Introduction to Robotics: Homework II

Instructor: Konstantinos Chatzilygeroudis (`costashatz@upatras.gr`)

5 November 2025

Introduction

The homework involves determining the transformation matrices and robot joint positions associated with a robotic manipulation task. Specifically, you are required to identify the transformation matrices for all objects in the environment, the relative transformation matrix between the manipulator's end-effector and the object being manipulated, and the transformation matrix for the end-effector that enables the manipulated object to be precisely inserted into a designated hole. Once all the transformation matrices are determined, you are required to calculate using inverse kinematics the joint positions of the manipulator in order to a) grasp the cube, and b) place the cube right next to the target hole (so that moving it perpendicularly toward the box's surface would place it directly into the hole).

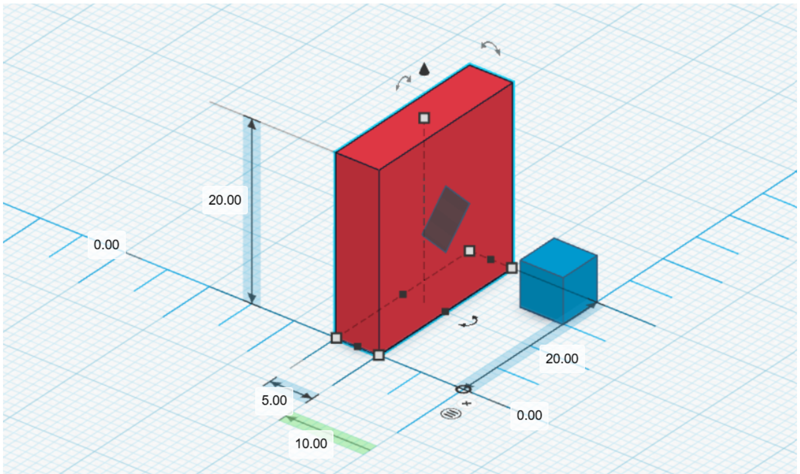

Figure 1: Visualization of the manipulation scenario. **Measurements are in cm.**

1 Description

The setup includes an Elephant myArm 300 Pi manipulator equipped with a gripper (Fig. 2), a box featuring a cubic hole at its center (represented as the red rectangular parallelepiped in Fig. 1), and a cube (shown as the blue cube in Fig. 1) that the manipulator must handle and insert into the hole. The Elephant myArm 300 Pi is a 7-DOF robotic arm that surpasses conventional 6-axis manipulators, offering human-like flexibility and precision. Powered by a Raspberry Pi, it supports ROS, Python, and advanced control, making it ideal for research, teaching, and application development in robotics. The manipulator's reach is shown in Fig. 4. Also, its joint limits are shown in Table 1. **The kinematic structure of the robot is given in Fig. 3. The transformation from the frame (x_7, y_7, z_7) to the end-effector frame (x_{ef}, y_{ef}, z_{ef}) is given by a translation of 75 mm in z_7 axis and 8 mm in x_7 axis (the orientation doesn't change).**

2 Task

Given Fig. 1, Fig. 2, Fig. 3, Fig. 4, Fig. 5, Fig. 6, Fig. 7, Fig. 9, Fig. 8, Fig. 10, Fig. 11 and Table 1:

1. Choose the location of the world frame and assign a fixed body frame to each object;
2. Select an appropriate transformation matrix for the base frame of the manipulator with respect to the world frame, ensuring it can perform the manipulation task;
3. Determine the relative transformation matrix between the manipulator's end-effector and the cube, allowing the manipulator to grasp the cube (aka, what should be the orientation and position of the cube with respect to the end-effector such that it can be grasped);
4. Calculate the cube's world-frame transformation matrix both when it is positioned right next to the hole (so that moving it perpendicularly toward the box's surface would place it directly into the hole) and after it has been precisely positioned inside the box by the manipulator.
5. Using inverse kinematics calculate the joint positions of the arm such that the end-effector is able to grasp the cube (verify your result with forward kinematics);
6. Using inverse kinematics calculate the joint positions of the manipulator such that the cube (grasped by the arm) is positioned in the "placing" position right next to the hole (verify your result with forward kinematics);
7. Visualize the results using the programming language of your choice.

