

Dynamic Vehicle Modelling and Controlling Techniques for Autonomous Vehicle Systems

R. Sushma¹, J. Satheesh Kumar²

¹Student, Department of Electronics and Instrumentation Engineering, Dayananda Sagar College of Engineering, Bangalore, Karnataka, India

²Associate Professor, Department of Electronics and Instrumentation Engineering, Dayananda Sagar College of Engineering, Bangalore, Karnataka, India

E-mail: ¹sushmarajanna1712@gmail.com, ²jsatheeshngl@gmail.com

Abstract

The driving scenario of an automated vehicle is the crucial technology in the design of autonomous cars. This suggested approach aims to address the shortcomings of autonomous cars, such as their poor real-time performance and low control precision. The process for building a virtual simulation environment for autonomous vehicle testing and validation is described in this study. Model Predictive Control and Proportional Integral and Derivative Control are used in MATLAB simulation to build three car models. These are related to the 2D and 3D animation used in collision detection and visualization. The virtual engine visualization is included throughout the model. A variety of test circumstances are used to validate the simulation model, and the model's performance is assessed in the presence of various barriers. The simulation's findings demonstrate that the autonomous vehicle has a strong potential for self-adaptation even in challenging and complex working environments. No instances of car sideslip or track departure have been noted. It is discovered that this autonomous car performs remarkably well overall when compared to other autonomous vehicles. The suggested approach is essential for enhancing autonomous vehicle driving safety, maintaining vehicle control in challenging situations, and improving the advancement of intelligent vehicle driving assistance.

Keywords: Autonomous vehicle, model predictive control, highway lane, visualization

1. Introduction

The goal of this research is to provide a virtual proving environment for the operation of autonomous vehicles. Why is virtual environment needed when everything can

be simply simulated in a real-world environment with actual kilometers? The issue is that autonomy is pushing forward the automotive industry, and hence the companies are attempting to create fully autonomous vehicles. Additionally, they need to certify and guarantee the safety of those vehicles because it removes all liability from the driver, and the companies are responsible for any problem or damage [9]. So, this issue can be solved by creating this model simulation so that, the cars can be modelled and placed in a busy metropolitan setting, so that they can communicate with one another.

The fact that the cars are interacting with one another adds to the complexity of the issue because the other dynamic objects are also decision-makers and can add further complexity to the traffic situation. In order to discuss this subject, the vehicle's parts must be put to test as well as the environment in which it operates and the activities it does in various situations. In a busy metropolitan traffic environment, there are countless potential situations. Because all conceivable outcomes add up to infinity, and a vehicle cannot be completely verified by taking all possible outcomes into account. Instead, it must be ensured that the algorithms and the manner in which actions are conducted provide safety assurances. To do this, the issue must be divided into several levels, including the component car level and the traffic level. So, to ensure that the car does a safe drive in a standard circumstance, the components of autonomous vehicles which specify the algorithms and control loops are designed first [7] & [9]. Initially, a vehicle model is created that is Model Predictive Control (MPC) [1][2] using adaptive cruise control system, and again a vehicle model is created for the purpose of Proportional Integral and Derivative (PID) cruise control, both containing vehicle following mode and free drive mode along with driving mode selector [8]. All these are connected to a collision detector, and the simulation stops if collision occurs [3][6]. A designed car model is shown in figure 1.



Figure 1. Car Model

2D animation and 3D animation visualization simulations are created. This method is vital for boosting autonomous vehicle control and driving safety, and for the development of intelligent vehicle driving assistance [12].

2. Methodology

This work is divided into 3 sections and every section has its own function. The 1st section consists of preparing the vehicle model of MPC and PID cruise control which is used for the creation of perfect vehicle model. Collision detection is used for an emergency purpose; if collision occurs, the simulation stops [6]. The 2nd section consists of preparing a visualization of the vehicle model in 2D and 3D animation. The 3rd section consists of an unreal engine visualization, which has 3 different cars in a highway lane with 3D animation which are ego vehicle and lead vehicle [13].

