A Primer on Forward Kinematics and the Jacobian

Jonathan Tse

October 24, 2005

1 Introduction

The study of robot control is an continuosly evolving field. The original impetus for study was for the control of robotic arms used in manufacturing automation, which is now a fairly developed subject. It is standard to use position or force control in industrial robotics applications, such as manufacturing automation. The reason for this is because manufacturing automation typically involves many position-specific, repetitive tasks, such as part placement or spot welding.

2 Forward Kinematics

2.1 Overview

Industrial robots are typically constructed in the form of a multi-jointed manipulating arm. Typically, each joint of the arm is capable of revolute movement. However, it is also possible to have a prismatic joint, which allows for linear travel along an arm segment. This paper will focus only on revolute joints. The end of the arm is called the "end-effector" and is usually a tool or manipulator of some sort. Since the end effector usually will be interacting with an external object, it makes logical sense that there exist a way to describe the position and orientation of the end effector with respect to the coordinate system of the external object, hereafter referred to as the reference frame. Forward kinematics allows us to specify the position and orientaton of the end effector as a function of joint angles and arm segment lengths. Later, forward kinematics will also allow us to express the velocity and the force of the end effector as a function of angular velocity and torque of the joints, respectively.

2.2 1-Degree of Freedom

Before we begin, a quick note on notation. Subscripts on the left of a matrix denote which reference frame we are translating from, and superscripts on the left of a matrix or vector denote the reference frame of their "output." Now, we examine the simple case of a one degree of freedom revolute joint. This is the case of a robot which has a single joint which can pivot about the z-axis.

To begin, we define a universal reference frame, frame 0, as the frame of reference of the base of the robotic arm. Next, we attach another reference frame, frame 1, to the joint with its origin coincident with frame 0's. By examining the basis vectors for frame 1 and expressing them in terms of the universal frame, frame 0, we have the following, where θ_0 is the angle we rotate about the z-axis:

$$\mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \mathbf{e}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \mathbf{e}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{1}$$

$$\mathbf{e}'_1 = \begin{bmatrix} \cos\theta_0 \\ \sin\theta_0 \\ 0 \end{bmatrix}, \mathbf{e}'_2 = \begin{bmatrix} -\sin\theta_0 \\ \cos\theta_0 \\ 0 \end{bmatrix}, \mathbf{e}'_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{2}$$

Combining $\mathbf{e}'_1, \mathbf{e}'_2, \mathbf{e}'_3$ into a matrix, we have the rotation matrix for expressing vectors from frame 1 in frame 0.

$${}^0_1R = \begin{bmatrix} \cos\theta_0 & -\sin\theta_0 & 0 \\ \sin\theta_0 & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

However, ${}^0_1R$ only describes a rotation. In order to completely describe a transformation, we will have to account for the case where the origins of two different reference frames are not coincident. In this case, we can create a 4×4 transformation matrix that will allow us to both rotate and translate a vector from frame 1 to frame 0. This is done by augmenting the rotational matrix with a row of all zeros and a column which consists of the vector in frame 0 describing the position of the origin of frame 1 and a single 1. The new transformation matrix ${}^0_1T$ is shown below, where ${}^0\mathbf{o}_1$ is the vector describing the origin of frame 1 in frame 0. ${}^0\mathbf{o}_1 = \mathbf{0}$ because the origins of frame 0 and 1 are coincident as described in Chapter 2.3 of [Craig].

$$ {}^0_1T = \left[\begin{array}{ccc|c} & {}^0_1R & & {}^0\mathbf{o}_1 \\ 0 & 0 & 0 & 1 \end{array}\right] = \begin{bmatrix} \cos\theta_0 & -\sin\theta_0 & 0 & 0 \\ \sin\theta_0 & \cos\theta_0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4} $$

We now have the capability to express vectors in frame 1 in frame 0.

$$ {}^0\mathbf{x} = {}^0_1 T \left({}^1\mathbf{x}\right) \tag{5} $$

. Typically, vectors in three dimensions have are 3×1, but since ${}^0_1T$ is a 4×4 matrix, we will need to augment both $\mathbf{x}$ vectors with a row containing 1.

$$ \mathbf{x} = \begin{bmatrix} \mathbf{x}_{old} \\ 1 \end{bmatrix} \tag{6} $$

To completely describe our robotic arm, we add another reference frame, frame 2, attached to the end effector of the robot. Since our imaginary robot has only one degree of freedom, i.e. there is no rotation between frame 1 and 2, the ${}^1_2T$ will have the identity as its rotation matrix, and a origin vector describing the length of the arm segment, which in this case has length ℓ_1 in the x-direction of frame 1. Thus,

$$ {}^1_2T = \begin{bmatrix} 1 & 0 & 0 & \ell_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7} $$