Guidelines:

- You have to use the Product of Exponentials for solving the forward kinematics problem;
- You have to solve the inverse kinematics problem using the Newton-Raphson algorithm;
- In order to solve the inverse kinematics problem you have to be able to compute the Jacobians of the manipulator (compute the geometric Jacobians);
- Your inverse kinematics solver should be generic and applicable to any desired end-effector pose.

3 Deliverables

- Detailed report;
- Code for visualization and containing the transformation matrices.

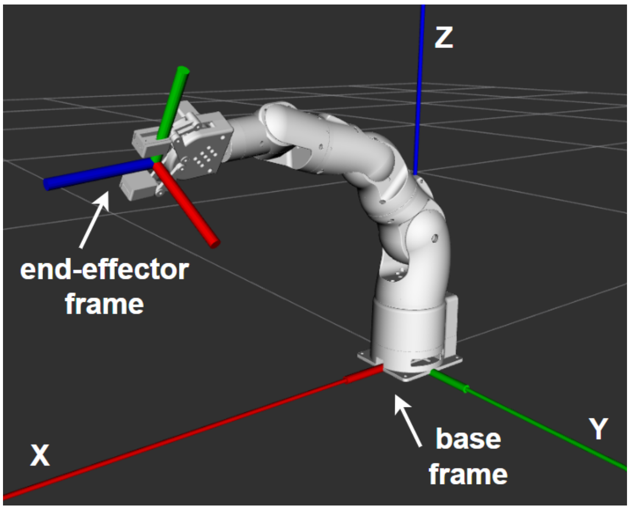

Figure 2: Visualization of the Elephant myArm manipulator.

