Inverse kinematics of the UR5 robot

Starting from the function **IKinBody**, 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 solution. 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:

$$\mathbf{Iteration\ 3:}$$

$$\texttt{joint vector:}$$
$$0.221,\ 0.375,\ 2.233,\ 1.414$$

$$\text{SE(3)}\ \texttt{end-effector config:}$$
$$\begin{array}{llll} 1.000 & 0.000 & 0.000 & 3.275 \\ 0.000 & 1.000 & 0.000 & 4.162 \\ 0.000 & 0.000 & 1.000 & -5.732 \\ 0 & 0 & 0 & 1 \end{array}$$

$$\texttt{error twist V\_b:}$$
$$0.232,\ 0.171,\ 0.211,\ 0.345,\ 1.367,\ -0.222$$

$$\mathbf{angular\ error\ magnitude}\ ||\mathbf{omega\_b}||:\ \mathbf{0.357}$$

$$\mathbf{linear\ error\ magnitude}\ ||\mathbf{v\_b}||:\ \mathbf{1.427}$$

The function should also save the joint vector of each iteration as a row in a matrix. For a four-joint robot with 3 iterates (including the initial guess), the matrix is

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

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.

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

joint i	ω_i	v_i
1	$(0, 1, 0)$	$(W_1 + W_2, 0, L_1 + L_2)$
2	$(0, 0, 1)$	$(H_2, -L_1 - L_2, 0)$
3	$(0, 0, 1)$	$(H_2, -L_2, 0)$
4	$(0, 0, 1)$	$(H_2, 0, 0)$
5	$(0, -1, 0)$	$(-W_2, 0, 0)$
6	$(0, 0, 1)$	$(0, 0, 0)$

The desired end-effector configuration is

$$T_{sd} = \begin{bmatrix} 0 & 1 & 0 & -0.5 \\ 0 & 0 & -1 & 0.1 \\ -1 & 0 & 0 & 0.1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where linear distances are in meters. Use $\epsilon_\omega = 0.001$ rad ($0.057°$) and $\epsilon_v = 0.0001$ (0.1 mm). Choose an initial guess θ^0 so that the numerical inverse kinematics converges after 3-5 Newton-Raphson steps. For example, you could use the sliders in the V-REP UR5 interactive scene to find an initial guess. (See the discussion in the first module of Course 1, or just go directly to http://hades.mech.northwestern.edu/index.php/V-REP_Introduction.) You can type the final joint angles found by your function into the UR5 interactive scene (Scene 2) to confirm that your function works properly and the joint angles of the solution achieve the desired end-effector configuration.

Once you have done this, you are ready to assemble your documents for submission. **You will submit a single .zip file of a directory with the following contents:**

1. **Your commented code in a directory called "code."** Your code should be lightly commented, so it is clear to the reader what the code is doing. No need to go overboard, but keep in mind your reviewer may not be fluent in your programming language. Your code comments must include an example of how to use the code. Only turn in functions that you wrote or modified; you don't need to turn in other MR functions that your code uses. If your code is in MATLAB or Python, just turn in the text files with your functions. If your code is in Mathematica, turn in (a) your .nb notebook file and (b) a .pdf printout of your code, so a reviewer can read your code without having to have the Mathematica software. Ideally your code would automatically generate a .csv file, but it is also acceptable if it simply prints out the comma-separated values, which then must be copy-and-pasted into a text editor to create the .csv file.

2. **A log showing your code's output, called "log.txt" (a cut-and-pasted plain text file) or "log.xxx", where xxx could be pdf, jpg, png, etc. (e.g., a screenshot).** This file should show your call of `IKinBodyIterates` with your initial guess, as well as all of the Newton-Raphson iterations until convergence, as described above.

3. **A text file called "iterates.csv".** This text file should have been created by your `IKinBodyIterates` function when it created the log file in part 2, above.

4. **A screenshot called "screenshot.xxx", where xxx could be jpg, png, etc., showing the UR5 at the solution.** This should be made with the UR5 interactive scene, and the 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 V-REP video animating the Newton-Raphson iterations.** Use the V-REP csv animation scene for the UR5. The video should show V-REP "playing" your .csv file. (See the V-REP description in module 1 of Course 1, or go directly to http://hades.mech.northwestern.edu/index.php/V-REP_Introduction, to learn about making videos with V-REP.) 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 V-REP 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 others can view. Your video should be taken from a virtual camera angle that makes it easy to see the end-effector configuration.

6. **(OPTIONAL) A plain text file called "README.txt" or similar.** This has any other information that may help the reviewer understand your submission.