ECE 494/761 Advanced Controls / Robotics (3cr)

Introduction

Welcome to ECE 494 / 761 Advanced Controls / Robotics. This course covers the mathematics behind modeling and control or robotic arms.

The course syllabus along with lecture notes (i.e. the textbook), recorded lectures, and homework sets are posted on Bison Academy.

ECE 494/761: Robotics

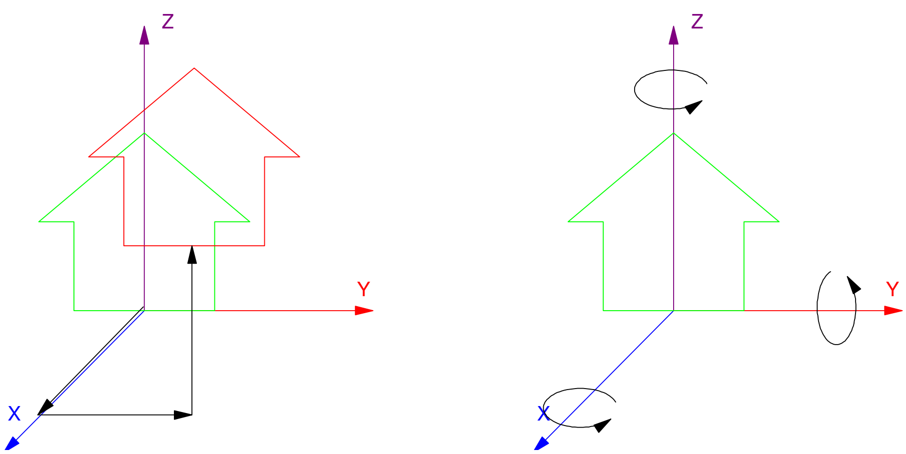
Class	Topic	Video YouTube Playlist	Code	Homework
1	0 Syllabus & Course Overview Slides #0	0: Syllabus Course Overview		
2	1 Matlab Review Slides #1	1: Matlab Review	Bouncing Ball Shoot	No homework #1
3	2 Rotation Matrices Slides #2	2: Rotation Matrices	Display3D Rotate	HW #2
4	3 Translation Matrices Slides #3	3: Translation Matrices	Translate	HW #3

Syllabus from www.BisonAcademy.com/ECE761/Index.htm

Topics that we cover are as follows:

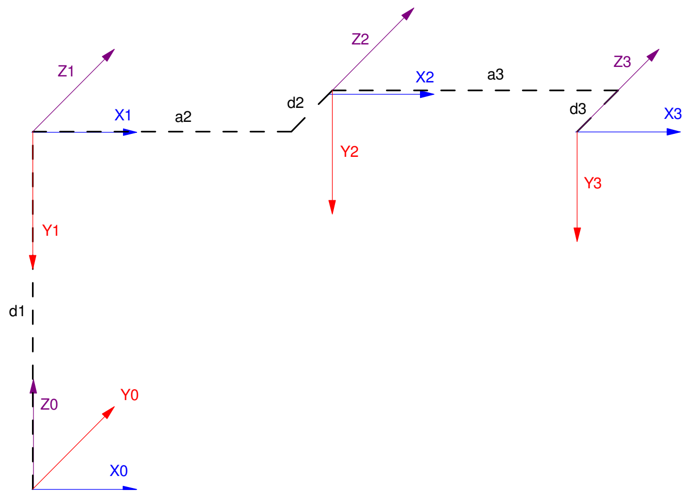
Topics & Course Overview:

Rotation & Translation Matrices: This course starts out covering rotation translation matrices. These matrices are 4x4 matrices that allow you to define where a point is in 3-space when you move along the XYZ axis (translation) or rotate about the XYZ axis (rotation).



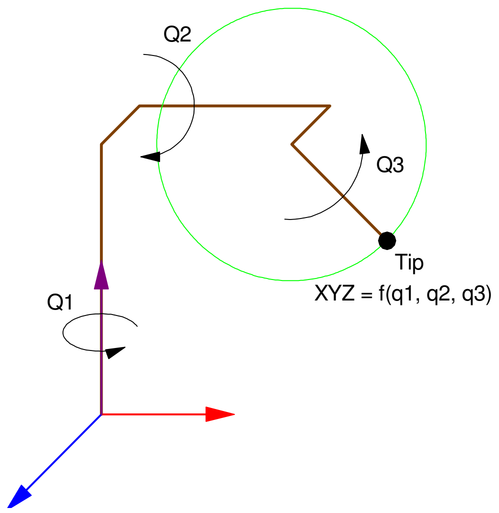
Translation and Rotation Matrices move an object or rotate it

Reference Frames: With a robotic arms, you typically have different joints such as the shoulder, elbow, and wrist. Reference frames define the world (XYZ) relative to a given joint. As you travel from one joint to the next, your position changes (modeled with a translation matrix) and you rotate (modeled with a rotational matrix).



