Homework 2 Due:20/11/2022

Q1 A second-order transfer function is defined as follows,

$$G(s) = \frac{w_n^2}{s^2 + 2\zeta w_n s + w_n^2}$$

where $w_n \in \mathbb{R}^+$ is the natural frequency and $0 < \zeta < 1$ is the damping ratio. The settling time needs to be modeled and it is known that the formula has the following form,

$$ts(w_n, \zeta) = \frac{\cdot}{\zeta w_n}$$

- [10p] Use `for` loops and the `meshgrid` function in order to calculate the model paramter of the settling time. You may minimize the Integral Squared Error(ISE) of the difference between the actual settling time and the settling time calculated using the formula.
- [10p] Use the matlab `fit` function.

The rise time can be modeled via,

$$tr(w_n, \zeta) = \frac{\cdot\zeta + \cdot}{w_n}$$

or via,

$$tr(w_n, \zeta) = \frac{\cdot\zeta^2 + \cdot\zeta + \cdot}{w_n}$$

- [20p] Use the matlab `fit` function.

Q2 [30p] Implement the function `info=rlocus_info(Gs)` which

- [30p] calculates the breaking points on the real axis on the root locus of the provided transfer function $G(s)$.
- [30p] calculates the jw-axis crossing on the root locus of the provided transfer function $G(s)$.

The input argument is a `tf` object and the function returns a struct that encapsulates the breaking points(not the candidates) and the gain values of the jw-axis crossing with the crossing frequency w.

- Prepare a report `*.pdf` for Q1 and submit 1 PDF file and the `rlocus_info.m` file.
- The given main function is not going to be submitted, only the necessary implementation needs to be submitted.
- Your submission will be tested with a tester `main.m` file. Your code needs to run without error, or your grade will be zero.
- Each functionality will be tested and added to your grade.
- Late submissions will be deduced 10p for each day late.
- Cheating is not allowed, once cheating is detected all involved submissions will be graded zero.

```
%% UPLOAD ONLY THIS FILE
function  info=rlocus_info(Gs)
        % convert Gs into a symbolic expression

        % calculate real axis breaking points
        bpoints=0;

        % calculate jw-crossings
        k=0;
        w=0;

        info=struct("bpoints",bpoints,"k",k,"w",w); %% DO NOT
           CHANGE THIS struct and the variable names
end
```

deliverables/rlocus_info.m

```
%% DO NOT UPLOAD THIS FILE
clear;clc;

Gs=tf(1,[1 2*0.2*10 10^2]);
info=rlocus_info(Gs);
```

deliverables/main.m