EN3563 Robotics
Laboratory Experiment 01

Title: Spatial Descriptions and Orientation Representations using Robotics Toolbox

1. Introduction

The relationship between two rigid bodies can be established by first attaching a coordinate frame to each rigid body and describing the relative position and relative orientation between the two coordinate frames (Fig. 1). To describe relative orientation, we use a rotation matrix.

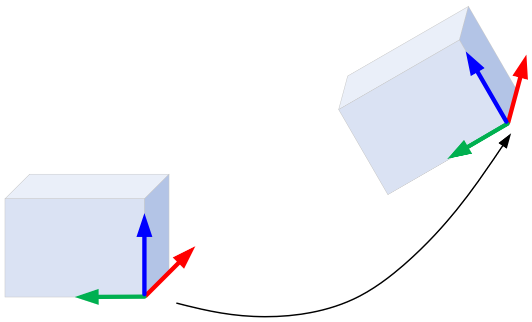

Figure 1: Establishing a relationship between two rigid bodies

The Robotics Toolbox is a MATLAB toolbox software that supports research and teaching into arm-type and mobile robotics. It contains functions and classes to represent orientation and pose in 2D and 3D (SO(2), SE(2), SO(3), SE(3)) as matrices, quaternions, twists, triple angles, and matrix exponentials. You are expected to have correctly configured the Robotics Toolbox to proceed with the remainder of this laboratory experiment.

From this experiment you will learn the following:

- Usage of MATLAB Robotics Toolbox
- Reinforce the understanding of spatial descriptions and orientation representations.

2. Theory

This section describes the underlying theories associated with spatial descriptions and orientation representations.

2.1. Rotation Matrix

A rotation matrix is an $n \times n$ matrix that belongs to the special orthogonal group $SO(n)$ of order n. It can be used to represent relative orientation between two coordinate frames.

2D Rotation

Figure 2 shows two coordinate frames, with frame $o_1x_1y_1$ oriented at an angle θ with respect to frame $o_0x_0y_0$. The resulting rotation matrix is given by,

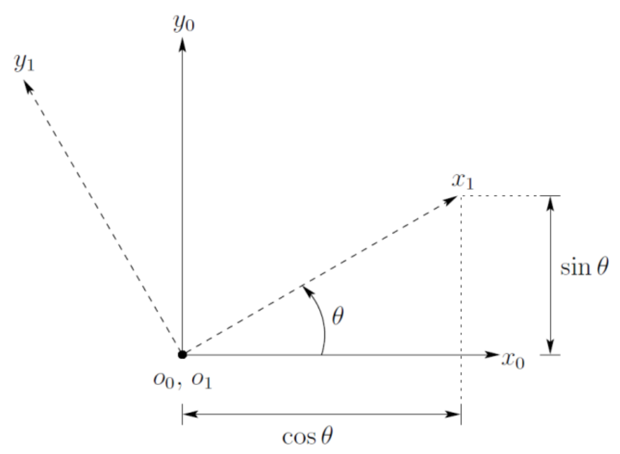

Figure 2: Coordinate frame $o_1x_1y_1$ is oriented at an angle θ with respect to $o_0x_0y_0$

$$R_1^0 = \begin{bmatrix} x_1 \cdot x_0 & y_1 \cdot x_0 \\ x_1 \cdot y_0 & y_1 \cdot x_0 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}. \tag{1}$$

3D Rotation

To obtain the rotation matrix for three dimensions, each axis of the frame $o_1x_1y_1$ is projected onto frame $o_0x_0y_0$. The resulting rotation matrix is given by,

$$R_1^0 = \begin{bmatrix} x_1 \cdot x_0 & y_1 \cdot x_0 & z_1 \cdot x_0 \\ x_1 \cdot y_0 & y_1 \cdot y_0 & z_1 \cdot y_0 \\ x_1 \cdot z_0 & y_1 \cdot z_0 & z_1 \cdot z_0 \end{bmatrix}. \tag{2}$$

For each axis, the basic rotation matrices are given by,

$$R_{z,\theta} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

$$R_{x,\theta} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \tag{4}$$

$$R_{y,\theta} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \tag{5}$$

2.2. **Rotational Transformations in Two Frames with Common Origin**

Rotation matrix R_1^0 can be used to transform the coordinates of a point from one frame to another.

$$p^0 = R_1^0 p^1 \tag{6}$$

2.3. **Rotation Matrix as an Operator**

Rotation matrix R can be operated on a vector (e.g. position vector) to rotate it in the same coordinate frame.

$$q^0 = Rp^0 \tag{7}$$

2.4. **Parameterization of Rotations**

The nine elements in a general SO(3) rotation matrix R are not independent. An arbitrary rotation can be represented using three independent quantities.