Armed with our new knowledge, we can easily solve the problem of finding the location of the end of the arm in terms of the universal frame, frame 0. Using equations 4, 7, and 5, we have

$$ {}^0\mathbf{x} = {}^0_1T {}^1_2T {}^2\mathbf{o}_2 \tag{8} $$

${}^2\mathbf{o}_2$ is the origin of the end effector's reference frame, frame 2, in terms of frame 2, which means it's the zero vector. Expanding our previous expression, we have

$$ {}^0\mathbf{x} = \begin{bmatrix} \cos\theta_0 & -\sin\theta_0 & 0 & 0 \\ \sin\theta_0 & \cos\theta_0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & \ell_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \ell_1 \cos\theta_0 \\ \ell_2 \sin\theta_0 \\ 0 \\ 1 \end{bmatrix} \tag{9} $$

Note the 1 in the bottom row of the vector. That 1 is an artifact of using the 4×4 matrix. Extracting the actual vector for the end of the arm in frame 0 is also easy. Simply ignore the fourth row, and we have

$$ {}^0\mathbf{o}_2 = \begin{bmatrix} \ell_1 \cos\theta_0 \\ \ell_1 \sin\theta_0 \\ 0 \end{bmatrix} \tag{10} $$

The robotic arm in this example is essentially capable of tracing out a circle in the x-y plane of radius ℓ_1. The vector that sweeps the circle out is obviously equivalent to ${}^0\mathbf{o}_2$ if θ_0 sweeps from 0 to 2π, so our matrix math is correct.

3 The Jacobian

3.1 Overview

Using forward kinematics, we now have an easy way of using matricies to find the position of the end effector of a robotic arm. Since the purpose of robot arms is to move around, it would seem logical that we find expressions for the velocity, angular velocity, and linear force at the end effector.

The most intuitive approach to this problem is an recursive one. If we have a 3-joint arm, the first joint will have some velocity. The second joint's velocity will be the sum of its own velocity and the first joint's velocity, the third

joint's velocity will be the sum of its velocity and the second joint's velocity, and so on. This iterative approach also applies to torques. If we want to apply some static force on the end-effector, we can iterate down the joint links, summing torques along the way.

This approach results in a system of coupled equations with as many terms are there are joints. Systems of equations such as these are clearly suited to being expressed in matrix form. The Jacobian, in particular, is essentially a partial derivative in matrix form. For example, if we have the system of equations

$$\begin{aligned} y_1 &= f_1(x_1, x_2, x_3) \\ y_2 &= f_2(x_1, x_2, x_3) \\ y_3 &= f_3(x_1, x_2, x_3) \end{aligned} \tag{11}$$

we can re-write it in terms of vectors and matricies as in Chapter 5.7 of [Craig].

$$\mathbf{y} = F\mathbf{x} \tag{12}$$

After partial differentiation, we have

$$\begin{aligned} \delta y_1 &= \tfrac{\delta f_1}{\delta x_1}\delta x_1 + \tfrac{\delta f_1}{\delta x_2}\delta x_2 + \tfrac{\delta f_1}{\delta x_3}\delta x_3 \\ \delta y_2 &= \tfrac{\delta f_2}{\delta x_1}\delta x_1 + \tfrac{\delta f_2}{\delta x_2}\delta x_2 + \tfrac{\delta f_2}{\delta x_3}\delta x_3 \\ \delta y_3 &= \tfrac{\delta f_3}{\delta x_1}\delta x_1 + \tfrac{\delta f_3}{\delta x_2}\delta x_2 + \tfrac{\delta f_3}{\delta x_3}\delta x_3 \end{aligned} \tag{13}$$

From Chapter 5.7 of [Craig], we can write this in terms of matrcies and vectors.

$$\delta\mathbf{y} = \frac{\delta F}{\delta \mathbf{x}}\delta\mathbf{x} \tag{14}$$

$$\delta\mathbf{y} = J(\mathbf{x})\delta\mathbf{x} \tag{15}$$

Using the Jacobian experssion, equation 15, we can express things like time derivatives. If we divide both sides by δt, we have $\frac{\delta \mathbf{y}}{\delta t} = J(\mathbf{x})\frac{\delta \mathbf{x}}{\delta t}$. We see from Chapter 5.7 of [Craig] that we can treat the Jacobian, $J(\mathbf{x})$, as a mapping of velocties in $\mathbf{x}$ to velocities in $\mathbf{y}$.

