Control
Lab Three

Prepared By

Ahmed Aly Gamal El-Din El-Ghannam

Electronics and Communications - Level 4
ID: 19015292

December 2023

Contents

Listings

1 Introduction

This report showcases the design methodology for a PID controller to accomplish automatic cruise control for the vehicle system shown in figure 1. All the MATLAB codes used to produce this result can be found in the lab's Github Repository.

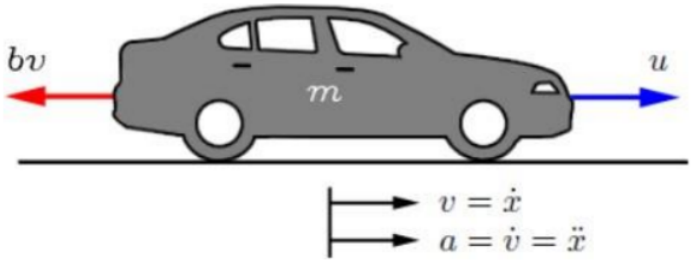

Figure 1: Free Body Diagram of a Vehicle

2 Uncompensated System Performance

The open-loop system is represented by the following transfer function:

$$\mathrm{H(s)} = \frac{\mathrm{V(s)}}{\mathrm{U(s)}} = \frac{1}{\mathrm{ms} + \mathrm{b}} \tag{1}$$

where ***m*** is the vehicle's mass and ***b*** is the damping coefficient with values 1000kg and 50N.s/m respectively.

Code snippet 1 displays the step response, as well as multiple other parameters.

```
%% SYSTEM WITHOUT CONTROLLER
% Mass of Car
m = 1000;
% Damping Coefficient
b = 50;
% Open-loop TF of the system
TF_OL_vOverU = tf([0 1], [m b]);

step(TF_OL_vOverU)
stepinfo(TF_OL_vOverU)
```

Code Snippet 1: Open-loop Performance

As shown in figure 2, the step response is not only extremely slow, but also highly inaccurate. The step info shown in figure 3 highlights this in great detail.

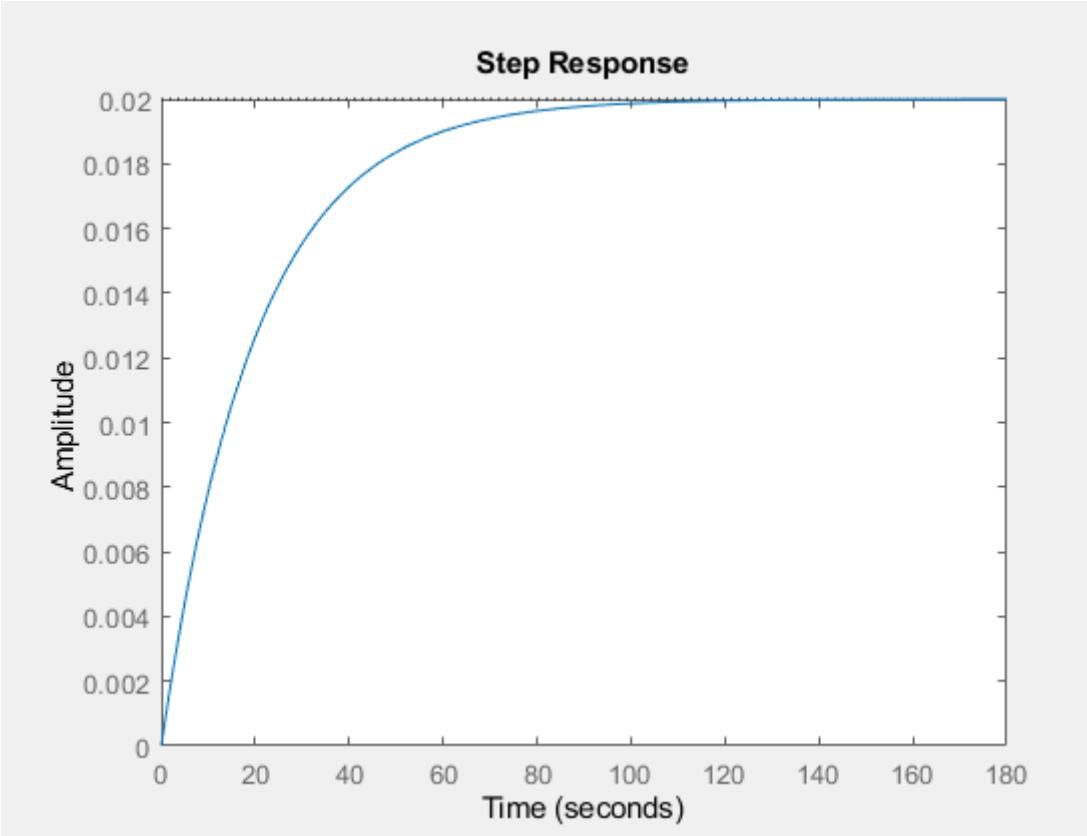

Figure 2: Open-loop System - Step Response

```
    RiseTime: 43.9401
SettlingTime: 78.2415
 SettlingMin: 0.0181
 SettlingMax: 0.0200
   Overshoot: 0
  Undershoot: 0
        Peak: 0.0200
    PeakTime: 210.9168
```

Figure 3: Open-loop System - Step Response Parameters' Values

3 PID Controller Design

It is required to design a PID controller with the following specifications:

- Rise Time < 5 seconds
- Maximum Overshoot $< 10\%$
- Steady-state Error $< 2\%$

The design philosophy is constrained by the values of 3 parameters and their effect on each of the required specifications: K_p, K_i, and K_d.

