Representing Position and Orientation

Prof. Gerardo Flores

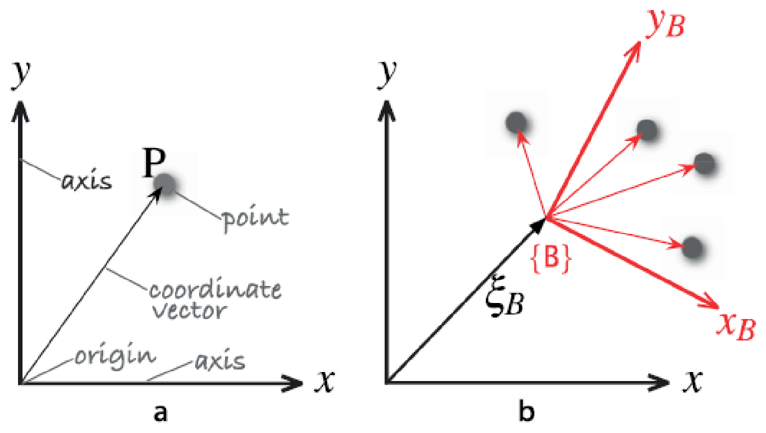
Aug 28, 2024

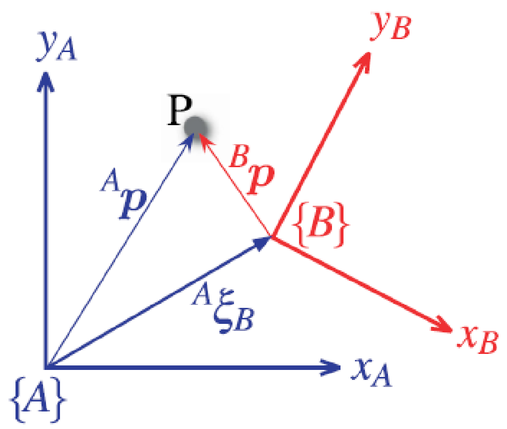
Robotics and Automation course,

TAMIU

Coordinated frames

Coordinate frame: It consists of an origin (or a simple point in space) and two or three coordinate axes for two- or three-dimensional spaces.





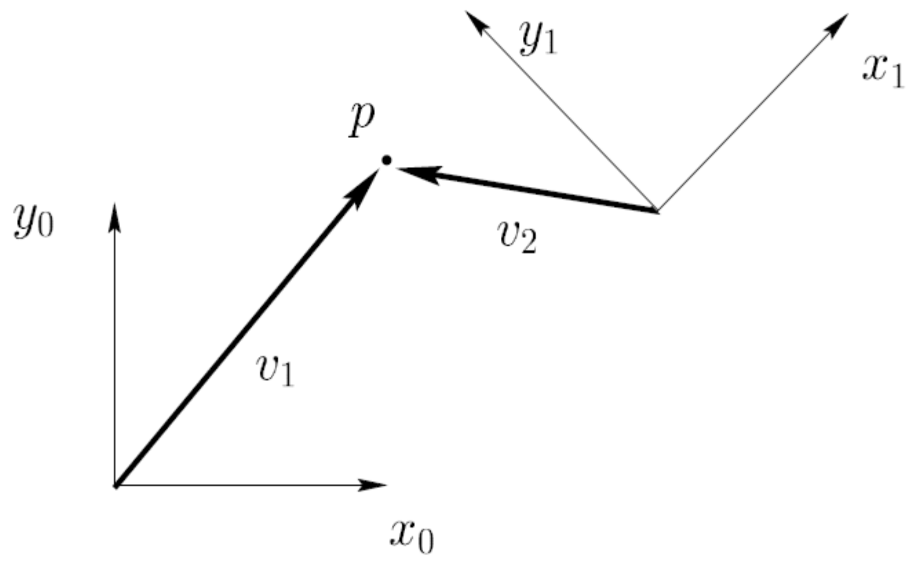


Fig. 2.1 Two coordinate frames, a point p, and two vectors v_1 and v_2.

Coordinated frames

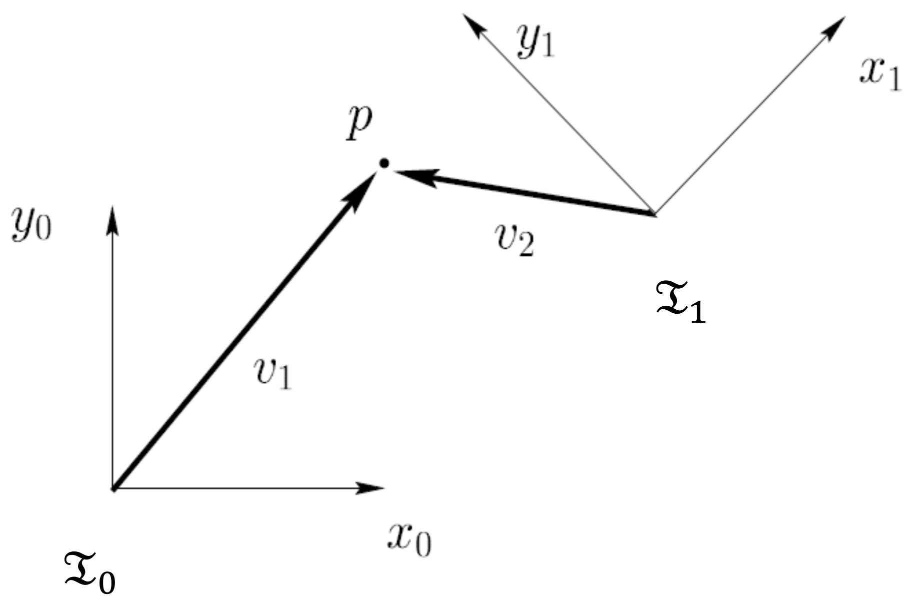


Fig. 2.1 Two coordinate frames, a point p, and two vectors v_1 and v_2.

Geometrically, a point corresponds to a specific location in space. We stress here that p is a geometric entity, a point in space, while both p^0 and p^1 are coordinate vectors that represent the location of this point in space with respect to coordinate frames $o_0x_0y_0$ and $o_1x_1y_1$, respectively.

- There can be as many coordinate frames as needed. In Figure 2.1, we have two coordinate frames, and the point p can be represented in either of these frames.
- The vectors v_1 and v_2 represent the position of p in frames $\mathfrak{T}_0$ and $\mathfrak{T}_1$, respectively.
- This idea can be repeated in the three dimensional space.
- The same point p thus is represented as follows:

$$p^0 = \begin{bmatrix} 5 \\ 6 \end{bmatrix}, \qquad p^1 = \begin{bmatrix} -2.8 \\ 4.2 \end{bmatrix}$$

Coordinated frames

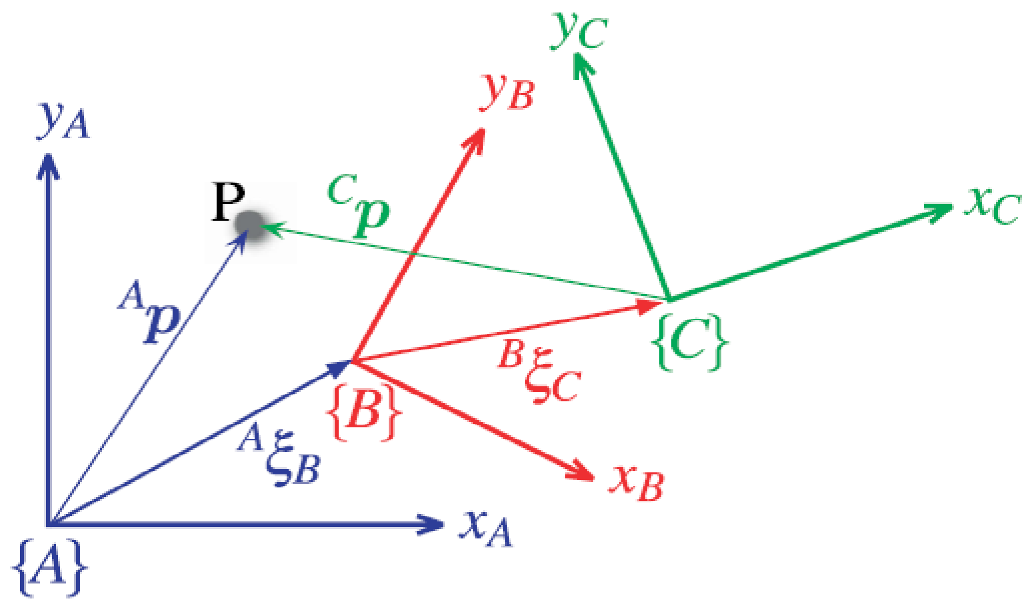
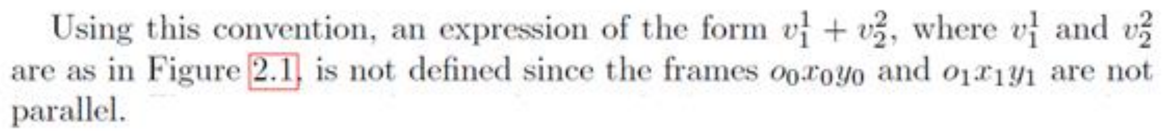


Fig. 2.3.
The point P can be described by coordinate vectors relative to either frame $\{A\}$, $\{B\}$ or $\{C\}$. The frames are described by relative poses

Fig. 2.3 Example with three coordinate frames.

