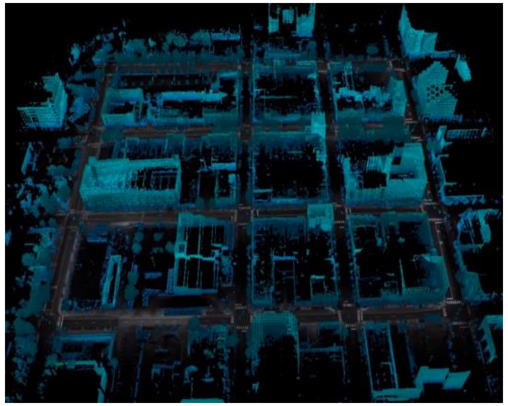
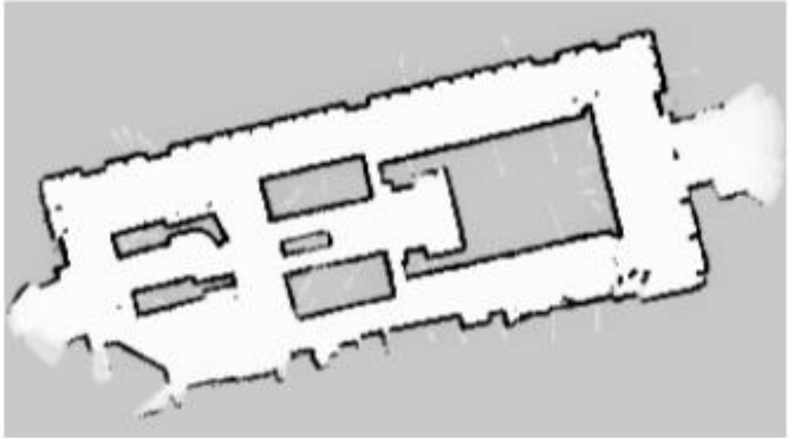
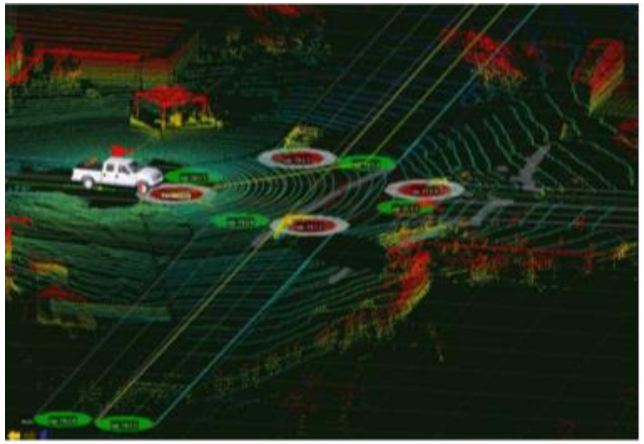
BYU Electrical & Computer Engineering





RIGID BODY TRANSFORMATIONS

ECEN 633: Robotic Localization and Mapping

Where is Everything?

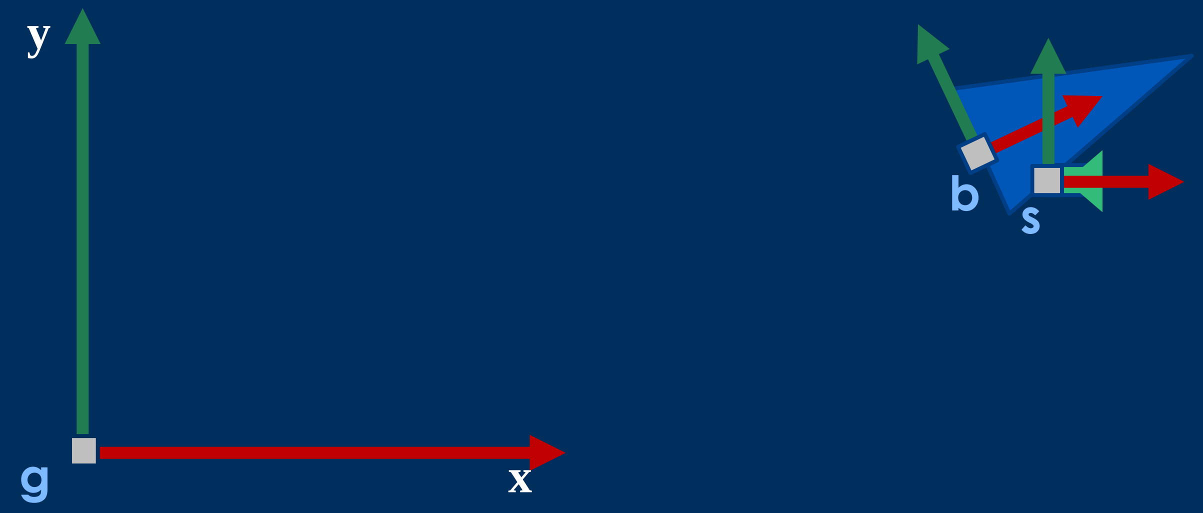
- In robotics, we need to know where things are!
  - The Robot/Vehicle
  - Other Objects/Agents
  - Sensors

  - Why do we need to know where the robot is?
  - What other things do we need to know the location of?
  - How do we find out where potential pedestrians/other vehicles are?
  - The data from the sensor tells us where they are with respect to what?

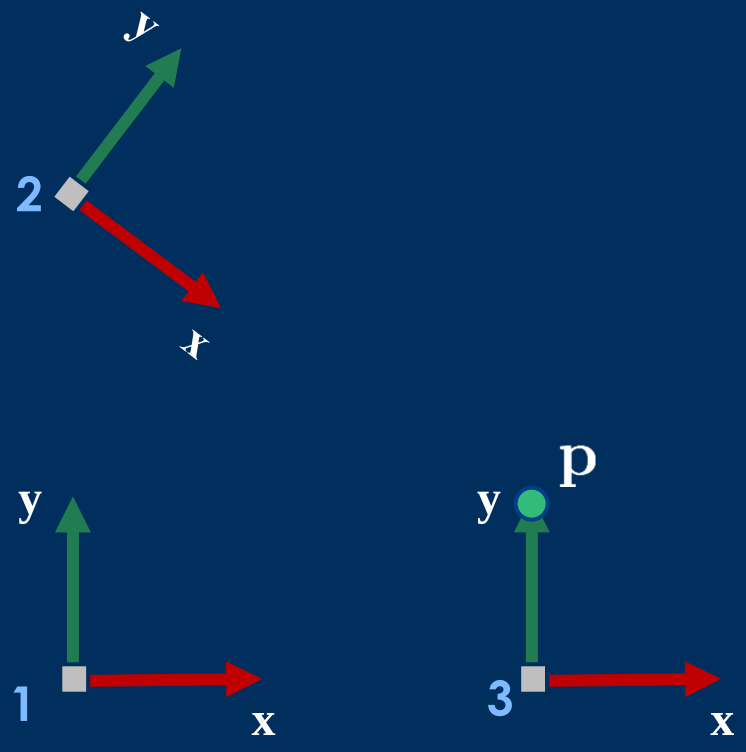
Images: Waymo

Coordinate Frames (Basis in linear algebra)

