Stony Brook University

Final Project

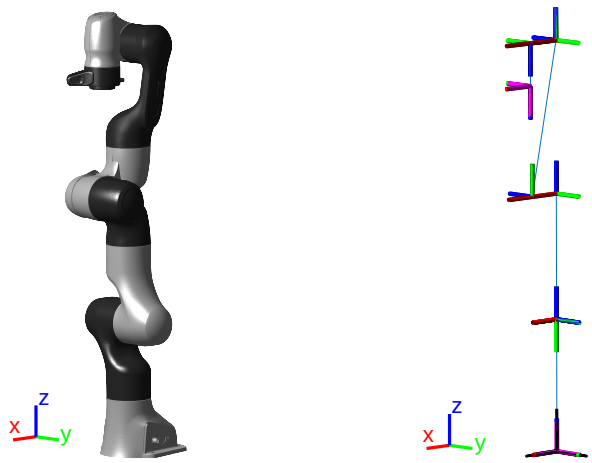
MEC 529: Introduction to Robotics

Spring 2023

Instructor	Amin Fakhari, Ph.D.
Assigned Date	Friday, Apr. 21, 2023
Due Date	Monday, May 15, 2023, 2:00 PM

Consider URDF and Specification files of the 6/7-DOF open-chain robot manipulator assigned to your group.

1. Using the URDF file, determine the screw axes ($\boldsymbol{\mathcal{S}}_i \in \mathbb{R}^6$ or $\boldsymbol{\mathcal{B}}_i \in \mathbb{R}^6$) of the robot joints when the robot is at its home (zero) configuration. Note that a fixed space frame $\{s\}$ is attached to the robot base and a body frame $\{b\}$ is attached to the robot end-effector. *Hint*: To have a better understanding of the pose of the robot axes at its home (zero) configuration, you can import the URDF file into MATLAB using the function `importrobot` (Fig. 1-a) and show it using the function `show` while its `Visuals` argument is `off` (Fig. 1-b). For more detail, refer to the attached sample MATLAB file.



(a) Mesh view (b) Frame view

Figure 1: A robot manipulator imported into MATLAB using the provided URDF file.

2. In order to make sure that the screw axes are determined correctly, use your forward kinematics function (`FK_SpaceForm` or `FK_BodyForm` from HW#5) to compute the end-effector configuration $\boldsymbol{T}_{sb} \in SE(3)$ for an arbitrary set of feasible joint angles $\boldsymbol{\theta} \in \mathbb{R}^n$ (refer to the Specification file to find the joint limits), and then, use the attached `triad` function to visually verify that the end-effector configuration $\boldsymbol{T}_{sb}$ coincides with the end-effector $\{b\}$-frame of the robot imported into MATLAB at the same configuration $\boldsymbol{\theta}$.

3. By considering the robot joint limits from the Specification file, (roughly) determine the robot reachable workspace in Cartesian space $\mathbb{R}^3$.

4. Determine if the robot is kinematically redundant for performing a general task by its end-effector in Cartesian space $\mathbb{R}^3$. Moreover, determine as many (boundary and internal) singular configurations $\boldsymbol{\theta}^*$ of the robot as possible, and verify them using the rank or condition number of space Jacobian $\boldsymbol{J}_s \in \mathbb{R}^{6\times n}$, body Jacobian $\boldsymbol{J}_b \in \mathbb{R}^{6\times n}$, or geometric Jacobian $\boldsymbol{J}_g \in \mathbb{R}^{6\times n}$. Note that the geometric Jacobian $\boldsymbol{J}_g$ is defined as $\begin{bmatrix}\boldsymbol{\omega}_s \\ \dot{\boldsymbol{p}}\end{bmatrix} = \boldsymbol{J}_g(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}$ and you can compute it using

$$\boldsymbol{J}_g = \begin{bmatrix}\boldsymbol{J}_{g,\omega} \\ \boldsymbol{J}_{g,v}\end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ -[\boldsymbol{p}] & \boldsymbol{I}\end{bmatrix}\boldsymbol{J}_s$$

