Lecture 2: Rigid Motions and Homogeneous Transformations

Reading: SHV Chapter 2

Mechanical Engineering
Hanz Richter, PhD

Representing Points, Vectors and Frames

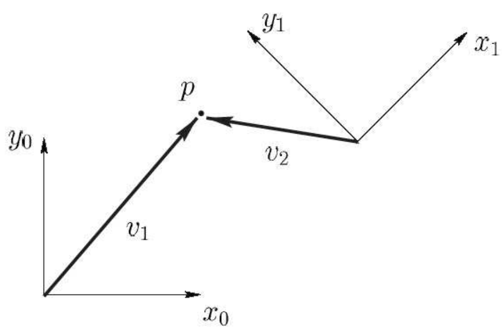

Figure 2.1: Two coordinate frames, a point p, and two vectors v_1 and v_2.

Frames: $o_0x_0y_0$, $o_1x_1y_1$
Coordinates of point p relative to frame 0:
$p^0 = [5|6]^T$
Components of vector v_2 relative to frame 1:
$v_2^1 = [-2.89|4.2]^T$

Our goals in this chapter are:

1. To convert the coordinates of a point from one frame to another (typically from the end effector frame to the base (world) frame. See Prob. 2-39.
2. To describe the position and orientation of a rigid object (typically attached to the end effector frame) in a compact matrix notation: the *homogeneous transformation*.
3. For this, we need to understand how a sequence of rotations can be understood in terms of matrix multiplications.

Linear Transformations and Matrices

- A linear transformation T is a mapping which takes a vector v and returns another vector Tv. T satisfies $T(v_1 + v_2) = T(v_1) + T(v_2)$ and $T(\alpha v) = \alpha T(v)$, with α being a scalar. The vectors v belong to a vector space. In this course, we work only with $\mathbb{R}^2$ and $\mathbb{R}^3$ as vector spaces, with real scalars.
- Once a basis is chosen for the vector space ($\mathbb{R}^2$ or $\mathbb{R}^3$), the linear transformation is equivalent to multiplication of a matrix A by the vector v.
- A linear transformation is defined by what it does to a basis. For example, choose the standard basis for $\mathbb{R}^2 : e_1 = [1|0]^T$ and $e_2 = [0|1]^T$ and consider the transformation T defined by $T(e_1) = [1|1]^T$ and $T(e_2) = [1| - 3]^T$.

Linear Transformations and Matrices...

- *The columns of A are the images of the basis vectors under T.*. Find the matrix A of T w.r.t. the standard basis and then transform the vector $v = [2|1]^T$.
- Now find the matrix B of T with respect to the basis $u_1 = [-1|1]^T$ and $u_2 = [2|2]^T$ (express these as a combination of e_1 and e_2 and use linearity to find the images. Express these images in the new basis to find the columns of B).
- Express v in this basis and transform it under B.
- Use Matlab to interpret the results geometrically.

Rotation Matrices

We saw that multiplication by the matrix of T had the effect of rotating v by 45 degrees clockwise and changing the length from $\sqrt{5}$ to $\sqrt{(10)}$.
In general, given v, the effect of Av is a rotation and a change in length.
In robotics we are interested in a special kind of matrix which only rotates, but does not change the length of v. These are called *orthogonal matrices*. The set of all $n \times n$ orthogonal matrices is called $SO(n)$, for Special Orthogonal Group of order n. Matrices in $SO(n)$ have the following properties:

- $R^{-1} = R^T \in SO(n)$
- Any two different columns and any two different rows are mutually orthogonal ($r_i.r_j = 0$ whenever $i \neq j$).
- The length of each row and column is 1.
- $\det R = 1$. Note: if $\det R = -1$, R is still orthogonal, but not a member of $SO(n)$.

2D Rotation Matrices by Angle of Rotation

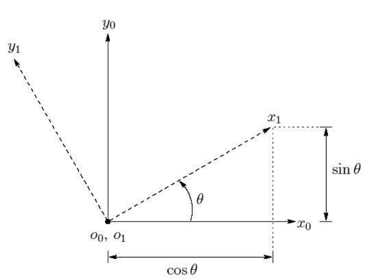

Figure 2.2: Coordinate frame $o_1x_1y_1$ is oriented at an angle θ with respect to $o_0x_0y_0$.

As we derive next, the coordinates of a point p^0 (in frame 0) can be obtained from the coordinates of the point in frame 1 (p^1) by multiplication:

$$p^0 = R_1^0 p^1$$

where the notation for R_1^0 is understood as “the rotation matrix to go from 1 to 0”. R_1^0 has the form

$$R_1^0 = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

This way of specifying R is not convenient due to a jump from 2π to 0. Also, the method does not extend well to 3D. **Show that** $R \in SO(2)$.

2D Rotation Matrices by Projection

A far more convenient way to find a rotation matrix is to find the projections of the axes of frame 1 onto those of frame 0, specifically:

$$R_1^0 = \begin{bmatrix} x_1.x_0 & y_1.x_0 & z_1.x_0 \\ x_1.y_0 & y_1.y_0 & z_1.y_0 \\ x_1.z_0 & y_1.z_0 & z_1.z_0 \end{bmatrix}$$

Recall that the projection of a unit vector onto another is given by the dot product.
Examine Examples 2.1 and 2.2 at home. As a class example, we find the 3D rotation matrix corresponding to a -45 degree rotation about the x_0-axis. Discuss the sign in relation to the right-hand rule.