	RISE TIME	OVERSHOOT	SETTLING TIME	Steady ERROR
K_p	Decrease	Increase	Small Change	Decrease
K_i	Decrease	Increase	Increase	Eliminate
K_d	Small Change	Decrease	Decrease	No Change

Figure 4: Effects of Changing K_p, K_i, and K_d

The two parameters that are way off their required values are rise time and steady-state error. Therefore, the first course of action was to increase K_p and/or K_i and compensate the unintended increase in overshoot by slightly increasing the value of K_d.

Through rigorous testing, the ratio in equation 2 was found to not only accomplish the required specifications with rational values for K_p, K_i, and K_d; but also to immensely enhance the system's performance at higher magnitudes.

$$\mathrm{K_p} : \mathrm{K_i} : \mathrm{K_d} = 100 : 4 : 1 \tag{2}$$

Figure 5 shows the parameters entered to PID Controller Simulator. A PID controller is added with the aforementioned ratio multiplied by 4.75. These are the lowest and most balanced integer values that achieve the required specifications. Higher values can grant better performance but at a higher cost.

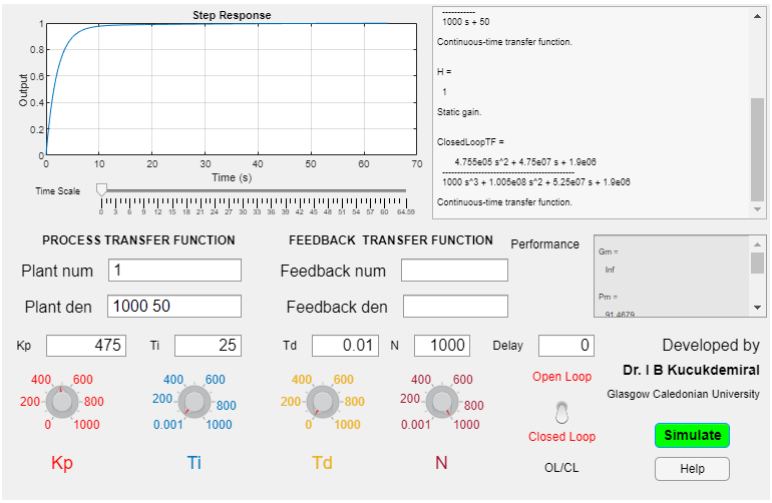

Figure 5: Open-loop System - PID Controller Simulator

Figures 6 and 7 showcases the closed-loop system step response and parameters after adding the PID controller.

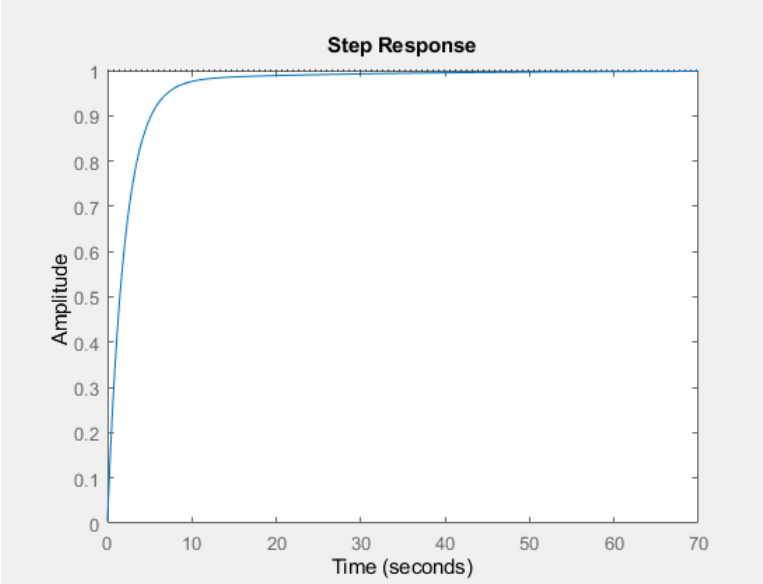

Figure 6: Closed-loop System - Step Response

```
    RiseTime: 4.9505
SettlingTime: 11.2548
 SettlingMin: 0.9030
 SettlingMax: 0.9987
   Overshoot: 0
  Undershoot: 0
        Peak: 0.9987
    PeakTime: 74.2126
```

Figure 7: Closed-loop System - Step Response Parameters' Values

Code snippet 2 was used to generate the previous figures. It is noticed that all the design criteria have been met successfully.

```
%% ADDING PID CONTROLLER
Kp = 500
Ki = 20
Kd = 5

TF_OL_PIDController = pid(Kp, Ki, Kd);

TF_CL_SystemWithController = feedback(TF_OL_PIDController *
   TF_OL_vOverU, 1);

step(TF_CL_SystemWithController)
stepinfo(TF_CL_SystemWithController)
```

Code Snippet 2: Open-loop Performance