Quaternion Based Kinematics

Matt Richard

January 26, 2015

1 Abstract

This paper compares the use of quaternions over the traditional use of homogeneous transformation matrices for computing forward and inverse kinematics on a 5-DOF arm. The forward kinematics are derived with both homogeneous transformation matrices and quaternions and the results are computationally compared. Quaternions prove to be more computationally efficient and use much less memory. The inverse kinematics are derived using homogeneous transformation matrices and dual quaternions. Quaternions alone lack the ability to encode translations, so dual quaternions are used because they allow for encoding a rotation and translation into a single mathematical object.

2 Introduction

Quaternions were first introduced by Charles Rowan Hamilton in 1843 [3]. Hamilton originally used them to compute vector division, but later on they proved useful for performing rotations on vectors in $\mathbb{R}^3$. The applications for quaternions are in classical and quantum mechanics, robotics, computer graphics, geometric analysis, and aerospace [1].

All of these fields rely heavily on geometry transformations and rotations. When dealing with rotations quaternions have become so popular in these fields because they eliminate the possiblity of singularities and gimble lock. "Gimbal lock is the loss of one degree of freedom in a three-dimensional, three-gimbal mechanism that occurs when the axes of two of the three gimbals are driven into a parallel configuration, "locking" the system into rotation in a degenerate two-dimensional space." [5]

The traditional method for computing kimematics for a robotic manipulator is to use homogeneous tranformation matrices, but there are many benefits of using quaternions instead. A rotation matrix requires nine values to be stored with some many values duplicates, whereas the equivalent quaternion only have four elements needing to be stored. Also, using homogeneous transformation matrices results in twelve values needing to be stored and as with a rotation matrix many values are duplicates. On the other hand, a dual quaternion has eight values needing to be stored. Thus, using a quaternion over a rotation matrix and using a dual quaternion over homogeneous transformation matrices result in better storage efficiency. Not only do quaternions reduce storage requirements, they also may remove singularities in the results as seen in [2].

In this paper, quaternions and homogenous transformation matrices are used to compute the forward kinematics of a 5-DOF robot manipulator. The result of the two methods for computing forward kinematics are then compare computationally. After, the inverse kinematics will be derived using the traditional method of using homogeneous transformation matrices and also using dual quaternions.

3 Quaternions

Quaternions are an extension to complex numbers called hyper complex numbers of rank 4. Where as complex numbers have a single imaginary component $\boldsymbol{i}$, quaternions have three: $\boldsymbol{i}$, $\boldsymbol{j}$, and $\boldsymbol{k}$. These components have the property

$$\boldsymbol{i}^2 = \boldsymbol{j}^2 = \boldsymbol{k}^2 = \boldsymbol{ijk} = -1 \tag{1}$$

Quaternions are the sum of a scalar and a vector and are commonly written in four ways

$$\begin{aligned} q &= (q0, \boldsymbol{q}) && (2)\\ &= q0 + \boldsymbol{q} && (3)\\ &= (q0, q1, q2, q3) && (4)\\ &= q0 + q1\boldsymbol{i} + q2\boldsymbol{j} + q3\boldsymbol{k} && (5) \end{aligned}$$

Quaternion addition is stright forward and is the sum of each of the components. So, given quaternions q and p, their sum is

$$q + p = (q0 + p0) + (q1 + p1)\,\boldsymbol{i} + (q2 + p2)\,\boldsymbol{j} + (q3 + p3)\,\boldsymbol{k}. \tag{6}$$

Quaternion multiplication is not stright forward. Given quaternions q and p, their product is

$$qp = q0p0 + \boldsymbol{q} \cdot \boldsymbol{p} + q0\boldsymbol{p} + p0\boldsymbol{q} + \boldsymbol{q} \times \boldsymbol{p}, \tag{7}$$

but this can be obtained by expanding and reordering the equation $(q0 + q1\boldsymbol{i} + q2\boldsymbol{j} + q3\boldsymbol{k})\,(p0 + p1\boldsymbol{i} + p2\boldsymbol{j} + p3\boldsymbol{k})$. Also, because of the cross product in the quaternions multiplication formula, quaternion multiplication is not commutative [3]. This non-communativity can be reasserted with the following fundamental quaternion component products shown below.

$$\begin{aligned} \boldsymbol{ij} &= \boldsymbol{k} = -\boldsymbol{ji} && (8)\\ \boldsymbol{jk} &= \boldsymbol{i} = -\boldsymbol{kj} && (9)\\ \boldsymbol{ki} &= \boldsymbol{j} = -\boldsymbol{ik} && (10) \end{aligned}$$

Before we continue we must define a few more properties of quaternions: the complex conjugate, norm, inverse, and the notion of a pure quaternion. The complex conjugate of a quaternion is similar to the idea of the complex conjugate for a complex number; we simply just negate the imaginary components. So, the complex conjugate of quaterion q is

$$q^* = q0 - q1\boldsymbol{i} - q2\boldsymbol{j} - q3\boldsymbol{k}. \tag{11}$$

The norm, also known as the length, of a quaternion is

$$\|q\| = qq^* = \sqrt{q0^2 + q1^2 + q2^2 + q3^2}, \tag{12}$$

and the inverse is defined as

$$q^{-1} = \frac{q^*}{\|q\|}, \|q\| \neq 0. \tag{13}$$

If $\|q\| = 1$ then q is a unit quaternion and the inverse becomes $q^{-1} = q^*$. Finally, a pure quaternion is a quaternion whose real part is zero. So,

$$q = 0 + q1\boldsymbol{i} + q2\boldsymbol{j} + q3\boldsymbol{k} \tag{14}$$

is a pure quaternion. This will become more important as we introduce rotations with quaternions in the next section.

