Rigid Body Motions

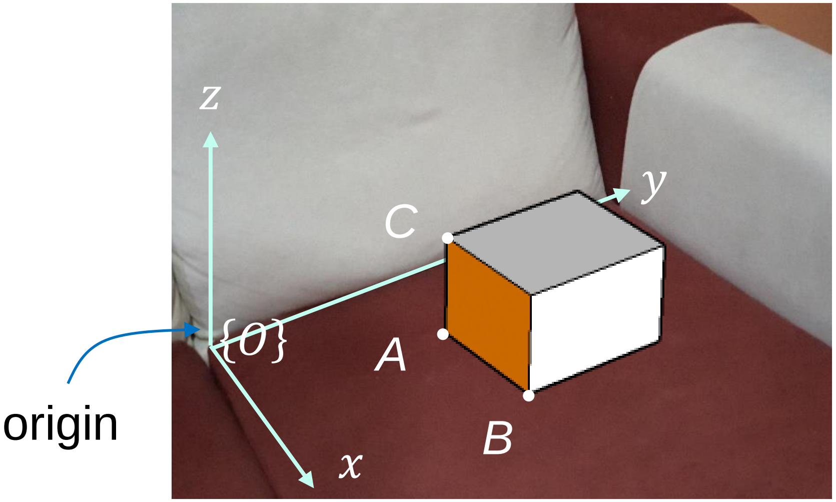
ZA-2203 Robotic Systems

Topics

- Coordinate systems
- Describing position
- Describing orientation
- More on orientation and rotation
- Describing pose (configuration)
- Describing motion (transformation): translation and rotation
- Describing velocities (screw theory)

Coordinate systems

Coordinate systems



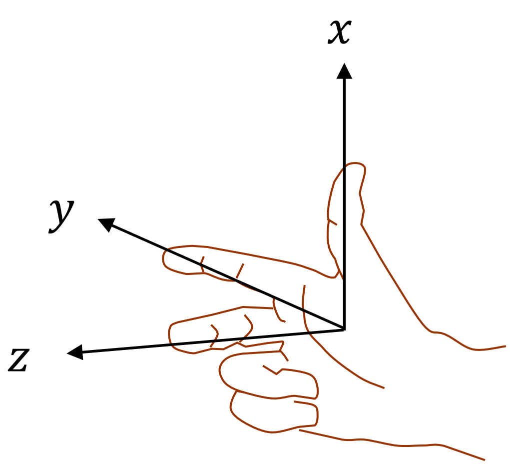
$$A = \begin{bmatrix} x_A \\ y_A \\ z_A \end{bmatrix}$$

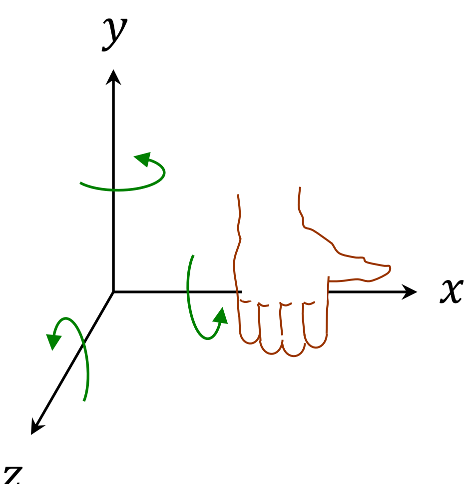
x_A means x coordinate of point A, likewise for other coordinate variables

$$B = \begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix}$$

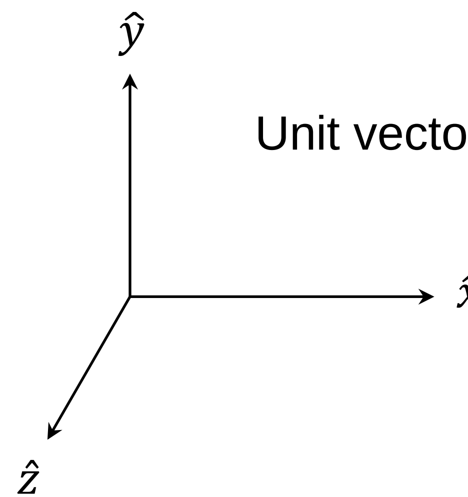
$$C = \begin{bmatrix} x_C \\ y_C \\ z_C \end{bmatrix}$$

Coordinate frame
- has origin
- orthogonal axes
- stationary





Right-hand rule (RHR)



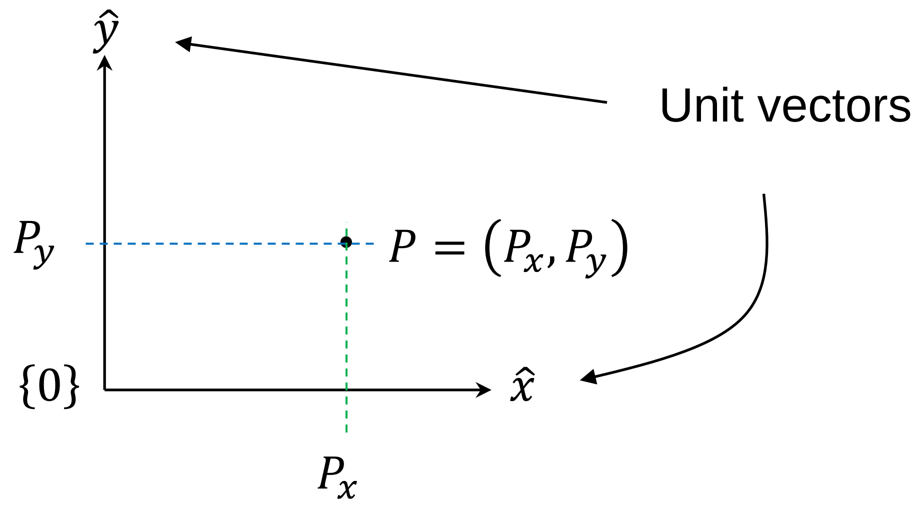
Unit vectors

Describing position

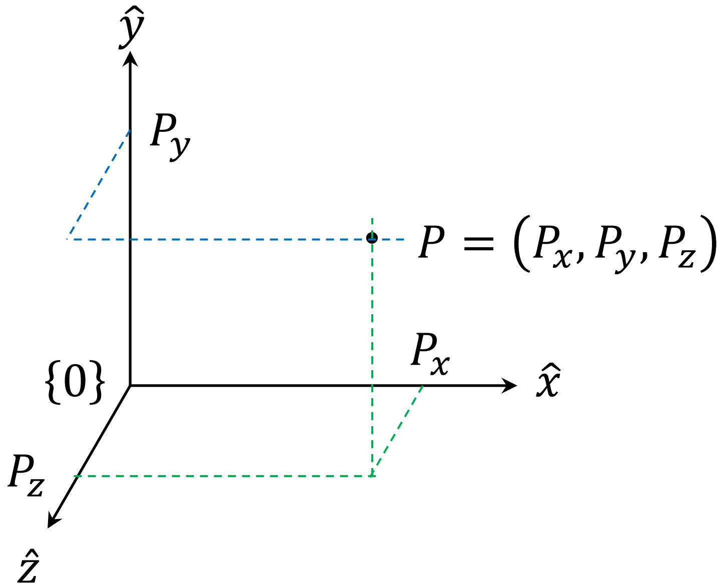
Position of a point

Position of a point: Cartesian coordinates

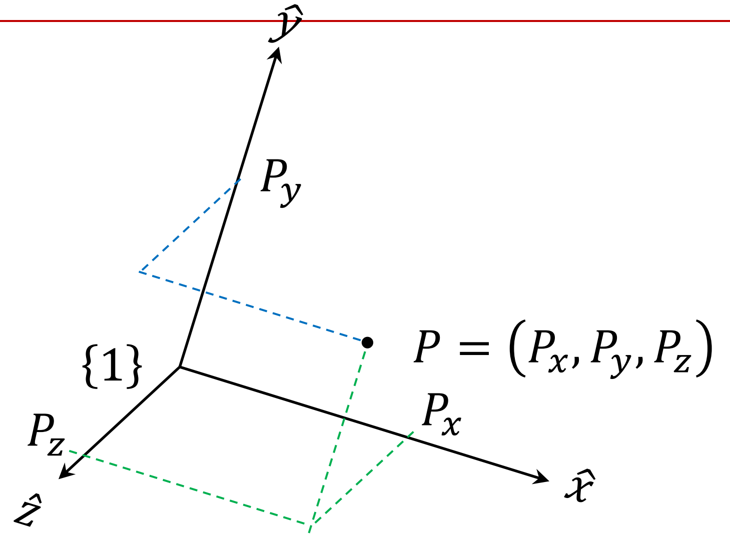
P_x means x coordinate of point P,
P_y means y coordinate of point P.
We have taken the liberal in using different notation conventions. Please interpret accordingly.



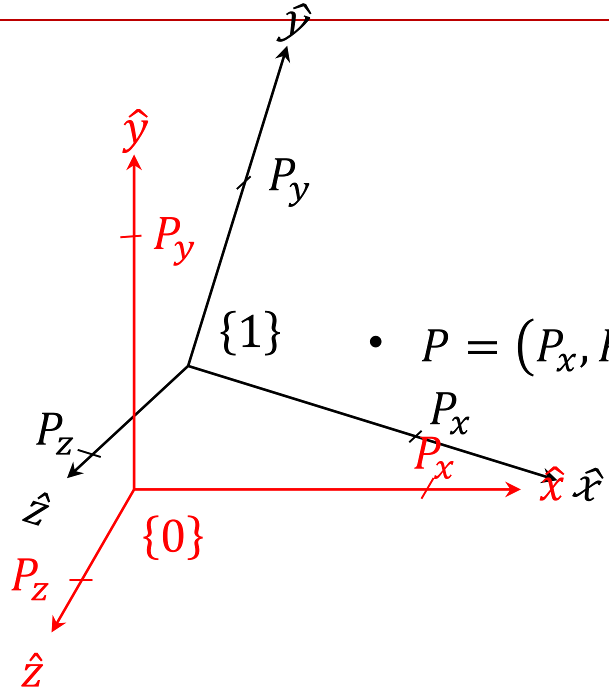
Position of a point: Cartesian coordinates



Position of a point: Cartesian coordinates



Position of a point: Cartesian coordinates



- $P = (P_x, P_y, P_z)$

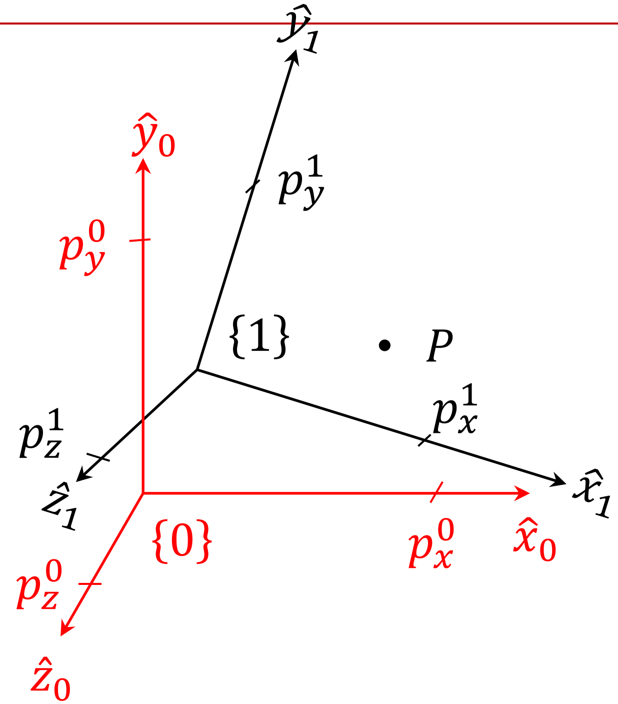
Position of a point: Cartesian coordinates

p_x^0 means x coordinate of point P in frame $\{0\}$,
p_y^0 means y coordinate of point P in frame $\{0\}$,
Likewise for other coordinate variables.
In this notation, the subscript is the subject of interest, and the superscript is the reference frame.
Alternative notation are below:

p_{0x} or ${}^0p_x$

In p^0, the superscript refers to the reference frame. Sometimes, subscript is used in this context, i.e. in p_0, the subscript is used to indicate the reference frame.

Different notation conventions have been used by different authors.
We will be liberal in using different notation conventions.
Please interpret accordingly.



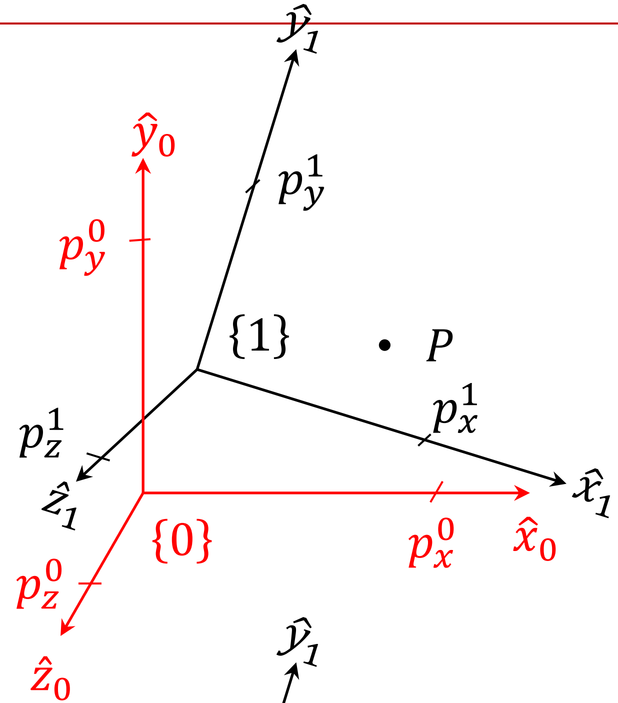
$$P^0 = (p_x^0, p_y^0, p_z^0)$$

$$P^1 = (p_x^1, p_y^1, p_z^1)$$

Cartesian coordinates

Position of a point: Cartesian coordinates

3D space

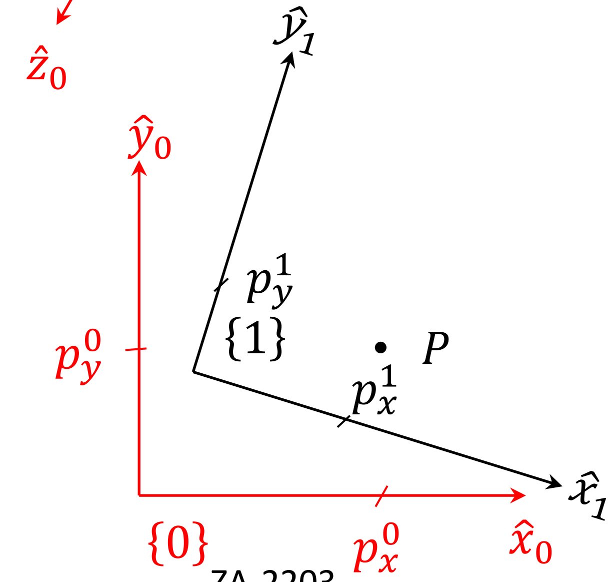


$$P^0 = \left(p_x^0, p_y^0, p_z^0\right)$$

$$P^1 = \left(p_x^1, p_y^1, p_z^1\right)$$

[Cartesian coordinates]

2D plane



$$P^0 = \left(p_x^0, p_y^0\right)$$

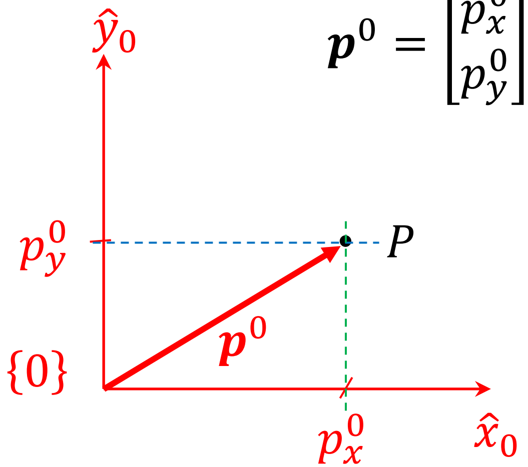
$$P^1 = \left(p_x^1, p_y^1\right)$$

[Cartesian coordinates]

Position of a point: Vector

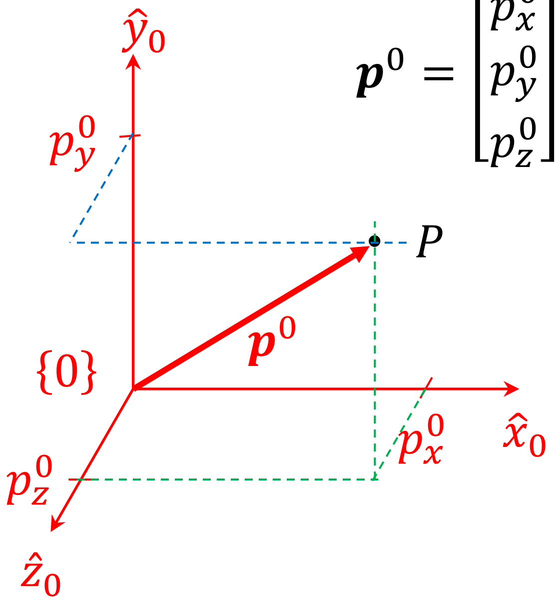
2D

$$\boldsymbol{p}^0 = \begin{bmatrix} p_x^0 \\ p_y^0 \end{bmatrix}$$



3D

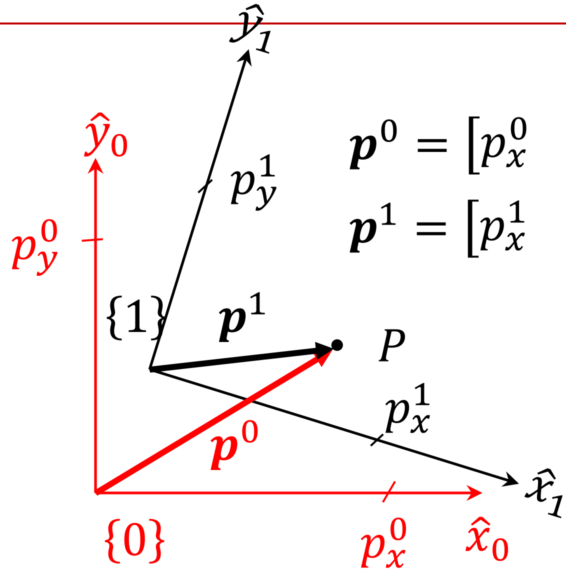
$$\boldsymbol{p}^0 = \begin{bmatrix} p_x^0 \\ p_y^0 \\ p_z^0 \end{bmatrix}$$



Vector is actually a displacement (more specifically a **translation** displacement).
It specify the position of P by specifying how much translation displacement in each dimension (x,y,z) is point P from the origin {0}.
It is useful in arithmetic, whereas we cannot perform arithmetic on points (Cartesian coordinates).

