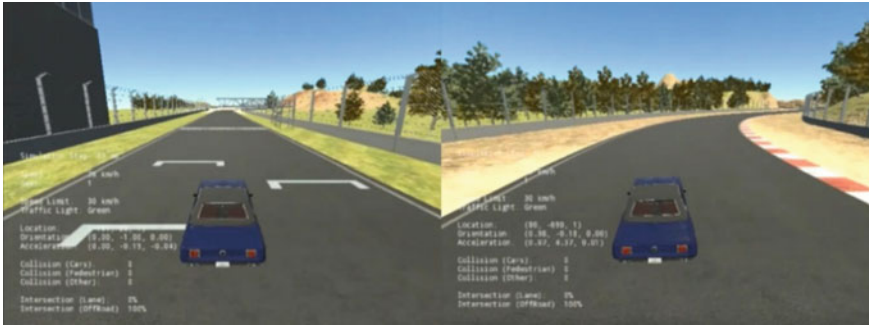
The waypoint tracking controller uses a pure pursuit controller which is a geometric path tracking controller for lateral control (Fig. 6). In the pure pursuit method, the centre of the rear axle in the vehicle is taken as the reference point and at a fixed distance from this point, a target point is determined on the path. This fixed distance is termed as the look-ahead distance ( $l_d$ ). The angle between the line connecting these two points and the heading of the vehicle is called as alpha ( $\alpha$ ). Based on this the constant steering angle required to meet the target point is calculated. This steering angle is then finally converted into steering commands that must be followed by the vehicle to accurately track the path. While the vehicle heads towards the desired path, the target point also shifts ahead on the path and thus gradually reduces the steering angle. This allows for a safe and gentle manoeuvre of the vehicle towards the path that needs to be tracked.

For precise tracking of the trajectory at all vehicle speeds, the lateral controller should also take into account the forward velocity of the vehicle. Hence, the look-ahead distance ( $l_d$ ) is assigned as a function of the vehicle forward speed ( $v_f$ ) by adding a proportionality constant ( $K_{dd}$ ) with its units in seconds. This constant can be tuned to change the stability and tracking performance of the pure pursuit controller for a particular course.

$$l_d = K_{dd}v_f \tag{4}$$

The steering angle output from the pure pursuit controller is given by Eq. 5. For practical purposes, this steering angle is limited from  $-1.22$  rad to  $1.22$  rad.

$$\delta = \tan^{-1}\left(\frac{2L \sin \alpha}{l_d}\right) = \tan^{-1}\left(\frac{2L \sin \alpha}{K_{dd}v_f}\right) \tag{5}$$



**Fig. 7** Snapshots of the simulation in CARLA Simulator

Finally, in the lateral controller, the steering angle output is bound to  $-1$  to  $1$  range limits to generate the steering command for steer actuation.

### 3 Simulation

A realistic simulation environment is essential for the development of an autonomous vehicle because it enables testing of the autonomous vehicle under various conditions and scenarios without even stepping into it. The simulation allows monitoring the vehicle's performance to ensure that the vehicle operates safely before deploying it on actual roads. One such highly realistic simulator used by a large number of researchers is CARLA Simulator. CARLA is an open-source simulator built using the Unreal game engine solely for autonomous driving research. It features highly detailed virtual worlds with vast selectable maps, numerous vehicle models, pedestrians, buildings and traffic signs including a wide range of adjustable environmental conditions (Fig. 7).

To test the accuracy and precision of the developed waypoint tracking controller, a path tracking simulation is carried out in CARLA using the specifications listed in Table 1. The PID controller is tuned to improve its performance in tracking the reference speed profile and the best resulting gain values  $K_P$ ,  $K_I$  and  $K_D$  are chosen as 2.5, 1 and 0.225, respectively for the simulation. The value of  $K_{dd}$  is chosen as 1 for this path to accurately track all the waypoints in the test bench.

### 4 Results and Conclusion

The throttle, brake and steer outputs from the waypoint tracking controller are shown in Fig. 8. The throttle and brake outputs range from 0 to 1, corresponding to the acceleration output from the PID controller. These throttle and brake outputs function as



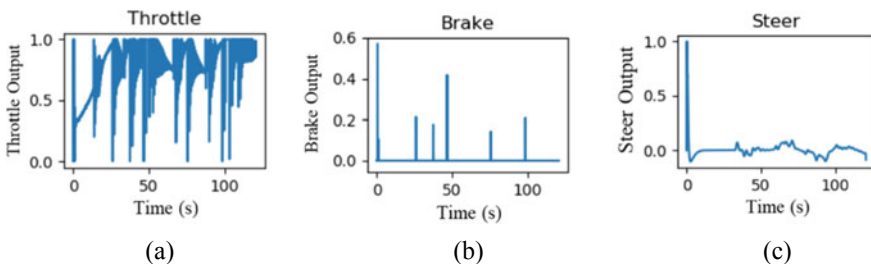
**Table 1** Simulation specifications

Parameter	Value
Processor	Intel i5-7300HQ @ 2.5 GHz
Simulator resolution	640 × 480
Simulator Frames Per Second (FPS)	30
Proportional Gain ( $K_D$ )	2.5
Integral Gain ( $K_I$ )	1
Derivative Gain ( $K_D$ )	0.225
Proportional constant ( $K_{dd}$ )	1