When an ego vehicle is assigned with some initial speed and position, the vehicle follows the lane accordingly. Speed control is one of the major drawbacks in autonomous vehicle and controlling that speed and developing automatic braking system is a difficult task. With the help of developing a MPC and PID Cruise Control, both can be achieved. Along with that, the vehicle following in the same lane is also an important task. An ego vehicle follows the lane and overtakes the lead vehicle without collision, without track changing and without sideslip [10], using a sensor. The result of this model is that it has a very good self-adaptivity under severe and complicated working conditions and no such cases as sideslip of car or departure of track were observed.

3. System Flow

Autonomous vehicle following highway lane begins with creating a vehicle model which is of MPC and PID Cruise Control as shown in the figure 2. A vehicle model is created using transfer function, which is the acceleration to speed converter. This contains initial speed and initial position of the vehicle which can be varied. Then a vehicle model is created, that is MPC using adaptive cruise control system which has vehicle following mode and free drive-mode along with driving-mode selector [1][2].

Another model is created for the purpose of PID cruise control which also contains vehicle following mode and free drive mode along with driving mode selector [8]. All these are connected to a collision detector which is used to detect when vehicles are nearby and yet

to collide and stop that from occurring. A rear collision is used to compare the constant value that is to detect the distance between the vehicles. If collision occurs, then the simulation stops [3][6].

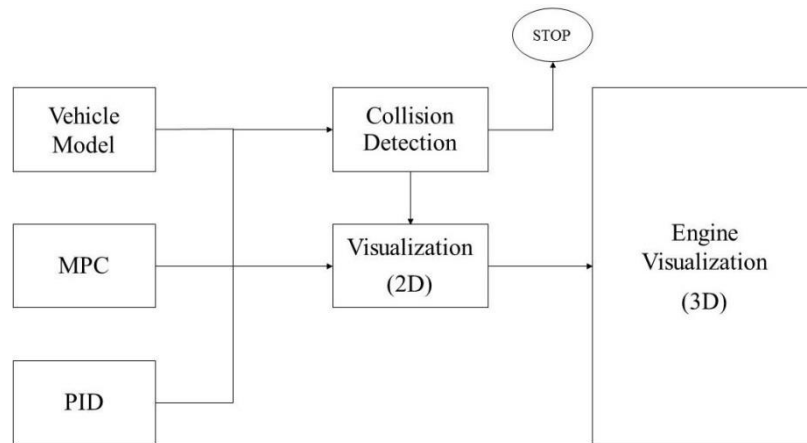


Figure 2. Block Diagram

2D animation and 3D animation visualization simulations are created. Where 2D animation contains MATLAB function model with position and angle of the vehicle, and 3D animation contains VR sink which has VR signal expander and bus creator connected to it which shows the vehicle model in virtual world. All these are connected to an unreal engine visualization which has a 3D simulation vehicle with ground following and simulation. 3D scene configuration works as sensor. And there is a scope connected to this model which shows the graphical representation of the speed of the vehicles [13].

3.1 Model Predictive Control

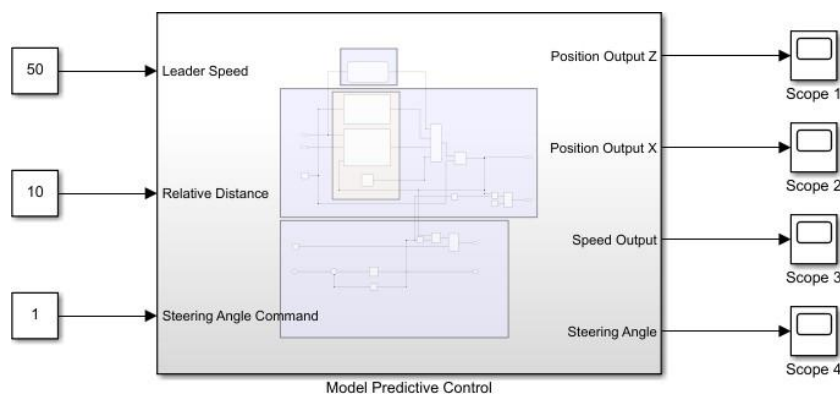


Figure 3. Model Predictive Control

Model Predictive Control (MPC) refers to a class of sophisticated control techniques that employ a process model to forecast the future behaviour of the controlled system. A

control algorithm based on a process prediction model is known as predictive control. Using knowledge about the process and predicted future input, this is utilized to forecast the future output. Instead of focusing on the model's structure, it highlights the model function [4].