Coordinated frames


Using this convention, an expression of the form $v_1^1 + v_2^2$, where v_1^1 and v_2^2 are as in Figure 2.1, is not defined since the frames $o_0x_0y_0$ and $o_1x_1y_1$ are not parallel.

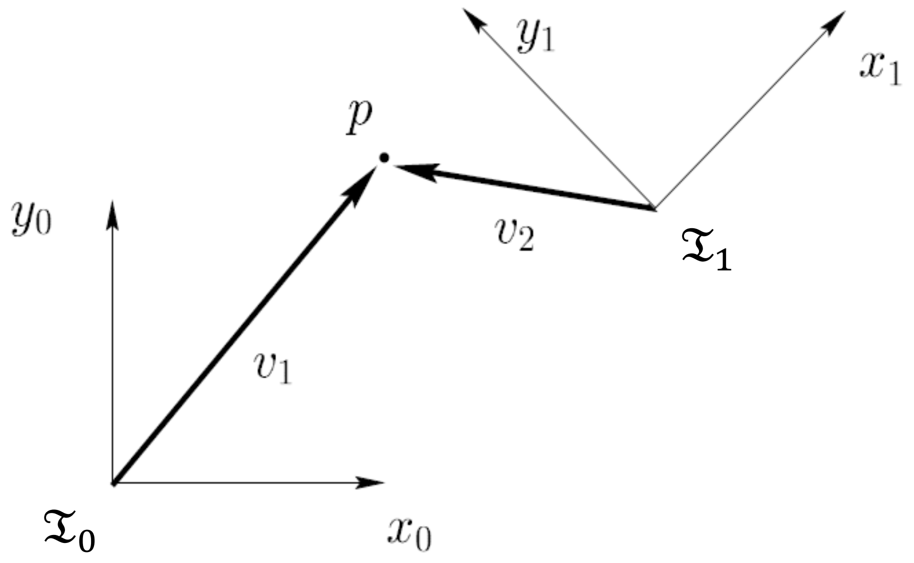
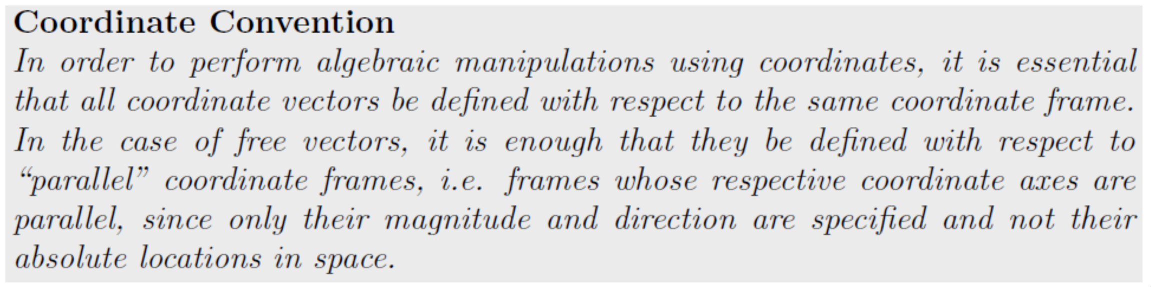


Fig. 2.1 Two coordinate frames, a point p, and two vectors v_1 and v_2.


Coordinate Convention
In order to perform algebraic manipulations using coordinates, it is essential that all coordinate vectors be defined with respect to the same coordinate frame. In the case of free vectors, it is enough that they be defined with respect to "parallel" coordinate frames, i.e. frames whose respective coordinate axes are parallel, since only their magnitude and direction are specified and not their absolute locations in space.

- We can represent each of the two vectors in Fig. 2.1 in each of the frames $\mathfrak{T}_0$ and $\mathfrak{T}_1$ as described by:

$$v_1^0 = \begin{bmatrix} 5 \\ 6 \end{bmatrix}, \quad v_1^1 = \begin{bmatrix} 7.77 \\ 0.8 \end{bmatrix}, \quad v_2^0 = \begin{bmatrix} -5.1 \\ 1 \end{bmatrix}, \quad v_2^1 = \begin{bmatrix} -2.89 \\ 4.2 \end{bmatrix}$$

- We have used the notation:

$$v_{\text{vector index}}^{\text{frame}}$$

Using this convention, an expression of the form $v_1^1 + v_2^0$ *is not defined* since the frames $\mathfrak{T}_0$ and $\mathfrak{T}_1$ are not parallel

Coordinated frames

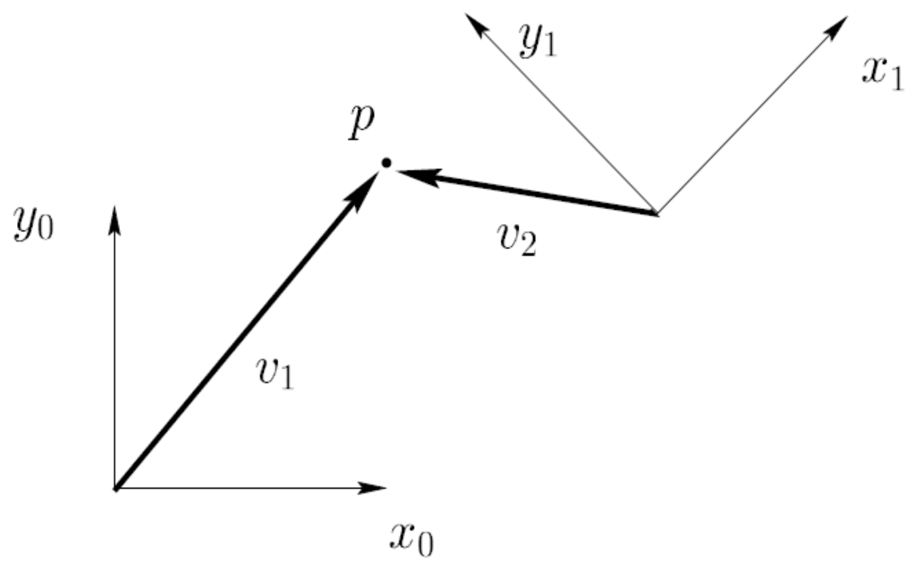


Fig. 2.1 Two coordinate frames, a point p, and two vectors v_1 and v_2.

