Manipulator Kinematics

Frank Dellaert
Center for Robotics and Intelligent Machines
Georgia Institute of Technology

February 6, 2020

Contents

1 Introduction

After an introduction to the necessary concepts from geometry, I start with simple planar manipulators with only revolute joints. This allows us to develop intuition for how forward kinematics can be implemented by concatenating a chain of rigid transforms. I then show how using exponential maps of differential twists, while an advanced concept, make describing robot arms more convenient. After that, it is a simple matter to generalize everything to 3D and include prismatic joints.

Please also see the texts by Murray, Li, and Sastry [3]and Lynch and Park [**?**].

2 Planar Geometry

We start off by defining and giving some intuitions for geometry in the plane.

2.1 Planar Rotations aka SO(2)

I spend a lot of time on 2D rotations below, but bear with it! It is the simplest space to think in, and we can get the most essential concepts across, which will then generalize elegantly to the other transformation groups.

Basic Facts

A point P^b in a rotated coordinate frame B can be expressed in a reference frame S by multiplying with a 2×2 orthonormal **rotation matrix** R^s_b,

$$p^s = R^s_b p^b$$

where the indices on R^s_b indicate the source and destination frames. The set of 2×2 rotation matrices, with matrix multiplication as the composition operator, form a *commutative* group called the **Special Orthogonal group** $SO(2)$. This means it satisfies the following properties, for all $R, R_1, R_2, R_3 \in SO(2)$:

1. Closed: $R_1 R_2 \in SO(2)$.
2. Identity: $I_2 R = R = R I_2$ with I_2 the 2×2 identity matrix.
3. Inverse: $R R^{-1} = I_2 = R^{-1} R$. Note that $R^{-1} = R^T$.
4. Associativity: $(R_1 R_2) R_3 = R_1 (R_2 R_3)$.
5. Commutativity: $R_1 R_2 = R_2 R_1$.

In fact, the 2D rotation matrices form a 1-dimensional subgroup of the general linear group of 2×2 invertible matrices $GL(2)$, i.e., $SO(2) \subset GL(2)$. Informally, $SO(2)$ is also a **manifold**, because it is a smooth continuous subset of the 2×2 matrices. We say it is a 1-D manifold in the 4-dimensional ambient space of 2×2 matrices. It is clear why this is so: the matrix is constrained to be orthonormal, which provides three non-redundant constraints on its four entries.

However, this is not quite enough to fully define the manifold, as rotations composed with a reflection also satisfy those constraints: together, they form the orthogonal group O(2). The "S" in SO(2) picks one of the two connected components of the O(2) manifold, namely those orthonormal matrices that have unit determinant, i.e., $|R| = 1$.

Finally, since $SO(2)$ is *both* a group and a manifold, we call it - informally, again - a 1-dimensional **Lie group**. An essential property of Lie groups is that the group operation is smooth, i.e., a small change to either R_1 or R_2 yields a small change in their product $R_1 R_2$. Similarly, the group action of $SO(2)$ on a 2D point $p \in \mathbb{R}^2$ is also smooth: a small change in R will yield a small change in Rp.

Intuitions

Loosely following [3, p. 51], let us consider the curve $R_b^s(t)$ describing the trajectory of a rotating body B in the base frame S, the **spatial coordinate frame**. Hence, a point P^b in the **body coordinate frame** B describes the following trajectory in spatial coordinates:

$$p^s(t) = R_b^s(t) p^b$$

Something that we intuitively know is that the point $p^s(t)$ describes a circular trajectory around the origin. In addition, the further from the origin the faster the point moves, and the **velocity vector** $v^s(t)$ is always orthogonal to the vector $p^s(t)$. Indeed, I show below that the velocity vector is given by

$$v^s(t) = \omega(t) \left[p^s(t)\right]^{\perp} \tag{2.1}$$

where I introduce the notation $p^{\perp}$ to mean the orthogonalization of a vector, i.e.,

$$\begin{bmatrix} x \\ y \end{bmatrix}^{\perp} \triangleq \begin{bmatrix} -y \\ x \end{bmatrix},$$

and $\omega(t)$ is the 1-dimensional, time-varying **angular velocity**. The time-function $\omega(t)$ completely describes the motion of the rotating body, and (2.1) does capture our intuition about the instantaneous velocity of a point on a circular trajectory.

Proof. The velocity of the point $p^s(t)$ at time t in the spatial coordinate frame is

$$v^s(t) = \frac{d}{dt}p^s(t) = \dot{R}^s_b(t)p^b$$

We know that we want to end up with an expression involving $p^s(t)$, so let us substitute the fixed p^b by the time varying quantity

$$p^b = (R^s_b(t))^{-1}\, p^s(t)$$

which yields a map from spatial coordinates $p^s(t)$ to spatial velocity:

$$v^s(t) = \dot{R}^s_b(t)\,(R^s_b(t))^{-1}\, p^s(t)$$

One can easily prove that the matrix

$$\hat{\omega}(t) \triangleq \dot{R}^s_b(t)\,(R^s_b(t))^{-1} \tag{2.2}$$

is skew-symmetric, and hence can be written as

$$\hat{\omega}(t) \triangleq \begin{bmatrix} 0 & -\omega(t) \\ \omega(t) & 0 \end{bmatrix} = \omega(t) \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix},$$

where we defined the **hat operator** $\wedge$ that maps a scalar to a 2×2 skew-symmetric matrix. Finally, note that

$$\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} -y \\ x \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix}^{\perp},$$

and hence

$$v^s(t) = \hat{\omega}(t)p^s(t) = \omega(t)\,[p^s(t)]^{\perp} \tag{2.3}$$

□

The skew-symmetric matrices of the form

$$\hat{\omega} \triangleq \begin{bmatrix} 0 & -\omega \\ \omega & 0 \end{bmatrix}$$

form a vector space and are elements of the **Lie algebra** $\mathfrak{so}(2)$ associated with $SO(2)$, and hence can be added together, multiplied with a scalar, etc. To extract the angular velocity from an $\mathfrak{so}(2)$ element we can define the **vee operator**:

$$\begin{bmatrix} 0 & -\omega \\ \omega & 0 \end{bmatrix}^{\vee} \triangleq \omega$$

Hence, $\mathfrak{so}(2)$ is isomorphic to the vector space $\mathbb{R}$.

The Exponential Map

Can we get a closed form solution for $p^s(t)$ if the angular velocity ω is constant? Again following [3, p. 27], from the definition of $v^s(t)$ and Equation 2.3, we get the following expression in the spatial coordinates $p^s(t)$:

$$\frac{d}{dt}p^s(t) = \hat{\omega}p^s(t).$$

This is a time-invariant differential equation, yielding the solution

$$p^s(t) = \exp(\hat{\omega}t)p^s(0)$$

where $\exp(\hat{\omega}t)$ is the **matrix exponential**, defined by the familiar series

$$\exp(\hat{\theta}) \stackrel{\Delta}{=} I + \hat{\theta} + \frac{\hat{\theta}^2}{2!} + \frac{\hat{\theta}^3}{3!} + \dots$$

