EEL 4930/5934: Autonomous Robots
Homework #5 (Spring 2023)

Tasks Overview:

A. 3D Robot localization from 3 landmarks
B. Inverse kinematics (DH notation of manipulators)
C. Kalman filtering for 2D object tracking in images
D. Bonus part: bounding box tracker using KF (+20 points)

Grading Breakdown

EEL 4930	**EEL 5934**
• Part A: 25	• Part A: 20
• Part B.1: 25	• Part B.1: 20
• Part B.2: extra! (+5 bonus points)	• Part B.2: 10
• Part C: 50	• Part C: 50

Tasks Overview: Lecture 7-8 materials. Download the **HW5_Blank** folder from canvas.

Submission: [Through Canvas only; Due: April 26, 2023 by 11.59pm]

- A single zip file with a folder (code) and PDF (report)
  - A PDF report (Part-A and Part-B): scanned pages of your handwritten solutions are fine as long as they are clear and readable; no need to write/draw anything for Part-C.
  - Your completed code (Part-C)
    - Do not add any more python files, just complete the functions.
  - If you solve the bonus point (Part-D): mention this in the report so that we can check.
    - Implement your new KF class: `class KF_4D(object)` in `KF.py` and write the test code in a new `circleBBoxTracker.py` file.
- Assignments more than 20 MB file size will get negative penalty (-10% to -50%)

Part A: 3D Robot localization from 3 landmarks

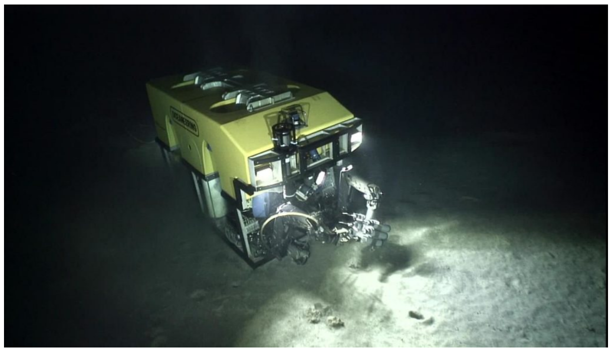
Given: coordinates of 3 non-planar points

$$^GP_1 = \begin{bmatrix} ^Gx_1 \\ ^Gy_1 \\ ^Gz_1 \end{bmatrix},\ ^GP_2 = \begin{bmatrix} ^Gx_2 \\ ^Gy_2 \\ ^Gz_2 \end{bmatrix},\ \text{and}\ ^GP_3 = \begin{bmatrix} ^Gx_3 \\ ^Gy_3 \\ ^Gz_3 \end{bmatrix}$$

$$^AP_1 = \begin{bmatrix} ^Ax_1 \\ ^Ay_1 \\ ^Az_1 \end{bmatrix},\ ^AP_2 = \begin{bmatrix} ^Ax_2 \\ ^Ay_2 \\ ^Az_2 \end{bmatrix},\ \text{and}\ ^AP_3 = \begin{bmatrix} ^Ax_3 \\ ^Ay_3 \\ ^Az_3 \end{bmatrix}$$

Find the position $^G\mathbf{P}_A$ and rotation matrix $^G\mathbf{R}_A$

$$^G_A\mathbf{T} = \left[\begin{array}{c|c} ^G_AR & ^GP_A \\ \hline 0\ \ 0\ \ 0 & 1 \end{array}\right]$$



Part B: Inverse kinematics (DH notation of manipulators)

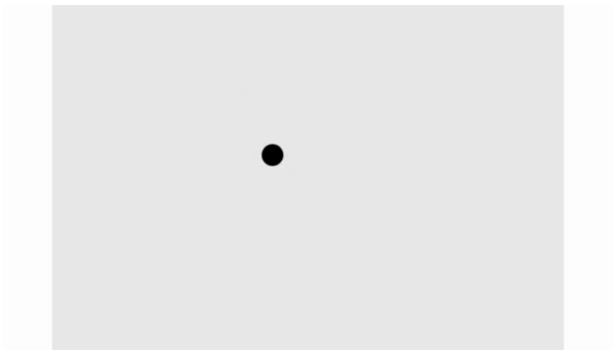
Present your work in detail explaining your methodology at every step.