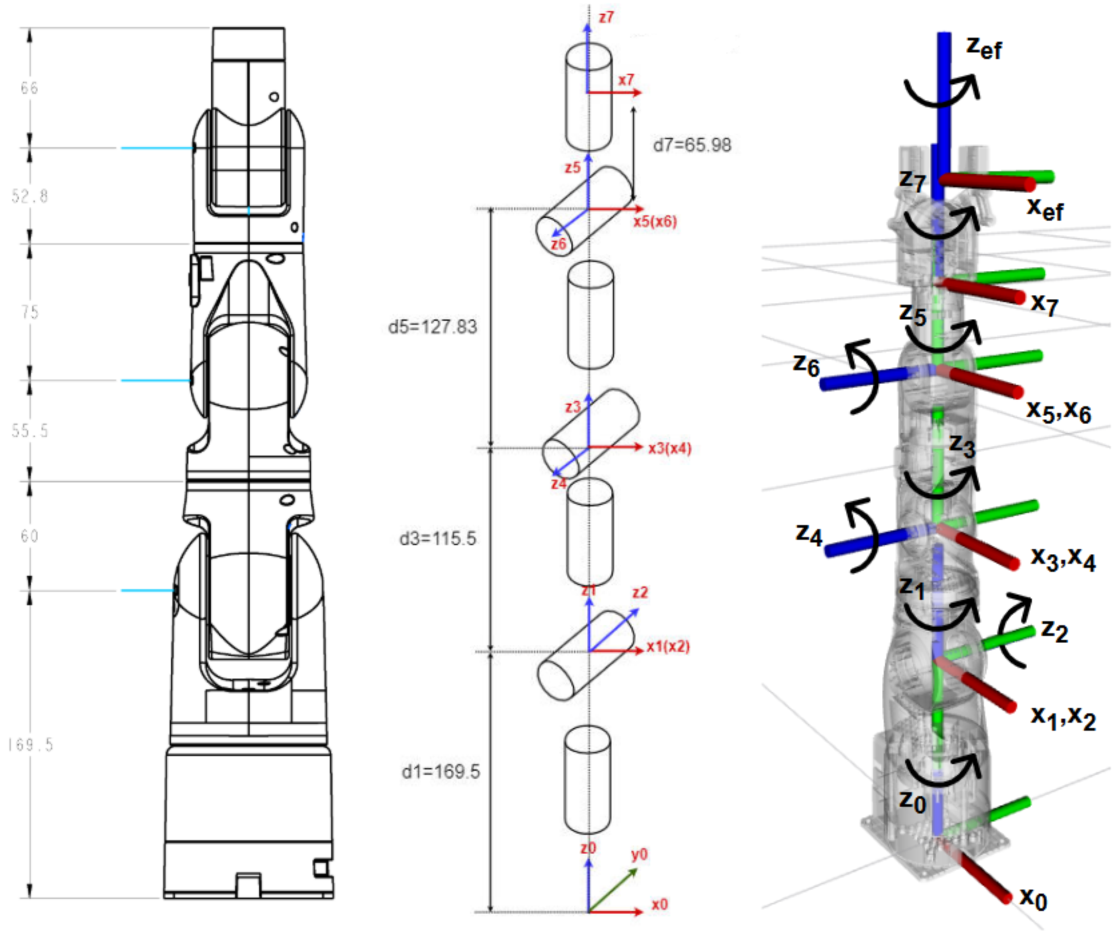

Figure 3: Elephant myArm manipulator kinematic structure (shown in the reference zero configuration). **Measurements are in mm.**

Joints	**Min angle (°)**	**Max angle (°)**
Joint 1	-160	160
Joint 2	-70	115
Joint 3	-170	170
Joint 4	-113	75
Joint 5	-170	170
Joint 6	-115	115
Joint 7	-180	180

Table 1: Joint limits (in degrees) for the Elephant myArm manipulator, where zero joint angles represent the default vertical position.

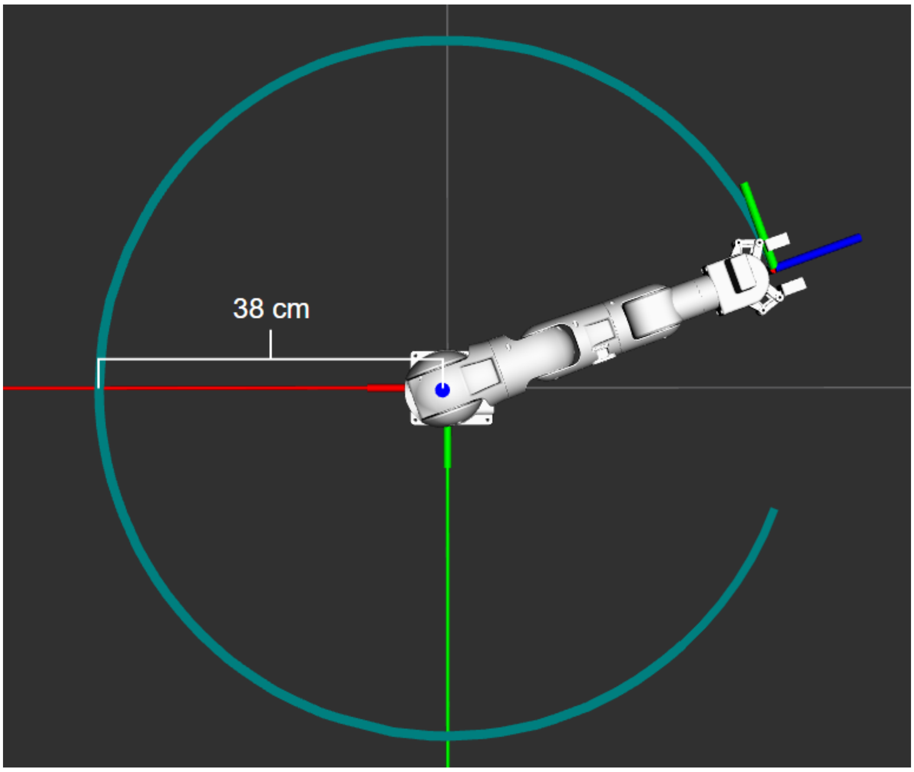

Figure 4: Elephant myArm manipulator's reach (top view).

