Faculty of Engineering and Applied Sciences
Department of Electronics and Computer Engineering

ECEN315: Fundamentals of Control

# Robot Line Follower

Made By:
Ahmed Ayman 202000689
Mohab Ali 202000083
Mohamed Shawky 202000218
Zeyad Hisham 202002027
Mohab Hassan 202001315

- # Introduction

A robot line follower is an autonomous vehicle that is designed to follow a predefined path, typically a black line on a white surface or vice versa. These robots use sensors to detect the line and adjust their movement accordingly, allowing them to navigate through a course without human intervention.

- # Methodology

Robot line followers utilize a combination of sensors, microcontrollers, and motors to navigate along a defined path. The key component is the sensor system, typically consisting sensors positioned at the bottom of the robot. These sensors continuously monitor the line, detecting the difference in light intensity between the line and the surface. For example, if the line is black on a white surface, the sensors will detect a lower light intensity on the black line compared to the white surface. The sensors send this data to the microcontroller, which acts as the "brain" of the robot However the software component of a line follower robot is responsible for processing the data collected by the sensors mounted on the robot. The software uses algorithms to interpret the visual data and make decisions about the robot's movement. For instance, the robot may use a PID controller algorithm to adjust its speed and direction based on the distance from the line. The software also enables the robot to adapt to changes in the line's path. On the other hand, the hardware component of a line follower robot consists of the physical parts that enable the robot to move and sense its environment. The robot's chassis, motors, and wheels work together to provide the necessary mobility, while sensors such as infrared, ultrasonic, or cameras detect the line and provide feedback to the software. The hardware also includes the power source, such as batteries or a power cord, and the control systems that regulate the robot's movements. The quality and precision of the hardware components are crucial in ensuring the robot's accuracy and reliability.

- ## Hand analysis

Since our function is $5^{th}$ order we didn't do performance calculations using hand analysis

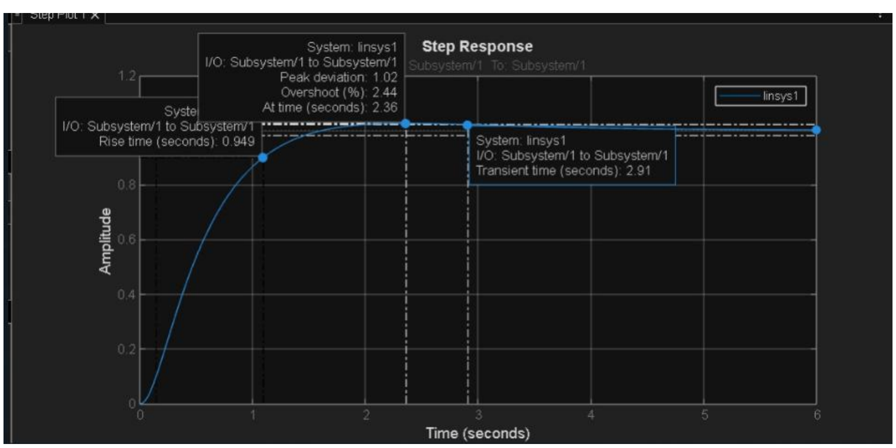
- ## Matlab Code

```
Kp = 0.12;

Ki = 0.1;

Kd = 0.02;

numerator = 800;

denominator = [1, 15.8, 62, 40];

sys = tf(numerator, denominator);

C = pid(Kp, Ki, Kd);

sys_cl = feedback(C * sys, 1);

step_response_cl = stepinfo(sys_cl);

disp('PID Controller Gains:');

disp(['Kp: ', num2str(Kp)]);

disp(['Ki: ', num2str(Ki)]);

disp(['Kd: ', num2str(Kd)]);



disp('Closed-Loop Response Characteristics:');

disp(['Rise Time: ', num2str(step_response_cl.RiseTime), ' seconds']);

disp(['Overshoot: ', num2str(step_response_cl.Overshoot), ' %']);
```

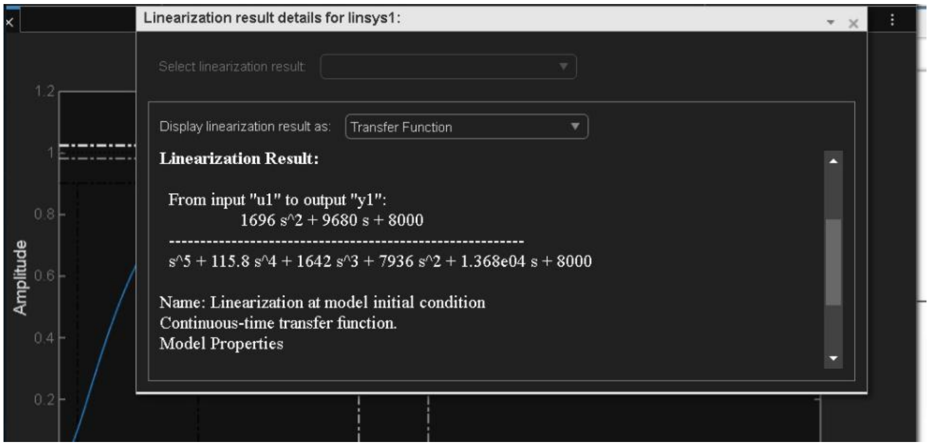
```
disp(['Settling Time: ', num2str(step_response_cl.SettlingTime), ' seconds']);


subplot(1,1,1);
step(sys_cl);
title('Step Response of Closed-Loop System');
xlabel('Time (s)');
ylabel('Amplitude');
```