Now, if we define $\theta \stackrel{\Delta}{=} \omega t$, because

$$\hat{\theta}^2 = \begin{bmatrix} 0 & -\theta \\ \theta & 0 \end{bmatrix} \begin{bmatrix} 0 & -\theta \\ \theta & 0 \end{bmatrix} = \theta^2 \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$\hat{\theta}^3 = \theta^2 \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 0 & -\theta \\ \theta & 0 \end{bmatrix} = \theta^3 \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

etc., we can recognize the even and odd series for $\cos$ and $\sin$, respectively, yielding

$$\exp(\hat{\omega}t) \stackrel{\Delta}{=} \begin{bmatrix} I - \frac{\theta^2}{2!} + \dots & -\theta + \frac{\theta^3}{3!} \dots \\ \theta - \frac{\theta^3}{3!} \dots & I - \frac{\theta^2}{2!} + \dots \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

Hence, to no great surprise, we find that the spatial coordinates trace out a circle:

$$p^s(t) = \begin{bmatrix} \cos\omega t & -\sin\omega t \\ \sin\omega t & \cos\omega t \end{bmatrix} p^s(0)$$

2.2 2D Rigid Transforms aka SE(2)

Basic Facts

In robot manipulators, the pose of the end-effector or tool is computed by concatenating several rigid transforms. Let p^b be the 2D coordinates of a point in the frame

B, and p^s the coordinates of the same point in a base frame S. We can transform p^b to p^s by a **2D rigid transform** T_b^s, which is a rotation followed by a 2D translation,

$$p^s = T_b^s \otimes p^b \triangleq R_b^s p^b + t_b^s$$

with $T_b^s \triangleq (R_b^s, t_b^s)$, where $R_b^s \in SO(2)$ and $t_b^s \in \mathbb{R}^2$. In all of the above, a subscript B indicates the frame we are transforming *from*, and the superscript S indicates the frame we are transforming *to*. In well-formed equations, subscripts on one symbol match the superscript of the next symbol.

The set of 2D rigid transforms forms a group, where the group operation corresponding to composition of two rigid 2D transforms is defined as

$$T_s^t = T_s^t \oplus T_b^s = \left(R_s^t,\, t_s^t\right) \oplus \left(R_b^s,\, t_b^s\right) \triangleq \left(R_s^t R_b^s,\, R_s^t t_b^s + t_s^t\right) \tag{2.4}$$

The group of rotation-translation pairs T with this group operation is called the **special Euclidean group** $SE(2)$. It has an identity element $e = (I, 0)$, and the inverse of a transform $T = (R, t)$ is given by $T^{-1} = (R^T, -R^T t)$.

2D rigid transforms can be viewed as a subgroup of a general linear group of degree 3, i.e., $SE(2) \subset GL(3)$. This can be done by embedding the rotation and translation into a 3×3 invertible matrix defined as

$$T_b^s = \begin{bmatrix} R_b^s & t_b^s \\ 0 & 1 \end{bmatrix}$$

With this embedding you can verify that matrix multiplication implements composition, as in Equation 2.4:

$$T_s^t T_b^s = \begin{bmatrix} R_s^t & t_s^t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_b^s & t_b^s \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_s^t R_b^s & R_s^t t_b^s + t_s^t \\ 0 & 1 \end{bmatrix} = T_b^t$$

By similarly embedding 2D points in a three-vector, the so-called **homogeneous coordinates** of the 2D vector, a 2D rigid transform acting on a point can be implemented by matrix-vector multiplication:

$$\begin{bmatrix} R_b^s & t_b^s \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p^b \\ 1 \end{bmatrix} = \begin{bmatrix} R_b^s p^b + t_b^s \\ 1 \end{bmatrix}$$

In what follows we will work with these transform matrices exclusively.

Intuitions

The intuition for rigid transforms in the plane stems from the following not-so-obvious theorem:

Theorem. *Every 2D rigid transform can be expressed as a rotation around a point, possibly at infinity (in which case it is a pure translation).*

Consider the trajectory traced out by spatial coordinates of a point p^b on a rigidly moving body, expressed in homogeneous coordinates:

$$\begin{bmatrix} p^s(t) \\ 1 \end{bmatrix} = T^s_b(t) \begin{bmatrix} p^b \\ 1 \end{bmatrix}.$$

Then the theorem says that at any given moment the velocity vector $v^s(t)$ of the point must be consistent with a rotational trajectory around some instantaneous center of rotation (ICR) $q^s(t)$, i.e.,

$$v^s(t) = \omega(t) \left[p^s(t) - q^s(t)\right]^\perp \tag{2.5}$$

where $\omega(t)$ is again the angular velocity over time. Rather than proving this, let us rewrite this to derive an equivalent hat operator for $SE(2)$:

$$\begin{aligned} v^s(t) &= \omega(t) \left[p^s(t) - q^s(t)\right]^\perp \\ &= \hat{\omega}(t) \left[p^s(t) - q^s(t)\right] \\ &= \hat{\omega}(t)p^s(t) - \hat{\omega}(t)q^s(t) \end{aligned}$$

Recognizing that

$$v(t) \stackrel{\Delta}{=} -\hat{\omega}(t)q^s(t) = -\omega(t) \left[q^s(t)\right] \perp$$

is the instantaneous velocity of the origin, we can write

$$\begin{bmatrix} v^s(t) \\ 0 \end{bmatrix} = \begin{bmatrix} \hat{\omega}(t)p^s(t) + v(t) \\ 0 \end{bmatrix} = \begin{bmatrix} \hat{\omega}(t) & v(t) \\ 0 & 0 \end{bmatrix} \begin{bmatrix} p^s(t) \\ 1 \end{bmatrix} \tag{2.6}$$

The 3×3 matrices of the form

$$\begin{bmatrix} \hat{\omega} & v \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -\omega & v_x \\ \omega & 0 & v_y \\ 0 & 0 & 0 \end{bmatrix}$$

are elements of the Lie algebra $\mathfrak{se}(2)$ associated with $SE(2)$.

Compare (2.6) above with Equation 2.3 for rotations. The pattern that emerges is that **elements of the Lie algebra map spatial coordinates to spatial velocity**, although we have to use homogeneous coordinates for $SE(2)$. Spelled out in full, we obtain

$$\begin{bmatrix} v^s_x \\ v^s_y \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & -\omega & v_x \\ \omega & 0 & v_y \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p^s_x \\ p^s_y \\ 1 \end{bmatrix} = \begin{bmatrix} -\omega p^s_y + v_x \\ \omega p^s_x + v_y \\ 0 \end{bmatrix}$$

We define a **2D differential twist** $\mathcal{V}$ as the three-dimensional vector $\mathcal{V} \triangleq (v_x, v_y, \omega)$. The first two components make up the linear velocity $v \in \mathbb{R}^2$, and the last component is the angular velocity $\omega \in \mathbb{R}$. Hence, we also frequently write $\mathcal{V} = (v, \omega)$. This quantity can can be interpreted as the instantaneous "velocity" of the time-varying transform $T_b^s(t)$, just like angular velocity is for 2D rotations. At any given time, the instantaneous center of rotation (ICR) is given by

$$q^s(t) = [v(t)]^{\perp} / \omega(t). \tag{2.7}$$

Note that for a pure translation this point lies at infinity, in the direction orthogonal to $v(t)$.

The $SE(2)$ hat operator maps 2D differential twists to elements of $\mathfrak{se}(2)$

$$\hat{\mathcal{V}} \triangleq \begin{bmatrix} \hat{\omega} & v \\ 0 & 0 \end{bmatrix}$$

and the $SE(2)$ vee operator extracts a 2D differential twist:

$$\begin{bmatrix} \hat{\omega}(t) & v(t) \\ 0 & 0 \end{bmatrix}^{\vee} \triangleq (v, \omega) = \mathcal{V}$$

The lie algebra $\mathfrak{se}(2)$ is isomorphic to the vector space $\mathbb{R}^3$, so we can add differential twists and multiply them with scalars.