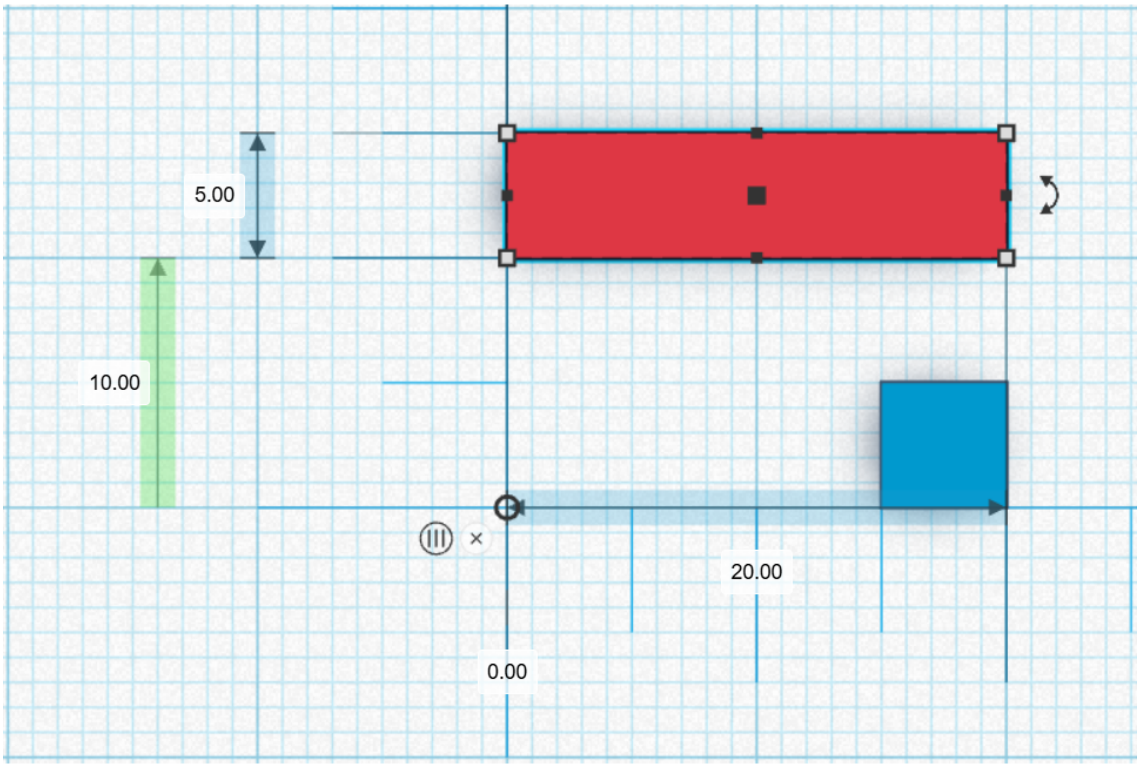

Figure 5: Scenario measurements. **Measurements are in cm.**

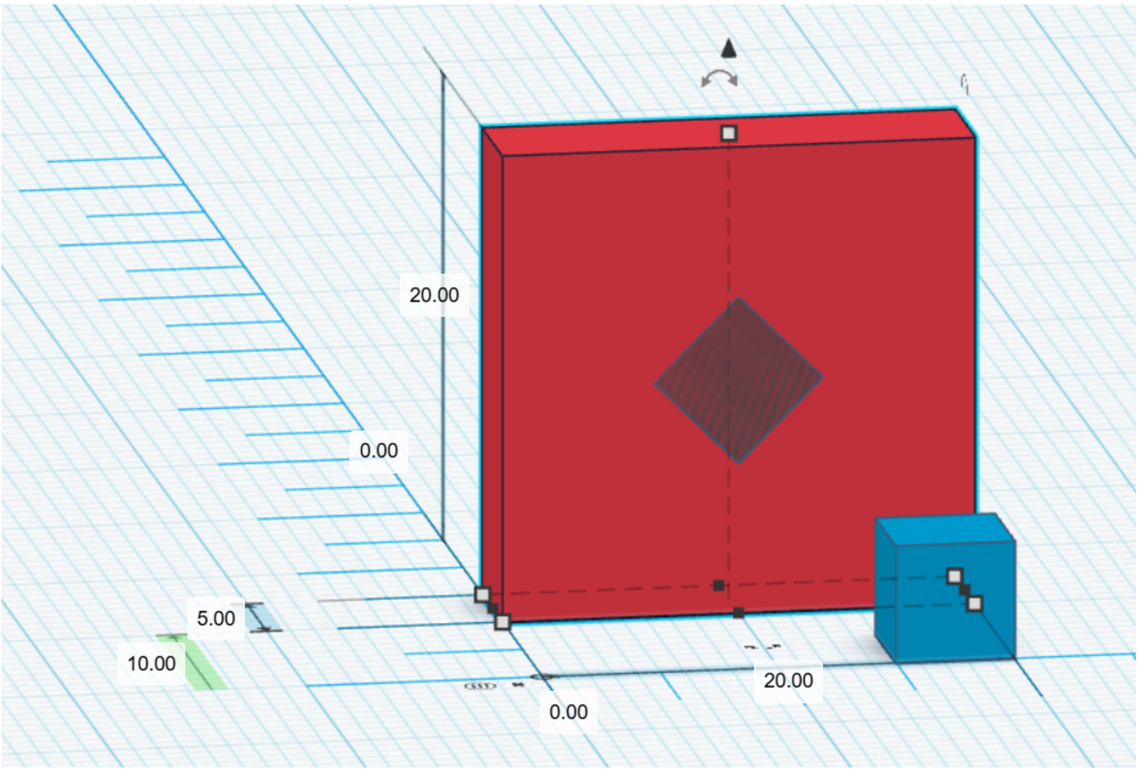