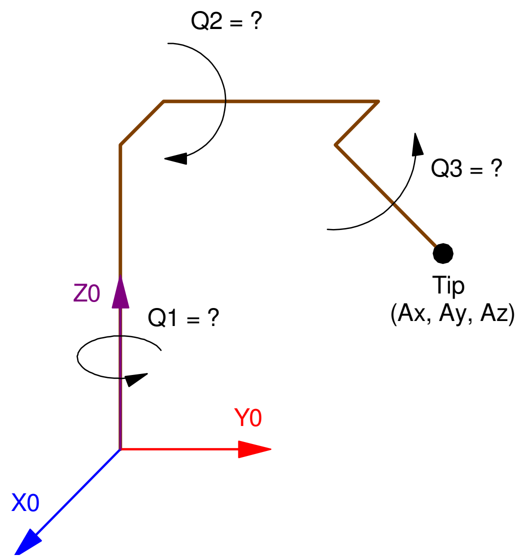
Reference Frames define the joints for a robotic arm

Forward Kinematics: Once you assign reference frames to a robotic manipulator, you can then travel from the base to the tip of the robotic arm. Forward kinematics is where you calculate the tip position given the robot joint angles.



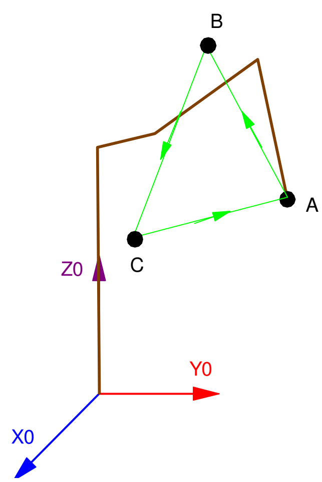
Forward Kinematics: Given the joint angles, determine the tip position

Inverse Kinematics: Typically, the joint angles are what is directly controlled with motors. The tip position is what you care about, however. Inverse Kinematics is where you compute what joint angles put you at a desired tip position XYZ.



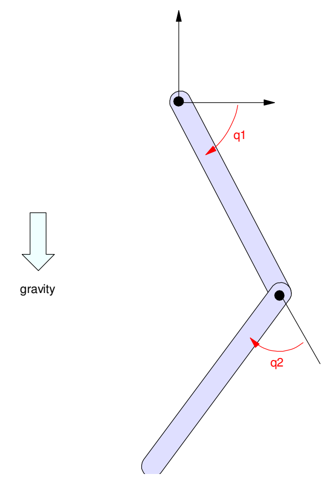
Inverse Kinematics: Given the tip position, determine the joint angles

Path Planning: Once you define how to go to position A with inverse kinematics, the problem arises of how to go from position A to position B. The path you take is called *path planning*. There are several ways to go from point A to B. The first derivative of the path defines the joint velocities. The second derivative defines the joint accelerations. For a path to be reasonable, both the joint velocities and accelerations need to be bounded.



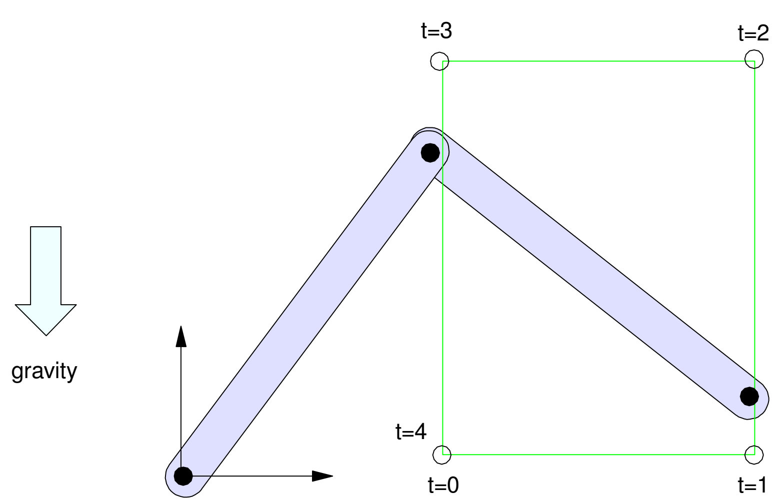
Path Planning: Going from one point to the next while keeping joint velocities and accelerations finite.

