WPI

RBE 501 – Robot Dynamics

Spring 2022 – Instructor: L. Fichera

Homework 2

INSTRUCTIONS

- **This homework assignment includes two sections:**
  - **Section 1**: Preliminaries – you will create MATLAB functions that you will later need to solve the problems in Section 2
  - **Section 2**: Position and Velocity Kinematics of three different robotic arms - to solve this section, you will have to have Peter Corke's Robotics toolbox installed on your system (ver 10.4 or newer). See https://petercorke.com/toolboxes/robotics-toolbox/.
- **Submit your solutions for section 1 through MATLAB grader:** https://grader.mathworks.com/courses/65880-rbe-501-robot-dynamics-spring-2022/assignments/184315-homework-2
- **Submit your solutions for section 2 through Gradescope:**
  - You will find separate entries on Gradescope to submit:
    1. A zip file containing your MATLAB code
       - All source files must be professionally commented; ideally, anyone not familiar with the code should be able to understand what it does just from reading the comments;
       - All functions must include a help section;
       - Use I/O and/or figures to illustrate what your code does. Code that runs silently, that is, without generating any visible output, will not be graded;
    2. A PDF file detailing your solution; the PDF **must** contain:
       - All necessary schematics/diagrams used to derive the solution;
       - Commentary to explain your thought process (for instance: "*To solve the inverse kinematics, I apply method X*"); if unsure how much level of detail is enough, err on the side of verbosity; no points will be awarded for just reporting the right solution;
       - All calculations (even though they are numerically performed in MATLAB);
       - Instructions on how to run your code (for instance: "*The inverse kinematics is solved in source file* `ik.m`");
  - Check your code before submitting it! **Code that runs with an error will not be graded.**
  - The PDF must be clearly legible. Plots and figures must be of professional quality. Figure DPI >= 300. Remember to add axis labels and units.
- **Due date: Wednesday 23-Feb-22 at 3:00 pm (start of class)**

<u>Section 1: Preliminaries (25 points total – 5 points for each correct answer)</u>

This section must be completed online at:

https://grader.mathworks.com/courses/65880-rbe-501-robot-dynamics-spring-2022/assignments/184315-homework-2

1. Converting exponential coordinates of rotation to rotation matrices (5 points)

2. Converting twists to homogeneous transformation matrices (5 points)

3. Product of Exponentials Formula (5 points)

4. Adjoint Transformations (5 points)

5. Jacobians in the space frame (5 points)

The remainder of this page was intentionally left blank.

Section 2: Kinematics and Velocity Control of three robotic arms (75 points total)

Problem 1 (25 points):

The 3-DoF manipulator in Figure 1 consists of a three degrees-of-freedom manipulator, with link lengths as follows: L1: 0.3 m, L2: 0.3 m, L3: 0.3 m. A model of this manipulator was pre-created for your convenience: open the zip archive with the starting MATLAB code provided on Canvas, then run the `hw2problem1.m` file.

Note: *running this script will first display the robot, then it will result in an error. The error is expected, and it occurs because the forward kinematics of this robot is not implemented yet – we will work on this next.*

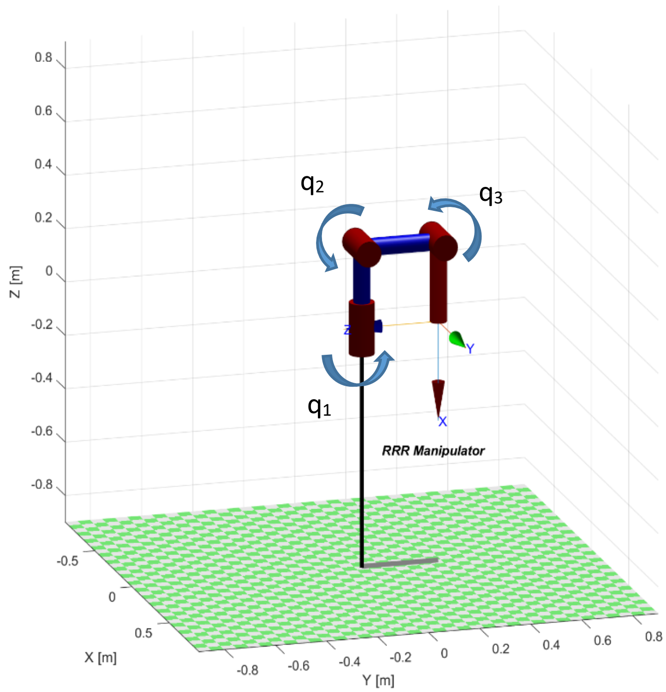

Figure 1: Three DoF manipulator.

a. **Calculate each of the screw axes $\xi_i = (\omega_i, v_i)$ with respect to the space frame (5 points)**
   Once you are done, complete line 38 of `hw2problem1.m` to hardcode the screw axes into the script.
b. **Calculate the manipulator kinematics using the product of exponentials formula (5 points)**
   To complete this step, you will have to first calculate the homogeneous transformation matrix for the home configuration (line 44), then use this information to calculate the forward kinematics (line 62). You should be able to use the `fkine` function developed in Section 1 of this homework.
c. **Calculate the space Jacobian of the manipulator (5 points)**
   Calculate the space Jacobian by adding code to line 96. You should be able to use the `jacob0` function developed in Section 1 of this homework.
d. **Calculate the inverse kinematics for 30 different configurations using a numerical approach (10 points)**
   To solve this step, fill out the code on lines 150-151. These two lines of code should generate a vector `deltaQ` that should be applied to the vector of joint variables to move closer to the target pose. In your report, comment on what numerical method you chose to implement; take note of the configurations for

which the inverse kinematics algorithm fails (if any), investigate the cause of the failure (ill-conditioned Jacobian? Unreachable configuration?), and attempt to find a different solution for these configurations.

Problem 2 (25 points):

Repeat the same steps of problem 1 for the *elbow manipulator* shown below. Assume the same link lengths as in problem 1. A model of this manipulator was pre-created for your convenience: open the zip archive with the starting MATLAB code provided on Canvas, then run the `hw2problem2.m` file.

Note: *unlike the prior problem, the starting code provided in* `hw2problem2.m` *is minimal. You are welcome to structure this script the way you want. Remember to comment your code adequately and to use I/O and/or figures to illustrate what your code does. Code that runs silently, that is, without generating any visible output, will not be graded.*

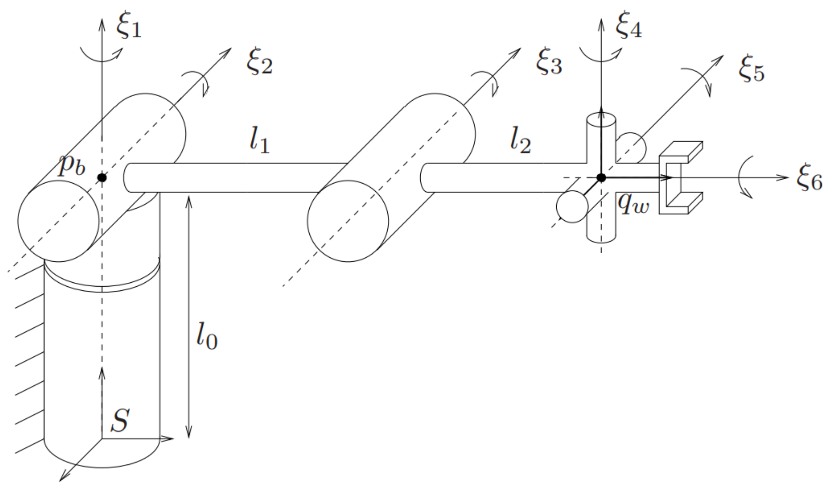

Figure 2: Elbow Manipulator

The remainder of this page was intentionally left blank.

Problem 3 (25 points):

Repeat the same steps of problem 1 for the Staubli TX-40 *robot.* The robot schematic and dimensions are shown in Figure 3. Run `hw2problem3.m` to create a model of the robot.

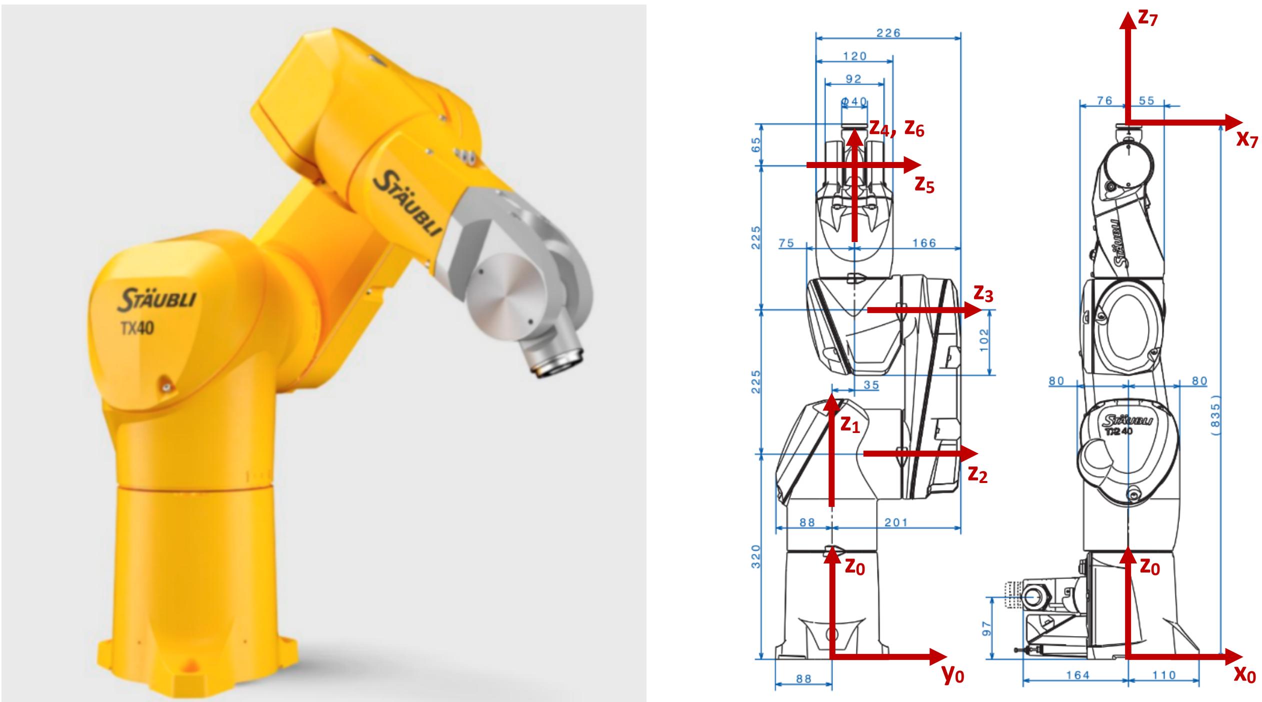

Figure 3: (a) The Staubli TX-40 robotic arm; (b) Joint axes and dimensions. All dimensions are in millimeters. Joints are numbered consecutively from 1 to 6. Frame {0} is the space frame. Frame {7} is the end-effector frame. ***Note****: the schematic does* *not* *show joint frames – only the joint axes (z_1, z_2, …, z_6). Recall that joint frames are not needed to calculate the kinematics with the Product of Exponentials Formula.*