WEEK 1

We'll be starting with Week 1 of the course. This week deals with modeling on MATLAB and Simulink.

MATLAB AND SIMULINK

MATLAB is a multi-paradigm numerical computing environment and proprietary programming language developed by MathWorks. MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages.

Simulink is a MATLAB-based graphical programming environment for modeling, simulating, and analyzing multi-domain dynamical systems. Its primary interface is a graphical block diagramming tool and a customizable set of block libraries. As to why we need to learn MATLAB and Simulink, it's because we have to plot the trajectories of robots keeping in mind the kinematics involved in it.

To start with MATLAB and Simulink, you need to do the Matlab Onramp course. The first task of this week would be doing the MATLAB and Simulink Onramp courses. You can access these courses here.

CONTROL SYSTEMS

Control systems help to control the movements and functions of the robot. We need the controllers because the dynamics vary with time. When the robot moves up in a slope and then down in the slope, or first travels on smooth concrete, then on a carpeted floor. So physical modeling of the "System" becomes crucial for designing a good controller.

We give a reference state to a controller. The controller also has sensor feedback, using the reference state and the sensor feedback controller generates a control signal needed to reach the reference state. This control signal is fed to the "System". The system dynamics determine how the system behaves to this control input. If the controller is good, hopefully, the "System" will reach our desired reference state.

- To start with the basics of control theory read this page
- To get started with control systems and theory - Control of Mobile Robots course by Magnus Egerstedt (Georgia Tech) (You only need to watch the first 9 videos of this course)
- Understanding PID Control - Playlist by MATLAB explains PID control in detail. You only need to watch the first 4 videos of this playlist. Also, read about the controller over here.

ASSIGNMENT

1. THE FIRST TASK OF THIS WEEK WOULD BE DOING THE MATLAB AND SIMULINK ONRAMP COURSES MENTIONED ABOVE. YOU WILL BE REQUIRED TO DOWNLOAD/SAVE THE CERTIFICATES OF THESE COURSES AND SUBMIT THEM.

2. THE SECOND TASK OF THIS WEEK IS TO WRITE A MATLAB CODE TO IMPLEMENT A BASIC PID CONTROLLER ON THE GIVEN SYSTEM. USE THE CODE GIVEN BELOW TO COMPLETE THE TASK. IT HAS ALL THE INSTRUCTIONS ON WHAT YOU NEED TO DO. YOU CAN TAKE THE HELP OF THIS TUTORIAL FOR DOING THIS TASK. YOU NEED TO SUBMIT THE .M FILE FOR THIS TASK. THE NAME OF THE FILE SHOULD BE **"WEEK-1_ASSIGNMENT_T2.M"**

```
%% What to do
% Do not worry much if you do not completely understand the code, the main
% objective is to implement a PID controller on the given system system.

%% Second Order System general format
%                  wn^2
% sys =  ------------------------
%        (s^2 + 2*zeta*wn*s + wn^2)
% You can read more about the second order system on internet if you are
% interested
wn = 3;
zeta = 0;

%% System
% Built-in MATLAB functions used. Need not worry much about this.
system = tf(wn^2, [1, 2*zeta*wn, wn^2]);

%% Insert your code here
% For the given 'system' above implement a pid controller and try using
% different values of kp, ki and kd until you observe the least overshoot
% and stable output.
% As a bonus you can also try out some auto pid tuning features of MATLAB

%% Output
step(sys)
%to give step input to the system (basically simulate the system)
```

3. THE THIRD TASK WILL BE TO IMPLEMENT THE SAME SYSTEM GIVEN ABOVE, ON SIMULINK. YOU CAN TAKE THE HELP OF THIS TUTORIAL FOR DOING SO. YOU NEED TO SUBMIT A .SLX FILE FOR THIS TASK. THE NAME OF THE FILE SHOULD BE **"WEEK-1_ASSIGNMENT_T3.SLX"**. SOME BLOCKS THAT WILL HELP YOU ARE

    - Transfer fcn
    - Step
    - Scope
    - PID

All your files - 2 certificates, the .m, and the .slx file all need to be zipped and you will need to submit that in the classroom. The file name should be **"Your_Name-W1.zip".**

The deadline for submission is 13th July (Tuesday) 11:59 PM.

FEW EXTRA POINTS:

1. To understand more about any function in MATLAB or block in Simulink, refer to the MATLAB official documentation online. It has a very detailed explanation for everything. Make this your habit whenever you come across something new, this will help you a lot in improving your knowledge.
2. Once you are done with PID, Try changing the values of *zeta* and *wn* run the code, again and again, to see how it affects the new system.

INSTALLATION GUIDELINES

1. Download MATLAB and Simulink from the official website, many universities provide free access to MATLAB. Just log in with your university mail and check if you are able to download the software for free. Download the latest release which is **R2021b**. (*BITS has Mathworks license*)
2. Although we highly suggest using MATLAB and Simulink for this course, in case you don't have free access to these through your university, You can check out these substitutes for it.

   You can download GNU Octave, Scilab, or Xcos. You may follow the given playlists for a better understanding of the substitutes mentioned above.

   - GNU Octave
   - Scilab
   - Xcos