4 Quaternion Rotation Operator

It's been stated repeatedly that quaternions allow for performing rotations of vectors. In this section, we introduce the rotation operator. Given a normalized quaternion of the form

$$q = \cos\theta + \boldsymbol{u}\sin\theta \tag{15}$$

the operator

$$L_q(\boldsymbol{v}) = q\boldsymbol{v}q^* \tag{16}$$

will rotate the vector $\boldsymbol{v}$ through an angle 2θ about the vector $\boldsymbol{u}$.

Essentially, we are encoding the axis of rotation and the angle of which to rotate about that axis in a quaternion. In order to multiply a vector $\boldsymbol{v}$ by quaternion q, we convert $\boldsymbol{v}$ into a pure quaternion $v = 0 + \boldsymbol{v}$. We are allowed to do this because there is an isomorphism between a vector in $\mathbb{R}^3$ and a quaternion. The result of the rotation operator is also a pure quaternion, so we can use this isomorphism again to extract the rotated vector from the result.

4.1 Example 1

Say we want to rotate the unit vector $\boldsymbol{v} =< 1, 0, 0 >$ around the z axis through and angle of $\frac{\pi}{2}$. After applying the rotation operator we expect the vector to be $\boldsymbol{w} =< 0, 1, 0 >$, where $\boldsymbol{w} = L_q(\boldsymbol{v})$. To begin, we will first build our rotation quaternion. Since the rotation operator will rotate $\boldsymbol{v}$ twice the angle used to build the rotation quaternion, we need to divide the angle we want to rotate $\boldsymbol{v}$ through by 2. So, $\theta = \frac{\pi}{2}\frac{1}{2} = \frac{\pi}{4}$. Also, since we want to rotate around the z axis, that means the vector $\boldsymbol{u} =< 0, 0, 1 >= \boldsymbol{k}$. So we have

$$\begin{aligned} q &= \cos\theta + \boldsymbol{u}\sin\theta && (17)\\ &= \cos\left(\frac{\pi}{4}\right) + \boldsymbol{k}\sin\left(\frac{\pi}{4}\right) && (18)\\ &= \frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2}\boldsymbol{k} && (19) \end{aligned}$$

Now, we can apply the rotation operator to $\boldsymbol{v}$.

$$\begin{aligned} \boldsymbol{w} &= L_q(\boldsymbol{v}) && (20)\\ &= q\boldsymbol{v}q^* && (21)\\ &= \left(\frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2}\boldsymbol{k}\right)(\boldsymbol{i})\left(\frac{\sqrt{2}}{2} - \frac{\sqrt{2}}{2}\boldsymbol{k}\right) && (22)\\ &= \left(\frac{\sqrt{2}}{2}\boldsymbol{i} + \frac{\sqrt{2}}{2}\boldsymbol{k}\boldsymbol{i}\right)\left(\frac{\sqrt{2}}{2} - \frac{\sqrt{2}}{2}\boldsymbol{k}\right) && (23)\\ &= \left(\frac{\sqrt{2}}{2}\boldsymbol{i} + \frac{\sqrt{2}}{2}\boldsymbol{j}\right)\left(\frac{\sqrt{2}}{2} - \frac{\sqrt{2}}{2}\boldsymbol{k}\right) && (24)\\ &= \left(\frac{\sqrt{2}}{2}\right)^2\boldsymbol{i} + \left(\frac{\sqrt{2}}{2}\right)^2\boldsymbol{j} - \left(\frac{\sqrt{2}}{2}\right)^2\boldsymbol{i}\boldsymbol{k} - \left(\frac{\sqrt{2}}{2}\right)^2\boldsymbol{j}\boldsymbol{k} && (25)\\ &= \frac{1}{2}\boldsymbol{i} + \frac{1}{2}\boldsymbol{j} + \frac{1}{2}\boldsymbol{j} - \frac{1}{2}\boldsymbol{i} && (26)\\ &= \boldsymbol{j} && (27) \end{aligned}$$

Thus, $\boldsymbol{w} =< 0, 1, 0 >$, which is what we expected.

4.2 Example 2

For a more interesting example, let's once again rotate the unit vector $\boldsymbol{v} =< 1, 0, 0 >$, but this time about the axis defined by the vector $< 1, 1, 1 >$ through an angle of $\frac{2\pi}{3}$. This example comes from [3]. To begin, we need to compute our unit vector $\boldsymbol{u}$ and our angle of rotation θ. The vector $\boldsymbol{u}$ must be a unit vector in the direction of the vector $< 1, 1, 1 >$, thus giving us $\boldsymbol{u} =< \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}} >$. Now if we assign θ to be our desired angle of rotation $\frac{2\pi}{3}$, this will rotate the vector $\boldsymbol{v}$ through an angle of 2θ. So, we want θ to be one-half our desired angle of rotation, hence $\theta = \frac{\pi}{3}$.

Now we are able to compute our rotation quationion:

$$q = \cos\theta + \boldsymbol{u}\sin\theta \tag{28}$$
$$= \cos\left(\frac{\pi}{3}\right) + \left(\frac{1}{\sqrt{3}}\boldsymbol{i} + \frac{1}{\sqrt{3}}\boldsymbol{j} + \frac{1}{\sqrt{3}}\boldsymbol{k}\right)\sin\left(\frac{\pi}{3}\right) \tag{29}$$
$$= \frac{1}{2} + \left(\frac{1}{\sqrt{3}}\boldsymbol{i} + \frac{1}{\sqrt{3}}\boldsymbol{j} + \frac{1}{\sqrt{3}}\boldsymbol{k}\right)\frac{\sqrt{3}}{2} \tag{30}$$
$$= \frac{1}{2} + \frac{1}{2}\boldsymbol{i} + \frac{1}{2}\boldsymbol{j} + \frac{1}{2}\boldsymbol{k} \tag{31}$$