The Exponential Map

Can we get a closed form solution for $p^s(t)$ for a constant-twist trajectory? Using the exponential map as before, with a constant differential twist $\mathcal{V} = (v, \omega)$, we obtain:

$$\begin{bmatrix} p^s(t) \\ 1 \end{bmatrix} = \exp(\hat{\mathcal{V}}t) \begin{bmatrix} p^s(0) \\ 1 \end{bmatrix}$$

where $\exp(\hat{\omega}t)$ is the matrix exponential, now defined on 3×3 matrices.

We can obtain a closed-form solution for the exponential map above using some more intuition, and the fact that for a constant differential twist the ICR is also constant and equal to $q = v^{\perp}/\omega$. Hence, we can simply conjugate a rotation at the origin with a translation to q:

$$\exp\left(\hat{\mathcal{V}}t\right) = \begin{bmatrix} I & q \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R(\omega t) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & -q \\ 0 & 1 \end{bmatrix} \tag{2.8}$$

where the 2×2 rotation matrix $R(\theta)$ is given as before:

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

Multiplying through, we obtain

$$\exp\left(\hat{\mathcal{V}}t\right) = \begin{bmatrix} R(\omega t) & [I - R(\omega t)]\, v^{\perp}/\omega \\ 0 & 1 \end{bmatrix} \tag{2.9}$$

Note that for $\omega \approx 0$ we have a division by zero above. However, note that

$$\lim_{\omega \to 0} R(\omega t)/\omega = \lim_{\omega \to 0} \begin{bmatrix} \cos\omega t & -\sin\omega t \\ \sin\omega t & \cos\omega t \end{bmatrix} /\omega = \begin{bmatrix} 1 & -t \\ t & 1 \end{bmatrix}$$

and hence

$$\lim_{\omega \to 0} [I - R(\omega t)]\, v^{\perp}/\omega = \begin{bmatrix} 0 & t \\ -t & 0 \end{bmatrix} v^{\perp} = vt$$

In practice, we handle this in the code by checking for small ω and setting

$$\exp\left(\hat{\mathcal{V}}t\right) \approx \begin{bmatrix} R(\omega t) & vt \\ 0 & 1 \end{bmatrix}$$

in that case.

2.3 Exercise

A rigid body is moving in $SE(2)$ with a constant spatial velocity $\mathcal{V} = (3, 2, 1)$. What is the velocity v^s of the point $p^s = (6, 7)$?

3 Serial Link Manipulators

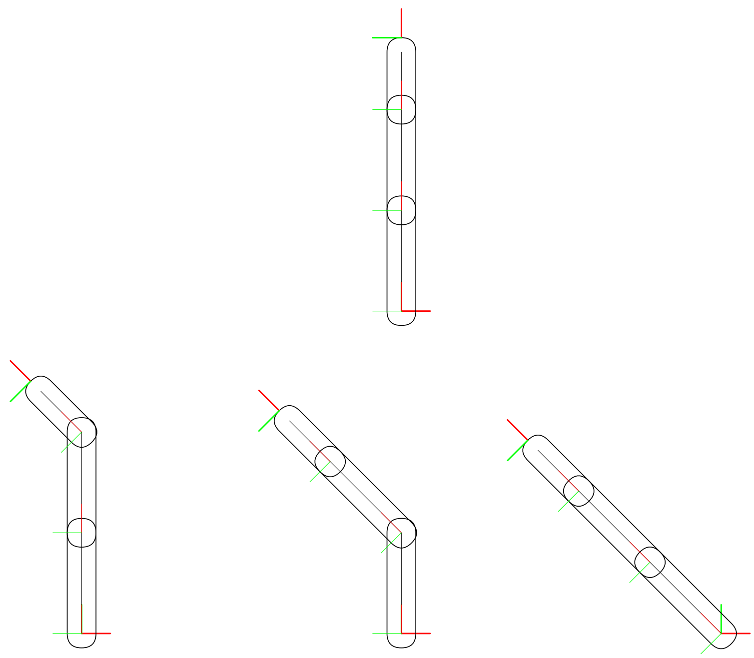

Figure 3.1: Top: the rest state of a planar RRR serial manipulator. Bottom: actuating the three degrees of freedom, respectively moving θ_3, θ_2, and θ_1.

This section describes the basic concepts, closely following [3, 1].

A **serial link manipulator** has several **links**, numbered 0 to n, connected by **joints**, numbered 1 to n. Joint j connects link $j-1$ to link j. We will only consider either a **revolute** joint with a **joint angle** θ_j, or a **prismatic** joint with a **link offset** d_j. More complex joints can be described as combinations of these basic joints.

We can treat both revolute and prismatic joins uniformly by introducing the concept of a **generalized joint coordinate** q_j, and specifying the joint type using a string, e.g., the classical Puma robot is RRRRRR, and the SCARA pick and place robot is RRRP. The vector $q \in Q$ of these generalized joint coordinates is also called the **pose** of the manipulator, where Q is called the **joint space** of the manipulator.

There are two more coordinate frames to consider: the **base frame** S, usually attached directly to link 0, and the **tool frame** T, attached to the end-effector of the robot. The tool frame T moves when the joints move.

All essential concepts can be easily developed for 2D or **planar manipulators** with revolute joints only, an example of which is shown in Figure 3.1. The top panel shows the manipulator at rest, along with five 2D coordinate frames: the base frame S, the tool frame T, and one coordinate frame for each of the three links, situated at each joint. Note that at rest, the first link is rotated by $\pi/2$. For this RRR manipulator, the generalized joint coordinates are $q = \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 \end{bmatrix}^T$, and the effect of changing each individual joint angle θ_j is shown at the bottom of the figure.

Key questions we will answer are how to determine the pose of the end-effector tool, given joint angles, and the reverse question: what joint angles should we command to get the tool to be at a desired pose? We start with the first question, which is known as forward kinematics.

3.1 Exercise

What is the configuration string of the Fetch robot, excluding the base?

4 Forward Kinematics

The problem of forward kinematics can now be stated [3]:

> Given generalized joint coordinates $q \in Q$, we wish to determine the pose $T_t^s(q)$ of the tool frame T relative to the base frame S.

We define a link coordinate frame $T_j^s(q)$ for every link j, and we define the link coordinate frame T_0^s to be identical to the base frame S. Since the tool frame T moves with link n, we have

$$T_t^s(q) = T_n^s(q) X_t^n$$

where X_t^n specifies the unchanging pose of the tool T in the frame of link n. The link coordinate frame $T_n^s(q)$ itself can be expressed recursively as

$$T_n^s(q) = T_{n-1}^s(q_1 \dots q_{n-1}) T_n^{n-1}(q_n),$$

finally yielding

$$\boxed{T_t^s(q) = T_1^s(q_1) \dots T_j^{j-1}(q_j) \dots T_n^{n-1}(q_n) X_t^n.} \tag{4.1}$$

4.1 Exercise

Draw a simple two-link RP manipulator (similar to the original Unimate robot, but in 2D) and provide the above forward kinematics formula.