The inputs given are relative distance of the vehicle, speed of leader vehicle and steering angle [5]. The outputs obtained are position of the vehicle, speed of the vehicle and the steering angle of the vehicle, which is shown in figure 3.

A speed versus time graph is obtained for the MPC, which are shown in figures 4,5,6 & 7. It represents the position output, speed output and steering angle of the vehicle model.

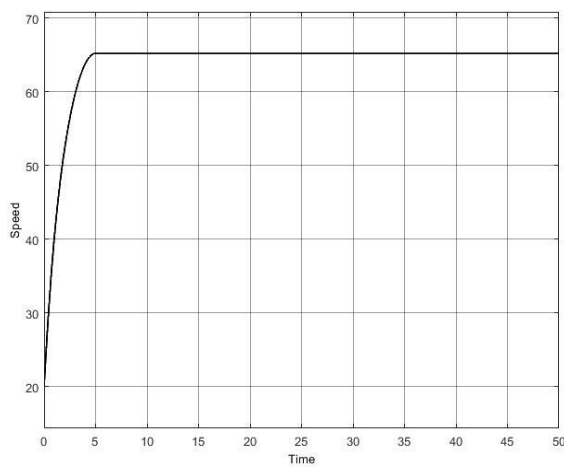


Figure 4. Position Output Z (Scope 1)

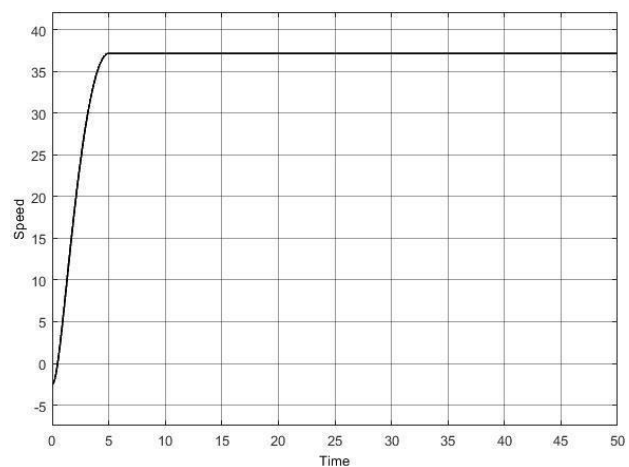


Figure 5. Position Output X (Scope 2)

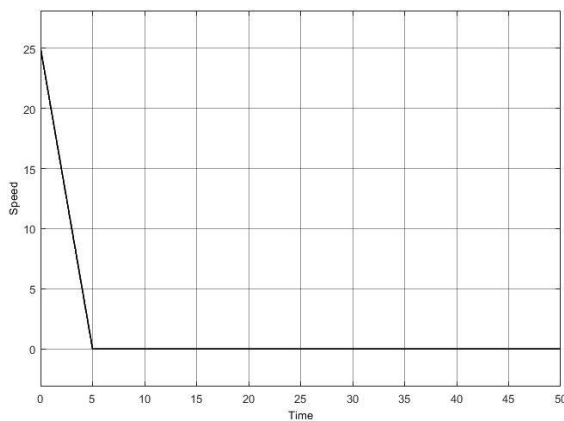


Figure 6. Speed Output (Scope 3)

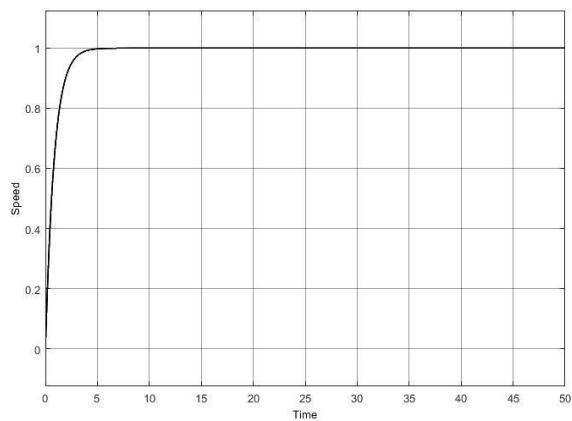


Figure 7. Steering Angle (Scope 4)

3.2 Proportional Integral Derivative

By managing the engine's power output, the PID algorithm returns the speed of the vehicle to the speed that was selected as the target. PID controllers use a proportional control strategy together with extra integral and derivative adjustments to assist the device

automatically that accounts for system changes. To push the process level to the set point, process conditions are measured, feedback is computed, and output is changed as necessary for the PID controller [8].