We can now apply the rotation operation on our basis vector $\boldsymbol{v}$.

$$\boldsymbol{w} = L_q(\boldsymbol{v}) \tag{32}$$
$$= q\boldsymbol{v}q^* \tag{33}$$
$$= \left(\frac{1}{2} + \frac{1}{2}\boldsymbol{i} + \frac{1}{2}\boldsymbol{j} + \frac{1}{2}\boldsymbol{k}\right)\boldsymbol{i}\left(\frac{1}{2} - \frac{1}{2}\boldsymbol{i} - \frac{1}{2}\boldsymbol{j} - \frac{1}{2}\boldsymbol{k}\right) \tag{34}$$
$$= \left(\frac{1}{2}\boldsymbol{i} + \frac{1}{2}\boldsymbol{i}^2 + \frac{1}{2}\boldsymbol{j}\boldsymbol{i} + \frac{1}{2}\boldsymbol{k}\boldsymbol{i}\right)\left(\frac{1}{2} - \frac{1}{2}\boldsymbol{i} - \frac{1}{2}\boldsymbol{j} - \frac{1}{2}\boldsymbol{k}\right) \tag{35}$$
$$= \left(-\frac{1}{2} + \frac{1}{2}\boldsymbol{i} + \frac{1}{2}\boldsymbol{j} - \frac{1}{2}\boldsymbol{k}\right)\left(\frac{1}{2} - \frac{1}{2}\boldsymbol{i} - \frac{1}{2}\boldsymbol{j} - \frac{1}{2}\boldsymbol{k}\right) \tag{36}$$
$$= -\frac{1}{4} + \frac{1}{4}\boldsymbol{i} + \frac{1}{4}\boldsymbol{j} + \frac{1}{4}\boldsymbol{k} + \frac{1}{4}\boldsymbol{i} - \frac{1}{4}\boldsymbol{i}^2 - \frac{1}{4}\boldsymbol{i}\boldsymbol{j} - \frac{1}{4}\boldsymbol{i}\boldsymbol{k} \tag{37}$$
$$+ \frac{1}{4}\boldsymbol{j} - \frac{1}{4}\boldsymbol{j}\boldsymbol{i} - \frac{1}{4}\boldsymbol{j}^2 - \frac{1}{4}\boldsymbol{j}\boldsymbol{k} - \frac{1}{4}\boldsymbol{k} + \frac{1}{4}\boldsymbol{k}\boldsymbol{i} + \frac{1}{4}\boldsymbol{k}\boldsymbol{j} + \frac{1}{4}\boldsymbol{k}^2 \tag{38}$$
$$= -\frac{1}{4} + \frac{1}{4} + \frac{1}{4} - \frac{1}{4} + \left(\frac{1}{4} + \frac{1}{4} - \frac{1}{4} - \frac{1}{4}\right)\boldsymbol{i} + \left(\frac{1}{4} + \frac{1}{4} + \frac{1}{4} + \frac{1}{4}\right)\boldsymbol{j} \tag{39}$$
$$+ \left(\frac{1}{4} - \frac{1}{4} + \frac{1}{4} - \frac{1}{4}\right)\boldsymbol{k} \tag{40}$$
$$= 0 + 0\boldsymbol{i} + \boldsymbol{j} + 0\boldsymbol{k} \tag{41}$$
$$= \boldsymbol{j} \tag{42}$$

Therefore, the result is $\boldsymbol{w} =< 0, 1, 0 >$.

5 Dual Quaternions

In a single quaternion we can either encode a rotation or a translation, but not both. So in order to represent a translation and a rotation, we use a dual quaternion. The first quaternion encodes the rotation and the second encodes the translation. Dual quaternion notation is as follows.

$$Q(q, p) = (q, p) \tag{43}$$
$$= ([q0, < q1, q2, q3 >], < p_x, p_y, p_z >) \tag{44}$$

There are three properties we will need to know to computer inverse kinematics using dual quaternions: sum, product, and inverse. Dual quaternion addition is the sum of each of the components. Given two dual quaternions

$$Q_1 = (q_1, p_1) \tag{45}$$
$$Q_2 = (q_2, p_2) \tag{46}$$

their sum is

$$Q_1 + Q_2 = (q_1 + q_2, p_1 + p_2), \tag{47}$$

their product is

$$Q_1 Q_2 = \left(q_1 p_1, q_1 p_2 q_1^{-1} + p_1\right), \tag{48}$$

and, finally, the inverse of a dual quaternion is

$$Q_1^{-1} = \left(q^{-1}, -q^{-1} p q\right). \tag{49}$$

6 PhantomX Pincher

The forward and inverse kinematics that follow are computed for the PhantomX Pincher arm made by Trossen Robotics. The PhantomX Pincher is a 5 DOF arm, with one of the servos dedicated to opening and closing the gripper. Hence, we will treat the arm as a 4 DOF arm. Below is a picture of the PhantomX Pincher arm with each of the joints labeled.

