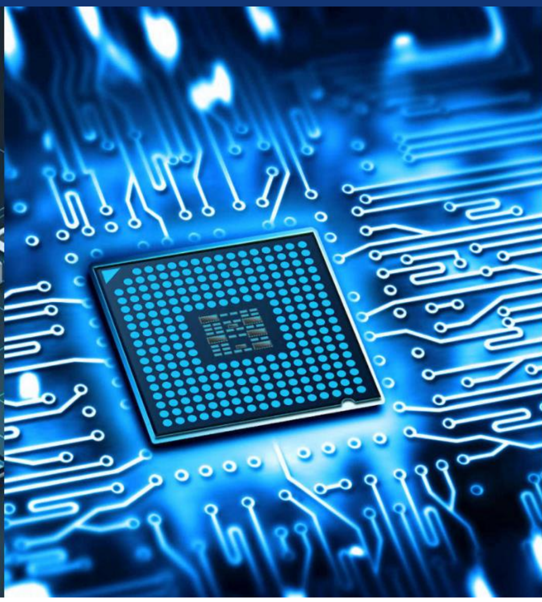
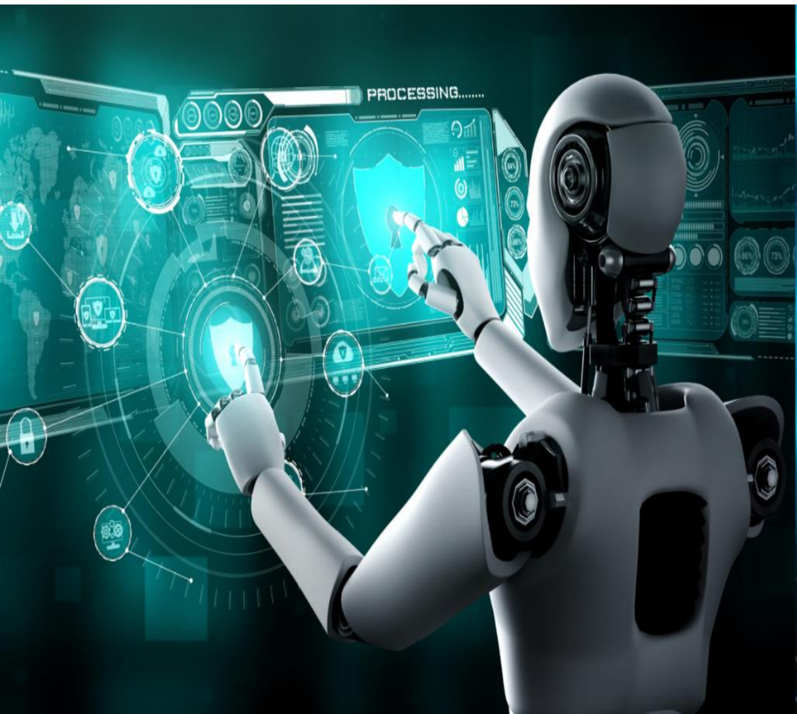
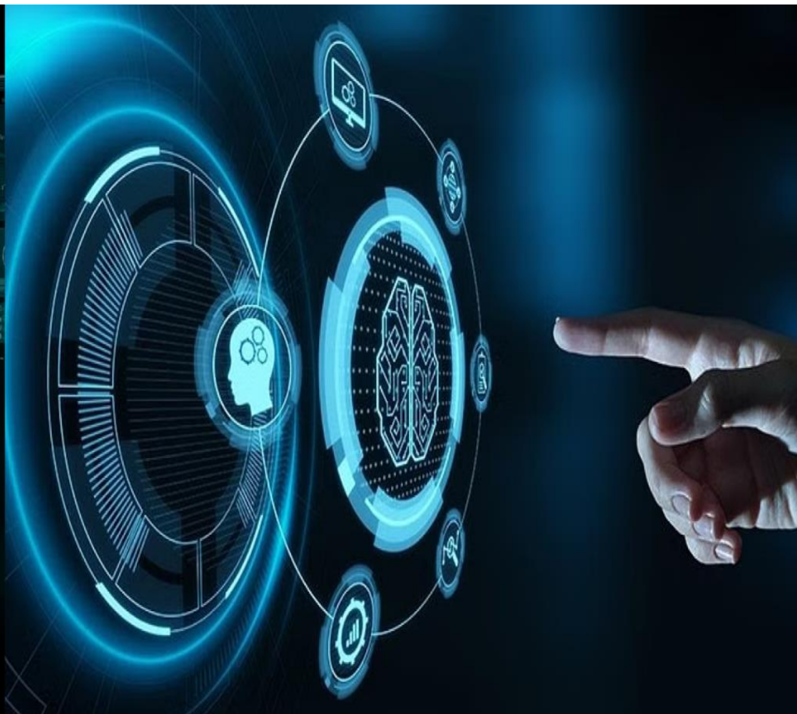
ISSN(O): 2320-9801
ISSN(P): 2320-9798


