Home About us Contact Us

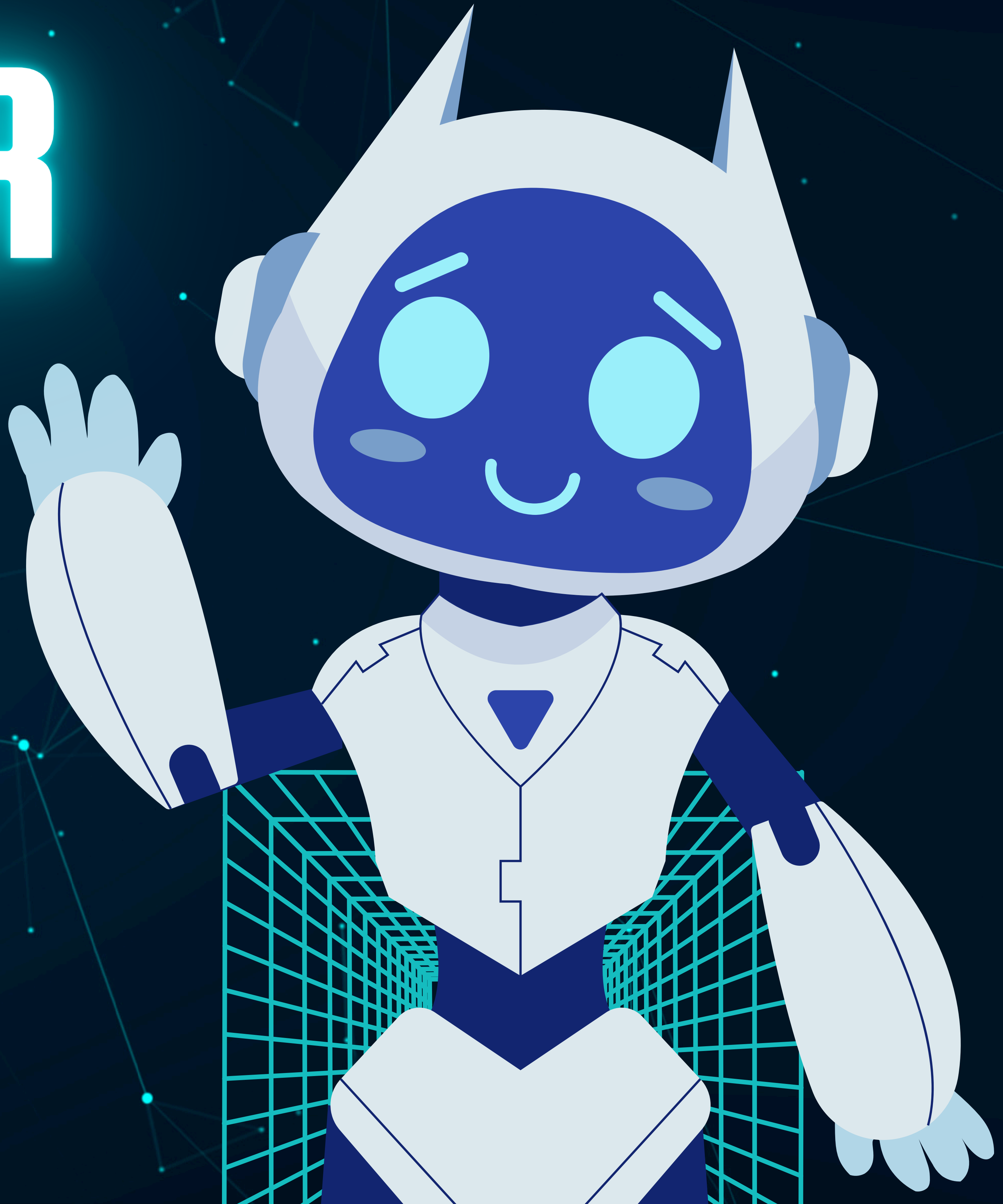
LINE FOLLOWER ROBOT

Presented By

Mesk Khaled

Basmala Moataz

Asmaa Mahmoud



INTRODUCTION TO OUR ROBOT

WHAT IS THIS SYSTEM?