5 Describing Serial Manipulators

Equation 4.1 is correct but we need to tie it to the robot's geometry. If we agree on making the link coordinate frame $T_j^s(q)$ coincide with joint axis j and fixed to link j, then the link-to-link transform $T_j^{j-1}(q_j)$ can written as

$$T_j^{j-1}(q_j) = X_j^{j-1} Z_j^j(q_j)$$

where X_j^{j-1} is a fixed transform telling us where joint j is located in the coordinate frame of the previous link, and $Z_j^j(q_j)$ represents the transform at the joint itself, parameterized by the generalized joint coordinate q_j. Substituting this into 4.1 yields

$$T_t^s(q) = X_1^s Z_1^1(q_1) \ldots X_j^{j-1} Z_j^j(q_j) \ldots X_n^{n-1} Z_n^n(q_n) X_t^n \tag{5.1}$$

For planar revolute joints, the $Z_j^j(q)$ transform can always be made of the form

$$Z_j^j(q) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.2}$$

possibly with an offset applied to the joint angle. Most often the coordinate frames are chosen such that the X_j^{j-1} transforms are as simple as possible.

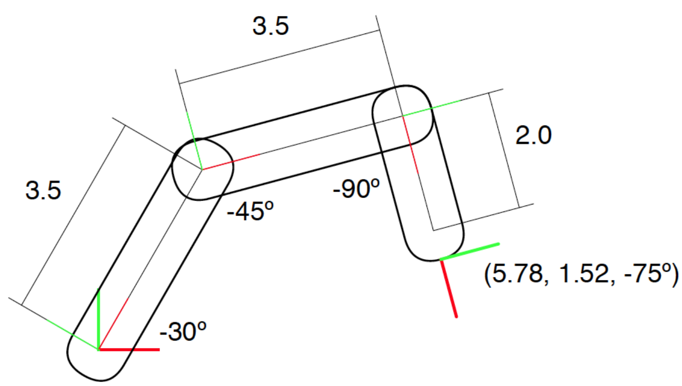

Figure 5.1: Example of simple RRR manipulator with all three joints actuated: $\theta_1 = -30^o$, $\theta_2 = -45^o$, and $\theta_3 = -90^o$, respectively.

As an example, for the simple planar RRR manipulator we have

$$T_t^s(\theta_1, \theta_2, \theta_3) = \left\{X_1^s Z_1^1(\theta_1)\right\} \left\{X_2^1 Z_2^2(\theta_j)\right\} \left\{X_3^2 Z_3^3(\theta_n)\right\} X_t^3$$

I chose the first joint coordinate frames as rotated by 90 degrees, which takes care of the joint angle offset, and chose the tool pose with respect to the third link frame as 2.5 m. along the x-axis:

$$X_1^s = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ and } X_t^3 = \begin{bmatrix} 1 & 0 & 2.5 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The next two transforms are simply translations along the link's X axis:

$$X_2^1 = \begin{bmatrix} 1 & 0 & 3.5 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ and } X_3^2 = \begin{bmatrix} 1 & 0 & 3.5 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

When multiplied out, we obtain

$$T_t^s(q) = \begin{pmatrix} -\sin\beta & -\cos\beta & -3.5\sin\theta_1 - 3.5\sin\alpha - 2.5\sin\beta \\ \cos\beta & -\sin\beta & 3.5\cos\theta_1 + 3.5\cos\alpha + 3.5\cos\beta \\ 0 & 0 & 1 \end{pmatrix} \tag{5.3}$$

with $\alpha = \theta_1 + \theta_2$ and $\beta = \theta_1 + \theta_2 + \theta_3$, the latter being the tool orientation with respect to rest.

5.1 Exercise

Provide the multiplied-out forward kinematics formula for your RP manipulator.

6 Product of Exponentials

The above exposition is cumbersome in that it involves a lot of intermediate coordinate frames. Murray et. al. [3] developed a different approach that only involves two coordinate frames: the base frame S and the tool frame T. In addition, it is very easy to determine the parameters corresponding to each joint. The end-result will express the forward kinematics as a **product of exponentials** (POE) of *differential twists*, which we define below for the planar case.

6.1 Product of Conjugations

Before tackling POE, however, let us look at a different formulation which only involves regular transforms, albeit re-arranged in a slightly different way. As an example, for a two link arm we have

$$\begin{aligned} T_t^s &= T_a^s(q_1)T_b^a(q_2)T_t^b \\ &= X_{a'}^s Z_a^{a'}(q_1)X_{b'}^a Z_b^{b'}(q_2)X_t^b \end{aligned}$$

where, as before, $X_{a'}^s$ represents the pose of the first joint axis with respect to the base frame, and $X_{b'}^a$ the pose of the second joint axis in the first link frame. Finally, X_t^b is the tool frame in the second link frame. These three transformations are constant and hence can also be derived from the joint poses at rest, e.g.

$$X_{b'}^a = T_{b0}^{a0} = T_s^{a_0}T_{b_0}^s$$

$$X_t^b = T_{t0}^{b0} = T_s^{b0}T_{t0}^s$$

Hence, we can rewrite the forward kinematics as

$$\begin{aligned} T_t^s &= \left\{T_{a0}^s Z_a^{a'}(q_1)T_s^{a0}\right\}\left\{T_{b0}^s Z_b^{b'}(q_2)T_s^{b0}\right\}T_{t0}^s \\ &= Z_a^s(q_1)Z_b^s(q_2)T_{t0}^s \end{aligned}$$

where we defined the **conjugated joint transform** as

$$Z_j^s(q_j) = T_{j0}^s Z_j^{j'}(q_j)\left(T_{j0}^s\right)^{-1}$$

The general **product of conjugations formula** is then

$$T_t^s(q) = Z_1^s(q_1)\dots Z_j^s(q_j)\dots Z_n^s(q_n)T_t^s(0) \tag{6.1}$$

The advantage is that we only have to inspect the joint poses for the manipulator at rest. There is considerable freedom in choosing the T_{j0}^s frames: for rotational joints we have to specify the correct orientation (2DOF) and a point on the joint axis (2DOF). For prismatic joints, only the orientation matters.

6.2 2D Unit Twists

The intuition behind the product of conjugations is to think about the rotation of 2D space around a joint: the conjugation machinery pulls us towards the joint, executes the joint transform, and pushes us back. The idea behind the product of exponentials formula is that can be done using a single matrix exponentiation rather than three transformations as needed for conjugations. In particular, we want to find the rigid transform $T_s^s(\theta)$ that takes points in frame S to rotated points in the same frame. If the joint axis happens to be the origin, then the corresponding 3×3 matrix is very easy to write down,

$$T_s^s(\theta) = \begin{bmatrix} R(\theta) & 0 \\ 0 & 1 \end{bmatrix} \tag{6.2}$$

where the 2×2 rotation matrix $R(\theta)$ is given as usual:

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

When the joint axis is *not* the origin but some arbitrary point p, we saw that we can write this by conjugating Equation 6.2 with a translation T_p^s to and from the joint axis p:

$$T_s^s(\theta) = T_p^s \left\{ T_p^p(\theta) \right\} \left(T_p^s \right)^{-1} = \begin{bmatrix} I & p \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R(\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & -p \\ 0 & 1 \end{bmatrix} \tag{6.3}$$

The above is a special case of the exponential map $\exp : \mathbb{R}^3 \rightarrow SE(2)$. For use in forward kinematics we only work with **unit twists** $\mathcal{S}$, with $\omega = 1$. Indeed, for revolute joints, the finite twist that corresponds to a 1 radian of rotation around the joint axis p is the **unit twist** $\mathcal{S} = (-p^{\perp}, 1) = (p_y, -p_x, 1)$.

