MEEG671: Introduction to Robotics

Final Project

A 7 degree-of-freedom robot arm (LBR iiwa 7 R800, KUKA) (Fig. 1) is mounted on a base as seen in Fig. 2. An *adapter* is mounted on the robot flange. The *adapter* holds a camera as shown in Fig. 3 on one side, and a rectangular shape on its other side.

The goal of this project is to program a trajectory for the robot to position the *rectangular shape* on top of a rectangular *target* located in the workspace of the robot arm (indicative solution is shown in Fig. 4). The exact location and orientation of the *target* can be computed using an Aruco[1] marker, that is printed close to the target, and is seen by the camera (see Fig. 5).

The robot starts from configuration $\mathbf{q_1}$ and needs to position the *rectangular shape* as close as possible to the target, matching its corners A, B, C, D with those of the target. During the motion, the robot should not hit in any way the surface/table the target is put on.

Design the trajectory in joint space so that the robot arm moves from its starting configuration and finishes when the *rectangular shape* is on top of the target.

In order to locate the target position, the camera takes a picture of the target and the Aruco marker. The software returns the position and orientation of the Aruco marker reference system with respect to the camera reference system, which is shown in Fig. 5.

When creating the robot trajectory, ensure you don't exceed the angular position and angular velocity limits of all the robot joints. The entire robot motion should **last exactly 10 seconds, in which, for the last 2 seconds, the robot should hold the same configuration, i.e., the robot should not be moving for the last 2 seconds**.

Deliverables:

- Provide the Matlab Code and a Technical Report (max. 5 pages) of the methods used to design the desired robot joint trajectory. All files should be included in a LastName_FirstName.zip file.

- Provide a LastName_FirstName.txt file that will have n rows and 7 columns of data. Each row will correspond to the robot configuration **in radians** from joints 1 to 7. Use 8 decimal points for the joint angles and a single space between the values for two joints. Each of the n rows corresponds to a robot configuration commanded to the robot every 5 ms. Therefore, you need to check if the generated trajectory exceeds the velocity limits when commanded every 5ms. The first row of your file should be the initial robot configuration $\mathbf{q_1}$. No other characters should be included in the .txt file. A sample file is attached (sample.txt). Since the robot motion should last 10 seconds, and each row corresponds to 5ms, then the file should have $n = 2,000$ rows.

- Each student will present his/her work in a 5-minute meeting with the instructor. All the deliverables (code, report, trajectory file) should be sent to the instructor before the presentation at a date that will be announced. You should be able to show/prove that the robot executes the motion task along the guidelines, and the robot does not hit the table while staying within position and velocity limits.

[1] https://www.pyimagesearch.com/2020/12/21/detecting-aruco-markers-with-opencv-and-python/

The manual of the robot is attached. You can use any information you need from the manual. Make sure you use the technical details of the LBR iiwa 7 R800 robot and not the LBR iiwa 14 R820, which is also included in the manual.

Additional Information:

The angular configuration of the joints of the robot in the beginning is given by:
$\mathbf{q_1} = \begin{bmatrix} 58.2686 & 75.3224 & 11.7968 & 45.9029 & -22.1081 & -31.2831 & -42.3712 \end{bmatrix} \quad \text{(deg)}$

The Aruco marker software provides the position and orientation of the Aruco marker reference system with respect to the camera reference system $< X_c, Y_c, Z_c >$ as follows:
transform translation x: -0.205780720039398 (m)
transform translation y: -0.109793029482687 (m)
transform translation z: 0.561252115509121 (m)
roll x: 174.1750404305652 (deg)
pitch y: -17.3967534123935 (deg)
yaw z: -1.9587388578232 (deg)

Note that the above orientation angles follow the Euler ZYX notation (Hint: Use Matlab function eul2rotm([$rot\_ang\_in\_rad$],'ZYX') to convert them to a rotation matrix).