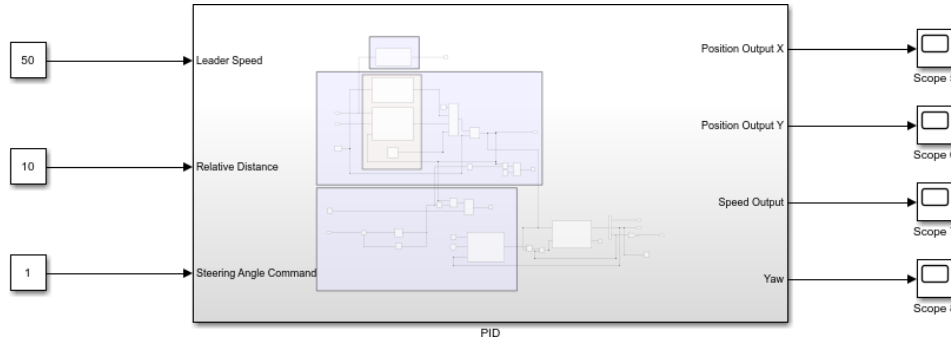


Figure 8. Proportional Integral Derivative

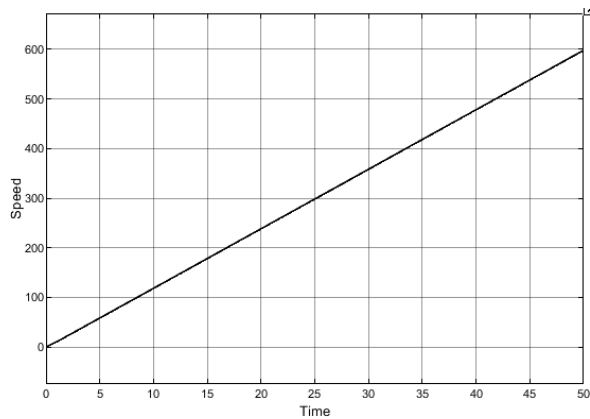


Figure 9. Position Output X (Scope 5)

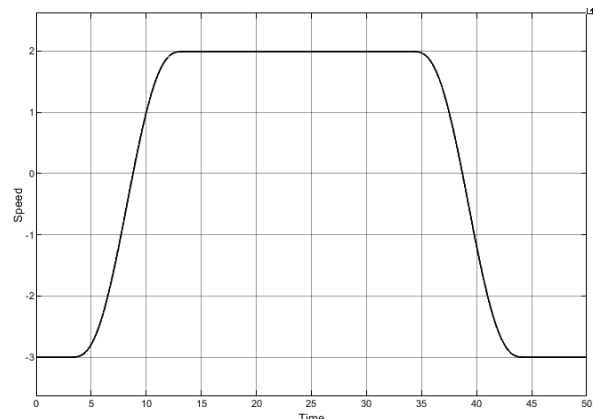


Figure 10. Position Output Y (Scope 6)

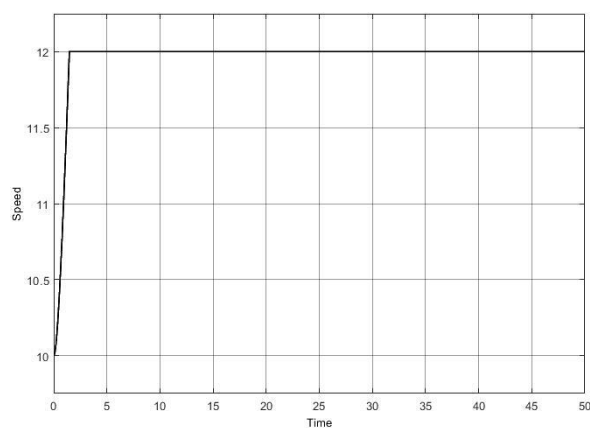


Figure 11. Speed Output (Scope 7)