Figure 6: Scenario measurements. **Measurements are in cm.**

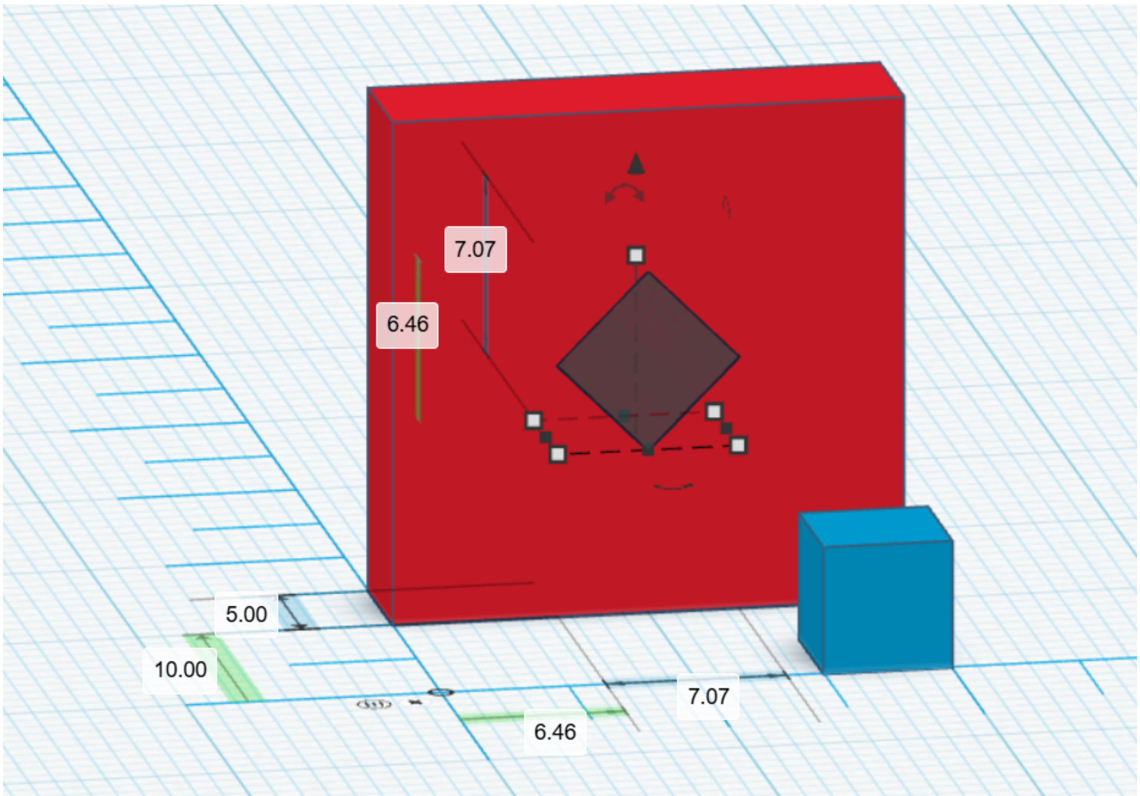

Figure 7: Scenario measurements. **Measurements are in cm.**