In the case of static end-effector forces and joint torques, we can apply the same recursive algorithm as before. In order to find all of the forces and torques, we lock all of the joints in place, effectively freezing time. Finding the forces on each section of the arm is trivial. If we define the force on the end effector as ${}^n\mathbf{f}_n$, where n is the frame of the end effector, and $n-1$ is the frame of the joint preceding it, we know

$${}^{n-1}\mathbf{f}_{n-1} = {}^{n-1}_{n}R\,{}^n\mathbf{f}_n \tag{16}$$

Essentially, equation 16 rotates the force vector from the frame of the end effector to some desired frame further "down" the robot. The torques for each joint require slighltly more computation. From a rudimentary knowledge of vector calculus and physics, we know that $\vec{\tau} = \vec{r} \times \vec{f}$, which is easily verifiable from a physics textbook such as [Wolfson and Pasachoff]. Looking at Chapter 5.9 of [Craig], we see that we can express this in terms of joint torques $\mathbf{t}$, joint links (lever arms) $\mathbf{r}$, and forces $\mathbf{f}$, we have the torque balance expression for joint number k where $0 \leq k \leq n$.

$${}^k\mathbf{t}_k = {}^k\mathbf{t}_{k+1} + {}^k\mathbf{r}_{k+1} \times {}^k\mathbf{f}_{k+1} \tag{17}$$

In other words, the torque about joint k is equal to the torque about joint $k+1$ expressed in the frame of k plus the torque about joint $k+1$ due to the force $\mathbf{f}$. Re-writing equation 17 with a rotation matrix and understanding that ${}^k\mathbf{f}_k = {}^k\mathbf{f}_{k+1}$ from equation 16, we have

$${}^k\mathbf{t}_k = {}_{k+1}{}^kR^{k+1}\mathbf{t}_{k+1} + {}^k\mathbf{r}_{k+1} \times {}^k\mathbf{f}_k \tag{18}$$

3.2 Virtual Work

From our overview of the Jacobian, the dynamics we covered in Section 3.1, our knowledge of basic physics, and Chapter 5.10 of [Craig], we know that if a force or torque acts through a displacement, work is done. Due to conservation of energy, work done by a linear force on a system will be the same as the work done by the equivalent torque on the system. In order to perform our torque and force balances in Section 3.1, we fixed the system in place. As such, we can write the equation below, where $\mathbf{f}$, $\mathbf{x}$, τ, and Θ are vectors.

$$\mathbf{f} \cdot \delta\mathbf{x} = \tau \cdot \delta\Theta \tag{19}$$

Here we see that the rectangular work differential is equal to the rotational work differential. By the definition of the dot product, equation 15, and some matrix algebra, we can re-write equation 19 as the following

$$\mathbf{f}^T J \delta\theta = \tau^T \delta\Theta \tag{20}$$

$$\mathbf{f}^T J = \tau^T \tag{21}$$

$$\tau = J^T \mathbf{f} \tag{22}$$

If the Jacobian in equation 22 is written in a particular frame, it will translate forces to torques in that frame.

3.3 1 Degree of Freedom

With our newfound knowledge, we can apply the jacobian to the one degree of freedom (DoF) arm we discussed in section 2.2.

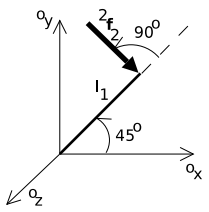

Figure 1: 1-DoF arm with a desired force ${}^2\mathbf{f}_2$ exerted at a 90° angle to the end effector in the frame of the end-effector (frame 2).

From 1, we can express ${}^2\mathbf{f}_2$ as

$${}^2\mathbf{f}_2 = \begin{bmatrix} 0 \\ f_2 \\ 0 \end{bmatrix} \tag{23}$$

and ${}^0\mathbf{f}_2$ as

$${}^0\mathbf{f}_2 = {}^0_1R^1_2R\,{}^2\mathbf{f}_2 = \begin{bmatrix} \cos\theta_0 & -\sin\theta_0 & 0 \\ \sin\theta_0 & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ f_2 \\ 0 \end{bmatrix} = \begin{bmatrix} -f_2 \sin\theta_0 \\ f_2 \cos\theta_0 \\ 0 \end{bmatrix} \tag{24}$$

Obtaining the Jacobian is slightly more difficult. Remembering from equation 15 that the Jacobian can relate velocities, if we use forward kinematics to solve for the velocity of the end-effector, we can easily obtain the Jacobian as we described in Section 3.1. We will use the vectors ω and ν to denote angular velocity and linear velocity, respectively.