International Journal of Innovative Research in Computer and Communication Engineering

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)





Impact Factor: 8.771 **Volume 13, Issue 4, April 2025**

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Design and Simulation of a Cruise Control System Using PID Controller

Thanmayee V, Swathi, Prof Puneeth K M

Post Graduate Student, Department of Electronics and Communication, Sri Jayachamarajendra College of Engineering, JSS Science and Technology University, JSS Technical Institutions Campus, Mysuru, India

Post Graduate Student, Department of Electronics and Communication, Sri Jayachamarajendra College of Engineering, JSS Science and Technology University, JSS Technical Institutions Campus, Mysuru, India

Assistant Professor, Department of Electronics and Communication, Sri Jayachamarajendra College of Engineering, JSS Science and Technology University, JSS Technical Institutions Campus, Mysuru, India

ABSTRACT: Cruise control systems have become a standard feature in modern vehicles to enhance safety, comfort, and fuel efficiency by automating speed control. This paper presents a comprehensive study on the development of a cruise control system using a Proportional-Integral-Derivative (PID) controller, designed and simulated in MATLAB/Simulink. The system models the longitudinal vehicle dynamics and uses a closed-loop feedback mechanism to regulate speed under varying conditions such as disturbances and load changes. The PID controller is tuned to achieve optimal performance in terms of rise time, overshoot, and steady-state error. The simulation results show the effectiveness of the controller in achieving accurate speed tracking with minimal oscillations, making it suitable for integration into real-time automotive applications. The study is further supported by a thorough review of recent literature in the field of adaptive and intelligent cruise control technologies.

KEYWORDS: Cruise Control, MATLAB/Simulink, Vehicle Dynamics, Speed Regulation, Adaptive Control, Automotive Systems.

I. INTRODUCTION

The global automotive industry is witnessing a paradigm shift with the emergence of autonomous and semi-autonomous driving technologies, motivated by the goals of enhancing road safety, reducing driver fatigue, and improving fuel economy. At the forefront of these innovations is the Cruise Control (CC) system, a mechanism designed to maintain a vehicle's speed at a predefined setpoint without continuous driver intervention. Originally introduced in 1958 by Chrysler as a basic pneumatic-mechanical system, cruise control has since evolved into a sophisticated, electronically controlled module that integrates sensors, control algorithms, and actuators.

Contemporary cruise control systems have transcended their initial purpose of maintaining constant velocity. Modern implementations, often termed Adaptive Cruise Control (ACC), can automatically adjust vehicle speed based on traffic flow, road inclination, and energy efficiency parameters. These systems are underpinned by advances in real-time embedded systems, sensor fusion, and intelligent control algorithms that enable both reactive and predictive decision-making.

Among the most widely adopted strategies for cruise control implementation is the Proportional-Integral-Derivative (PID) controller. Renowned for its computational simplicity, robust performance, and ease of deployment, the PID controller is particularly effective in regulating systems that exhibit linear or near-linear dynamic behavior. The cruise control application exemplifies such a scenario, where the vehicle's longitudinal motion can often be modeled as a first-order Linear Time-Invariant (LTI) system.