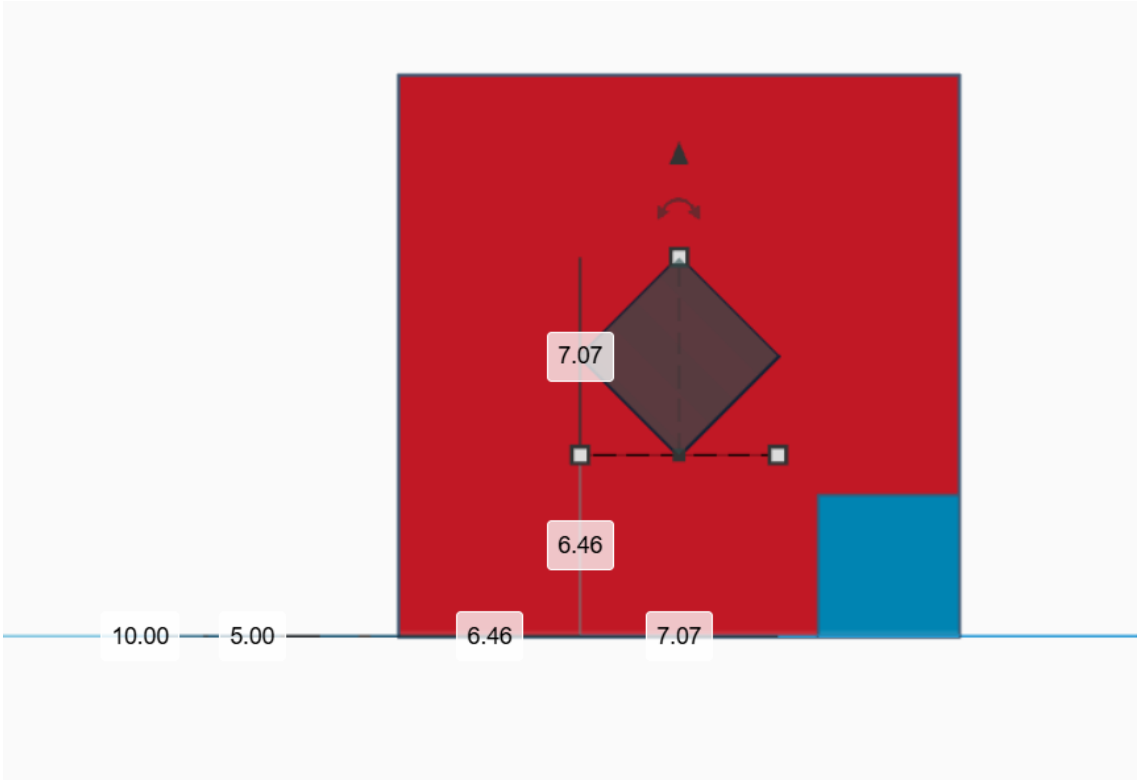

Figure 8: Scenario measurements. **Measurements are in cm.**