Position of a point: Vector

2D



$$\boldsymbol{p}^0 = \begin{bmatrix} p_x^0 & p_y^0 \end{bmatrix}^T$$
$$\boldsymbol{p}^1 = \begin{bmatrix} p_x^1 & p_y^1 \end{bmatrix}^T$$

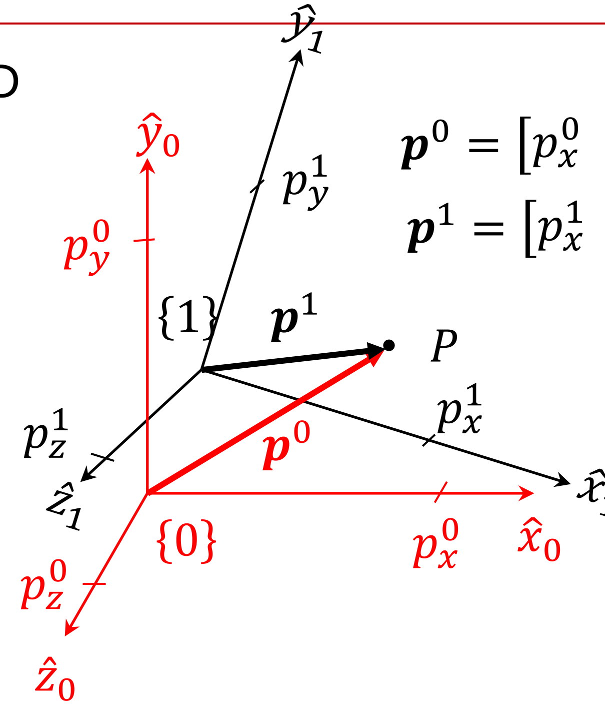
$$\hat{x}_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \hat{y}_0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \; in \; \{0\}$$

We can write

$$\boldsymbol{p}^0 = p_x^0 \hat{x}_0 + p_y^0 \hat{y}_0$$

$$\boldsymbol{p}^0 = p_x^0 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + p_y^0 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} p_x^0 \\ p_y^0 \end{bmatrix}$$

3D



$$\boldsymbol{p}^0 = \begin{bmatrix} p_x^0 & p_y^0 & p_z^0 \end{bmatrix}^T$$
$$\boldsymbol{p}^1 = \begin{bmatrix} p_x^1 & p_y^1 & p_z^1 \end{bmatrix}^T$$

$$\hat{x}_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \; \hat{y}_0 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \; \hat{z}_0 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \; in \; \{0\}$$

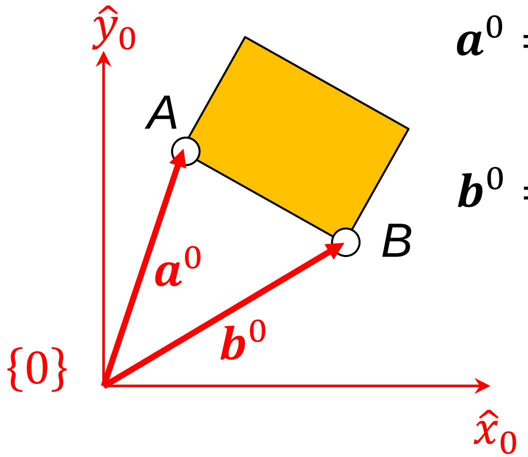
We can write

$$\boldsymbol{p}^0 = p_x^0 \hat{x}_0 + p_y^0 \hat{y}_0 + p_z^0 \hat{z}_0$$

Given the same point in space, the coordinates used to represent the position of the point depend on which reference coordinate frame is being used.

Position of a rigid body: Points

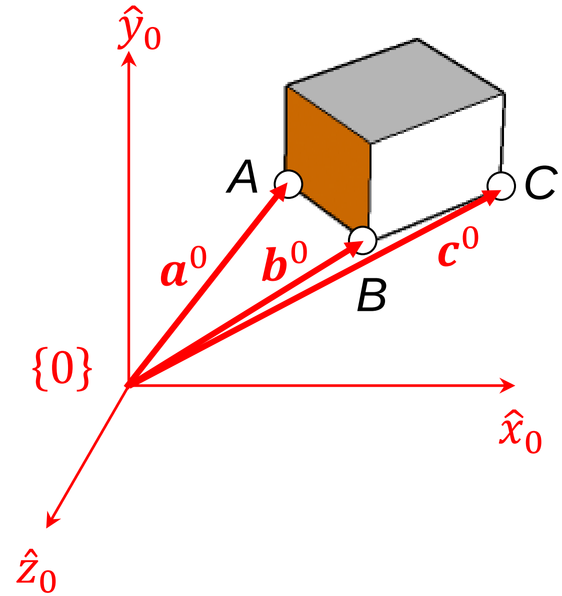
2D



$$\boldsymbol{a}^0 = \begin{bmatrix} a_x^0 \\ a_y^0 \end{bmatrix}$$

$$\boldsymbol{b}^0 = \begin{bmatrix} b_x^0 \\ b_y^0 \end{bmatrix}$$

3D



$$\boldsymbol{a}^0 = \begin{bmatrix} a_x^0 \\ a_y^0 \\ a_z^0 \end{bmatrix}$$

$$\boldsymbol{b}^0 = \begin{bmatrix} b_x^0 \\ b_y^0 \\ b_z^0 \end{bmatrix}$$

$$\boldsymbol{c}^0 = \begin{bmatrix} c_x^0 \\ c_y^0 \\ c_z^0 \end{bmatrix}$$

The configuration of a rigid body needs to specify where (position) it is in the space and how it is oriented (orientation).

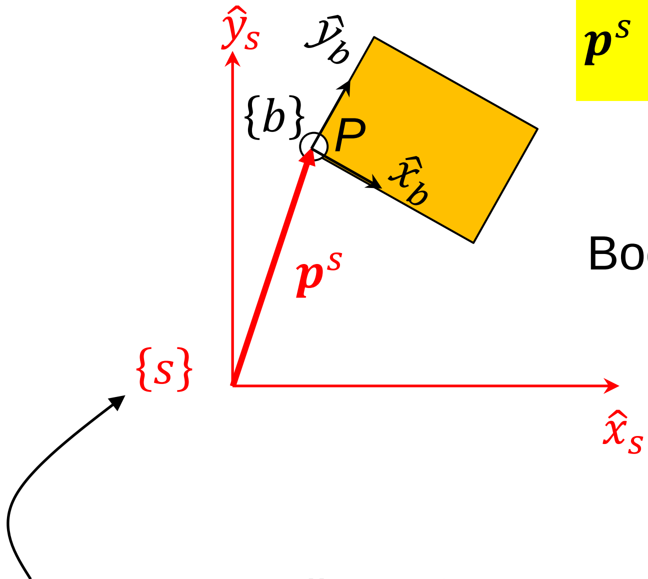
The position (vector) of the points sufficiently describe the **configuration** of the rigid bodies: both position and orientation.

However, this representation requires dealing with the constraints when moving the body.

It will become easier to deal rigid body motion if we decompose the representation into **position** and **orientation**.

Position of a rigid body: Body frame

2D

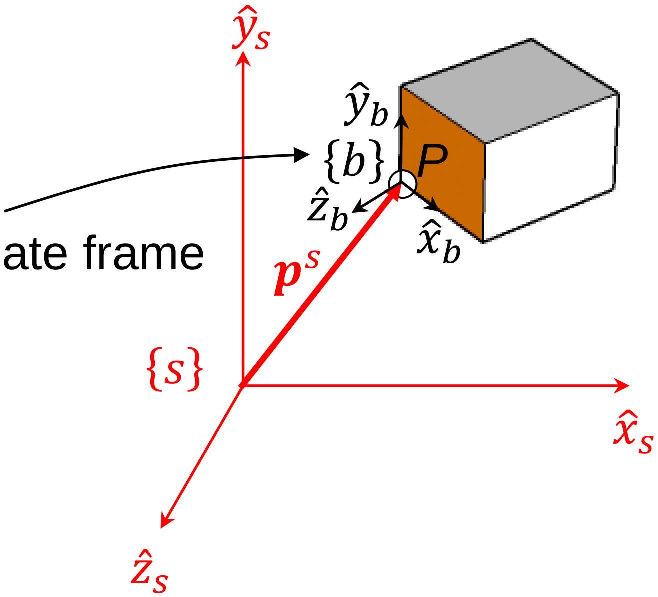


$$\boldsymbol{p}^s = \begin{bmatrix} p_x^s \\ p_y^s \end{bmatrix}$$

Space coordinate frame
(world, global, base or inertia)

3D

Body coordinate frame



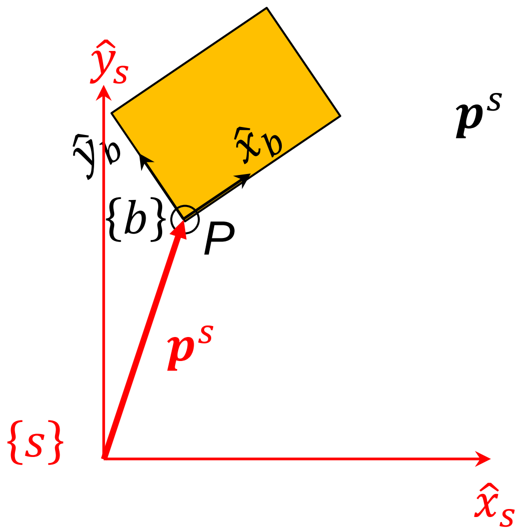
$$\boldsymbol{p}^s = \begin{bmatrix} p_x^s \\ p_y^s \\ p_z^s \end{bmatrix}$$

Attach a coordinate frame to the body.
The origin of the body coordinate frame {b} in the space coordinate frame {s} gives the position of the body in space.
The orientation of the {b} with respect to {s} gives the orientation of the body.

Describing orientation

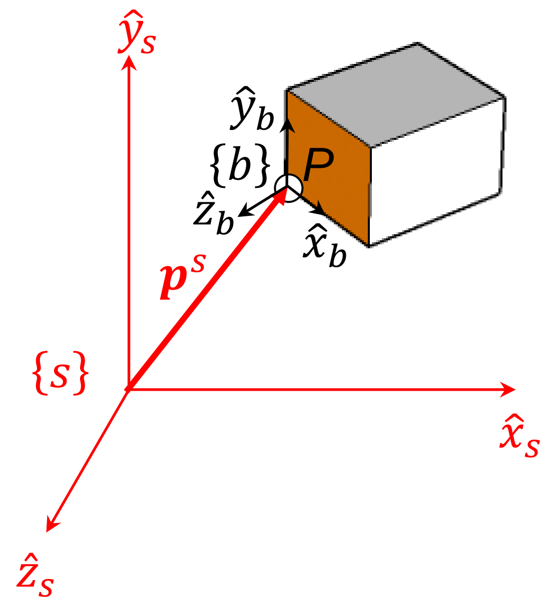
Orientation of a rigid body: Rotation

2D



$$\boldsymbol{p}^s = \begin{bmatrix} p_x^s \\ p_y^s \end{bmatrix}$$

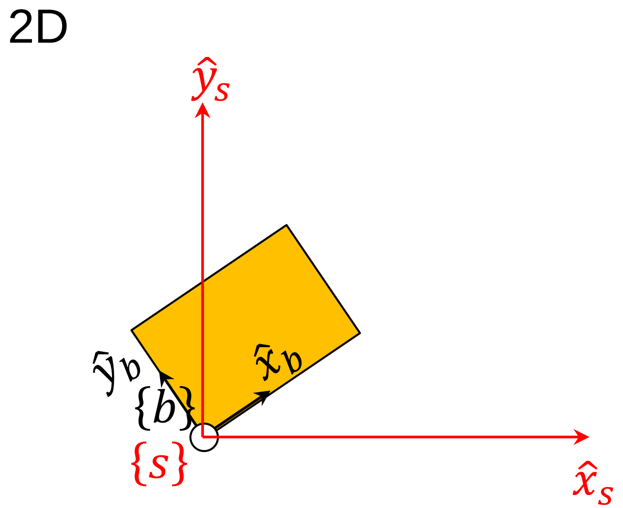
3D

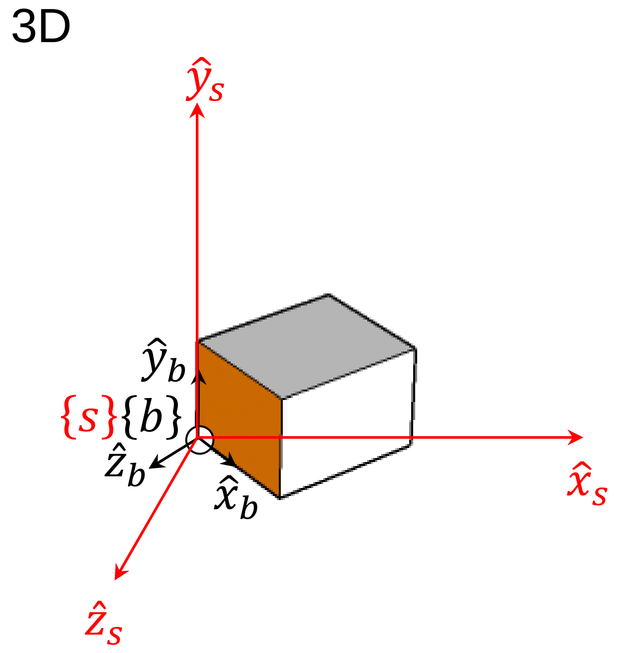


$$\boldsymbol{p}^s = \begin{bmatrix} p_x^s \\ p_y^s \\ p_z^s \end{bmatrix}$$

We consider orientation separately from the position …

Orientation of a rigid body: Rotation



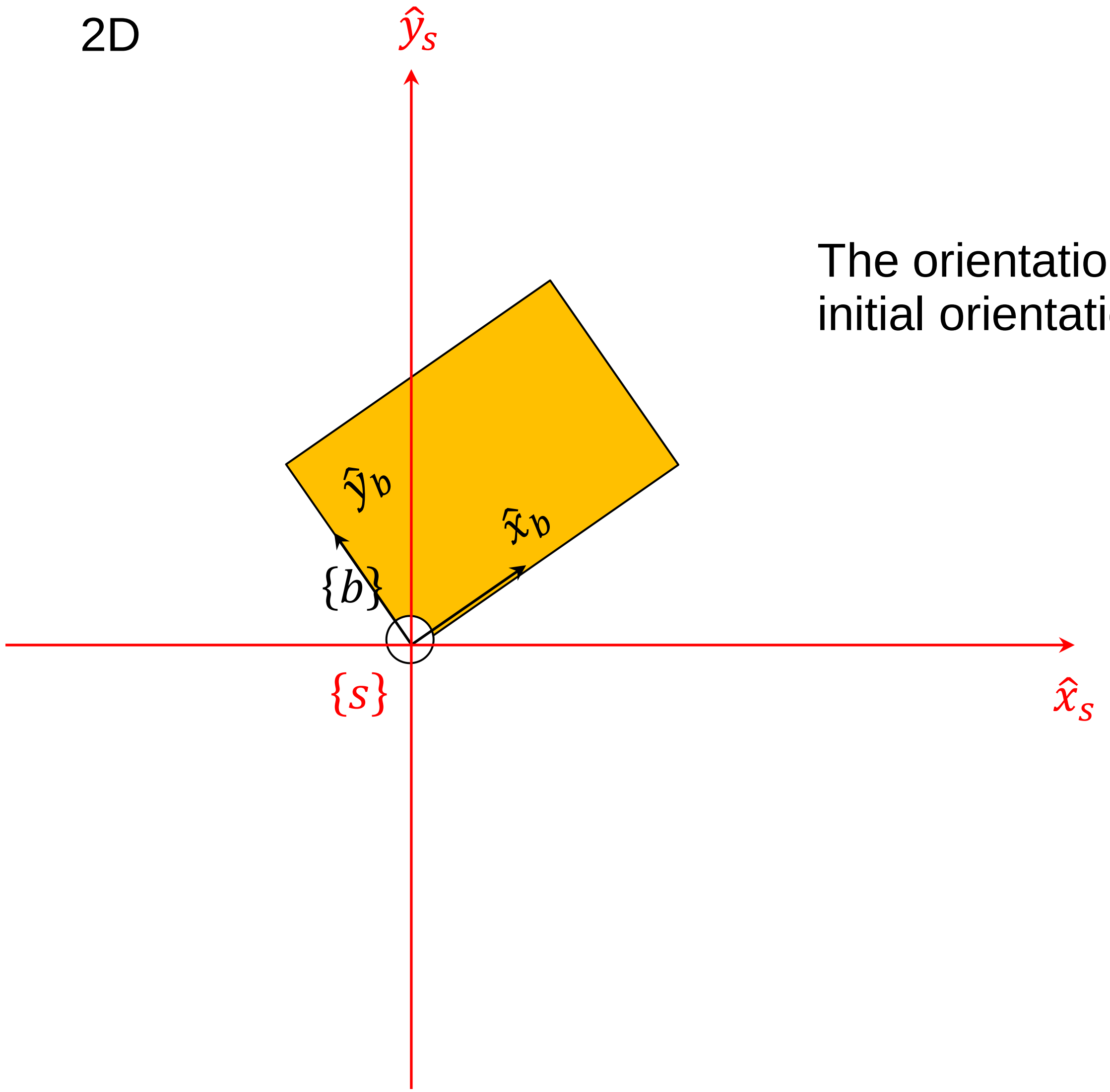


Let's remove the position displacement $\boldsymbol{p}^s$ …

We consider the orientation of {b} with reference to {s}. Alternatively, think of the orientation of {b} is a rotation from the initial orientation of {s}.