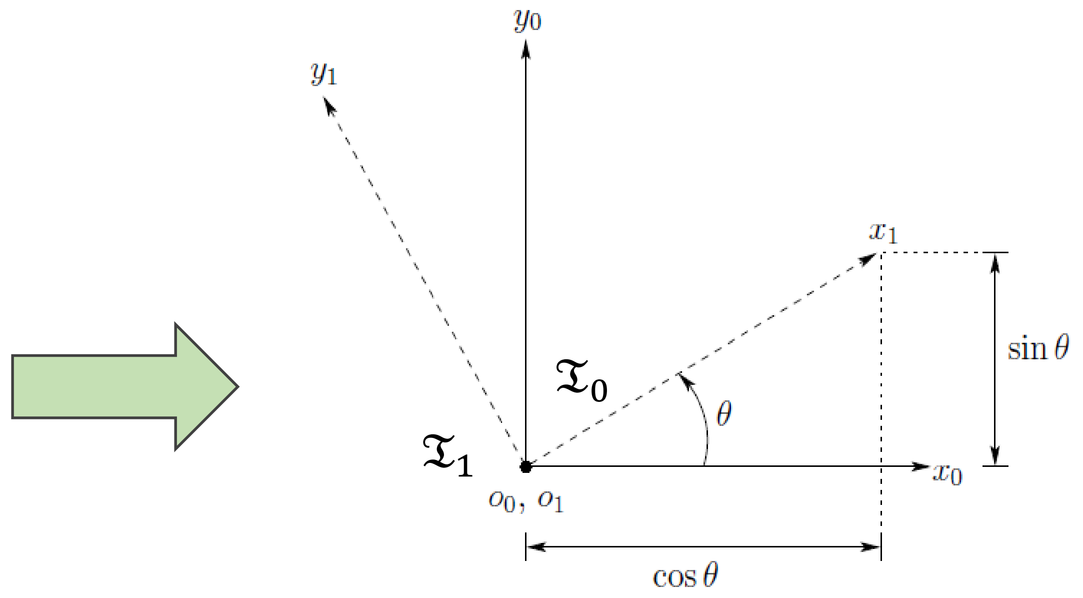
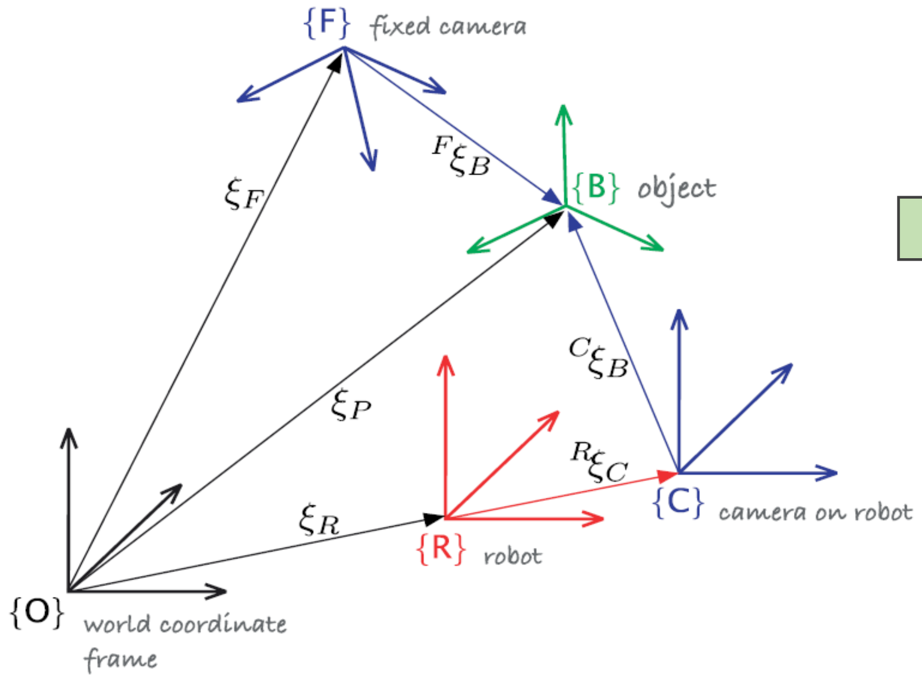
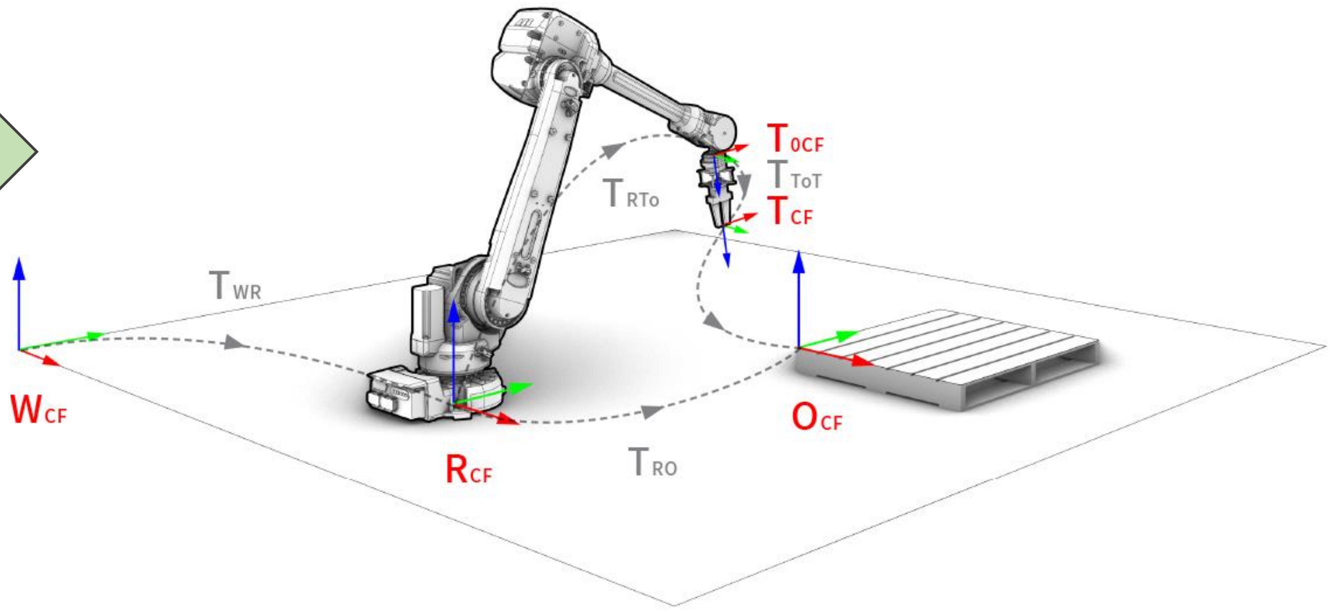


Fig. 2.2 Coordinate frame $\mathfrak{T}_1$ is oriented at an angle θ with respect to frame $\mathfrak{T}_0$.

Coordinated frames





Rotations in the plane

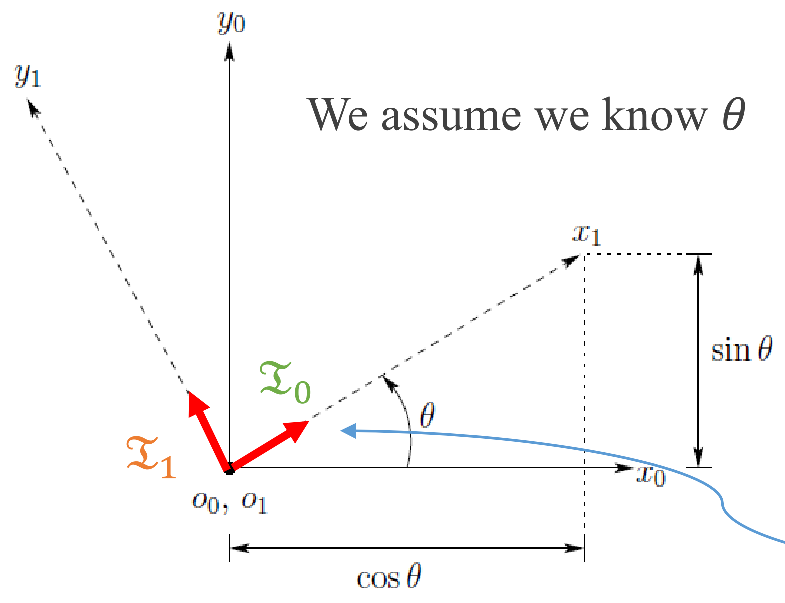


Fig. 2.2 Coordinate frame $\mathfrak{T}_1$ is oriented at an angle θ with respect to frame $\mathfrak{T}_0$.

The goal is to represent the relative orientation of the frames $\mathfrak{T}_0$ and $\mathfrak{T}_1$.

- The most convenient way of it is to specify the coordinate vectors for the axes of frame $\mathfrak{T}_1$ w.r.t. coordinate frame $\mathfrak{T}_0$ as follows:

$$R_1^0 = [x_1^0 \mid y_1^0] \quad \text{Rotation matrix}$$

where x_1^0 and y_1^0 are the coordinates in the frame $\mathfrak{T}_0$ of **unit vectors** (x_1, y_1).

$$x_1^0 = \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}, \qquad y_1^0 = \begin{bmatrix} -\sin\theta \\ \cos\theta \end{bmatrix}$$

The **superscript** denotes the reference frame.

$$R_1^0 = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{2.1}$$

Rotations in the plane

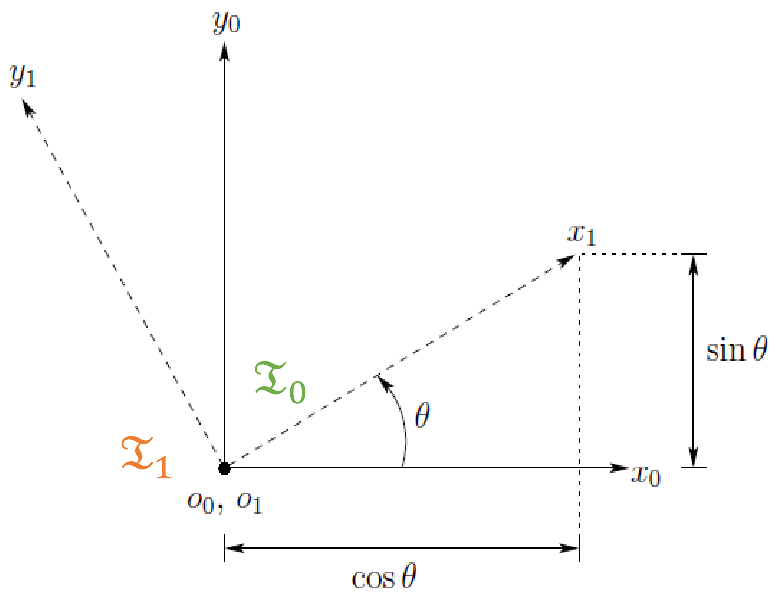


Fig. 2.2 Coordinate frame $\mathfrak{T}_1$ is oriented at an angle θ with respect to frame $\mathfrak{T}_0$.

Rotation matrix

$$R_1^0 = [x_1^0 \mid y_1^0] = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

Thus, such a rotation matrix transforms vectors from frame $\mathfrak{T}_1$ to frame $\mathfrak{T}_0$, or in other words, it expresses vectors originally in frame $\mathfrak{T}_1$ on frame $\mathfrak{T}_0$.

To use rotation matrices, we need to know the angles between frames, which can be measured using **sensors** such as IMU.

