ME 449 Assignment 2
Due 1:30 PM, Monday October 25, 2021

1 Screw Axes and the Jacobian (24 pts)

Figure 1 shows the 6-dof ARMin III rehabilitation robot [1]. Write $M = T_{sb}(0)$ and the 6×6 body Jacobian $J_b(0)$ at the home configuration shown.

2 Numerical Inverse Kinematics (76 pts)

Starting from `IKinBody` in the MR code library, write a new function, `IKinBodyIterates`. This function prints out a report for each iteration of the Newton-Raphson process, for iterates 0 (the initial guess) to the final answer. Each iteration reports the iteration number i, the joint vector θ^i, the end-effector configuration $T_{sb}(\theta^i)$, the error twist $\mathcal{V}_b$, and the angular and linear error magnitudes, $\|\omega_b\|$ and $\|v_b\|$ (something like the table at the end of Chapter 6.2.2). For a four-joint robot, a typical iterate might look like:

```
Iteration 3:

joint vector:
0.221, 0.375, 2.233, 1.414

SE(3) end-effector config:
1.000  0.000  0.000  3.275
0.000  1.000  0.000  4.162
0.000  0.000  1.000 -5.732
0      0      0      1

error twist V_b:
0.232, 0.171, 0.211, 0.345, 1.367, -0.222

angular error magnitude ||omega_b||:  0.357

linear error magnitude ||v_b||:  1.427
```

The function should also save the joint vector of each iteration as a row in a matrix. For a four-joint robot with three iterates (including the initial guess), the matrix would be

$$\begin{bmatrix} \theta_1^0 & \theta_2^0 & \theta_3^0 & \theta_4^0 \\ \theta_1^1 & \theta_2^1 & \theta_3^1 & \theta_4^1 \\ \theta_1^2 & \theta_2^2 & \theta_3^2 & \theta_4^2 \end{bmatrix}.$$

When your function completes, it should save the matrix as a .csv file, where each row of the text file consists of the comma separated joint values for that iterate. You can learn more about generating .csv files here.

Test your new function for the UR5 robot of Example 4.5 of Chapter 4.1.2 (Figure 4.6). The home configuration of the end-effector M is given in the book, as well as the numerical values of the constants $L_1, L_2, H_1, H_2, W_1, W_2$. The screw axes $\mathcal{B}_i$ in the end-effector frame are

$$\begin{array}{ll}
\text{joint 1:} & (0,1,0,W_1+W_2,0,L_1+L_2) \\
\text{joint 2:} & (0,0,1,H_2,-L_1-L_2,0) \\
\text{joint 3:} & (0,0,1,H_2,-L_2,0) \\
\text{joint 4:} & (0,0,1,H_2,0,0) \\
\text{joint 5:} & (0,-1,0,-W_2,0,0) \\
\text{joint 6:} & (0,0,1,0,0,0)
\end{array}$$

The desired end-effector configuration is

$$T_{sd} = \begin{bmatrix} 0 & 0 & -1 & 0.25 \\ 1 & 0 & 0 & -0.15 \\ 0 & -1 & 0 & 0.1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where linear distances are in meters. Use $\epsilon_\omega = 0.001$ rad ($0.057°$) and $\epsilon_v = 0.0001$ (0.1 mm).

You will use the Newton-Raphson method with two different initial guesses θ^0 to determine how the initial guess affects convergence to a set of joint angles that satisfies the desired end-effector configuration. Choose initial guesses θ^0 so that the numerical inverse kinematics

1. converges after 3–5 Newton-Raphson steps (short_iterates) and
2. converges after 10 Newton-Raphson steps, or never converges (long_iterates).

You can use the sliders in the CoppeliaSim UR5 interactive scene (Scene 1) to choose initial guesses. (Check out `http://hades.mech.northwestern.edu/index.php/CoppeliaSim_Introduction` for more information.) You can type the final joint angles found by your function into the UR5 interactive scene to confirm that your function works properly and the desired end-effector configuration is achieved.

Your pdf report should provide the following figures:

1. A 3D plot that shows the (x, y, z) position of the end-effector at each iterate, with a line between successive iterations. (See the sample figure at end of document).
2. A plot of the magnitude of the linear error (on the y-axis) as a function of the iterate number (on the x-axis).
3. A plot of the magnitude of the angular error as a function of the iterate number.

The results from both initial guesses should be shown in each figure. (Three figures total.) Label all axes and figures, and provide a legend that labels the two different initial guesses.

Once you have done this, assemble your documents for submission. **You will submit one zip file (FamilyName_GivenName_asst2.zip; for me, it would be Lynch_Kevin_asst2.zip).**

- The zip file should contain:
  1. The pdf file FamilyName_GivenName_asst2.pdf. (See below.)
  2. Your commented code in a directory called "code." You only need to provide your modified function(s), not the rest of the MR code.
  3. A text file called "short_iterates.csv," created by your `IKinBodyIterates` function and used to generate the video that takes 3–5 iterations to converge.