The configuration of the robot at the instance the picture is captured is given by:
$\mathbf{q_c} = \begin{bmatrix} 97.31 & -4.70 & 162.57 & 103.53 & -1.96 & -62.33 & 113.31 \end{bmatrix} \quad \text{(deg)}$

Grading Criteria:

Your grade will be out of 10, and the following points will be deducted before the meeting/presentation with the instructor:

- 1 point if any of the instructions above (see deliverables) are not followed, e.g. wrong zip filename, wrong format of the LastName_FirstName.txt file with the robot motion, missing functions, etc. [Following instructions criterion]
- 1 point if the robot does not start from the initial point or fails to bring the object close to the target. [Inv. Kinematics criterion]
- 1 point if the orientation of the robot at the end of its motion is not such as to bring the object on top of the target. [Orientation criterion]
- 1 point if the robot motion exceeds position limits at any joint. [Position range criterion]
- 1 point if the robot motion exceeds velocity limits at any joint. [Velocity range criterion]
- 1 point if the robot hits the surface/table with any point of its body. [Obstacle avoidance criterion]

Your final grade will also be based on the Q&A during your presentation after the initial deductions listed above.

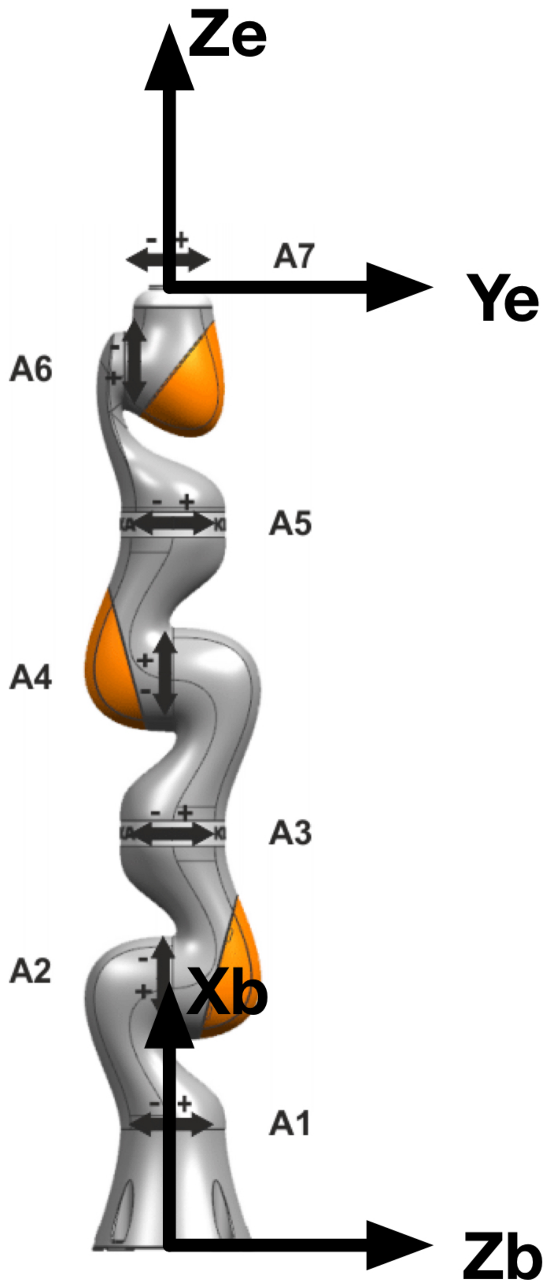

Figure 1: LBR iiwa 7 R800 robot arm, KUKA. Base and end-effector frames are shown.