Rotations in the plane

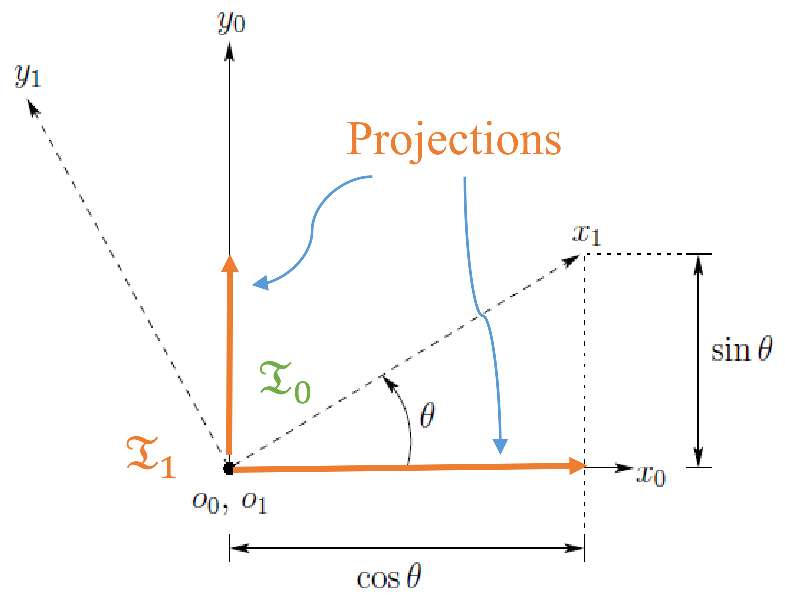
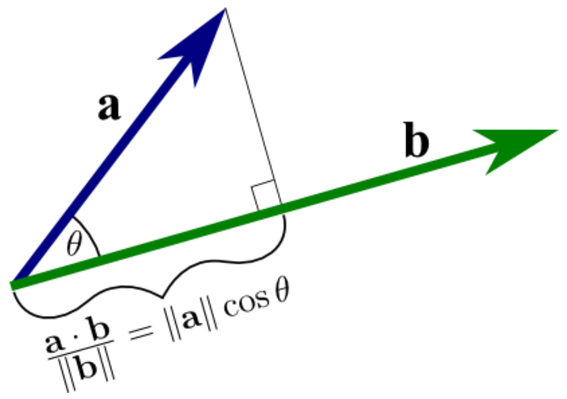


Fig. 2.2 Coordinate frame $\mathfrak{T}_1$ is oriented at an angle θ with respect to frame $\mathfrak{T}_0$.

Rotation matrix

- Another way of building rotational matrices is:
  - Projecting the axes of frame $\mathfrak{T}_1$ onto the coordinate axes of frame $\mathfrak{T}_0$.
  - For that, we use the **dot product**.

Dot product of two **unit vectors** gives the projection of one onto the other



$\mathbf{a}$
$\mathbf{b}$
θ
$\frac{\mathbf{a}\cdot\mathbf{b}}{\|\mathbf{b}\|} = \|\mathbf{a}\|\cos\theta$

When the vectors are not unitary, we use:

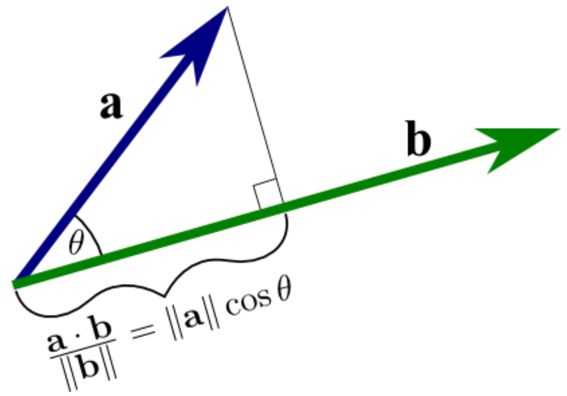
$$proj_b a = \left(\frac{a\cdot b}{\|b\|^2}\right) b \longrightarrow$$ Projection of a onto b.

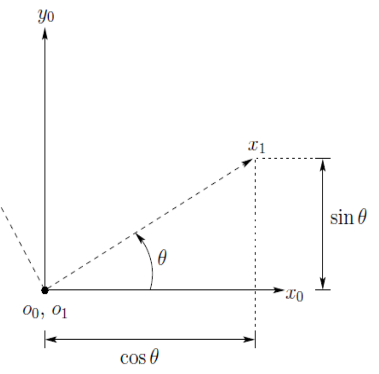
Rotations in the plane

Rotation matrix

- Another way of building rotational matrices is:
  - Projecting the axes of frame $\mathfrak{T}_1$ onto the coordinate axes of frame $\mathfrak{T}_0$.
  - For that, we use the **dot product**.

Dot product of two **unit vectors** gives the projection of one onto the other


a
b
θ
$\frac{\mathbf{a}\cdot\mathbf{b}}{\|\mathbf{b}\|} = \|\mathbf{a}\|\cos\theta$



Projection of x_1 onto x_0

- Using the dot product we obtain:

$$x_1^0 = \begin{bmatrix} x_1 \cdot \boxed{x_0} \\ x_1 \cdot \boxed{y_0} \end{bmatrix}, \qquad y_1^0 = \begin{bmatrix} y_1 \cdot \boxed{x_0} \\ y_1 \cdot \boxed{y_0} \end{bmatrix}$$

- Which can be combined to obtain the rotation matrix:

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

Question: How do we get this from the above?

$$R_1^0 = [x_1^0 \mid y_1^0] = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

Rotations in the plane

- Similarly, we can get the rotation matrix that describes the frame $\mathfrak{T}_0$ with respect the frame $\mathfrak{T}_1$

$$R_0^1 = \begin{bmatrix} x_0 \cdot x_1 & y_0 \cdot x_1 \\ x_0 \cdot y_1 & y_0 \cdot y_1 \end{bmatrix}$$

- c

Question: How R_1^0 and R_0^1 are related to each other?

$$R_1^0 = \begin{bmatrix} x_1 \cdot x_0 & y_1 \cdot x_0 \\ x_1 \cdot y_0 & y_1 \cdot y_0 \end{bmatrix} \quad R_0^1 = \begin{bmatrix} x_0 \cdot x_1 & y_0 \cdot x_1 \\ x_0 \cdot y_1 & y_0 \cdot y_1 \end{bmatrix}$$

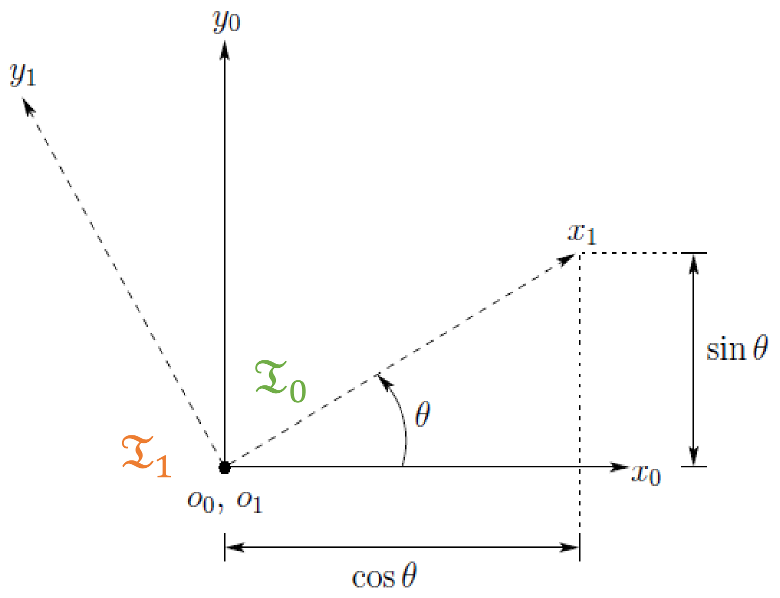


Fig. 2.2 Coordinate frame $\mathfrak{T}_1$is oriented at an angle θ with respect to frame $\mathfrak{T}_0$.

Develop a solution to the problem and perform a comparative analysis using Python.

Rotations in the plane

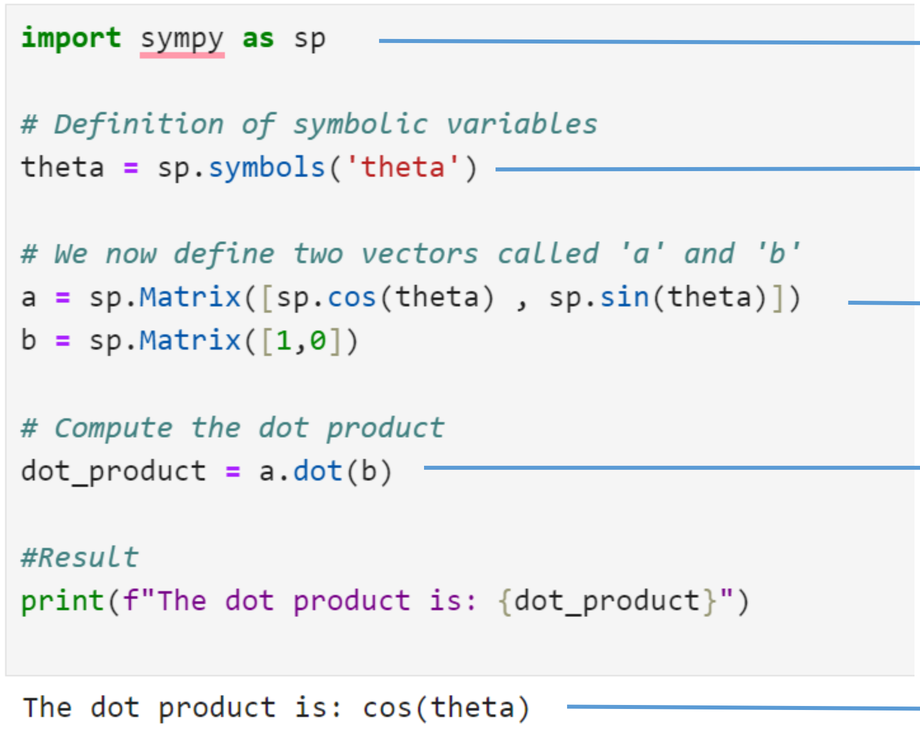
Excercises:

1. Compute the rotation matrix using symbolic variables, then compute its transpose and inverse. Analyze the results and discuss their similarities.