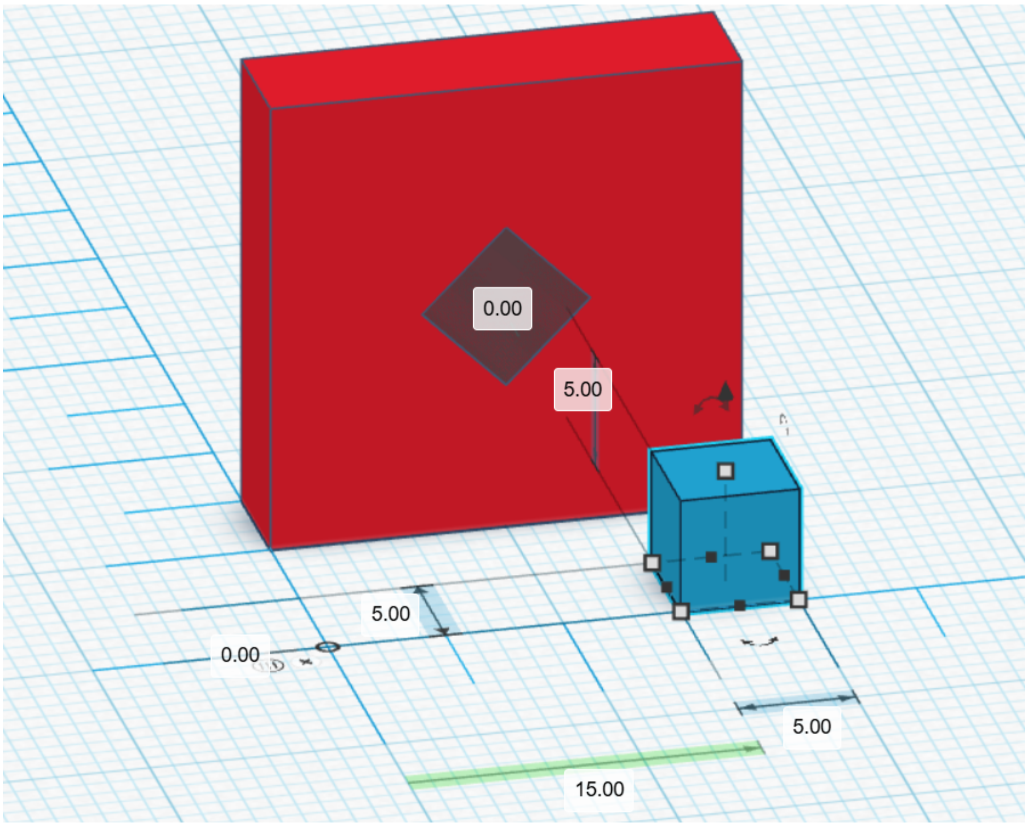

Figure 9: Scenario measurements. **Measurements are in cm.**

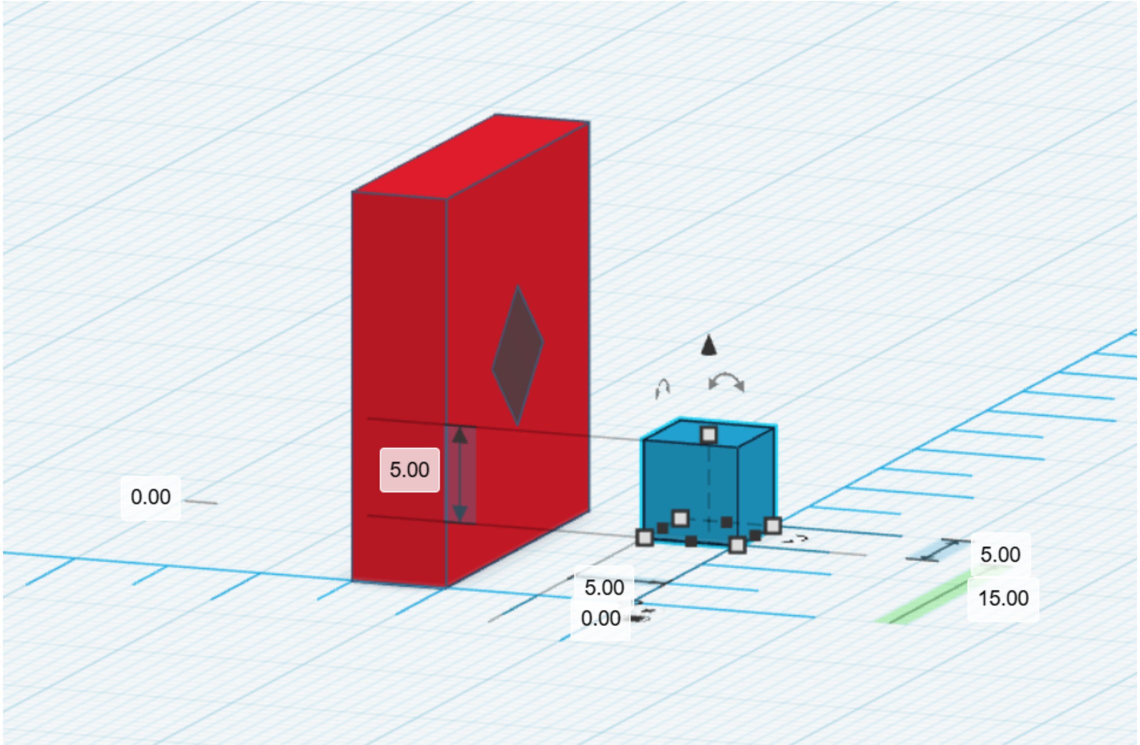