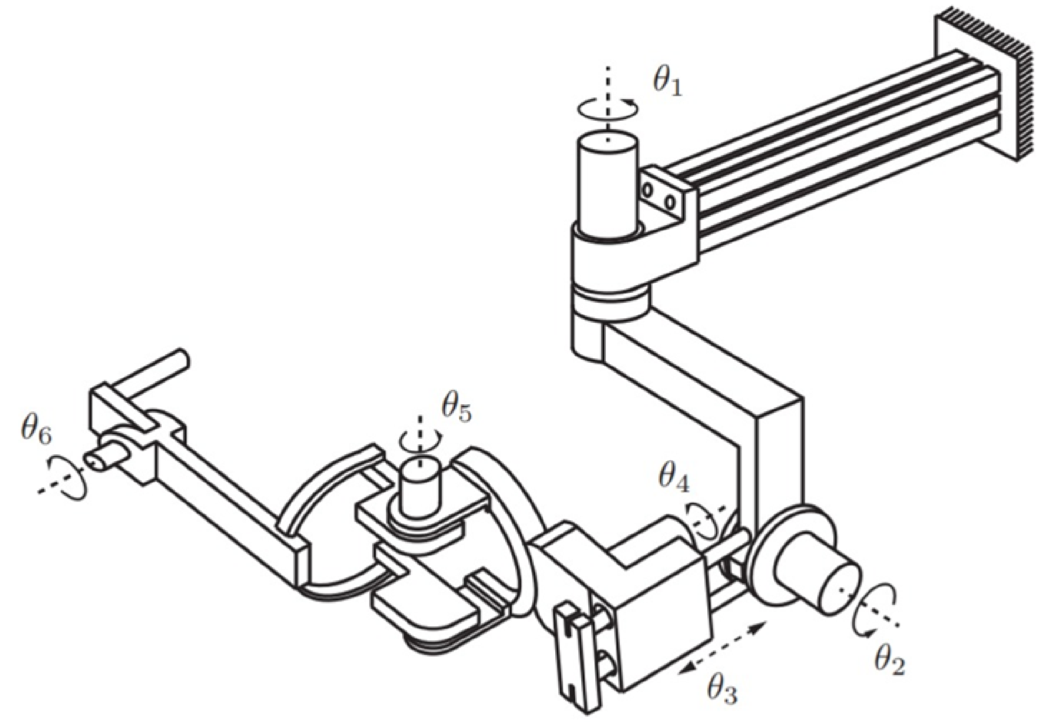
4. A text file called "long_iterates.csv," created by your `IKinBodyIterates` function and used to generate the video that takes 10 or more iterations to converge, or never converges.
5. Two CoppeliaSim videos animating the Newton-Raphson iterations, **one for each initial guess**. Use the CoppeliaSim csv animation scene for the UR5 (Scene 2). The videos should show CoppeliaSim "playing" your .csv file. (Go to `http://hades.mech.northwestern.edu/index.php/CoppeliaSim_Introduction` to learn about making videos with CoppeliaSim.) The video is just a sequence of configurations of the robot, equal to the number of iterates in your .csv file. You should uncheck the "Interpolate" checkbox in the CoppeliaSim UR5 animation scene to make this video. Your video should be a "reasonable" size (e.g., a few MB, less than 10 MB) and use a standard codec (e.g., some variant of .mp4) that common video viewers, in Mac OS, Windows, or Linux, can view. Your video should be taken from a virtual camera angle that makes it easy to see the end-effector configuration. **The video file names should follow the convention FamilyName_GivenName_short.mp4 and FamilyName_GivenName_long.mp4 (or possibly a different common file extension).**

- The pdf file should contain:

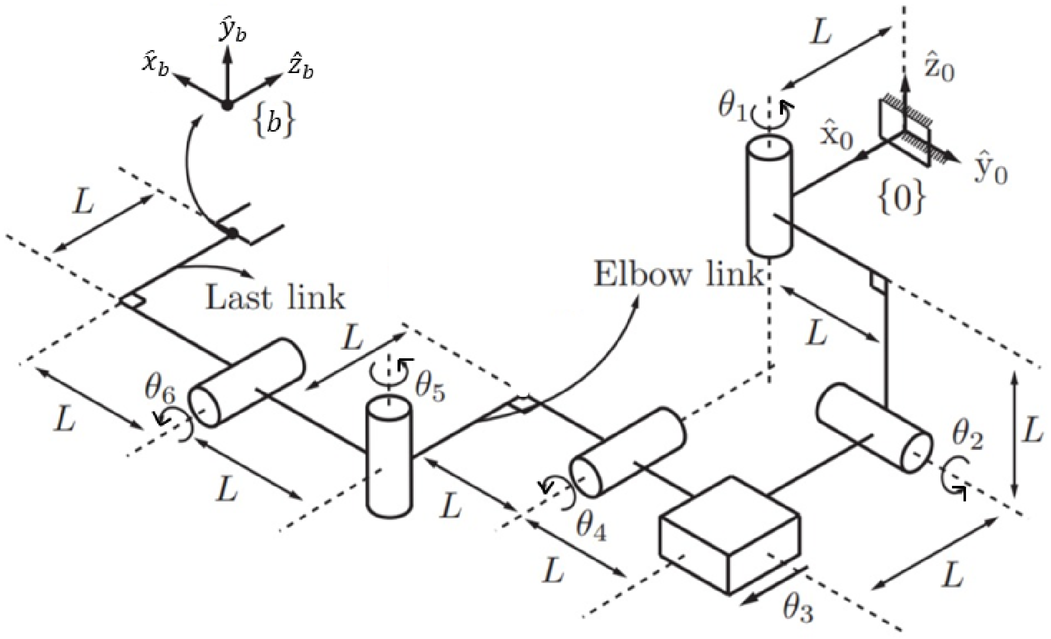
1. Any information that will help the grader understand your entire submission.
2. The M and $J_b(0)$ matrices for the rehabilitation robot ARMin III.
3. One screen log for each of your two guesses. This log should show how your code is called and the text output, i.e., the Newton-Raphson iterates that are printed to the screen when you call `IKinBodyIterates` with your initial guess.
4. A CoppeliaSim screenshot showing the UR5 at the solution configuration calculated after 3–5 iterations for your "good" initial guess. This screenshot should clearly show the UR5's end-effector configuration as well as the $SE(3)$ configuration reported by the scene's interface, confirming that your code calculated a good solution.
5. A figure showing the progression of end-effector (x, y, z) positions during the solution process. (Plots of both initial guesses in the same figure.)
6. A figure showing the magnitudes of the linear error as a function of iterations (both initial guesses).
7. A figure showing the magnitude of the angular error as a function of iterations (both initial guesses).
8. An explanation why your long_iterates initial guess has trouble converging. Also, take a look at the joint vectors in successive iterates from both the "short" and "long" initial guesses. Do you see any strange behavior? Explain.

References

[1] Tobias Nef, Marco Guidali, and Robert Riener. "ARMin III–arm therapy exoskeleton with an ergonomic shoulder actuation". In: *Applied Bionics and Biomechanics* 6.2 (2009), pp. 127–142.



(a) Rehabilitation robot ARMin III [1]. Figure courtesy of ETH Zürich.



(b) Kinematic model of the ARMin III.

Figure 1 The ARMin III Rehabilitation Robot [1]

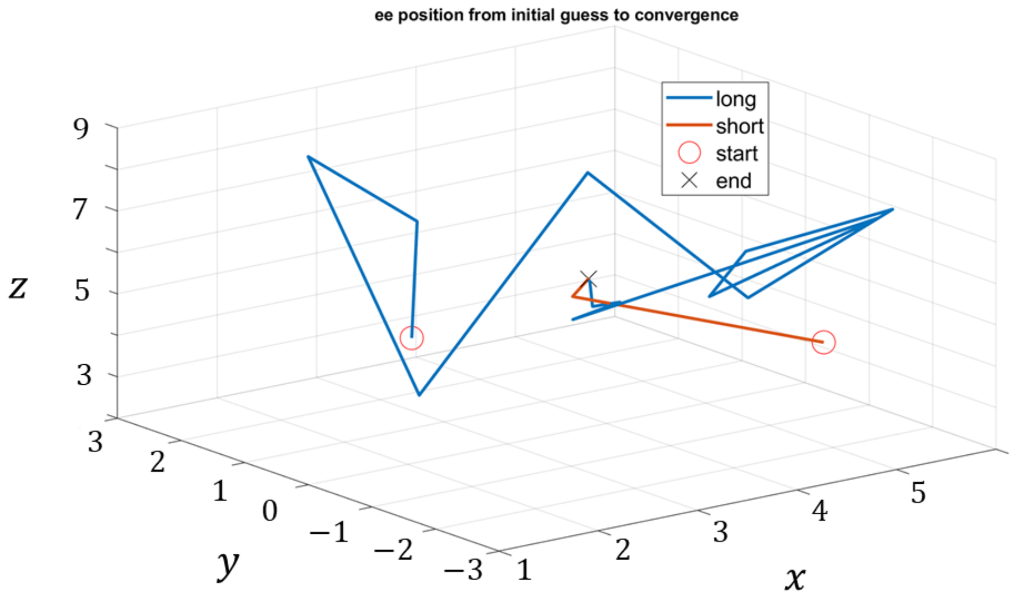


Figure 2 Sample figure for 3D plot, except your axis labels should include units (e.g., mm, cm, or m).