Homework 3:
DH Parameters and Inverse Kinematics

MEAM 520, University of Pennsylvania
Katherine J. Kuchenbecker, Ph.D.

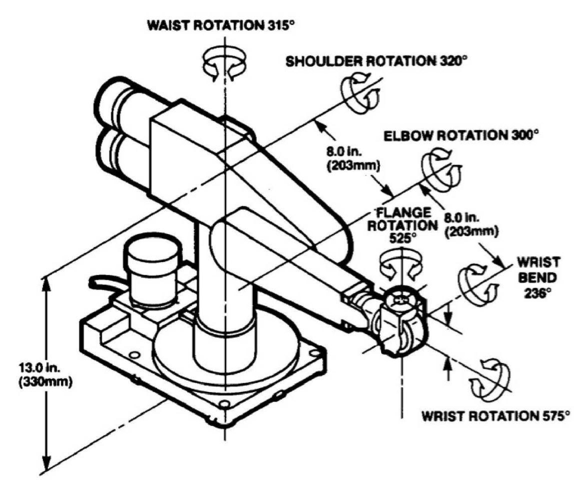
September 30, 2012

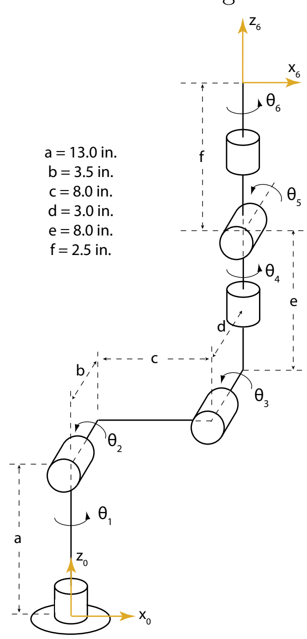
This assignment is due on **Tuesday, October 9**, by 5:00 p.m. sharp. You should aim to turn the paper part in during class that day. If you don't finish until later in the day, you can turn it in to Professor Kuchenbecker's office, Towne 224. The code must be emailed according to the instructions at the end of this document. Late submissions of either or both parts will be accepted until 5:00 p.m. on Wednesday, but they will be penalized by 25%. After that deadline, no further assignments may be submitted.

You may talk with other students about this assignment, ask the teaching team questions, use a calculator and other tools, and consult outside sources such as the Internet. To help you actually learn the material, what you write down should be your own work, not copied from a peer or a solution manual.

Forward Kinematics of the PUMA 260 *(30 points)*

The first two problems center on the forward kinematics of the PUMA 260. We will be using this robot for the hands-on manipulator labs in this class. It is an articulated robot (RRR) with lateral offsets, plus a spherical wrist (RRR). The drawing below on the left shows the robot and the arrangement of its joints.





The schematic above on the right shows the zero configuration we have chosen for use in this class (a different pose from the drawing at the left). The joint angle arrows show the positive direction for each revolute joint (θ_1 to θ_6). All of the joints are shown at $\theta_i = 0$. The diagram also gives the measurements for the constant dimensions (a to f), all in inches.

1. **DH Parameters for the PUMA 260** ***(14 points)***
   Annotate the full-page schematic of the PUMA (provided later in this document) with appropriately placed coordinate frames, and then write a table of the corresponding DH parameters; use degrees for the angles. Do this in pencil so that you can make corrections if needed. You may find it useful to follow the steps provided in SHV Section 3.4.

2. **DH Function in MATLAB** ***(4 points)***
   Download the file called `dh_starter.m` from Homework 3 on the MEAM 520 wiki; all of the files you need for this assignment will probably be zipped together for convenience. Change the file name and the function definition (on the first line of the file) to use your PennKey rather than the word `starter`.

   Update this file so that it calculates the four-by-four transformation matrix `A` for the set of DH parameters passed in by the user: `a`, `alpha`, `d`, and `theta`. The input angles have units of degrees, so use appropriate trigonometric functions. Test your function by calling it from the command line with various inputs before moving on to the next step.