6.3 Example

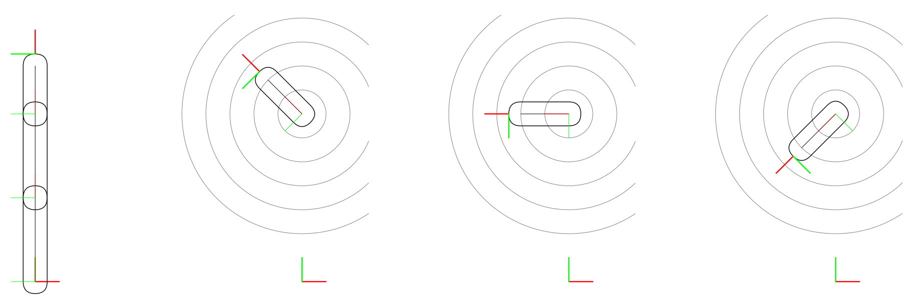

Figure 6.1: The effect of a twist around joint 3 on link 3.

To illustrate the idea with the planar example, let us look at a single joint, say joint 3. Since the joint axis in rest is at $(0, 7)$, the corresponding unit twist is

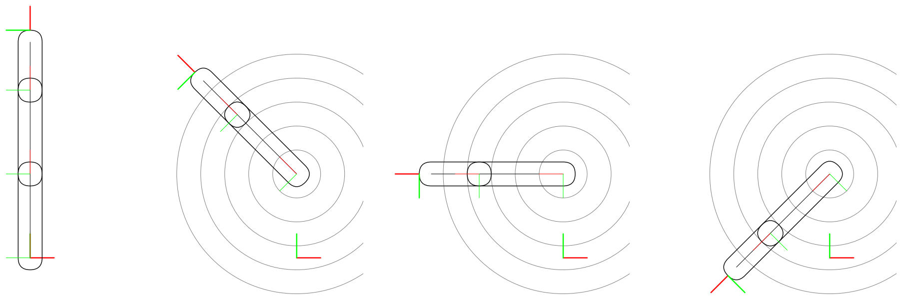

Figure 6.2: The effect of a twist around joint 2, on links 2 and 3.

$\mathcal{S}_3 = (7, 0, 1)$, with associated transform matrix given by

$$T_s^s(\theta_3) = \exp\left(\hat{\mathcal{S}}_3\theta_3\right) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 7 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 \\ \sin\theta_3 & \cos\theta_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -7 \\ 0 & 0 & 1 \end{bmatrix}$$
$$= \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 7\sin\theta_3 \\ \sin\theta_3 & \cos\theta_3 & 7(1-\cos\theta_3) \\ 0 & 0 & 1 \end{bmatrix}$$

If we view the transform $\exp\left(\hat{\mathcal{S}}_3\theta_3\right)$ as acting on all points, expressed in the base frame S, then it also applies to the entire link 3. This is illustrated in Figure 6.1, where the first panel shows the links of the arm in the rest configuration, and the next panels show the effect of the global transform.

In forward kinematics we are mostly interested in the pose of the tool. Hence, if $T_t^s(0)$ is the tool pose for a zero joint angle, then it follows that for a non-zero angle we have

$$T_t^s(\theta_3) = T_s^s(\theta_3)\, T_t^s(0) = \exp\left(\hat{\mathcal{S}}_3\theta_3\right) T_t^s(0)$$

Now we can ask what happens if we move joint 2. Since the joint axis in rest is at $(0, 3.5)$, the corresponding unit twist is $\mathcal{S}_2 = (3.5, 0, 1)$, with associated transform matrix

$$T_s^s(\theta_2) = \exp\left(\hat{\mathcal{S}}_2\theta_2\right) = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 3.5\sin\theta_2 \\ \sin\theta_2 & \cos\theta_2 & 3.5(1-\cos\theta_2) \\ 0 & 0 & 1 \end{bmatrix}$$

In Figure 6.2, I show the effect of the exponential map $\exp\left(\hat{\mathcal{S}}_2\theta_2\right)$ on the last two links. Now, since a twist acts on the entire space, if we apply it to the tool frame

$T_t^s(\theta_3)$ *after* it has been moved by $\exp\left(\hat{\mathcal{S}}_3\theta_3\right)$, it stands to reason that the effect of moving the two last joints is given by

$$T_t^s(\theta_2, \theta_3) = \exp\left(\hat{\mathcal{S}}_2\theta_2\right) T_t^s(\theta_3) = \exp\left(\hat{\mathcal{S}}_2\theta_2\right) \left\{\exp\left(\hat{\mathcal{S}}_3\theta_3\right) T_t^s(0)\right\}$$

This formula generalizes in the obvious way for the entire joint configuration,

$$T_t^s(q) = \exp\left(\hat{\mathcal{S}}_1\theta_1\right) \exp\left(\hat{\mathcal{S}}_2\theta_2\right) \exp\left(\hat{\mathcal{S}}_3\theta_3\right) T_t^s(0) \tag{6.4}$$

where the last remaining transform $\exp\left(\hat{\mathcal{S}}_1\theta_1\right)$ uses the unit twist $\mathcal{S}_1 = (0, 0, 1)$, as the axis of the first joint is the origin. The tool at rest is given by

$$T_t^s(0) = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 9.5 \\ 0 & 0 & 1 \end{pmatrix}$$

When multiplied out, we get exactly the same as in Equation 5.3.

6.4 General Case

In general, for any serial manipulator with n joints, we have the following product of exponentials expression for the forward kinematics,

$$\boxed{T_t^s(q) = \exp\left(\hat{\mathcal{S}}_1\theta_1\right) \ldots \exp\left(\hat{\mathcal{S}}_j\theta_j\right) \ldots \exp\left(\hat{\mathcal{S}}_n\theta_n\right) T_t^s(0)} \tag{6.5}$$

and, while the left-to-right order has to follow the manipulator structure, the formula above does *not* depend on the order in which the actual joints are actuated.

6.5 Exercise

Provide the POE for your RP example.

7 Three-dimensional Geometry

The story above generalizes almost entirely to three dimensions. We start with some geometry.

7.1 Rotations in 3D aka SO(3)

Basic Facts

Rotating a point in 3D around the origin from a moving body frame B to a base frame S can be done by multiplying with a 3×3 orthonormal rotation matrix

$$p^b = R_b^s p^b$$

where the indices on R_b^s indicate the source and destination frames. The columns of R_b^s represent the axes of frame B in the S coordinate frame:

$$R_b^s = \begin{bmatrix} \hat{x}_b^s & \hat{y}_b^s & \hat{z}_b^s \end{bmatrix}$$

The 3D rotations together with composition constitute the **special orthogonal group** $SO(3)$. It is made up of all 3×3 orthonormal matrices with determinant 1, with matrix multiplication implementing composition. However, *3D rotations do not commute*, i.e., in general $R_2 R_1 \neq R_1 R_2$.

