Course Overview

ECE 494/761: Advanced Controls / Robotics

Department of Electrical and Computer Engineering
www.BisonAcademy.com,

Introduction

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

All material is located on Bison Academy

- Syllabus
- Lectures (pdf & recorded)
- Homework assignments

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

Translation & Rotation Matrices

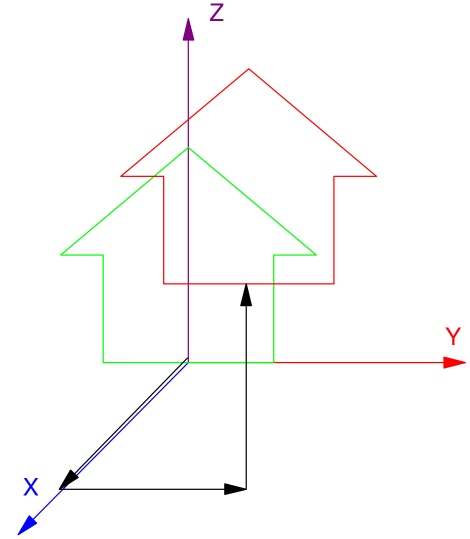
- First topic covered

Translation Matrices:

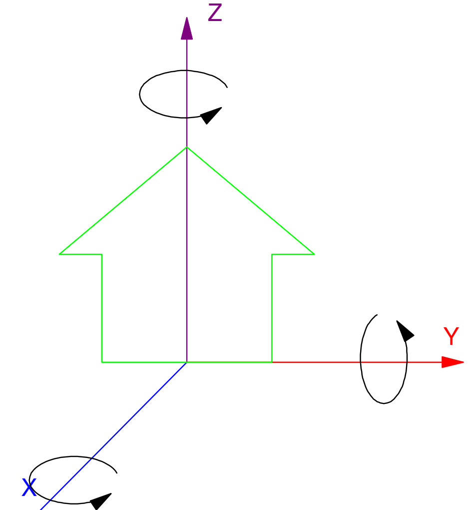
- How to move a point in the (x, y, z) direction
- How to move an object in the (x, y, z) direction

Rotation Matrices:

- Where is a point of you rotate about the (x, y, z) axis?
- Rotate an obect about its (x, y, z) axis



Translation



Rotation

Reference Frames:

- Second Topic

Define a reference frame at each joint

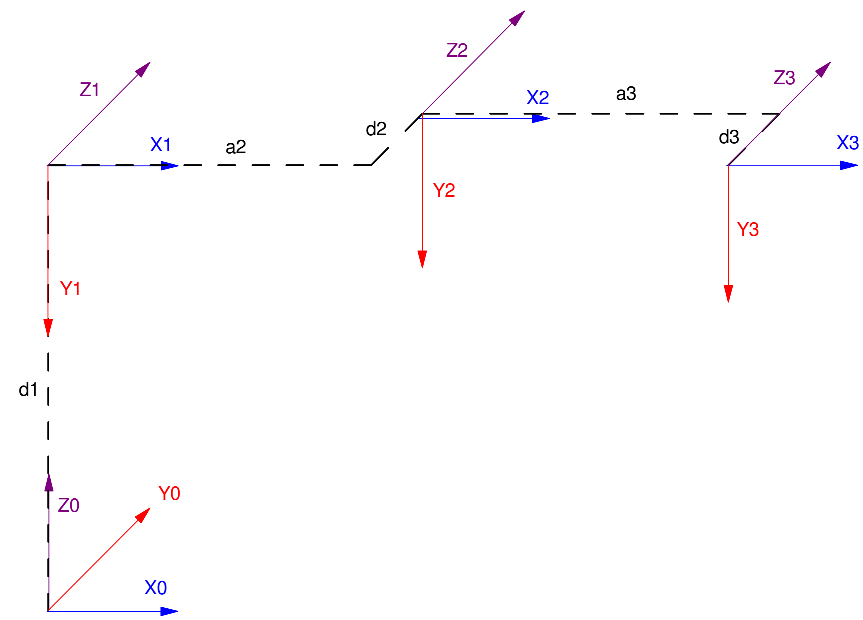
- Base, Shoulder, Elbow, Wrist

Define how you move from frame to frame

- Tranlations
- Rotations

This then defines the robotic arm

- Where each joing is located
- Where the tip is located



Forward Kinematics:

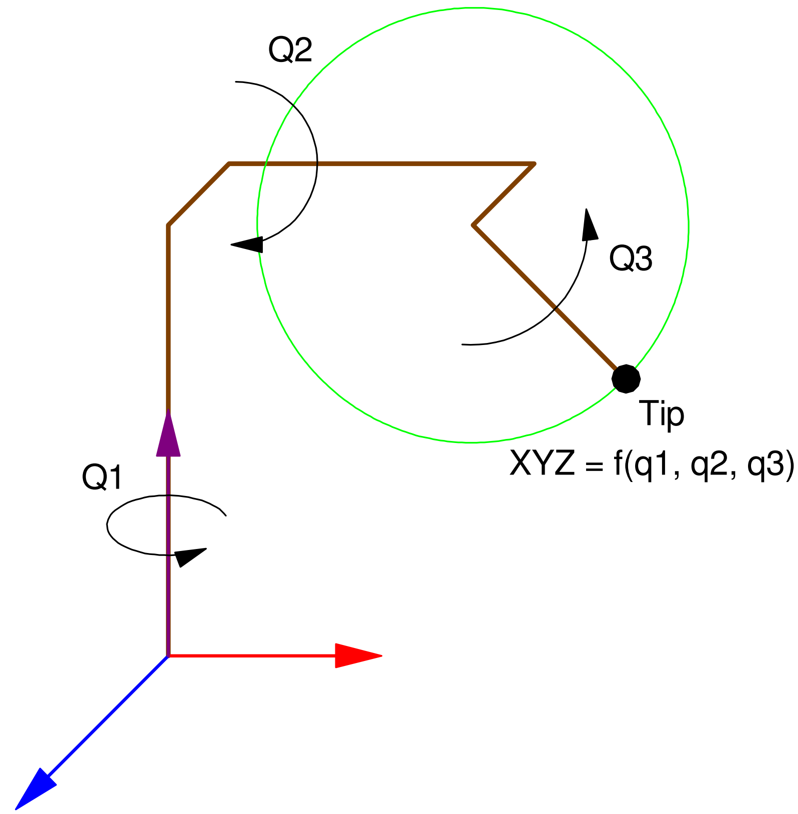
- Given the joint angles, where's the tip?

Once reference frames are defined, you know

- Where each joint is located
- Where the tip is located

This allows you to animate the motion or a robotic arm as you rotate each joint

- Matlab animations



Inverse Kinematics:

- Given the tip position, what are the joint angles?

Finding the tip position given the joint angles is pretty straight forward

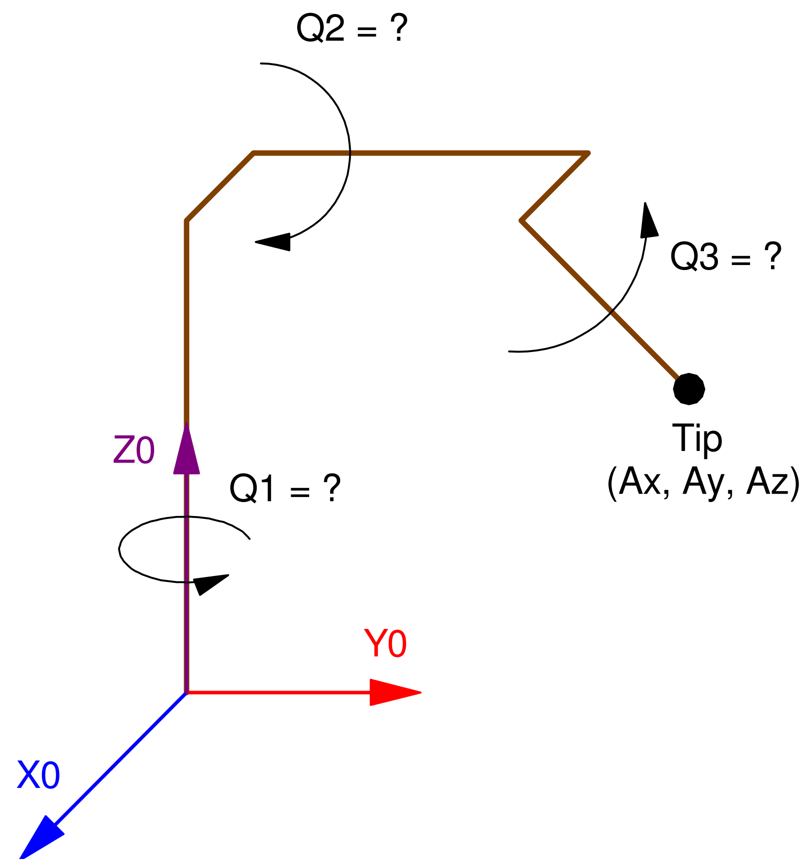
- Plug angles into the translation and rotation matrices