3. **Animating the PUMA 260** ***(12 points)***
   Download the file called `puma_robot_starter.m` from Homework 3 on the MEAM 520 wiki and put it in the same folter as the DH function you just wrote. Change the file name to use your PennKey rather than the word `starter`, and enter your name as the value of the variable `student_name`.

   Update this script to animate the movement of the PUMA 260, using the origins of your seven frames (frame 0 to frame 6) as the points to be plotted. Ignore joint limits. All of your code should be between the two lines of asterisks. The first step is to calculate the robot's six A matrices (`A1` through `A6`) given the current values of the joint variables; you should use the DH function you created in the previous step to do this. Note that the robot's measurements are already defined for you as `a` through `f`.

   Then calculate the position of the origin of each frame (`o1` through `o6`) in the base frame; `o0` is done for you. All seven origin locations should be put together in the variable `points_to_plot` in order, with the origin of frame 6 as the last column. Use the seven available motion modes to test your calculations, and fix everything that you notice is not correct. You are welcome to add more motion modes to test other trajectories as needed, but please don't modify the ones that are already defined.

   Instructions for turning in your MATLAB files appear at the end of this document.

Inverse Kinematics *(30 points)*

These questions both deal with inverse kinematics of 3-DOF robots.

A. **Making the SCARA Draw a Vertical Circle** ***(12 points)***
Read Example 3.10 on pages 108–109 of the textbook. It derives the inverse kinematics of a SCARA manipulator with the goal of placing the end-effector at $[o_x\ o_y\ o_z]^T$. The provided solution includes formulas for θ_2, θ_1, and d_3 (as well as θ_4, which we will be ignoring), but unfortunately it includes two mistakes. Furthermore, the book's convention is to list the arguments to the atan2 function in the opposite order from MATLAB, which is confusing; I prefer to show the numerator and denominator of the tangent in a fraction. Finally, note that $d_1 = 0$ in this analysis. The corrected answers are as follows:

$$\cos\theta_2 = \frac{o_x^2 + o_y^2 - a_1^2 - a_2^2}{2a_1a_2} \tag{1}$$

$$\theta_2 = \operatorname{atan2}\left(\frac{\pm\sqrt{1-\cos^2\theta_2}}{\cos\theta_2}\right) \tag{2}$$

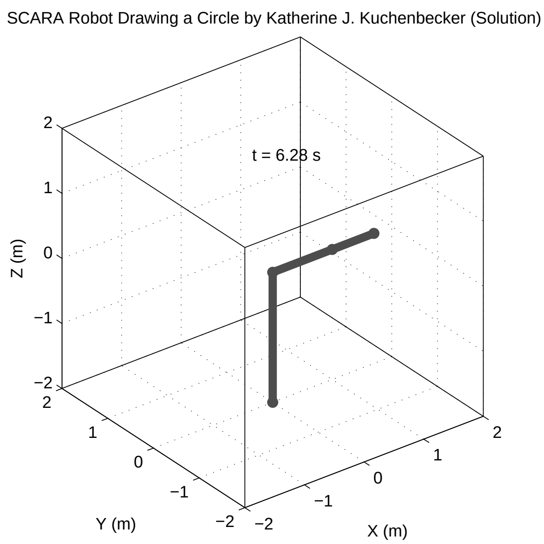
$$\theta_1 = \operatorname{atan2}\left(\frac{o_y}{o_x}\right) - \operatorname{atan2}\left(\frac{a_2\sin\theta_2}{a_1 + a_2\cos\theta_2}\right) \tag{3}$$