Orientation of a rigid body: Rotation

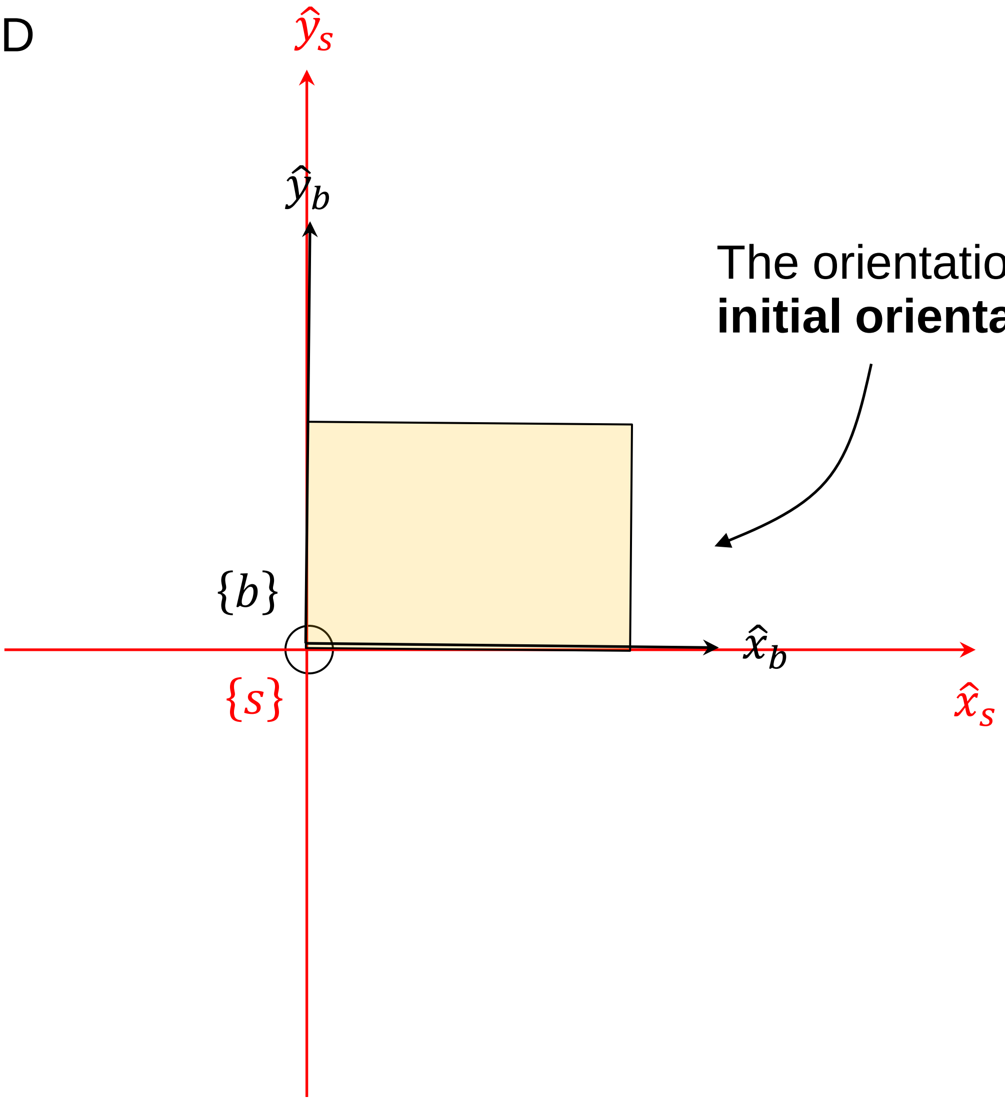
2D



The orientation of {b} is a rotation from the initial orientation of {s}.

Orientation of a rigid body: Rotation

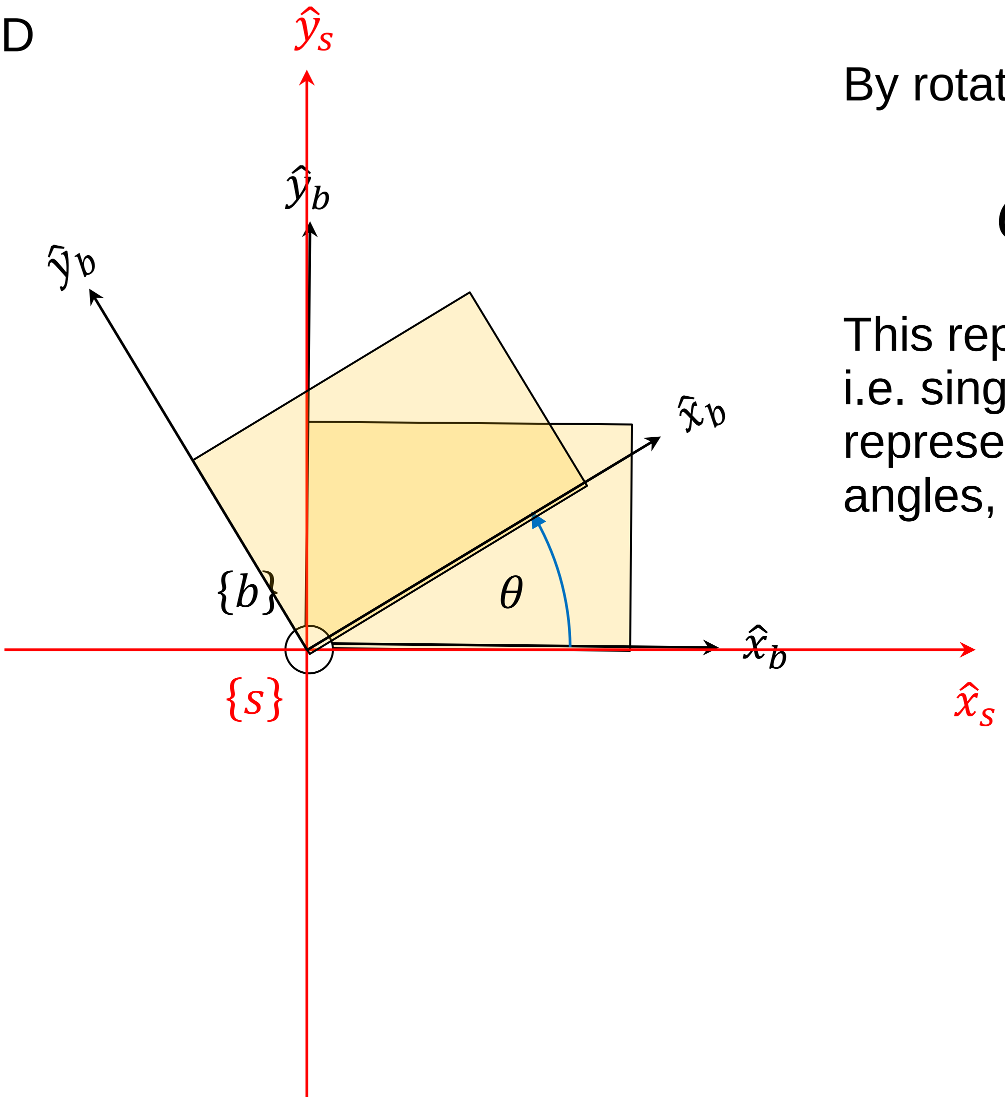
2D



The orientation of {b} is a rotation from the **initial orientation** of {s}.

Orientation of a rigid body: Rotation

2D



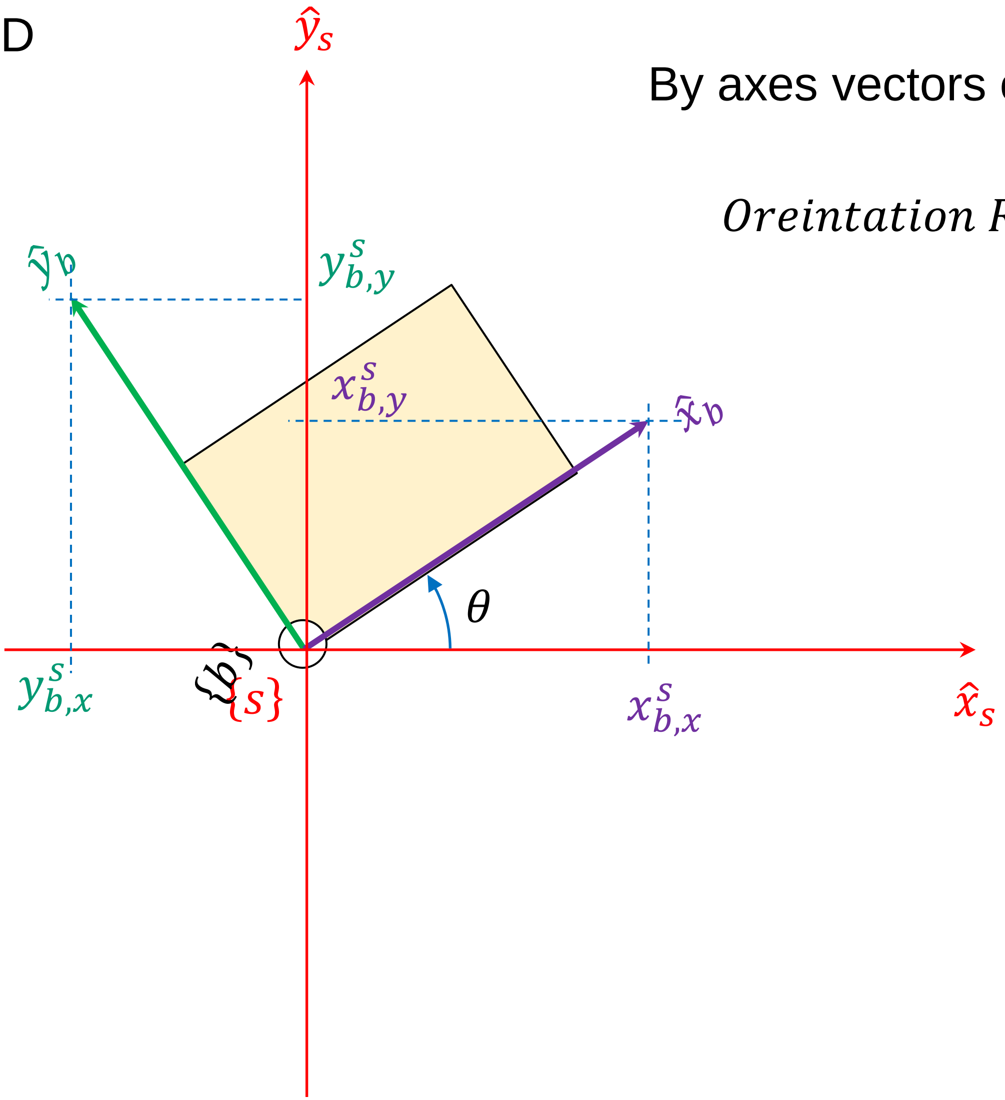
By rotation angle (RHR),

$$\textit{Orientation}\ R = \theta$$

This representation is not continuous, i.e. singularity at $0°$. In 3D space, the representation will specify three angles, i.e. for the three axes.

Orientation of a rigid body: Rotation

2D

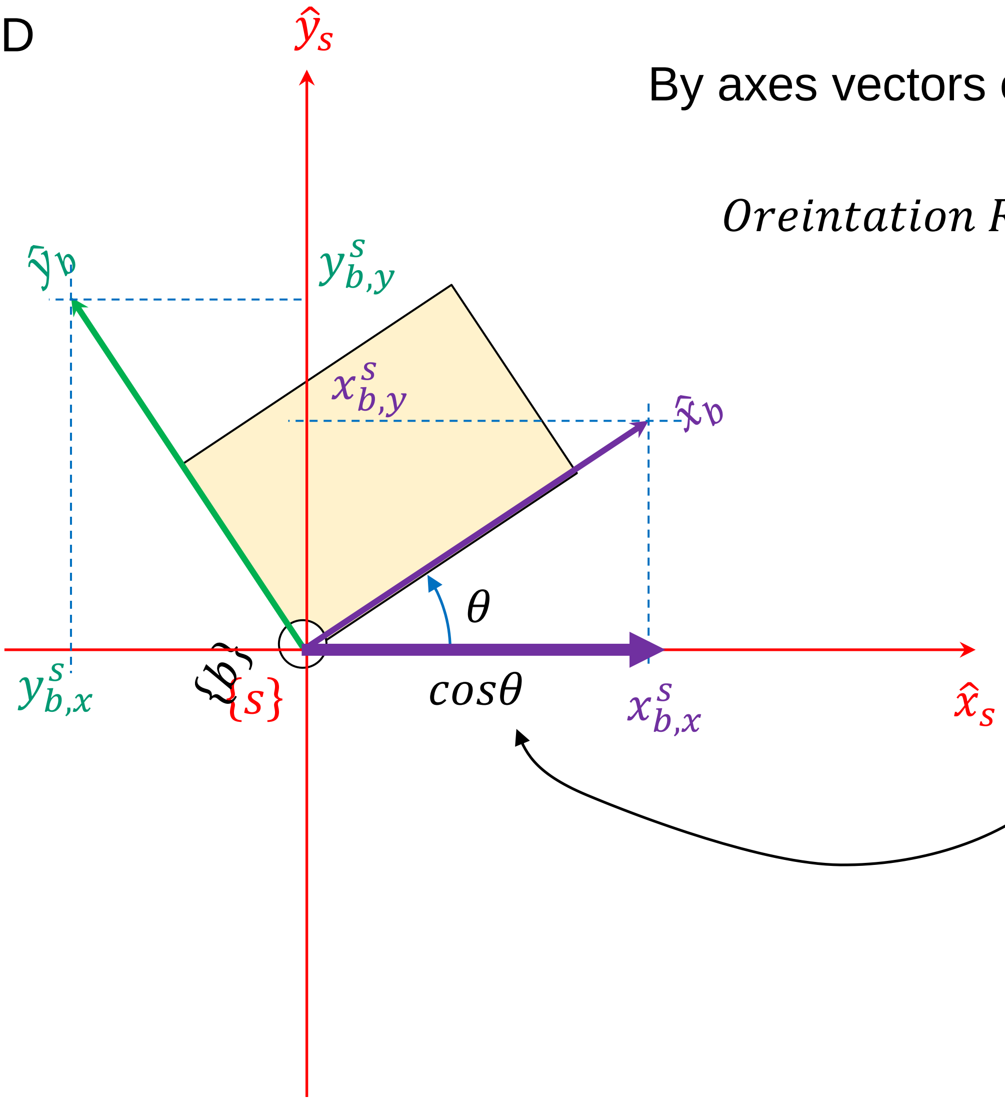


By axes vectors of {b} in {s},

$$Oreintation\ R = [\hat{x}^s_b \quad \hat{y}^s_b] = \begin{bmatrix} x^s_{b,x} & y^s_{b,x} \\ x^s_{b,y} & y^s_{b,y} \end{bmatrix}$$

Orientation of a rigid body: Rotation

2D



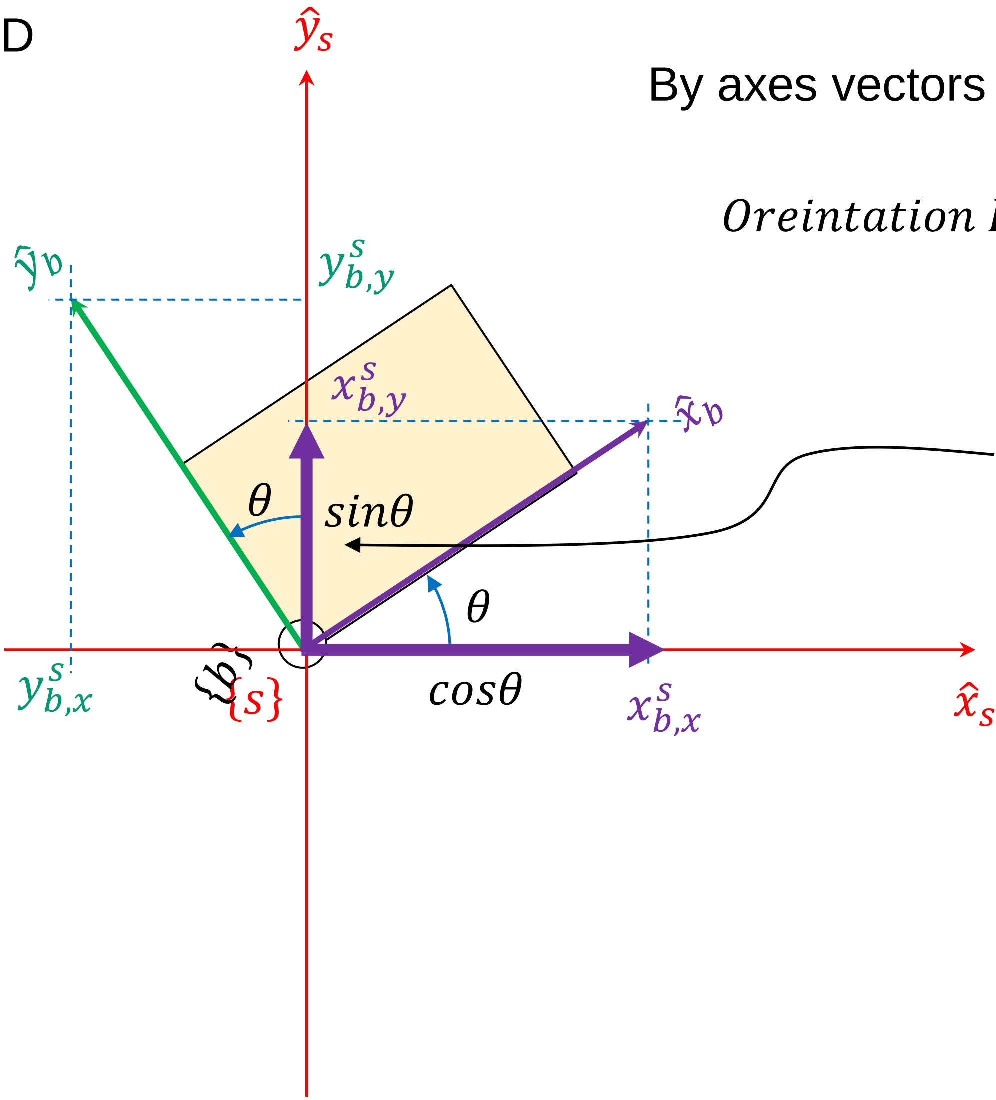
By axes vectors of {b} in {s},

$$Oreintation\ R = [\hat{x}^s_b \quad \hat{y}^s_b] = \begin{bmatrix} x^s_{b,x} & y^s_{b,x} \\ x^s_{b,y} & y^s_{b,y} \end{bmatrix}$$