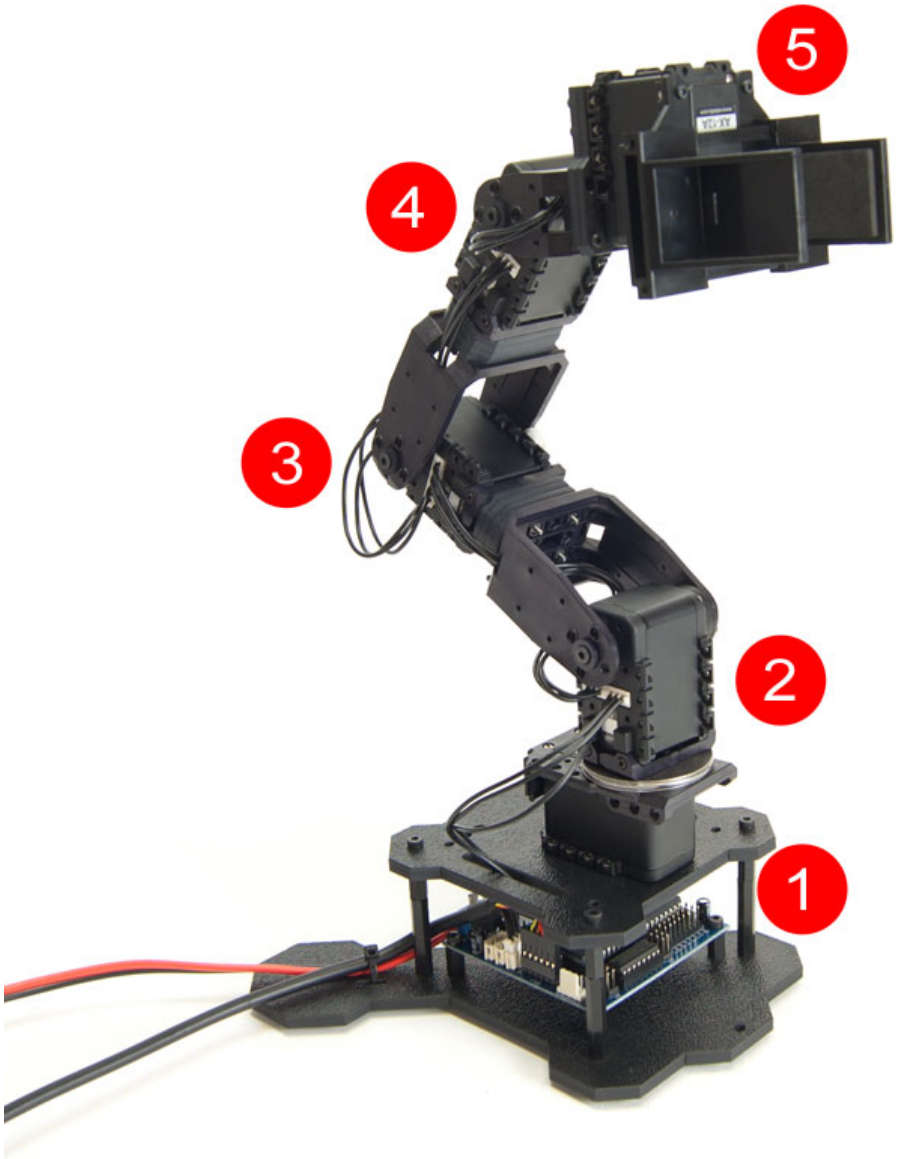

Figure 1: PhantomX Pincher robotic manipulator with each joint labeled.

The PhantomX Pincher has a 35cm vertical reach and a 31 cm horizontal reach. It uses Dynamixel AX-12A Robot Actuators as the servo actuators with a 29 degree resolution that include feedback such as speed, temperature, position, voltage, and load. Position feedback provided the ability to compute the forward kinematics which were computationally timed, comparing quaternions to homogeneous transformation matrices, with the results shown in a later section. Lastly, the servo controller is the ArbotiX-M Robocontroller, which can be controlled using serial communication or remotely via Xbee.

7 Forward Kinematics

7.1 Homogeneous Transformation Matricies

The traditional method for computing forward kinematics is by using homogenous transformation matricies to represent rotations and translations. The figure below shows a skeletal drawing of the PhantomX Pincher robotic manipulator. The values a_1 through a_4 are the lengths of each of the arms links, and θ_1 through θ_4 represent the angle of rotation for each joint.

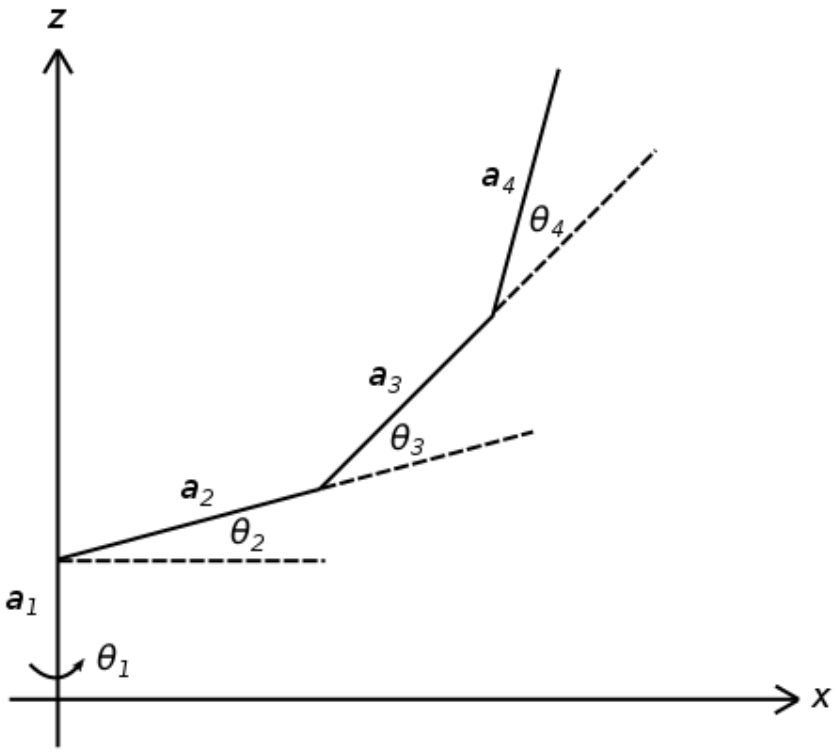

Figure 2: Skeletal drawing of the PhantomX Pincher manipulator.