2. Write a code in Python that plots a vector in two-dimensional space along with its rotation by 45 degrees.

Rotations in the plane

Example of dot product between vectors $a = [\cos\theta \ , \ \sin\theta]^T$ and $b = [1 \ , \ 0]^T$ in Python.
It is recommended to use Jupyter notebook:
https://jupyter.org/try-jupyter/notebooks/?path=notebooks/Intro.ipynb



```python
import sympy as sp

# Definition of symbolic variables
theta = sp.symbols('theta')

# We now define two vectors called 'a' and 'b'
a = sp.Matrix([sp.cos(theta) , sp.sin(theta)])
b = sp.Matrix([1,0])

# Compute the dot product
dot_product = a.dot(b)

#Result
print(f"The dot product is: {dot_product}")
```

```
The dot product is: cos(theta)
```

We use symbolic variables since one of the vectors is a function of the variable θ.

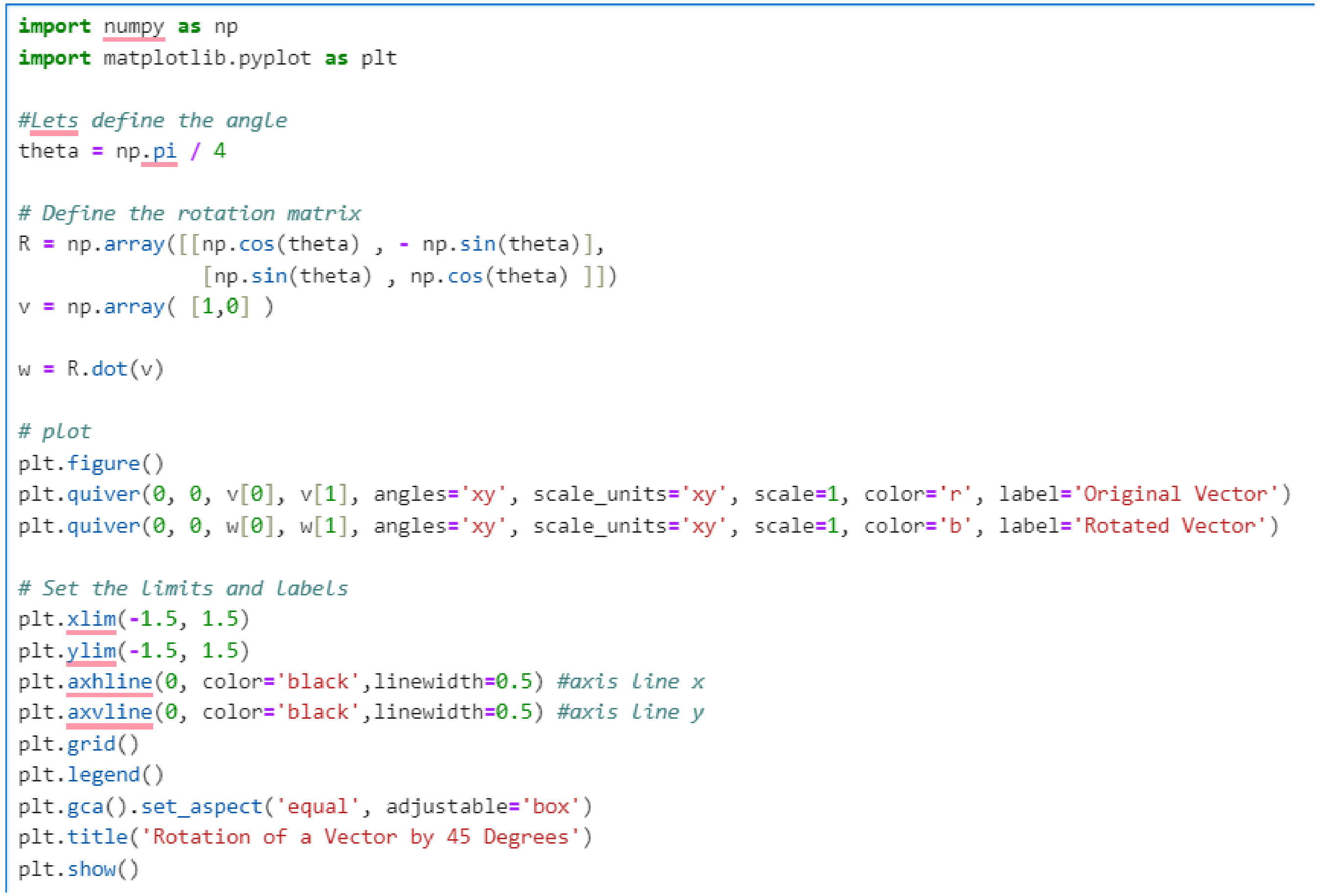
Theta is defined as a **symbolic variable**.

Vectors definition.

Dot product operation.

Result.

Example



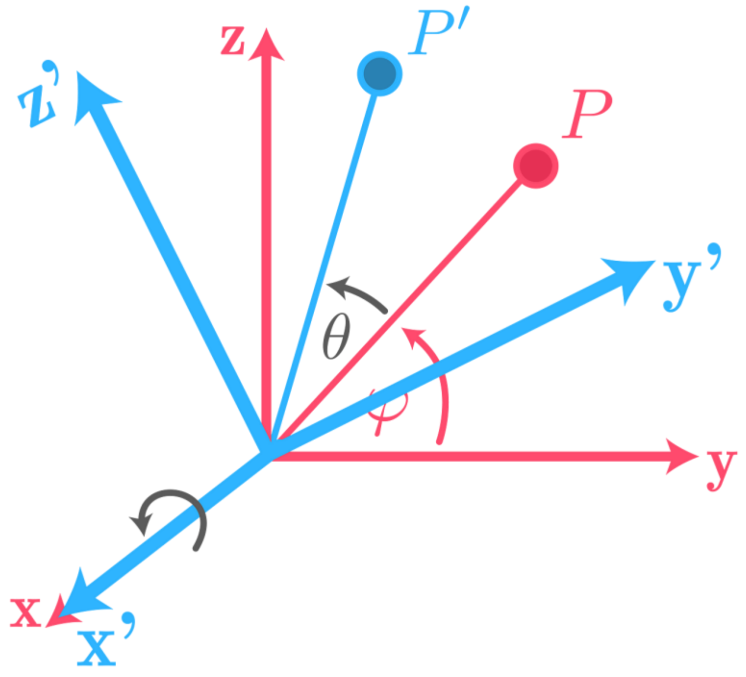
```python
import numpy as np
import matplotlib.pyplot as plt

#Lets define the angle
theta = np.pi / 4

# Define the rotation matrix
R = np.array([[np.cos(theta) , - np.sin(theta)],
               [np.sin(theta) , np.cos(theta) ]])
v = np.array( [1,0] )

w = R.dot(v)

# plot
plt.figure()
plt.quiver(0, 0, v[0], v[1], angles='xy', scale_units='xy', scale=1, color='r', label='Original Vector')
plt.quiver(0, 0, w[0], w[1], angles='xy', scale_units='xy', scale=1, color='b', label='Rotated Vector')

# Set the limits and labels
plt.xlim(-1.5, 1.5)
plt.ylim(-1.5, 1.5)
plt.axhline(0, color='black',linewidth=0.5) #axis line x
plt.axvline(0, color='black',linewidth=0.5) #axis line y
plt.grid()
plt.legend()
plt.gca().set_aspect('equal', adjustable='box')
plt.title('Rotation of a Vector by 45 Degrees')
plt.show()
```

Rotations in three dimensions

- The projection technique described above scales nicely to the 3D case:

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

- Each axis of the frame $\mathfrak{T}_1$ is projected onto coordinate frame $\mathfrak{T}_0$.



Rotations in three dimensions

Example

Suppose the frame $o_1x_1y_1z_1$ is rotated through an angle θ about the z_0 −axis and it is desired to find the resulting transformation matrix R_1^0.

By convention the positive sense of the angle θ about the axis z follows the right-hand rule.