$$x^s_{b,x} = \|\hat{x}_b\| cos(\theta) = cos\theta$$

Orientation of a rigid body: Rotation

2D



By axes vectors of {b} in {s},

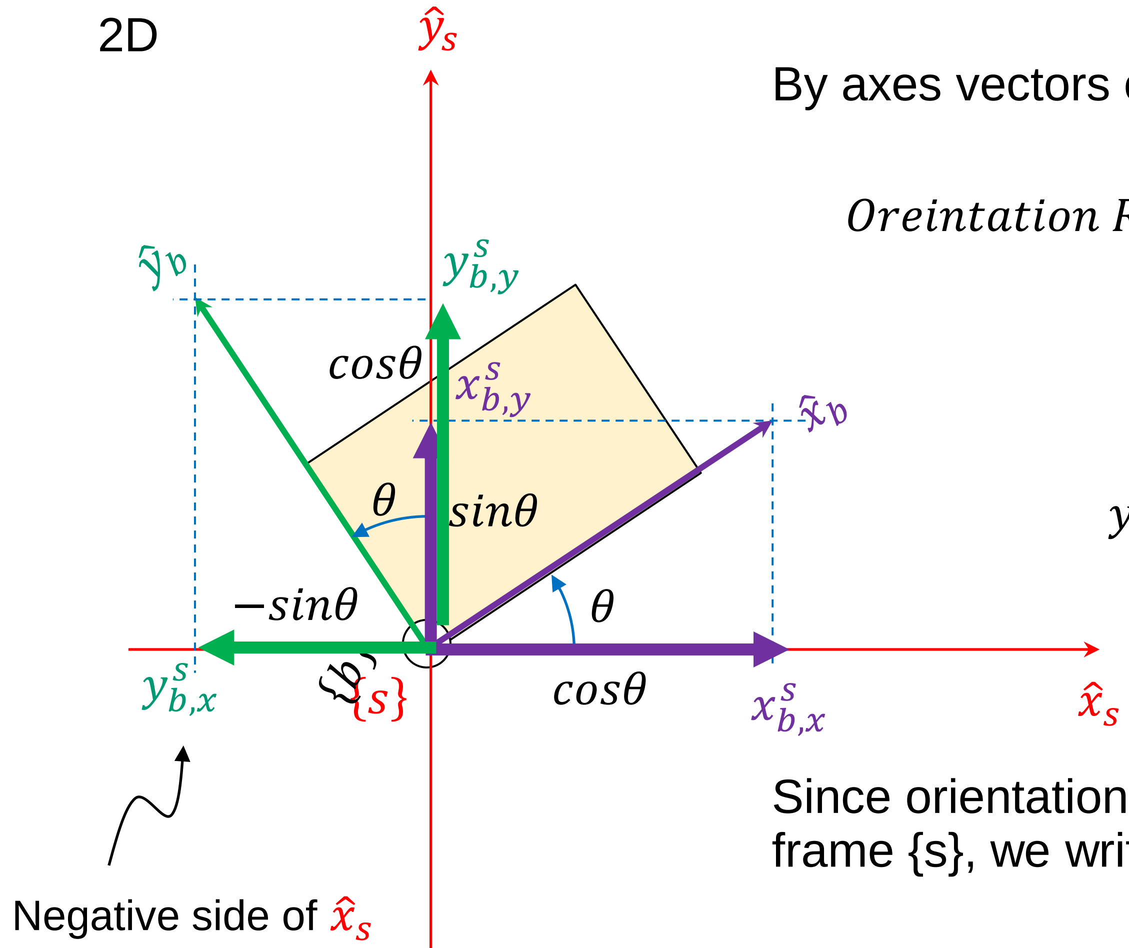
$$Oreintation\ R = [\hat{x}^s_b \quad \hat{y}^s_b] = \begin{bmatrix} x^s_{b,x} & y^s_{b,x} \\ x^s_{b,y} & y^s_{b,y} \end{bmatrix}$$

$$x^s_{b,x} = \|\hat{x}_b\| cos(\theta) = cos\theta$$

$$x^s_{b,y} = \|\hat{x}_b\| sin(\theta) = sin\theta$$

Orientation of a rigid body: Rotation

2D



By axes vectors of {b} in {s},

$$Oreintation\ R = [\hat{x}^s_b \quad \hat{y}^s_b] = \begin{bmatrix} x^s_{b,x} & y^s_{b,x} \\ x^s_{b,y} & y^s_{b,y} \end{bmatrix}$$

$$x^s_{b,x} = \|\hat{x}_b\|cos(\theta) = cos\theta$$
$$x^s_{b,y} = \|\hat{x}_b\|sin(\theta) = sin\theta$$
$$y^s_{b,x} = -\|\hat{y}_b\|sin(\theta) = -sin\theta$$
$$y^s_{b,y} = \|\hat{y}_b\|cos(\theta) = cos\theta$$

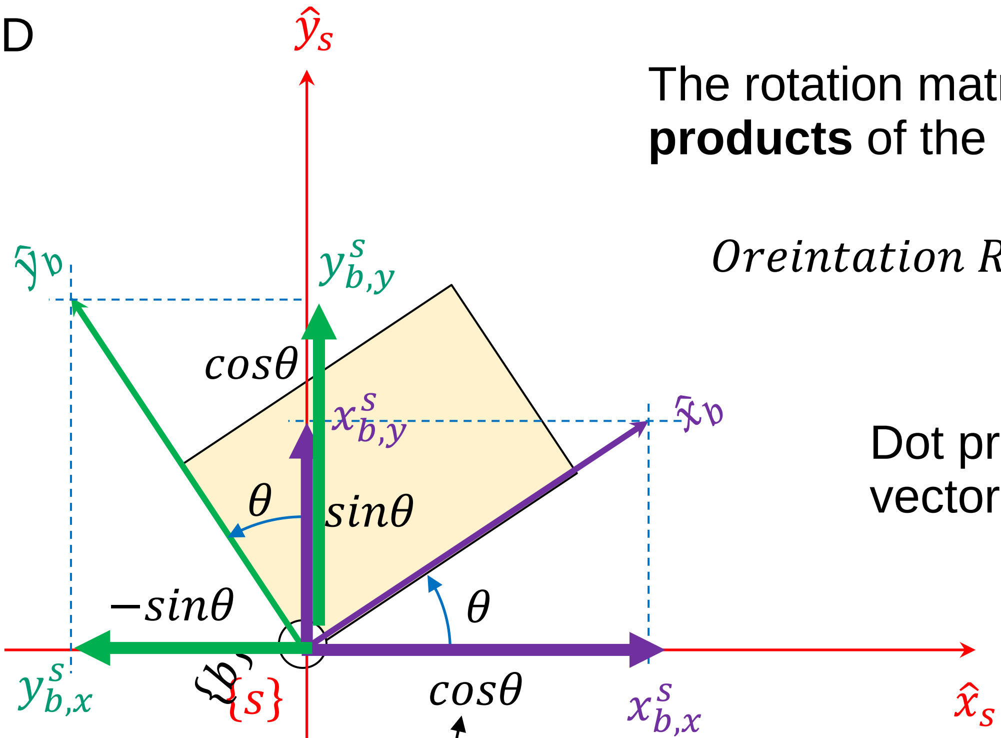
Since orientation is rotation of frame {b} in frame {s}, we write

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

R is called a **rotation matrix**.

Orientation of a rigid body: Rotation

2D

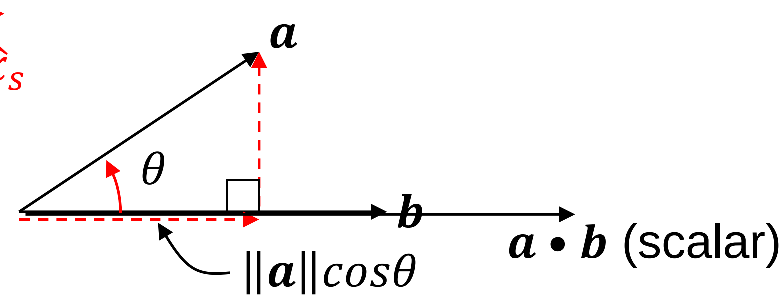


The rotation matrix can be expressed as **dot products** of the axes

$$Oreintation\ R^s_b = [\hat{x}^s_b \quad \hat{y}^s_b] = \begin{bmatrix} x^s_{b,x} & y^s_{b,x} \\ x^s_{b,y} & y^s_{b,y} \end{bmatrix}$$

Dot product of two vectors project one vector on to another vector.

$$\boldsymbol{a} \bullet \boldsymbol{b} = \|\boldsymbol{a}\|\|\boldsymbol{b}\|cos\theta$$



$$x^s_{b,x} = \hat{x}_b \bullet \hat{x}_s$$

$$x^s_{b,y} = \hat{x}_b \bullet \hat{y}_s$$

$$y^s_{b,x} = \hat{y}_b \bullet \hat{x}_s$$

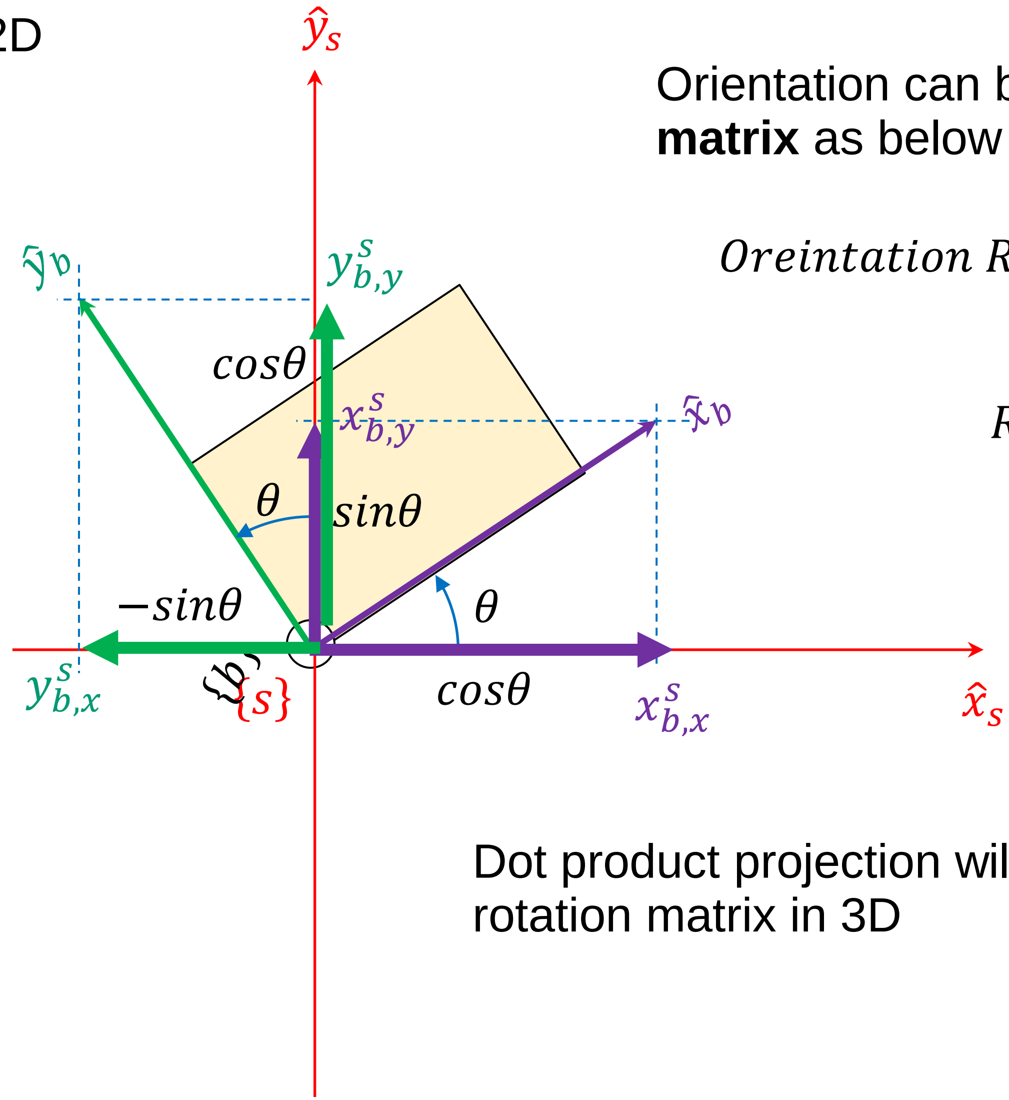
$$y^s_{b,y} = \hat{y}_b \bullet \hat{y}_s$$

$$R^s_b = \begin{bmatrix} \hat{x}_b \bullet \hat{x}_s & \hat{y}_b \bullet \hat{x}_s \\ \hat{x}_b \bullet \hat{y}_s & \hat{y}_b \bullet \hat{y}_s \end{bmatrix}$$

$$\boldsymbol{a} \bullet \boldsymbol{b} = cos\theta\ if\ \|\boldsymbol{a}\| = \|\boldsymbol{b}\| = 1$$

Orientation of a rigid body: Rotation

2D



Orientation can be represented by a **rotation matrix** as below

$$Oreintation\ R^s_b = [\hat{x}^s_b \quad \hat{y}^s_b] = \begin{bmatrix} x^s_{b,x} & y^s_{b,x} \\ x^s_{b,y} & y^s_{b,y} \end{bmatrix}$$

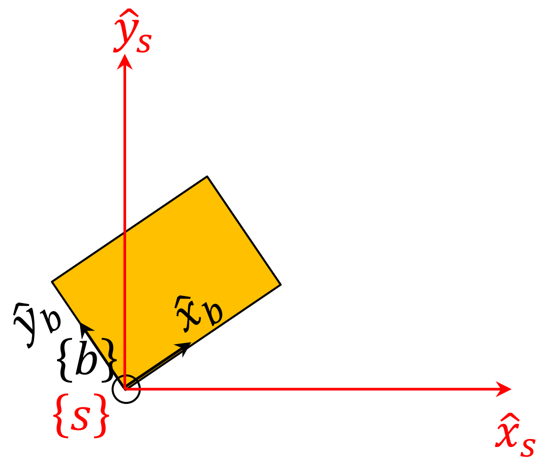
$$R^s_b = \begin{bmatrix} \hat{x}_b \bullet \hat{x}_s & \hat{y}_b \bullet \hat{x}_s \\ \hat{x}_b \bullet \hat{y}_s & \hat{y}_b \bullet \hat{y}_s \end{bmatrix}$$

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

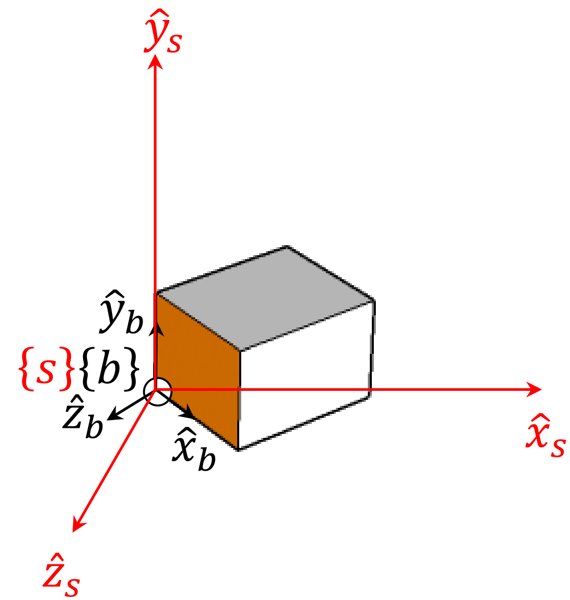
Dot product projection will come handy to formulate the rotation matrix in 3D

Orientation of a rigid body: Rotation

2D



3D



$$R_b^s = [\hat{x}_b^s \quad \hat{y}_b^s] = \begin{bmatrix} x_{b,x}^s & y_{b,x}^s \\ x_{b,y}^s & y_{b,y}^s \end{bmatrix}$$

$$R_b^s = \begin{bmatrix} \hat{x}_b \bullet \hat{x}_s & \hat{y}_b \bullet \hat{x}_s \\ \hat{x}_b \bullet \hat{y}_s & \hat{y}_b \bullet \hat{y}_s \end{bmatrix}$$

$$R_b^s = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix}$$

$$R_b^s = [\hat{x}_b^s \quad \hat{y}_b^s \quad \hat{z}_b^s] = \begin{bmatrix} x_{b,x}^s & y_{b,x}^s & z_{b,x}^s \\ x_{b,y}^s & y_{b,y}^s & z_{b,y}^s \\ x_{b,z}^s & y_{b,z}^s & z_{b,z}^s \end{bmatrix}$$

$$R_b^s = \begin{bmatrix} \hat{x}_b \bullet \hat{x}_s & \hat{y}_b \bullet \hat{x}_s & \hat{z}_b \bullet \hat{x}_s \\ \hat{x}_b \bullet \hat{y}_s & \hat{y}_b \bullet \hat{y}_s & \hat{z}_b \bullet \hat{y}_s \\ \hat{x}_b \bullet \hat{z}_s & \hat{y}_b \bullet \hat{z}_s & \hat{z}_b \bullet \hat{z}_s \end{bmatrix}$$

Properties of rotation matrix

- Because the columns of the rotation matrix are axes, they are orthogonal to each other. The rotation matrix is an **orthogonal matrix**.

$$R_b^s = [\hat{x}_b^s \quad \hat{y}_b^s] \quad R_b^s = [\hat{x}_b^s \quad \hat{y}_b^s \quad \hat{z}_b^s]$$

- Since all column vectors are unit vectors, the determinant of the rotation matrix is 1, i.e. $det\{R\} = 1$ (RHR). *R* does not change the length of vector it multiplies.
- The rotation matrix belongs to the **special orthorgonal group $\boldsymbol{SO}(\boldsymbol{n})$** where $n = 3$ is the dimension of the matrix. Two useful properties:

$$R^{-1} = R^T$$

- Product is a rotation matrix $R_1 R_2 \in SO(n)$.
- The above property is useful to compute its inverse when solving transformation problems.
- Other useful properties: 1. closure, 2. associativity, 3. identity element existence, 4. inverse element existence.

Directional cosine representation

- Rotation matrices contain *sin* and *cosine* of the angle(s) between the axis vectors. Representation in this form is sometimes referred as **direction cosine**.
  - The elements in *R* can be given (flattened) in a single column vector: 4 parameters for 2x2 and 9 parameters for 3x3 R matrices.
- This is an implicit representation of the orientation.
  - For 2x2, there are 4 parameters to represent 1 dof in orientation on a plane. There are 3 constraints: two columns are unit vectors, columns are orthogonal.
  - Likewise, for 3x3, there are 9 parameters to represent 3 dof in orientation in the space. There are 6 constraints: three columns are unit vectors, columns are orthogonal.
  - Implicit representation avoids singularity in the representation.