Figure 10: Scenario measurements. **Measurements are in cm.**

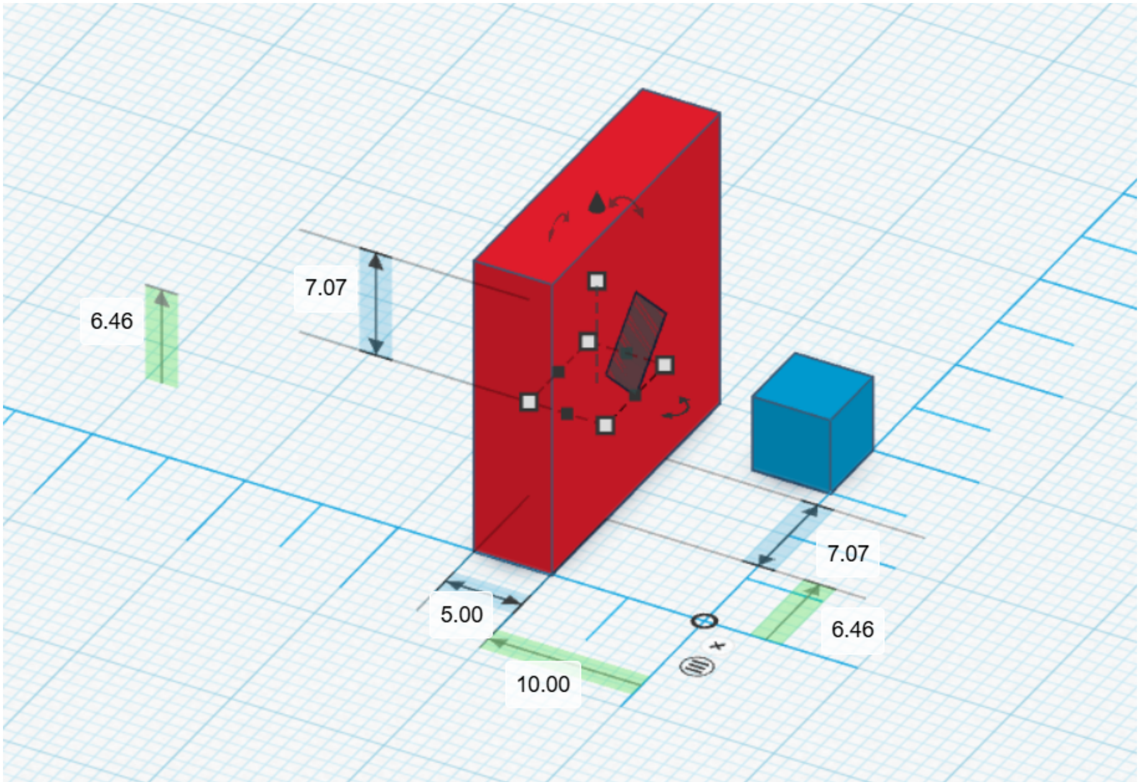

Figure 11: Scenario measurements. **Measurements are in cm.**