MIT - Mechanical Engineering Department

2.12 Introduction to Robotics - Fall 2016 - HW2

Released: September 22, 2016 Due: September 28, 2016

Notes:

- Turn in individual problems in **separate pages** and put your name on **each page**.
- Show the process to get to the answer and not just the solution.
- Clearly highlight solutions, i.e., draw a rectangle around them.

Problem 1

RRP Surgical Manipulator

The manipulator in Figure 1 is a three dof surgical manipulator. We define coordinate frame $\{O_0, (x_0, y_0, z_0)\}$ at the end of link 0 with axis z_0 pointing down. The length between origins O_0 and O_1 is zero (for illustration purposes these points are separated). Joint 1 rotates link 1 about axis x_0, and joint 2 rotates link 2 about axis y_1, both in the right hand sense. The third joint is prismatic, extending the end effector along z_2. Coordinate frames $\{O_1, (x_1, y_1, z_1)\}$ and $\{O_2, (x_2, y_2, z_2)\}$ are attached to the ends of link 1 and to the end effector. Figure 1 also shows the arm configuration for $(\theta_1 = \pi/6, \theta_2 = \pi/6, d_3 = 20\ cm)$. Unit vectors, $\hat{x}_i, \hat{y}_i, \hat{z}_i$ for $i = 0, 1, 2$ are attached to each coordinate frame. Answer the following questions:

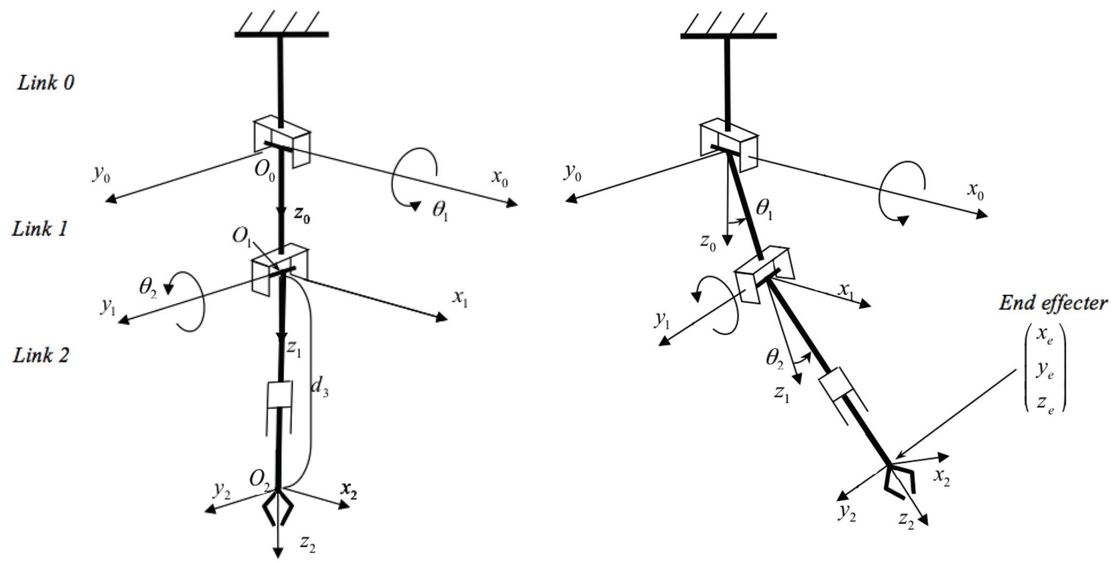

Figure 1: Left: Home Position. Right: Configuration $(\theta_1 = \pi/6, \theta_2 = \pi/6, d_3 = 20\ cm)$.

a. Obtain the coordinate unit vectors $\hat{x}_1, \hat{y}_1, \hat{z}_1$ from frame 1 as viewed from frame 0. That is, express them in coordinates of $\{O_0, (x_0, y_0, z_0)\}$. Construct the homogeneous transformation ${}^0_1T$ from frame 1 to 0.

b. Obtain the unit vectors $\hat{x}_2, \hat{y}_2, \hat{z}_2$ in coordinates of frame 1 and the homogeneous transformation ${}^1_2T$.

c. Construct the homogeneous transformation between frame 2 and 0 and extract the orientation of the end effector, i.e, vectors $\hat{x}_2, \hat{y}_2, \hat{z}_2$, as viewed from frame 0.

d. Obtain the end effector position $(x_e, y_e, z_e)^T$, in frame 0 coordinates.