1. Compute R_1^0 by hand and verify your result in Python.
2. Show a figure of such a rotation in Python.

Solution

$$R_1^0 = \begin{bmatrix} x_1 \cdot x_0 & y_1 \cdot x_0 & z_1 \cdot x_0 \\ x_1 \cdot y_0 & y_1 \cdot y_0 & z_1 \cdot y_0 \\ x_1 \cdot z_0 & y_1 \cdot z_0 & z_1 \cdot z_0 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$x_1 \cdot x_0 = \cos\theta, \qquad y_1 \cdot x_0 = -\sin\theta, \qquad z_0 \cdot z_1 = 1$$
$$x_1 \cdot y_0 = \sin\theta, \qquad y_1 \cdot y_0 = \cos\theta$$

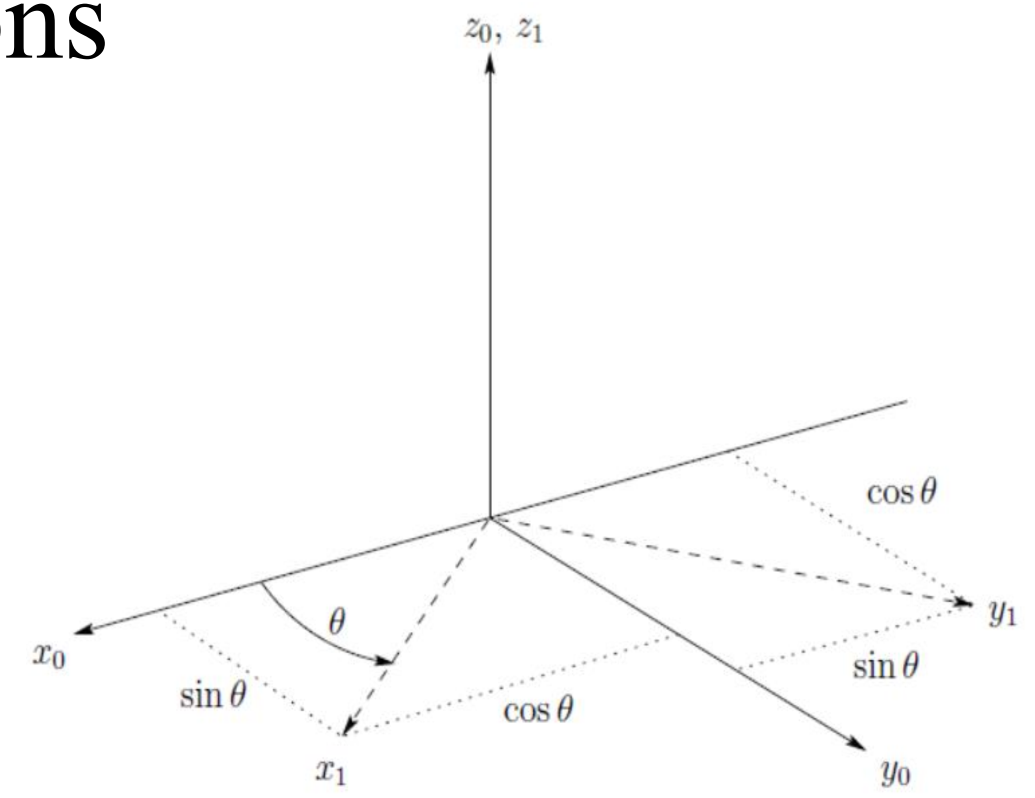


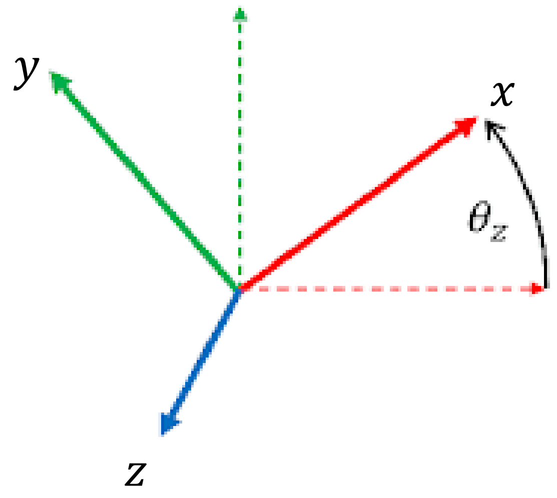
Fig. 2.3 Rotation about z_0 by an angle θ.

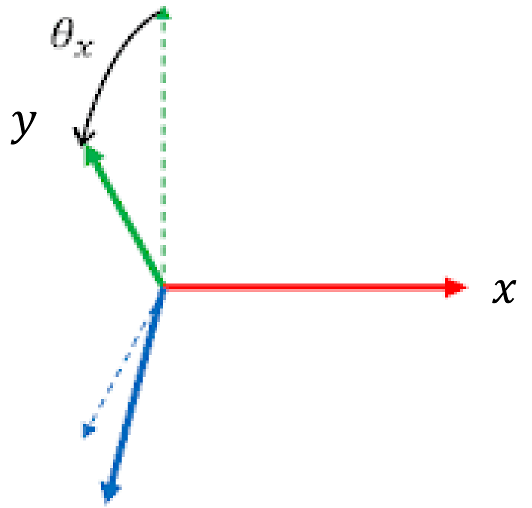
Properties of rotation matrices

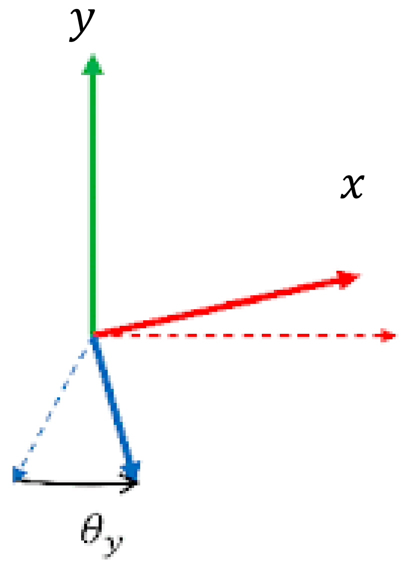
Properties of the Matrix Group $SO(n)$

- $R \in SO(n)$
- $R^{-1} \in SO(n)$
- $R^{-1} = R^T$
- The columns (and therefore the rows) of R are mutually orthogonal
- Each column (and therefore each row) of R is a unit vector
- $\det R = 1$

Basic rotation matrices







$$R_{z,\theta} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} = R_1^0, \quad R_{x,\theta} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}, \quad R_{y,\theta} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

- Notice that

$$R_{z,0} = I$$
$$R_{z,\theta} R_{z,\phi} = R_{z,\theta+\phi}$$
$$\left(R_{z,\theta}\right)^{-1} = R_{z,-\theta}$$

Rotational tranformations

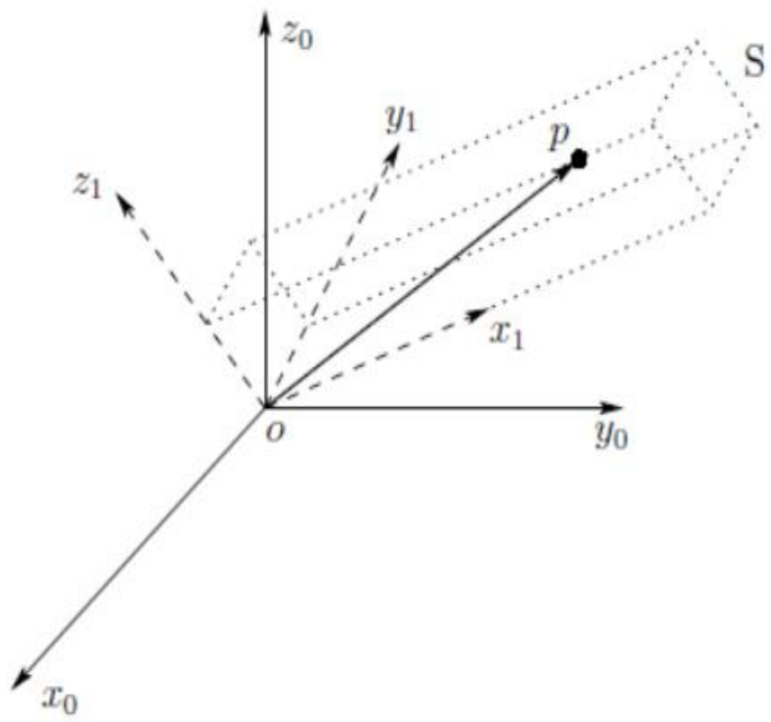


Fig. 2.5 Coordinate frame attached to a rigid body.

The goal is to express p^1 in the frame p^0. Where $p^1 = \begin{bmatrix} u \\ v \\ w \end{bmatrix}$. For that, we simply use the rotation matrix that represents or models the rotation of frame $o_1x_1y_1z_1$ with respect to frame $o_0x_0y_0z_0$, i. e., R_1^0.