- We use coordinate frames to represent where things are in space.
- Commonly used frames:
  - Global/inertial frame – somewhere fixed in the world (**g**)
  - Robot/robot "base" frame – somewhere fixed on robot (**b or r**)
  - Sensor frame – fixed relative to the robot (**s**)



Coordinate Frames – Details: Representing Positions

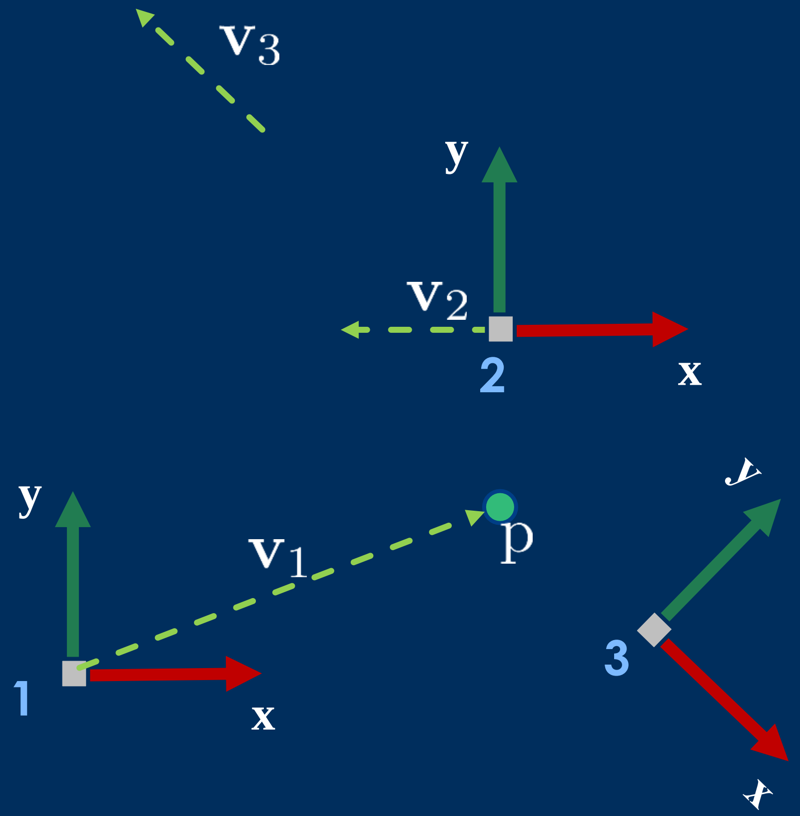


$$^{1}\mathbf{p} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$$

$$^{2}\mathbf{p} = \begin{bmatrix} 4 \\ 0 \end{bmatrix}$$

$$^{3}\mathbf{p} = \quad ?$$

Coordinate Frames – Details: Representing Vectors



Vectors – Direction and Magnitude
Used for displacement, force, velocity, etc.

$$^{1}\mathbf{v}_1 = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$$

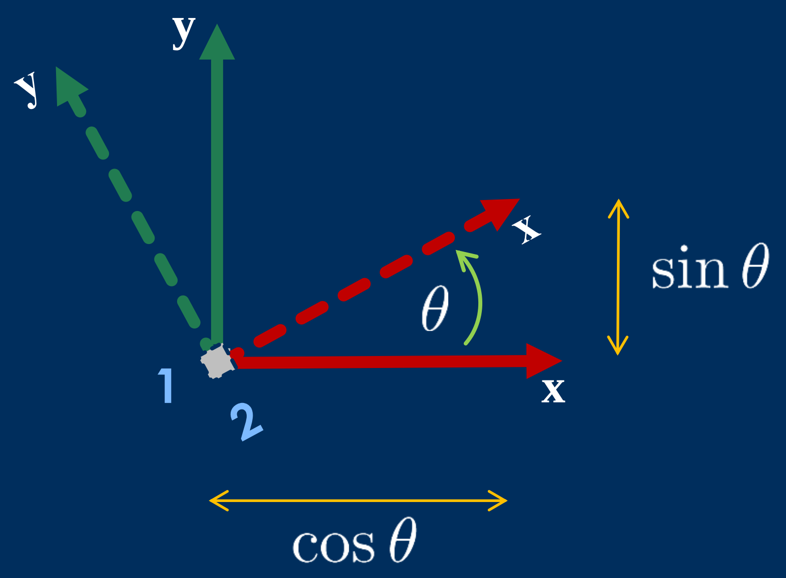
$$^{1}\mathbf{v}_2 = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$$

$$^{1}\mathbf{v}_3 = \quad ?$$

$$^{2}\mathbf{v}_2 = \quad ?$$

$$^{3}\mathbf{v}_2 = \quad ?$$

Coordinate Frames – Details: Representing Rotations



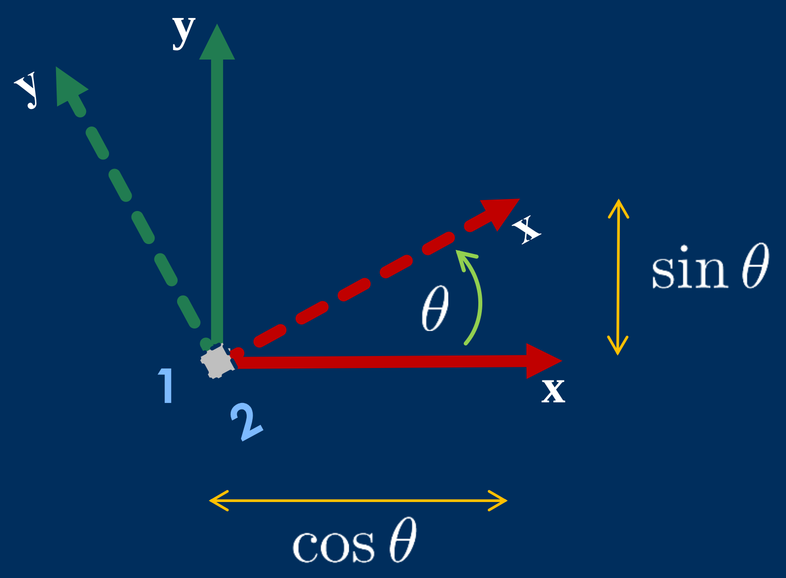
Methods of Representing Rotation:

- Change in angle θ
  - Cons:
    - Theta wraps at 2 pi (non-continuous)
    - Doesn't scale to 3D very well
- Rotation Matrix

$$R_2^1 = \left[\ {}^1x_2 \ \middle|\ {}^1y_2\ \right] = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

  - R_2^1 is a matrix whose column vectors are the unit vectors of frame 2 with respect to frame 1.

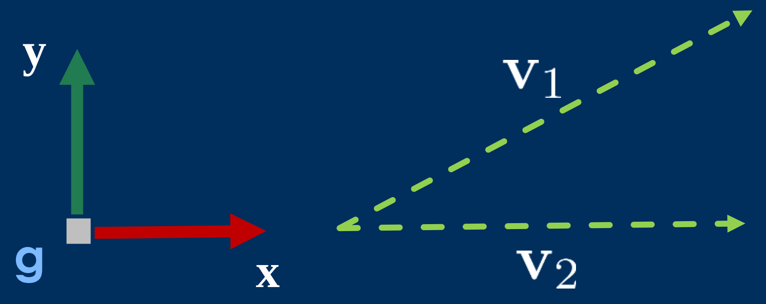
Coordinate Frames – Details: Rotation Matrices



Rotation Matrix via Projection:

- Dot product of two unit-vectors projects one onto the other

Reminder – Dot Product



$$\mathbf{a} = \begin{bmatrix} a^x \\ a^y \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} b^x \\ b^y \end{bmatrix}$$

$$\mathbf{a} \cdot \mathbf{b} = \sum_i a^i b^i$$
$$= a^x b^x + a^y b^y$$

$${}^g\mathbf{v}_1 = \begin{bmatrix} {}^g v_1^x \\ {}^g v_1^y \end{bmatrix} = \begin{bmatrix} 3 \\ 1.5 \end{bmatrix}$$

$${}^g\mathbf{v}_2 = \begin{bmatrix} {}^g v_2^x \\ {}^g v_2^y \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \end{bmatrix}$$

$${}^g\mathbf{v}_1 \cdot {}^g\mathbf{v}_2 = 3 * 3 + 0 * 1.5 = 9$$

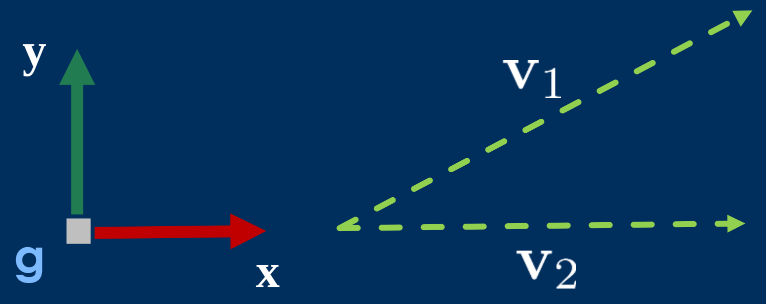
Measures how much two vectors are pointing in same direction:

- If not at all dot product is 0
- Maximized if two vectors are parallel

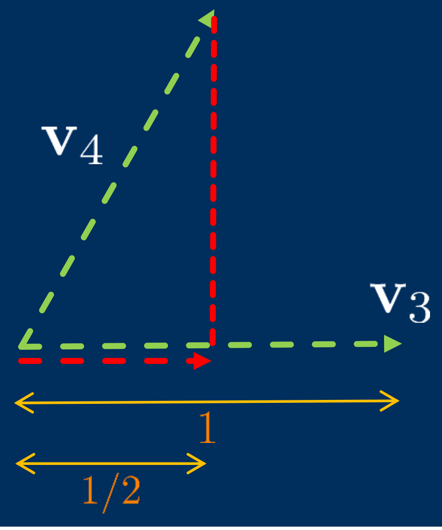
Reminder – Dot Product

$$\mathbf{a} = \begin{bmatrix} a^x \\ a^y \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} b^x \\ b^y \end{bmatrix}$$

$$\mathbf{a} \cdot \mathbf{b} = \sum_i a^i b^i = a^x b^x + a^y b^y$$



Unit Vector Projection Example:

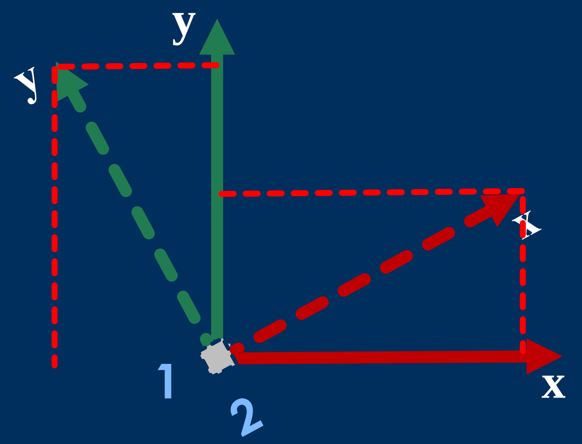


$$^g\mathbf{v}_3 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad ^g\mathbf{v}_4 = \begin{bmatrix} 1/2 \\ \sqrt{3}/2 \end{bmatrix}$$

$$^g\mathbf{v}_3 \cdot {}^g\mathbf{v}_4 = 1 * 1/2 + 0 * \sqrt{3}/2 = 1/2$$

When vectors are unit vectors, the dot product "projects" one onto the other or determines how much of 1 vector is along the direction of the other!