where $\boldsymbol{\omega}_s \in \mathbb{R}^3$ is the angular velocity of the end-effector frame $\{b\}$ expressed in $\{s\}$, $\boldsymbol{p} \in \mathbb{R}^3$ and $\dot{\boldsymbol{p}} \in \mathbb{R}^3$ are the position and linear velocity of the origin of the end-effector frame $\{b\}$ expressed in $\{s\}$, respectively, $\boldsymbol{I} \in \mathbb{R}^{3\times 3}$ is an identity matrix, and $[\boldsymbol{p}] \in so(3)$. You can use the Jacobian functions `J_SpaceForm` or `J_BodyForm` you have written for HW#6.

5. (*Optional Extra Credit Part*) At two arbitrary configurations of the robot, one in the neighborhood of a singular configuration and another away from a singular configuration, compute Jacobian $\boldsymbol{J}_g$, and separate them into $\boldsymbol{J}_{g,\omega} \in \mathbb{R}^{3\times n}$ (the portion corresponding the effect of the joint rates on the angular velocity) and $\boldsymbol{J}_{g,v} \in \mathbb{R}^{3\times n}$ (the portion corresponding the effect of the joint rates on the linear velocity). Calculate the directions and lengths of the principal semi-axes of the three-dimensional angular-velocity manipulability ellipsoid (based on $\boldsymbol{J}_{g,\omega}$) and the directions and lengths of the principal semi-axes of the three-dimensional linear-velocity manipulability ellipsoid (based on $\boldsymbol{J}_{g,v}$). Then, plot the ellipsoids on the robot imported into MATLAB, centered at $\boldsymbol{p}$, the origin of the end-effector frame $\{b\}$, and each ellipsoid in a separate figure for each configuration. Choose the proper scaling for the ellipsoids so that they can be easily visualized (e.g., the ellipsoids should usually be smaller than the robot but not so small that they cannot be easily seen). If the results near a singular configuration do not match what you expect, try with another singular configuration, and discuss the reason.

6. The inverse kinematics of a robot refers to the calculation of the joint variables $\boldsymbol{\theta} \in \mathbb{R}^n$ from a desired configuration of the end-effector frame $\{b\}$ with respect to a fixed space frame $\{s\}$.

   (a) Using Newton–Raphson method, write a general MATLAB function `IK_BodyForm` that returns the joint variables $\boldsymbol{\theta} \in \mathbb{R}^n$ by taking the desired end-effector configuration $\boldsymbol{T}_{sd} \in SE(3)$, an initial guess $\boldsymbol{\theta}_0 \in \mathbb{R}^n$, and a vector $\boldsymbol{\epsilon} = (\epsilon_\omega, \epsilon_v) \in \mathbb{R}^2$ which is concatenation of small threshold values $\epsilon_\omega \in \mathbb{R}$ (in radians) and $\epsilon_v \in \mathbb{R}$ (in meters) on the final error. Since this numerical inverse kinematics also needs calculation of the body Jacobian and the geometric forward kinematics in each iteration, consider a matrix $\mathbf{B} \in \mathbb{R}^{6\times n}$ in which each column of the matrix corresponds to the screw axes $\boldsymbol{\mathcal{B}}_i \in \mathbb{R}^6$ of the robot joints expressed in frame $\{b\}$ when the robot is at its home configuration (i.e., $\mathbf{B} = [\boldsymbol{\mathcal{B}}_1, ..., \boldsymbol{\mathcal{B}}_n]$), and also a transformation matrix $\boldsymbol{M} \in SE(3)$ which is the configuration of $\{b\}$ relative to $\{s\}$ when the robot is in its home configuration, as the inputs of the function `IK_BodyForm`. Therefore, $\boldsymbol{\theta} =$ `IK_BodyForm`$(\boldsymbol{T}_{sd}, \boldsymbol{\theta}_0, \boldsymbol{\epsilon}, \mathbf{B}, \boldsymbol{M})$, and in this function, you can use the forward kinematics function $\boldsymbol{T}_{sb} =$ `FK_BodyForm`$(\mathbf{B}, \boldsymbol{M}, \boldsymbol{\theta})$ and the body Jacobian function $\boldsymbol{J}_b =$ `J_BodyForm`$(\mathbf{B}, \boldsymbol{\theta})$ you have written for HW#5 and HW#6.