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

Compositions of rotations

Suppose we add a third coordinate frame $o_2x_2y_2z_2$

A given point p can thus be represented by coordinated specified with respect to any of these frames:

- $o_0x_0y_0z_0 \longrightarrow p^0$
- $o_1x_1y_1z_1 \longrightarrow p^1$
- $o_2x_2y_2z_2 \longrightarrow p^2$

The relationship is:

$$p^0 = R_1^0 p^1$$
$$p^1 = R_2^1 p^2$$
$$p^0 = R_2^0 p^2$$

c

We combine the first two equations and we get:

$$p^0 = R_1^0 R_2^1 p^2$$

From where we can deduce that

$$R_2^0 = \cancel{R_1^0 R_2^1}$$

Which is the composition law for rotational transformations

Example

- Suppose a rotation matrix R represents a rotation of angle ϕ about the current y −axis followed by a rotation of angle θ about the current z −axis. Compute the matrix R.

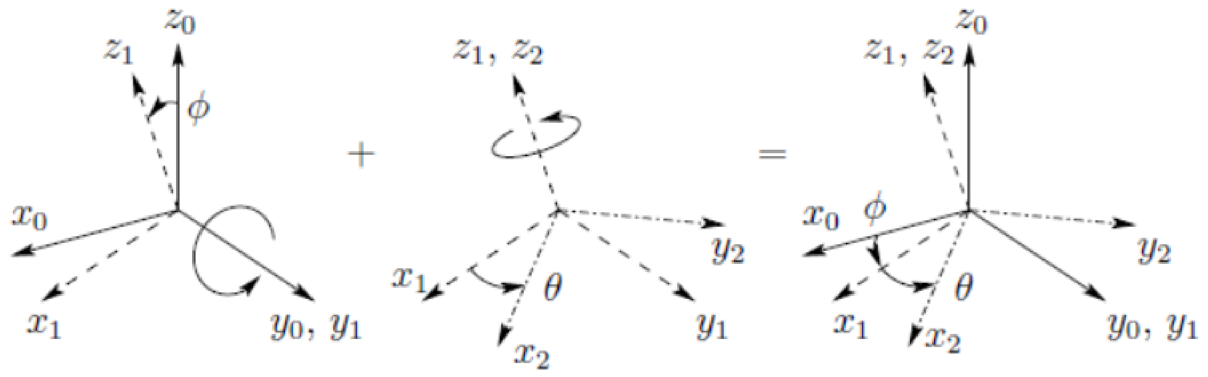


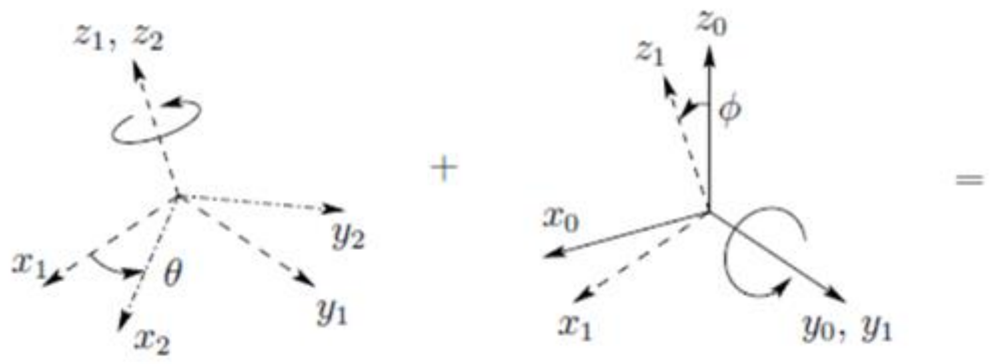
Fig. 2.9 Composition of rotations about current axes.

$$\begin{aligned} R &= R_{y,\phi} R_{z,\theta} \\ &= \begin{bmatrix} c_\phi & 0 & s_\phi \\ 0 & 1 & 0 \\ -s_\phi & 0 & c_\phi \end{bmatrix} \begin{bmatrix} c_\theta & -s_\theta & 0 \\ s_\theta & c_\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} c_\phi c_\theta & -c_\phi s_\theta & s_\phi \\ s_\theta & c_\theta & 0 \\ -s_\phi c_\theta & s_\phi s_\theta & c_\phi \end{bmatrix} \end{aligned}$$

- Write a Python code that calculates that matrix R.

Example

- Suppose that the above rotations are performed in the reverse order, that is, first a rotation about the current z −axis followed by a rotation about the current y −axis. Compute the R'.



$$
\begin{aligned}
R' &= R_{z,\theta}R_{y,\phi} \\
&= \begin{bmatrix} c_\theta & -s_\phi & 0 \\ s_\theta & c_\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_\phi & 0 & s_\phi \\ 0 & 1 & 0 \\ -s_\phi & 0 & c_\phi \end{bmatrix} \\
&= \begin{bmatrix} c_\theta c_\phi & -s_\theta & c_\theta s_\phi \\ s_\theta c_\phi & c_\theta & s_\theta s_\phi \\ -s_\phi & 0 & c_\phi \end{bmatrix}
\end{aligned}
$$

- Write a Python code that calculates that matrix R' **Are they the same?**

Euler angles

- A common method of specifying a rotation matrix in terms of three independent quantities is to use **Euler Angles** (ϕ, θ, ψ).
- Since we have three angles, we need three succesive rotations as follows:
  1. First rotate about the z-axis by the angle ϕ.
  2. Next rotate about the current y-axis by the angle θ.
  3. Finally, rotate about the current z-axis by the angle ψ.

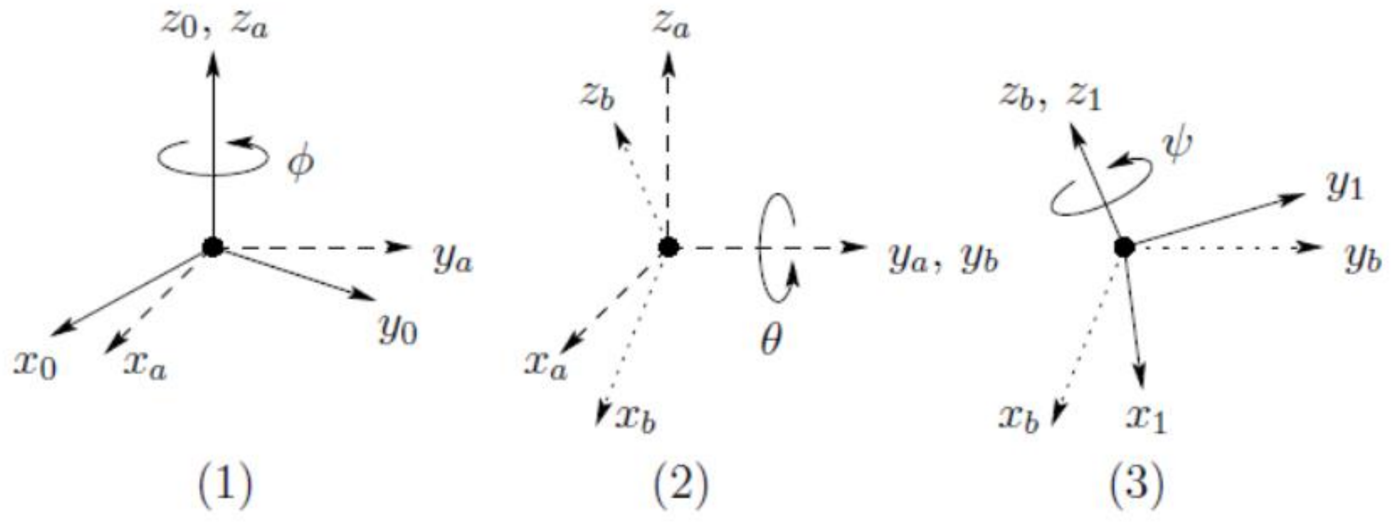


Fig. 2.11 Euler angle representation.

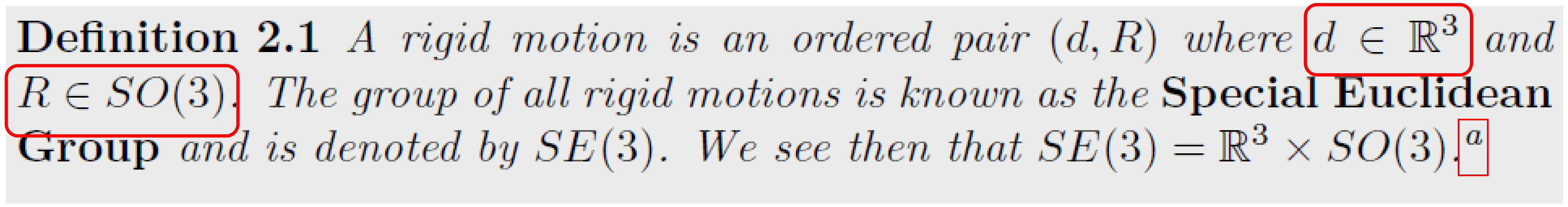
Euler angles

- The resulting rotational transformation can be generated as follows:

$$\left.\begin{aligned}
R_{ZYZ} &= R_{z,\phi}R_{y,\theta}R_{z,\psi} \\
&= \begin{bmatrix} c_\phi & -s_\phi & 0 \\ s_\phi & c_\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix}\begin{bmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
&= \boxed{\begin{bmatrix} c_\phi c_\theta c_\psi - s_\phi s_\psi & -c_\phi c_\theta s_\psi - s_\phi c_\psi & c_\phi s_\theta \\ s_\phi c_\theta c_\psi + c_\phi s_\psi & -s_\phi c_\theta s_\psi + c_\phi c_\psi & s_\phi s_\theta \\ -s_\theta c_\psi & s_\theta s_\psi & c_\theta \end{bmatrix}}
\end{aligned}\right\} \text{ZYZ – Euler angle tranformation}$$

Rigid motions

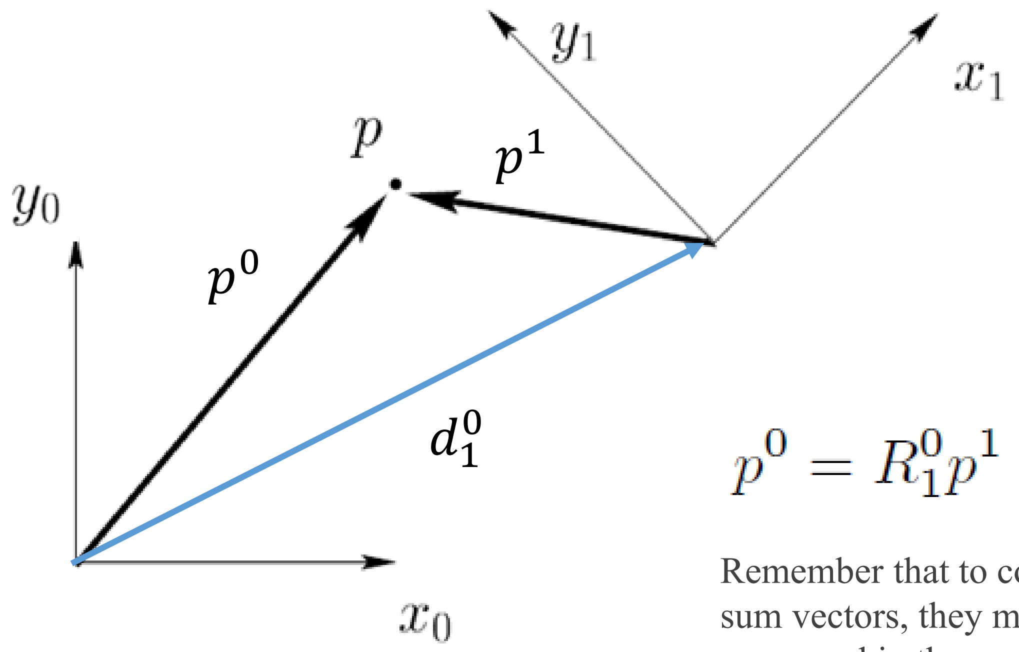
- We have seen how to represent positions and orientations. We now combine these two concepts to define a **rigid motion**.



Definition 2.1 *A rigid motion is an ordered pair* (d, R) *where* $\boxed{d \in \mathbb{R}^3}$ *and* $\boxed{R \in SO(3)}$. *The group of all rigid motions is known as the* **Special Euclidean Group** *and is denoted by* $SE(3)$. *We see then that* $SE(3) = \mathbb{R}^3 \times SO(3)$.[a]

- Therefore, a rigid motion is a **pure translation** together with a **pure rotation**.

Rigid motions



$$p^0 = R_1^0 p^1 + d_1^0$$

Remember that to correctly sum vectors, they must first be expressed in the same coordinate frame. If vectors are in different frames, you must transform them to a common coordinate system before performing the addition.

- If we have two rigid motions

$$p^0 = R_1^0 p^1 + d_1^0 \qquad p^1 = R_2^1 p^2 + d_2^1$$

Then, we can get a third rigid motion as follows,

$$p^0 = \underbrace{R_1^0 R_2^1}_{R_2^0} p^2 + \underbrace{R_1^0 d_2^1 + d_1^0}_{d_2^0}$$

$$p^0 = \underbrace{R_2^0 p^2}_{\text{Pure rotation}} + \underbrace{d_2^0}_{\text{Pure translation}}$$

Homogeneous transformations

$$p^0 = \underbrace{R_1^0 R_2^1 p^2}_{R_2^0} + \underbrace{R_1^0 d_2^1 + d_1^0}_{d_2^0}$$

$$\downarrow$$

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

where 0 denotes the row vector $(0, 0, 0)$

Thus, the rigid motions can be represented by the set of matrices of the form:

$$\boxed{H = \begin{bmatrix} R & d \\ 0 & 1 \end{bmatrix}; R \in SO(3), \; d \in \mathbb{R}^3}$$

$$\downarrow$$

Homogeneous transformations

Q: What are the dimensions of H?

Homogeneous transformations

- Problem: Remember that

$$H = \begin{bmatrix} R & d \\ 0 & 1 \end{bmatrix}; R \in SO(3),\ d \in \mathbb{R}^3$$

Then, prove that

$$H^{-1} = \begin{bmatrix} R^T & -R^T d \\ 0 & 1 \end{bmatrix}$$

Also, corroborate your result in Python.

Key: Think about the definition of the inverse of a matrix.

Homogeneous transformations

- Think about the previous transformation

$$p^0 \quad = \quad R^0_1 p^1 + d^0_1$$

- In order to c that transformation by matrix multiplication, we must augment the vectors p^0, p^1 as follows:

$$\underbrace{P^0 \quad = \quad \begin{bmatrix} p^0 \\ 1 \end{bmatrix} \qquad P^1 \quad = \quad \begin{bmatrix} p^1 \\ 1 \end{bmatrix}}_{\text{Homogeneous representations of vectors}} \quad \Rightarrow \quad \boxed{P^0 \quad = \quad H^0_1 P^1}$$

Homogeneous transformations

- A set of basic homogeneous transformations generating $SE(3)$ is given byv

$$\mathrm{Trans}_{x,a} = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad Rot_{x,\alpha} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_\alpha & -s_\alpha & 0 \\ 0 & s_\alpha & c_\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathrm{Trans}_{y,b} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad Rot_{y,\beta} = \begin{bmatrix} c_\beta & 0 & s_\beta & 0 \\ 0 & 1 & 0 & 0 \\ -s_\beta & 0 & c_\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathrm{Trans}_{z,c} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad Rot_{x,\gamma} = \begin{bmatrix} c_\gamma & -s_\gamma & 0 & 0 \\ s_\gamma & c_\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Homogeneous transformations

- Problem: Compute

$$\begin{aligned} H &= Rot_{x,\alpha} Trans_{x,b} Trans_{z,d} Rot_{z,\theta} \\ &= \begin{bmatrix} c_\theta & -s_\theta & 0 & b \\ c_\alpha s_\theta & c_\alpha c_\theta & -s_\alpha & -d s_\alpha \\ s_\alpha s_\theta & s_\alpha c_\theta & c_\alpha & d c_\alpha \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

Homogeneous transformations

- In computer vision we also use homogenous transformations:

$$H = \left[\begin{array}{c|c} R_{3\times 3} & d_{3\times 1} \\ \hline \underbrace{f_{1\times 3}}_{\boxed{0 \quad 0 \quad 0}} & \underbrace{s_{1\times 1}}_{\boxed{1}} \end{array}\right] = \left[\begin{array}{c|c} Rotation & Translation \\ \hline perspective & scale\ factor \end{array}\right]$$

For our purposes we always take the last row vector of H to be $(0, 0, 0, 1)$.

Problems

1. Using the fact that $v_1 \cdot v_2 = v_1^T v_2$, show that the dot product of two free vectors does not depend on the choice of frames in which their coordinates are defined.

Problems

2. Show that the length of a free vector is not changed by rotation, i.e., that $\|v\| = \|Rv\|$.

Problems

3. Show that the distance between points is not changed by rotation i.e., that $\|p_1 - p_2\| = \|Rp_1 - Rp_2\|$.

Problems

4. If a matrix R satisfies $R^T R = I$, show that the column vectors of R are of unit length and mutually perpendicular.

Problems

5. If a matrix R satisfies $R^T R = I$, then
   a) show that $\det R = \pm 1$
   b) Show that $\det R = \pm 1$ if we restrict ourselves to right-handed coordinate systems.

Problems

16. Suppose that three coordinate frames $o_1x_1y_1z_1$, $o_2x_2y_2z_2$ and $o_3x_3y_3z_3$ are given, and suppose

$$R_2^1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2} & -\frac{\sqrt{3}}{2} \\ 0 & \frac{\sqrt{3}}{2} & \frac{1}{2} \end{bmatrix}; R_3^1 = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

Find the matrix R_3^2.

$$R_3^2 = R_1^2 R_3^1 \quad \text{where} \quad R_1^2 = (R_2^1)^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1/2 & \sqrt{3}/2 \\ 0 & -\sqrt{3}/2 & 1/2 \end{bmatrix}.$$

Therefore,

$$R_3^2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1/2 & \sqrt{3}/2 \\ 0 & -\sqrt{3}/2 & 1/2 \end{bmatrix} \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & -1 \\ \sqrt{3}/2 & 1/2 & 0 \\ 1/2 & -\sqrt{3}/2 & 0 \end{bmatrix}$$