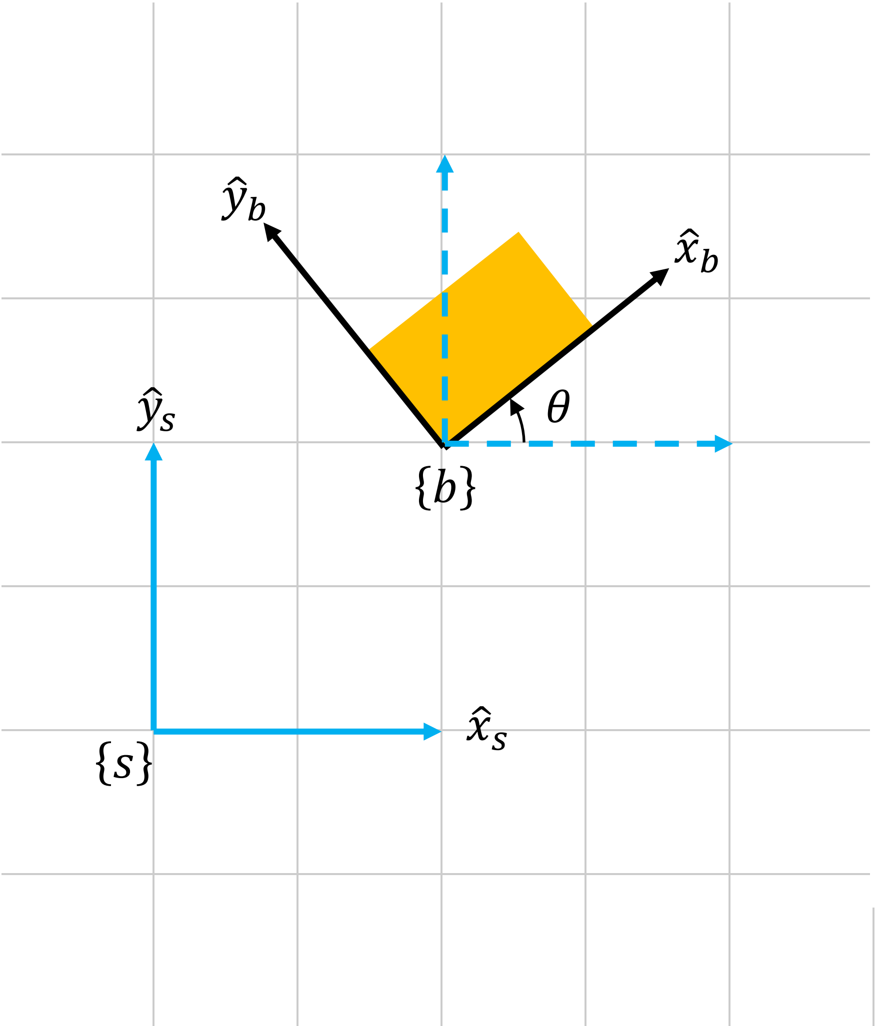
$$R_b^s = \begin{bmatrix} cos\theta \\ sin\theta \\ -sin\theta \\ cos\theta \end{bmatrix} \quad R_b^s = \begin{bmatrix} x_{b,x}^s & x_{b,y}^s & x_{b,z}^s & y_{b,x}^s & y_{b,y}^s & y_{b,z}^s & z_{b,x}^s & z_{b,y}^s & z_{b,z}^s \end{bmatrix}^T$$

More on orientation and rotation

Three uses of rotation matrix

- Rotation matrix can be used for three purposes:
  1. **Represent orientation** of a vector or coordinate frame (attached to an object)
  2. **Rotate** a vector or a coordinate frame in the same reference coordinate frame
  3. **Change reference frame** of a vector (point) or a coordinate frame
- In second case, the rotation matrix is used to describe a rotational motion or rotational transformation.
- In second and third cases, the rotation matrix is used as an operator, i.e. to be multiplied on a vector or frame.

Represent orientation



The orientation of a rigid body in space is represented by the orientation of the coordinate frame {b} attached to it, with reference to a space coordinate frame {s}.

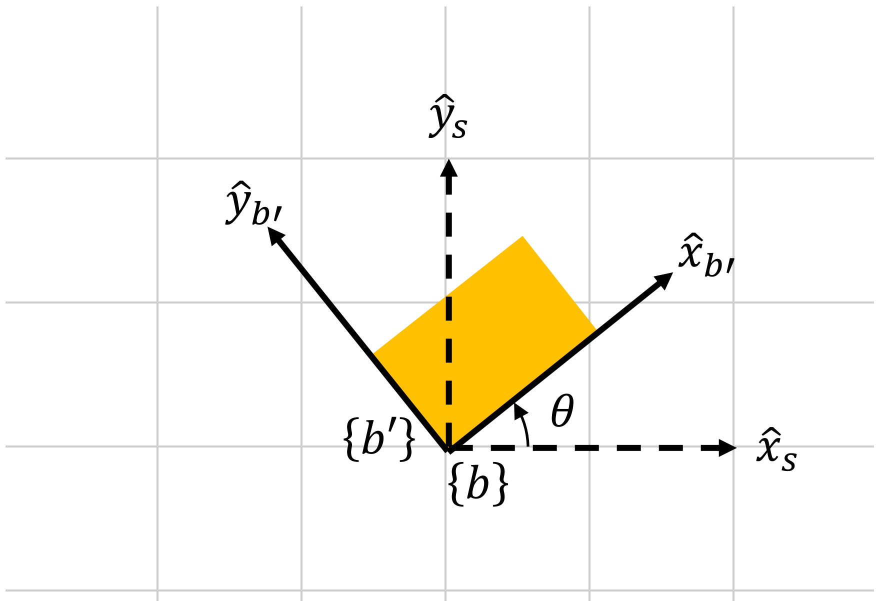
Rotation matrix can be used to represent the orientation of the coordinate frame. Example for 2D space (plane):

$$R_b^s = [\hat{x}_b^s \quad \hat{y}_b^s] = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix}$$

Likewise for 3D,

$$R_b^s = [\hat{x}_b^s \quad \hat{y}_b^s \quad \hat{z}_b^s]$$

Rotate a coordinate frame



Orientation of {b'}, $R_{b'}$ is the result of rotating {b}, R_b by the rotation expressed by R.

$$R_{b'} = R R_b$$

The rotation of a rigid body in space can be represented by the rotation of a coordinate frame {b} attached to it from its initial orientation {b} to its new orientation {b'}.

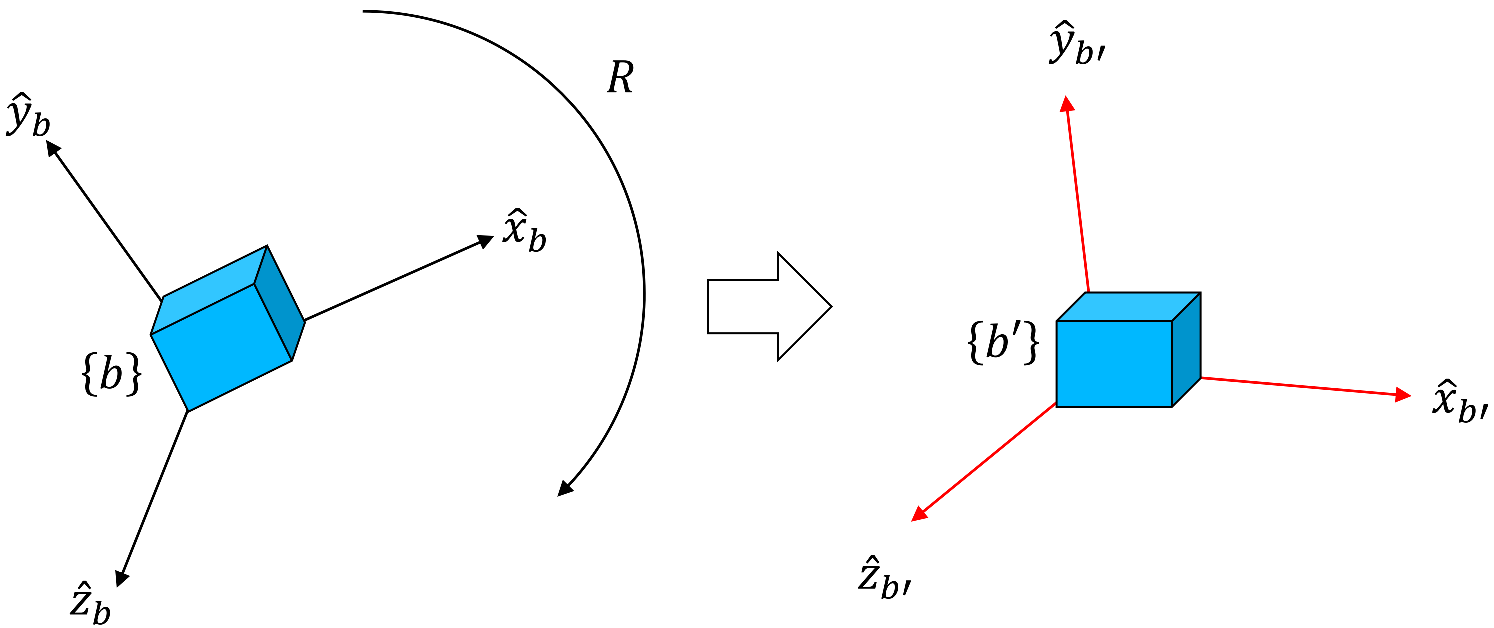
Rotation matrix can be used to represent the rotation of the coordinate frame. Example for 2D space (plane):

$$R = [\hat{x}_{b'}^{b} \quad \hat{y}_{b'}^{b}] = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix}$$

Likewise for 3D,

$$R = [\hat{x}_{b'}^{b} \quad \hat{y}_{b'}^{b} \quad \hat{z}_{b'}^{b}]$$

Rotation of a coordinate frame

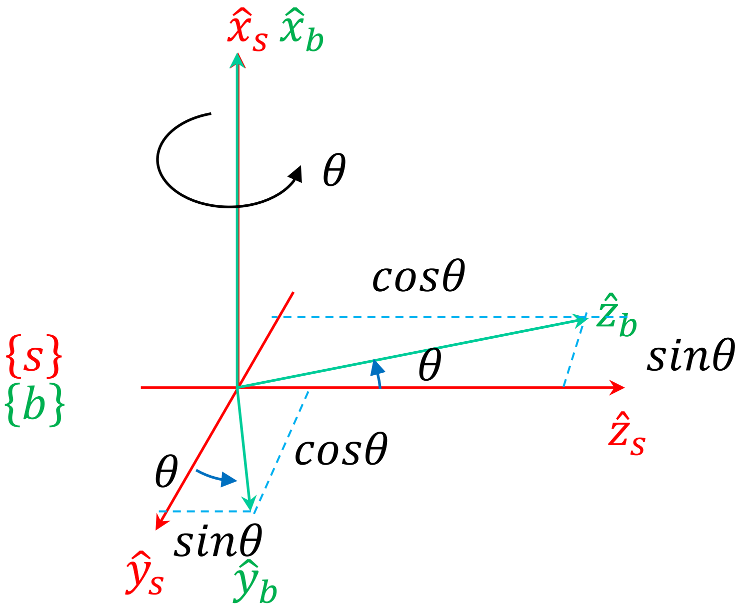


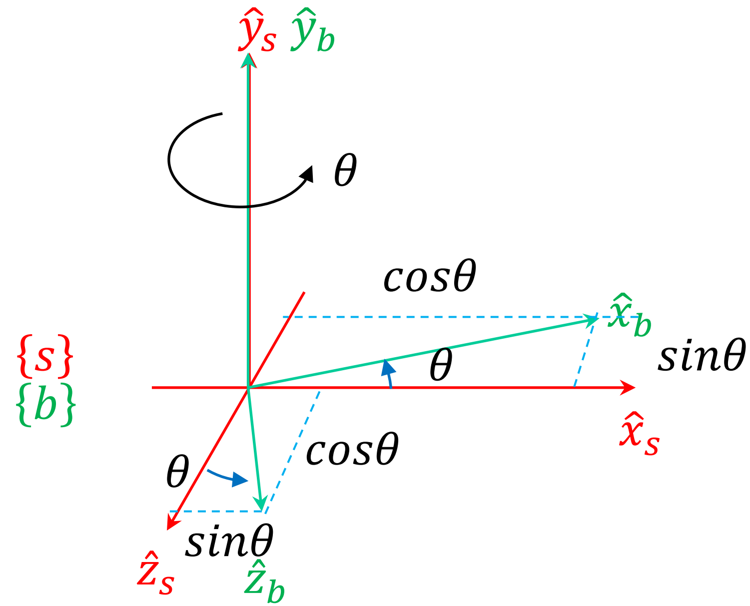
$$Orientation\ of\ \{b'\}\ wrt\ \{b\}\ R_{b'}^{b} = RR_b$$

$$R_b\ is\ the\ reference\ coordinate\ frame\ (unrotated), R_b = I,$$
$$Rotation\ R = R_{b'}^{b}$$

In general, R can be composed of a series of independent rotations around the principal axes (x,y,z)

Basic rotation matrices in 3D





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

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

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

Composition of rotations

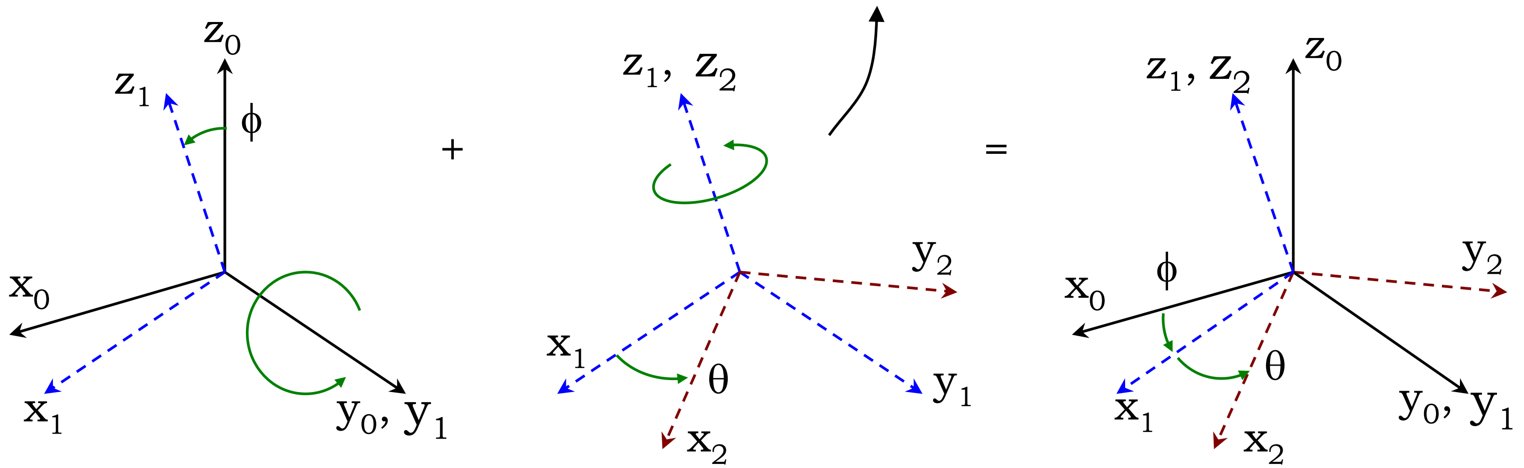
- Cumulative effect of rotations about different axes can be computed. There are two ways of computing the total rotation depending on:
  - If all the rotations are specified **with respect to their current frame**
  - If all the rotations are specified **with respect to a fixed (same) frame**

- Rotations with respect to **current (relative) frames – post-multiply** the rotations to compute overall rotation
- Rotation with respect to a **fixed (same) frame – pre-multiply** the rotations to compute overall rotation

Rotations with respect to current frames

- If we rotate from frame {0} to frame {1} by R_1^0 followed by rotation of frame {1} to frame {2} by R_2^1, the overall rotation is given by **post multiplication**:

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



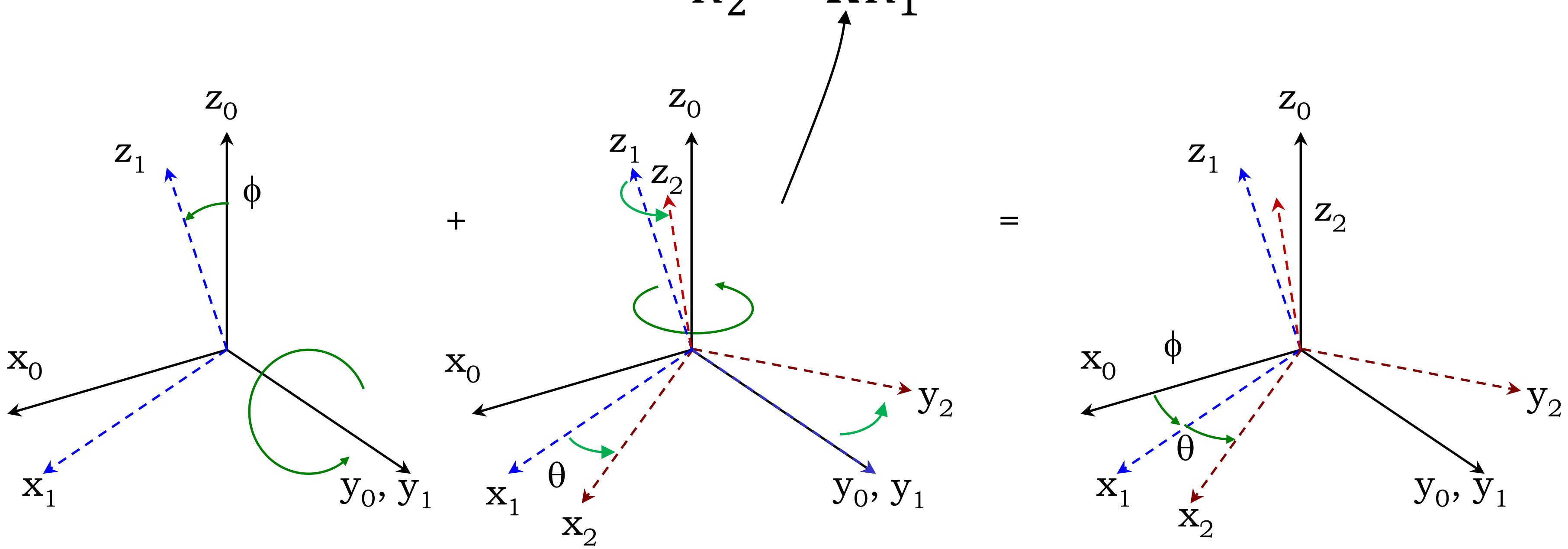
R_j^i is used to denote rotation from frame **{i}** to **{j}**, which also means orientation of frame **{j}** with reference to frame **{i}**.

