Homework #1

Due Date : Mar 16, 2020 (23.59pm)

Problem #1.

- **Background**

In the field of control engineering, engineers are often faced with the problem of stability analysis. To do this, control engineers try to find the gain cross-over and phase cross-over frequencies because they contain pretty important information for analyzing stability of the given control system. The gain cross-over and phase cross-over frequencies are closely related to the open-loop system's frequency response $G_o(j\omega) = G_o(s)|_{s=j\omega}$ which is the complex rational function in angular frequency $\omega[rad/s]$. The gain cross-over frequency ω_{gc} and the phase cross-over frequency ω_{pc} satisfy the following conditions in complex plane:

$$\| G_o(j\omega) \|_{\omega=\omega_{gc}} = 1, \qquad \angle G_o(j\omega)|_{\omega=\omega_{pc}} = \pi$$

Therefore, in view of numerical method, this problem can be cast into one of root finding problems associated with scalar nonlinear functions.

- **Problem Statement**

Our first problem is related to the development of an appropriate C-program to automatically calculate these frequencies under the assumption that the mathematical system model $G_o(s) = \frac{N(s)}{D(s)}$ of interest is given. To do this, note that

1) The only known prior information about the system model $G_o(s) = \frac{N(s)}{D(s)}$ is that the maximum orders of its denominator $D(s)$ and numerator $N(s)$ do not exceed 3. That is, the system model is generally represented as follows:

$$G_o(s) = \frac{N(s)}{D(s)} = \frac{b_3 s^3 + b_2 s^2 + b_1 s + b_0}{s^3 + a_2 s^2 + a_1 s + a_0}$$

2) When you compose your own program, you must adopt one of representative root finding algorithms introduced in our class such as 'Bisection', 'Newton', 'Secant', and 'Fixed-point Iteration'.
3) In your program, the user can input the coefficients $\{a_i\}$ and $\{b_i\}$ of $D(s)$ and $N(s)$ as well as their orders m and n. The user can insert the coefficients to two decimal places. But, it is always assumed that $a_m = 1$. Once the user inputs the data about the open loop system model $G_o(s)$, your program should print out it on the screen and ask whether the data is correct or not. If the answer is "No", the above mentioned data input process should be re-iterated. For example,

```
Input the order of denominator             : 3
What are the coefficients of denominator?  : [1 15 75 250]

Input the order of numerator               : 2
What are the coefficients of numerator?    : [40 1000]

The inserted open loop system model is as follows:

                          40.00 s + 1000.00
Go(s) = ----------------------------------
```

```
S^3 + 15.00 s^2 + 75.00 s +  250.00

Is it correct [y/n]? y
```

4) Specify the numerical method used for calculating ω_{gc} and ω_{pc}. Furthermore, show the iteration process and values of ω_{gc} and ω_{pc} calculated by your algorithm. Round off to the nearest hundredth when you display the calculation results. Note that the units of ω_{gc} and ω_{pc} are $[rad/s]$.

```
Calculating Gain Crossover and Phase Crossover Frequencies...

[Newton's Method]
Iteration #  | 어쩌구
0            | 저쩌구
1            | 어쩌구
2            | 저쩌구

Gain  Crossover Frequency        Wgc =  9.35 [rad/s]
Phase Crossover Frequency        Wpc = 12.75 [rad/s]
```

5) Your program can properly handle the following exceptional cases:
   ① if there is no solution, give a message as follows:

```
Gain  Crossover Frequency        Wgc =   -   [rad/s]
```

   ② if there are one more solutions, print out the smallest value as the cross-over frequency.

Problem #2.

- **Background**

Most autonomous vehicles have a GPS(global positioning system) receiver in order to obtain their position information. To determine the vehicle's position, the GPS receiver makes use of the concept of time-of-arrival ranging. This concept entails measuring the time it takes for a signal transmitted by the i^{th} satellite at the known location $(x_{s,i}, y_{s,i}, z_{s,i})$ to reach the GPS receiver. Recall that the range between the i^{th} satellite and the GPS receiver is calculated by the time difference $(t_r - t_{s,i})$ between the received and transmitted signals, multiplied by the speed of light c. Unfortunately, the GPS receiver clock is not synchronized with the satellite clock in general, hence the actual range measurement of the GPS receiver(pseudorange) should be represented by taking the unknown receiver clock offset ΔT into account.

$$R_i = f(\boldsymbol{x}) = \sqrt{\left(x - x_{s,i}\right)^2 + \left(y - y_{s,i}\right)^2 + \left(z - z_{s,i}\right)^2} - \Delta T \cdot c$$

where t_r and t_s are the receiver and the satellite time, respectively. These are provided by their own clocks.

