# <u>Autonomous systems // Final project // Yael Keidar 314711771</u>

**Introduction-**

In the field of autonomous systems, designing and implementing self-navigating vehicles is crucial for advancing robotic navigation technologies. This project focuses on creating an autonomous car capable of performing two fundamental tasks: line following and obstacle avoidance. These tasks are essential for effective and safe autonomous navigation. Utilizing an Arduino microcontroller as the central processing unit, the car is programmed to follow a path by detecting a line on the ground while simultaneously identifying and navigating around obstacles encountered along its route.

This project represents a practical application of the theoretical concepts learned throughout the course. It leverages the skills and techniques acquired to address real-world challenges in autonomous vehicle design. The project includes the implementation, and testing phases, each of which is critical for achieving a fully functional autonomous navigation system.

During the course, we acquired valuable tools and techniques, including:

<u>1. Using Encoders</u>: Encoders provide detailed data about the motor's performance, enabling us to create graphs of X-Y coordinates and velocity-time (V-t). These graphs were instrumental in analyzing the vehicle's motion and performance.

<u>2. Sensor Integration</u>: We learned to effectively use sensors to detect obstacles in the vehicle's path. This capability is essential for dynamic obstacle avoidance and ensuring the vehicle's safe navigation.

By applying these tools and techniques, we designed a system capable of precise line following and adaptive obstacle avoidance. The following sections of this report detail the project's design, implementation, and testing processes. We will explore the challenges faced during development and the solutions implemented to ensure the autonomous vehicle's reliable and efficient operation.

**Experiment-**

During the lab sessions, we applied the concepts and techniques acquired from the seminars to develop our autonomous car. The project initiated with the implementation of a fundamental line-following algorithm. Using sensors, we detected the contrast between the line and the surrounding surface, enabling the car to stay precisely on its designated path. This initial setup provided a solid foundation for further enhancements.

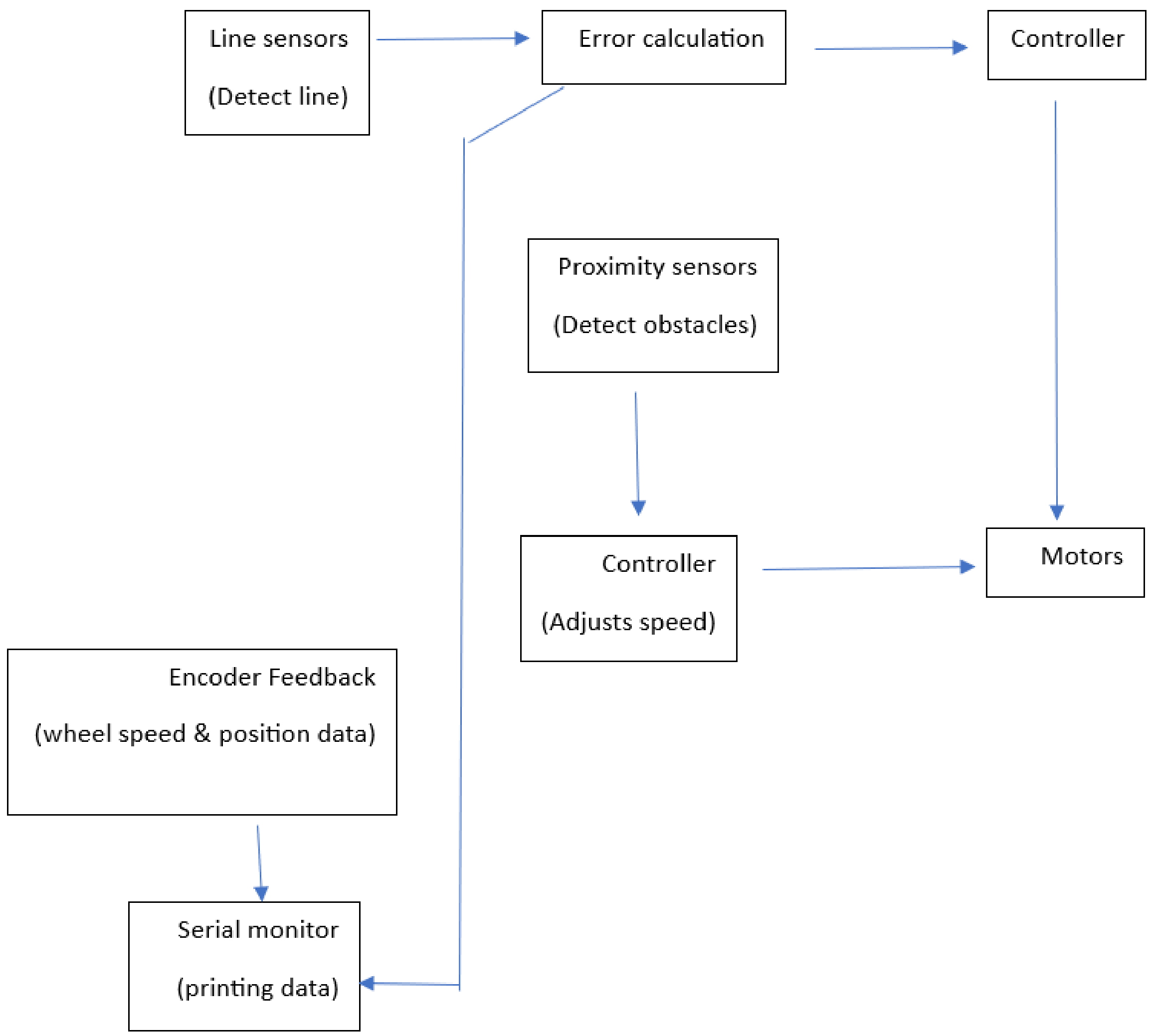
Expanding on this base, we enhanced the functionality of the code to incorporate obstacle detection and avoidance capabilities. We integrated proximity sensors, which are distance sensors specifically designed to monitor the car's surroundings. These sensors were programmed to detect obstacles in the car's path. Upon detecting an obstacle, the code triggered a series of predefined maneuvers, such as turning or reversing, allowing the car to navigate around the obstruction and then return to its line-following task. This dynamic response ensured that the car could handle unexpected obstacles effectively.