- Simulink of robot line follower


- Simulink Response

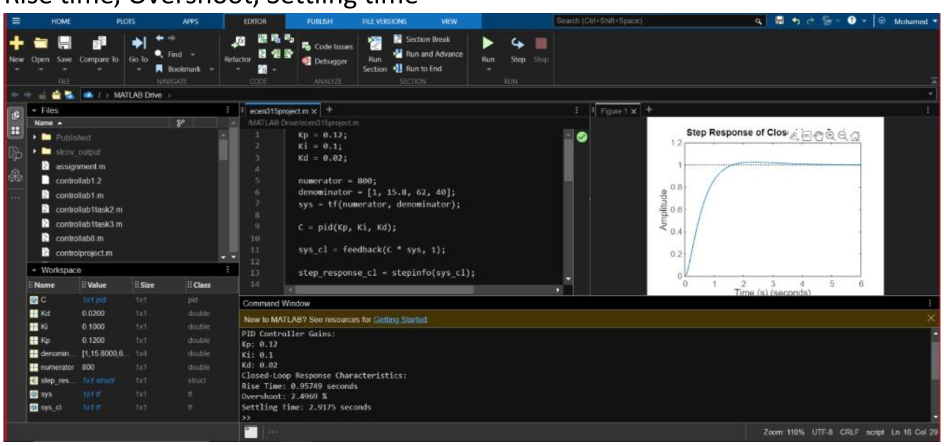


- Transfer Function




- Transient Response

Rise time, Overshoot, Settling time



Here the PID Controller Gains from the simulation we did was:

Kp: 0.12

Ki: 0.1

Kd: 0.02

Closed-Loop Response Characteristics:

Rise Time: 0.95749 seconds

Overshoot: 2.4969 %

Settling Time: 2.9175 second

- Software and hardware comparison

The system is represented by a transfer function with the given numerator and denominator coefficients.

First, the PID controller gains are set: Kp = 0.12, Ki = 0.1, and Kd = 0.02. These values determine the proportional, integral, and derivative components of the controller, respectively. The transfer function of the system is created using the tf function, and the PID controller is defined using the pid function.

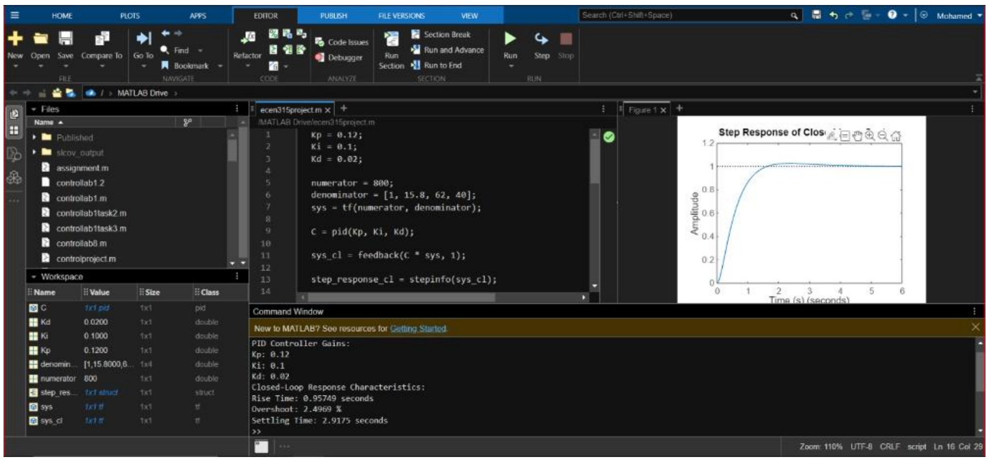
Next, the closed-loop system is formed by combining the system transfer function and the PID controller using the feedback function. This step incorporates the feedback loop and creates the closed-loop transfer function.

To analyze the performance of the closed-loop system, the step response characteristics are obtained using the stepinfo function. The rise time, overshoot, and settling time are extracted from the step response and displayed in the command window.

Finally, a subplot is created to plot the step response of the closed-loop system. The step function is used to generate the step response, and the plot is labeled with a title, x-axis label

(Time), and y-axis label (Amplitude). In the end both the software and hardware come up with the same result.

## Hardware




## Software