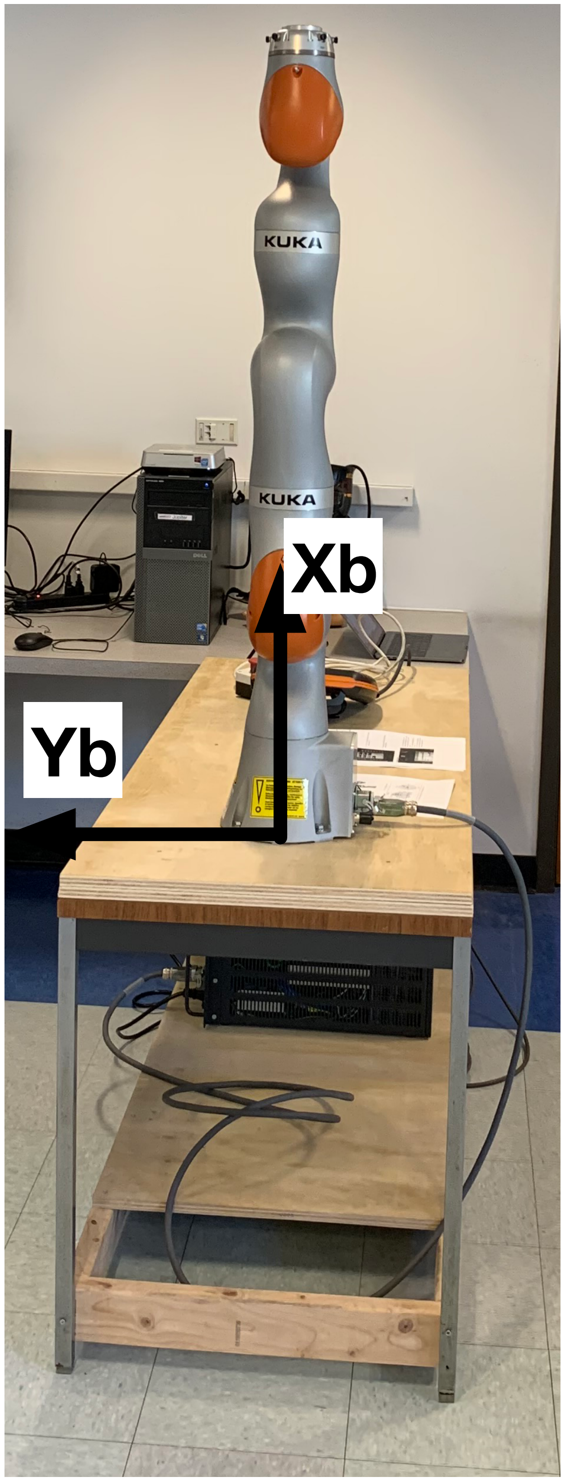

Figure 2: Robot mounting.

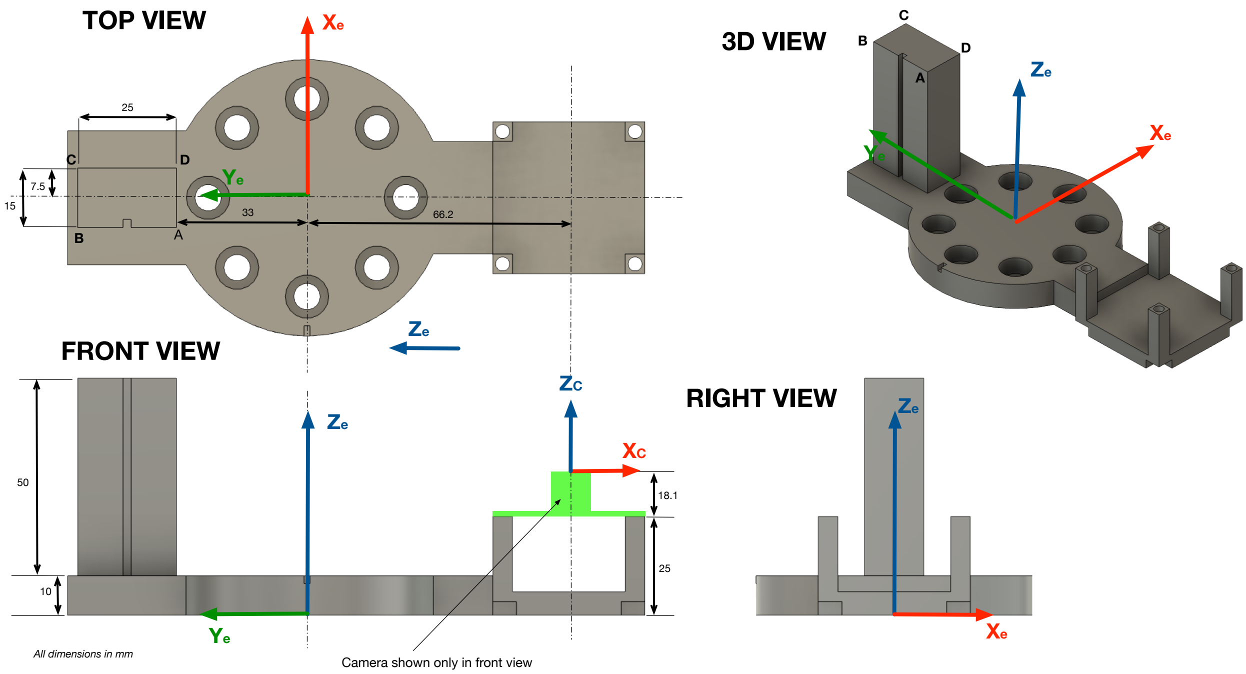

Figure 3: Adapter mounted on the robot end-effector flange. Camera reference system $< X_c, Y_c, Z_c >$ and robot end effector reference system $< X_e, Y_e, Z_e >$ are shown. All dimensions are in mm. Rectangular shape with A, B, C and D corners shown.

