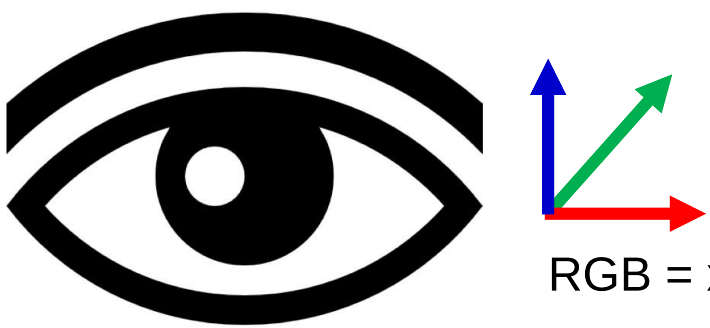
M3228.000300
SLAM 101

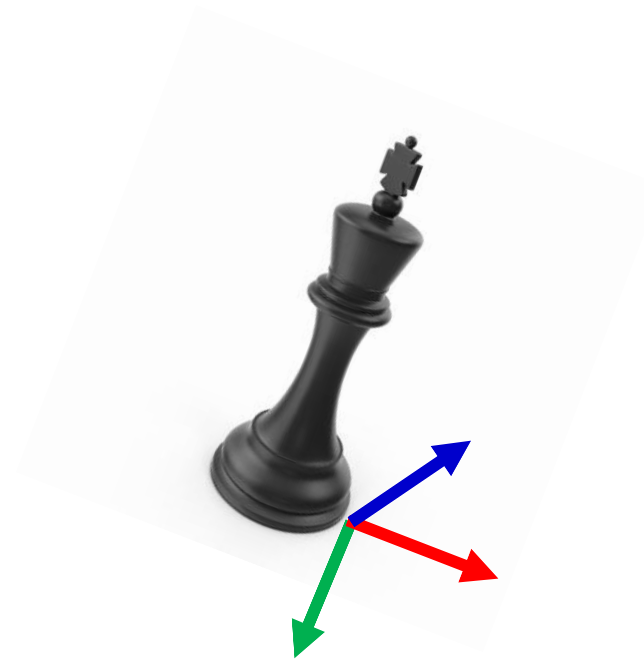
Lecture 02 State Representation

Ayoung Kim

State

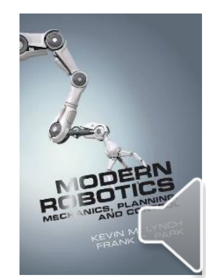
- How to describe the robot pose
  - Position = x,y,z
  - Orientation = ?
  - Pose = position and orientation
- Prior to everything
  - Define coordinate frame





RGB = x,y,z axis

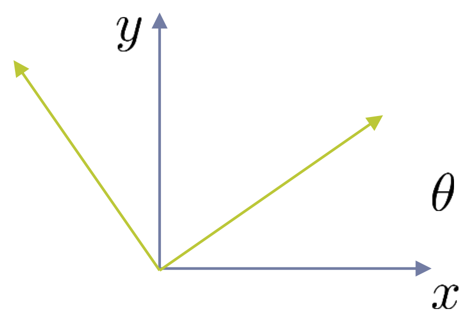
Modern Robotics: Mechanics, Planning, and Control
Book by Frank C. Park and Kevin M. Lynch



Rotation

Rotation matrix

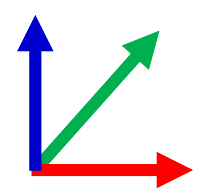
- Rotation can be expressed as a matrix
  - Rotation matrix R, and translation t
  - R = matrix, t = vector
  - 2D rotation matrix = 2x2 matrix

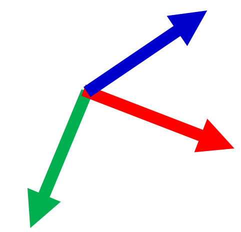


$$[1, 0] \rightarrow [\cos\theta, \sin\theta] \ \& \ [0, 1] \rightarrow [-\sin\theta, \cos\theta]$$

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

  - 3D rotation matrix = 3x3 matrix
    - Not straightforward
    - Euler angles, quaternion, screw param


Base frame


Target frame

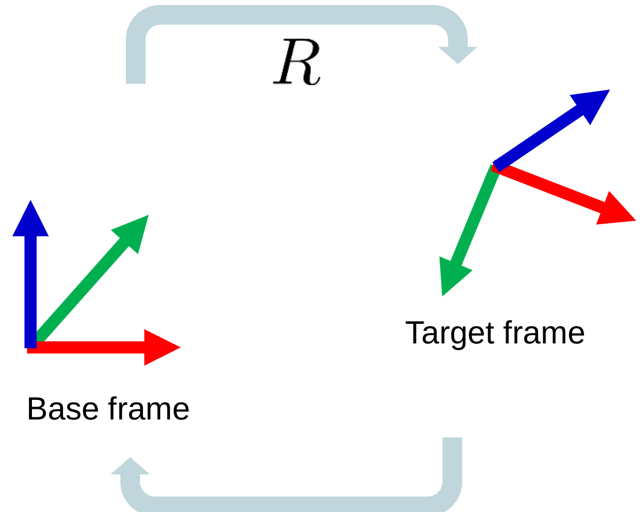
Rotation Matrix

- Properties
  - Inverse is transpose

    $$R^\top R = RR^\top = I$$

  - Determinant of proper R is $+1$
    - $det(R) = \pm 1$ from equation above
  - Rotation matrices are orthogonal matrix
    - (=orthonormal matrix)
  - $R \in SO(3)$

    Special orthogonal group



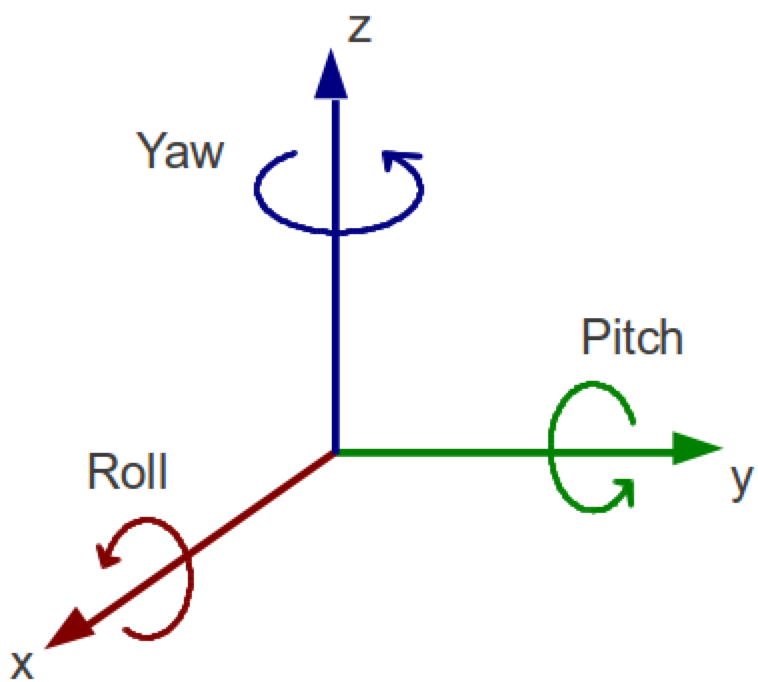
$$R^{-1} = R^\top$$

3D Rotation Matrix

- Euler angle
  - Separate three rotation matrices w.r.t. (x-axis, y-axis, z-axis)
  - Matrix multiplication to compose rotations
- Screw parameter
  - Generalized representation using screw theory
- Quaternion
  - Four parameters instead of three