Intuition

The intuition for 3D rotations stems from **Euler's theorem**:

Theorem. *Every 3D rotation can be expressed as a rotation around a single axis.*

Consider the trajectory traced out by spatial coordinates of a point p^b on a rigid rotating body in 3D:

$$p^s(t) = R_b^s(t) p^b.$$

Then the theorem says that at any given moment the velocity vector $v^s(t)$ of the point must be consistent with a rotational trajectory around an axis $n^s(t)$, i.e.,

$$v^s(t) = \omega(t) \left[n^s(t) \times p^s(t) \right] = \Omega(t) \times p^s(t) \tag{7.1}$$

where $n^s(t)$, a unit vector, is the instantaneous rotation axis, and $\Omega(t) \stackrel{\Delta}{=} \omega(t) n^s(t)$ is the time-varying angular velocity vector: its direction defines the rotation axis, and its magnitude the angular velocity. Following the pattern established above, we can rewrite this to find a Lie algebra for $SO(3)$:

$$\begin{aligned} v^s(t) &= \Omega(t) \times p^s(t) \\ &= \widehat{\Omega}(t) p^s(t) \end{aligned}$$

where the cross product was rewritten as multiplication with the skew-symmetric matrix $\widehat{\Omega}$:

$$\widehat{\Omega} \triangleq \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

Hence, we see that the Lie algebra $\mathfrak{so}(3)$ associated with $SO(3)$ is the space of skew-symmetric matrices, and the above is the hat operator for $\mathfrak{so}(3)$. The lie algebra $\mathfrak{so}(3)$ is isomorphic to the vector space $\mathbb{R}^3$, so we can add angular velocity vectors and multiply them with scalars.

The Exponential Map

We can obtain a closed-form solution for the exponential map for $SO(3)$ by thinking about the finite displacement of a point p^b undergoing a constant rotational motion specified by $\Omega = \omega n^s$:

$$\begin{aligned} p^s &= p^b + n^s \times n^s \times p^b + \left(n^s \times p^b\right) \sin \omega t - \left(n^s \times n^s \times p^b\right) \cos \omega t \\ &= \left[I + N \sin \omega t + N^2 (1 - \cos \omega t)\right] p^b \end{aligned}$$

where $N \triangleq \widehat{n^s}$ is the skew-symmetric matrix associated with the rotation axis. Hence, the exponential map for $SO(3)$ is

$$\exp\left(\hat{\Omega} t\right) = I + N \sin \omega t + N^2 (1 - \cos \omega t) \tag{7.2}$$

This is the well known **Rodrigues' formula**.

7.2 3D Rigid transforms aka SE(3)

Basic Facts

A point p^b on a rigid moving in three-space can be transformed by a 3D rigid transform, which is a 3D rotation followed by a 3D translation, to be expressed in a fixed base frame S,

$$p^s = R_b^s p^b + t_b^s$$

where $R_b^s \in SO(3)$ and $t_b^s \in \mathbb{R}^3$. We denote this transform by $T_b^s \triangleq (R_b^s, t_b^s)$. The **special Euclidean group** $SE(3)$, with the group operation defined similarly as in Equation 2.4. Moreover, the group $SE(3)$ is a subgroup of a general linear group $GL(4)$ of degree 4, by embedding the rotation and translation into a 4×4 invertible matrix defined as

$$T_b^s = \begin{bmatrix} R_b^s & t_b^s \\ 0 & 1 \end{bmatrix}$$

Again, by embedding 3D points in a four-vector, a 3D rigid transform acting on a point can be implemented by matrix-vector multiplication:

$$\begin{bmatrix} R_b^s & t_b^s \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p^b \\ 1 \end{bmatrix} = \begin{bmatrix} R_b^s p^b + t_b^s \\ 1 \end{bmatrix}$$

Intuitions

The intuition for rigid transforms in 3D stems from the far-from-obvious **Chasles' theorem**:

Theorem. *Every 3D rigid transform can be expressed as a rotation around an axis, followed by a translation along that axis (or vice versa, the two operations in this case commute).*

The axis in question is called the **screw axis**, and the trajectory traced out by spatial coordinates of a point p^b on a rigidly moving body describes a helical (screw) motion around the axis. We take the velocity induces by the rotation, and simply add a translational component in the direction of the axis

$$v^s(t) = \Omega(t) \times [p^s(t) - q^s(t)] + \lambda\Omega(t)$$

where $q^s(t)$ is *any* point on the screw axis, and λ is defined as the **pitch** of the screw motion, i.e., the proportion of the translational vs. the rotational component. We see that

$$\begin{bmatrix} v^s(t) \\ 0 \end{bmatrix} = \begin{bmatrix} \hat{\Omega}(t) & v(t) \\ 0 & 0 \end{bmatrix} \begin{bmatrix} p^s(t) \\ 1 \end{bmatrix}$$

where $v(t) \triangleq q^s(t) \times \Omega(t) + \lambda\Omega(t)$ is again the velocity of the origin. Hence, the 4×4 matrices of the form

$$\begin{bmatrix} \hat{\Omega} & v \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -\omega_z & \omega_y & v_x \\ \omega_z & 0 & -\omega_x & v_y \\ -\omega_y & \omega_x & 0 & v_z \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

are elements of the Lie algebra $\mathfrak{se}(3)$ associated with $SO(3)$.

In 3D, a **differential twist**[1] is given by an angular velocity $\Omega \in \mathbb{R}^3$ and a linear velocity $v \in \mathbb{R}^3$, collected in a column vector $\mathcal{V} \in \mathbb{R}^6$:

$$\mathcal{V} \triangleq \begin{bmatrix} \Omega \\ v \end{bmatrix}$$

At any given time, the point on the screw axis closest to the origin is given by

$$q = [\Omega/\omega] \times v/\omega = \frac{\Omega \times v}{\omega^2}. \tag{7.3}$$

Compare that with Equation 2.7, and it looks very familiar.

The Exponential Map

We can use the point q above by deriving a closed form for the exponential map by conjugation: move to the screw axis, execute the screw motion, and move back. For a constant differential twist $\mathcal{V} = (\Omega, v)$, with $\omega = \|\Omega\| \neq 0$, the exponential map $\exp : \mathfrak{so}(3) \to SE(3)$ is then given by

$$\exp\left(\hat{\mathcal{V}}t\right) = \begin{bmatrix} I & q \\ 0 & 1 \end{bmatrix} \begin{bmatrix} e^{\hat{\Omega}t} & (\lambda t)\Omega \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & -q \\ 0 & 1 \end{bmatrix} \tag{7.4}$$

where the rotation matrix $R(\omega t)$ is given by Rodrigues' formula. The above is but a simple generalization of a circular arc in 2D to a screw motion in 3D.

Multiplying through, and making use of $\Omega = n\omega$, $q = (n \times v)/\omega = Nv/\omega$, and $\lambda = n^T v/\omega$ we get

$$\begin{aligned} \exp\left(\hat{\mathcal{V}}t\right) &= \begin{bmatrix} e^{\hat{\Omega}t} & \left[I - e^{\hat{\Omega}t}\right] q + (\lambda t)\Omega \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} e^{\hat{\Omega}t} & \left[I - e^{\hat{\Omega}t}\right] Nv/\omega + nn^T vt \\ 0 & 1 \end{bmatrix} \end{aligned}$$

7.3 Exercise

A rigid body is moving in $SE(3)$ with a constant spatial velocity $\mathcal{V} = (0, 1, 0, 0, 2, 0)$. What is the velocity v^s of the point $p^s = (6, 7, 8)$?

[1] Note we follow a different convention from [3] in that we reserve the first three components for rotation, and the last three for translation.