Coordinate Frames – Details: Rotation Matrices



Rotation Matrix via Projection:

- Dot product of two unit-vectors projects one onto the other

$$R_2^1 = \left[\ {}^1\mathbf{x}_2 \ \middle|\ {}^1\mathbf{y}_2\ \right]$$

$${}^1\mathbf{x}_2 = \begin{bmatrix} \mathbf{x}_2 \cdot \mathbf{x}_1 \\ \mathbf{x}_2 \cdot \mathbf{y}_1 \end{bmatrix} \quad {}^1\mathbf{y}_2 = \begin{bmatrix} \mathbf{y}_2 \cdot \mathbf{x}_1 \\ \mathbf{y}_2 \cdot \mathbf{y}_1 \end{bmatrix}$$

$$R_2^1 = \left[\begin{array}{c|c} \mathbf{x}_2 \cdot \mathbf{x}_1 & \mathbf{y}_2 \cdot \mathbf{x}_1 \\ \mathbf{x}_2 \cdot \mathbf{y}_1 & \mathbf{y}_2 \cdot \mathbf{y}_1 \end{array}\right]$$

Coordinate Frames – Details: Rotation Matrices

$$R_2^1 = \left[\begin{array}{c|c} \mathbf{x}_2 \cdot \mathbf{x}_1 & \mathbf{y}_2 \cdot \mathbf{x}_1 \\ \mathbf{x}_2 \cdot \mathbf{y}_1 & \mathbf{y}_2 \cdot \mathbf{y}_1 \end{array}\right]$$

Do the same thing reversed to find:

$$R_1^2 = \left[\begin{array}{c|c} \mathbf{x}_1 \cdot \mathbf{x}_2 & \mathbf{y}_1 \cdot \mathbf{x}_2 \\ \mathbf{x}_1 \cdot \mathbf{y}_2 & \mathbf{y}_1 \cdot \mathbf{y}_2 \end{array}\right]$$

Dot product is commutative, so:

$$\mathbf{x}_i \mathbf{y}_j = \mathbf{y}_j \mathbf{x}_i$$

And,

$$R_1^2 = (R_2^1)^\top$$

R_1^2 is the geometric inverse of R_2^1, so

$$(R_2^1)^\top = (R_2^1)^{-1}$$

Rotation Matrix Properties:

- $(R)^\top = (R)^{-1}$ (Transpose is inverse)

Spong - Ch2

Coordinate Frames – Details: Rotation Matrices

$$R_2^1 = \left[\begin{array}{c|c} \mathbf{x}_2 \cdot \mathbf{x}_1 & \mathbf{y}_2 \cdot \mathbf{x}_1 \\ \mathbf{x}_2 \cdot \mathbf{y}_1 & \mathbf{y}_2 \cdot \mathbf{y}_1 \end{array}\right]$$

Column vectors of R_2^1 are unit-length and mutually orthogonal.

Thus, R is an **orthogonal** matrix.

Rotation Matrix Properties:

- $(R)^\top = (R)^{-1}$ (Transpose is inverse)
- R is **orthogonal**
  - Columns (and rows) are mutually orthogonal
  - Each column and row is a unit vector

Spong - Ch2

Coordinate Frames – Details: Rotation Matrices

$$R_2^1 = \left[\begin{array}{c|c} \mathbf{x}_2 \cdot \mathbf{x}_1 & \mathbf{y}_2 \cdot \mathbf{x}_1 \\ \mathbf{x}_2 \cdot \mathbf{y}_1 & \mathbf{y}_2 \cdot \mathbf{y}_1 \end{array} \right]$$

Determinant of R_2^1 can only be positive or negative 1.

If restrict to right-handed coordinate frames (for 3D rotations), then $\det R_2^1 = +1$

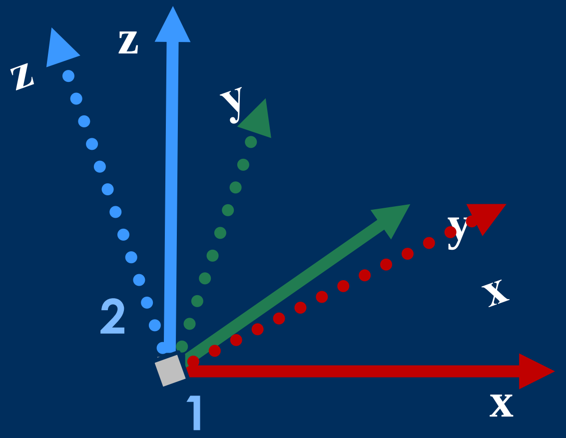
Under this restriction, rotation matrices with dimension $n \times n$ are part of the **Special Orthogonal Group** and are referred to with the symbol $\mathrm{SO}(n)$

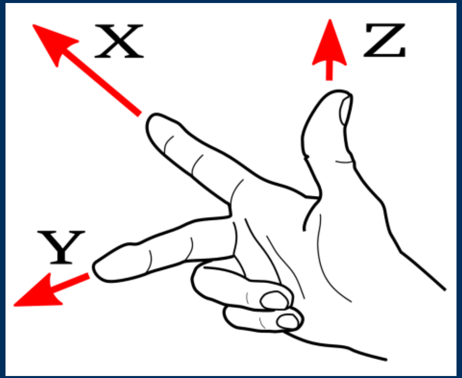
Thus, for any $R \in \mathrm{SO}(n)$:

- $(R)^\top = (R)^{-1}$ (Transpose is inverse)
- R is **orthogonal**
  - Columns (and rows) are mutually orthogonal
  - Each column and row is a unit vector
- $\det R = +1$

Spong - Ch2

Coordinate Frames – Details: Rotations in 3D





Right-handed Coordinate Frames

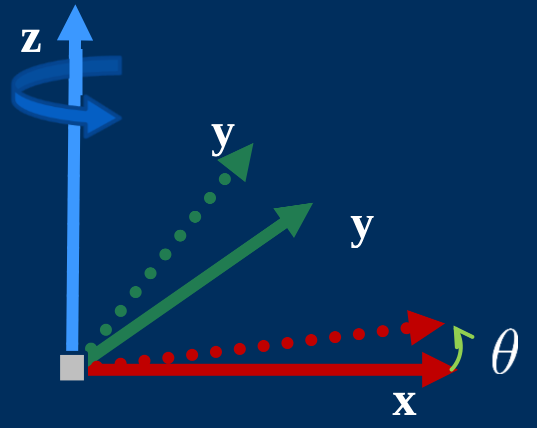
Rotation matrices in 3D are elements of $\mathrm{SO}(3)$