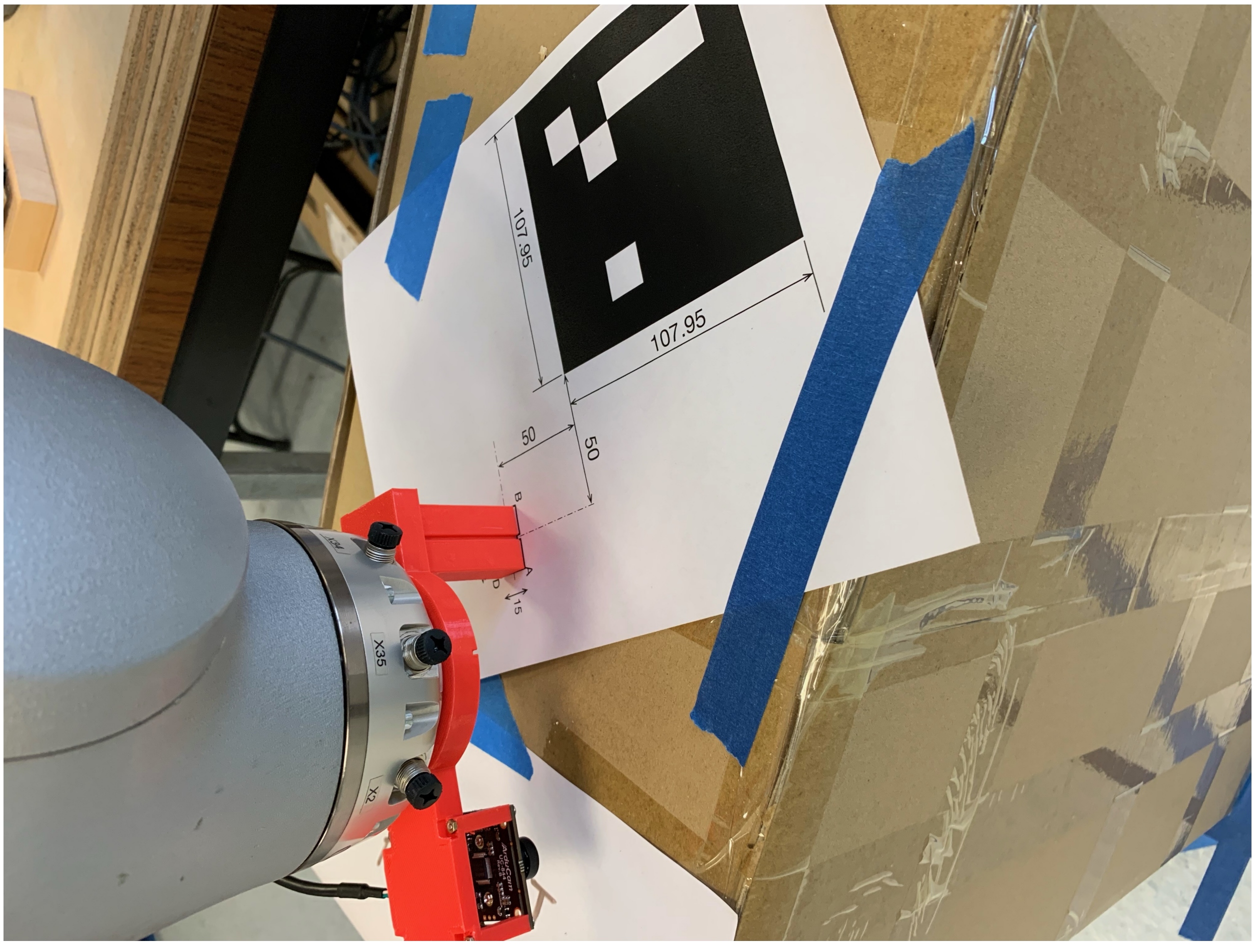

Figure 4: The *rectangular shape* is placed on top of the target.