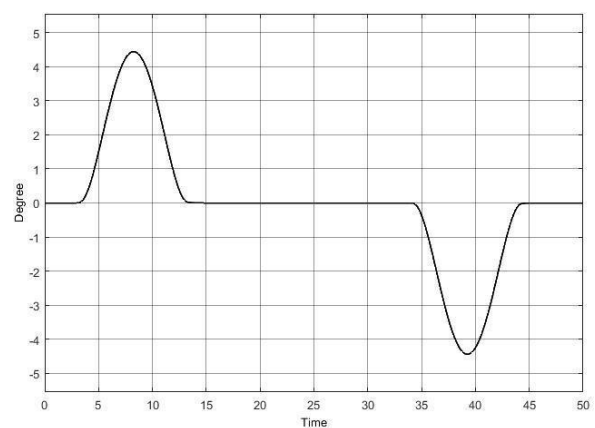


Figure 12. Yaw (Scope 8)

The inputs given are relative distance of the vehicle, speed of leader vehicle and steering angle. The outputs obtained are position of the vehicle, speed of the vehicle and the yaw (angle of vehicle) as shown in figure 8 [11]. A speed versus time graph is obtained for the

PID as shown in figure 9,10,11 and 12. It represents the position output, speed output and yaw of the vehicle model.

4. Results and Discussion

According to the simulation findings, the autonomous vehicle has high self-adaptivity even under difficult and demanding working conditions, and no instances of sideswiping or track departure were noted. This was determined to have impressively complete performance when compared to other autonomous cars. With a thorough brake control system, as illustrated in Figure 13, the car overtakes another vehicle without colliding or side sliding. The graph in Figure 14 displays the speed fluctuation of the vehicle [11].