In addition to obstacle avoidance, we incorporated encoder readings into our system. Encoders provide valuable data on the rotational speed and position of the car's wheels, offering real-time feedback on its movement. This data was crucial for controlling the car's speed and ensuring accurate navigation. By analyzing encoder data, we generated X-Y position graphs and velocity-time

(V-t) graphs, which revealed the car's movement patterns and performance characteristics. These visualizations were instrumental in fine-tuning the control algorithms and improving the car's overall responsiveness and accuracy.

The lab sessions were characterized by iterative testing and refinement. We continuously updated the code based on test results, assessed the car's performance under various conditions, and analyzed the data from sensors and encoders. This iterative process allowed us to identify and address issues, optimize the control strategies, and enhance the reliability and efficiency of the autonomous navigation system.

<u>Block diagram-</u>



As illustrated in the block diagram, the system operates as follows-

Line Sensors: These sensors detect the contrast between the line and the surface, enabling the car to follow the line accurately. The data from these sensors is used to compute the error, which represents how far the car is from the line’s center.

Error Calculation: The error calculation determines the car’s deviation from the line and sends this error value to the Controller. This calculation is also printed to the Serial Monitor for analysis purposes.

PID Controller: The controller receives the error value and adjusts the motors' speed to correct the car’s position. This adjustment ensures that the car stays on course by modifying the speed of the left and right motors based on the calculated error.

Proximity Sensors: These sensors monitor the car's surroundings to detect obstacles. When an obstacle is detected, the data is sent to the Controller, which temporarily overrides the controller to adjust the motors' speed and navigate around the obstacle.

Motors: The motors are driven based on commands from the Controller and the obstacle avoidance logic. They adjust the car's movement to follow the line and navigate around obstacles as needed.

Encoder Feedback: Encoders provide data on the rotational speed and position of the wheels, which is used to track the car’s movement. This data is also printed to the Serial Monitor for additional analysis.

Serial Monitor: Outputs data from the error calculation and encoder feedback, allowing for real-time monitoring and debugging of the system’s performance.

This integrated approach ensures that the car can effectively follow a line, avoid obstacles, and provide comprehensive feedback for analysis and improvement.

**Summary-**

This project demonstrated the application of autonomous navigation principles by developing a car capable of following a line and avoiding obstacles using an Arduino microcontroller. Throughout the project, we applied the skills and knowledge gained from the course, including sensor integration, encoder data analysis, and real-time control systems.

The implementation involved modifying a basic line-following algorithm to incorporate obstacle detection and avoidance, which was achieved by integrating additional sensors and refining the control logic. The use of encoders allowed for precise monitoring of the car's movement, enabling the creation of velocity-time (V-t) graphs that provided valuable insights into the system's dynamics.

During the lab, we encountered a few difficulties:

1. Understanding how to implement a function in the code to ensure that the car could detect and avoid obstacles.

2. Correctly integrating and working with the sensors to achieve reliable data.

3. Properly implementing the encoder functionality in the code for accurate measurement and control.

However, while the overall results were positive, we encountered difficulties specifically with the encoder implementation. The encoder data did not perform as expected, leading to challenges in achieving precise control and consistent movement. This issue affected the accuracy of the position and velocity measurements and highlighted the need for further refinement in the encoder integration.

Overall, this project provided hands-on experience in autonomous systems design, reinforcing the theoretical concepts covered in the course. The successful completion of the project emphasized the importance of iterative testing and data analysis in developing robust and efficient autonomous navigation systems.