$$
{}^1\omega_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \dot{\theta}_0 \quad {}^1\nu_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}
$$

$$
{}^2\omega_2 = {}^1\omega_1 \qquad {}^2\nu_2 = {}^2_1R({}^1\nu_1 + {}^1\omega_1 \times {}^1\mathbf{o}_2) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \left(\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \dot{\theta}_0 \times \begin{bmatrix} \ell_1 \\ 0 \\ 0 \end{bmatrix} \right) = \begin{bmatrix} 0 \\ \ell_1 \\ 0 \end{bmatrix} \dot{\theta}_0 \tag{25}
$$

Our result for ${}^2\nu_2$ passes a quick sanity test, since the linear velocity is $v = \omega r$, and since we chose frame 2 as our coordinate system. Therefore, ${}^2\nu_2$ is has the magnitude of $\omega r = \dot{\theta}_0 \ell_1$ in the y direction.

We can re-write ${}^2\nu_2$ as ${}^2\dot{\mathbf{o}}_2$, which makes our ${}^2\nu_2$ expression from 25 capable of being expressed as

$$
{}^2\dot{\mathbf{o}}_2 = {}^2J^2\dot{\theta}_0 \tag{26}
$$

Thus, the Jacobian in frame 2 is the folowing:

$$
{}^2J = \begin{bmatrix} 0 \\ \ell_1 \\ 0 \end{bmatrix} \tag{27}
$$

We can also express the Jacobian in terms of the universal frame, or frame 0.

$$
{}^0_1R^0J = {}^0_1R^1_2R^2J = \begin{bmatrix} \cos\theta_0 & -\sin\theta_0 & 0 \\ \sin\theta_0 & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ \ell_1 \\ 0 \end{bmatrix} = \begin{bmatrix} -\ell_1 \sin\theta_0 \\ \ell_1 \cos\theta_0 \\ 0 \end{bmatrix} \tag{28}
$$

Again, we perform a quick sanity check. Assume $\theta_0 = 45^\circ$ and $\dot{\theta} = \dot{\theta}_0$.

$$
{}^0\nu_2 = {}^0J^0\dot{\theta}_0 = \begin{bmatrix} -\ell_1 \sin 45^\circ \\ \ell_1 \cos 45^\circ \\ 0 \end{bmatrix} \dot{\theta}_0 = \begin{bmatrix} -\dot{\theta}_0 \frac{\ell_1\sqrt{2}}{2} \\ \dot{\theta}_0 \frac{\ell_1\sqrt{2}}{2} \\ 0 \end{bmatrix} \tag{29}
$$

In other words, if the link is currently 45° from the x-axis of frame 0 and has angular velocity $\dot{\theta}_0$, we know that $v = \omega r$, so $v = \dot{\theta}_0 \ell_1$, where v is perpendicular to the vector that describes ℓ_1. This means ${}^2\nu_2 = \begin{bmatrix} 0 & \ell_1\dot{\theta}_0 & 0 \end{bmatrix}^T$, which we verify from equation 25. Converting to frame zero, we use the rotation matricies ${}^1_2R$ and ${}^0_1R$,

$$
\begin{aligned}
{}^0\nu_2 = {}^0_1R^1_2R^2\nu_2 &= \begin{bmatrix} \cos\theta_0 & -\sin\theta_0 & 0 \\ \sin\theta_0 & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ \ell_1\dot{\theta}_0 \\ 0 \end{bmatrix} \\
{}^0\nu_2 &= \begin{bmatrix} -\ell_1 \sin\theta_0\dot{\theta}_0 \\ \ell_1 \cos\theta_0\dot{\theta}_0 \\ 0 \end{bmatrix} = \begin{bmatrix} -\ell_1 \sin 45^\circ\dot{\theta}_0 \\ \ell_1 \cos 45^\circ\dot{\theta}_0 \\ 0 \end{bmatrix} = \begin{bmatrix} -\dot{\theta}_0 \frac{\ell_1\sqrt{2}}{2} \\ \dot{\theta}_0 \frac{\ell_1\sqrt{2}}{2} \\ 0 \end{bmatrix}
\end{aligned} \tag{30}
$$

We can see that equation 30 yields the same result as 29, so our Jacobian expression for the 1-DoF case is correct.