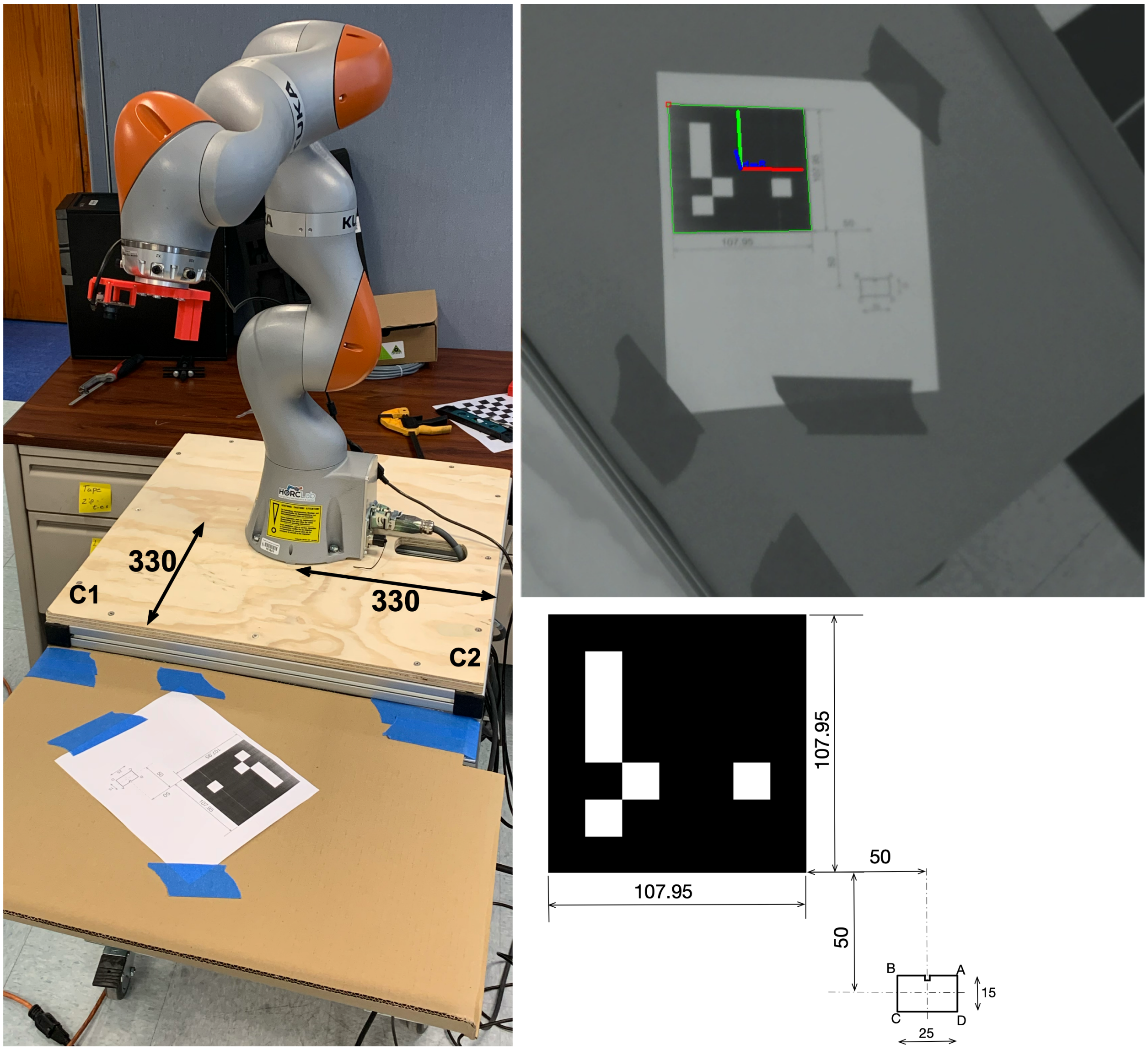

Figure 5: Robot taking a picture of the Aruco marker and target sheet (left), as well as the captured image with Aruco marker reference system identified (right, top). The reference system is placed at the center of the Aruco marker. All dimensions are in mm. The printed sheet with the Aruco marker and target is shown in bottom right. The two corners of the table C1 and C2 are shown along with their distances from the center of the robot base.