Going the other way gets tricky

- Nonlinear equations
- Multiple solutions

It's necessary, however

- The tip position is really what you care about.



Path Planning:

Once you have inverse kinematics, you can find the joint angles that put you at a point.

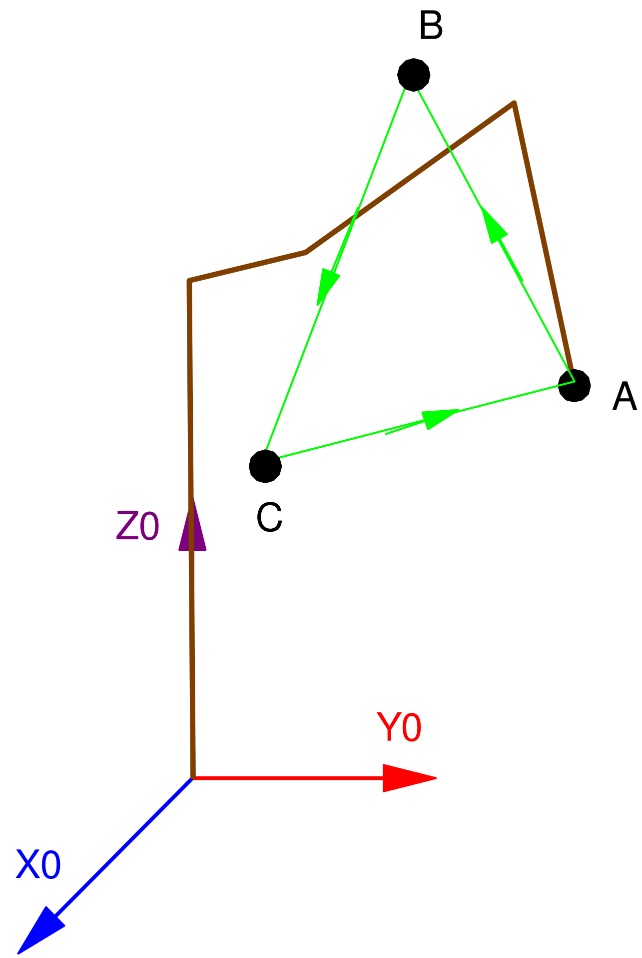
Path planning looks at how to travel from point A to point B

The goal is to do so while

- Keeping joint velocities finite, and
- Keeping joint accelerations finite

Several methods are covered in this course

- Spline curve fitting
- Cosine interpolation
- etc.



Dynamics:

- What is the differential equation relating joint torques and joint angles?