We can represent the forward kinematics as a series of rotations and translations as seen below. Each R represents a rotation and each T represents a translation. The transfomation below begins with $R(z, \theta_1)$ which is a rotation about the z axis through an angle of θ_1. Next, we have $T(z, a_1)$ which represents a translation up the z axis over a length of a_1. The remaining rotations are all about the y axis and each remaining translation is over the x axis. These rotations and translations are all in reference to their local coordinate system.

$$T = R(z, \theta_1)\, T_1(z, a_1)\, R(y, \theta_2)\, T(x, a_2)\, R(y, \theta_3)\, T(x, a_3)\, R(y, \theta_4)\, T(x, a_4) \tag{50}$$

We will combine each translation with a rotation and obtain the following homogenous transformation matricies. The notation below is as follows. $c_i = \cos(\theta_i)$, $c_{ij} = \cos(\theta_i + \theta_j)$, $s_i = \sin(\theta_i)$, and $s_{ij} = \sin(\theta_i + \theta_j)$.

$$A_1 = R(z, \theta_1) T(z, a_1) = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & a_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{51}$$

$$A_2 = R(y, \theta_2) T(x, a_2) = \begin{bmatrix} c_2 & 0 & s_2 & a_2 c_2 \\ 0 & 1 & 0 & 0 \\ -s_2 & 0 & c_2 & -a_2 s_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{52}$$

$$A_3 = R(y, \theta_3) T(x, a_3) = \begin{bmatrix} c_3 & 0 & s_3 & a_3 c_3 \\ 0 & 1 & 0 & 0 \\ -s_3 & 0 & c_3 & -a_3 s_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{53}$$

$$A_4 = R(y, \theta_4) T(x, a_4) = \begin{bmatrix} c_4 & 0 & s_4 & a_4 c_4 \\ 0 & 1 & 0 & 0 \\ -s_4 & 0 & c_4 & -a_4 s_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{54}$$

Multiplying A_1 through A_4 together, we get

$$T = A_1 A_2 A_3 A_4 = \begin{bmatrix} c_1 c_{234} & -s_1 & c_1 s_{234} & c_1 (a_4 c_{234} + a_3 c_{23} + a_2 c_2) \\ s_1 c_{234} & c_1 & s_1 s_{234} & s_1 (a_4 c_{234} + a_3 c_{23} + a_2 c_2) \\ -s_{234} & 0 & c_{234} & -a_4 s_{234} - a_3 s_{23} - a_2 s_2 + a_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{55}$$

Thus, our forward kinematics are

$$x = c_1 (a_4 c_{234} + a_3 c_{23} + a_2 c_2) \tag{56}$$
$$y = s_1 (a_4 c_{234} + a_3 c_{23} + a_2 c_2) \tag{57}$$
$$z = -a_4 s_{234} - a_3 s_{23} - a_2 s_2 + a_1 \tag{58}$$

7.2 Quaternions

We can also compute the forward kinematics by using two quaternions. One quaternion represents the rotation and the other is a pure quaternion represeting translation. The quaternion representing the rotation for each joint is

$$q_1 = \cos\left(\frac{\theta_1}{2}\right) + \sin\left(\frac{\theta_1}{2}\right) \boldsymbol{k} \tag{59}$$
$$q_2 = \cos\left(\frac{\theta_2}{2}\right) + \sin\left(\frac{\theta_2}{2}\right) \boldsymbol{j} \tag{60}$$
$$q_3 = \cos\left(\frac{\theta_3}{2}\right) + \sin\left(\frac{\theta_3}{2}\right) \boldsymbol{j} \tag{61}$$
$$q_4 = \cos\left(\frac{\theta_4}{2}\right) + \sin\left(\frac{\theta_4}{2}\right) \boldsymbol{j} \tag{62}$$

The quaternions representing the translation at each link are

$$\boldsymbol{v}_1 = a_1\boldsymbol{k} \tag{63}$$
$$\boldsymbol{v}_2 = a_2\boldsymbol{i} \tag{64}$$
$$\boldsymbol{v}_3 = a_3\boldsymbol{i} \tag{65}$$
$$\boldsymbol{v}_4 = a_4\boldsymbol{i} \tag{66}$$

Now we apply the rotation operator to each link separately. The first link v_1 is only affected by the first joint q_1, thus we only use the rotation operator with q_1 on v_1. The second link v_2 is not only affected by the first but it is also affected by the second joint. So, we will apply the rotation operator twice. First with q_1, then with q_2. In a similar manner, the third link is affected by the three joints below it, and the fouth link is affected by every joint. The resulting vectors are

$$\boldsymbol{w}_1 = L_{q_1}(\boldsymbol{v}_1) = q_1\boldsymbol{v}_1q_1^* \tag{67}$$
$$\boldsymbol{w}_2 = L_{q_2q_1}(\boldsymbol{v}_2) = q_2(q_1\boldsymbol{v}_2q_1^*)q_2^* \tag{68}$$
$$\boldsymbol{w}_3 = L_{q_3q_2q_1}(\boldsymbol{v}_3) = q_3(q_2(q_1\boldsymbol{v}_3q_1^*)q_2^*)q_3^* \tag{69}$$
$$\boldsymbol{w}_4 = L_{q_4q_3q_2q_1}(\boldsymbol{v}_4) = q_4(q_3(q_2(q_1\boldsymbol{v}_4q_1^*)q_2^*)q_3^*)q_4^* \tag{70}$$

By summing up the x components together, the y components together, and the z together of the resulting vectors, we obtain the x, y, and z components of the end effector's position, respectively.

$$x = \boldsymbol{w}_{1x} + \boldsymbol{w}_{2x} + \boldsymbol{w}_{3x} + \boldsymbol{w}_{4x} \tag{71}$$
$$y = \boldsymbol{w}_{1y} + \boldsymbol{w}_{2y} + \boldsymbol{w}_{3y} + \boldsymbol{w}_{4y} \tag{72}$$
$$z = \boldsymbol{w}_{1z} + \boldsymbol{w}_{2z} + \boldsymbol{w}_{3z} + \boldsymbol{w}_{4z} \tag{73}$$

Our final rotation quaternion is

$$q = q_4q_3q_2q_1 \tag{74}$$

thus the final orientation of the end effector is

$$w = L_q(\boldsymbol{v}_4) = qv_4q^* \tag{75}$$

7.3 Forward Kinematic Timings

Using the PhantomX Pincher arm, timing was done over five runs and each run is an average over 100 calculations.

