Institute for Aerospace Studies
UNIVERSITY OF TORONTO

LABORATORY II

Robot Arm

ROB301 Introduction to Robotics
Fall 2015

1 Introduction

This is the second lab of ROB301 Introduction to Robotics. The focus for this week is a robotic arm. We will investigate robot manipulators at a high-level. We shall be addressing the kinematics and dynamics of robotic arms later in the course. More in-depth analyses will be available in ECE470 or AER525. Lab II has no written deliverables; however, students must attend the session and demonstrate functionality to the TAs.

2 Purpose

This lab introduces several fundamental concepts and issues that frequently arise with the use of robotic manipulators. But the main purpose is to allow students to gain greater familiar with the LEGO robot kits. Only open-loop control is expected with some opportunity for data collection in the later tasks.

3 Equipment & Software

The hardware platform used for ROB301 are LEGO Mindstorms EV3 kits. Included on each workstation is a folder labelled “Lab 2” which contains a copy of these lab instructions as well as a PDF of the step-by-step LEGO assembly instructions for a robotic arm. This apparatus must be constructed by at least one member of the team in order to complete the tasks of the day. Approximate build time is 45–60 minutes. Please note that the Touch sensor and Color sensor included in the PDF are not necessary for the following tasks and may be excluded from the build or included and ignored.

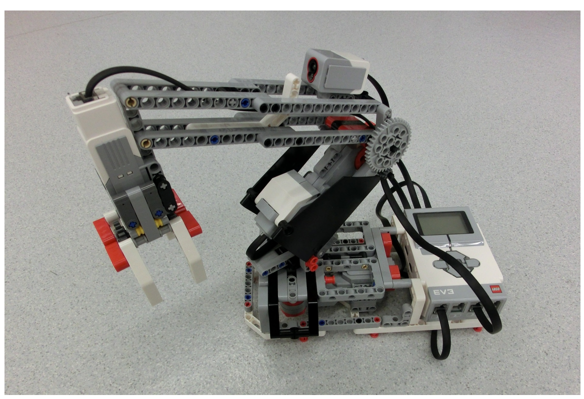


Figure 1: LEGO robot arm

The robotic arm consists of three actuators, two large servo motors for positioning and a medium servo motor to open and close a gripper. As there are only two actuators dedicated to driving the arm kinematics, the robotic arm’s capabilities and degrees of freedom are limited.

4 Task Programming

Four different areas relevant to robotic manipulators are investigated in this section. First, the capabilities of the robotic arm are detailed as the workspace is defined. Second, the motion of the manipulator through space is done by specifying simple and more complex trajectories. Third, the gripper is used to allow the robotic arm to interact with its environment by picking up and moving objects. Finally, the robot arm is used as a tool to observe its surroundings.

4.1 Workspace Identification

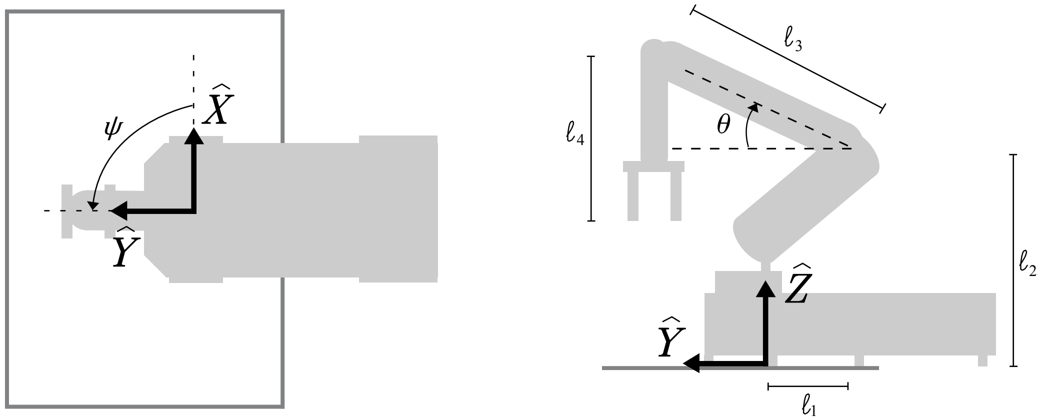
The workspace of a robotic manipulator is defined as the area or volume in which the end-effector can reach. Different applications warrant manipulators with different capabilities. Complex manipulators which have many joints (and degrees of freedom) can interact with objects in their environment is complex ways. There exist conventions and formulations to mathematically characterize the kinematics of these robots which will not be addressed in this lab. Fortunately, our robot arm is able to move in three dimensional space while still being simple enough to characterize mathematically by inspection and without the need for a rigorous modelling framework.

The robot arm constructed for this lab uses two large servo motor actuators to position the end-effector. We will characterize the workspace of our robotic arm. To do this, we need to describe mathematically the position and orientation of the end-effector. We start by affixing axes to the robot in order to base our investigation. The $\hat{Z}$ axis passes through the main revolute joint responsible for "yaw" or moving side-to-side. The angle ψ is measured about the $\hat{Z}$ axis. The plane formed by the work-surface is the location of the X-Y plane where the $\hat{X}$ axis points out to the side of the robot arm base. This is where $\psi = 0$ as seen in Figure 2. The $\hat{Y}$ axis is the remaining mutually orthogonal vector whose direction is determined using the right-hand rule. The "pitch" angle θ is the angle the upper part of the arm makes with the horizontal.

