UDC Control Sub-Team Recruitment Task

February 10, 2024

Instructions

This task is designed to assess your skills in control system engineering and programming. Attached, you will find relevant learning materials that should assist you in addressing the task and interview questions.

1. First start by reading the task questions to have an idea about the questions you are going to face.

2. Review Learning Materials:Begin by carefully reviewing the attached learning materials. Pay close attention to the concepts and the examples.

3. Independent Research:If a question requires additional information not covered in the provided materials, feel free to perform independent research. Clearly indicate if information is obtained from external sources and provide proper references.

4. Provide explanations for your answers. We are interested in your thought process and understanding of the underlying principles.

5. Submit your response in a zip file containing all the required files and a report explaining your work on the following mail :udc.learning@gmail.com

6. Deadline for task submission :

Additional notes

- Use clear and concise language in your responses.

- If you encounter difficulties with any question, provide your best attempt along with a brief explanation of your thought process.

- Some of the material does not have a task tackling them but you have to study all of it as you will be asked about it in the interview.

- Do your best , if you did not understand a topic completely just answer with what you understand.

We appreciate your effort and dedication to completing this task. Feel free to reach out if you have any clarifications or questions.

Tasks

I. Physical Modeling

Landing gear suspension systems are designed to absorb and dissipate the kinetic energy of landing impact, which reduces the impact loads transmitted to the air-frame. In this question, you're required to model a suspension system using Simulink. A suspension system could be simplified to a double spring-damper system.

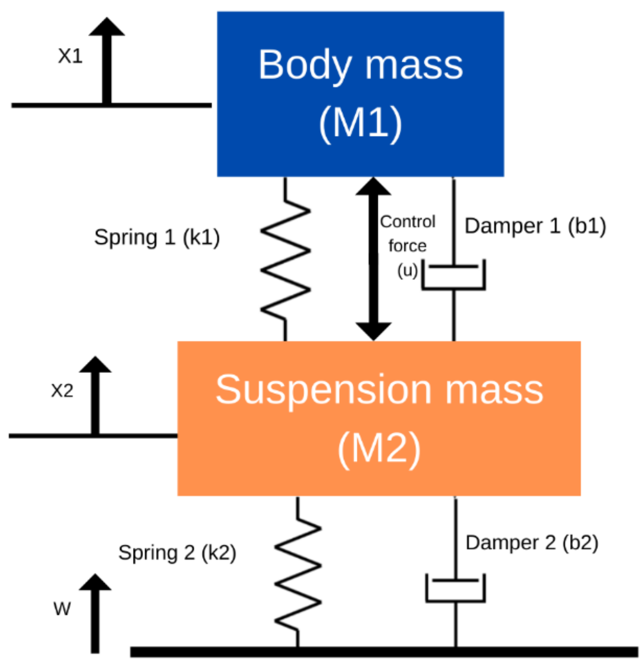

Figure 1: Suspension System Model

Use the following equations to implement the suspension system model on Simulink:

$$\int\int \frac{d^2x}{dt^2} = \int \frac{dx}{dt} = x$$
$$\frac{1}{M}\sum F = \frac{d^2x}{dt^2}$$

Springs and dampers induce forces that affect each mass. Forces due to springs are functions of spring constant and displacement difference, while forces due to dampers are functions of damper constant and velocity difference.

Below are the equations of forces that affect both the body mass and the suspension mass:

$$F_{spring1} = k_1(X1 - X2)$$
$$F_{damper1} = b_1(V_1 - V_2)$$
$$F_{spring2} = K_2(x_2 - W)$$
$$F_{damper2} = b_2(V_2 - \frac{dW}{dt})$$

Here are some helpful notes:

1. W is the ground profile (i.e., road disturbance). If $W = 0$, this means that the road is flat and there are no disturbances. It could be modeled as a step input.
2. The control force, u, could be modeled as a step input.
3. $X_1 - X_2$ is the output of the system, and it represents how much the body will oscillate. For example, if the vehicle hits a step on the road, the value of $X_1 - X_2$ will tell us the range of oscillation that happened due to the hit. Use a Simulink block that displays the oscillation ($X_1 - X_2$).
4. Pay attention to how each component (springs and dampers) affects each mass. For example, the effect of spring 1 with respect to mass 1 is not the same as its effect with respect to mass 2.
5. Use the following parameters to run the simulation:

Parameter	Value
M_1	2500 Kg
M_2	320 Kg
K_1	80000 N/m
B_1	350 N.s/m
K_2	500000 N/m
B_2	15020 N.s/m

Table 1: System parameters

Notes :

1. Set all the parameters in the control force block (U) to zero.
2. The vehicle hits a 10-cm pothole (a hollow part in the road) after 5 seconds of operation. This means that, in the disturbance (W) block, you have to set the step time to 5 and the final value to -0.1.
3. Make sure that the solver has a fixed step size of 0.001. You can access the solver settings from the bottom right of Simulink.

