Assignment II
Robotics for IEM

October 2023

1 General Notice

1. Important Dates: This assignments starts on **28/09/2023** and ends at **12/10/2023**. This means you should hand in your assignment on *BRIGHTSPACE* no later than 23:59 at **12/10/2023**.

2. You need to hand in a mini-report (made in LaTeX) along with code files. The mini-report has to be brief, clear and easy to read. **We cannot grade what we cannot read.** In this mini-report, you only need to write down the answers to the questions below. In particular, you should write down how you get your answers (e.g., the computation process) rather than only the final results. You can include figures of your drawings in the mini-report, for instance for drawing frame assignments etc.

   Code template files will be provided. These template files provide frameworks and you only need to fill in what is required. DO NOT change the frameworks of the codes (e.g., the format of the inputs of a function).

3. Your mini-report will be graded by the TAs, but the code files will be graded automatically by Matlab scripts. This means that it is your responsibility to make sure that your codes run successfully without reporting any errors. **Unrunable (erroneous) codes and incorrect answers will be graded ZERO by the automatic scripts.** Incorrect answers are those of which the differences from the correct answers exceed some manually set threshold.

4. To avoid version conflicts, you are encouraged to use **MATLAB (2022 or higher)** in UWP, provided by the university, although other versions should work. The recommended robotics toolbox by Peter Corke is **RTB10.4.** A manual named RoboticsToolboxManual about the toolbox is available on *BRIGHTSPACE*.

2 Hand in Materials for this Assignment

Each student has to hand in a zipped folder on the corresponding assignment page on *BRIGHTSPACE*. The name of the this zipped folder should be according to the syntax *SNumber_Assignmenty.zip*. For example, if S123456 is submitting the first assignment, then the zipped folder should be named *S123456_Assignment1.zip*. The contents of the zipped folder is listed below.

- PDF file of the mini-report made in LaTeXwith the answers (and derivations) to the questions in Section 3 of this document.
- MATLAB Code files necessary for this specific assignment, being `ik.m` and `jacob.m`

3 Questions

Question 1 - Inverse Kinematics (60%)

Throughout this assignment, we will use the lightweight, low-cost robotic manipulator AX-18A SMART ROBOTIC ARM (see Figure 1). The following questions might require the use of the Robotics Toolbox developed by Peter Corke, which you can download here. The needed functionalities of the Toolbox are described in the manual downloadable from BrightSpace.

Figure 1: AX-18A robotic manipulator

1. [35%] Assuming $\theta_4 = 0, \theta_5 = \frac{\pi}{2}$, derive the inverse kinematics for the robot shown in Figure 1, of which the schematics are given in Figure 2. You can use geometrical and analytical approaches. ${}^0p_6 = [P_{E,x}, P_{E,y}, P_{E,z}]^\top$ equals the position of the end-effector expressed in frame zero and is given in equation (2). The links' parameters are given by: $L_1 = 17$ cm, $L_2 = 17$ cm, $L_3 = 7$ cm, $L_4 = 4$ cm, $L_5 = 4$ cm, and $L_6 = 9$ cm. For this question, you should compute the function $f : \mathbb{R}^3 \mapsto \mathbb{R}^3$ such that

$$\begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} = f\left(\begin{bmatrix} P_{E,x} \\ P_{E,y} \\ P_{E,z} \end{bmatrix}\right). \tag{1}$$

   It is **not** allowed to use the robotics Toolbox for this question, only geometrical and analytical methods are allowed. (Hint 1: Make use of trigonometric identities to simplify ${}^0p_6$). (Hint 2: You can check your answer using a forward kinematics function.)

2. [10%] Now consider θ_4 and θ_5 to be zero. The total number of solutions for the inverse kinematics equals $2 \times 2 = 4$. Describe (or draw) the corresponding poses for the different solutions and put them in the mini-report. (Hint: Consider the positions of the first and elbow joint.)

3. [15%] Fill in the incomplete Matlab function `ik.m` with the corresponding inverse kinematics solution you found in subquestion (1). This function takes as inputs the three coordinates of the position of the end-effector $(P_{E,x}, P_{E,y}, P_{E,z})$. The output of the function are the joint angles $[\theta_1, \theta_2, \theta_3, \theta_4, \theta_5]$ (radians). Do **not** use the robotics toolbox for this question. If you where not able to find a solution in Q1.1 program the solution for $\theta_4 = 0, \theta_5 = 0$. This solution is given below.

For $\theta_5, \theta_4 = 0$ we find.

$$^{0}p_6 = \begin{bmatrix} P_{E,x} \\ P_{E,y} \\ P_{E,z} \end{bmatrix} = \begin{bmatrix} -c_1(4c_{2+3} + 20s_{2+3} + 17s_2) \\ -s_1(4c_{2+3} + 20s_{2+3} + 17s_2) \\ 17c_2 + 20c_{2+3} - 4s_{2+3} + 17 \end{bmatrix}$$

(a) Finding θ_1

$$\theta_1 = Atan2(y, x)$$

Or use the second solution

$$\theta_1 = Atan2(y, x) + \pi$$

(b) Finding θ_3 by law of cosines

$$\cos(\theta_3 + \beta) = \frac{x^2 + y^2 + (z - 17^2) - L_2^2 - L_4^2 - (L_3 + L_5 + L_6)^2}{2L_2\sqrt{L_4^2 + (L_3 + L_5 + L_6)^2}}$$

Where

$$\beta = Atan2(L_4, L_3 + L_5 + L_6)$$

And

$$\sin(\theta_3 - \beta) = \pm\sqrt{1 - \cos(\theta_3 + \beta)^2}$$

To find θ_3 compute Atan2 (elbow up and down)

$$\theta_3 = Atan2(\pm\sqrt{1 - \cos(\theta_3 + \beta)^2}, \cos(\theta_3 - \beta)) - \beta$$