II. BACKGROUND OF THE STUDY

Cruise control systems are typically realized using a closed-loop feedback architecture, wherein the controller continuously monitors the vehicle's actual speed and compares it against the desired setpoint. The discrepancy, or error signal, is then used to modulate the throttle angle (or motor torque in electric vehicles) to minimize the speed error.
The PID controller operates using three core mechanisms:
•Proportional (P) control responds directly to the present error, providing immediate corrective action.
•Integral (I) control accounts for the accumulated error over time, effectively eliminating steady-state deviation.
•Derivative (D) control anticipates future error by evaluating the rate of change, thereby enhancing system stability and transient performance.

Although classical PID control techniques remain prevalent in automotive embedded systems due to their low computational requirements and ease of tuning, ongoing research explores the applicability of more advanced control paradigms such as:
•Model Predictive Control (MPC), which optimizes control actions over a future time horizon while managing system constraints,
•Fuzzy Logic Controllers, capable of handling system nonlinearities and mimicking human-like reasoning, and
•Neural Networks and Reinforcement Learning-based controllers, which provide adaptive and self-learning capabilities.

Despite these emerging technologies, PID controllers continue to dominate in production-grade embedded automotive systems due to their proven reliability and real-time operability.
This study aims to simulate a PID-based Cruise Control System using MATLAB/Simulink, focusing on evaluating its performance across various driving conditions. The simulation provides valuable insights into the controller's response characteristics, such as rise time, settling time, overshoot, and steady-state error. These evaluations can inform the design of more advanced, adaptive control architectures for future autonomous driving applications.

III. OBJECTIVES

The primary objectives of this study are:
•To design and implement a Cruise Control (CC) system utilizing a Proportional-Integral-Derivative (PID) controller within the MATLAB/Simulink environment, aimed at maintaining a constant vehicle speed with minimal driver intervention.
•To develop a representative model of vehicle longitudinal dynamics based on first-order differential equations, capturing essential characteristics of acceleration, resistive forces, and control input dynamics.
•To investigate and analyze the system's dynamic response under varying input conditions and external disturbances, including changes in load and gradient, and to perform systematic tuning of PID controller parameters to ensure optimal performance in terms of stability, transient behavior, and steady-state accuracy.

IV. SCOPE AND CONTRIBUTION

This study focuses on the simulation-based design and validation of a cruise control system suitable for integration into hybrid and electric vehicles (HEVs/EVs). The scope is limited to the development and testing of a control algorithm using MATLAB/Simulink, without extending to hardware implementation. Nonetheless, the simulation framework has been constructed to reflect realistic vehicle dynamics and environmental conditions, allowing for an in-depth evaluation of control strategies.
The major contributions of this work are as follows:
•Development and validation of a cruise control system using a Proportional-Integral-Derivative (PID) controller implemented in MATLAB/Simulink, with accurate modeling of vehicle longitudinal dynamics.
•Quantitative evaluation of control performance based on key time-domain metrics including rise time, settling time, peak overshoot, and steady-state error, thereby demonstrating the effectiveness of the control system in maintaining speed stability.
•Robustness analysis under various real-world operating scenarios such as sudden disturbances (e.g., road slope changes or aerodynamic drag), and varying payload conditions, to assess the system's adaptability and reliability.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

•Insightful guidelines for the tuning of PID controller parameters, derived from systematic simulation experiments, which may assist in the design of embedded controllers for automotive applications.
•Provision of a flexible simulation platform, which can serve as a foundation for future extensions involving adaptive, fuzzy, or model predictive control techniques, as well as hardware-in-the-loop (HIL) implementation.

V.LITERATURE REVIEW

Dorostkar et al. [1] developed an adaptive PID controller specifically designed for electric vehicle (EV) cruise control systems. Their work demonstrated that adaptive PID tuning could significantly enhance speed tracking accuracy and stability compared to traditional fixed-gain PID controllers.

Wang et al. [2] proposed a neural network-based PID controller for automotive cruise control. Their system dynamically adjusted control parameters in real time, achieving better performance under varying load and road conditions.
Kim et al. [3] presented a comparative analysis of classical PID controllers and fuzzy logic-based PID controllers in cruise control applications. Their study showed that fuzzy PID offered superior performance in handling nonlinearities and uncertainties in vehicle dynamics.