3 Interpretations of the Rotation Matrix

- We just saw how R converts the coordinates of a point from frame to frame.
- R can be thought of the matrix of some linear transformation w.r.t. standard basis (the frame). Multiplication of a vector v by R yields a rotated vector (still unit length). Rv has coordinates in the same frame.
- By definition, R packs the orientation of frame 1 relative to frame 0 in a compact matrix notation.

Properties of Rotation Matrices

- All properties of $SO(n)$ hold.
- If we wish to find R_0^1, we switch the indices 0 and 1 in the definition of R_1^0. Since the dot product commutes, we find $R_0^1 = (R_1^0)^T$ (see page 40).
- Also, if $v^0 = R_1^0 v^1$ then, since R_1^0 is invertible (nonzero determinant), we must have $v^1 = (R_1^0)^{-1} v^0$. On the other hand, we can say $v^1 = R_0^1 v^0$, therefore

$$R_0^1 = (R_1^0)^{-1} = (R_1^0)^T$$

This confirms one of the orthogonality requirements.

Change of Basis: Similarity Transforms

Suppose we have two frames related by a particular R_1^0. In addition, we have a rotation matrix A defined in frame 0 (interpret A as a vector rotator s.t. $w^0 = Av^0$). What is the matrix B realizing the same rotation in frame 1? -We derive Eq. 2.11 in SHV.
We have $v^1 = R_0^1 v^0$, therefore $v^0 = (R_0^1)^{-1} v^1$. Also, $w^1 = R_0^1 w^0$, but $w^0 = Av^0$, so making substitutions: $w^1 = R_0^1 A v^0 = R_0^1 A (R_0^1)^{-1} v^1$.
Therefore

$$B = R_0^1 A (R_0^1)^{-1}$$

Complete the steps necessary to obtain Eq. 2.11 in SHV.
We say that A and B are related by *similarity transformation*. These transformations will be used frequently.

Chained Rotations

If we attach a coordinate frame to each link of a manipulator, we see that the end effector frame will be related to the base frame by a series of rotations (and offsets, which will be treated later).
We need to obtain the overall rotation matrix corresponding to a sequence of rotations. Two cases arise: rotations relative to the current frame (recursively defined); or rotations relative to a fixed frame (usually the world frame).

Chained Rotations: Current Frame

Suppose we have a series of rotations defined w.r.t. the last frame:
R_1^0, R_2^1, R_3^2 ... etc. Then, if we start with a point p^0 we have

$$p^0 = R_1^0 p^1 = R_1^0 R_2^1 p^2 = R_1^0 R_2^1 R_3^2 p^3 ...$$

Therefore, after k rotations we have

$$p^0 = R_1^0 R_2^1 R_3^2 .... R_k^{k-1} p^k = R_k^0 p^k$$

The composite rotation is simply the product (**in the same order!**) of all rotations.
Remember that rotation does not commute (except in the infinitesimal case), nor does matrix multiplication.

Chained Rotations: Fixed Frame

Suppose we have a series of rotations defined always w.r.t. the first frame: $R_1^0,\ R_2^0,\ R_3^0$ ... etc. Then, if we start with p^0 we have

$$p^0 = R_1^0 p^1$$

The next rotation is defined relative to frame 0. We can transform it to frame 1 according to the similarity transformation formula $R_2^1 = (R_1^0)^{-1} R_2^0 R_1^0$. Then we have

$$p^0 = R_1^0 (R_1^0)^{-1} R_2^0 R_1^0 p^2 = R_2^0 R_1^0 p^2$$

As an exercise continue the process to show that the overall rotation is obtained by multiplying the individual ones **in reverse**.

Examples

- 2-14
- 2-8

Parameterization of Rotations

- A 3-by-3 matrix has 9 entries. However, we know that any rotation can be described in terms of only 3 angles (one rotation about each coord. axis, for instance).
- This means that the 9 entries are related (not independent). We can actually write 6 equations expressing the conditions for the matrix to belong to $SO(3)$.
- There are three useful parameterizations: **Euler angles**, **Roll, Pitch and Yaw** and **Axis/Angle** representation. We study only the first two.
- The Euler angles are three specific successive rotations about *current frames*.
- The Roll, Pitch and Yaw rotations are performed relative to a *fixed frame*.

Euler Angles

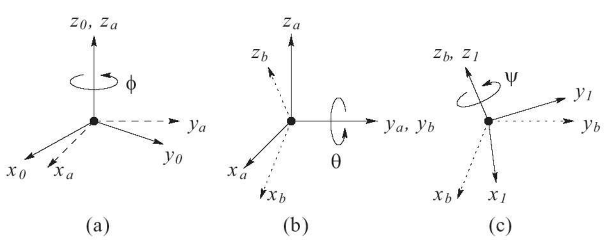

Figure 2.10: Euler angle representation.