We can now take the transpose of our Jacobian to obtain an expression for the torques about all of the joints, using equations 24 and 28.

$$
{}^0J^T = \begin{bmatrix} -\ell_1 \sin\theta_0 \\ \ell_1 \cos\theta_0 \\ 0 \end{bmatrix}^T = \begin{bmatrix} -\ell_1 \sin\theta_0 & \ell_1 \cos\theta_0 & 0 \end{bmatrix} \tag{31}
$$

$$
{}^0\tau = {}^0J^T{}^0\mathbf{f} = \begin{bmatrix} -\ell_1 \sin\theta_0 & \ell_1 \cos\theta_0 & 0 \end{bmatrix} \begin{bmatrix} -f_2 \sin\theta_0 \\ f_2 \cos\theta_0 \\ 0 \end{bmatrix} = f_2\ell_1(\sin^2\theta_0 + \cos^2\theta_0) = f_2\ell_1 \tag{32}
$$

In this specific case, the torque is a 1×1 vector, which represents the torque about the first joint. From our knowledge of physics, we can clearly see that this is true. If a force of magnitude f_2 is desired to be exerted perpendicular to the joint $\ell_1$at the end effector, the motor at the joint will have to exert a $\tau = fr\sin\theta = f_2\ell_1 \sin 90° = f_2\ell_1$.

4 3 Degrees of Freedom

4.1 Overview

Now that we have seen simple examples for forward kinematics and the torque-force relation given to us by the Jacobian, we can examine more complicated cases.

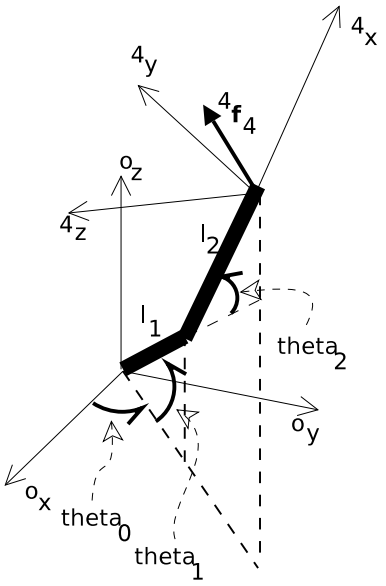

Figure 2: Setup for 3-DoF Arm

Figure 2 shows a 3-DoF arm, with the bottommost joint having 2-DoF. For the sake of simplicity, we can model that joint as a two 1-DoF joints with their axes of rotation orthogonal to one another. The first 1-DoF joint, with the z-axis as its axis of rotation, has a link of length 0, and the second joint, with the y-axis as its axis of rotation, has a link of length ℓ_1. The third joint has the y-axis as its axis of rotation, and has a link of length ℓ_2. One quick notation note is that the subscripts of the θ angles are always one less than the number of the joint they are rotating.
Additionally, we want the arm to exert a force ${}^4\mathbf{f}_4$ in the frame of the end-effector.

$$ {}^4\mathbf{f}_4 = \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} \tag{33} $$

In order to describe this arm, we will begin by deriving all of the forward kinematics equations for the end effector, which will involve finding the rotation and translation matricies. Using those matricies, we can derive the Jacobian and therefore derive the torques of each joint.

4.2 Forward Kinematics

For readability, we will denote $\cos\theta_k$ as c_k, and $\sin\theta_k$ as s_k

$$ {}^0_1R = \begin{bmatrix} c_0 & -s_0 & 0 \\ s_0 & c_0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad {}^0_1T = \begin{bmatrix} c_0 & -s_0 & 0 & 0 \\ s_0 & c_0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{34} $$

$$ {}^1_2R = \begin{bmatrix} c_1 & 0 & -s_1 \\ 0 & 1 & 0 \\ s_1 & 0 & c_1 \end{bmatrix} \quad {}^1_2T = \begin{bmatrix} c_1 & 0 & -s_1 & 0 \\ 0 & 1 & 0 & 0 \\ s_1 & 0 & c_1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{35} $$

$$ {}^2_3R = \begin{bmatrix} c_2 & 0 & -s_2 \\ 0 & 1 & 0 \\ s_2 & 0 & c_2 \end{bmatrix} \quad {}^2_3T = \begin{bmatrix} c_2 & 0 & -s_2 & \ell_1 \\ 0 & 1 & 0 & 0 \\ s_2 & 0 & c_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{36} $$

$$ {}^3_4R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad {}^3_4T = \begin{bmatrix} 1 & 0 & 0 & \ell_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{37} $$