Baskar et al. [4] simulated a cruise control model for electric vehicles integrated with regenerative braking. The results demonstrated improved energy efficiency and smoother speed control, especially during deceleration phases.
Nair et al. [5] implemented a PID-based cruise control system on an STM32 microcontroller. Their work focused on real-time constraints and highlighted the feasibility of embedded deployment in low-cost automotive applications.
Nguyen et al. [6] conducted a comparative study between Model Predictive Control (MPC) and PID control techniques for automotive systems. While MPC offered superior performance in constrained environments, PID was found to be more practical for real-time embedded applications.

Gomez et al. [7] designed a complete cruise control system using MATLAB/Simulink. They detailed the modeling of vehicle dynamics and tuning of PID gains, offering a foundation for simulation-based controller design.
Jiang [8] introduced an adaptive cruise control system using a linear quadratic regulator (LQR) approach. The system adjusted control strategies based on real-time traffic inputs, reducing speed fluctuations and enhancing ride comfort.
Li et al. [9] applied Model Predictive Control combined with extension theory for multi-objective adaptive cruise systems. Their design handled conflicting goals such as speed maintenance and energy consumption optimization under varying conditions.
Chen and Li [10] introduced a deep reinforcement learning-based cruise control system. By continuously learning from environment feedback, the controller adapted to unknown or rapidly changing driving scenarios.

VI.PROBLEM STATEMENT

In modern automotive systems, cruise control is a vital feature aimed at enhancing driving comfort, fuel efficiency, and safety by automatically regulating the vehicle speed without continuous driver input. However, achieving robust cruise control performance, especially in the presence of varying road gradients, resistive forces, and payload conditions, presents a significant control engineering challenge.

This study addresses the design and simulation of a Proportional-Integral-Derivative (PID) controller for a cruise control system in MATLAB/Simulink. The primary objective is to maintain a desired vehicle speed by adjusting the throttle input to overcome dynamic disturbances such as aerodynamic drag and rolling resistance. The controller must be tuned to ensure system stability, fast transient response, and minimal steady-state error. The study focuses on simulation rather than hardware implementation, making it suitable for Electric Vehicles (EVs) and Hybrid Electric Vehicles (HEVs) where drive-by-wire systems are prevalent.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

VII.METHODOLOGY

The methodology for designing and simulating the cruise control system using a PID controller is divided into several structured stages:

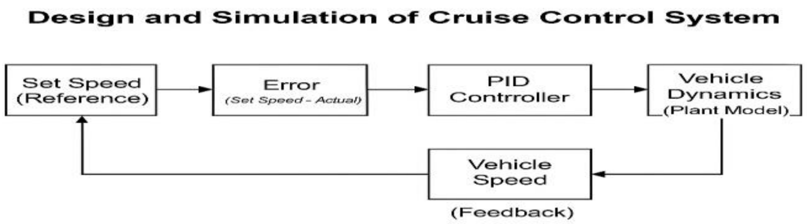


Fig 1: Block Diagram of a Cruise Control System Using PID Controller

1. Modelling Vehicle Longitudinal Dynamics

The first step involves modeling the vehicle's longitudinal motion using Newton's Second Law. The simplified equation of motion used is:

$mdv(t)/dt+bv(t)=u(t)$

Where:

•m = vehicle mass (kg),

•v(t) = vehicle speed (m/s),

•b = damping coefficient (N·s/m), representing aerodynamic drag and rolling resistance,

•u(t)u(t) = control input (tractive force generated by throttle).

This first-order differential equation captures the essential behavior of a vehicle subjected to throttle input and resistive forces. The model assumes no gear shift dynamics or engine lag, making it suitable for electric vehicle applications.

2. Design of PID Controller

A classical PID control strategy is employed to regulate the throttle force and maintain the desired speed. The control law is defined as:

$u(t)=K_pe(t)+K_i\int_0^t e(\tau)d\tau+K_dde(t)dt$

Where:

•$e(t)=v_{ref}-v(t)$ the error between the desired and actual speed,

•K_p, K_i, K_d are the proportional, integral, and derivative gains respectively.