8 Spatial Manipulators

8.1 Kinematic Chains in Three Dimensions

In three dimensions the matrices are now 4×4, but exactly the same expressions are use to describe a kinematic chain:

$$T_t^s(q) = T_1^s(q_1) \dots T_j^{j-1}(q_j) \dots T_n^{n-1}(q_n) X_t^n. \tag{8.1}$$

and to describe any serial manipulator we can again alternate fixed link transforms X_j^{j-1} and parameterized joint transforms $Z_j^j(q_j)$:

$$T_t^s(q) = X_1^s Z_1^1(q_1) \dots X_j^{j-1} Z_j^j(q_j) \dots X_n^{n-1} Z_n^n(q_n) X_t^n \tag{8.2}$$

In 3D we typically obey the convention that the axis of rotation is chosen to be the Z-axis, hence the suggestive naming of the corresponding matrix parameterized by a joint angle. We can now also introduce **prismatic joints**, which are typically parameterized as translations *along* the Z-axis.

8.2 Denavit-Hartenberg Conventions

No introduction to serial manipulators is complete without mentioning the **Denavit-Hartenberg convention**, which is a particular choice of coordinate frames to make equation 8.2 as simple as possible. In particular, as suggested by the alternation of matrices named X and Z, we ensure that

1. all joint axes are aligned with the Z-axis, and the corresponding transform is parameterized by two parameters, a rotation θ around Z and a displacement d along Z;

2. the X-axis is chosen to be the **common perpendicular** between two successive joint axes, and the link geometry is described by two parameters, a rotation α around X and a displacement a along X.

It might be surprising that only four parameters (θ, d, α, a) are needed to specify the location of one frame relative to another. However, these frames are special since they have two independent conditions imposed: the X-axis intersects the next Z-axis, and is perpendicular to it [2].

Subject to these two constraints, there are two popular variants in use, the proximal and distal variants, that differ on where they put the coordinate frame on the link. In the **distal** variant, the link coordinate frame $T_j^s(q)$ is made to coincide with joint axis $j+1$, and link frame n is defined to be identical to the tool frame. This convention is a bit awkward to work with.

In the simpler, **proximal** variant, also denoted the **modified Denavit-Hartenberg convention** in [1], the link coordinate frame $T^s_j(q)$ is made to coincide with joint axis j, and the transform $T^{j-1}_j(q_j)$ between links is written as:

$$T^{j-1}_j(q_j) = X^{j-1}_j(\alpha_{j-1}, a_{j-1}) Z^j_j(\theta_j, d_j)$$

where q_j is either θ_j for a revolute joint, or to d_j for a prismatic joint, and

$$X^{j-1}_j(\alpha_{j-1}, a_{j-1}) = T_{Rx}(\alpha_{j-1}) T_x(a_{j-1}) \qquad Z^j_j(\theta_j, d_j) = T_{Rz}(\theta_j) T_z(d_j)$$

8.3 Product of Exponentials in 3D

To generalize the product of exponentials formula, we need the concept of a 3D **unit twist** $\mathcal{S}$, which consist of an **axis of rotation** $\bar{\omega}$ combined with a **linear motion vector** $\bar{v}$, which defines a rotational or translational motion around the joint.

For a *prismatic* joint, a unit twist is simply $\mathcal{S} = (0, \bar{v})$, with $\bar{v}$ a specifying the direction of motion, and the exponential map is just a translation:

$$\exp(\mathcal{S}t) = \begin{bmatrix} I & \bar{v}t \\ 0 & 1 \end{bmatrix}$$

For a *revolute* joint, the unit twist is given by $\mathcal{S} = (\bar{\omega}, p \times \bar{\omega})$, where $\bar{\omega}$ is a unit vector specifying the axis of rotation, and p is *any* point on the joint axis. In this case the exponential map simplifies to

$$\exp(\mathcal{S}\theta) = \begin{bmatrix} I & p \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R(\bar{\omega}\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & -p \\ 0 & 1 \end{bmatrix}$$

The general formula for the rotation matrix $R(\bar{\omega}\theta)$ is given in the appendix, but is easy whenever the rotation axis $\bar{\omega}$ is aligned with a coordinate axis. For example, for a rotation around the Z-axis, i.e., $\omega = [0, 0, 1]^T$ we have

$$R(\bar{\omega}\theta) = R([0\,0\,\theta]^T) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

and similarly, for respectively the X-axis and Y-axis, we have

$$R([\theta\,0\,0]^T) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$$

$$R([0\,\theta\,0]^T) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

Finally, we obtain the same product of exponentials as in the planar case,

$$\boxed{T_t^s(q) = \exp\left(\hat{\mathcal{S}}_1 q_1\right) \dots \exp\left(\hat{\mathcal{S}}_j q_j\right) \dots \exp\left(\hat{\mathcal{S}}_n q_n\right) T_t^s(0)} \qquad (8.3)$$

8.4 Detailed Example: The Pincher Robot

Figure 8.1: Pincher robot at rest, with all joint angles at 0. The chosen base frame S and tool frame T are shown as RGB coordinate frames.

The Pincher robot is shown in Figure 8.1 in its rest configuration. I chose to put the base frame S at the intersection of the vertical joint 1 axis and the horizontal joint 2 axis, which makes the transforms for the first two joint axes easy, with unit twists $\mathcal{S}_1 = (0, 0, 1, 0, 0, 0)$ and $\mathcal{S}_2 = (1, 0, 0, 0, 0, 0)$, respectively:

$$e^{\mathcal{S}_1 \theta_1} = \begin{pmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \text{ and } e^{\mathcal{S}_2 \theta_2} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta_2 & -\sin\theta_2 & 0 \\ 0 & \sin\theta_2 & \cos\theta_2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Note that the joint angles are measured from the configuration at rest, and positive joint angles will make the arm lean backwards.

The third joint axis, at rest (all servos at 512) is 10.5 cm above the second one. We can just conjugate a rotation around the x-axis, corresponding to the unit twist $\mathcal{S}_3 = (1, 0, 0, 0, 10.5, 0)$:

$$e^{\mathcal{S}_3\theta_3} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta_3 & -\sin\theta_3 & 10.5\sin\theta_3 \\ 0 & \sin(\theta_3) & \cos(\theta_3) & 10.5(1-\cos\theta_3) \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The fourth joint is the same, except it is higher, with unit twist $\mathcal{S}_4 = (1, 0, 0, 0, 21, 0)$:

$$e^{\mathcal{S}_4\theta_4} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta_4 & -\sin\theta_4 & 21\sin\theta_4 \\ 0 & \sin\theta_4 & \cos\theta_4 & 21(1-\cos\theta_4) \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Finally, the fifth DOF controls the gripper and is not included in the inverse kinematics. The tool frame, at rest, is 6.5 cm above the fourth joint:

$$T_t^s(0) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 27.5 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Multiplying all these together, we get the forward kinematics as follows:

$$T_t^s(q) = \exp\left(\hat{\mathcal{S}}_1\theta_1\right)\exp\left(\hat{\mathcal{S}}_2\theta_2\right)\exp\left(\hat{\mathcal{S}}_3\theta_3\right)\exp\left(\hat{\mathcal{S}}_4\theta_4\right)T_t^s(0) =$$

$$\begin{pmatrix} \cos\theta_1 & -\sin\theta_1\cos\beta & \sin\theta_1\sin\beta & \frac{1}{2}\sin\theta_1\left(21\sin\theta_2 + 21\sin\alpha + 13\sin\beta\right) \\ \sin\theta_1 & \cos\theta_1\cos\beta & -\cos\theta_1\sin\beta & -\frac{1}{2}\cos\theta_1\left(21\sin\theta_2 + 21\sin\alpha + 13\sin\beta\right) \\ 0 & \sin\beta & \cos\beta & \frac{1}{2}\left(21\cos\theta_2 + 21\cos\alpha + 13\cos\beta\right) \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

where $\alpha = \theta_2 + \theta_3$ and $\beta = \theta_2 + \theta_3 + \theta_4$. It is easy to see that the first joint angle θ_1 rotates the entire arm around the vertical, and that the other angles dictate the kinematics in that rotated frame. In fact, not coincidentally, when setting θ_1 to zero we can recognize exactly the same structure as the three-link kinematics chain from Equation 5.3, in the Y-Z plane:

$$T_t^s(0, \theta_2, \theta_3, \theta_4) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\beta & -\sin\beta & -\frac{1}{2}\left(21\sin\theta_2 + 21\sin\alpha + 13\sin\beta\right) \\ 0 & \sin\beta & \cos\beta & \frac{1}{2}\left(21\cos\theta_2 + 21\cos\alpha + 13\cos\beta\right) \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

It is clear from this that, with three joints in this plane and the rotation around joint 1, we can reach any 3D pose within the workspace. However, we have no way of rotating the tool around its Z-axis.