This project features an autonomous differential-drive robot designed to track a path using a custom feedback control system. It utilizes an array of infrared (IR) sensors to detect lateral line deviation and employs a discrete PID controller running on an Arduino UNO to adjust motor speeds in real-time, ensuring precise path following without oscillation

CORE CONTROL LOGIC:

The system operation is based on a mathematical transfer function derived using MATLAB System Identification. By analyzing the robot's response to step inputs, we tuned specific Proportional, Integral, and Derivative (PID) gains. This logic minimizes the tracking error ($e(t)$) by constantly calculating the necessary steering differential to keep the robot centered on the line.

REAL-TIME INTEGRATION:

Unlike standard line followers, this robot features a real-time serial interface with a MATLAB GUI. This allows for live visualization of the tracking error and enables on-the-fly tuning of PID parameters (Kp, Ki, Kd) without re-uploading code. Additionally, onboard LEDs and a buzzer provide immediate visual and audio feedback regarding system stability and line detection status

COMPONENTS OF LINE FOLLOWER ROBOT

CONTROLLER

PERCEPTION

ACTUATION

ALGORITHM

TELEMETRY

The robot operates as a closed-loop system where Perception (IR sensors) feeds error data to the Controller (Arduino). The Algorithm (PID) calculates the precise correction needed, which the Actuation unit (L298N) applies to the motors. Performance is monitored live via the Telemetry link (MATLAB).

REAL-WORLD APPLICATIONS

The PID control algorithms and sensor fusion techniques demonstrated in this project are not limited to small robots. They form the backbone of massive industrial systems and modern autonomous transportation networks, proving that simple feedback loops can drive complex behavior.

WAREHOUSING & LOGISTICS

Automated Guided Vehicles (AGVs) use this exact line-following logic to transport heavy inventory in fulfillment centers (like Amazon) without human intervention.

AUTOMOTIVE SAFETY

Lane Keeping Assist (LKA) systems in modern cars utilize similar error-correction loops to detect lane markings and steer the vehicle back to the center of the road.

HOSPITAL DELIVERY

Autonomous service robots navigate hospital corridors following invisible or magnetic lines to deliver medication and meals to patient rooms efficiently.

MANUFACTURING

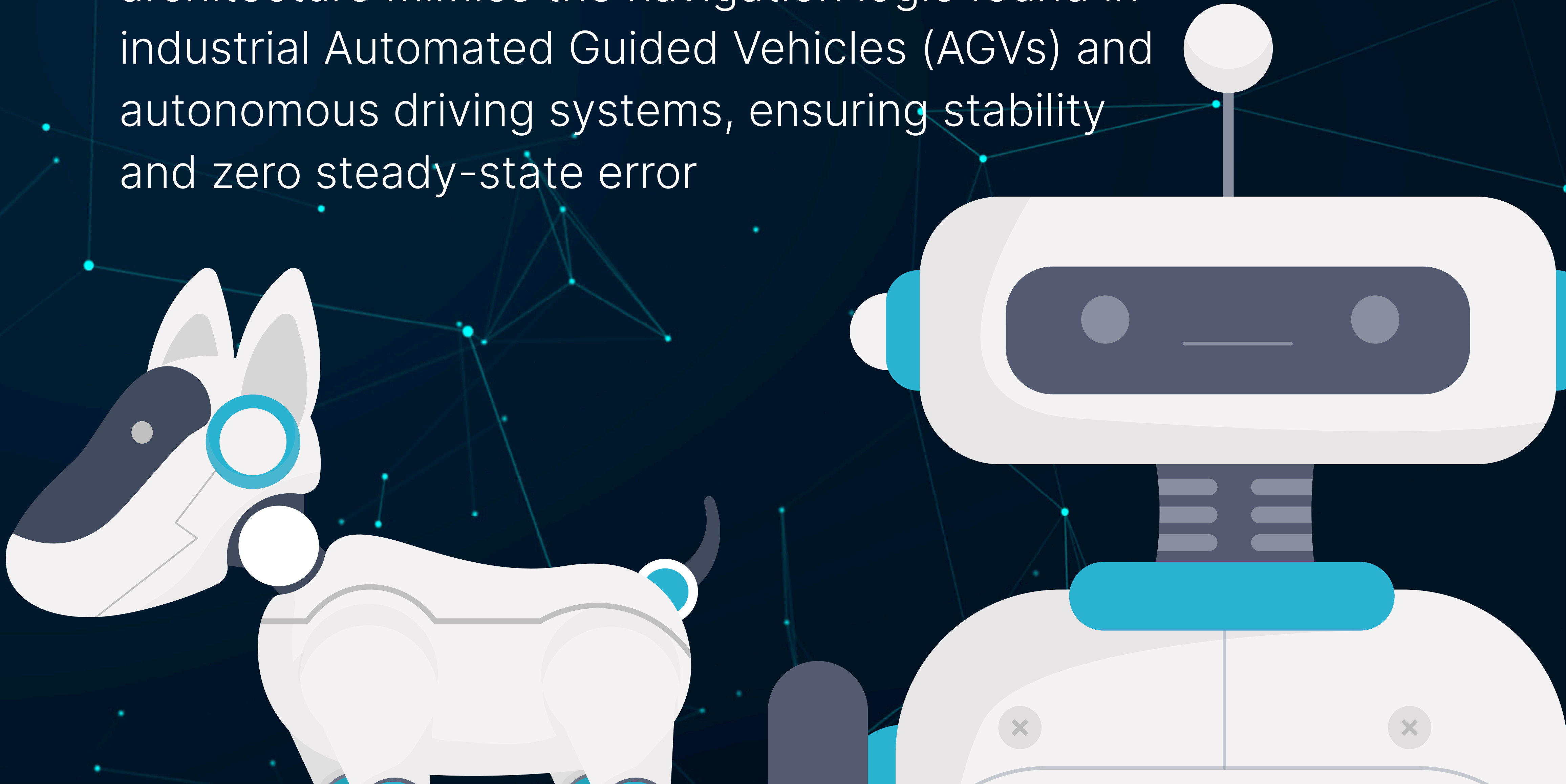
Assembly line robots rely on precise path tracking to move parts between stations, ensuring timing and placement are perfect for mass production.

