9/7/2022

* Intro to coordinate frames
  + They are frames of reference
  + There’s a coordinate frame at every joint in a robot arm
* What’s a robot
  + A mechanical device that can perform various complex tasks
* Transformation matrices
  + Used to find orientation or position from two or more coordinate frames
* Forward kinematics (Movement)
  + Input angles of each joint and the output would be the end effector’s position and orientation
* Inverse kinematics
  + Input end of vector’s position and orientation and the output would be angles of each joint
* Any object in space can move in 6 degrees of freedom
* If you know all the torques in each joint, you can calculate the end of vector’s force and velocity
* Forward dynamics (Torques and forces)
  + Input all the torques in the joints, and the output would be the force and acceleration of the end of vector
* Inverse dynamics
  + Input
* What is trajectory
  + A path for each joint to follow
* Path planning
  + Moving the robot arm from A to B but avoiding certain places in space
* Force control
  + Maintaining a constant force on a surface while moving the robot
* Things to learn this semester
  + Tic Tac Toe robot
  + Pathplanning
  + Natural language processing
* A class is nothing more than a set of functions and the variables that the function acts on
  + Classdef – class definition
  + Properties – variables of that class
  + Methods – functions
  + What threads all of these functions together is the ‘object’
  + Every class must have a contructor function
* Learn live editor in Matlab

9/12/2022

* Yaw, Pitch, and Roll corresponds to the x, y, z axis respectively
* Robotics manipulator implies that parts and tools will be moved.
* We need to represent positions and orientations of all the parts, tools, and of the mechanism.
* Locating on object
  + Coordinate system (B) has been attached to the body in a known way.
  + A description of {B} relative to {A} now suffices to give the orientation of the body.
  + Theres a transformation that occurs going from {B} to {A} or vice versa to find the orientation and position of the body.
* 2D Rotations
  + Cos a = x’/r 🡪 x’ = r \*cos a = x \* cos a
  + Sin a = y’/r 🡪 y’ = r
* Rotation Matrix
* Why do we do dot product?

9/14/2022

* Homogenous transformation matrices
  + The right side of the matrix equals 0

9/19/2022



9/21/2022

Robot Kinematics Overview

* Basic Concepts
  + Robot Kinematics - studying robot motion without resorting to force and mass properties. Dealing with position, velocity, and acceleration
  + Kinematic Chain – A set of rigid bodies connected by kinematic pairs
  + Two basic Types of kinematic pairs (Joint) R&P
    - Rotating Pair – Revolute
    - Sliding Pair – Prismatic
  + You can connect these two types of pairs to make new kinematic pairs
    - Many kinds
* Robot Manipulators
  + Kinematic Chains: Link (Rigid Body) + Joint (Kinematic Pair)
* Basic Topologies of Kinematic Chain
  + Open Chain
  + Tree
  + Necklace
* Basic Problems in Robotic Kinematics
  + Direct (or Forward) Kinematics
  + Inverse Kinematics
* Denavit-Hartenberg Notation (DH Parameters)
  + Purpose – To uniquely define the architecture of a rbot manipulatior (kinematic chain)
  + Where the joint to moving, that’s the z-axis

09/28/2022

Quaternions

* A common problem with Euler angles is gimbal lock.
  + This results when two axes coincide after a rotation by some integer multiple of 90 degrees about a third axis, resulting in a singularity, i.e. a loss of a degree of freedom
* A quaternion is a four-part hyper-complex number used in three-dimensional rotaions and orientations. It is an extension of complex numbers
* A quaternion number is represented in the form a+bi+cj+dk, where a,b,c, and d parts are realnumebrs, and I, j , and k are the basis elements, satisfying the equation: i^2 = j^2 = k^2 = ijk = -1.
* All rotations in 3-D can be described by an axis and an angle about that axis.

Optimization

* Fminsearch Function
  + To minimize f(x,y) = 2+x-y+2x^2+2xy+y^2, rewrite as f(x1, x2)
* Fminsearch(f, [ ])
  + We start with an initial guess of all our DH parameters and the offset values. The function will then optimize for the least error
  + Create our model – use link and serial link inside this objective function
  + Our model is fkine ( )
    - Our data points are the dh parameters and the offset
    - We can predict the end effector position and orientation
    - Our x and y variables are the theta 1, theta 2, depth, and distance.
      * Take the distance from clipboard coordinate frame to tip of pen at end effector.
    - The objective function that you put into fminsearch will give only one error
      * Put modeled data into objective function - Fkine( ) – our error value
      * Put Measured DH parameters into fminsearch
      * Fminsearch( @A, …) A is an array of all our measured dh parameters
      * Objective function:
        + Input measured dh parameters.
        + Build robot like how you normally build robot with the same parameters
        + Read file to get all measurements and modeled data
        + Use fkine to get
        + Find error by getting the distance between the two
        + Fminsearch will return the optimized dh parameters
* Kinematics
* How to find workspace
  + For(i=1:n)
    - For(j=1:n)
      * T = fkine(…)
        + We are moving the robot very slightly and capturing the location of the robot

10/05/2022

Kinematics in Detail

* Jacobian Model (simplified)
  + Keeping everything the same, change only 1 parameter and see what the output is.
  + Tells you how important each joint is in the system
    - We can measure the mobility of a design
    - Jacobian matrices are used in inverse kinematics algorithms