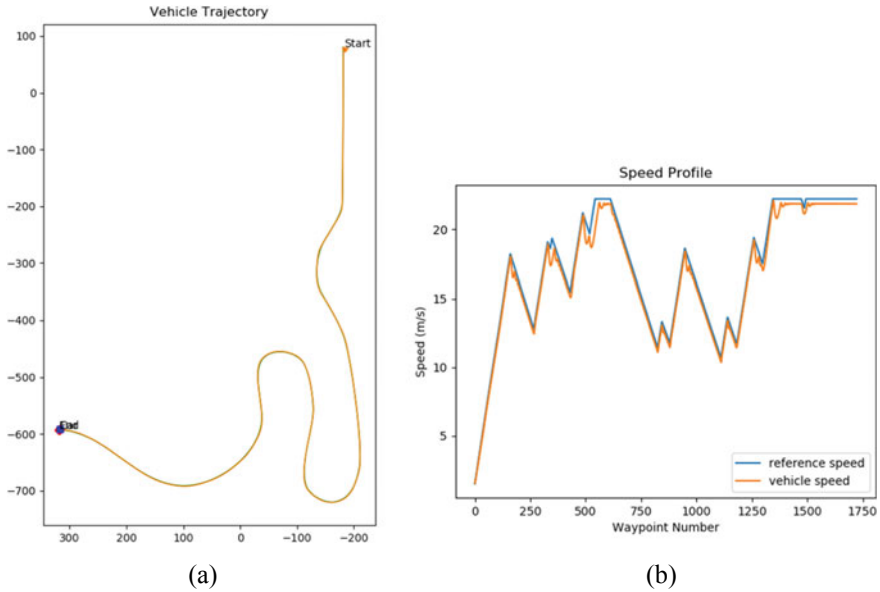
the actuation signals for adjusting the throttle and brake pedal position of the vehicle to accelerate accordingly. The steer output ranges from  $-1$  to  $1$ , corresponding to the steering angle output from the pure pursuit controller. This steer output actuates the steering of the vehicle to turn the vehicle accordingly.

The test results indicate that the vehicle tracks 100% of the specified waypoints in the trajectory. Further, the waypoint tracking controller performs well in closely tracking the reference speed profile (Fig. 9).

This work discussed the development and simulation of a waypoint tracking controller for autonomous vehicles. The waypoint tracking controller is developed by implementing a PID controller for longitudinal control and a pure pursuit controller for lateral control. The PID gains are tuned to precisely track the reference speed profile. A pure pursuit controller is implemented for lateral control to accurately follow the desired path. The developed waypoint controller is deployed in CARLA Simulator. The simulation results indicate that the vehicle tracks 100% of the waypoints specified in the trajectory. The waypoint tracking controller also performs well in closely tracking the reference speed profile. The developed waypoint tracking controller is precise and suitable for application in autonomous vehicles for urban driving environments.



**Fig. 8** Waypoint tracking controller outputs: **a** throttle **b** brake **c** steer



**Fig. 9** Simulation results: **a** vehicle trajectory **b** vehicle speed profile

## References

1. SAE Standard J3016: Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles. [https://www.sae.org/standards/content/j3016\\_202104](https://www.sae.org/standards/content/j3016_202104) (2021). Accessed 16 Oct 2021
2. Python Programming Language Homepage. <https://www.python.org>. Accessed 16 Oct 2021
3. CARLA Simulator Homepage, <https://carla.org>. Accessed 16 Oct 2021
4. OpenDRIVE, <https://www.asam.net/standards/detail/opendrive>. Accessed 16 Oct 2021
5. OpenStreetMaps, <https://www.openstreetmap.org>. Accessed 16 Oct 2021
6. Rajamani R (2012) Introduction to longitudinal control. In: Vehicle dynamics and control. 2nd edn., pp 123–150, Springer, USA
7. Sailan K, Kuhnert KD (2013) Modeling and design of cruise control system with feedforward for all terrain vehicles. In: Second International Conference on Advanced Information Technologies and Applications. Dubai, UAE
8. Samuel M, Hussein M, Mohamad MB (2016) A review of some pure-pursuit based path tracking techniques for control of autonomous vehicle. *Int J Comput Appl* 135:35–38
9. Hoffman GM, Tomlin CJ, Montemerlo M, Thrun S (2007) Autonomous automobile trajectory tracking for off-road driving: controller design, experimental validation and racing. In: Proceedings of the 2007 American control conference, pp 2296–2301, New York, NY, USA
10. Falcone P, Borrelli F, Asgari J, Tseng H, Hrovat D (2007) Predictive active steering control for autonomous vehicle systems. *IEEE Trans Control Syst Technol* 15(3):566–579