II. System Identification

System identification is a methodology for building mathematical models of dynamic systems using measurements of the input and output signals of the system.

System identification involves the following steps:

- Measuring the input and output signals from the system in time or frequency domain.
- Selecting a model structure that describes the relationship between the input and output variables.
- Applying an estimation method to estimate the values of the adjustable parameters in the model structure.
- Evaluating the estimated model to see if it is adequate for the application needs. The evaluation can be based on goodness-of-fit measures, validation data, residual analysis, etc.

System identification can be used for various purposes, such as control design, simulation, prediction, optimization, fault detection, and system analysis.

You are given measured data in time domain for the input signal (ESC signal) and output (force) of Brushless DC motor with a propeller attached. The system behaves much like a simple lag which is a delay between input and output. The system can be modeled as the following :

$$\frac{output}{input} = \frac{K}{\tau s + 1}$$

The system has only a gain and one pole with no zeros.

Your task is to use this data and obtain the constants K and τ which describe the system.

You may find this video useful in understanding how the whole process works: https://rb.gy/ohzala .

Notes: in PID Tuner tool choose Arbitrary I/O Data instead of step response. Use start time $T_0 = 0$ and Sampling time $= 1/60$ sec .

III. Linux Installation

Install Ubuntu Linux version 20.04.6 on WSL2(Windows subsystem for Linux), follow this tutorial: http://tinyurl.com/4xn2b7je.

Then practice navigating in the terminal using Linux commands as you will be asked in interview about it.

In the submission , you required to attach a screenshot of your terminal after writing the following command 'whoami' .

Learning material

I. Control System

1. Introduction to control theory:https://rb.gy/rc8miv.
2. What control system engineer do :http://tinyurl.com/2s47mkmr.
3. Explanation to transfer function:
   https://shorturl.at/cgqFS
   https://rb.gy/pkejqc.
4. Step Response explained:https://shorturl.at/ejozS.
5. System Identification :https://shorturl.at/aeiqS.
6. System Identification Methods : https://shorturl.at/cBOP1.
7. Linearization : https://shorturl.at/afpM6.
8. PID Control : https://shorturl.at/fjPRW.

II. Open Source Autopilot Basics

PX4/Ardupilot are an open-source flight control firmware that are designed to be used in unmanned aerial vehicles (UAVs), also known as drones. They are developed by the PX4/Ardupilot open-source community and can be used for a variety of applications, including aerial photography, mapping, inspection, and delivery. The PX4/Ardupilot firmware run on a micro-controller, which is typically embedded in the drone's flight control board. It receives sensor data from the drone's sensors, such as accelerometers, gyroscopes, and GPS, and uses this information to stabilize the drone and execute its flight plan.
We will focus on the PX4 firmware since its highly configurable and can be customized to suit different types of drones and applications. It supports a wide range of hardware platforms, including many popular flight controllers, and can be integrated with various types of software and development environments.

- PX4 User guide: https://docs.px4.io/main/en/

1. Stick to the following sections :
2. Getting Started
3. Basic Assembly:Mounting the Flight Controller- Mounting the GPS/Compass-Vibration Isolation- Cable wiring- mRo Pixhwak Wiring Quick-start.
4. Standard Configuration.

- Introduction to Pixhawk Flight Controller playlist :https://t.ly/RH9vN.

- How drones work : https://t.ly/HrVS9.

Ground Stations

QGroundControl is an open-source ground control station (GCS) software used to control unmanned vehicles, such as drones, robots, and autonomous vehicles. It is designed to work with the open-source PX4 flight control firmware, which provides a complete solution for the development of unmanned aerial and ground vehicles. QGroundControl provides a user-friendly interface for configuring, monitoring, and controlling unmanned vehicles. It allows users to plan and execute flight missions and view telemetry data in real-time. QGroundControl is cross-platform software, which means it can run on Windows, Linux, and mac-OS operating systems. It supports a wide range of UAV platforms, including quad-copters, fixed-wing aircraft, and multi-rotors. QGroundControl is also highly extensible, allowing developers to create custom plugins and interfaces to integrate with other software or hardware.

- An intro guide video to qground: https://t.ly/CRvVh.

III. Programming

Graphical programming

Simulink is a powerful visual programming environment and simulation tool developed by MathWorks. It is widely used in various engineering disciplines, particularly in control system design, signal processing, and model-based design. Simulink allows users to model, simulate, and analyze dynamic systems using a graphical block diagram approach, making it intuitive for engineers and researchers.

- Introductory playlist to simulink : https://t.ly/piN9_.

Linux OS

- what is Linux:https://t.ly/RSNgg.
- Basic Linux Commands:http://tinyurl.com/nhbhs6u4.