Screening Task
for
P(11-3) - Project Research Assistant

Please attempt any one of the tasks. Remember that your submission will be graded on the quality of your work and validity of the results.

We would want to test your Scilab coding skills together with some basic knowledge on signal processing. As a test, you are expected to do the following.

Task 1:

1. See this or this link. They have a list of octave compatible files that are used in its “signal” package.
2. See this spreadsheet. It has a list of signal processing toolbox functions. They have been marked as either blank or "done" or "problem".
3. Referring to the spreadsheet, you need to choose those functions which have a blank remark. Choose any 2.
4. Find their equivalent octave implementation in the links provided in point 1. If you find the implementation elsewhere on the Internet, you may very well use it as long as it is valid.
5. Code these functions in Scilab. Use Scilab-5.5.2 to do so.
6. Write example Scilab code to demonstrate the converted function’s usability and validity.
7. Zip your functions and examples together and keep them ready for submission. You may include a README file in case if you want to explain your submission.

NOTE: The git repository for our scilab signal processing toolbox is available here. Use this toolbox on Scilab-5.5.2. Load the toolbox in scilab by issuing the command “exec loader.sce” on the scilab console. Your scilab’s working directory must be the toolbox directory in order to use it. Feel free to browse through the toolbox by issuing the command "help" on the scilab console and reading through the help entry for this toolbox.

In case if you are unable to find any octave implementation online, you may try finding them in Octave directly. However, you will have to install and load the “signal” package in octave using the “pkg install -forge signal” and “pkg load signal” commands respectively. Thereafter you may issue the command “type function_name” to see the code behind it. For example, “type grpdelay”.

Task 2:

1. Implement a PID controller simulation completely in Xcos. Choose a second order underdamped transfer function. Plot the setpoint and the controlled variable in a single plot. Plot the manipulated variable in a different plot.
2. Write a scilab code to find the unit step response of the second order transfer function of the form 10/(s^2+0.2*s+400) and write the input and output data in to a .txt file.
3. Write a Scilab function which accepts a vector "a" (can be 1xn) and returns the maximum value contained in "a" along with its index. Try to call this function from xcos, if possible. If not, specify reasons.
4. Write a scilab code which accepts a transfer function in terms of a polynomial from the user. The code finds its roots and plots the pole-zero plot, bode plot, and calculates the gain margin and phase margin

Submission procedure for both Task 1 and Task 2:

1. Put your code along with other supporting files (if any) in a folder. You may put a README file inside the folder, if you want, to give us more information about your submission. Rename that folder as "j**ob-code-YourName**", without quotes. For example, **P(11-3)-satish.**
2. Compress the folder in ZIP format. Avoid any other compression format.
3. Mail the zip file to **info@fossee.in.**

Make sure to put the mail subject-line as "j**ob-code-YourName** without quotes. For example, "**P(11-3)-satish**" No extension in the deadline will be considered for submission of screening task