Writeup for *CS3630* Homework 3

Nathan Korzekwa
Frank the Tank

Section 2.2 (Deliverable)
The tool pose $T_t^s(0)$ is given by the following matrix:

$$T_t^s(0) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 54 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

which was obtained by composing the 3D rotatation matrix around the X-axis, $R([\theta, 0, 0]^T)$, with a simple displacement vector, given by:

$$\vec{p} = \begin{bmatrix} 0 \\ 54 \\ 0 \\ 1 \end{bmatrix}$$

where all values are in pixels, given by the distance from the origin of the base frame.

Section 2.3 (Deliverable)
The exponential map for joint 1 is obtained by concatenating the rotation matrix around the joint's axis with a translation matrix from joint 1's origin to the base frame's origin. By measurement, I have found that, using the base configuration as detailed in the PDF, the translation matrix is near zero. Since joint 1 is the only rotation around the Z axis, this translation matrix is really non-consequential, since it's orthogonal to the other joints. Nevertheless, 0 works fine, and the resulting exponential map is as follows:

$$exp(\xi_1\theta_1) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} cos(\theta_1) & -sin(\theta_1) & 0 & 0 \\ sin(\theta_1) & cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

This can be written as $IR_z(\theta_1)I$, which obviously just reduces to $R_z(\theta_1) = R([0, 0, \theta]^T)$.

Section 2.4 (Deliverable) The exponential map for joint 2 is obtained in the same way as the map for joint 1, except the translation matrix is no longer I. In the defined zero configuration, the measured offset from the origin of the base frame is given by:

$$\vec{p} = \begin{bmatrix} 0 \\ 0 \\ 39 \\ 1 \end{bmatrix}$$

Where values are in pixels as always.
From this, we continue like we did with joint 1, except we use a rotation matrix around the X axis:

$$exp(\xi_2\theta_2) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 39 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(\theta_2) & -sin(\theta_2) & 0 \\ 0 & sin(\theta_2) & cos(\theta_2) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -39 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

which can be compactly represented as $exp(\xi_2\theta_2) = PR_x(\theta_2)P^{-1}$. Because *fuck that guy.*

Section 2.5 (Deliverable) *maek 1t stahp plz. ... plz.*
For the last joint (our illustrious joint 3), we follow the exact same process that we did for joint 2:

$$exp(\xi_3\theta_3) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 35 \\ 0 & 0 & 1 & 50 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(\theta_3) & -sin(\theta_3) & 0 \\ 0 & sin(\theta_3) & cos(\theta_3) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -35 \\ 0 & 0 & 1 & -50 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Section 2.6 (Deliverable)

$$exp(\xi_1\theta_1)exp(\xi_2\theta_2)exp(\xi_3\theta_3)T_t^s(0) =$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} cos(\theta_1) & -sin(\theta_1) & 0 & 0 \\ sin(\theta_1) & cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 39 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(\theta_2) & -sin(\theta_2) & 0 \\ 0 & sin(\theta_2) & cos(\theta_2) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -39 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 35 \\ 0 & 0 & 1 & 50 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(\theta_3) & -sin(\theta_3) & 0 \\ 0 & sin(\theta_3) & cos(\theta_3) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -35 \\ 0 & 0 & 1 & -50 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 54 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Section 4 (Inverse Kinematics)
See the attached file "ScreencapWorkspace.png" for a decent approximation of the robot's workspace. This type of setup would only be useful for a very limited number of situations. It can't reach down to its base except in two infinitesimal locations on it's two sides, and it's restricted in movement to one 2D cross-section of the world space. However, it would be decent at doing something such as manipulating air-borne objects, assuming a suitable end-effector.

One improvement that could be made to this arm would be the addition of more degrees of freedom via more joints. To be truly effective, the joints must be able to loop back around to the origin of the base frame, allowing full access to the plane spanning the XY plane that contains the origin of the base frame. A more robust solution, in my mind, would be allowing the limbs to change size, within limits. This would open up the entire semi-circle enclosed by the full revolution of joint 1, with all other joints 0 to the workspace.

Section 5 (Real Inverse Kinematics)
For the deliverables for this part of the assignment, please refer to the attached PNG files. All of them show the iterative Jacobian method for solving for the angles, and they are all near or at convergence in the pictures. One implementation note: I did not limit the angles that the joints could take on, so the workspace in this instance is much larger than in the previous parts.
In finding the Jacobian for a system of three joints, I needed to derive the sample provided in the manipulators PDF myself. So I created two equations to get the X and Y coordinates from the joint angles, and took their partial derivatives fo reach angle. These equations are as follows:

$$x_2(\theta_0, \theta_1, \theta_2) = r_x + L_0 sin(\theta_0) + L_1 sin(\theta_0 + \theta_1) + L_2 sin(\theta_0 + \theta_1 + \theta_2)$$

and similarly,

$$y_2(\theta_0, \theta_1, \theta_2) = r_y + L_0cos(\theta_0) + L_1cos(\theta_0 + \theta_1) + L_2cos(\theta_0 + \theta_1 + \theta_2)$$

thus giving the partials

$$\frac{\partial x_2}{\partial \theta_0} = L_0cos(\theta_0) + L_1cos(\theta_0 + \theta_1) + L_2cos(\theta_0 + \theta_1 + \theta_2)$$

$$\frac{\partial x_2}{\partial \theta_1} = L_1cos(\theta_0 + \theta_1) + L_2cos(\theta_0 + \theta_1 + \theta_2)$$

$$\frac{\partial x_2}{\partial \theta_2} = L_2cos(\theta_0 + \theta_1 + \theta_2)$$

so, it was from here that I put the partials into the formulas derived and given in the manipulators.pdf file and used them to implement the Gradient Descent algorithm that gets the angles.