From the above relationship, it is obvious that at least four pseudoranges($i = 1{\sim}4$) are needed for determining the vehicle position and the receiver clock offset, $\boldsymbol{x} = [x \quad y \quad z \quad \Delta T]^T$. The system of equations for four satellite psuedoranges is multivariate and nonlinear. Consequently, these coupled nonlinear equations should be also solved by relying on the numerical techniques.

- **Problem Statement**

For the following satellite ephemeris data and pseudorange measurements, the Newton-Raphson method to determine the vehicle position and the GPS clock offset. Use the similar program structure used in Problem 1.

satellite	$x_{s,i}$	$y_{s,i}$	$z_{s,i}$	R_i
#1	20000	19400	19740	19992.498593
#2	18700	1800	18500	18992.103621
#3	18900	17900	20000	19463.491788
#4	17900	21000	18500	19383.574838

Problem #3.

In order to calculate the position of satellite radar image center, $\vec{R}_t = [r_x^t \quad r_y^t \quad r_z^t]^T$, one should solve the following coupled nonlinear equations related to the imaging geometry.

① range equation

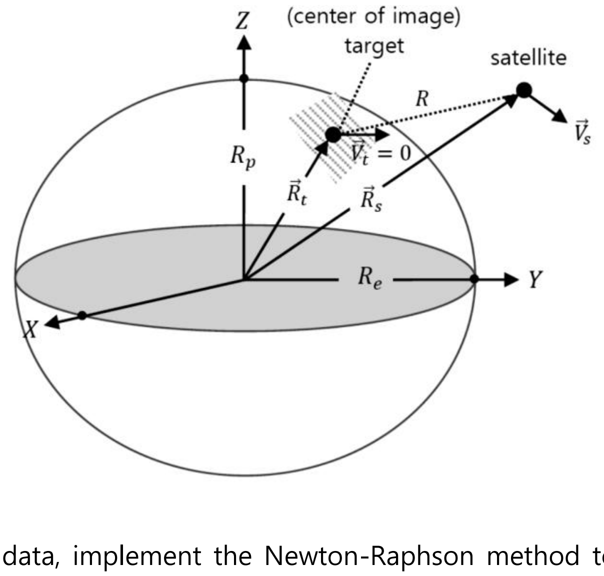
$$R = |\vec{R}_s - \vec{R}_t|, \quad R = (2.998 \times 10^8) \times \frac{5.69116[ms]}{2}$$

② zero Doppler

$$f_{DC} = -\frac{2}{\lambda} \times \frac{(\vec{V}_s - \vec{V}_t) \cdot (\vec{R}_s - \vec{R}_t)}{R} = 0, \quad \lambda = 0.056[m], \ \vec{V}_t = \mathbf{0}$$

③ Earth model

$$\frac{(r_x^t)^2 + (r_y^t)^2}{R_e^2} + \frac{(r_z^t)^2}{R_p^2} = 1, \quad R_e = 6378114[m], \quad R_p = 6356759[m]$$



Given the satellite ephemeris data, implement the Newton-Raphson method to calculate $\vec{R}_t$. Use the similar program structure used in Problem 1.

$$\vec{R}_s = \begin{bmatrix} r_x^s \\ r_y^s \\ r_z^s \end{bmatrix} = \begin{bmatrix} -3760226.95 \\ 4429412.04 \\ 4182432.32 \end{bmatrix} [m], \quad \vec{V}_s = \begin{bmatrix} v_x^s \\ v_y^s \\ v_z^s \end{bmatrix} = \begin{bmatrix} -1414.48007 \\ 4432.34777 \\ -5949.22393 \end{bmatrix} [m/s]$$