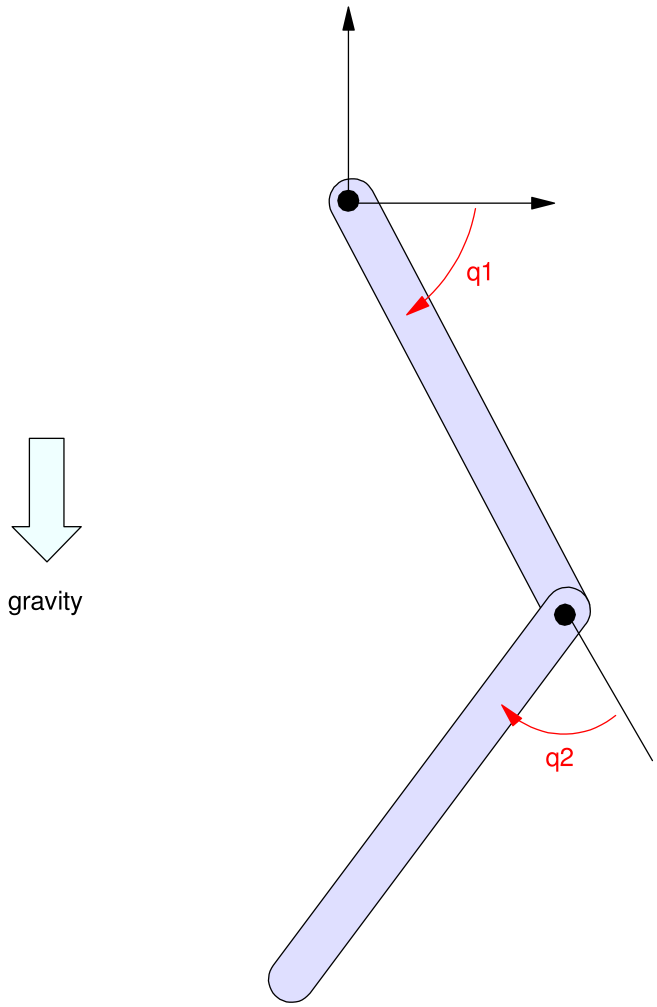
After path planning comes dynamics and controls.

Before controlling a robotic arm, the dynamics need to be defined.

This course uses

- A 2-link robotic arm
- In 2 dimensions (x, y)

More complex robotic arms are left for later studies.

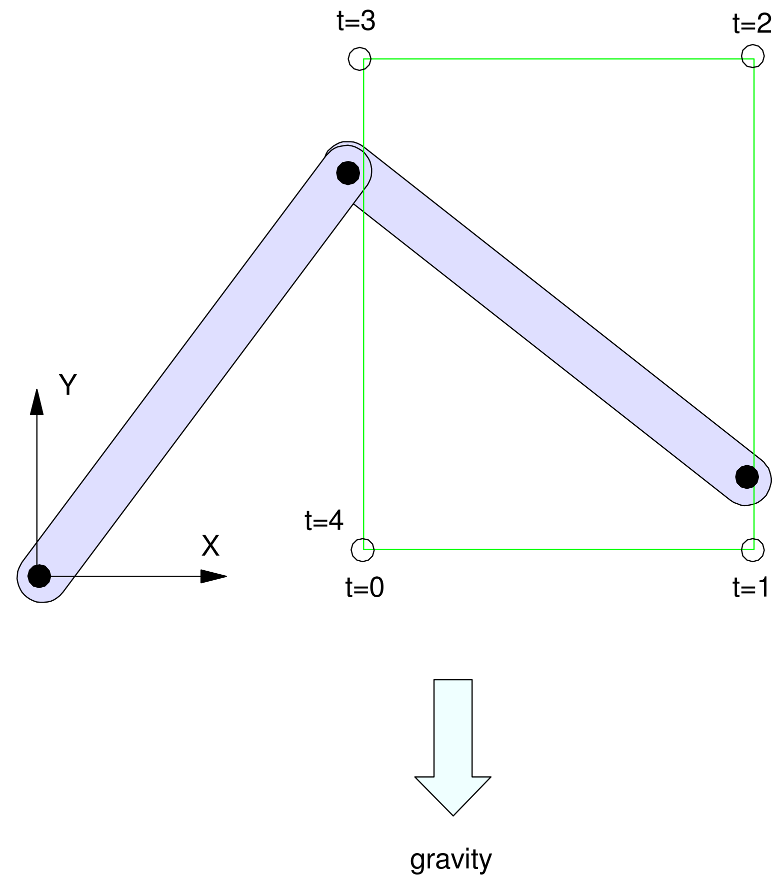


Controls:

Given

- Path Planning and
- Dynamics

determine the joint torques that cause a robotic arm to follow a prescribed path.



This uses

- Feedback (to stabilize the system)
- Feedforward Control
  - Cancel gravity
  - Cancel coriolis forces
  - Compensate for inertia

The more terms you add, the better the tracking

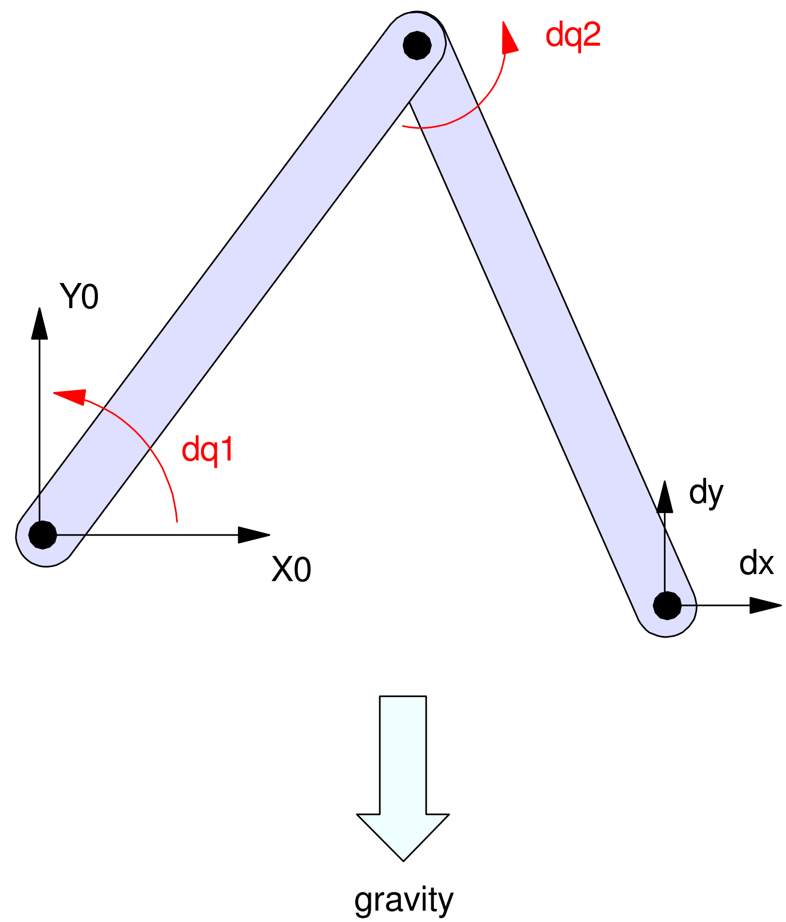
Jacobians:

Jacobians do several things

- The relate tip velocites to joint velocities
- The relate tip forces to joint torques

With Jacobians, you can control the tip position (x, y, z) rather than joint angles

- Another way to control the tip position



Impact Forces:

- Last topic

Model modeling robotic arms that come into contact with a surface.

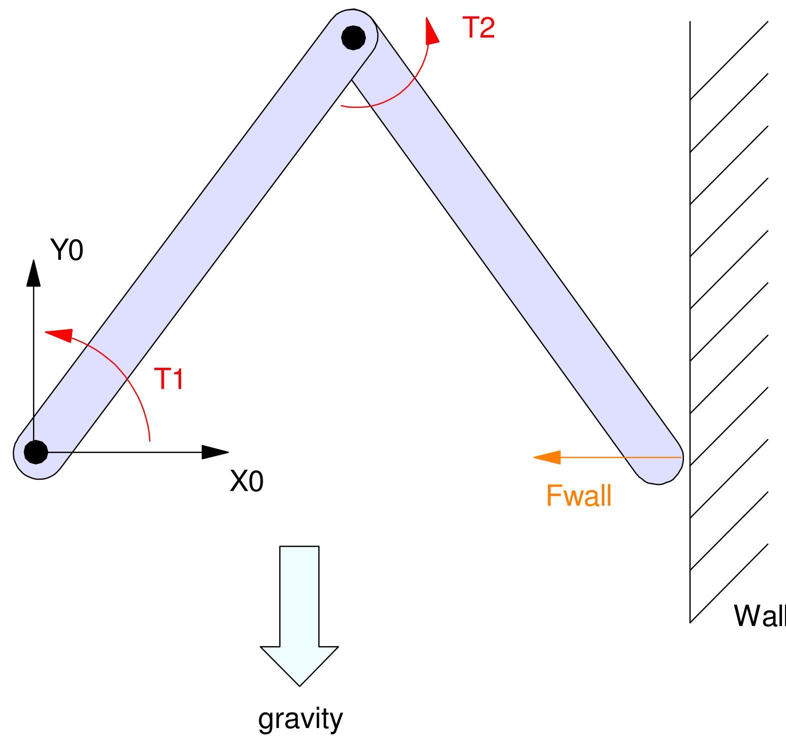
Free Space:

- N degrees of freedom

Contact;

- N-1 degrees of freedom

Tip forces translate to joint torques through the Jacobian

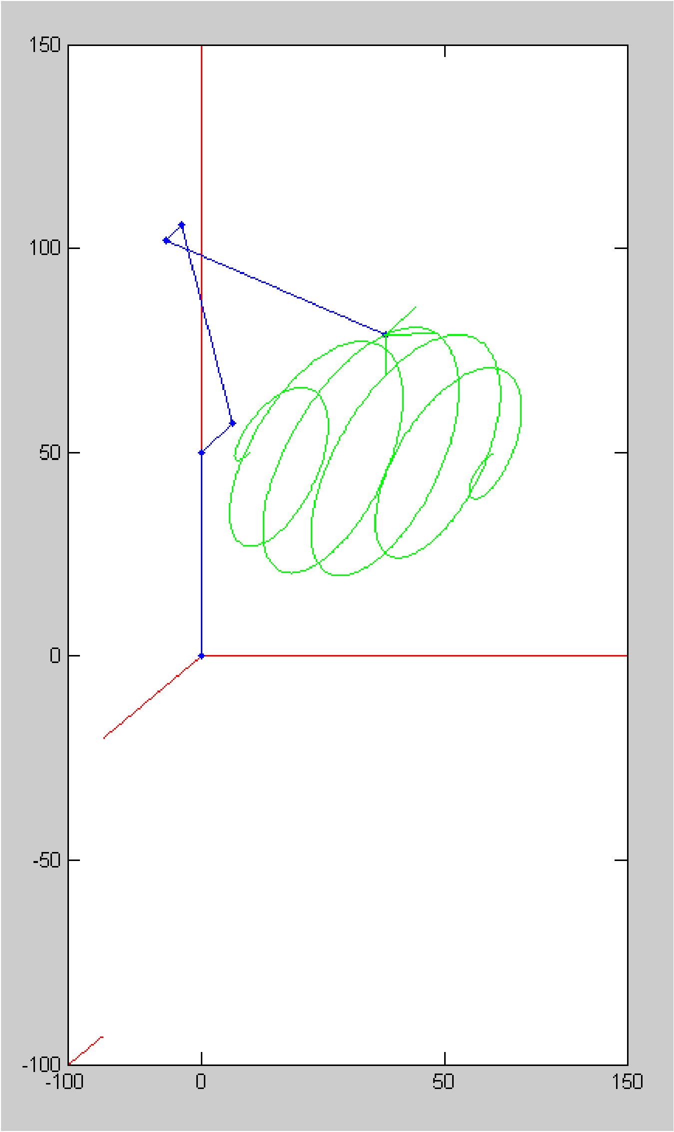


Summary

After taking this course, you should have a basic understanding of the mathematics behind modeling and control of robotic arms.

Hopefully, you'll also have fun with the Matlab animations

- Matlab isn't designed for animation but it does a pretty good job at it.


150
100
50
0
-50
-100
-100
0
50
150