Run	Homogeneous Coordinates (μs)	Quaternions (μs)
1	2.95	0.04
2	3.17	0.05
3	3.35	0.01
4	2.57	0.04
5	1.85	0.04

As you can see, quaternions out performed homogeneous coordinates by a factor of 50-100.

8 Inverse Kinematics

The inverse kinematics PhantomX Pincher arm can be simplified by using polar coordiates. The arm reduces down to a three-link manipulator in the r-z plane, with rotation θ_1. Given the x, y, and z coordinates of the end effector, θ_1 can be easily found and is simply $\arctan\left(\frac{y}{x}\right)$. Thus, remaining is to find θ_2, θ_3, and θ_4.

8.1 Homogeneous Transformation Matricies

To start off with the derivation of the inverse kinematics of the PhantomX Pincher using homogeneous transformations matricies, we will use the results from the forward kinematics derived earlier using homogeneous transformation matricies

$$\begin{bmatrix} c_1c_{234} & -s_1 & c_1s_{234} & c_1\left(a_4c_{234}+a_3c_{23}+a_2c_2\right) \\ s_1c_{234} & c_1 & s_1s_{234} & s_1\left(a_4c_{234}+a_3c_{23}+a_2c_2\right) \\ -s_{234} & 0 & c_{234} & -a_4s_{234}-a_3s_{23}-a_2s_2+a_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{76}$$

In order to derive the inverse kinematics we must be given the end-effector orientation

$$\begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_y & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{77}$$

We begin by equating the two matricies. We can then take the square root of the sum of p_x^2 and p_y^2 to eliminate c_1 from the equations.

$$p_r = \sqrt{p_x^2+p_y^2} = a_4c_{234}+a_3c_{23}+a_2c_2 \tag{78}$$

Also, c_{234} is given to us by end-effector transformation matrix. So, we can subtract a_4c_{234} from both sides of the equation and relabel it as

$$r = p_r - a_4c_{234} = a_3c_{23}+a_2c_2 \tag{79}$$

We use the label r because we are essential using cylidrical coordinates with r being the radius from the base of the arm to the postion of the end-effector. Now, using $p_z = -a_4s_{234}-a_3s_{23}-a_2s_2+a_1$, we also know s_{234} so we may move it to the otherside and relabel as z

$$z = -p_z - a_4s_{234}+a_1 = a_3s_{23}+a_2s_2 \tag{80}$$

In cylindrical coordinates, z is the height of the end effector. Squaring both z and r and adding them together gives us the law of cosines and eliminates c_{23}, s_{23} and s_2 from the equation

$$r^2+z^2 = a_2^2+a_3^2+2a_2a_3c_3 \tag{81}$$

We may now solve for c_3

$$c_3 = \frac{r^2+z^2-a_2^2-a_3^2}{2a_2a_3} \tag{82}$$

Then,

$$s_3 = \pm\sqrt{1-c_3^2} \tag{83}$$

Thus,

$$\theta_3 = \arctan\left(s_2, c_2\right) \tag{84}$$

Now, plug θ_3 back into the equations with r and z and solve for θ_2

$$r = k_1c_2 - k_2s_2 \tag{85}$$

$$z = k_1s_2 + k_2c_2 \tag{86}$$

where $k_1 = a_2+a_3c_3$, and $k_2 = a_3s_3$.

Now, let

$$t = \sqrt{k_1^2 + k_2^2} \tag{87}$$

$$\gamma = \arctan(k_2, k_1) \tag{88}$$

Then,

$$k_1 = t\cos\gamma \tag{89}$$

$$k_2 = t\sin\gamma \tag{90}$$

Plugging these back into equations (85) and (86) we get

$$\frac{r}{t} = \cos\gamma c_2 + \sin\gamma s_2 \tag{91}$$

$$\frac{z}{t} = \cos\gamma s_1 + \sin\gamma c_1 \tag{92}$$

With a couple trigonometric identities these can be rewritten as

$$\cos(\gamma + \theta_2) = \frac{r}{t} \tag{93}$$

$$\sin(\gamma + \theta_2) = \frac{z}{t} \tag{94}$$

and so

$$\gamma + \theta_2 = \arctan(z, r) \tag{95}$$

Therefore

$$\theta_2 = \arctan(z, r) - \arctan(k_2, k_1) \tag{96}$$

Finally,

$$\theta_4 = \phi - \theta_2 - \theta_3 \tag{97}$$

Credit goes to [4] for the above derivation.

8.2 Dual Quaternions

We will now derive the inverse kinematics for the PhantomX Pincher arm using dual-quaternions. Just like in deriving the inverse kinematics using homogeneous transformation matrices, we first need the dual-quaternion of the end-effector to provide us with the end-effector's position and orientation. The end-effector dual-quaternion will be given by

$$N_1 = ([w, < a, b, c >], < p_x, p_y, p_z >) \tag{98}$$

Next we need to build the forward and inverse dual-quaternions for each joint. Some new notation will be used: $\bar{c}_i = \cos\frac{\theta_i}{2}$, $\bar{c}_{ij} = \cos\left(\frac{\theta_i+\theta_j}{2}\right)$, $\bar{s}_i = \sin\frac{\theta_i}{2}$, and finally $\bar{s}_{ij} = \sin\left(\frac{\theta_i+\theta_j}{2}\right)$. The forward dual-quaternions for each joint is

$$Q_1 = ([\bar{c}_2, \bar{s}_2 k], < a_2c_2, a_2s_2, 0 >) \tag{99}$$

$$Q_2 = ([\bar{c}_3, \bar{s}_3 k], < a_3c_3, a_3s_3, 0 >) \tag{100}$$

$$Q_3 = ([\bar{c}_4, \bar{s}_4 k], < a_4c_4, a_4s_4, 0 >) \tag{101}$$

and the inverse dual-quaternions are

$$Q_1^{-1} = ([\bar{c}_2, -\bar{s}_2 k], < -a_2, 0, 0 >) \tag{102}$$

$$Q_2^{-1} = ([\bar{c}_3, -\bar{s}_3 k], < -a_3, 0, 0 >) \tag{103}$$

$$Q_3^{-1} = ([\bar{c}_4, -\bar{s}_4 k], < -a_4, 0, 0 >) \tag{104}$$

Now, to derive the inverse kinematics we will compute six equations which are shown below.

$$Q_1Q_2Q_3 = N_1 \tag{105}$$
$$Q_2Q_3 = Q_1^{-1}N_1 \tag{106}$$
$$Q_3 = Q_2^{-1}Q_1^{-1}N_1 \tag{107}$$

We will first compute the left side of the equations, and, after, the right side. Before we continue with the left side of the equations we will first assign new labels to each of the left equations: $M_3 = Q_3$, $M_2 = Q_2M_3 = Q_2Q_3$, and $M_1 = Q_1M_2 = Q_1Q_2Q_3$. The dual-quaternion products of the left side of the equations are

$$M_3 = Q_3 \tag{108}$$
$$= ([\bar{c}_4, \bar{s}_4k], < a_4c_4, a_4s_4, 0 >), \tag{109}$$

$$M_2 = Q_2M_3 \tag{110}$$
$$= Q_2Q_3 \tag{111}$$
$$= ([\bar{c}_{34}, \bar{s}_{34}k], < a_4c_{34} + a_3c_3, a_4s_{34} + a_3s_3, 0 >), \tag{112}$$

and

$$M_1 = Q_1M_2 \tag{113}$$
$$= Q_1Q_2Q_3 \tag{114}$$
$$= ([\bar{c}_{234}, \bar{s}_{234}k], < M_{11}, M_{12}, 0 >) \tag{115}$$

where $M_{11} = a_4c_{234} + a_3c_23 + a_2c_2$ and $M_{12} = a_4s_{234} + a_3s_{23} + a_2s_2$.

Now we compute the right side of the equations. We will also be relabeling the right side of the equations to $N_2 = Q_1^{-1}N_1$ and $N_3 = Q_2^{-1}N_2 = Q_2^{-1}Q_1^{-1}N_1$. First, we again have our initial end effector dual-quaternion

$$N_1 = ([w, < a, b, c >], < p_x, p_y, p_z >) \tag{116}$$

Then,

$$N_2 = Q_1^{-1}N_1 \tag{117}$$
$$= ([N_{21}, < N_{22}, N_{23}, N_{24} >], < N_{25}, N_{26}, p_z >) \tag{118}$$

where $N_{21} = w\bar{c}_2 + c\bar{s}_2$, $N_{22} = a\bar{c}_2 + b\bar{s}_2$, $N_{23} = b\bar{c}_2 - a\bar{s}_2$, $N_{24} = c\bar{c}_2 - w\bar{s}_2$, $N_{25} = p_xc_2 + p_ys_2 - a_2$, and $N_{26} = p_yc_2 - p_xs_2$, and finally

$$N_3 = Q_2^{-1}N_2 \tag{119}$$
$$= Q_2^{-1}Q_1^{-1}N_1 \tag{120}$$
$$= ([N_{31}, < N_{32}, N_{33}, N_{34} >], < N_{35}, N_{36}, p_z >) \tag{121}$$

where $N_{31} = w\bar{c}_{23} + c\bar{s}_{23}$, $N_{32} = a\bar{c}_{23} + b\bar{s}_{23}$, $N_{33} = b\bar{c}_{23} - a\bar{s}_{23}$, $N_{34} = c\bar{c}_{23} - w\bar{s}_{23}$, $N_{35} = p_xc_2 + p_ys_2 - a_2 - a_3$, and $N_{36} = p_yc_2 - p_xs_2$.

Now, we combine the left and right side of the equations shown earlier by matching M_1 with N_1, M_2 with N_2, and M_3 with N_3. The results are shown below.

$M_1 = N_1$:

$$\bar{c}_{234} = w \tag{122}$$
$$0 = a \tag{123}$$
$$0 = b \tag{124}$$
$$\bar{s}_{234} = c \tag{125}$$
$$a_4c_{234} + a_3c_{23} + a_2c_2 = p_x \tag{126}$$
$$a_4s_{234} + a_3s_{23} + a_2s_2 = p_y \tag{127}$$
$$0 = p_z \tag{128}$$

$M_2 = N_2$:

$$\bar{c}_{34} = w\bar{c}_2 + c\bar{s}_2 \tag{129}$$
$$0 = a\bar{c}_2 + b\bar{s}_2 \tag{130}$$
$$0 = b\bar{c}_1 - a\bar{s}_1 \tag{131}$$
$$\bar{s}_{34} = c\bar{c}_2 - w\bar{s}_2 \tag{132}$$
$$a_4c_{34} + a_3c_3 = p_xc_2 + p_ys_2 - a_2 \tag{133}$$
$$a_4s_{34} + a_3s_3 = p_yc_2 - p_xs_2 \tag{134}$$
$$0 = p_z \tag{135}$$

$M_3 = N_3$:

$$\bar{c}_4 = w\bar{c}_{23} + c\bar{s}_{23} \tag{136}$$
$$0 = a\bar{c}_{23} + b\bar{s}_{23} \tag{137}$$
$$0 = b\bar{c}_{23} - a\bar{s}_{23} \tag{138}$$
$$\bar{s}_4 = c\bar{c}_{23} - w\bar{s}_{23} \tag{139}$$
$$a_4c_4 = p_xc_2 + p_ys_2 - a_2 - a_3 \tag{140}$$
$$a_4s_4 = p_yc_2 - p_xs_2 \tag{141}$$
$$0 = p_z \tag{142}$$

We may now begin deriving the inverse kinematics. We first begin by finding $\theta_2 + \theta_3 + \theta_4$. This can be obtained from equations (122) or (125). We will use (122):

$$\bar{c}_{234} = w \tag{143}$$
$$\frac{\theta_2 + \theta_3 + \theta_4}{2} = \arccos w \tag{144}$$
$$\theta_2 + \theta_3 + \theta_4 = 2\arccos w \tag{145}$$

Now, we may continue on and find θ_3. By squaring both (126) and (127) and adding them together, we obtain

$$c_3 = \frac{\left(p_x - a_4c_\phi\right)^2 + \left(p_y - a_4s_\phi\right)^2 - a_3^2 - a_2^2}{2a_2a_3} \tag{146}$$

with $\phi = \theta_2 + \theta_3 + \theta_4$. Then,

$$s_3 = \sqrt{1 - c_3^2} \tag{147}$$

so

$$\theta_3 = \arctan\left(\frac{s_3}{c_3}\right) \tag{148}$$

By pluging θ_3 back into (126) and (127) we can solve for s_2 and c_2:

$$c_2 = \frac{\left(a_2 + a_3c_3\right)\left(p_x - a_4c_\phi\right) - a_3s_3\left(p_y - a_4s_\phi\right)}{\left(p_x - a_4c_\phi\right)^2 + \left(p_y - a_4s_\phi\right)^2} \tag{149}$$
$$s_2 = \frac{\left(a_2 + a_3c_3\right)\left(p_y - a_4s_\phi\right) - a_3s_3\left(p_x - a_4c_\phi\right)}{\left(p_x - a_4c_\phi\right)^2 + \left(p_y - a_4s_\phi\right)^2} \tag{150}$$
$$\tag{151}$$

thus we have

$$\theta_2 = \arctan\left(\frac{s_2}{c_2}\right) \tag{152}$$

Since we now have θ_2 and θ_3 we can easily find θ_4 by subtracting θ_2 and θ_3 from ϕ. So,

$$\theta_4 = \phi - \theta_3 - \theta_2 \tag{153}$$

In summary,

$$\theta_1 = \arctan\left(\frac{y}{x}\right) \tag{154}$$

$$c_2 = \frac{(a_2 + a_3 c_3)(p_x - a_4 c_\phi) - a_3 s_3 (p_y - a_4 s_\phi)}{(p_x - a_4 c_\phi)^2 + (p_y - a_4 s_\phi)^2} \tag{155}$$

$$s_2 = \frac{(a_2 + a_3 c_3)(p_y - a_4 s_\phi) - a_3 s_3 (p_x - a_4 c_\phi)}{(p_x - a_4 c_\phi)^2 + (p_y - a_4 s_\phi)^2} \tag{156}$$

$$\theta_2 = \arctan\left(\frac{s_2}{c_2}\right) \tag{157}$$

$$\phi = \theta_2 + \theta_3 + \theta_4 = \arctan\left(\frac{p_y}{p_x}\right) \tag{158}$$

$$c_3 = \frac{(p_x - a_4 c_\phi)^2 + (p_y - a_4 s_\phi)^2 - a_3^2 - a_2^2}{2 a_2 a_3} \tag{159}$$

$$s_3 = \sqrt{1 - c_3^2} \tag{160}$$

$$\theta_3 = \arctan\left(\frac{s_3}{c_3}\right) \tag{161}$$

$$\theta_4 = \phi - \theta_3 - \theta_2 \tag{162}$$

9 Conclusion

Quaternions proved much more useful when computing forward kinematics in not only computation but in storage as well. Not only are they more efficient to work with, they also eliminate the possiblity of singularities and gimble lock. Although, using dual quaternions to derive the inverse kinematics for the PhantomX Pincher arm resulted in similar equations as using homogeneous transformation matrices, the fact that the equations are equivalent shows that using dual quaternions provides a useful alternative method for deriving inverse kinematics.

References

[1] Yavuz Aydin and Serdar Kucuk. Quaternion based inverse kinematics for industrial robot manipulators with euler wrist. *Mechatronics, 2006 IEEE International Converence on*, pages 581–586, 2006.

[2] Janez Funda, Russell H. Taylor, and Richard P. Paul. On homogeneous transforms, quaternions, and computational efficiency. *IEEE T. Robotics and Automation*, 6(3):382–388, 1990.

[3] Jack B. Kuipers. *Quaternions and rotation sequences : a primer with applications to orbits, aerospace, and virtual reality*. Princeton Univ. Press, Princeton, NJ, 1999.

[4] Wikibooks. Robotics kinematics and dynamics/serial manipulator position kinematics — wikibooks, the free textbook project, 2009. [Online; accessed 7-April-2015].

[5] Wikipedia. Gimbal lock — wikipedia, the free encyclopedia, 2015. [Online; accessed 3-May-2015].