Figure 13. Autonomous Vehicle

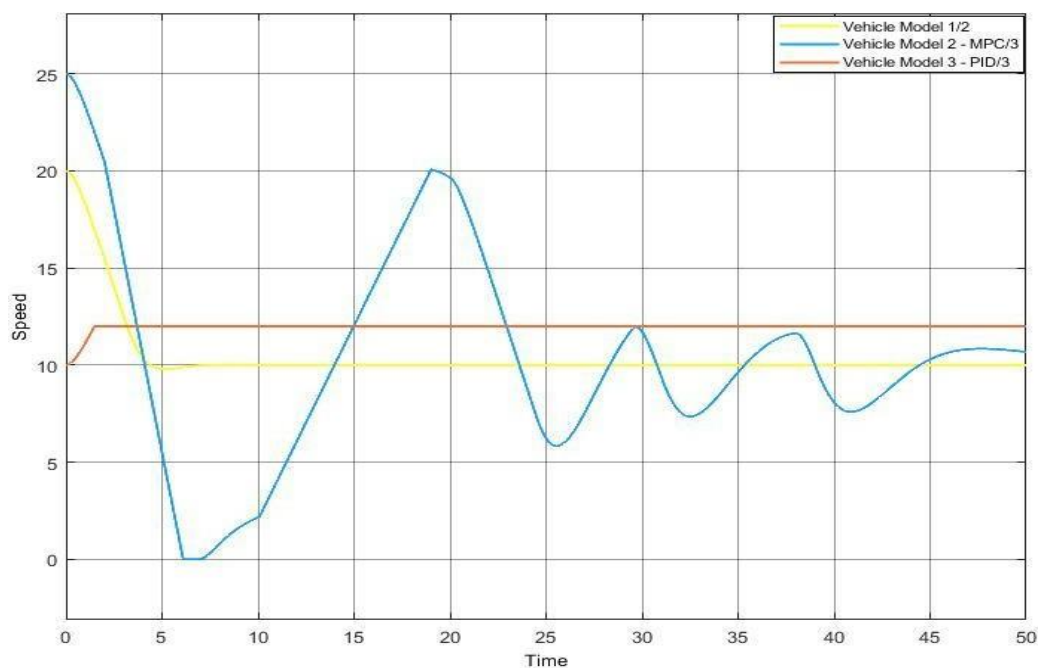


Figure 14. Speed of Vehicles

5. Conclusion

The autonomous vehicle concept is a well-established concept that has aided in numerous driving situations, including improved driver assistant systems and automatic braking systems. This paper achieves in creating a virtual validation environment for autonomous vehicle functions. Firstly, three vehicle models are created using MPC and PID with appropriate simulation. These are connected to collision detection and visualization which has 2D and 3D animation. This is then connected to unreal engine visualization. Finally, using MATLAB or Simulink software a simulation model is created and autonomous vehicle with different obstacles under different vehicle speed are tested through simulation experiment. The simulation finding show that the autonomous vehicle has good self-adaptivity under the challenging operating conditions and no incidents of sideslip of car or departure of track were noticed. The suggested method improves vehicle control in difficult situations, increases driving safety, and helps in the progression of smart vehicle driving support, and is found to have remarkable comprehensive performance compared to other autonomous vehicles.

References

1. Max Schwenzer, Muzaffer Ay, Thomas Bergs & Dirk Abel, “Review on model predictive control: an engineering perspective”, *The International Journal of Advanced Manufacturing Technology* volume 117, pages 1327–1349 (2021)
2. Kai Zhang, Yeqi Lu, Xiaoyue Huang, “Design of Adaptive Cruise Control considering multi scene application”, IEEE, DOI: 10.1109/CAC53003.2021.9728110, China Automation Congress (CAC), 2021.
3. Z. Wang, G. Li, H. Jiang, Q. Chen, and H. Zhang, “Collision-free navigation of autonomous vehicles using convex quadratic programming-based model predictive control,” *IEEE/ASME Trans. Mechatronics*, vol. 23, no. 3, pp. 1103–1113, Jun. 2018.
4. C. Liu, S. Lee, S. Varnhagen, and H. E. Tseng, “Path planning for autonomous vehicles using model predictive control” in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Beach, CA, USA, Jun. 2017, pp. 174–179.
5. B. Zhang, C. Zong, G. Chen, and B. Zhang, “Electrical vehicle path tracking based model predictive control with a Laguerre function and exponential weight” *IEEE Access*, vol. 7, pp. 17082–17097, 2019.

6. J. Funke, M. Brown, S. M. Erlien, and J. C. Gerdes, "Collision avoidance and stabilization for autonomous vehicles in emergency scenarios," *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 4, pp. 1204–1216, Jul. 2017.
7. M. Park, S. Lee, and W. Han, "Development of steering control system for autonomous vehicle using geometry-based path tracking algorithm" *ETRI J.*, vol. 37, no. 3, pp. 617–625, Jun. 2015.
8. P. R. Ouyang, J. Acob, and V. Pano, "PID with sliding mode control for trajectory tracking of robotic system," *Robot. Comput.-Integr. Manuf.*, vol. 30, no. 2, pp. 189–200, Apr. 2014.
9. H. Yoshida, N. D. A. o. J. Department of Mechanical Systems Engineering, M. Omae, T. Wada, "Toward next active safety technology of intelligent vehicle," *J. Robot. Mechatron.*, vol. 27, no. 6, pp. 610–616, Dec. 2015.
10. K. D. Kim, "Collision free autonomous ground traffic: A model predictive control approach," in *Proc. ACM/IEEE 4th Int. Conf. Cyber-Phys. Syst. ICCPS*, Philadelphia, PA, USA, Apr. 2013, pp. 51–60.
11. D. Soudbakhsh and A. Eskandarian, "A collision avoidance steering controller using linear quadratic regulator," *SAE Tech. Paper 2010-01-0459*, Detroit, MI, USA, Apr. 2010.
12. Yimin Zhou; Gang Wang; Guoqing Xu; Guoqiang Fu, "Safety driving assistance system design in intelligent vehicles", *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, DOI: 10.1109/ROBIO.2014.7090740, 2014.
13. Loi Tran, Michael Hennessey, John Abraham, "Simulation and Visualization of Dynamic Systems Using MATLAB, Simulink, Simulink 3D Animation, and SolidWorks", <https://doi.org/10.1115/IMECE2011-62704>, August 1, 2012.