- Via projections technique:

$$R_2^1 = \left[\begin{array}{c|c|c} \mathbf{x}_2 \cdot \mathbf{x}_1 & \mathbf{y}_2 \cdot \mathbf{x}_1 & \mathbf{z}_2 \cdot \mathbf{x}_1 \\ \mathbf{x}_2 \cdot \mathbf{y}_1 & \mathbf{y}_2 \cdot \mathbf{y}_1 & \mathbf{z}_2 \cdot \mathbf{y}_1 \\ \mathbf{x}_2 \cdot \mathbf{z}_1 & \mathbf{y}_2 \cdot \mathbf{z}_1 & \mathbf{z}_2 \cdot \mathbf{z}_1 \end{array}\right]$$

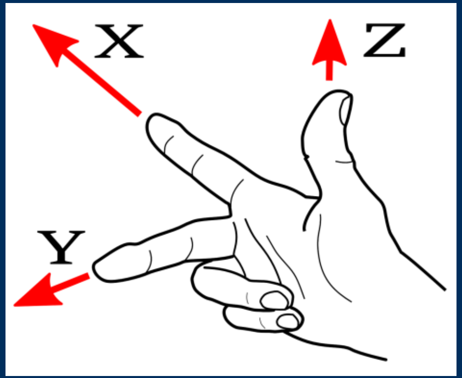
Image: Drew Noakes (stackoverflow)

Coordinate Frames – Details: Rotations in 3D

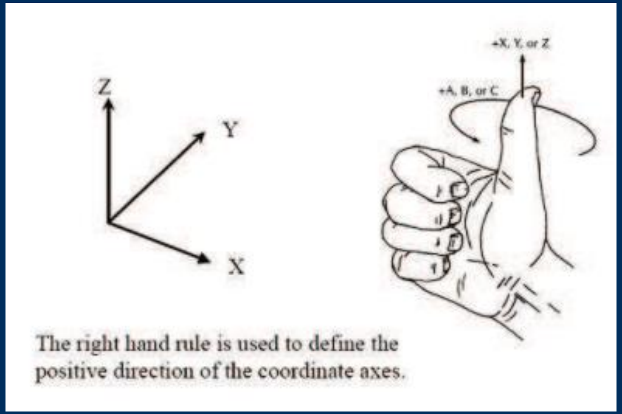


$$R_2^1 = \begin{bmatrix} \mathbf{x}_2 \cdot \mathbf{x}_1 & \mathbf{y}_2 \cdot \mathbf{x}_1 & \mathbf{z}_2 \cdot \mathbf{x}_1 \\ \mathbf{x}_2 \cdot \mathbf{y}_1 & \mathbf{y}_2 \cdot \mathbf{y}_1 & \mathbf{z}_2 \cdot \mathbf{y}_1 \\ \mathbf{x}_2 \cdot \mathbf{z}_1 & \mathbf{y}_2 \cdot \mathbf{z}_1 & \mathbf{z}_2 \cdot \mathbf{z}_1 \end{bmatrix}$$

$$R_2^1 = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



Right-handed Coordinate Frames

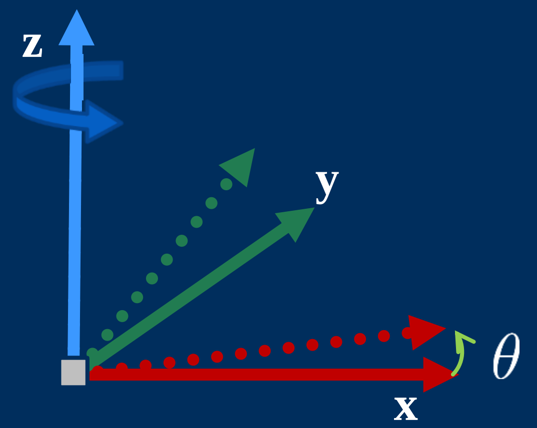


Positive Rotations

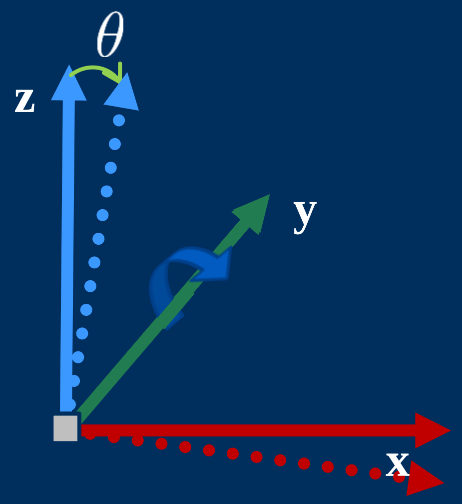
Image: Matthew Peet (control.asu.edu)

Coordinate Frames – Details: Rotations in 3D

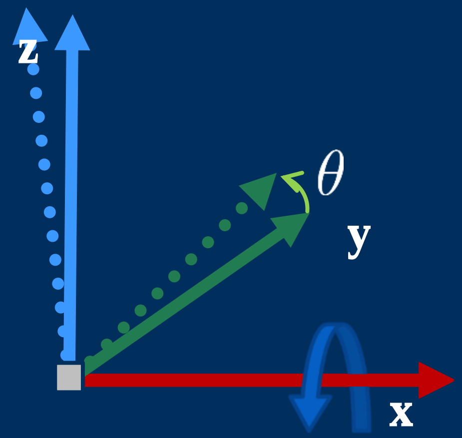
Rotation about Z-axis



Rotation about Y-axis

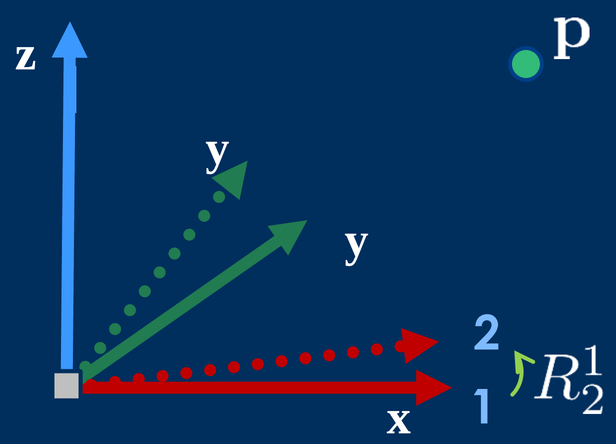


Rotation about X-axis



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

Coordinate Frames – Details: Changing Point Frame of Ref.



