HW5

Wednesday, October 5, 2022 1:58 PM

1. Complete your own function in "kinematics.py" to calculate the Geometric Jacobian.

   (a) Copy the "jacob" function from "kinematics_hw05.py" into your own "kinematics.py" file and complete all of the TODO sections. Make sure to examine the code, read the comments and try to really understand what you are doing.

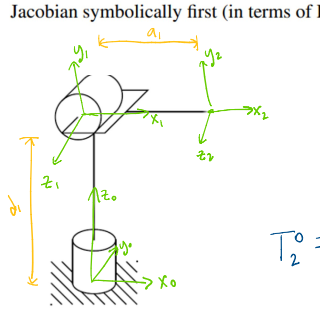
   (b) Download the learning suite file "hw05_test_jacobian.py" with your other files and run it to check the output of your function. Please be aware that if you run all the cells (including the commented cells which are markup and are similar to the markup cells in a jupyter notebook) it will generate the actual output you are to compare against.

   (c) Discuss what the zeros in the columns of the two different Jacobians mean. This may be much easier if you look at the visualization of the arm the robot described by the DH parameters given. An example of how to do this is shown in the "hw05_test_jacobian.py" file.

- The zeros in the last 3 rows of any column denote the inability of the robot to rotate in that direction (by itself)

- The zeros in the top 3 rows show where the next joint will go instantaneously is the joint is actuated.

(a) Calculate the Jacobian at the tip of the second link for the robot pictured below by hand (this would be the first two joints from HW 03, problem 2(e) where you should have calculated the DH parameters and forward kinematics already). Find this geometric Jacobian symbolically first (in terms of DH parameters).



	θ_z	d_z	a_z	α_z
1	q_1	d_1	0	$\pi/2$
2	q_2	0	a_1	0

$$T_2^0 = \begin{bmatrix} \cos(q_1)\cos(q_2) & -\sin(q_2)\cos(q_1) & \sin(q_1) & a_1\cos(q_1)\cos(q_2) \\ \sin(q_1)\cos(q_2) & -\sin(q_1)\sin(q_2) & -\cos(q_1) & a_1\sin(q_1)\cos(q_2) \\ \sin(q_2) & \cos(q_2) & 0 & a_1\sin(q_2)+d_1 \\ 0 & 0 & 0 & 1.0 \end{bmatrix}$$

$$J(q) = \begin{bmatrix} d_1\cos(q_1) & -(-a_1\sin(q_2)-d_1)\cos(q_1) \\ d_1\sin(q_1) & -(-a_1\sin(q_2)-d_1)\sin(q_1) \\ 0 & -a_1\sin^2(q_1)\cos(q_2)-a_1\cos^2(q_1)\cos(q_2) \\ \sin(q_1) & \sin(q_1) \\ -\cos(q_1) & -\cos(q_1) \\ 0 & 0 \end{bmatrix}$$

(b) Now let each link length be 30 cm and use your kinematics.py code to generate a serial arm that represents this arm. Compare the resulting Jacobian at a few different joint configurations using your own Jacobian code. How are the columns affected by the change in position and what does this mean physically (use at least one concrete example to discuss)? Also discuss what happens to the Jacobian when the second joint is moved to 90 degrees (so that the link is straight up or down).

```
from first set of q's, J is:
[[ 0.  -0. ]
 [ 0.3  0. ]
 [ 0.   0.3]
 [ 0.   0. ]
 [ 0.  -1. ]
 [ 1.   0. ]]
from second set of q's, J is:
[[-0.3 -0. ]
 [-0.   0 ]
 [ 0.   0.3]
 [ 0.   1. ]
 [ 0.  -0. ]
 [ 1.   0. ]]
from third set of q's, J is:
[[-0.2121 -0.    ]
 [ 0.2121  0.    ]
 [ 0.      0.3   ]
 [ 0.      0.7071]
 [ 0.     -0.7071]
 [ 1.      0.    ]]
When the second link is 90 deg and link_1 is 0, J is:
[[-0.  -0.3]
 [ 0.   0. ]
 [ 0.   0. ]
 [ 0.   0. ]
 [ 0.  -1. ]
 [ 1.   0. ]]
```

$q=[0,0]$

$q=[\frac{\pi}{2},0]$

$q=[\frac{\pi}{4},0]$

Jacobian Comparisons

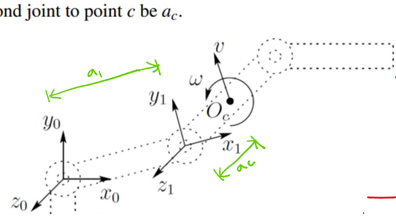
This means that instantaneously it will move in the negative x direction 0.3 m.

Changing q values changes where the instantaneous lengths show up.

Axis of rotation is $-y$ in frame 0

Any positive q would make the robot go in the negative x direction 0.3.

3. For the three-link planar manipulator shown below, compute the forward kinematics for the vector o_c (in terms of joint angles and DH parameters) and derive the manipulator Jacobian matrix in symbolic form for point o_c. Let the first link length be a_1 and the distance from the second joint to point c be a_c.