Euler Angle

- Rotation matrix using Euler angle α, β, γ
  - Separate three rotation matrices w.r.t. (x-axis, y-axis, z-axis)
  - Combination order may differ



$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c(\theta) & -s(\theta) \\ 0 & s(\theta) & c(\theta) \end{bmatrix}$$

$$R_y(\theta) = \begin{bmatrix} c(\theta) & 0 & s(\theta) \\ 0 & 1 & 0 \\ -s(\theta) & 0 & c(\theta) \end{bmatrix}$$

$$R_z(\theta) = \begin{bmatrix} c(\theta) & -s(\theta) & 0 \\ s(\theta) & c(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

[Image courtesy] https://devforum.roblox.com/

Euler Angle

- Rotation matrix using Euler angle
  - Z-Y-X Euler angle, Z-Y-Z Euler angles…

- Example: Z-Y-X Euler angle
  - Rotate around z → y → x

$$R = Rot(z, \alpha) \cdot Rot(y, \beta) \cdot Rot(x, \gamma)$$

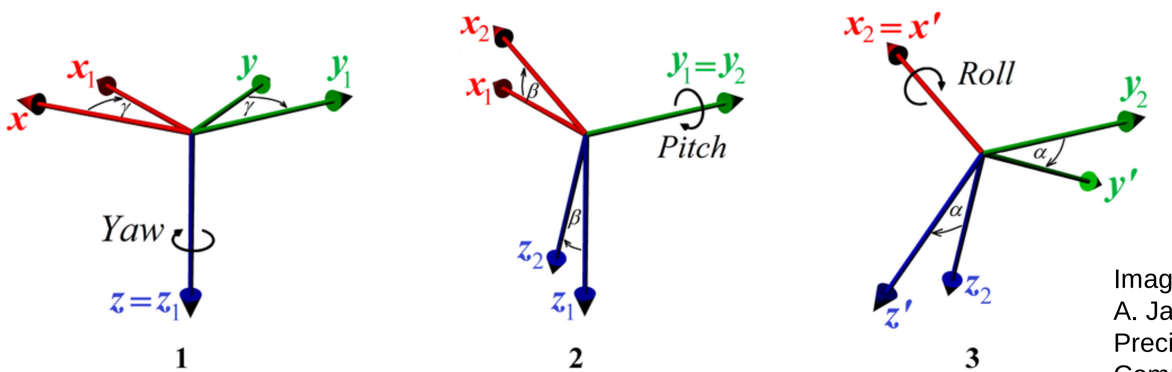
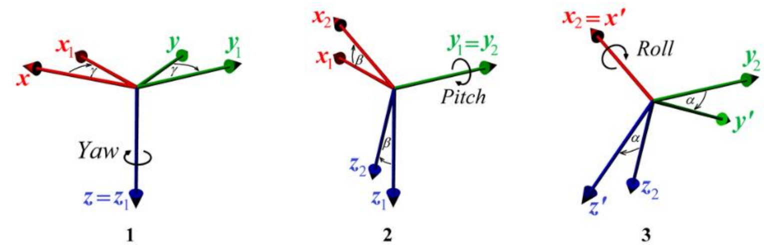


Image from
A. Janota et al., "Improving the Precision and Speed of Euler Angles Computation from Low-Cost Rotation Sensor Data" MDPI Sensors 2015.

$$\begin{aligned} R &= Rot(z, \alpha) \cdot Rot(y, \beta) \cdot Rot(x, \gamma) \\ &= \begin{bmatrix} c(\alpha) & -s(\alpha) & 0 \\ s(\alpha) & c(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c(\beta) & 0 & s(\beta) \\ 0 & 1 & \\ -s(\beta) & 0 & c(\beta) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c(\gamma) & -s(\gamma) \\ 0 & s(\gamma) & c(\gamma) \end{bmatrix} \end{aligned}$$

Euler Angle

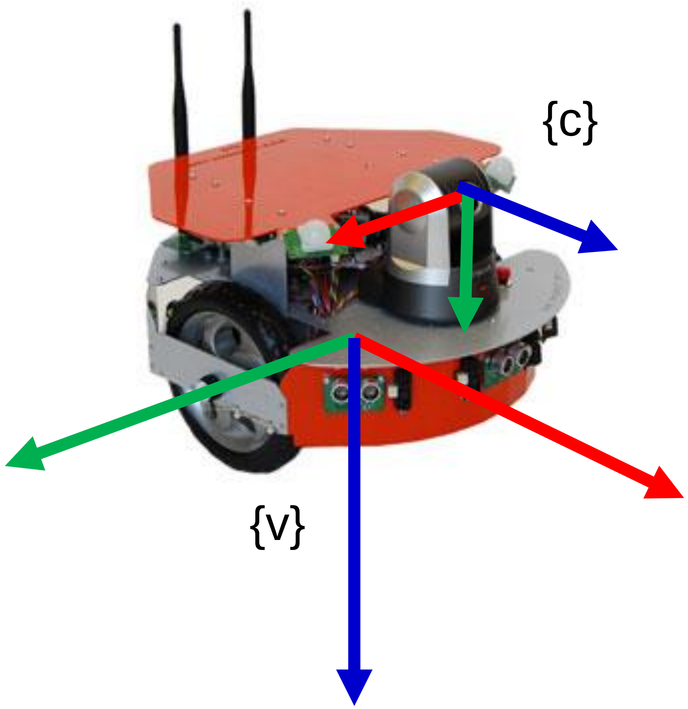
- Example: Z-Y-X Euler angle

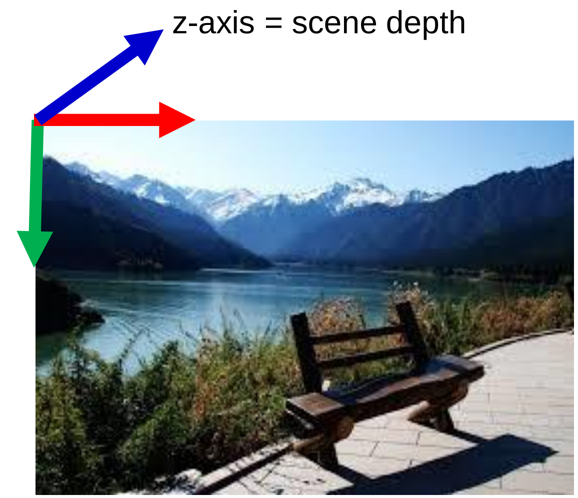


$$
\begin{aligned}
R &= Rot(z,\alpha)\cdot Rot(y,\beta)\cdot Rot(x,\gamma) \\
&= \begin{bmatrix} c(\alpha) & -s(\alpha) & 0 \\ s(\alpha) & c(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix}
\begin{bmatrix} c(\beta) & 0 & s(\beta) \\ 0 & 1 & \\ -s(\beta) & 0 & c(\beta) \end{bmatrix}
\begin{bmatrix} 1 & 0 & 0 \\ 0 & c(\gamma) & -s(\gamma) \\ 0 & s(\gamma) & c(\gamma) \end{bmatrix} \\
&= \begin{bmatrix} c(\alpha)c(\beta) & c(\alpha)s(\beta)s(\gamma)-s(\alpha)c(\gamma) & c(\alpha)s(\beta)c(\gamma)+s(\alpha)s(\gamma) \\ s(\alpha)c(\beta) & s(\alpha)s(\beta)s(\gamma)+c(\alpha)c(\gamma) & s(\alpha)s(\beta)c(\gamma)-c(\alpha)s(\gamma) \\ -s(\beta) & c(\beta)s(\gamma) & c(\beta)c(\gamma) \end{bmatrix}
\end{aligned}
$$

Euler Angle

- What is the camera pose w.r.t the vehicle

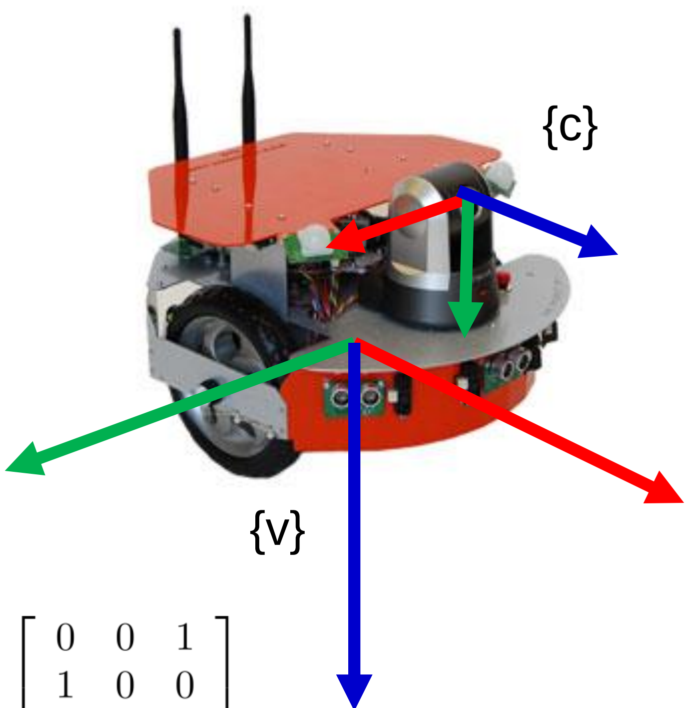




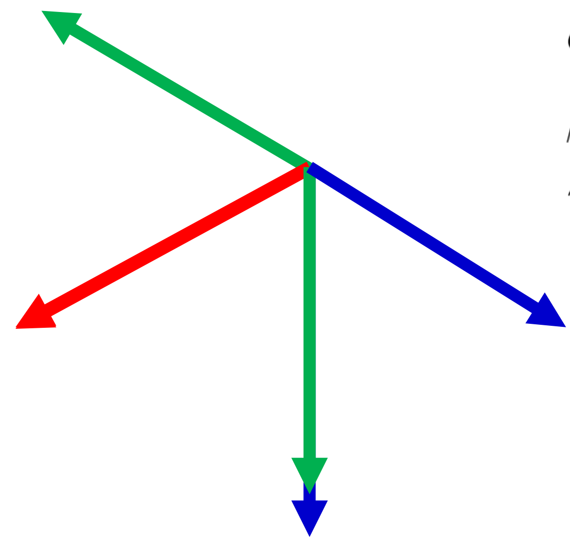
$$R_{vc} = \left[\quad \right]$$

Euler Angle

- What is the camera pose w.r.t the vehicle



$\alpha? \;\; \beta? \;\; \gamma?$



$\alpha = 90^\circ$

$\beta = 0^\circ$

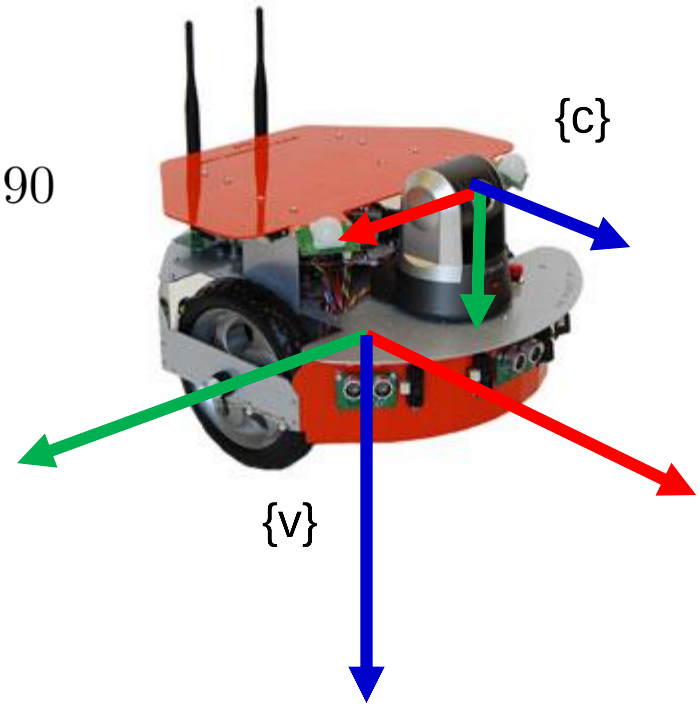
$\gamma = 90^\circ$

$$
\begin{aligned}
R_{vc} &= \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \\
&= \begin{bmatrix} c(\alpha)c(\beta) & c(\alpha)s(\beta)s(\gamma) - s(\alpha)c(\gamma) & c(\alpha)s(\beta)c(\gamma) + s(\alpha)s(\gamma) \\ s(\alpha)c(\beta) & s(\alpha)s(\beta)s(\gamma) + c(\alpha)c(\gamma) & s(\alpha)s(\beta)c(\gamma) - c(\alpha)s(\gamma) \\ -s(\beta) & c(\beta)s(\gamma) & c(\beta)c(\gamma) \end{bmatrix}
\end{aligned}
$$

Euler Angle

- 1) Use imagination
  - Align z → 90 degree
  - What next?
  - Alight x → 90 degree

$$\alpha = 90 \;\; \beta = 0 \;\; \gamma = 90$$



- 2) Use formula

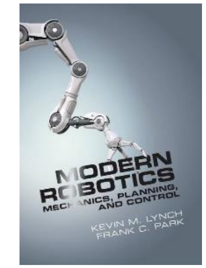
$$R_{vc} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

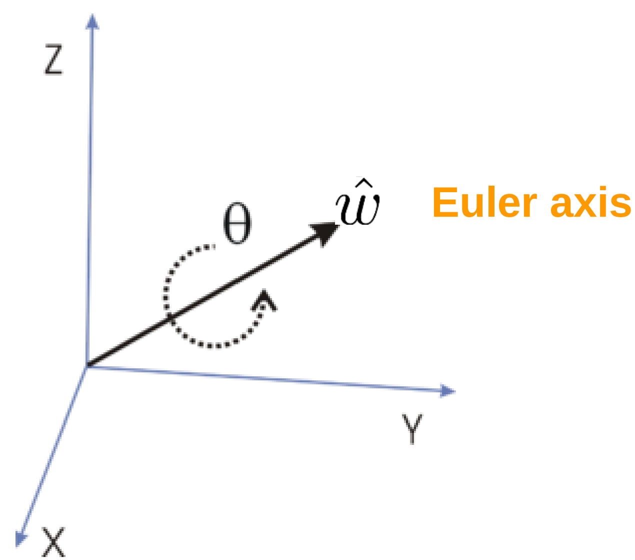
$$\begin{aligned} \alpha &= \tan^{-1}(R_{21}, R_{11}) \\ \beta &= \tan^{-1}(-R_{31}, R_{11}\cos(\alpha) + R_{21}\sin(\alpha)) \\ \gamma &= \tan^{-1}(R_{13}\sin(\alpha) - R_{23}\cos(\alpha), -R_{12}\sin(\alpha) + R_{22}\cos(\alpha)) \end{aligned}$$

- This has ambiguity (singularity) at +/- 90!

Screw parameter

- Euler's rotation theorem
  - Any rotation of a rigid body about a fixed point is equivalent to a single rotation by a given angle θ about a fixed axis
  - Euler axis = unit vector $\hat{w} = [w_1, w_2, w_3]$
  - How can we compose 3x3 rotation matrix from these screw parameters?



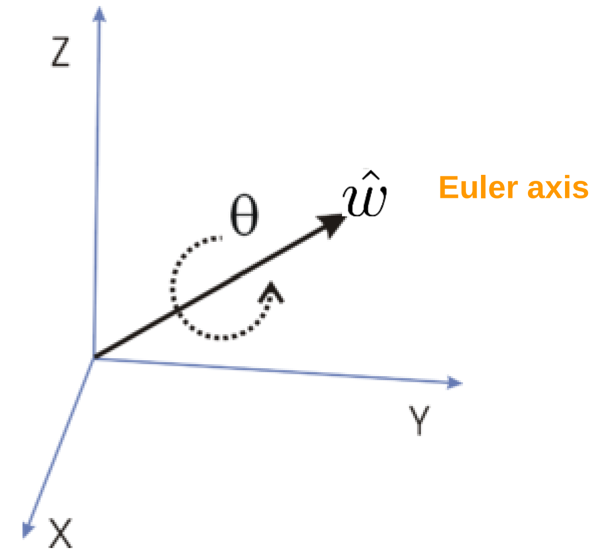


Screw parameter

- Screw parameters to SO(3)
  - Skew-symmetry of a matrix $[\hat{w}]$

$$[\hat{w}] = \begin{bmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{bmatrix}$$

    - Skew-symmetric $-[\hat{w}] = [\hat{w}]^\top$



  - Conversion

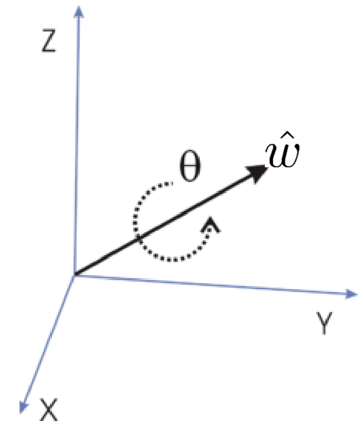
$$R = \cos(\theta) I_{3\times 3} + (1 - \cos(\theta))\hat{w}\hat{w}^\top - \sin(\theta)[\hat{w}]$$

Quaternion

- A quaternion q is defined as a complex number

$$q = [q_1, q_2, q_3, q_4] = q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k} + q_4$$

$\mathbf{i}$, $\mathbf{j}$ and $\mathbf{k}$ are three orthogonal unit vectors



- Represent rotation with *axis* + *scalar value*
  - Same idea as the screw parameter

    $q = [q_1, q_2, q_3]$ and q_4 is the scalar
- Unit quaternion (Euler parameters)

$$|q|^2 = q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1$$

- Relation to the screw parameter

$$\begin{aligned} q_0 &= \sin(\theta/2)\cdot w_1 \\ q_1 &= \sin(\theta/2)\cdot w_2 \\ q_2 &= \sin(\theta/2)\cdot w_3 \\ q_3 &= \cos(\theta/2) \end{aligned}$$

Rotation Representation

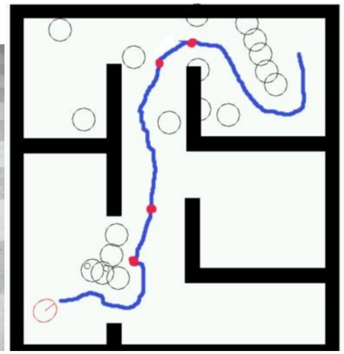
- Why Use Different Rotation Parameter?
  - They have different singularity
  - But not all are intuitive

	Pros	Cons	Application
Euler	Intuitive	Gimbal lock Slow to compute	Mobile robotics
Screw parameter	Forward kinematics		Manipulators
Quaternion	No Gimbal lock Fast to compute	Not intuitive	Computer graphics

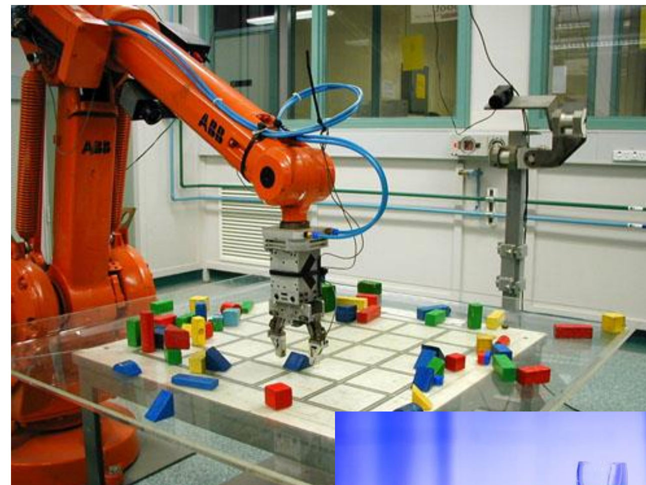
Rotation + Translation

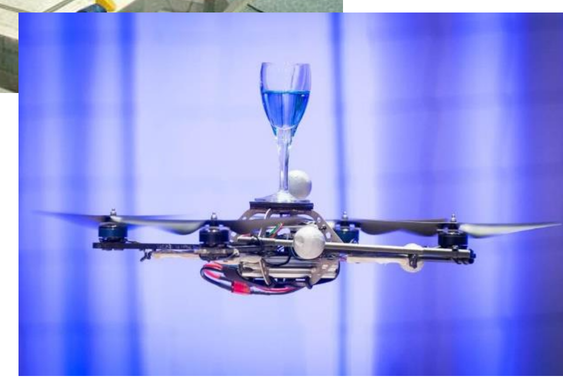
Motion Representation

- Move
- Interact



- Reach
- maintain posture





- Q.Where is the robot?
- Q. In what orientation?

Coordinate Representation

- Coordinate Representation = rotation + translation
  - Transformation matrix T = [R and t]
  - State X
  - Special Euclidean group
- 2 dimensional 3 DOF state representation
  - SE(2): 3x3 matrix that belongs to SE(2) group $[x, y, \theta]$
  - Vector form: x,y and heading
- 3 dimensional 6 DOF state representation
  - SE(3): 4x4 matrix that belongs to SE(3) group $[x, y, z, r, p, h]$
  - Vector form: x,y,z and roll, pitch, yaw

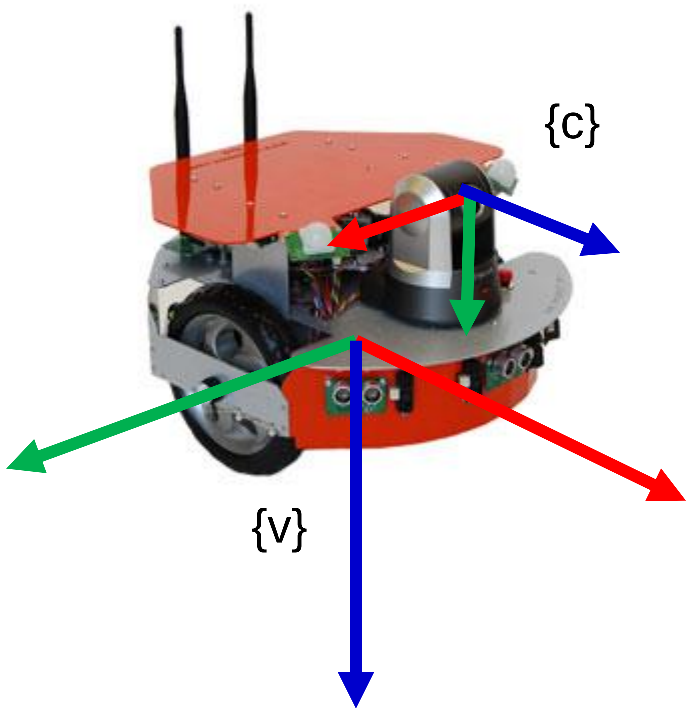
Coordinate Representation

- SE(3): Special Euclidean Group
  - 4x4 matrix

$$T = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \in SE(3)$$

R is 3×3 rotation matrix
t is 3×1 translation vector

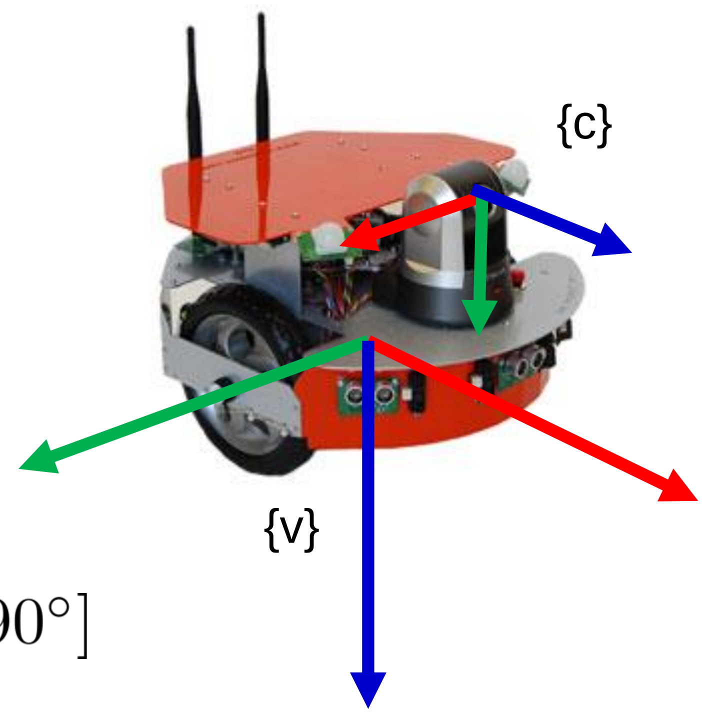
$$T_{vc} = \begin{bmatrix} R_{vc} & t_{vc} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & \Delta x \\ 1 & 0 & 0 & \Delta y \\ 0 & 1 & 0 & \Delta z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Coordinate Representation

- 6 DOF Vector form
  - Use three Euler angle for orientation description

$$X = [x, y, z, r, p, h]$$



$$T_{vc} = \begin{bmatrix} 0 & 0 & 1 & \Delta x \\ 1 & 0 & 0 & \Delta y \\ 0 & 1 & 0 & \Delta z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

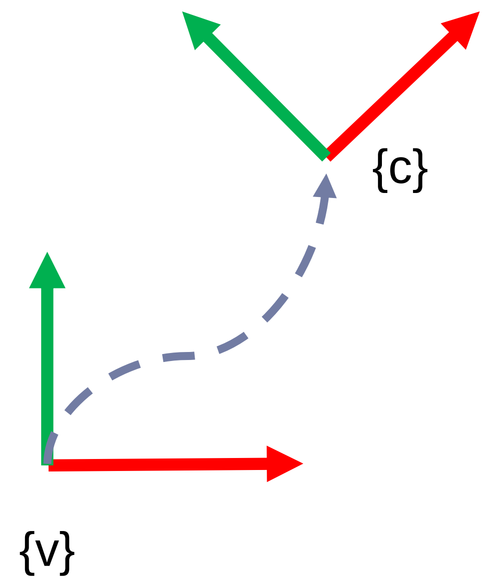
$$X_{vc} = [\Delta x, \Delta y, \Delta z, 90°, 0, 90°]$$

Coordinate Representation – 2D

- Example: Simpler one
  - 2D 3DOF representation

$$T_{vc} = \begin{bmatrix} \cos\theta & -\sin\theta & \Delta x \\ \sin\theta & \cos\theta & \Delta y \\ 0 & 0 & 1 \end{bmatrix}$$

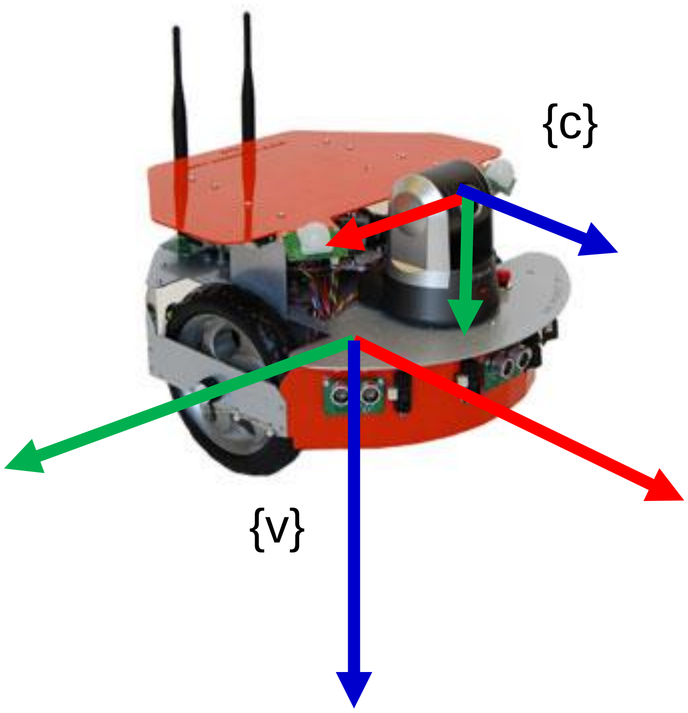
$$X_{vc} = [\Delta x, \Delta y, \theta]$$



State Representation Preference

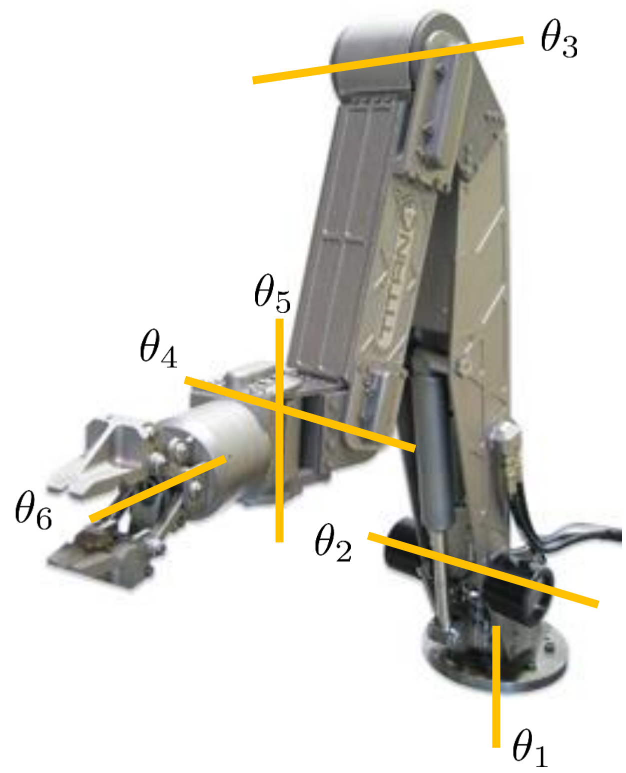
- Mobile robots

$$X = [x, y, z, r, p, h]$$



- Manipulators

$$T = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix} \in SE(3)$$

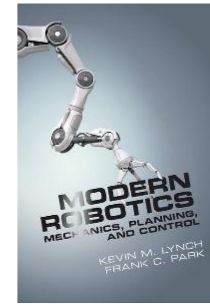


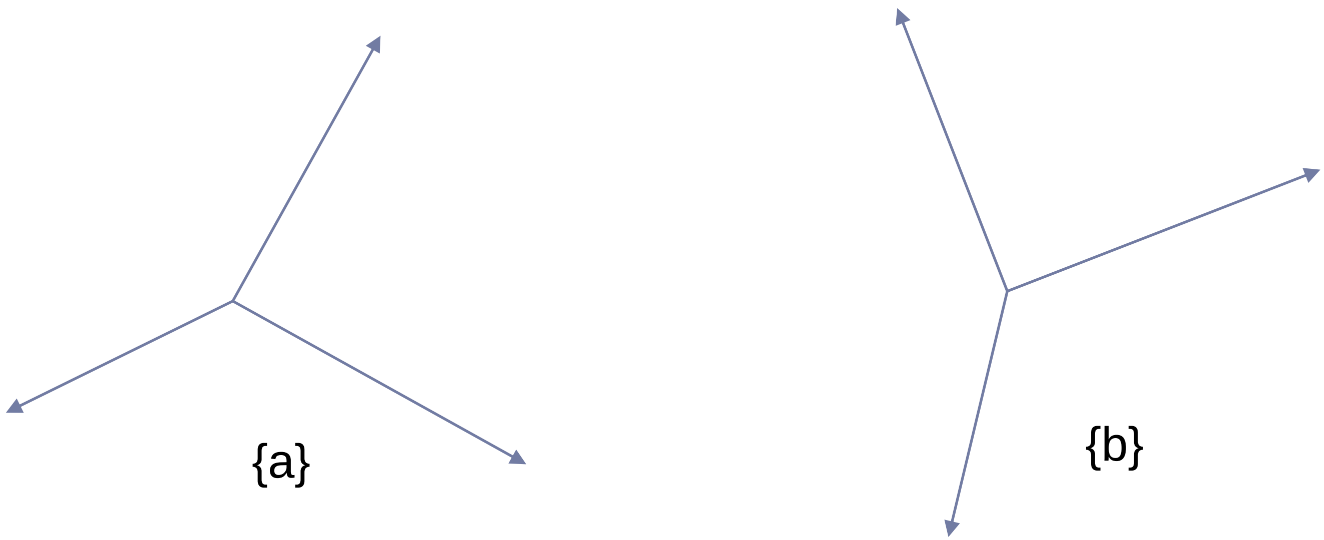
Why?

Rigid Body Transformation

Review: Rigid Body Transformation

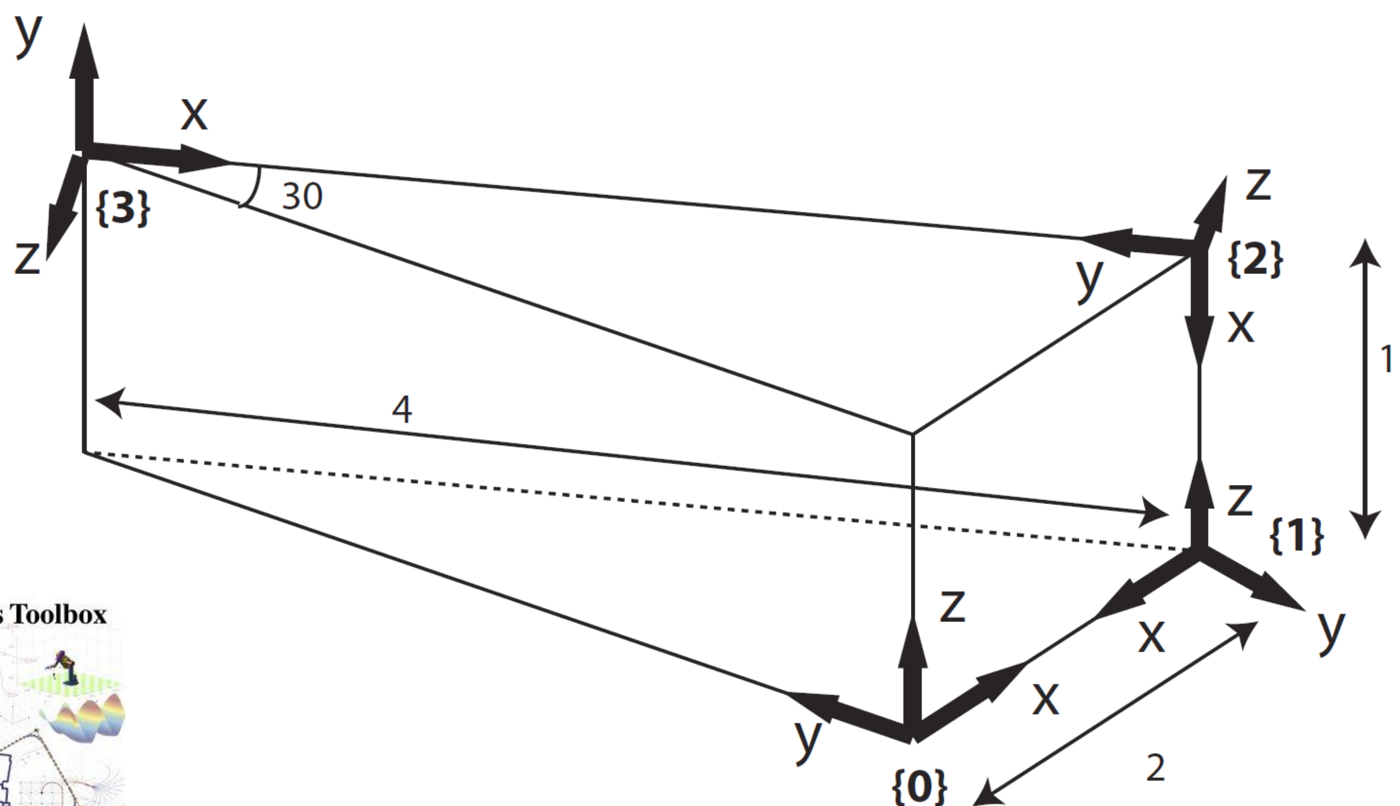
- Review "Introduction to robotics" from SNUON
- Coordinate {a} to coordinate {b}

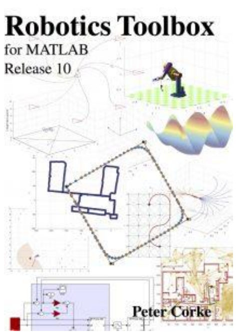




Example – Rotation + Translation

- What is the SE(3) between {0} and {2} ?





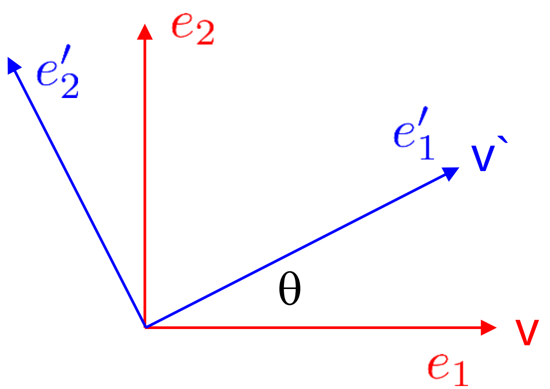
http://petercorke.com/wordpress/toolboxes/robotics-toolbox

Example – Coordinate Basis Change

- Consider 2D rotation
  - Rotation R from {v} to {v'}

$$[1,0] \rightarrow [\cos\theta, \sin\theta] \ \& \ [0,1] \rightarrow [-\sin\theta, \cos\theta]$$

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

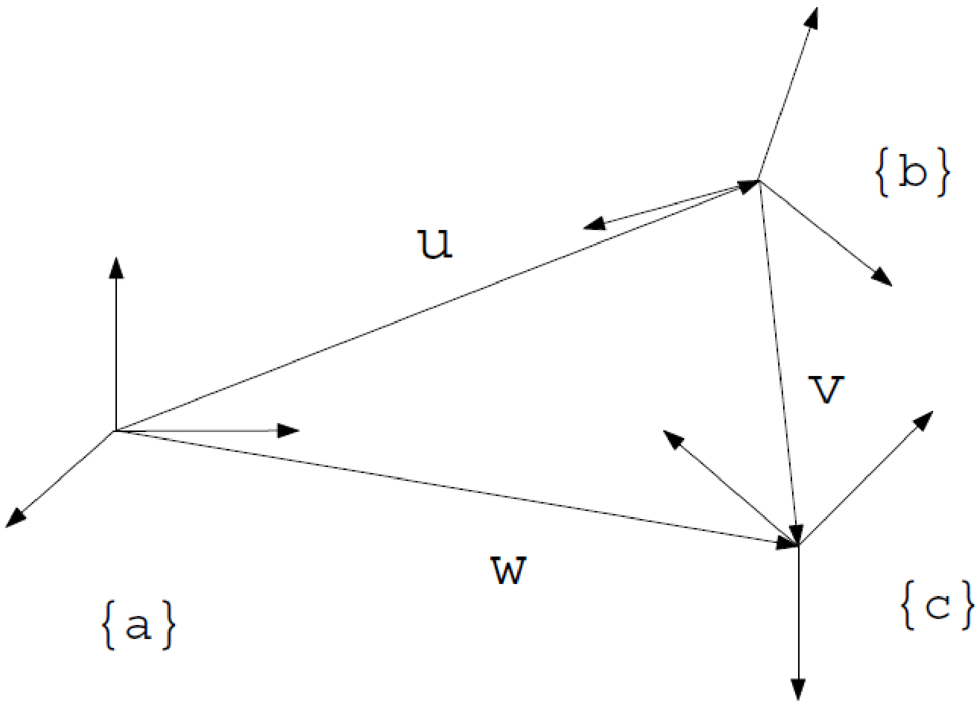


  - Two basis vectors e_1, e_2 transformed to e'_1, e'_2

$$\begin{bmatrix} e'_1 \\ e'_2 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \end{bmatrix}$$

$$v' = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} v = Rv$$

Example – Three Frames



$$\boxed{R_{ac} = R_{ab} R_{bc}}$$

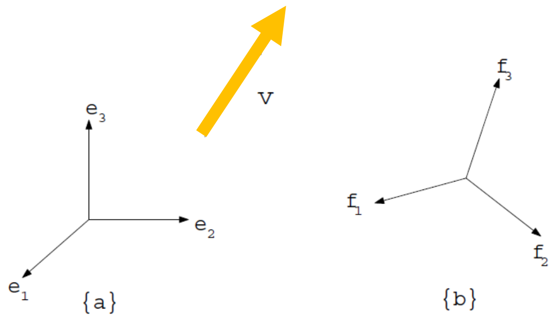
$$w_a = u_a + v_a$$

$$v_a = R_{ab} v_b$$

$$\boxed{w_a = u_a + R_{ab} v_b}$$

$$\begin{bmatrix} R_{ac} & w_a \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_{ab} & u_a \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_{bc} & v_b \\ 0 & 1 \end{bmatrix}$$

Example – Change of Basis



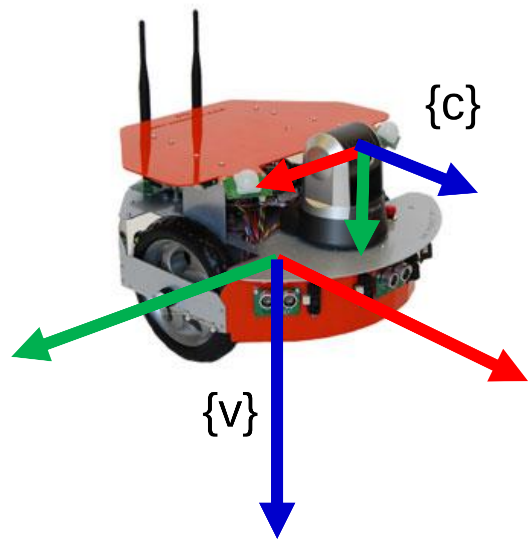
$$\mathbf{v} = v_{a,1}\mathbf{e}_1 + v_{a,2}\mathbf{e}_2 + v_{a,3}\mathbf{e}_3$$
$$= v_{b,1}\mathbf{f}_1 + v_{b,2}\mathbf{f}_2 + v_{b,3}\mathbf{f}_3$$

$$v_a = (v_{a,1}, v_{a,2}, v_{a,3}),\ v_b = (v_{b,1}, v_{b,2}, v_{b,3})$$

$$\mathbf{v} = \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3 \end{bmatrix} v_a$$
$$= \begin{bmatrix} \mathbf{f}_1 & \mathbf{f}_2 & \mathbf{f}_3 \end{bmatrix} v_b$$

$$\begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1 & \mathbf{f}_2 & \mathbf{f}_3 \end{bmatrix} R_{ba}$$

$$\boxed{R_{ba} v_a = v_b}$$

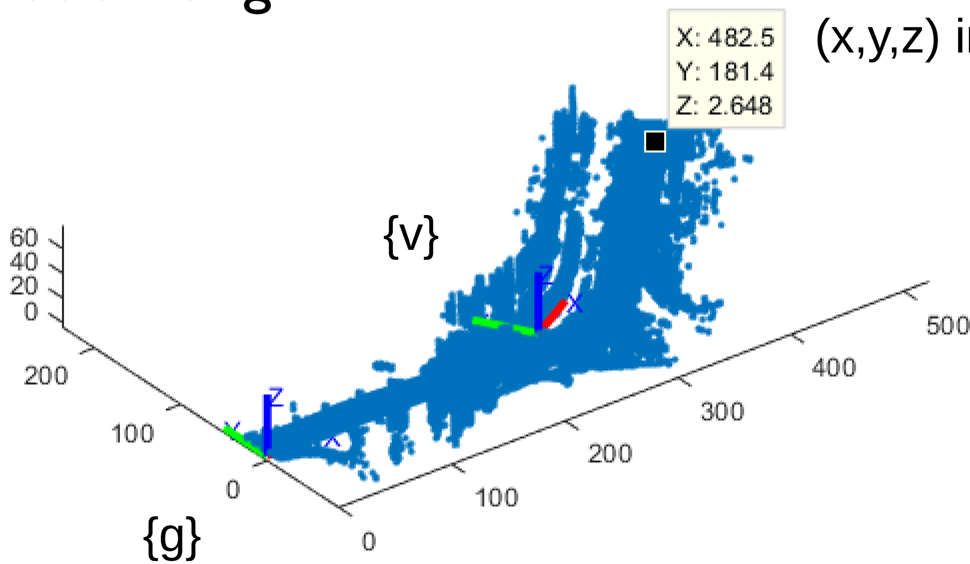


z_c

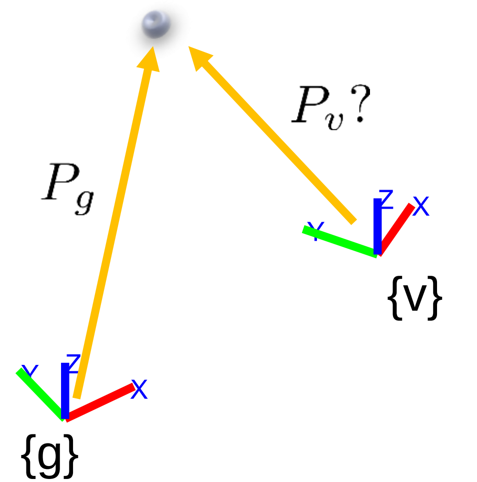
$$z_v = R_{vc} z_c$$

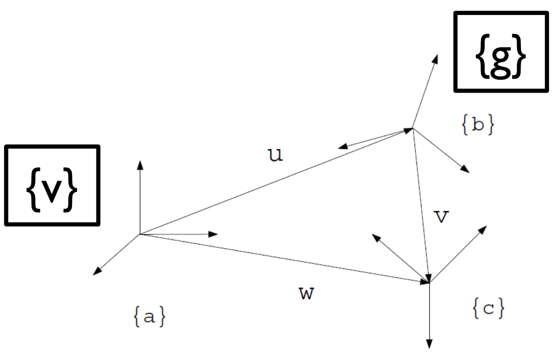
Example – Change of Basis

- Root shifting



(x,y,z) in the global coordinate {G}





$$\begin{bmatrix} R_{ac} & w_a \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_{ab} & u_a \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_{bc} & v_b \\ 0 & 1 \end{bmatrix}$$

$$w_a = u_a + R_{ab} v_b$$

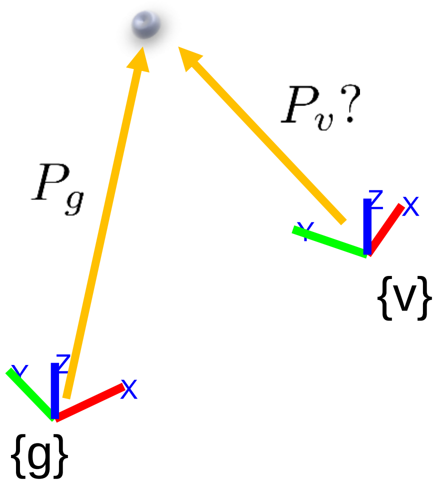
Example – Change of Basis

- Root shifting

$$\begin{bmatrix} R_{ac} & w_a \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_{ab} & u_a \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_{bc} & v_b \\ 0 & 1 \end{bmatrix}$$

$$w_a = u_a + R_{ab} v_b$$

$$P_v = t_{vg} + R_{vg} P_g$$

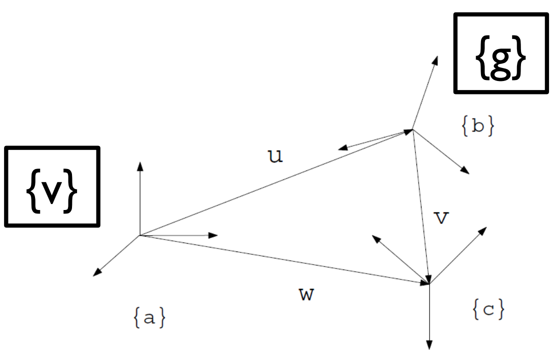


P_g: given from data
Point cloud in the global coordinate

R_vg & t_vg from inverse transform

$$\begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} R^\top & -R^\top t \\ 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} R_{gv} & t_{gv} \\ 0 & 1 \end{bmatrix}^{-1} = \begin{array}{cc} \boxed{\text{R\_vg}} & \boxed{\text{t\_vg}} \\ \end{array}\begin{bmatrix} R_{gv}^\top & -R_{gv}^\top t_{gv} \\ 0 & 1 \end{bmatrix}$$



$$\boxed{P_v = -R_{gv}^\top t_{gv} + R_{gv}^\top P_g = R_{gv}^\top (P_g - t_{gv})}$$

Thank you very much !!