$$ {}^0_4R = {}^0_1R^1_2R^3_2R^3_4R = \begin{bmatrix} c_0c_1c_2 - c_0s_1s_2 & -s_0 & -c_0c_1s_2 - c_0c_2s_1 \\ c_1c_2s_0 - s_0s_1s_2 & c_0 & -c_1s_0s_2 - c_2s_0s_1 \\ c_2s_1 + c_1s_2 & 0 & c_1c_2 - s_1s_2 \end{bmatrix} \tag{38} $$

$$ {}^0_4T = {}^0_1T^1_2T^3_2T^3_4T = \begin{bmatrix} c_0c_1c_2 - c_0s_1s_2 & -s_0 & -c_0c_1s_2 - c_0c_2s_1 & \ell_2(c_0c_1c_2 - c_0s_1s_2) + \ell_1c_0c_1 \\ c_1c_2s_0 - s_0s_1s_2 & c_0 & -c_1s_0s_2 - c_2s_0s_1 & \ell_2(c_1c_2s_0 - s_0s_1s_2) + \ell_1c_1s_0 \\ c_2s_1 + c_1s_2 & 0 & c_1c_2 - s_1s_2 & \ell_2(c_2s_1 + c_1s_2) + \ell_1s_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{39} $$

The position of the end effector is given by

$$ {}^0\mathbf{o}_4 = {}^0_4T^4\mathbf{o}_4 \tag{40} $$

$$ {}^0\mathbf{o}_4 = {}^0_4T \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \ell_2(c_0c_1c_2 - c_0s_1s_2) + \ell_1c_0c_1 \\ \ell_2(c_1c_2s_0 + s_0s_1s_2) + \ell_1c_1s_0 \\ \ell_2(c_2s_1 + c_1s_2) + \ell_1s_1 \\ 1 \end{bmatrix} \tag{41} $$

For a quick sanity check, let $\theta_0 = \theta_1 = \theta_2 = 0°$, i.e. the arm lies entirely along the x-axis. Ignoring the last row, which is an artifact of the translation matrix, we have ${}^0\mathbf{o}_4 = \begin{bmatrix} \ell_2 + \ell_1 & 0 & 0 \end{bmatrix}^T$, which is correct.

If we set $\theta_0 = 90°$ and $\theta_1 = \theta_2 = 0°$, i.e. the arm lies entirely along the y-axis, we have ${}^0\mathbf{o}_4 = \begin{bmatrix} 0 & \ell_2 + \ell_1 & 0 \end{bmatrix}^T$.

Now we set $\theta_0 = \theta_2 = 0°$ and $\theta_1 = 90°$, i.e. the arm is along the z-axis. These parameters give us ${}^0\mathbf{o}_4 = \begin{bmatrix} 0 & 0 & \ell_2 + \ell_1 \end{bmatrix}^T$.

If we tried more values of θ, we would see that our matrix for ${}^0_4T$ is correct. The ideal sanity check of course would be to orient the arm parallel to $\begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^T$. Indeed, if we try $\theta_0 = 45°$, $\theta_1 = \arctan\frac{1}{\sqrt{2}}$, and $\theta_2 = 0°$, our numerically-computed vector is ${}^0\mathbf{o}_4 = \begin{bmatrix} 0.5774\ell_2 + 0.5774\ell_1 & 0.5774\ell_2 + 0.5774\ell_1 & 0.5774\ell_2 + 0.5774\ell_1 \end{bmatrix}^T$, which is most definitley parallel to $\begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^T$.

4.3 Jacobian

Now that we have all of our forward kinematics expressions, we can use them to derive the Jacobian.

$$ {}^1\omega_1 = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_0 \end{bmatrix} \quad {}^1\nu_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{42} $$

$$ {}^2\omega_2 = \begin{bmatrix} 0 \\ \dot{\theta}_1 \\ \dot{\theta}_0 \end{bmatrix} \quad {}^2\nu_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{43} $$

We know that ${}^3_2R = ({}^3_2R)^T$, so using equation 36 we have

$$ {}^3\omega_3 = \begin{bmatrix} 0 \\ \dot{\theta}_1 + \dot{\theta}_2 \\ \dot{\theta}_0 \end{bmatrix} \quad {}^3\nu_3 = {}^3_2R({}^2\nu_2 + {}^2\omega_2 \times {}^2\mathbf{o}_3) = \begin{bmatrix} c_2 & 0 & -s_2 \\ 0 & 1 & 0 \\ s_2 & 0 & c_2 \end{bmatrix}^T \begin{bmatrix} 0 \\ \dot{\theta}_1 \\ \dot{\theta}_0 \end{bmatrix} \times \begin{bmatrix} \ell_1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -\ell_1s_2\dot{\theta}_1 \\ \ell_1\dot{\theta}_0 \\ -\ell_1c_2\dot{\theta}_1 \end{bmatrix} \tag{44} $$

$$
{}^4\omega_4 = \begin{bmatrix} 0 \\ \dot{\theta}_1 + \dot{\theta}_2 \\ \dot{\theta}_0 \end{bmatrix} \quad {}^4\nu_4 = {}^4_3R({}^3\nu_3 + {}^3\omega_3 \times {}^3\mathbf{o}_4) = \begin{bmatrix} -\ell_1 s_2 \dot{\theta}_1 \\ \ell_1 \dot{\theta}_0 \\ -\ell_1 c_2 \dot{\theta}_1 \end{bmatrix} + \begin{bmatrix} 0 \\ \dot{\theta}_1 + \dot{\theta}_2 \\ \dot{\theta}_0 \end{bmatrix} \times \begin{bmatrix} \ell_2 \\ 0 \\ 0 \end{bmatrix}
$$

$$
{}^4\nu_4 = \begin{bmatrix} -\ell_1 s_2 \dot{\theta}_1 \\ \ell_1 \dot{\theta}_0 + \ell_2 \dot{\theta}_0 \\ -\ell_1 c_2 \dot{\theta}_1 + \ell_2(\dot{\theta}_1 + \dot{\theta}_2) \end{bmatrix} \tag{45}
$$

From equation 45, we can derive the Jacobian.

$$
\delta\mathbf{o} = {}^4J\delta\theta = \begin{bmatrix} 0 & -\ell_1 s_2 & 0 \\ \ell_1 + \ell_2 & 0 & 0 \\ 0 & -\ell_1 c_2 + \ell_2 & \ell_2 \end{bmatrix} \begin{bmatrix} \dot{\theta}_0 \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}
$$

$$
{}^4J = \begin{bmatrix} 0 & -\ell_1 s_2 & 0 \\ \ell_1 + \ell_2 & 0 & 0 \\ 0 & -\ell_1 c_2 + \ell_2 & \ell_2 \end{bmatrix} \tag{46}
$$

Now, we express the Jacobian in reference frame 0, keeping in mind that ${}^0J = {}^0_4R{}^4J$

$$
{}^0J = \begin{bmatrix} -s_0(\ell_1 + \ell_2) & -\ell_1 s_2(c_0 c_1 c_2 - c_0 s_1 s_2) - (\ell_1 c_2 + \ell_2)(c_0 c_1 s_2 + c_0 c_2 s_1) & -\ell_2(c_0 c_1 s_2 + c_0 c_2 s_1) \\ c_0(\ell_1 + \ell_2) & -\ell_1 s_2(c_1 c_2 s_0 - s_0 s_1 s_2) - (\ell_1 c_2 + \ell_2)(c_1 s_0 s_2 + c_2 s_0 s_1) & -\ell_2(c_1 s_0 s_2 + c_2 s_0 s_1) \\ 0 & -\ell_1 s_2(c_2 s_1 + c_1 s_2) + (\ell_1 c_2 + \ell_2)(c_1 c_2 - s_1 s_2) & \ell_2(c_1 c_2 - s_1 s_2) \end{bmatrix} \tag{47}
$$

Again, a very quick sanity check. Assume $\theta_0 = \theta_1 = \theta_2 = 0°$. ${}^0\nu_4 = {}^0J\dot{\theta}$. Given our choices for θ_k,

$$
{}^0J = \begin{bmatrix} 0 & 0 & 0 \\ \ell_1 + \ell_2 & 0 & 0 \\ 0 & \ell_1 + \ell_2 & \ell_2 \end{bmatrix}
$$

As such,

$$
{}^0\nu_4 = {}^0J\dot{\theta} = \begin{bmatrix} 0 & 0 & 0 \\ \ell_1 + \ell_2 & 0 & 0 \\ 0 & \ell_1 + \ell_2 & \ell_2 \end{bmatrix} \begin{bmatrix} \dot{\theta}_0 \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ (\ell_1 + \ell_2)\dot{\theta}_0 \\ (\ell_1 + \ell_2)\dot{\theta}_1 + \ell_2\dot{\theta}_2 \end{bmatrix}
$$

From our knowledge of simple physics, $v = \omega r$, and a little algebra, we can see that the above expression for ${}^0\nu_4$ is true. Our Jacobian is correct. Using our Jacobian and equation 22, we can easily obtain the torques of each joint required to exert the force ${}^4\mathbf{f}_4$. For readibility, we have kept the matricies and vectors in their unexpanded form.

$$
\tau = \left({}^0J\right)^T {}^0_4R{}^4\mathbf{f}_4 \tag{48}
$$

Using MATLAB, we can find a vector expression for τ in terms of $\sin\theta_k$, $\cos\theta_k$, ℓ_1, ℓ_2, f_x, f_y, and f_z. The raw form of the expression is extremely unwieldly and needs simplification. We have included it in Appendix A for your conveience.

5 Conclusion

The power of linear algebra has enabled us to use forward kinematics to determine instantaneous position simply by knowing the geometry of a manipulator and its current angles, as well as translating desired forces to needed torques. As we can clearly see from our work above, the Jacobian is a powerful tool. Not only can it in its normal form act as a sort of partial derivative in matrix form, it can also translate forces to torques in a very simple matrix operation. Roboticists typically use a time-variant Jacobian, which continously updates the θ_k values in its entries, to calculate torques from forces at any given instant. This real-time translation from desired forces to needed torques is essential for many types of robot controls systems. Since the underlying mathematics of the Jacobian are relatively simple, we can apply it over and over again in our control systems to get a very quick and effective way of force control.

Appendix A - Torque Expression for 3-DoF Arm

τ_1

```
(-s0*(l1+l2)*(c0*c1*c2-c0*s1*s2)+c0*(l1+l2)*(s0*c1*c2-s0*s1*s2))*fx+(s0^2*(l1+l2)+c0^2*(l1+l2))*fy+(-s0*(l1+l2)*(-c0*c1*s2-c0*s1*c2)+c0*(l1+l2)*(-s0*c1*s2-s0*s1*c2))*fz
```

τ_2

```
((-(c0*c1*c2-c0*s1*s2)*l1*s2+(-c0*c1*s2-c0*s1*c2)*(l1*c2+l2))*(c0*c1*c2-c0*s1*s2)+(-(s0*c1*c2-s0*s1*s2)*l1*s2+(-s0*c1*s2-s0*s1*c2)*(l1*c2+l2))*(s0*c1*c2-s0*s1*s2)+
     (-(s1*c2+c1*s2)*l1*s2+(-s1*s2+c1*c2)*(l1*c2+l2))*(s1*c2+c1*s2))*fx+(-(-(c0*c1*c2-c0*s1*s2)*l1*s2+(-c0*c1*s2-c0*s1*c2)*(l1*c2+l2))*s0+(-(s0*c1*c2-s0*s1*s2)*l1*s2+
     (-s0*c1*s2-s0*s1*c2)*(l1*c2+l2))*c0)*fy+((-(c0*c1*c2-c0*s1*s2)*l1*s2+(-c0*c1*s2-c0*s1*c2)*(l1*c2+l2))*(-c0*c1*s2-c0*s1*c2)+(-(s0*c1*c2-s0*s1*s2)*l1*s2+
     (-s0*c1*s2-s0*s1*c2)*(l1*c2+l2))*(-s0*c1*s2-s0*s1*c2)+(-(s1*c2+c1*s2)*l1*s2+(-s1*s2+c1*c2)*(l1*c2+l2))*(-s1*s2+c1*c2))*fz
```

τ_3

```
((-c0*c1*s2-c0*s1*c2)*l2*(c0*c1*c2-c0*s1*s2)+(-s0*c1*s2-s0*s1*c2)*l2*(s0*c1*c2-s0*s1*s2)+(-s1*s2+c1*c2)*l2*(s1*c2+c1*s2))*fx+
     (-(-c0*c1*s2-c0*s1*c2)*l2*s0+(-s0*c1*s2-s0*s1*c2)*l2*c0)*fy+((-c0*c1*s2-c0*s1*c2)^2*l2+(-s0*c1*s2-s0*s1*c2)^2*l2+(-s1*s2+c1*c2)^2*l2)*fz
```

Appendix B - Acknowledgements

General Help

- Professor Gill Pratt for answering citation technique questions.
- Jon Chambers for answering a quick coordinate systems question.

Proofreaders

- Elizabeth Kneen

References

[Craig] Craig, John J. *Introduction to Robotics: Mechanics and Control.* 3rd ed. Upper Saddle River:Pearson Prentice Hall, 2005

[Wolfson and Pasachoff] Wolfson, Richard and Jay M. Pasachoff. *Physics for Scientists and Engineers.* 3rd ed. Reading: Addison-Wesley, 1999.

[Stewart] Stewart, James. *Calculus.* 5th ed. Belmont: Thomson Brooks/Cole, 2003.