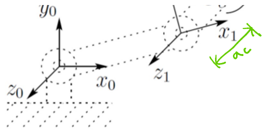


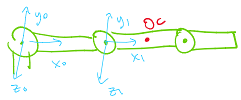
DH param

	θ_z	d_z	a_x	α_x
1	q_1	0	a_1	0
2	q_2	0	a_c	0

$a_1\sin(q_1)+a_c\sin(q_1+q_2)$

$(a\sin(q_1$





2	q_2	0	a_c	0

$$J = \begin{bmatrix} a_1 \sin(q_1) & a_1 \sin(q_1) + a_c \sin(q_1+q_2) \\ a_1 \cos(q_1) & -a_1 \cos(q_1) - a_c \cos(q_1+q_2) \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix}$$

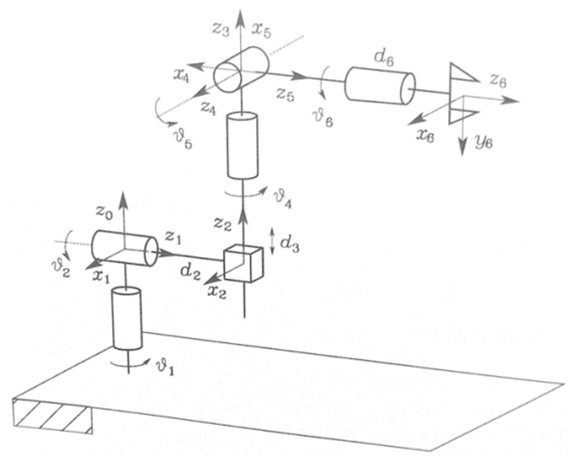
see code

4. Using your code from "kinematics.py," make a SerialArm object to represent the "Stanford" robot arm. The DH parameters for this robot are reported in Tables 2.3 and 2.5 of the book and you can see an image representing this arm below. However, the formulation in the book does not match the image assuming the shown robot is in the "zero" configuration with all its joints (i.e. $q = [0,0,0,0,0,0]$). Instead, the following DH parameters match the image below assuming that all joint variables "q" are zero.

Link	θ_i	d_i	a_i	α_i
1	q_1	0	0	$-\frac{\pi}{2}$
2	q_2	0.154	0	$\frac{\pi}{2}$
3	0	$q_3 + 0.25$	0	0
4	$q_4 - \frac{\pi}{2}$	0	0	$-\frac{\pi}{2}$
5	$q_5 - \frac{\pi}{2}$	0	0	$\frac{\pi}{2}$
6	$q_6 + \frac{\pi}{2}$	0.263	0	0

Please complete the following steps:

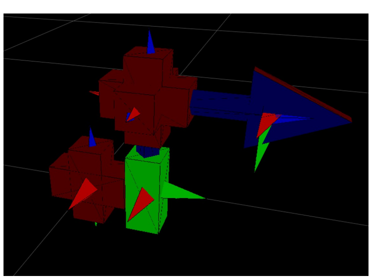
(a) Start by verifying that the DH parameters in the table above match the image below assuming all joint values "q" equal zero. Yep



(b) Make a SerialArm object using these defined DH parameters (hint: be careful about defining that one of these joints is a prismatic joint). Also include a transform as follows in the arm declaration: ✓

$$T^6_{tip} = \begin{bmatrix} & Rot_y(-\frac{\pi}{2}) & & \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(c) Confirm that the robot arm matches the frames shown below by 1) drawing the serial arm (as shown in "hw05_test_jacobian.py") with coordinate frames (see example from problem 1) or 2) using your "fk" function calling "add_frame" for each coordinate frame from 0 to 6 (probably in a for loop). (hint: Remember that for our "fk" function, that in addition to "q," we can pass an "index" variable. For example, to get T^0_1, we would pass in "1" for the index.)



(d) Finally, calculate the Jacobian when all joint angles are zero, and again when only the prismatic joint has moved in the positive direction 0.10 meters. Do the changes in the column values make sense? Please discuss.

```
The Jacobian for q=[0, 0, 0, 0, 0, 0] is:
[[-0.417  0.25   0.    -0.263 -0.     -0.   ]
 [ 0.     0.     0.     0.     0.      0.   ]
 [ 0.     0.     1.     0.     0.263   0.   ]
 [ 0.     0.     0.     0.     1.      0.   ]
 [ 0.     1.     0.     0.     0.      1.   ]
 [ 1.     0.     0.     1.     0.      0.   ]]
```

```
The Jacobian for q=[0, 0, 0.1, 0, 0, 0] is:
[[-0.417  0.45   0.    -0.263 -0.     -0.   ]
 [ 0.     0.     0.     0.     0.      0.   ]
 [ 0.     0.     1.     0.     0.263   0.   ]
 [ 0.     0.     0.     0.     1.      0.   ]
 [ 0.     1.     0.     0.     0.      1.   ]
 [ 1.     0.     0.     1.     0.      0.   ]]
```

the second joint will now cause more

```
[ 0.     1.     0.     0.     0.     1.   ]
[ 1.     0.     0.     1.     0.     0.   ]]
```

```
[ 0.     1.     0.     0.     0.     1.   ]
[ 1.     0.     0.     1.     0.     0.   ]]
```

These values make sense because the second joint will now cause more movement in the positive x-direction (in frame 0) dependent on how it moved.