Parameter	**Value**	
ℓ_1	7	cm
ℓ_2	16	cm
ℓ_3	19	cm
ℓ_4	11	cm

Table 1: Robot arm parameters

We seek to describe the position of the end-effector tip in Cartesian coordinates relative to the origin as a function of the yaw angle ψ and pitch angle θ. Start by considering only the Y-Z plane (when $\psi = 90°$) and determine the position vector from the origin to the end-effector as a function of θ and the parameters in Table 1. Next, consider movement in



(a) Top view

(b) Front view

Figure 2: Frames and relevant angles for robot arm

three-dimensional space and incorporate the effect of ψ as a variable on the position vector from the origin to the end-effector.

With the LEGO robot arm construction complete, find the motor position signals which define the boundary of the workspace? What motor angles cause the manipulator to reach the limits dictated by ψ and θ?

4.2 Trajectory & Path-Planning

The next robot manipulator application is the movement through the workspace using various trajectories. It is important for manipulators to be able to avoid obstacles as well as have the necessary manoeuvrability while approaching objects for the given task.

Simple Trajectories. Set the manipulator at $\psi = 0$ and on the X-Y plane. Write a program which will use the joint space to move the manipulator up to the maximum θ, then move the manipulator to the maximum ψ. It is easy to program these simple trajectories to utilize a single actuator at a time. Note that the rotational speed can be adjusted as needed.

Less Simple Trajectories. Set the manipulator at $\psi = 0$ and on the X-Y plane. Write a program which will move the manipulator up to the maximum θ and maximum ψ simultaneously. These larger, sweeping motions can be used to translate from one point to another directly.

Complex Trajectories. (*If time allows.*) Write a program which will cause the manipulator to trace a circle on the surface of the workspace.

4.3 Pick & Place

A common robot manipulator application is the interaction of the end-effector with the environment. By default, the gripper has been installed as the manipulator end-effector. This gripper will be used to pick up and move blocks. Position the robot arm onto the Pick & Place Mat according to the stand markers (Figure 3).

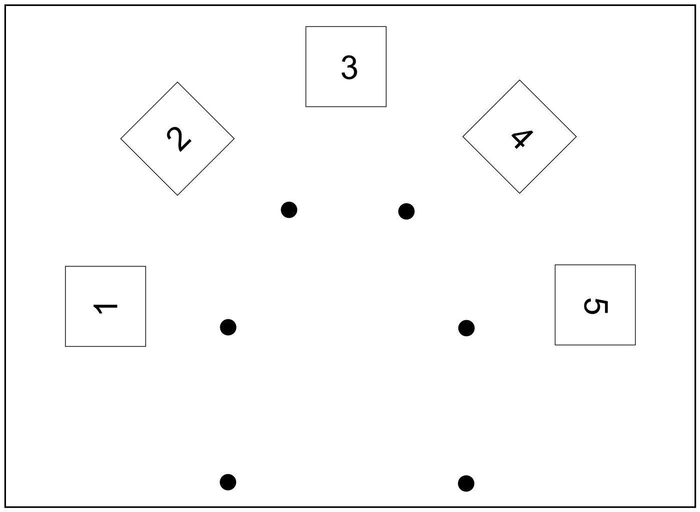


Figure 3: The Pick & Place Mat

Moving Blocks. Using the Pick & Place Mat, position a block onto Position 2 and Position 4. Set the manipulator at $\psi = 0$ and on the X-Y plane. Write a program to retrieve the block at Position 4 and move it to Position 1, then pick up the block at Position 2 and move it to Position 4, then pick up the block at Position 1 and move it to Position 5.

Stacking Blocks. Using the Pick & Place Mat, position a block onto Position 1 and Position 2. Set the manipulator at $\psi = 0$ and on the X-Y plane. Write a program to retrieve the block at Position 1 and move it to Position 3, then pick up the block at Position 2 and move it on top of the block already at Position 3. Have the manipulator return to Position 5 empty-handed. Then have the manipulator retrieve the top block from Position 3 and move it to Position 4, then pick up the lower block at Position 3 and move it on top of the block already at Position 4. Have the manipulator return to Position 5 empty-handed.

4.4 Object Detection

Finally, robot manipulators can be used for inspection purposes in addition to moving objects within the surrounding environment. Remove the gripper apparatus from the upper arm and replace it with a Touch sensor with the button pointing downward. As we learned performing the workspace identification, the end-effector for this manipulator will always point downward. Continue to use the Pick & Place Mat (Figure 3).

Write a program to use the end-effector as a sensor to detect whether an object is present or not at each of the five Positions. After each Position is checked, the display should report whether an object was present or not at each of the five Positions.

5 Conclusion

We have introduced some fundamental robotic arm concepts, experimented with simple applications and learned a little more about the LEGO kit.

6 Additional Resources

1. LEGO Mindstorms - `http://mindstorms.lego.com/`
2. Java Programming - `http://www.javaprogrammingforums.com/`
3. leJOS - `http://sourceforge.net/p/lejos/wiki/Home/`
4. leJOS EV3 API - `http://www.lejos.org/ev3/docs/overview-summary.html`