In general, $R_n^0 = R_1^0 R_2^1 R_3^2 \cdots R_n^{n-1}$

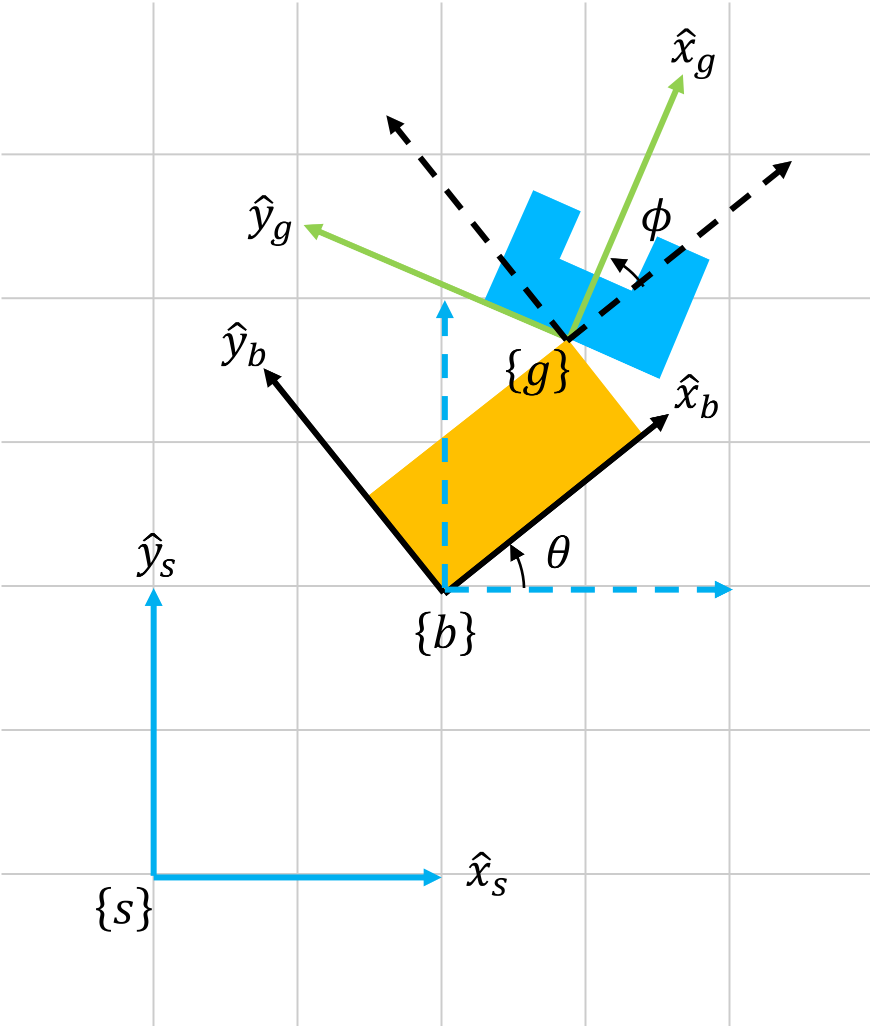
Rotation with respect to a fixed frame

- If we rotate from frame {0} to frame {1} by R_1^0 followed by rotation of frame {1} to frame {2} by a rotation specified with reference to frame {0}. Let R represent the second rotation. We will determine the overall rotation by **pre multiplication**.

$$R_2^0 = \boldsymbol{R} R_1^0$$



Change reference frame



We know the orientation of the gripper represented by {g} on the effector represented {b}

$$R_g^b = \begin{bmatrix} cos\phi & -sin\phi \\ sin\phi & cos\phi \end{bmatrix}$$

We know the orientation of {b} in the space {s}

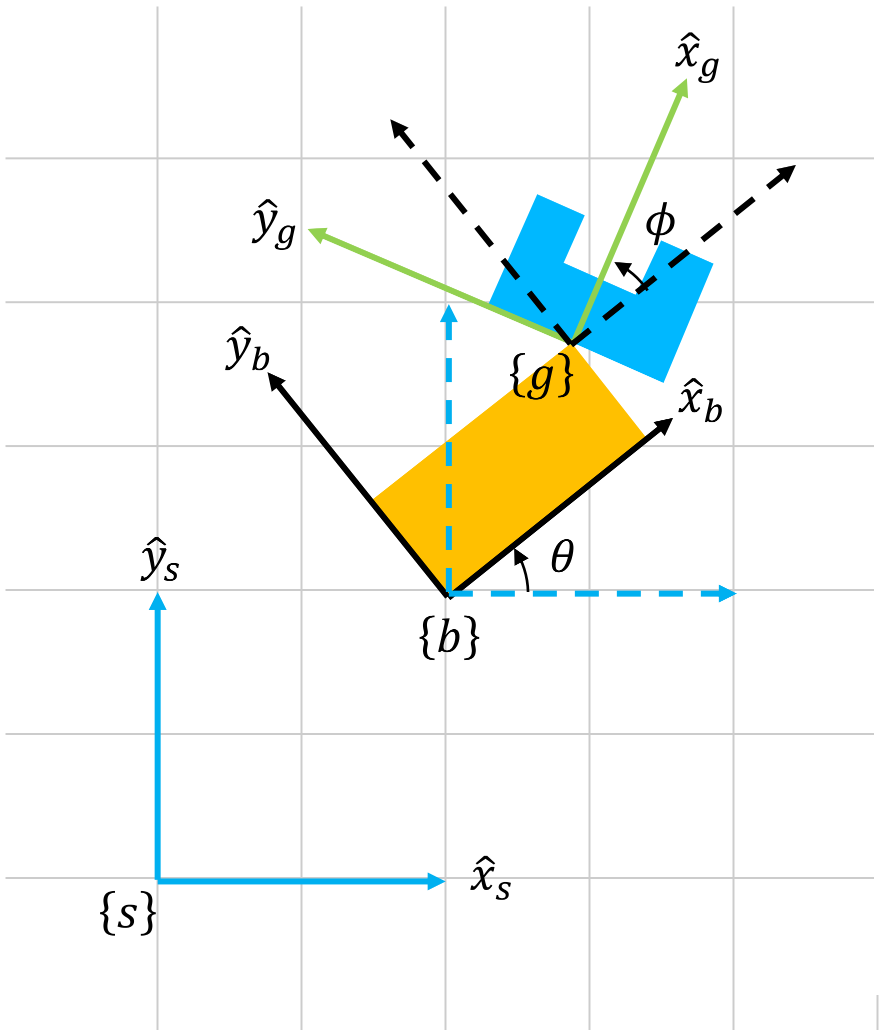
$$R_b^s = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix}$$

We can determine the orientation of the gripper {g} in space {s}, i.e. change its reference frame from {b} to {s}

$$R_g^s = R_b^s R_g^b$$

Likewise, if it is in 3D.

Change reference frame



We know the orientation of the gripper represented by {g} on the effector represented {b}

$$R_g^b = \begin{bmatrix} cos\phi & -sin\phi \\ sin\phi & cos\phi \end{bmatrix}$$

We know the orientation of {b} in the space {s}

$$R_b^s = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix}$$

We can determine the orientation of the gripper {g} in space {s}, i.e. change its reference frame from {b} to {s}

$$R_g^s = R_b^s R_g^b$$

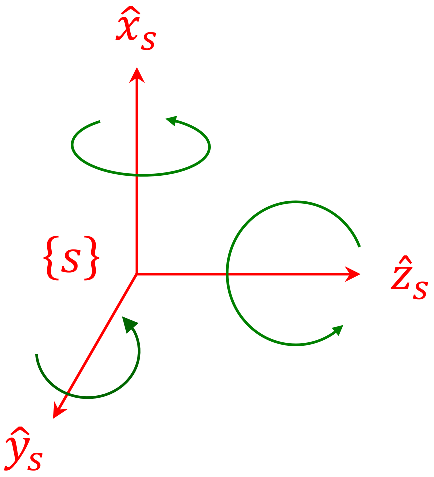
Likewise, if it is in 3D.

Directional cosine representation: cons

- The redundancy in the **directional cosine representation** requires significant care in maintaining the constraints during interpolation. **Makes interpolation of all parameters difficult**.
- In addition, determining the rotational (angular) **velocities** is not straight forward since the parameters are not angles, i.e. we cannot determine angular velocities as $\dot{R}$.
- There are alternative **angular representations**. We can represent orientation and rotation by angles.
- These angles can also be used to **parameterize the rotation matrix** in order to use the rotation matrix for ease of calculations, e.g. combining rotations.

Three-angles representations

- A rigid body uses 3 dof to achieve its desire orientation.
- An orientation can be achieved by a series of independent rotation around three arbitrary axes.
- Three angles representation suffers from the problem of singularity.

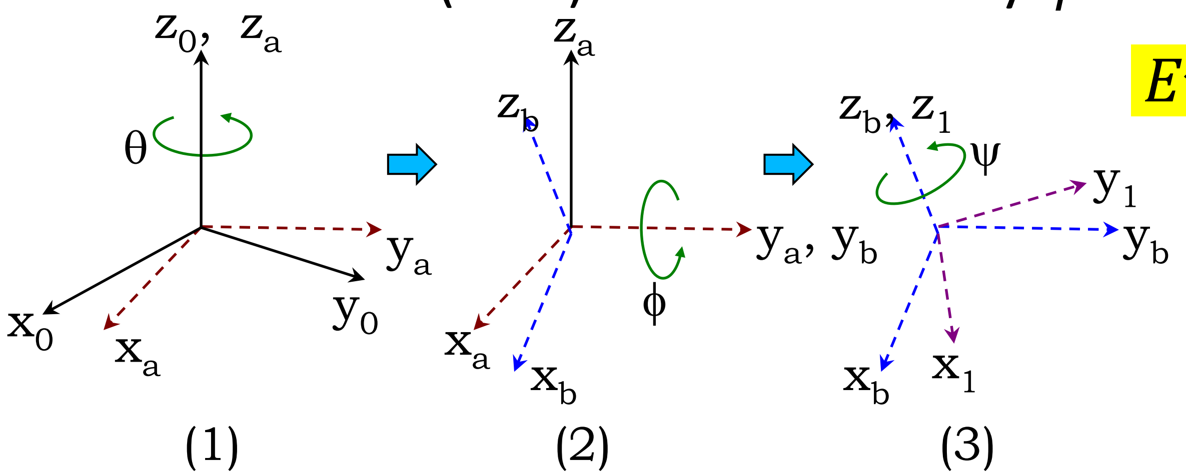


Possible sequences:
XYX XYZ XZX XZY
YXY YXZ YZY YZX
ZXZ ZXY ZYZ ZYX

Common names:
1. Euler angles
2. Yaw-pitch-roll angles
3. Axis-angle
4. Exponential coordinates

Euler Angles

- Rotate around **current** frame, i.e. relative rotation – post-multiplication.
- Most common is **ZYZ**: rotate about current z-axis by θ followed by rotate about (new) current y-axis by ϕ and finally rotate about (new) current z-axis by ψ.



$Euler\ angles$: θ, ϕ, ψ

XYX XYZ **XZX** XZY
YXY YXZ **YZY** YZX
ZXZ ZXY **ZYZ** ZYX

$$R_1^0 = R_{z,\theta} R_{y,\phi} R_{z,\psi} = \begin{bmatrix} c_\theta c_\phi c_\psi - s_\theta s_\psi & -c_\theta c_\phi s_\psi - s_\theta c_\psi & c_\theta s_\phi \\ s_\theta c_\phi c_\psi + c_\theta s_\psi & -s_\theta c_\phi s_\psi + c_\theta c_\psi & s_\theta s_\phi \\ -s_\phi c_\psi & s_\phi s_\psi & c_\phi \end{bmatrix}$$

(Forward problem)

$c_\alpha = cos\ \alpha \quad s_\alpha = sin\ \alpha \quad t_\alpha = tan\ \alpha$

Euler Angles: inverse problem

- Given the rotation matrix, we can solve linear equations to determine the Euler angles. However, there are multiple solutions.

$$Given\ R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} c_\theta c_\phi c_\psi - s_\theta s_\psi & -c_\theta c_\phi s_\psi - s_\theta c_\psi & c_\theta s_\phi \\ s_\theta c_\phi c_\psi + c_\theta s_\psi & -s_\theta c_\phi s_\psi + c_\theta c_\psi & s_\theta s_\phi \\ -s_\phi c_\psi & s_\phi s_\psi & c_\phi \end{bmatrix}$$

If not both r_{13} and r_{23} are *zero*, $|r_{33}| \neq 1$, $s_\phi \neq 0$ – **two solutions**

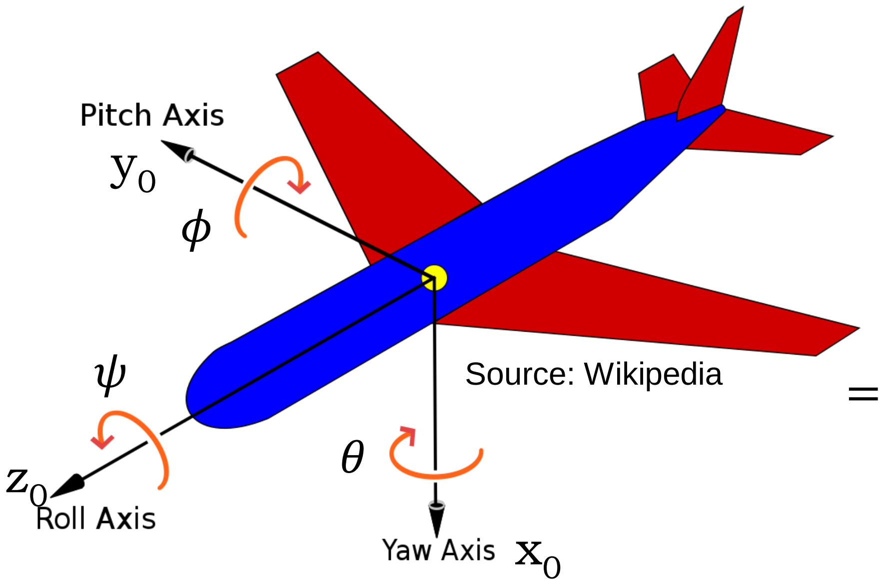
Solution 1 ($s_\phi > 0$)	**Solution 2 ($s_\phi < 0$)**
$\theta = \text{Atan2}(r_{13}, r_{23})$	$\theta = \text{Atan2}(-r_{13}, -r_{23})$
$\phi = \text{Atan2}\left(r_{33}, \sqrt{1 - r_{33}^2}\right)$	$\phi = \text{Atan2}\left(r_{33}, -\sqrt{1 - r_{33}^2}\right)$
$\psi = \text{Atan2}(-r_{31}, r_{32})$	$\psi = \text{Atan2}(r_{31}, -r_{32})$

If $r_{13} = r_{23} = 0$, $r_{33} = 1$, or $r_{13} = r_{23} = 0$, $r_{33} = -1$, cannot resolve θ and ψ.

(Singularity)

Fix frame Yaw-Pitch-Roll

- Rotate around **fixed** (stationary) frame – pre-multiplication.
- Rotate about x-axis (Yaw) by θ with respect to a base frame then rotate about y-axis (Pitch) by ϕ with respect to the same base frame and finally rotate about z-axis (Roll) by φ with respect to the same base frame. Performing the sequence in reverse order, Roll-Pitch-Yaw, will yield the same result.



Yaw-pitch-roll: θ, ϕ, ψ

$$R = R_{z,\psi} R_{y,\phi} R_{x,\theta}$$

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

(Forward problem)

Fix frame Yaw-Pitch-Roll: inverse problem

$$Given\ R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} c_\psi c_\phi & -s_\phi c_\theta + c_\psi s_\phi s_\theta & s_\psi s_\theta + c_\psi s_\phi c_\phi \\ s_\psi c_\phi & c_\psi c_\theta + s_\psi s_\phi s_\theta & -c_\psi s_\theta + s_\psi s_\phi c_\theta \\ -s_\phi & c_\phi s_\theta & c_\phi c_\theta \end{bmatrix}$$

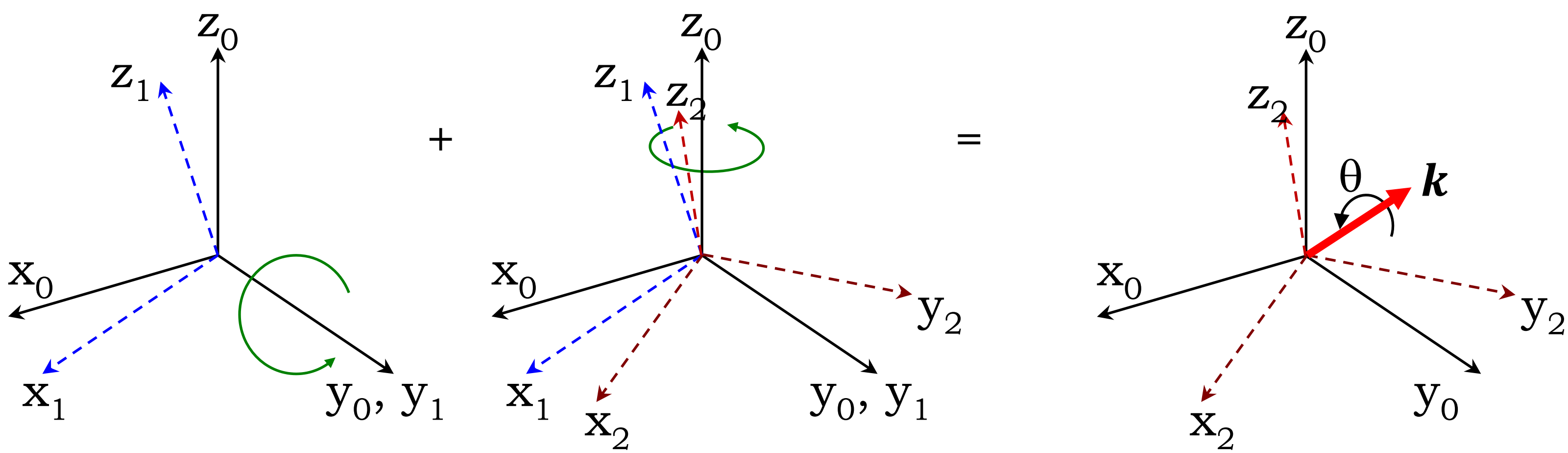
If not both r_{11} and r_{21} are *zero*, $|r_{31}| \neq 1$, $c_\phi \neq 0$ – **two solutions**

Solution 1 ($c_\phi > 0$)	**Solution 2 ($c_\phi < 0$)**
$\phi = \text{Atan2}\left(-r_{31}, \sqrt{1 - r_{31}^2}\right)$	$\phi = \text{Atan2}\left(-r_{31}, -\sqrt{1 - r_{31}^2}\right)$
$\theta = \text{Atan2}(r_{32}, r_{33})$	$\theta = \text{Atan2}(-r_{32}, -r_{33})$
$\psi = \text{Atan2}(r_{21}, r_{11})$	$\psi = \text{Atan2}(-r_{21}, -r_{11})$

If $r_{11} = r_{21} = 0$, $r_{31} = 1$, or $r_{11} = r_{21} = 0$, $r_{31} = -1$, cannot resolve yaw θ and roll ψ. (Singularity)

Axis-Angle

- There exists a **single axis of rotation** for every rotation produced by a given rotation matrix, i.e. every rotation matrix R can be represented by a rotation about an axis $\hat{\boldsymbol{k}}$ by an angle of θ.



$$R_2^0 = R_{\hat{\boldsymbol{k}},\theta} = R(\hat{\boldsymbol{k}},\theta)$$

$$= \begin{bmatrix} (k_x)^2 v_\theta + c_\theta & k_x k_y v_\theta - k_z s_\theta & k_x k_z v_\theta + k_y s_\theta \\ k_x k_y v_\theta + k_z s_\theta & (k_y)^2 v_\theta + c_\theta & k_y k_z v_\theta - k_x s_\theta \\ k_x k_z v_\theta - k_y s_\theta & k_y k_z v_\theta + k_x s_\theta & (k_z)^2 v_\theta + c_\theta \end{bmatrix}$$

$$v_\theta = \text{vers}\,\theta = 1 - c_\theta$$

Axis-angle: $(\hat{\boldsymbol{k}},\theta)$

$$k = [k_x \quad k_y \quad k_z]^T$$

Axis-angle: Rodrigues' formula

- The rotation matrix representing axis-angle can be determined from the **Rodrigues' formula**

$$R_{k,\theta} = e^{[\hat{\boldsymbol{k}}]\theta} = I + S(\hat{\boldsymbol{k}})sin\theta + S(\hat{\boldsymbol{k}})^2(1 - cos\theta)$$

where $\hat{\boldsymbol{k}} = [k_x \quad k_y \quad k_z]^T$ and

$$S(\hat{\boldsymbol{k}}) = \begin{bmatrix} 0 & -k_z & k_y \\ k_z & 0 & -k_x \\ -k_y & k_x & 0 \end{bmatrix}$$ is the skew-symmetric matrix of $\boldsymbol{k}$

- Base on linear differential equation theory.

Different notation conventions have been used by different authors for skew-symmetric matrix. In the above, $s(\hat{\boldsymbol{k}})$ represents the skew-symmetric matrix of a vector $\hat{\boldsymbol{k}}$. The Modern Robotics book uses $[\hat{\omega}]$ to represent the skew-symmetric matrix of a vector $\hat{\omega}$. And, unfortunately, similar notation $[\omega]$ is used to mean the matrix form of a variable ω.
We will be liberal in using different notation conventions. However, please be careful and interpret accordingly depending on the context.

Axis-angle: Exponential coordinates

The axis-angle representation uses *four* parameters (three for $\hat{\boldsymbol{k}}$ and one for θ) to specify a *3-DOF* overall rotation. Since the three axis of $\hat{\boldsymbol{k}}$ are orthogonal, it is possible to reduce one parameter in the representation. The rotation can be represented by a single vector $\boldsymbol{r}$ as:

$$r = [r_x \quad r_y \quad r_z]^T = [\theta k_x \quad \theta k_y \quad \theta k_z]^T$$

Note, since $\hat{\boldsymbol{k}}$ is a unit vector, that the length of the vector $\boldsymbol{r}$ is the equivalent angle θ and the direction of $\boldsymbol{r}$ is the equivalent axis $\hat{\boldsymbol{k}}$.

Axis-angle: $\hat{k}\theta$

The components of $\hat{\boldsymbol{k}}\theta$ are called **exponential coordinates** of the rotation matrix *R*. The above representation is called exponential coordinates as seen in the Rodrigues' formula.

Axis-Angle: inverse problem

$$Given\ R_{k,\theta} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} k_x^2 v_\theta + c_\theta & k_x k_y v_\theta - k_z s_\theta & k_x k_z v_\theta + k_y s_\theta \\ k_x k_y v_\theta + k_z s_\theta & k_y^2 v_\theta + c_\theta & k_y k_z v_\theta - k_x s_\theta \\ k_x k_z v_\theta - k_y s_\theta & k_y k_z v_\theta + k_x s_\theta & k_z^2 v_\theta + c_\theta \end{bmatrix}$$

$$\theta = cos^{-1}\left(\frac{Tr(R) - 1}{2}\right)$$

where Tr denotes the *trace* (sum of diagonal elements) of $\boldsymbol{R}$, and

$$\hat{\boldsymbol{k}} = \begin{bmatrix} k_x \\ k_y \\ k_z \end{bmatrix} = \frac{1}{2\, sin\, \theta} \begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix}$$

Axis of rotation is undefined if $\sin\theta = 0, \theta = 0$ (singularity). For cases when θ is integer multiple of π, there are different formulae (Sec 3.2.3.3).

Note that $R_{k,\theta} = R_{-k,-\theta}$

Quaternion: four parameters

- The axis-angle representation can be embedded in higher dimensional space to **avoid singularity**.
- **Quaternion** representation uses four parameters to represent the orientation. In mathematics, Quaternion is an extended complex number with one real part and three imaginary parts. Think of it as a four-dimensional axes system.

$$q = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} cos\frac{\theta}{2} \\ \hat{\boldsymbol{k}} sin\frac{\theta}{2} \end{bmatrix} \in \mathbb{R}^4$$

- where $\hat{\boldsymbol{k}}$ is the unit vector axis of rotation, $\|q\| = 1$.

$$R = \begin{bmatrix} {q_0}^2 + {q_1}^2 - {q_2}^2 - {q_3}^2 & 2(q_1q_2 - q_0q_3) & 2(q_0q_2 + q_1q_3) \\ 2(q_0q_3 + q_1q_2) & {q_0}^2 - {q_1}^2 + {q_2}^2 - {q_3}^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_0q_1 + q_2q_3) & {q_0}^2 - {q_1}^2 - {q_2}^2 + {q_3}^2 \end{bmatrix}$$

Quaternion: inverse problem

Given $R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$, we can determine

$$q_0 = cos\frac{\theta}{2} = \frac{1}{2}\sqrt{1 + r_{11} + r_{22} + r_{33}}$$

$$\begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = \widehat{\boldsymbol{k}} sin\frac{\theta}{2} = \frac{1}{4q_0}\begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix}$$

Due to the redundancy (4 parameters), there is always a nonzero parameter to choose as q_0 to avoid singularity.

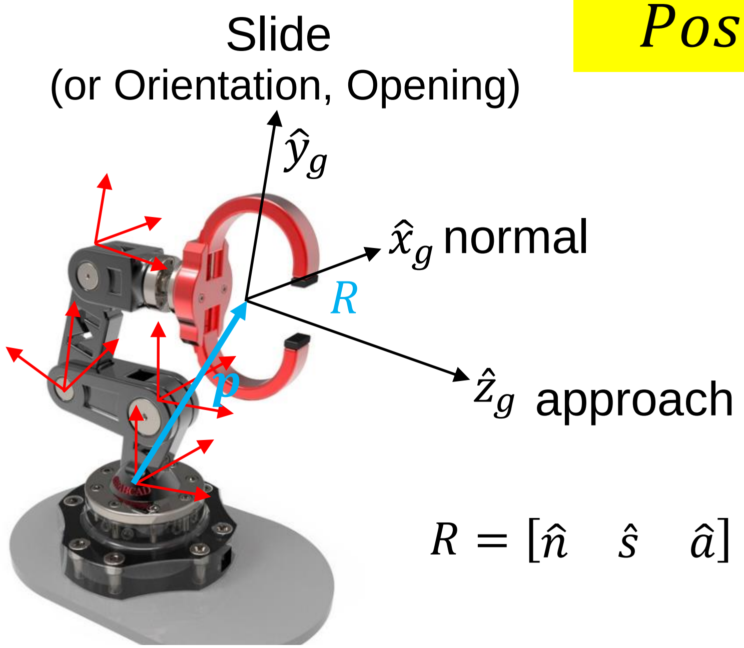
Configuration representation

Describing pose

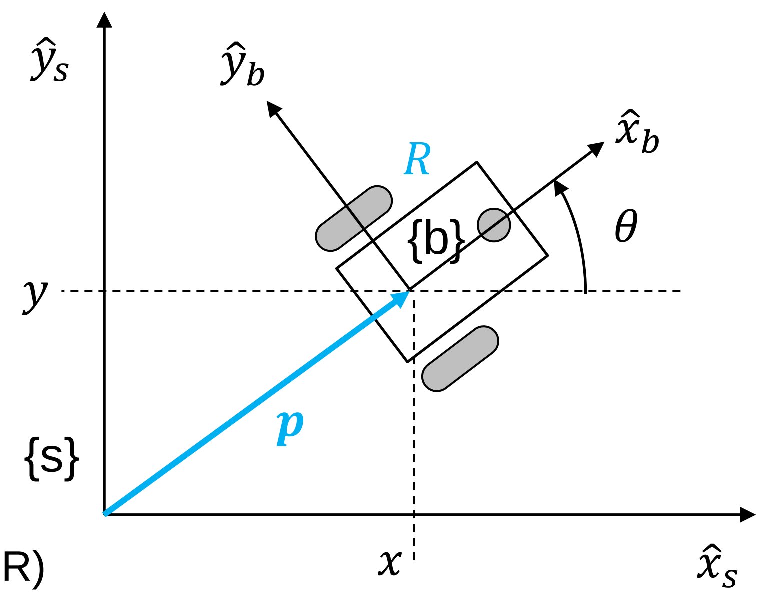
Representing pose: position, orientation

- With the representation of position and orientation we have developed so far, we can put them together to specify the **configuration** of a rigid body.
- In robotics, a robot configuration is also called a **pose** usually denoted by $\boldsymbol{\xi}$.

$$Pose\ \xi \quad = R, p$$



$$R = [\hat{n} \quad \hat{s} \quad \hat{a}]$$



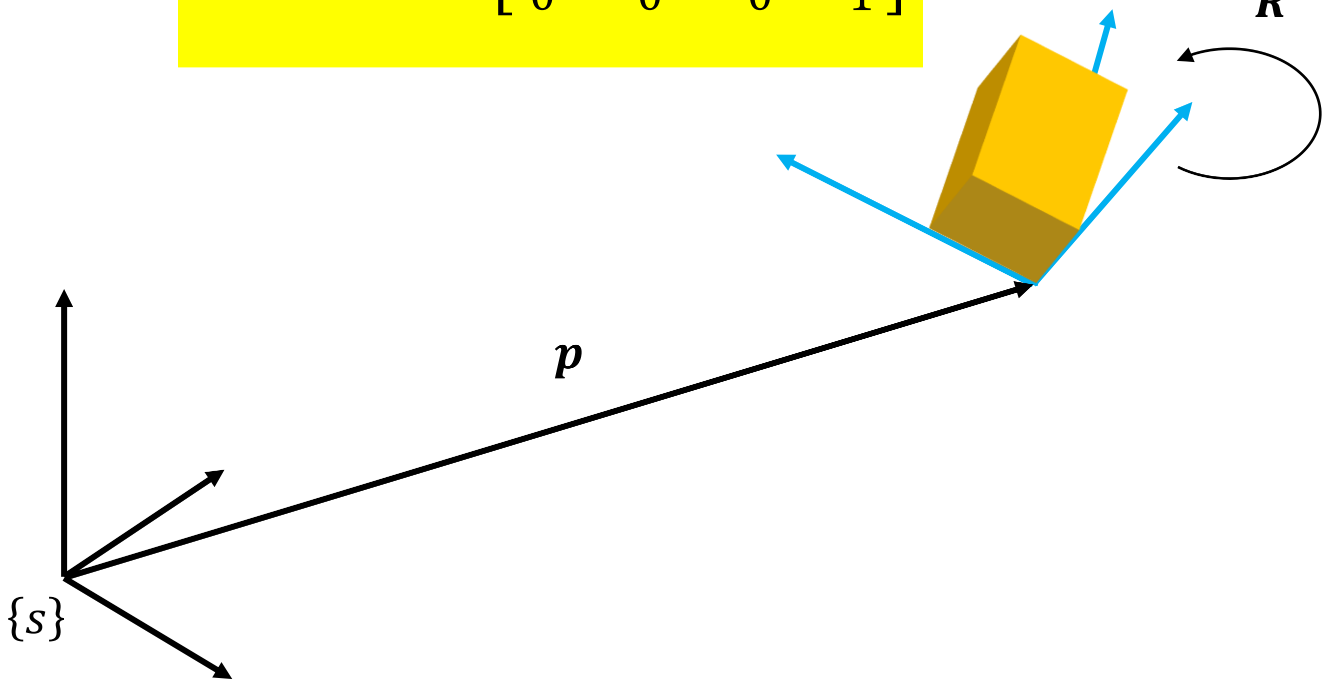
(2-axis representation: $[\hat{s} \quad \hat{a}]$. $\hat{n}$ is assumed with RHR)

Source: https://grabcad.com/library/grabcad4m-robot-arm-1

Homogeneous transformation matrix: pose

It is convenient to put both **orientation** and **position** in *one matrix*. A **homogeneous transformation** matrix puts together rotation ***R*** and position vector ***p*** in one 4 by 4 matrix:

$$\xi = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{p} \\ \underline{0} & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{21} & r_{31} & p_x \\ r_{12} & r_{22} & r_{32} & p_y \\ r_{13} & r_{23} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

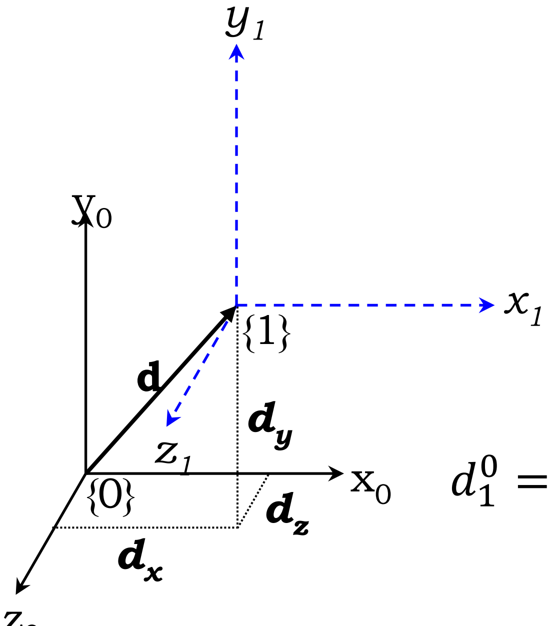


Rigid body motion (aka displacement, or transformation)

Describing motion: translation and rotation

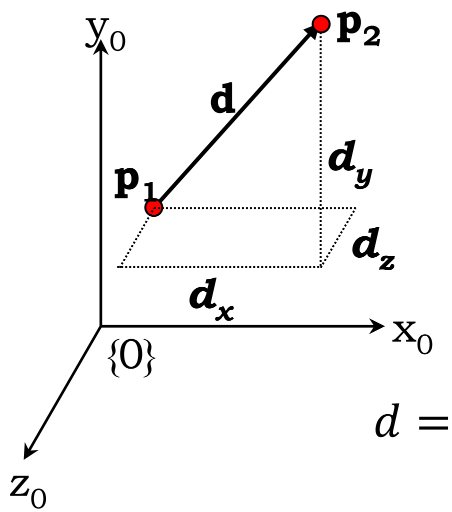
Describing translation motion

- We have learned to use **vector** to represent **position**.
- In addition, free **vector** can be used to represent change in position, i.e. **translation** motion or transformation.
- **Translation** transformation is achieved by **vector addition**.
- Only add vectors in the same (or parallel) reference frame.



$$d_1^0 = \begin{bmatrix} d_x^0 \\ d_y^0 \\ d_z^0 \end{bmatrix}$$

Translation from frame {0} to frame {1}

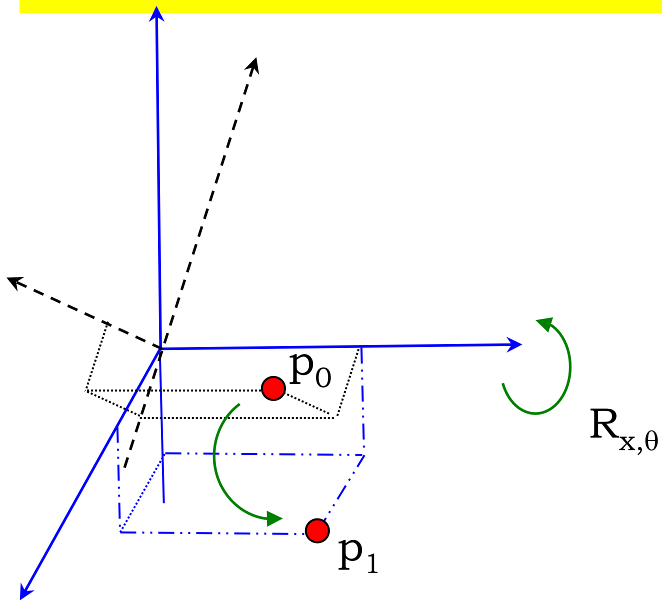


$$d = \begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix} \quad p_2 = p_1 + d$$

Translation within same coordinate frame, from p_1 to p_2

Describing rotation motion

- We have learned that **rotation matrix** can be used to represent **orientation**, **rotation** and **change of reference frame.**
- A **rotation matrix** is used to represent change in orientation, i.e. **rotation** motion or transformation.
- **Rotation** transformation is achieved by **matrix multiplication**.