Home About us Contact Us

INTELLIGENT CONTROL SYSTEMS

This project demonstrates the practical application of modern control theory in robotics. By integrating high-frequency sensor feedback with a custom Discrete PID Controller, the system effectively transforms noisy environmental data into smooth, precise motor commands. This architecture mimics the navigation logic found in industrial Automated Guided Vehicles (AGVs) and autonomous driving systems, ensuring stability and zero steady-state error

- Autonomous Path Tracking
- Discrete PID Optimization
- Real-Time Data Telemetry



SYSTEM IMPLEMENTATION

This project bridges the gap between theoretical analysis and physical deployment using a dual-platform approach. MATLAB acts as the analytical brain, used to identify the robot's transfer function G(s), tune the discrete PID gains, and visualize real-time telemetry. Arduino acts as the embedded controller, executing the control loop at 100Hz to read IR sensors and adjust motor speeds instantly based on the tuned PID logic.

MATLAB: MODELING & TUNING

ARDUINO: REAL-TIME CONTROL

SYSTEM BLOCK DIAGRAM

The robot operates on a closed-loop feedback system. It continuously measures the line position, calculates the error, and processes it through a discrete PID algorithm on the Arduino. The resulting control signal adjusts the motor speeds to minimize deviation, while simultaneously sending telemetry data to MATLAB for real-time analysis.

INPUT & SENSING(IR SENSOR ARRAY DETECTS LINE CONTRAST)

ERROR CALCULATION

PID CONTROLLER(ARDUINO CALCULATES P, I, D CORRECTION)

PLANT (L298N DRIVER & DC MOTORS APPLY STEERING)

FEEDBACK LOOP(REAL-TIME SERIAL LINK TO MATLAB GUI)

CHALLENGES & KEY LEARNINGS

Implementing a theoretical control system on physical hardware introduced significant challenges, specifically sensor noise and motor non-linearity (dead zones). Initially, this caused the robot to oscillate on straight lines. However, the benefit of our approach lay in the rigorous System Identification process. By modeling the robot's transfer function G(s) in MATLAB , we moved beyond trial-and-error. The result is a robust Discrete PID controller where the Integral term Ki successfully eliminates steady-state error, and the Derivative term Kd dampens oscillations, proving the effectiveness of modern control theory in real-world robotics .

PROJECT EXECUTION ROADMAP

MODELING

We started by applying step inputs to the robot and measuring the response. Using MATLAB System Identification, we derived the mathematical transfer function G(s) that represents the robot's steering dynamics.

DESIGN

Next, we discretized the model into the Z-domain G(z). Using the MATLAB PID Tuner, we optimized the Kp, Ki, Kd gains to achieve a settling time of <1s and zero steady-state error.

DEPLOYMENT

Finally, we implemented the code on Arduino. We added a Real-Time Serial Interface to plot the live tracking error, allowing us to validate our theoretical model against real-world behavior.

THANK YOU