Dynamics: Finally, we get to the control of a robotic arm. Here the problem is how to control the voltages to the motors of the robotic arm to follow a desired path. The first step in this process is to define the dynamics of a robotic arm. In this class, we look at a 2-link 2-dimensional robotic arm. More complex models that deal with 3+ links in 3-dimensions are left for future studies.



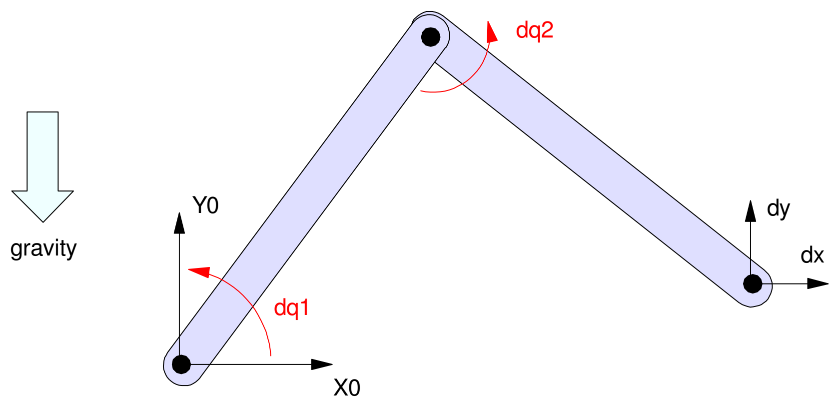
Dynamics: Determine the differential equation that describe the motion of a robotic manipulator

Controls: The dynamics that define the motion of a robotic arm are typically nonlinear and include gravity forces and coriolis forces. Using feedback, you can control the angle of a motor to track a desired path, defined by path planning. This usually results in a lag between the desired angles and the actual angles. By adding feed-forward terms, this lag can be reduced.



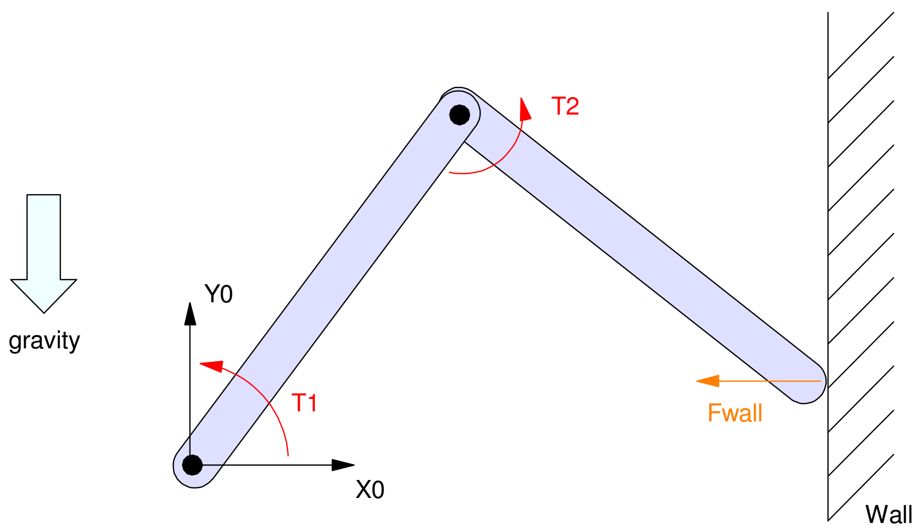
Controls: Define the joint torques to follow a prescribed path

Jacobians: By controlling the joint angles, you control the tip position of the robotic arm. By using Jacobians, you can control the XYZ position of the tip directly rather than indirectly through the joint angles.



Jacobians define how motion at the tip relates to motion of the joints.

Impact Forces: Finally, at the end of the semester we look at modeling robotic arms that come into contact with a surface.



Impact forces model the effect of hitting a surface on the dynamics of a robotic arm

Course Organization & Matlab Assignments

This course is ran as a flip-classroom. The lectures (pdf and YouTube videos) provide background for the day's topics. The homework set then asks you to write a Matlab program which uses these concepts and then demonstrate your final working code.

Documentation of your Matlab assignments can be with a word document along with screen shots of your Matlab animations, or preferable, a word document along with a video showing your robot animations in Matlab. Once you complete the homework assignments, you're done with the course.