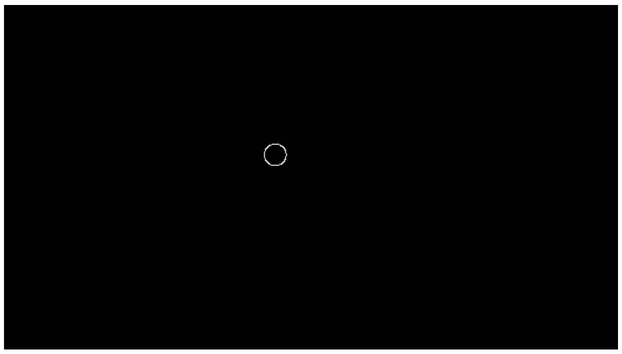
B.1 Craig's book exercise 4.2

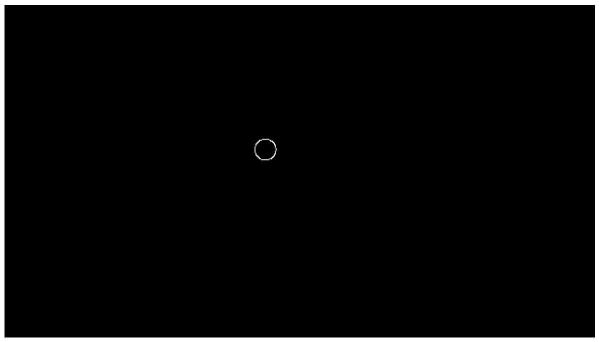
B.2 Craig's book exercise 4.16

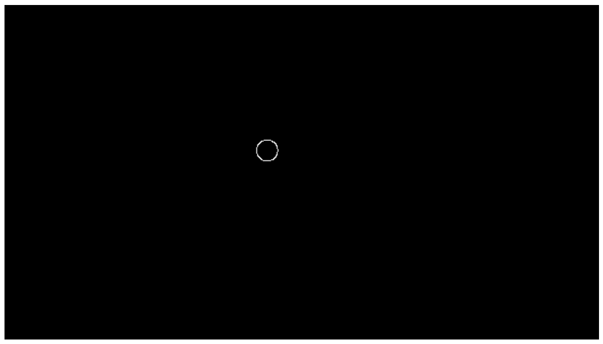
Part C: Kalman filtering for 2D object tracking in images

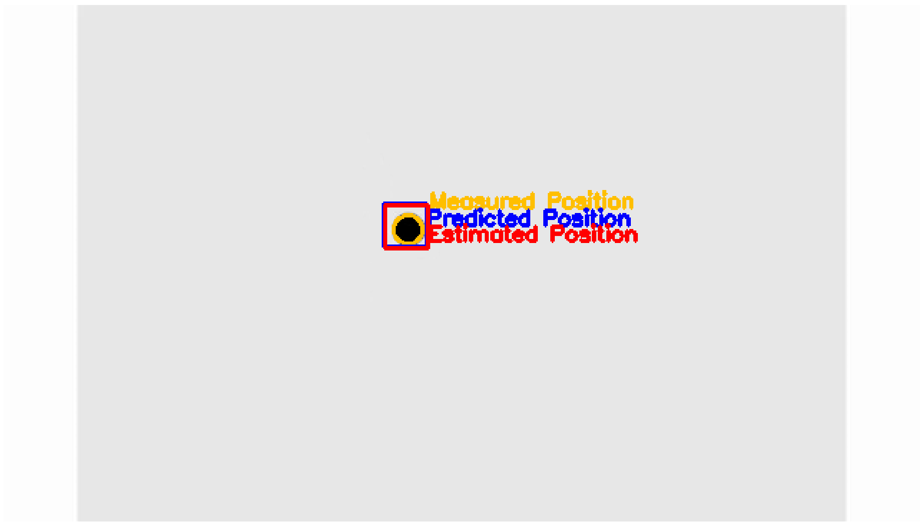
You will implement the basic Kalman Filtering algorithm for 2D object tracking in images. The detailed instructions are provided in the **HW5_Blank** folder; download it from canvas. The relevant materials are discussed in class (Lecture 8); your implementation is a simpler version, which is to track the center of a circle from observations. Complete the missing functions in the given template and check your output.


Sample input images from video


Detect edges using OpenCV


Convert to binary image by thresholding


Find the object contour: this is your detected circle



Finally apply the KF predict and update rule to find and track the estimation

Bonus Part D: Bounding box tracker using KF (maximum +20 points)

The detector of your tracker `circleDetector(image)` function now returns the center of the circle, which we track. Instead, return a bounding box (`x, y, w, h`) and modify your KF driver so that we can track (predict and update) the bounding box. That is, we will be tracking 4D instead of 2D - the principle still remains the same, but you will need to adjust the driver, parameters, and the predict-update rules. Implement your new KF in a class: `class KF_4D(object)` in `KF.py` and write the test code in a new `circleBBoxTracker.py` file.