Euler Representation

In Z-Y-X Euler angles, a general rotation is given by the outcome of following sequence:

Rotate about current Z axis by ϕ angle Rotate about current Y axis by θ angle Rotate about current X axis by ψ angle

Fixed Angle Representation

In fixed angles, a general rotation is given by the outcome of following sequence:

Rotate about fixed X axis by ψ angle Rotate about fixed Y axis by θ angle Rotate about fixed Z axis by ϕ angle

Roll-Pitch-Yaw

There can be some confusion around the terms *roll-pitch-yaw angles* due to various definitions on different textbooks or toolboxes. Carefully go through the documentation and conduct simple experiments to figure out the sequence of rotations as well as the axes used.

3. Procedure

Follow the subsequent procedure using notation as described in section 5.

Spatial Descriptions

3.1. Visualize the default 2D coordinate frame $\{0\}$ in a MATLAB figure. Limit the plot area to X: $-4, 7$ and Y:$-2, 7$ and enable grid.

3.2. Consider a 2D point p with position vector $[5 \;\; 6]^T$ in frame $\{0\}$. Visualize p using a blue arrow.

3.3. Rotate the default coordinate frame $\{0\}$ counterclockwise by 45^0 using animation techniques. Visualize the new coordinate frame $\{1\}$ in red color. Find p in frame $\{1\}$.

3.4. Consider another 2D point q with position vector $[-3 \;\; 2]^T$ in frame $\{1\}$. Visualize q using a red arrow.

3.5. Apply a 68^0 counterclockwise rotation to the position vector of p to obtain the new 2D point r. Visualize r using a green arrow.

Orientation Representations

3.6. In a new MATLAB figure, visualize the default 3D coordinate frame $\{0\}$. Limit the plot area to $-1, 2$ for all X, Y, and Z directions.

3.7. Another 3D coordinate frame $\{1\}$ is obtained as follows:

> Rotate the default coordinate frame about X axis for $+15^0$.
> Rotate the current coordinate frame about the new Y axis for $+25^0$.
> Rotate the current coordinate frame about the new Z axis for $+35^0$.

Using suitable rotation functions, obtain 3x3 rotation matrices for each rotation, and the final rotation R_1^0.

Using animation techniques, visualize successive rotations that leads to R_1^0. Ultimately, visualize frame $\{1\}$ using red color.

3.8. Find the default roll-pitch-yaw angle definition for the toolbox. Note, however, that it is customizable using parameters.

3.9. For the following 3x3 rotation matrix (same matrix from the lecture), use the toolbox to find ψ (roll about X axis), θ (pitch about Y axis) and φ (yaw about Z axis) angles in degrees. Confirm your answer by doing the opposite conversion, and also using the product of basic rotation matrices.

$$R = \begin{bmatrix} 0.8138 & 0.0400 & 0.5798 \\ 0.2962 & 0.8298 & -0.4730 \\ -0.5000 & 0.5567 & 0.6634 \end{bmatrix}$$

4. MATLAB Robotics Toolbox Reference

4.1. Position Vector

`plot_arrow`

Draw an arrow in 2D or 3D

4.2. Rotation Matrix

`rot2`: SO(2) rotation matrix

`rot3`: SO(3) rotation matrix

4.3. Coordinate Frame

`trplot2`: 2D Coordinate frame

`trplot`: 3D Coordinate frame

Related commands

`hold on`: multiple frames can be added

`axis`: limit the plot area

`grid`: grid can be added to the plot

4.4. Animations

`tranimate2`: Animate a 2D coordinate frame

`tranimate`: Animate a 3D coordinate frame

↳ options:

`cleanup`: remove the frame at end of animation

4.5. Rotations

`rotx`: SO(3) rotation about X axis

`roty`: SO(3) rotation about Y axis

`rotz`: SO(3) rotation about Z axis

4.6. Conversions

`rpy2r`: roll-pitch-yaw angles to SO(3) rotation matrix

`tr2rpy`: convert SO(3) or SO(3) matrix to roll-pitch-yaw angles

5. Notation

Use a meaningful notation in MATLAB similar to the following:

Notation	Meaning
p_in_a	Point **p** represented in frame {a}
R_a_in_b	Orientation of frame {a} with respect to frame {b}
R_x_30	30^0 rotation about X axis

Answer Sheet

Index No:

1. MATLAB code for 3.1 ~ 3.5.

2. Final output MATLAB figure for the operations in 3.1 ~ 3.5.

3. p^1 for 3.3:

4. R_1^0 for 3.7.

5. MATLAB code for 3.6 ~ 3.9.

6. Final output MATLAB figure for the operations in 3.6 ~ 3.9.

7. Default roll-pitch-yaw angle definition for the toolbox.

8. For 3.9,
   ψ: ____________ θ: _______________φ: _______________