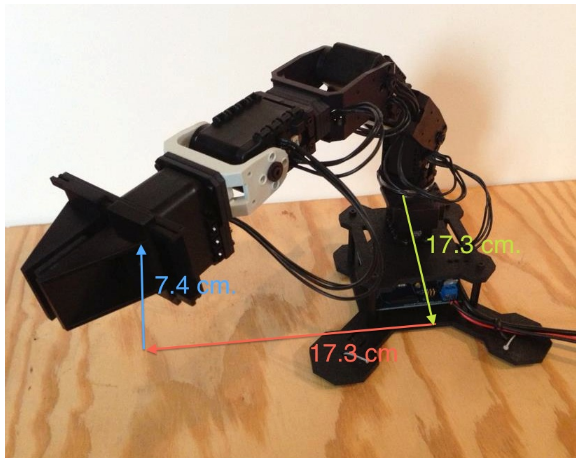

Figure 8.2: The pose of the robot with joint angles $q = (-45^o, -45^o, -45^o, 0)$.

To test the FK, let us plug in some angles:

$$T_t^s(-45^o, -45^o, -45^o, 0) = \begin{pmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 17.3 \\ -\frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 17.3 \\ 0 & -1 & 0 & 7.4 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

It can be verified in Figure 8.2 (and I verified it in reality) that the tool is indeed horizontal now, rotated 45 degrees to the left, and is at position $(17.3, 17.3, 7.4)$ with respect to the chosen base frame.

8.5 Exercise

Let us now consider the 3D Unimate robot, which is RRP, with the first rotation around vertical (yaw, around a vertical Z axis), and the second rotation is around the Y-axis (pitch). Provide the POE formula and calculate the end-effector pose for an arbitrary pose where the arm extends *upwards*.

A Appendix: Geometry Reference

A.1 Planar Rotations aka SO(2)

1. As a matrix group $SO(2) \subset GL(2)$:

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

2. Spatial velocity:

$$v^s(t) = \omega(t)\,[p^s(t)]^\perp$$

3. Lie algebra isomorphic to $\mathbb{R}$, the space of angular velocities ω. Hat operator given by:

$$\hat{\omega} \in \mathfrak{so}(2) \triangleq \begin{bmatrix} 0 & -\omega \\ \omega & 0 \end{bmatrix}$$

4. The exponential map:

$$\exp(\hat{\omega}t) = \begin{bmatrix} \cos\omega t & -\sin\omega t \\ \sin\omega t & \cos\omega t \end{bmatrix}$$

A.2 Rotations in 3D aka SO(3)

1. As a matrix group $SO(3) \subset GL(3)$:

$$R^s_b = \begin{bmatrix} \hat{x}^s_b & \hat{y}^s_b & \hat{z}^s_b \end{bmatrix}$$

2. Spatial velocity:

$$v^s(t) = \Omega(t) \times p^s(t)$$

3. Lie algebra isomorphic to $\mathbb{R}^3$, the space of angular velocity vectors Ω. Hat operator given by:

$$\widehat{\Omega} \triangleq \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

4. The exponential map, with $N = \hat{\Omega}/\omega$:

$$\exp\left(\hat{\Omega}t\right) = I + N\sin\omega t + N^2(1-\cos\omega t)$$

A.3 2D Rigid Transforms aka SE(2)

1. As a matrix group: $SE(2) \subset GL(3)$:

$$T_b^s = \begin{bmatrix} R_b^s & t_b^s \\ 0 & 1 \end{bmatrix}$$

2. Spatial velocity:

$$v^s(t) = \omega(t)\left[p^s(t) - q^s(t)\right]^{\perp}$$

3. Lie algebra isomorphic to $\mathbb{R}^3$, the space of 2D differential twist coordinates $\mathcal{V} \triangleq (v_x, v_y, \omega)$. Hat operator given by:

$$\hat{\mathcal{V}} \in \mathfrak{se}(2) = \begin{bmatrix} \hat{\omega} & v \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -\omega & v_x \\ \omega & 0 & v_y \\ 0 & 0 & 0 \end{bmatrix}$$

4. The exponential map, with $q = v^{\perp}/\omega$:

$$\exp\left(\hat{\mathcal{V}}t\right) = \begin{bmatrix} I & q \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \exp(\hat{\omega}t) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & -q \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \exp(\hat{\omega}t) & \left[I - \exp(\hat{\omega}t)\right] v^{\perp}/\omega \\ 0 & 1 \end{bmatrix}$$

A.4 3D Rigid transforms aka SE(3)

1. As a matrix group: $SE(3) \subset GL(4)$:

$$T_b^s = \begin{bmatrix} R_b^s & t_b^s \\ 0 & 1 \end{bmatrix}$$

2. Spatial velocity:

$$v^s(t) = \Omega(t) \times \left[p^s(t) - q^s(t)\right] + \lambda\Omega(t)$$

3. Lie algebra isomorphic to $\mathbb{R}^6$, the space of 3D differential twist coordinates $\mathcal{V} \triangleq (\Omega, v)$. Hat operator given by:

$$\hat{\mathcal{V}} \in \mathfrak{se}(3) = \begin{bmatrix} \hat{\Omega} & v \\ 0 & 0 \end{bmatrix}$$

4. The exponential map, with $q = (\Omega \times v)/\omega^2$:

$$\exp\left(\hat{\mathcal{V}}t\right) = \begin{bmatrix} I & q \\ 0 & 1 \end{bmatrix} \begin{bmatrix} e^{\hat{\Omega}t} & (\lambda t)\Omega \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & -q \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} e^{\hat{\Omega}t} & \left[I - e^{\hat{\Omega}t}\right] Nv/\omega + nn^T vt \\ 0 & 1 \end{bmatrix}$$

A.5 Unit twists

As defined in Lynch & Park, a unit twist is either a pure velocity or a screw:

- if $\omega = 0$ then $\mathcal{S} = (0, v/\|v\|)$, and corresponding twist $\mathcal{V} = \mathcal{S}v = (0, v)$;
- if $\omega \neq 0$ then $\mathcal{S} = (\Omega/\omega, v/\omega) = (n, v/\omega)$, and corresponding twist $\mathcal{V} = \mathcal{S}\omega = (\Omega, v)$.

References

[1] Peter Corke. *Robotics, Vision, and Control*. Springer, 2011.

[2] H. Lipkin. A note on Denavit-Hartenberg notation in robotics. In *Proc. ASME IDETC/CIE*, pages 921–926, 2005.

[3] R.M. Murray, Z. Li, and S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.