   (b) Using Newton–Raphson method, write a general MATLAB function `IK_SpaceForm` that returns the joint variables $\boldsymbol{\theta} \in \mathbb{R}^n$ by taking the desired end-effector configuration $\boldsymbol{T}_{sd} \in SE(3)$, an initial guess $\boldsymbol{\theta}_0 \in \mathbb{R}^n$, and a vector $\boldsymbol{\epsilon} = (\epsilon_\omega, \epsilon_v) \in \mathbb{R}^2$ which is concatenation of small threshold values $\epsilon_\omega \in \mathbb{R}$ (in radians) and $\epsilon_v \in \mathbb{R}$ (in meters) on the final error. Since this numerical inverse kinematics also needs calculation of the space Jacobian and the geometric forward kinematics in each iteration, consider a matrix $\mathbf{S} \in \mathbb{R}^{6\times n}$ in which each column of the matrix corresponds to the screw axes $\boldsymbol{\mathcal{S}}_i \in \mathbb{R}^6$ of the robot joints expressed in frame $\{s\}$ when the robot is at its home configuration

(i.e., $\mathbf{S} = [\boldsymbol{\mathcal{S}}_1, ..., \boldsymbol{\mathcal{S}}_n]$), and also a transformation matrix $\boldsymbol{M} \in SE(3)$ which is the configuration of $\{b\}$ relative to $\{s\}$ when the robot is in its home configuration, as the inputs of the function `IK_SpaceForm`. Therefore, $\boldsymbol{\theta} =$ `IK_SpaceForm`$(\boldsymbol{T}_{sd}, \boldsymbol{\theta}_0, \boldsymbol{\epsilon}, \mathbf{S}, \boldsymbol{M})$, and in this function, you can use the forward kinematics function $\boldsymbol{T}_{sb} =$ `FK_SpaceForm`$(\mathbf{S}, \boldsymbol{M}, \boldsymbol{\theta})$ and the space Jacobian function $\boldsymbol{J}_s =$ `J_SpaceForm`$(\mathbf{S}, \boldsymbol{\theta})$ you have written for HW#5 and HW#6.

(c) For an arbitrary and feasible end-effector configuration $\boldsymbol{T}_{sd} \in SE(3)$ in the robot workspace, use the numerical inverse kinematics function `IK_SpaceForm` or `IK_BodyForm` to find the corresponding joint variables $\boldsymbol{\theta}$. Verify your results using the forward kinematics function. Note that numerical inverse kinematics is intended to find a solution close to the initial guess. Make sure the solution does not violate the robot joint limits.

7. The goal of trajectory planning is to generate the reference inputs $\boldsymbol{\theta}_d(t)$ to the motion control system which ensures that the manipulator executes the planned trajectories. In the following parts, you are asked to plan different desired offline trajectories for the robot. In planning the trajectories, you should make sure that the trajectories do not violate the joint limits (i.e., the end-effector motion remains inside the robot workspace) and the joint velocity/acceleration limits, and also the robot does not pass near the singular configurations determined in (4). Since the control action on the robot is carried out in the joint space by motors, if the trajectory planning is performed in the end-effector operational space, you have to use inverse kinematics to find the corresponding time sequence of joint variables $\boldsymbol{\theta}_d(t)$ along the path. After computing $\boldsymbol{\theta}_d(t)$ in each part, simulate the motion of the robot imported into MATLAB (refer to the attached sample MATLAB file) and verify that the task is properly performed.

(a) Choose two arbitrary and feasible desired configurations $\boldsymbol{T}_{sd,\text{start}} \in SE(3)$ and $\boldsymbol{T}_{sd,\text{end}} \in SE(3)$ for the end-effector frame $\{b\}$ (as much as possible away from each other with different orientations). Plan a smooth trajectory with a straight-line path in the robot joint space $\boldsymbol{\theta}(s) \in \mathbb{R}^n$ and a trapezoidal velocity profile for time scaling $s(t) \in \mathbb{R}$ (where $t \in \mathbb{R}$ is time in seconds) to take the robot from the start configuration to end configuration with minimum motion time $T \in \mathbb{R}$.

(b) For the same configurations $\boldsymbol{T}_{sd,\text{start}}$ and $\boldsymbol{T}_{sd,\text{end}}$, plan a smooth trajectory with a straight-line path in $SE(3)$ for the configuration of the end-effector frame $\{b\}$ $\boldsymbol{T}(s) \in SE(3)$ and a 3rd-order polynomial for time scaling $s(t)$ to take the robot from the start configuration to end configuration with minimum motion time T (note that if the path does not remain in the robot workspace or it passes near a kinematic singularity, change the arbitrary configurations chosen in (7)a and repeat the steps).

(c) For the same configurations $\boldsymbol{T}_{sd,\text{start}}$ and $\boldsymbol{T}_{sd,\text{end}}$, plan a smooth trajectory with a straight-line path in Cartesian space $\mathbb{R}^3$ for the position of the origin of the end-effector frame $\{b\}$ $\boldsymbol{p}(s) \in \mathbb{R}^3$ and a straight-line path in $SO(3)$ for the orientation of the end-effector frame $\{b\}$ $\boldsymbol{R}(s) \in SE(3)$ and 3rd-order polynomials for time scaling $s(t)$ to take the robot from the start configuration to end configuration with minimum motion time T (note that if the path does not remain in the robot workspace or it passes near a kinematic singularity, change the arbitrary configurations chosen in (7)a and repeat the steps).

(d) Plan a smooth trajectory to move the origin of the end-effector frame $\{b\}$ on an arbitrary closed circular path in Cartesian space $\mathbb{R}^3$ with minimum motion time T. Use a 5th-order polynomial for time scaling $s(t)$ and keep the orientation of the end-effector frame $\{b\}$ fixed during this motion.

(e) Plan a smooth trajectory in task space in which the origin of the end-effector frame $\{b\}$ passes through at least four arbitrary via points (including the start and end points) in Cartesian space $\mathbb{R}^3$ at arbitrary specified times while keeping the orientation of the end-effector frame $\{b\}$ fixed during this motion. Use cubic polynomials with continuous velocities and accelerations at via points (i.e., splines) to design the trajectory (without separating the path and time scaling) with an arbitrary motion time T.

(f) Use a proper trajectory generation method to create an arbitrary English capital letter (except "I"!) in a plane in Cartesian space $\mathbb{R}^3$ by the motion of the origin of the end-effector frame $\{b\}$.

In the motion simulation of each part, plot the configurations $\boldsymbol{T}_{sd,\text{start}}$ and $\boldsymbol{T}_{sd,\text{end}}$ (or any other via points for instance for (7)e) using the `triad` function and also plot the path of the origin of the end-effector frame $\{b\}$ in Cartesian space $\mathbb{R}^3$ to verify the desired task is properly performed by the imported robot. Moreover, plot $\boldsymbol{p}(t)$, $\boldsymbol{\theta}_d(t)$, $\dot{\boldsymbol{\theta}}_d(t)$, and $\ddot{\boldsymbol{\theta}}_d(t)$ for each task along with the joint position/velocity/acceleration limits to verify all the joint limits are also satisfied.

Notes:

- Your report should include a description of your results for every part of the project with supporting figures.
- Add proper comments to your code, which detail what each part of the code is doing.
- Submit your report and code files in a single Zip file on Brightspace. A report without its supporting code files and code files without a supporting report is NOT acceptable.
- Make sure to submit all the files/functions required to properly execute your code.