Problem 2

Inspection Shuttle Manipulator

An astronaut is operating a shuttle manipulator with an inspection end-effector attached to the tip of a planar RRR arm with revolute joints $(\theta_1, \theta_2, \theta_3)$ in series, as in Figure 2. To facilitate the inspection, we attach the coordinate frame $\{O, (x, y)\}$ to the object, located at distance L from the base of the robot.

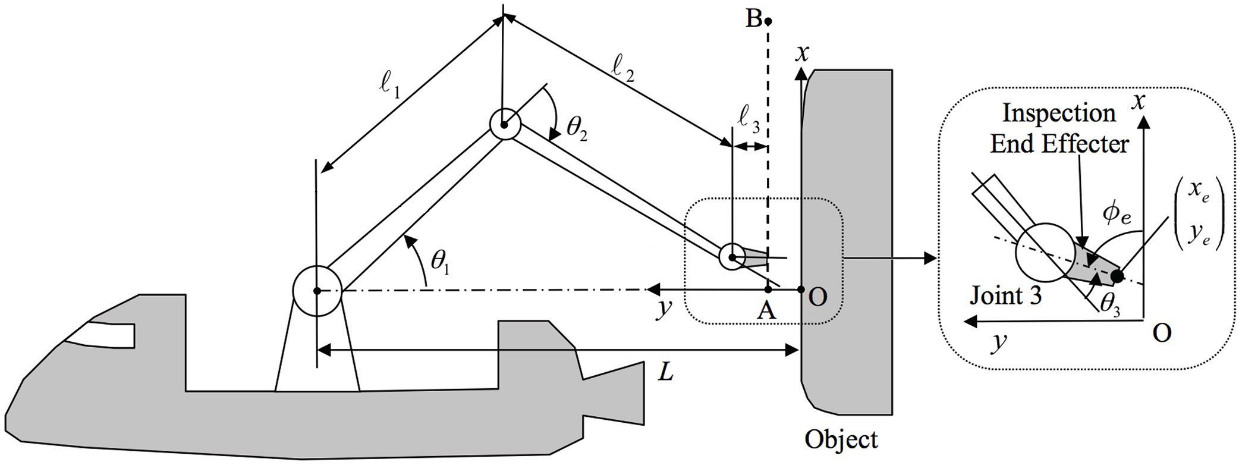

Figure 2: Shuttle manipulator inspecting an object surface. Note: All the angles are measured in the right hand sense, i.e., θ_2 as depicted in the figure is negative.

a. Obtain the forward kinematic equations relating the end effector position and orientation (x_e, y_e, ϕ_e) to the three joint angles $(\theta_1, \theta_2, \theta_3)$. Note that we are interested in the end effector position and orientation as viewed from the coordinate system attached to the object $\{O, (x, y)\}$.

b. Obtain the Jacobian matrix associated with the forward kinematics from part a).

c. We want to move the inspection end-effector along the object surface, i.e., the x-axis, at a constant speed, $v_{\text{rel}} = 20$ cm/s. We also want to maintain a constant safety gap between the inspection sensor and the object surface $y_{\text{offset}} = 10$ cm, and a constant horizontal orientation, $\phi = 90°$. Compute the time trajectories of the three joint angles and joint velocities as the end-effector moves from Point A at $(x = 0, y = 10\text{cm})$ to Point B at $(x = 15\text{cm}, y = 10\text{cm})$. The link lengths are $l_1 = 10$m, $l_2 = 10$m, and $l_3 = 40$cm, and the distance to the object is $L = 10.5$m. Plot position and velocity trajectories using for example MATLAB.

(Hint: You can think of using the Jacobian iteratively at many intermediate points).

Problem 3

Axis-Angle Representation for 3D Rotations

One of multiple ways to represent a 3D rotation is by a rotation axis $\hat{n}$ and a rotation magnitude θ. This is an important representation because (due to Chasles' Theorem) any 3D rigid body motion can be generated as a rotation about an axis and a translation along the same axis. Figure 3 shows vector $\vec{x}$ that is to be rotated around the axis defined by the unit vector $\hat{n}$, by θ radians, resulting in $\vec{x}'$.

a. Prove the following extression:

$$\vec{x}' = \hat{n}(\hat{n} \cdot \vec{x}) + (\sin\theta)\hat{n} \times \vec{x} - (\cos\theta)\hat{n} \times (\hat{n} \times \vec{x})$$

known as "Rodrigues's Formula". Note that there are multiple ways to verify that the expression is true, some considerably more involved that others.

b. Using the expression you just proved, show that we can write the rotation matrix associated with a rotation about an axis $\hat{n} = (n_1, n_2, n_3)$ by angle θ radians as:

$$R = \begin{bmatrix} n_1n_1(1-c\theta)+c\theta & n_1n_2(1-c\theta)-n_3s\theta & n_1n_3(1-c\theta)+n_2s\theta \\ n_1n_2(1-c\theta)+n_3s\theta & n_2n_2(1-c\theta)+c\theta & n_2n_3(1-c\theta)-n_1s\theta \\ n_1n_3(1-c\theta)-n_2s\theta & n_3n_2(1-c\theta)+n_1s\theta & n_3n_3(1-c\theta)+c\theta \end{bmatrix}$$

where $c\theta = \cos\theta$ and $s\theta = \sin\theta$.

(Hint: Consider the cross product matrix N, which satisfies $Nx = \vec{n} \times x$)

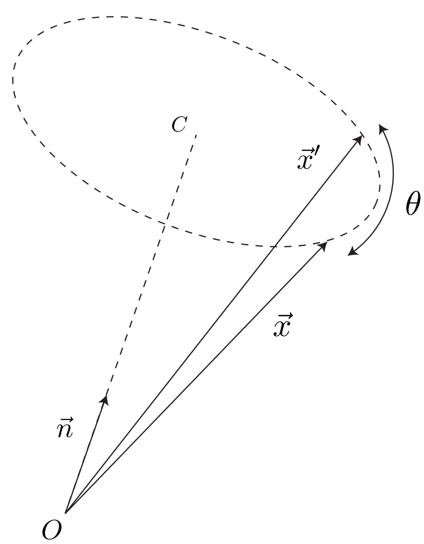

Figure 3: Vector $\vec{x}$ rotates about axis along unit vector $\hat{n}$ by angle θ generating vector $\vec{x}'$.

Problem 4

Iterative Inverse Kinematics [2.120 Only]

Solving algebraically an inverse kinematics problem can be hard, particularly for a high DOF spatial mechanism. Figure 4 illustrates the Space Shuttle problem where the end effector must trace the surface of an object from point A to point B in 3D space. We want to find an algorithm to compute joint trajectories without having to solve the inverse kinematics problem.

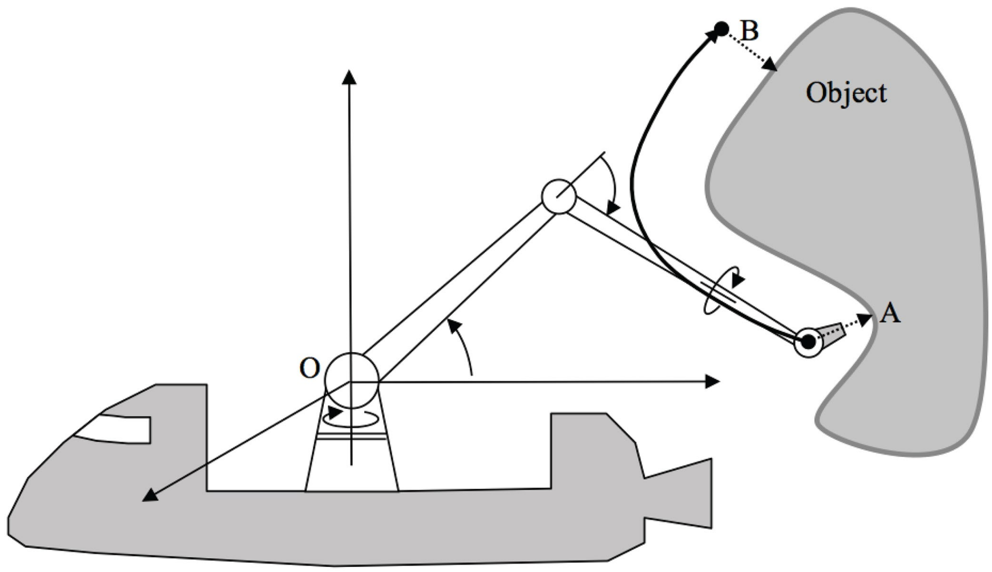

Figure 4: Space Shuttle manipulator tracking a curved surface.

For the general n degree-of-freedom case, the question can be stated as: Let $\vec{p}$ be the n-dimensional vector of end effector position and orientation, and $\vec{q}$ the m-dimensional vector of joint displacements where n could be different from m. Given:

- the forward kinematic equation: $\vec{p} = f(\vec{q})$, where $f(\cdot)$ is an n-dimensional differentiable function,
- an m-dimensional desired trajectory $\vec{p}(t)$ of the end-effector from point A to point B,
- and joint coordinates corresponding to the initial pose of the end-effector at point A.

Propose an algorithm to calculate the joint trajectories to track, as closely as possible, the desired end-effector trajectory $p(t)$. Note that there is no one single answer to this question and credit is given for reasonable approaches. An implementation is not required, pseudo-code is sufficient.