1. Rotate about z-axis by angle ϕ
2. Rotate about the *current* y-axis by θ
3. Rotate about the *current* z-axis by ψ

As an exercise, find each one of the rotation matrices and multiply to obtain Eq. 2.26

$$R_{ZYZ} = \begin{bmatrix} c_\phi c_\theta c_\psi - s_\phi s_\psi & -c_\phi c_\theta s_\psi - s_\phi c_\psi & c_\phi s_\theta \\ s_\phi c_\theta c_\phi + c_\phi s_\psi & -s_\phi c_\theta s_\psi + c_\phi c_\psi & s_\phi s_\theta \\ -s_\theta c_\psi & s_\theta s_\psi & c_\theta \end{bmatrix}$$

Note the shorthand notation for sine and cosine. To get an idea of the inverse kinematics problem, consider finding ϕ,θ and ψ given a numeric R_{ZYZ} matrix. See some solutions in pages 55-56.

Roll, Pitch and Yaw Parameterization

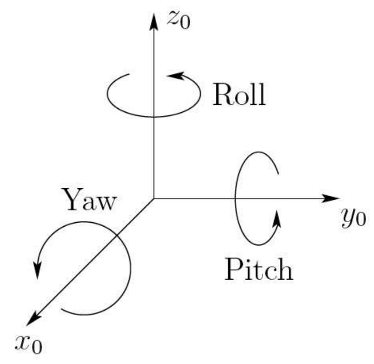

Figure 2.11: Roll, pitch, and yaw angles.

1. Rotate about x_0-axis by angle ψ (yaw)
2. Rotate about y_0-axis by θ (pitch)
3. Rotate about z_0-axis by ϕ (roll)

As an exercise, find each one of the rotation matrices and multiply to obtain Eq. 2.38

$$R = \begin{bmatrix} c_\phi c_\theta & -s_\phi c_\psi + c_\phi s_\theta s_\psi & s_\phi s_\psi + c_\phi s_\theta c_\psi \\ s_\phi c_\theta & c_\phi c_\theta + s_\phi s_\theta s_\psi & -c_\phi s_\psi + s_\phi s_\theta c_\psi \\ -s_\theta & c_\theta s_\psi & c_\theta c_\psi \end{bmatrix}$$

Note that the same rotations in reverse order and w.r.t current frames would give the same matrix. Also, the solutions to the inverse kinematics problem can be otained using techniques similar to those used for the Euler parameterization.

Translation+Rotation: Homogeneous Transformations

The objective is to account for a displacement of origin of the rotated frames. Suppose we have three frames: $o_0x_0y_0z_0$,$o_1x_1y_1z_1$ and $o_2x_2y_2z_2$. The vector from frame 0 to frame 1 is d_1 and the vector from frame 1 to frame 2 is d_2 (so far, the d's are coordinate-free). Finally, suppose p^2 is a point expressed in the coordinates of frame 2. Then we have

$$\begin{aligned} p^0 &= R_1^0 p^1 + d_1^0 \\ p^1 &= R_2^1 p^2 + d_2^1 \end{aligned}$$

Substituting

$$p^0 = (R_1^0 R_2^1) p^2 + (R_1^0 d_2^1 + d_1^0)$$

Homogeneous Transformations...

It is not difficult to show that a single rotation accompanied by a translation can be captured by a matrix multiplication of the form:

$$\begin{bmatrix} p^0 \\ 1 \end{bmatrix} = \begin{bmatrix} R_1^0 & d_1^0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p^0 \\ 1 \end{bmatrix}$$

The matrix, notated H_1^0, is 4-by-4. The zero is actually a 1-by-3 array. The d_1^0 is a column-vector of 3 components.
The set of all matrices of the above form is called $SE(3)$, for Special Euclidean Group.

Properties of $SE(3)$

- A series of rotations+displacements H_1,H_2..etc. is equivalent to a single rotation and displacement of the **product** of the individual H's.
- If the rotations contained in each H are w.r.t. to current frames, we use the same order for the multiplication: $H_1 H_2$....
- If the rotations contained in each H are w.r.t. to a fixed frame, we use the reverse order in the multiplication: $...H_2 H_1$.
- There is a formula for the inverse homogeneous transformation: See Eq.2.60.
- There is a set of 6 *homogeneous primitives* which may be combined to generate any arbitrary element of $SE(3)$: three translations and three rotations (kind of expected).

Generators of $SE(3)$

The six homogeneous primitives are

$$\text{Trans}_{x,a} = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{Rot}_{x,\alpha} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_\alpha & -s_\alpha & 0 \\ 0 & s_\alpha & c_\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{Trans}_{y,b} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{Rot}_{y,\beta} = \begin{bmatrix} c_\beta & 0 & s_\beta & 0 \\ 0 & 1 & 0 & 0 \\ -s_\beta & 0 & c_\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{Trans}_{z,c} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{Rot}_{z,\gamma} = \begin{bmatrix} c_\gamma & -s_\gamma & 0 & 0 \\ s_\gamma & c_\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Examples

- Example 2-10
- Exercise 2-37,
- Exercises 2-38,2-39