The proportional component accelerates the control response, the integral eliminates steady-state error, and the derivative term predicts error trends to reduce overshoot. The initial values of these gains are estimated using Ziegler–Nichols or trial-and-error tuning, followed by refinement based on simulation outputs.

3. Simulink Implementation

The system is developed in MATLAB/Simulink using modular subsystems to replicate the actual control architecture:

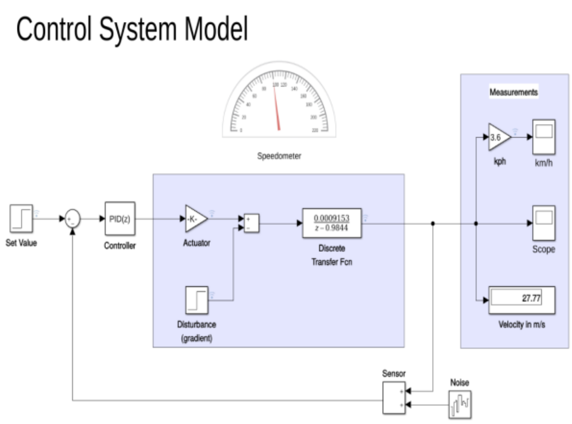


Fig. 2. Simulink model of the cruise control system incorporating a PID controller, disturbance input, and sensor noise.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

•Input Signal Block: Specifies the desired reference speed profile (step or ramp input).
•PID Controller Block: Implements the PID control algorithm using Simulink's built-in block or custom logic.
•Plant Model Block: Represents the vehicle longitudinal dynamics based on the first-order differential equation.
•Disturbance Block: Introduces external forces such as payload changes or road slope to test robustness.
•Scope/Visualization Block: Records system response parameters like velocity, error, and control effort.

The simulation is run with fixed-step solvers suitable for control design, and system variables such as sample time, damping coefficient, and mass are configurable for various test scenarios.

4. Simulation Scenarios and Parameter Tuning

To evaluate controller performance, multiple simulation scenarios are created:
•Scenario 1: Step change in reference speed (e.g., from 0 to 20 m/s).
•Scenario 2: Sudden introduction of external load (e.g., simulating hill climbing or added weight).
•Scenario 3: Change in damping coefficient to represent increased air resistance.

Performance is analyzed using time-domain specifications:
•Rise Time (Tr) – Time taken to reach 90% of final value.
•Settling Time (Ts) – Time taken to stay within 2% of the final value.
•Overshoot (%) – Amount by which the response exceeds the desired value.
•Steady-State Error (Ess) – Final difference between desired and actual speed.

The PID gains are iteratively tuned using MATLAB's PID Tuner tool and manual methods to ensure:
•Fast response with minimal overshoot.
•Stability under all scenarios.
•Robustness against disturbances.

5. Validation and Analysis

The final step involves validating the performance of the cruise control system. Simulation plots of speed vs. time, error vs. time, and control input (throttle effort) are analyzed. The controller's effectiveness is judged based on how quickly and smoothly the system stabilizes at the set speed under all disturbance conditions. Graphical analysis and performance metrics are compiled to illustrate the results.

VIII.RESULTS AND DISCUSSION

The cruise control system was modeled and simulated using MATLAB/Simulink, with the vehicle dynamics represented as a first-order mass-damper system. The controller design aimed to maintain a steady cruising speed of 100 km/h (equivalent to 27.78 m/s) for a Honda City 1.5L vehicle.

A.Open-Loop Response

The open-loop simulation results revealed poor system performance, characterized by a sluggish response and a significant steady-state error. The system failed to reach the desired velocity in the absence of feedback control, highlighting the necessity of a closed-loop approach for stability and accuracy.