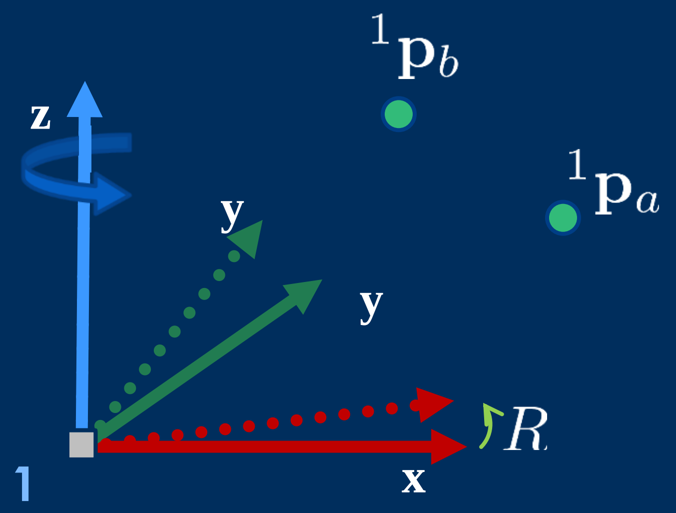
$$^1\mathbf{p} = \begin{bmatrix} 3 \\ 1 \\ 0 \end{bmatrix}$$

$$^2\mathbf{p} = R_1^2 \ {}^1\mathbf{p}$$

Represent frame 2 with
respect to frame 1

Rotation Matrix Use Case 1:
Can be used to transform coordinate representation of a point from one frame to another.

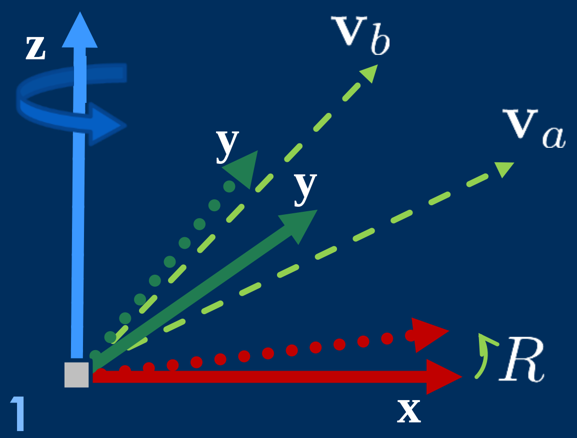
Coordinate Frames – Details: Rotating Points



$$ {}^{1}\mathbf{p}_b = R\, {}^{1}\mathbf{p}_a $$

Rotation Matrix Use Case 2:
Can be used to representation a rotation about a single frame and apply it to a point.

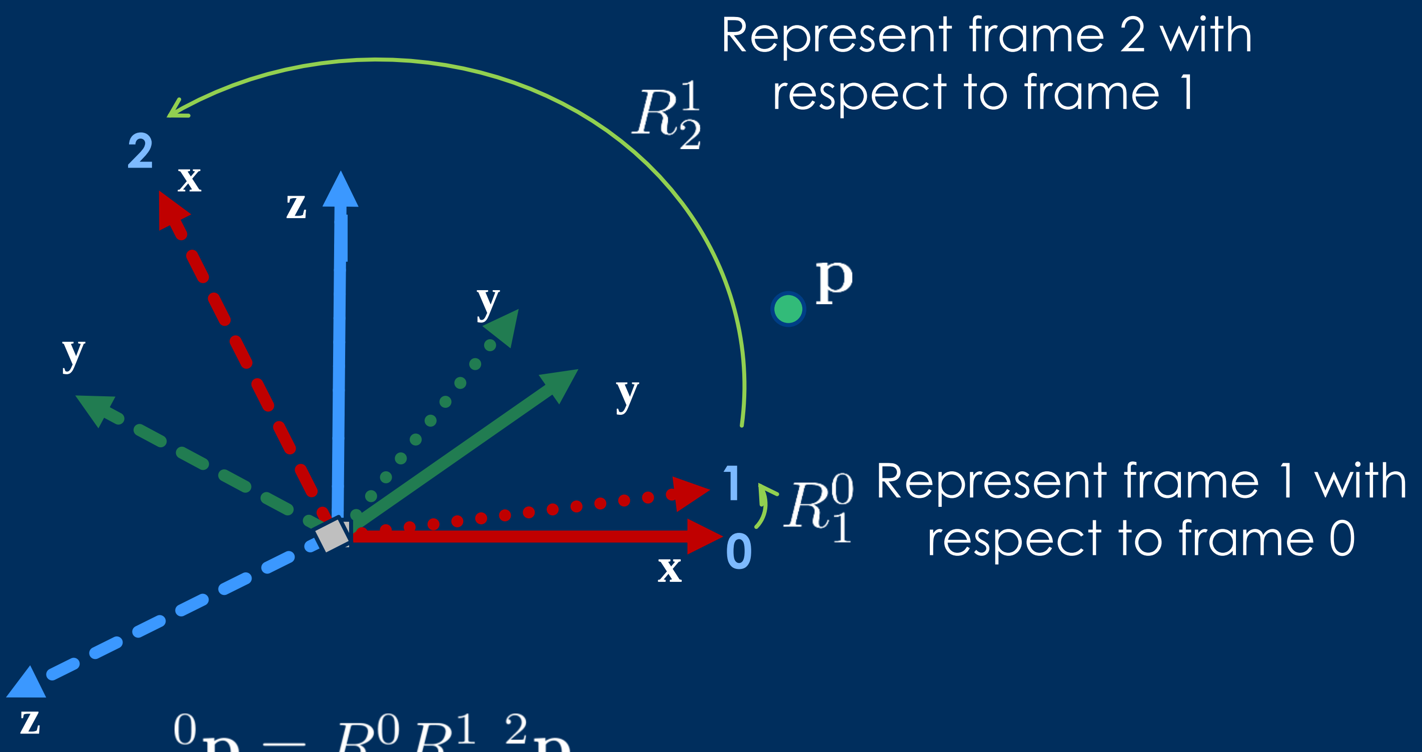
Coordinate Frames – Details: Rotating Vectors



$$ {}^1\mathbf{v}_b = R\, {}^1\mathbf{v}_a $$

Rotation Matrix Use Case 3:
Can be used to represent a rotation about a single frame and apply it to a vector.

Coordinate Frames – Details: Rotation Composition

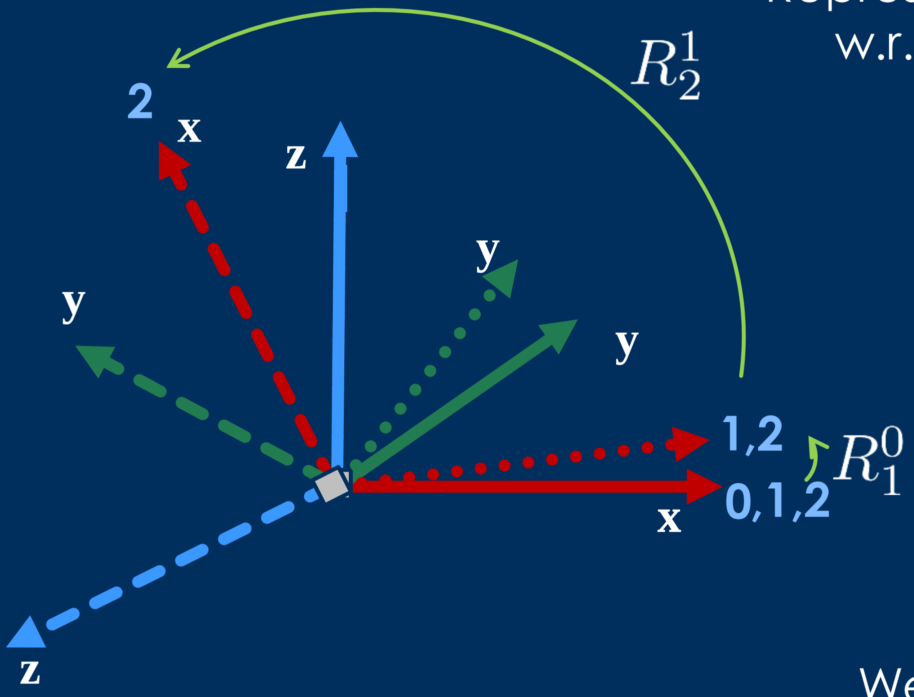


$$^0\mathbf{p} = R_1^0 R_2^1 \; ^2\mathbf{p}$$

$$R_2^0 = R_1^0 R_2^1$$

We can compose rotation matrices between consecutive frames in a sequence.

Coordinate Frames – Details: Rotation Composition (Rotation about *the current frame*)



Represent rotation
w.r.t. frame 1

The "new frame" in the sequence is called
the current frame.

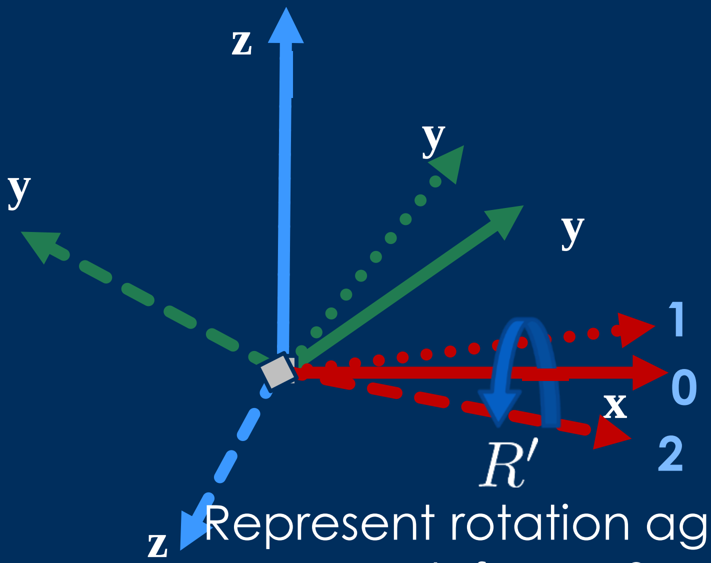
Represent rotation
w.r.t. frame 0

$$R = R_1^0 R_2^1$$

We can compose rotation matrices sequentially on
the right side (or by post-multiplication)
to generate a matrix that represents a sequence of rotations
about ***the current frame.***

Coordinate Frames – Details: Rotation Composition (Rotation about *the fixed frame*)

The initial or world frame is called
the fixed frame.



R_1^0 Represent rotation
w.r.t. frame 0

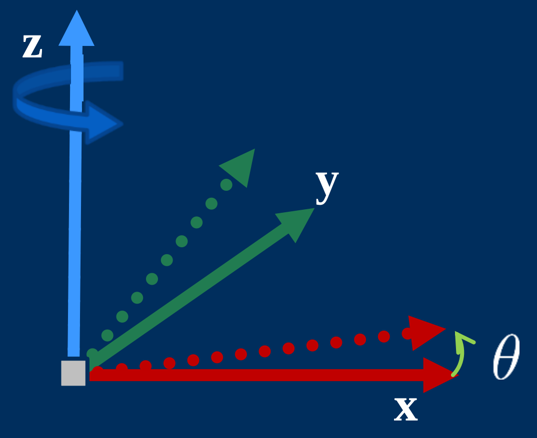
Represent rotation again
w.r.t. frame 0

$$R = R' \, R_1^0$$

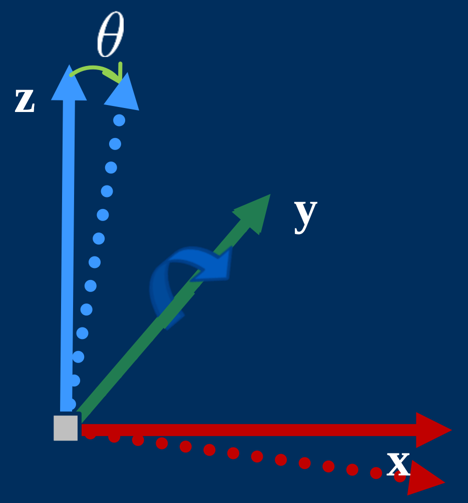
We can compose rotation matrices sequentially on
<u>the left side (or by pre-multiplication)</u>
to generate a matrix that represents a sequence of rotations
about ***the fixed frame.***

Revisit: Common Rotations in 3D

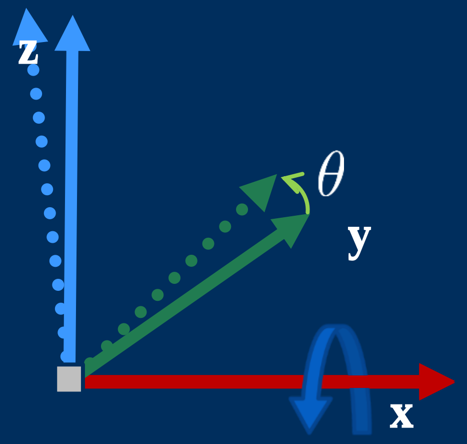
Rotation about Z-axis



Rotation about Y-axis



Rotation about X-axis



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

Matrix are formulated the same, but order depends on whether the reference frame is current vs. fixed.

Coordinate Frames – Details: Rotation Composition (Example)

Define the rotation matrix R that is defined by the following sequence of rotations in the specified order:

1. A rotation matrix of θ about the current x-axis.
2. A rotation of Φ about the current z-axis.
3. A rotation of α about he fixed z-axis.
4. A rotation of β about the current y-axis.
5. A rotation of δ around the fixed x-axis.

$$R = R_{x,\delta} R_{z,\alpha} R_{x,\theta} R_{z,\phi} R_{y,\beta}$$

Cite: Spong Ch2, Example 2.8

Coordinate Frames – Details: Rotation Parameterizations

Euler Angles

- Sequence of 3 rotations according to one of many conventions.
- Must be consistent in convention used.

- Very common
- Easy to understand

- Problems:
  - Gimbal lock
  - Conventions can be tricky to get right.

Axis Angle

- Represent by:
  - An axis about which rotation occurs
  - A magnitude (angle)

- Very common
- Summarizes the entire rotation

- Problems:
  - Requires conversion b

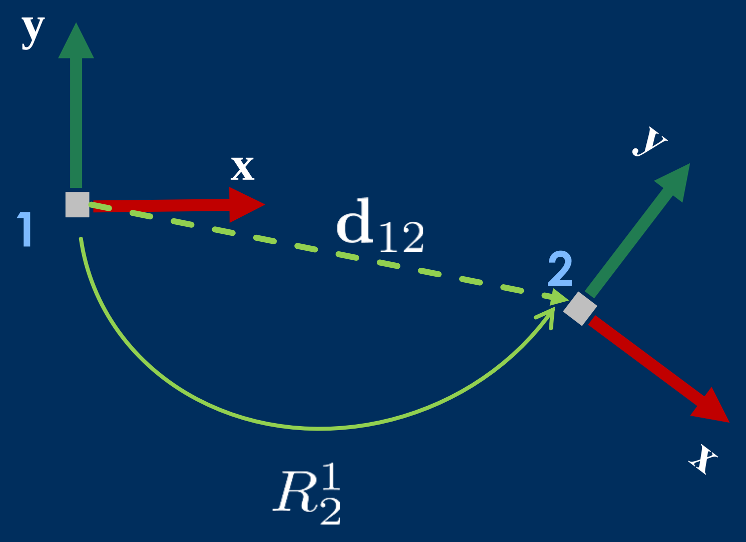
Quaternions

- Represent by:
  - An axis about which rotation occurs
  - A magnitude (angle)

- Very common
- Avoids Gimbal lock

- Problems:
  - Requires conversion to rotation matrix to rotate a point.

Coordinate Frames – Details: Rigid Motion



Rigid motion consists of both:
- Rotation and
- *Translation*

Changing the reference frame of a point

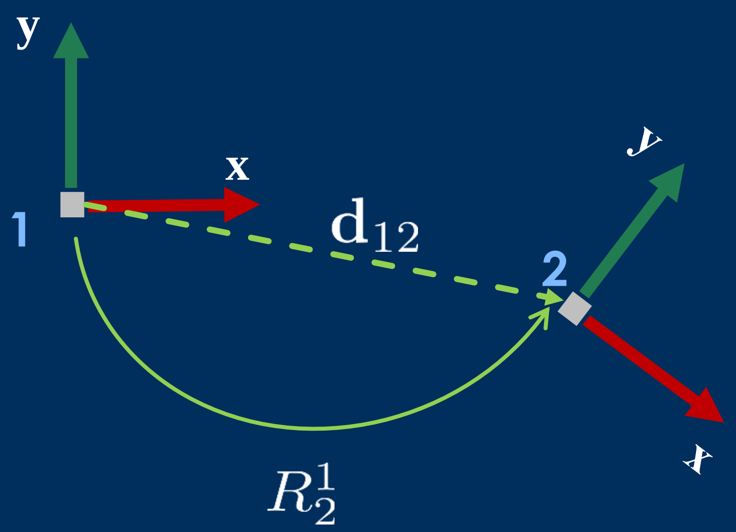
$$^1\mathbf{p} = R_2^1 \, ^2\mathbf{p} + \, ^1\mathbf{d}_{12}$$

Homogeneous Transformation Matrix

$$H_2^1 = \begin{bmatrix} R_2^1 & ^1\mathbf{d}_{12} \\ 0 & 1 \end{bmatrix}$$

$$^1P = H_2^1 \, ^2P$$

Coordinate Frames – Details: Homogeneous Transforms



$$^1\mathbf{p} = R_2^1 \, {}^2\mathbf{p} + \, {}^1\mathbf{d}_{12}$$

$$H_2^1 = \begin{bmatrix} R_2^1 & {}^1\mathbf{d}_{12} \\ 0 & 1 \end{bmatrix}$$

$$^1P = H_2^1 \, {}^2P$$

$$^1P = \begin{bmatrix} {}^1\mathbf{p} \\ 1 \end{bmatrix}$$

Coordinate Frames – Details: Homogeneous Transforms

- Use homogeneous transformation matrices to represent full rigid body transformations.
- Can compose homogeneous transformations in the same way as rotations:
  - Pre-multiply for fixed frame
  - Post-multiply for current frame
- While rotations are part of the Special Orthogonal Group, SO(d), homogeneous transfromations are members of the Special Euclidean group: SE(d)
- We often refer to a homogeneous transformation as a “pose”.

References

- "Robot Modeling and Control", Spong, Hutchinson, and Vidyasagar (Chapter 2)