Homework 3: Kinematics

24-760 Robot Dynamics & Analysis
Fall 2024

Name: ______________________________

Note: For homework submission, please submit a zipped folder of your Matlab files to "Homework 3 Programming" in Gradescope. Submit solutions for `Problem 2)` as a Matlab script following the given template `HW3_sudent.m` and include all used functions in .zip format. Fill in all the `TODO` sections and clearly label sections based on which part they are for. Please use the precise variable names that we define in the template and do not overwrite them in later sections.

Problem 1) Matlab

You may call and reuse the functions from the last homework.

1.1) In folder `studentDefinedTests`, complete the function `compare_rotm()`.

Generate a random *unit* vector `w` (as `axang2rotm` requires a unit vector for the axis) and rotation amount `t`. Compute the rotation matrix `rotm_matlab` generated with Matlab built-in function `axang2rotm`, and `rotm` generated with your function `angvel2skew` and the Matlab function `expm`.

The function should return `w, t, rotm` and `rotm_matlab`, and `rotm` and `rotm_matlab` should be the same.

1.2) In folder `studentDefinedTests`, complete the function `compare_tform()`.

Generate a random *unit* vector `w`, rotation amount `t`, and twist pitch `p`. Using `w` and `t`, compute the rigid body transformation matrix `g_w_matlab` generated with `axang2tform`, and `g_w` generated with your function `twist2rbvel` and the Matlab function `expm`.

Then using a pure translation with a velocity of `v = wp` and the amount `t`, compute the rigid body transformation matrix `g_v_matlab` generated with `trvec2tform` and `g_v` generated with your function `twist2rbvel` and the Matlab function `expm`.

Finally, using `w`, `t`, and `v`, compute the rigid body transformation matrix `g` generated with your function `twist2rbvel` and the Matlab function `expm` and `g_matlab` generated with the composition of `axang2tform` and `trvec2tform`.

The function should return `w, t, p, g_w, g_w_matlab, g_v, g_v_matlab, g` and `g_matlab`.

1.3) In folder `studentDefinedTests`, complete the function `compare_twist()`.

Generate a random transformation `g` as before and a random twist `V`. First, treating `V` as a body velocity, compute the conversion to spatial velocity `V_s_Ad_g` using `tform2adjoint`, and the conversion in homogeneous coordinates `V_s_tform` using `twist2rbvel` and `rbvel2twist`.

Repeat the test the other way, treating `V` as a spatial velocity and converting to body velocity and compute `V_b_Ad_g` and `V_b_tform`.

The function should return `V, g, V_s_Ad_g, V_s_tform, V_b_Ad_g` and `V_b_tform`.

Problem 2) ABB IRB 6620

In this problem you will be modeling the kinematics of the ABB IRB 6620, a large industrial robot capable of lifting 150kg[1]. We have included the key datasheet figures on the next page. (One ambiguity in the figure is the location of the 5th axis (E), which is in between the "887" and "200" measurements). A video of the robot in operation can be found here:
https://www.youtube.com/watch?v=LtTTpW_2flw

As always, you may use the functions you defined in the last homework as well as any of the Robotics System Toolbox functions. You will also need to make use of the Symbolic Math Toolbox for some of the calculations, which has some tutorials here if you have not used it before:
https://www.mathworks.com/help/symbolic/getting-started-with-symbolic-math-toolbox.html

2.1) Begin by writing out by hand the rigid body transformation from the stationary frame to the tool frame in the configuration shown on the next page, `g_st(0)`. Use a stationary base frame with the $x-y$ plane on the floor, the $+x$ axis pointing to the right in the bottom center figure, and the $+z$ axis aligned with Axis 1 in the top figure. Use a tool frame with a $+x$ axis pointing out of the end, the $y-z$ plane flush with the tool plate, and a $+z$ pointing upwards. Submit in the template script that assigns a numerical expression called `gst0`.

2.2) Find the general forward kinematics in Matlab. Define a set of rigid body transformations that represent the displacement between each successive link, and include an explanation of how you defined any intermediate frames. The angles should be represented symbolically, e.g. `syms q1 real`. Use these to define $g_{st}(q)$, where $q = [q_1, q_2, q_3, q_4, q_5, q_6]^T$, the forward kinematics, and compare to $g_{st}(0)$ which you found above. Submit in the template script that produces a symbolic expression called `gst`, a numerical expression `gst0_sym` and performs the comparison.

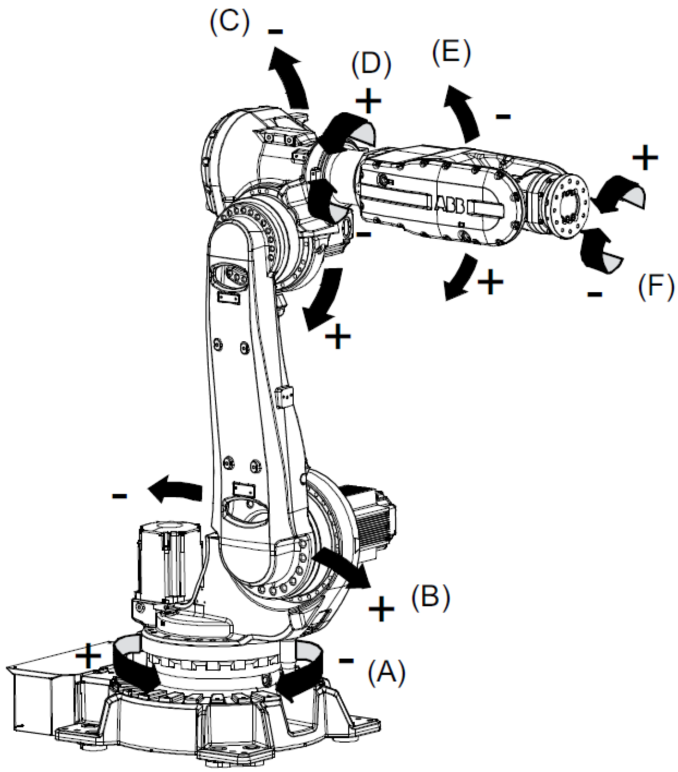
2.3) Now use the product of exponentials formula to recalculate the forward kinematics. Does this give the same forward kinematics? Submit in the template script that produces a symbolic expression called `gst_exp` and performs the comparison. (*Hint: You may need to use* `simplify`.)

2.4) Solve the inverse kinematics problem to find a configuration of the robot where the end effector is located $100mm$ in the $+y$ direction from the home configuration, with the same orientation. To do this, define an optimization problem using your forward kinematic map and solve for q. You do not need to consider joint limits or self-collision. Verify the solution by passing it into the forward kinematics $g_{st}(q)$ and comparing to the desired configuration with a tolerance of 0.1. Submit in the template script that produces a numerical vector called `q_sol` and performs the comparison. (*Hint: You may need to use* `matlabFunction` *and* `fminunc`)

[1] https://new.abb.com/products/robotics/industrial-robots/irb-6620

Manipulator axes

IRB 6620



xx0900000454

Pos	Description	Pos	Description
A	Axis 1	B	Axis 2
C	Axis 3	D	Axis 4
E	Axis 5	F	Axis 6

Dimensions IRB 6620