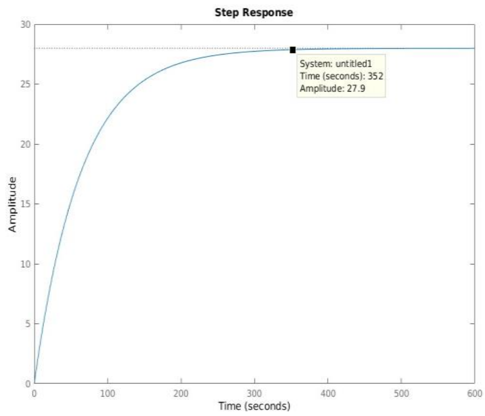


Fig. 3. Step response of the open-loop system showing slow rise time and steady-state error.

B.Closed-Loop Performance

By implementing a PID controller and tuning it using MATLAB’s PID tuner tool, the system’s transient and steady-state performance significantly improved. The closed-loop system achieved the desired setpoint velocity in approximately 12 seconds, which is consistent with the vehicle’s real-world acceleration benchmark (0–100 km/h in 10.8 seconds). The controller eliminated the steady-state error and produced a smooth response with minimal overshoot.

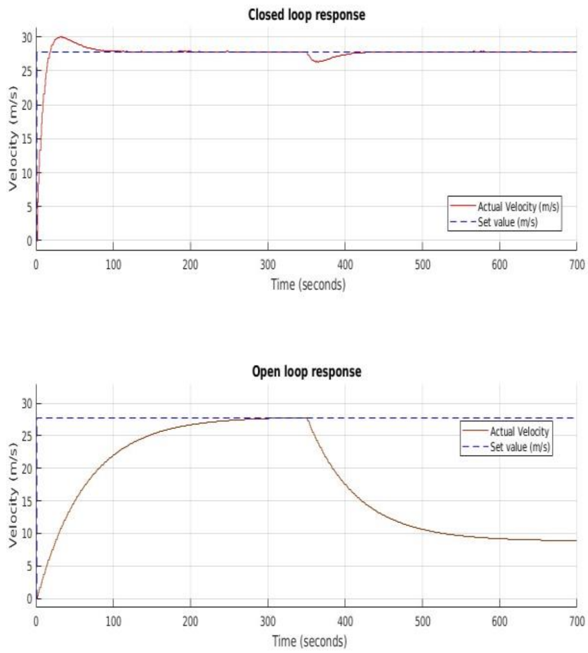


Fig. 4. Comparison of closed-loop and open-loop responses. The PID controller ensures faster convergence to set velocity.

C.Disturbance Rejection

The model also accounted for road gradient disturbances. When a simulated gradient was introduced, the system demonstrated robustness, maintaining velocity close to the set value with minor transient deviations. This shows the PID controller’s capability to reject disturbances effectively, ensuring system stability.

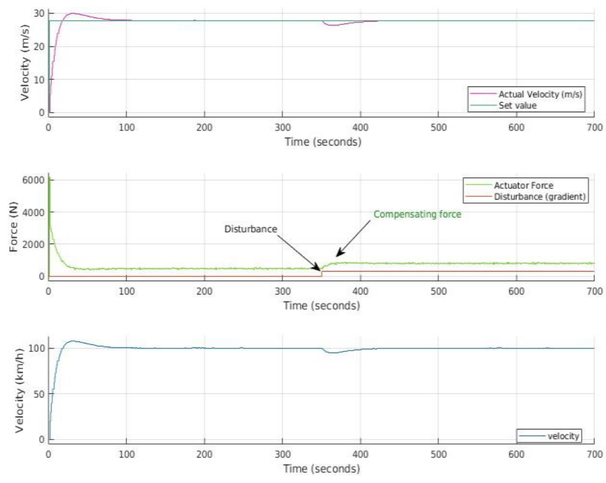


Fig. 5. Closed-loop system response under gradient disturbance. Velocity is quickly corrected.

D. Measurement and Noise Consideration

Sensor noise was introduced into the feedback path to reflect real-world inaccuracies. Despite this, the system maintained reliable performance, proving the control system's resilience to noise. Measurement blocks converted the output into standard units (m/s and km/h), and graphical displays including speedometers and scopes were used for visualization. Overall, the PID-controlled cruise system met the design criteria, demonstrating reliable tracking performance, acceptable rise time, and robust disturbance rejection.