(c) finding θ_2

$$\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} k_1 & k_2 \\ k_1 & -k_2 \end{bmatrix} \begin{bmatrix} c_2 \\ s_2 \end{bmatrix}$$

Use the following values.

$$k_1 = 4c_3 + 20s_3$$
$$k_2 = 20c_3 - 4s_3 + 17$$
$$b_1 = \pm\sqrt{x^2 + y^2} \quad \text{use + for first solution of } \theta_1 \text{ and visa versa}$$
$$b_2 = z - 17$$

Now we can extract $\theta_2 = Atan2(s_2, c_2)$ by inversion of the matrix

Question 2 - Jacobians (40%)

1. [25%] Consider the robotic arm in Figure 1, the AX-18A manipulator. Consider the position of the end-effector expressed in frame zero as stated below.

$$ {}^{0}p_6 = \begin{bmatrix} P_{E,x} \\ P_{E,y} \\ P_{E,z} \end{bmatrix} = \begin{bmatrix} 9s_1c_4s_5 + 9c_1c_{2+3}s_4s_5 - 9c_1s_{2+3}c_5 - c_1(4c_{2+3} + 11s_{2+3} + 17s_2) \\ -9c_1c_4s_5 + 9s_1c_{2+3}s_4s_5 - 9s_1s_{2+3}c_5 - s_1(4c_{2+3} + 11s_{2+3} + 17s_2) \\ -9s_{2+3}s_4s_5 + 9c_{2+3}c_5 + 17c_2 + 11c_{2+3} - 4s_{2+3} + 17 \end{bmatrix} \tag{2} $$

   Derive the basic Jacobian $J_{0,v} \in \mathbb{R}^{3\times5}$ relating the joint velocities to the end-effector's linear velocities. Do **not** use the robotics toolbox, only analytical methods are allowed (show derivations).

2. [15%] Fill in the provided incomplete Matlab function `jacobian.m`. This function takes as inputs the five joints angles $(\theta_1, ..., \theta_5)$ (unit:radians) and outputs the basic jacobian associated to the linear velocities, i.e., $J_{0,v} \in \mathbb{R}^{3\times5}$, evaluated for that particular joint configuration. The links' parameters are the same as the previous question.

Bonus Question (10 %)

Consider a robotic arm with sufficient accurate encoders for the joints positions. Also, this robot can take joint-velocity as input to move. We want to move the robotic arm from the current position to a target position. How can the Jacobian be used to compute the inverse kinematics towards the target position? Describe the steps and possible problems, assuming the followings are known

- Current joint positions (encoders)
- Current end-effector position and orientation
- Target position and orientation of the end-effector
- Jacobian matrix as function of the joint values read by the encoders.

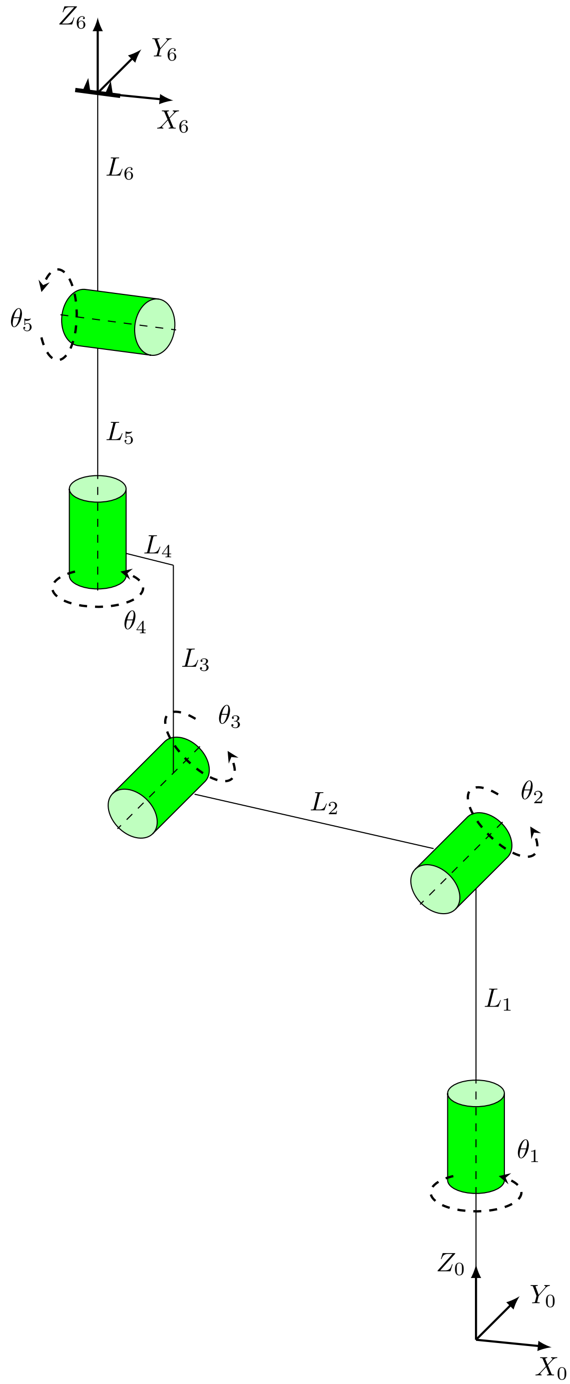

Figure 2: AX18A schematic, the following offsets exists from the zero configuration: $\theta_2 = \frac{\pi}{2}, \theta_3 = -\frac{\pi}{2}$