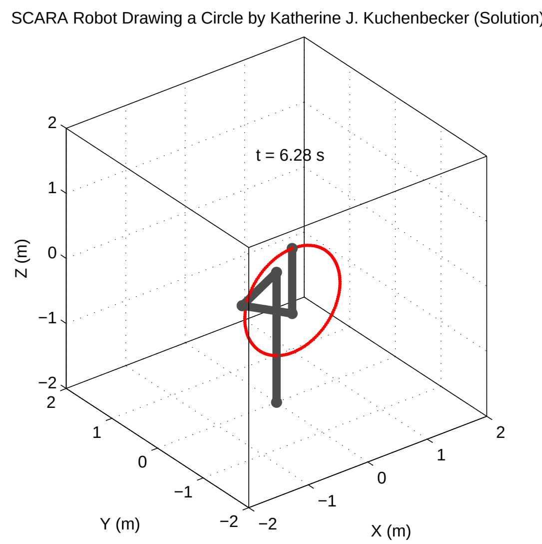
$$d_3 = -o_z \tag{4}$$

Download the script called `scara_robot_circle_starter.m` from Homework 3 on the MEAM 520 wiki. This script sets up an environment to animate the movement of a SCARA robot drawing a vertical circle over time. Rename the starter file to `scara_robot_circle_yourpennkey.m`, and put your name at the top of the file where it says `student_name = 'PUT YOUR NAME HERE';`

Download the file called `scara_robot_fk.p` from Homework 3 on the MEAM 520 wiki. This is a content-obscured MATLAB function that does the forward kinematics calculations for the SCARA and passes back the locations of the origins of all of its frames, for use in plotting. This is essentially the solution to the MATLAB part of Homework 2, so I am obscuring the contents instead of providing an m-file.

Run the code and watch what happens. As provided, the script should show a SCARA robot sitting still with all of its joint coordinates equal to zero, as shown below left.





Your job is to update this script so that the SCARA draws a circle in a plane parallel to the x-z plane, as shown above right. The desired `radius`, `y_offset`, `x_center`, and `z_center` are defined in the code (feel free to change these), along with the desired trajectory of the robot's tip. The provided starter code functions much as the code for Homework 2 did, except it provides you with `ox`, `oy`, and `oz` instead of the joint coordinates. You need to calculate `theta1` (in radians), `theta2` (in radians), and `d3` (in meters) from `a1`, `a2`, `ox`, `oy`, and `oz`. You should only need to update the code between the two lines of stars. Note that these angles are in radians, while the ones in the PUMA 260 problem are in degrees.

Once you get this to work, spend some time playing with the circle parameters to improve your understanding of inverse kinematics. Instructions for submitting your code are at the end of this assignment.

B. **SHV 3-12 – Inverse Kinematics for the Planar RPR** ***(18 points)***
Make sure to answer all of the questions stated in SHV 3-11. Here is a somewhat embellished list of the questions you should answer:

a. Given a desired position of the end effector, how many solutions are there to the inverse kinematics of the three-link planar arm shown in Figure 3.33? How does the number of solutions depend on the desired position, if at all?

b. If the orientation of the end effector is also specified, how many solutions are there? How does the number of solutions depend on the desired position and orientation, if at all?

c. Use the geometric approach to find the inverse kinematic solution(s) for the case when both the position and orientation of the end effector are specified as o_x, o_y, and α, remembering the concept of kinematic decoupling.

Do this problem in pencil on paper, and submit it with the other paper parts of this assignment.

Submitting Your Code

Follow these instructions to submit your code:

1. Start an email to `meam520@seas.upenn.edu`
2. Make the subject *Homework 3: Your Name*, replacing *Your Name* with your name.
3. Attach your three correctly named MATLAB files to the email. They should be `dh_yourpennkey.m`, `puma_robot_yourpennkey.m`, and `scara_robot_circle_yourpennkey.m`
4. Optionally include any comments you have about this assignment.
5. Send the email.

You are welcome to resubmit your code if you want to make corrections. To avoid confusion, please state in the new email that it is a resubmission, and include all three of your MATLAB files, even if you have updated only one or two of them.

Zero Configuration for the PUMA 260