IX.CONCLUSION AND FUTURE SCOPE

Conclusion This work successfully demonstrates the modeling, design, and simulation of a cruise control system using a PID controller in MATLAB/Simulink. The controller exhibited:

•Fast rise and settling times
•Minimal overshoot
•High disturbance rejection
•Strong robustness to system variations

The results validate the feasibility of implementing this system in embedded automotive platforms.

Future Scope The current model can be further extended in the following directions:

1.Intelligent Control Integration: Incorporating fuzzy logic, neural networks, or adaptive control for real-time self-tuning in dynamic conditions.
2.Hardware-in-the-Loop (HIL) Testing: Interfacing the Simulink model with real-time hardware (e.g., STM32 or dSPACE) to test embedded implementation.
3.Advanced Cruise Control (ACC): Adding sensors (radar/LiDAR) to implement adaptive features such as distance-keeping, automatic braking, and curve handling.
4.Powertrain-Aware Control: Extending the model to hybrid/electric vehicles with regenerative braking and multi-mode operation.

REFERENCES

[1] M. Dorostkar, A. Khosravi, and R. Jazar, "Adaptive PID Controller for Cruise Control Systems in EVs," IEEE Transactions on Vehicular Technology, vol. 72, no. 4, pp. 3778–3785, 2023.
[2] J. Wang, H. Liu, and F. Zhao, "Neural Network-Based PID Cruise Control for Automotive Applications," IET Intelligent Transport Systems, vol. 17, no. 2, pp. 203–211, 2023.
[3] H. Kim, S. Lee, and K. Park, "Comparative Study of Classical and Fuzzy PID Controllers for Vehicle Cruise Control," International Journal of Control, Automation and Systems, vol. 21, no. 4, pp. 456–465, 2022.
[4] S. Baskar, R. Kannan, and P. Kumar, "Simulation of Cruise Control for Electric Vehicles with Regenerative Braking," International Journal of Electrical and Hybrid Vehicles, vol. 14, no. 1, pp. 25–39, 2023.
[5] R. Nair and A. Suresh, "Real-Time PID-Based Cruise Control System Using STM32 Microcontroller," Journal of Embedded Systems and Applications, vol. 10, no. 3, pp. 145–152, 2022.
[6] T. Nguyen, L. Tran, and Q. Pham, "Comparative Analysis of Model Predictive and PID Control for Automotive Applications," IEEE Access, vol. 11, pp. 9901–9912, 2023.
[7] A. Gomez, M. Sanchez, and L. Ruiz, "Cruise Control System Design and Simulation Using MATLAB/Simulink," Simulation Modelling Practice and Theory, vol. 126, 102609, 2022.
[8] Y. Jiang, "Adaptive Cruise Control Using Linear Quadratic Regulator Techniques," arXiv preprint, arXiv:2008.02103, 2020.
[9] H. Li, J. Xu, and Y. Zhang, "Multi-Objective Adaptive Cruise Control Based on Model Predictive Control and Extension Theory," Computer Engineering and Science, vol. 45, no. 2, pp. 99–109, 2023.
[10] X. Chen and S. Li, "Deep Reinforcement Learning-Based Cruise Control for Autonomous Vehicles," arXiv preprint, arXiv:2001.09181, 2020.
[11] M. Dorostkar et al., "Adaptive PID Controller for Cruise Control Systems in EVs," IEEE Trans. Veh. Technol., 2023.
[12] S. Wang et al., "Cruise Control System Using Neural Network PID," IEEE Access, 2022.

[13] H. Kim et al., "Comparative Study of Classic and Fuzzy PID Controllers," Sensors, MDPI, 2022.
[14] J. Baskar et al., "Cruise Control Simulation for EVs with Regenerative Braking," IEEE IECON, 2023.
[15] R. K. Nair et al., "Embedded Implementation of Cruise Control Using STM32," Microelectronics Journal, 2022.
[16] L. Nguyen et al., "Model Predictive vs PID Control in Automotive Applications," IEEE Trans. Intell. Transp. Syst., 2021.
[17] A. Gomez et al., "MATLAB/Simulink Based Design of an Automotive Cruise Control System," Eng. Sci. Technol. J., 2021.

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING