Università degli studi di Genova

DIBRIS

Department of Computer Science and Technology,
Bioengineering, Robotics and System Engineering

Modelling and Control of Manipulators

Exam session: 09/01/2024

Professors:

Enrico Simetti
Giorgio Cannata

Tutors:

Andrea Tiranti
Francesco Giovinazzo
George Kurshakov

January 9, 2024

Mathematical expression	Definition	MATLAB expression
$< w >$	World Coordinate Frame	w
${}^a_bR$	Rotation matrix of frame $< b >$ with respect to frame $< a >$	aRb
${}^a_bT$	Transformation matrix of frame $< b >$ with respect to frame $< a >$	aTb
${}^aO_b$	Vector defining frame $< b >$ wit respect to frame $< a >$	aOb

Table 1: Nomenclature Table

1 Exam description

The Modelling and Control of Manipulator exam consists of a single assignment that summarizes all the topics discussed during the lectures. The following exercises should be solved using Matlab; using the tools implemented during the practical lessons.

For the exam, you have a compressed folder in which you can find the code template, which consists of a $main.m$ and a folder $/include$. The main file should be compiled following the template, if it is completed correctly the code can be launched and should output the results of the exam.

While in the $"include"$ folder you can find all the functions to be implemented for the exam; you are not bound to follow a standard template for the functions, try to respect at least the input and output variables of each function. You can add other functions (implemented by yourself) if you deem it necessary.

1.1 Exam

Given the CAD model of a 6 DOF manipulator in Figure 1:

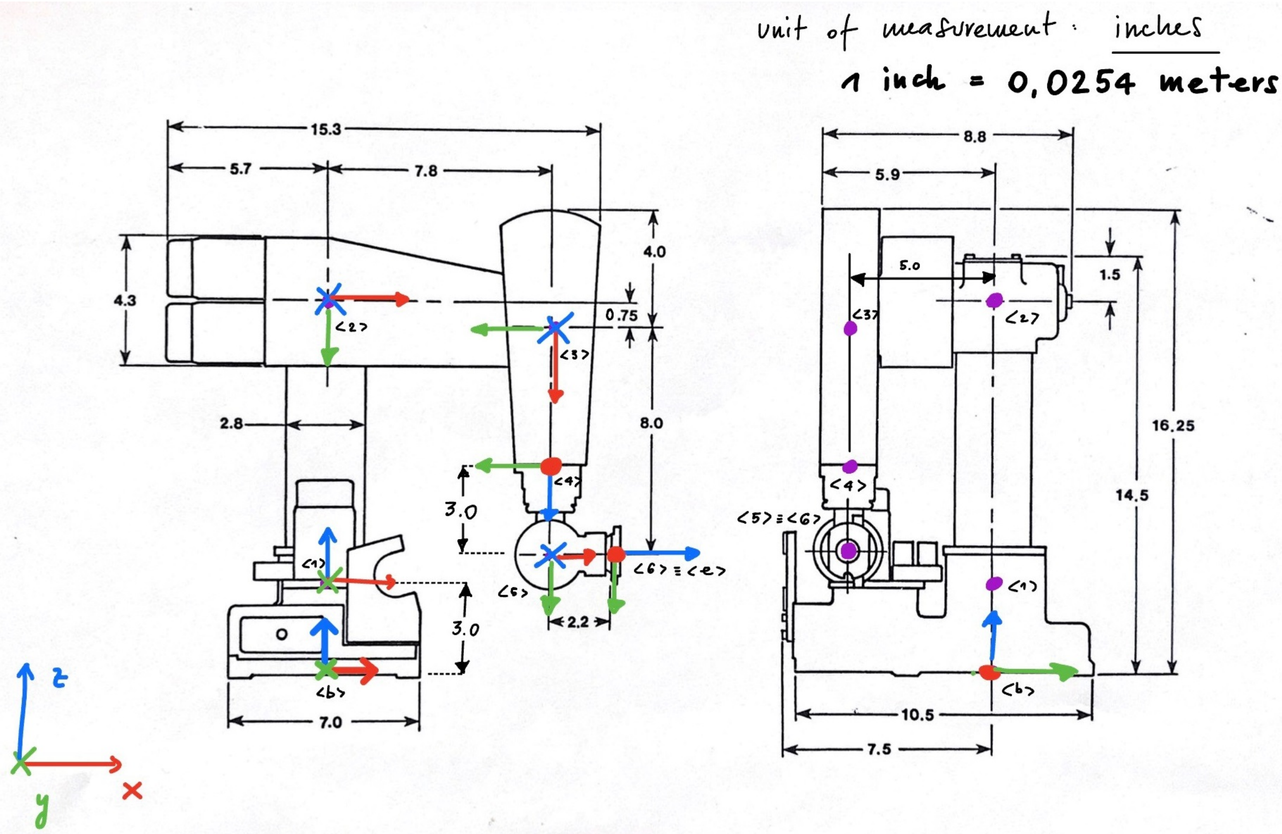

Figure 1: CAD model of the robot with the frames

Q1.1 Define all the model matrices, by filling the structures in the *BuildTree()* function. Note that the units of measurement of the CAD are expressed in inches, but you will have to work in meters. The Figure shows the position of the frames for $q = 0$.

Q1.2 Implement a function called *DirectGeometry()* which can calculate how the transformation matrix of link j with respect to link $j-1$ will rotate if joint j rotates. Then, develop a function called GetDirectGeometry() which returns all the model matrices given the following joint configuration:

- $\mathbf{q}^* = \begin{bmatrix} 0 & -\pi/2 & -\pi/2 & 0 & \pi/2 & 0 \end{bmatrix}^\top$;

Q1.3 Calculate all the transformation matrices between any two links and between a link and the base, filling respectively: *GetFrameWrtFrame()*, *GetTransformationWrtBase()*.

- Test the function *GetTransformationWrtBase()* by computing the transformation of the end effector with respect to the base ${}^b_eT$
- Test the function *GetFrameWrtFrame()* by computing the transformation matrix ${}^2_3T$ from frame $< 2 >$ to frame $< 3 >$

- Test the function *GetFrameWrtFrame()* by computing the transformation matrix ${}^6_33T$ from frame $< 6 >$ to frame $< 3 >$

Q1.4 Compute the Jacobian matrix for the manipulator for the previous joint configuration $\mathbf{q}^*$ using the function "*GetJacobian()*".

Q1.5 Compute the cartesian error between the robot end-effector frame ${}^b_eT$ and the goal frame ${}^b_gT$, knowing that:

- The initial configuration of the robot is $\mathbf{q}_0 = \begin{bmatrix}\pi/2 & -\pi/4 & -\pi/4 & 0 & \pi & 0\end{bmatrix}^\top$
- The goal position with respect to the base frame is ${}^bO_g = \begin{bmatrix}0.2 & 0 & 0.15\end{bmatrix}^\top$
- The goal frame is defined starting from the initial position of the end-effector. In detail, the roll-pitch-yaw parameters of the rotation matrix ${}^e_gR$ in the initial configuration are: $\eta = \begin{bmatrix}0 & \pi/6 & 0\end{bmatrix}$.

Q1.6 Compute the desired angular velocities and the linear reference velocities of the end-effector with respect to the base: ${}^b\nu^*_{e/0} = \alpha \cdot \begin{bmatrix}\omega^*_{e/0} \\ v^*_{e/0}\end{bmatrix}$, such that $\alpha = 0.8$ is the gain.

Q1.7 Compute the desired joint velocities.

Q1.8 Simulate the robot motion by implementing the function: *"KinematicSimulation()"*.