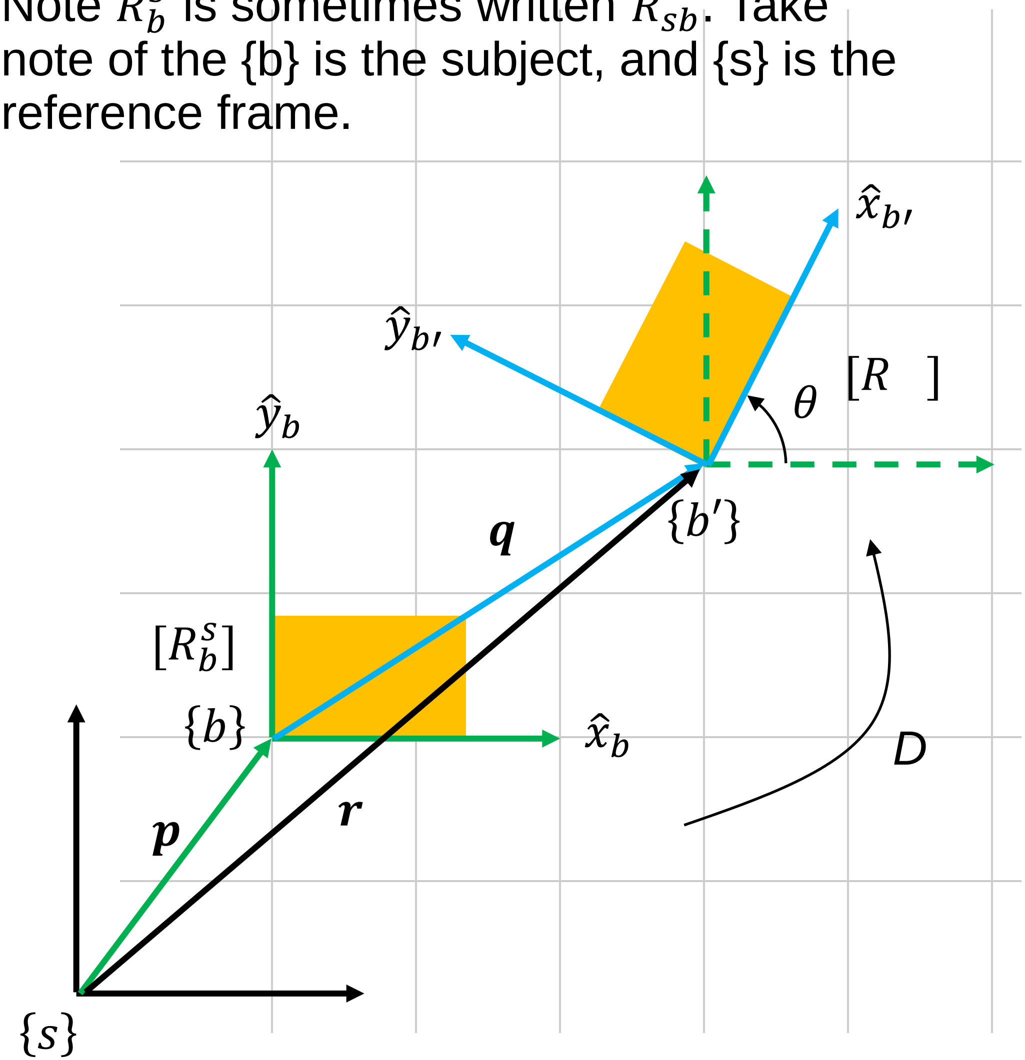


A point **p**, starting at position $\mathbf{p_0}$ (represented as a vector) is rotated about x-axis by an angle of θ, and its new position is $\mathbf{p_1}$

$$p_1 = R_{x,\theta} p_0$$

Combine translation and rotation

Note R_b^s is sometimes written R_{sb}. Take note of the {b} is the subject, and {s} is the reference frame.



Consider a rigid body being displaced by first a translation of ***q*** followed by a rotation of ***R***. The pose of the rigid body is transformed (displaced) from frame {b} to {b'}.

$$Displacement\ D = \left(R = R_{b'}^{b}, \boldsymbol{q}\right)$$

$$Old\ pose\ \xi_b^s = (R_b^s, \boldsymbol{p})$$
$$New\ pose\ \xi_{b'}^s = (R_{b'}^s, \boldsymbol{r})$$
$$R_{b'}^s = R_b^s R$$
$$r = R_b^s q + p$$

Note that the reference frame for both transformations ***q*** and ***R*** is **{b}**. *Rotation* is achieved through (post) *multiplication* while *translation* (reference frame changed) by *addition*.

Homogeneous transformation matrix

It is convenient to put both **rotation** and **translation** in *one matrix*. Transformation will then be achieved through matrix *multiplication* only. A **homogeneous transformation** matrix puts together rotation $\boldsymbol{R} \in \mathbb{R}^{3x3}$ and translation $\boldsymbol{t} \in \mathbb{R}^3$ in one 4 by 4 matrix:

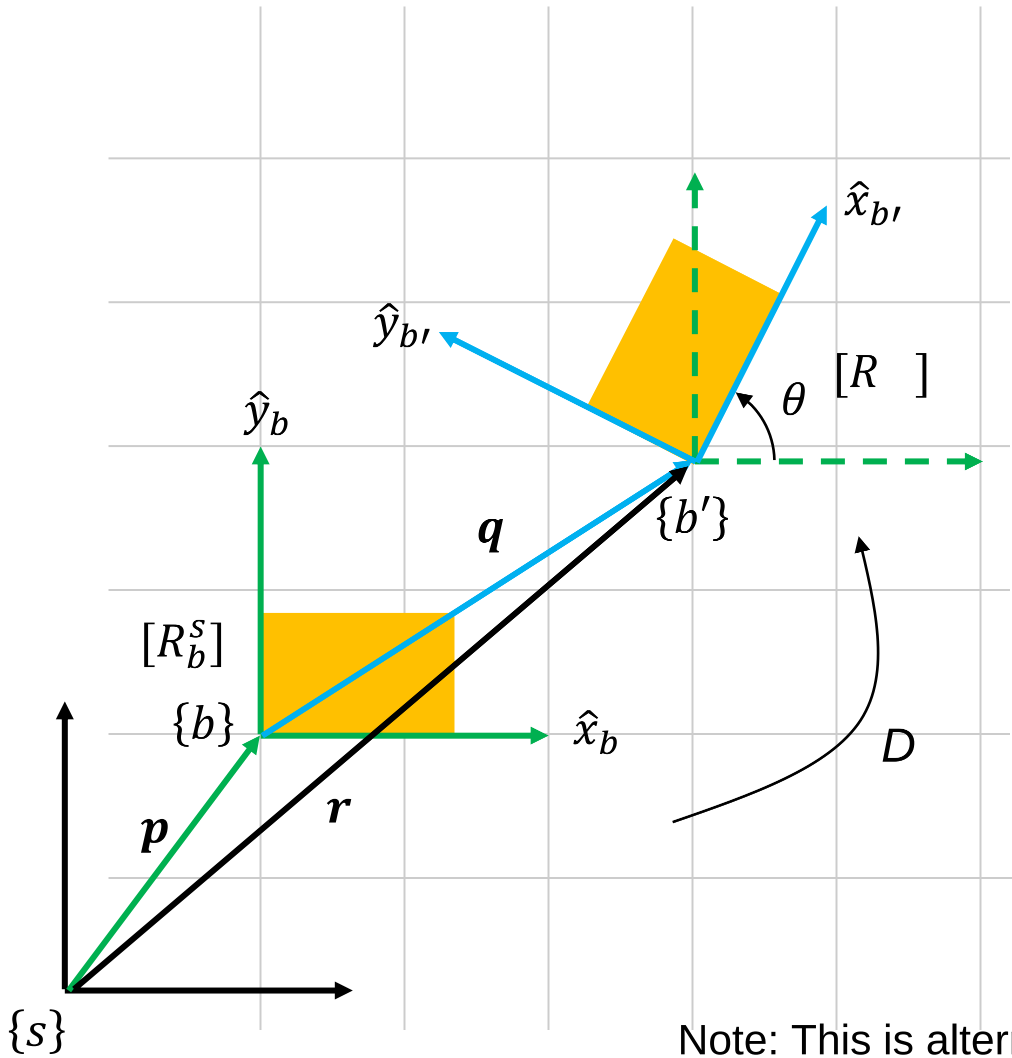
$$H = \begin{bmatrix} R & t \\ \underline{0} & 1 \end{bmatrix}$$

Displacement (transformation) of a vector or frame can be done by multiplication solely.

Homogeneous transformation matrices belongs to the special Euclidean group $SE(3)$. A useful property of this matrix is:

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

Homogeneous transformation: displacement



Consider a rigid body being displaced by first a translation of ***p*** followed by a rotation of R. The pose of the rigid body is transformed (displaced) from frame {b} to {b'}.

$$Displacement\ D = \begin{bmatrix} R = R^b_{b'} & \boldsymbol{q} \\ \underline{0} & 1 \end{bmatrix}$$

$$Old\ pose\ \xi^s_b = \begin{bmatrix} R^s_b & \boldsymbol{p} \\ \underline{0} & 1 \end{bmatrix}$$

$$New\ pose\ \xi^s_{b'} = \xi^s_b D = \begin{bmatrix} R^s_{b'} & \boldsymbol{r} \\ \underline{0} & 1 \end{bmatrix}$$

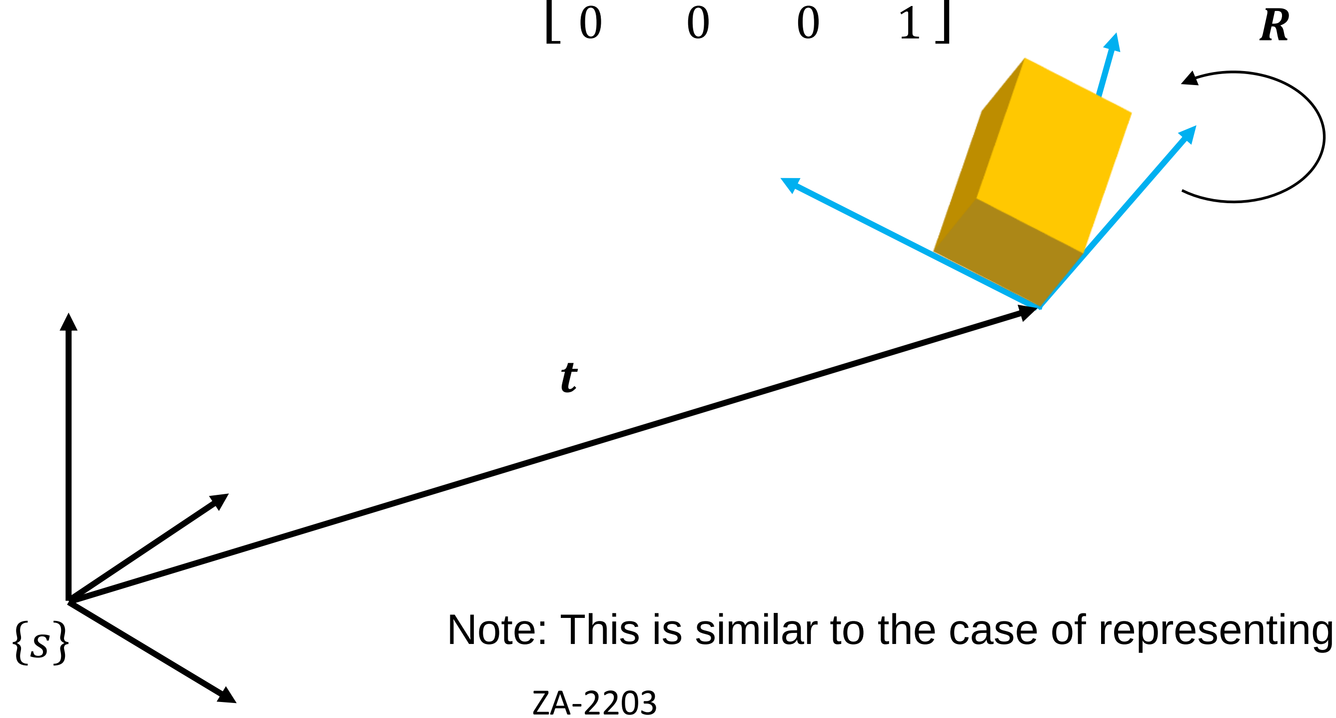
Careful with pre- or post-multiplication

Note: This is alternative representation to that in Slide 67.

Homogeneous transformation: 3D

Similar to 2D case, we can pack the **rotational** and **translational transformation (displacement)** in *one matrix*. A **homogeneous transformation** matrix puts together rotation ***R*** and translation vector ***t*** in one 4 by 4 matrix:

$$\boldsymbol{D} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \underline{0} & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{21} & r_{31} & p_x \\ r_{12} & r_{22} & r_{32} & p_y \\ r_{13} & r_{23} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



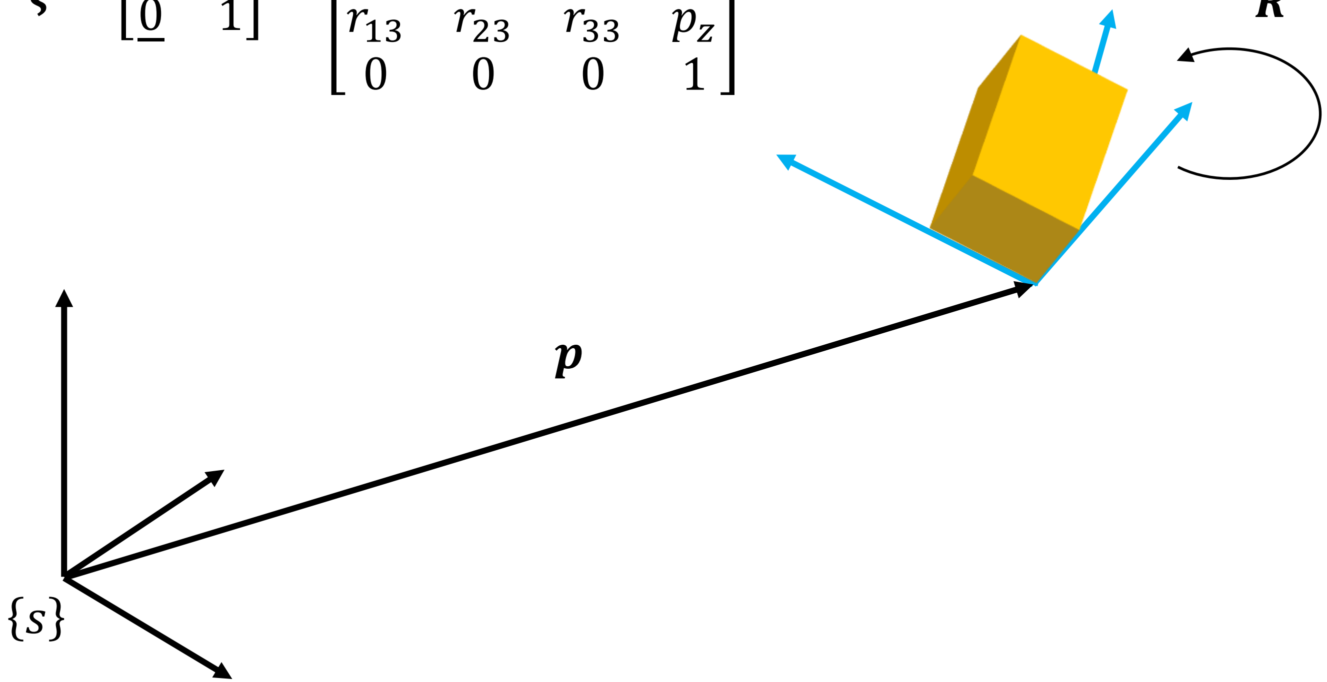
Note: This is similar to the case of representing pose.

Three uses of transformation matrix

- Similar to rotation matrix, homogeneous transformation matrix can be used in three ways
  - Represent a configuration (pose)
  - Displace a vector or frame
  - Change the reference frame of a vector or frame

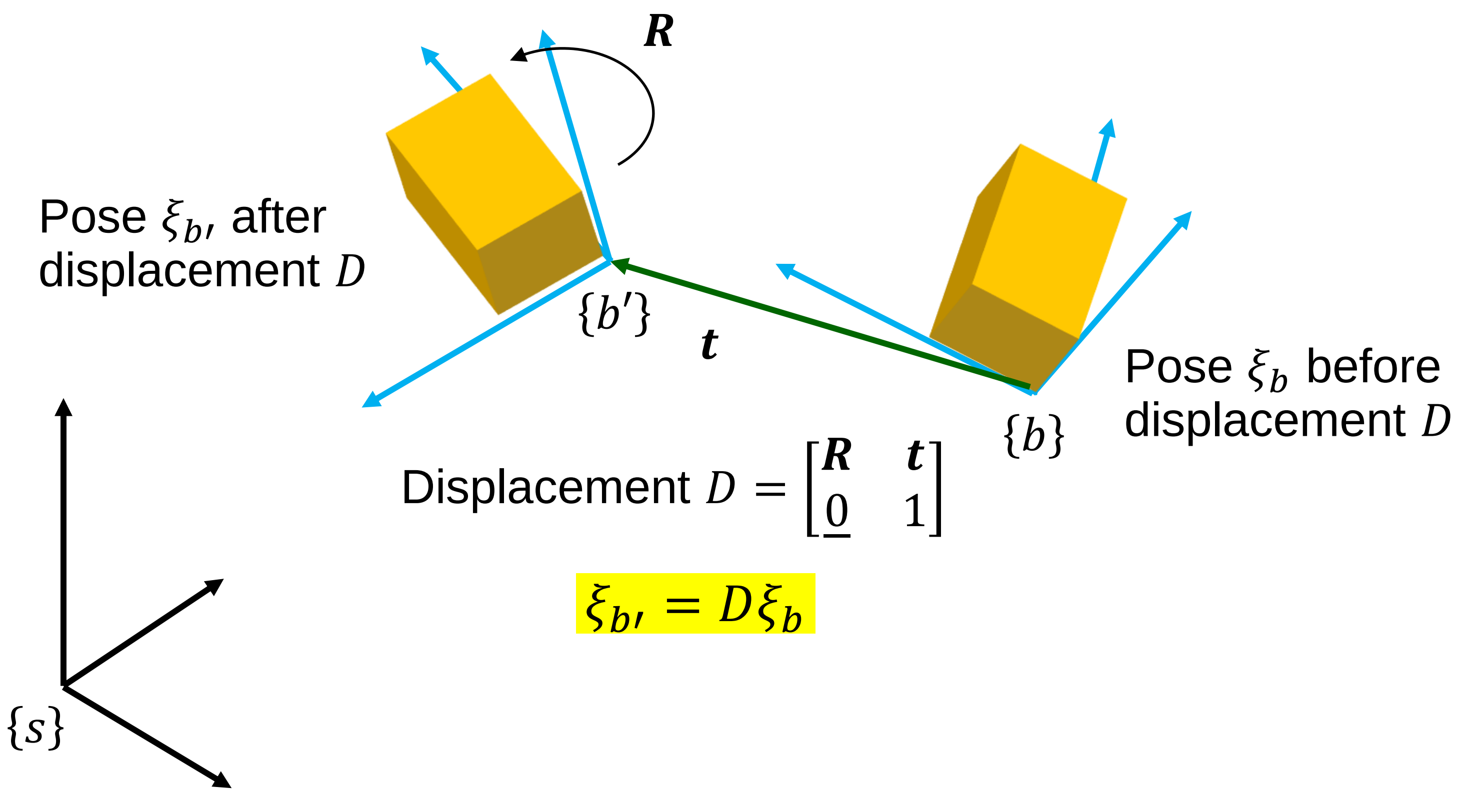
(repeat) T Matrix: represent a pose

$$\boldsymbol{\xi} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{p} \\ \underline{0} & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{21} & r_{31} & p_x \\ r_{12} & r_{22} & r_{32} & p_y \\ r_{13} & r_{23} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



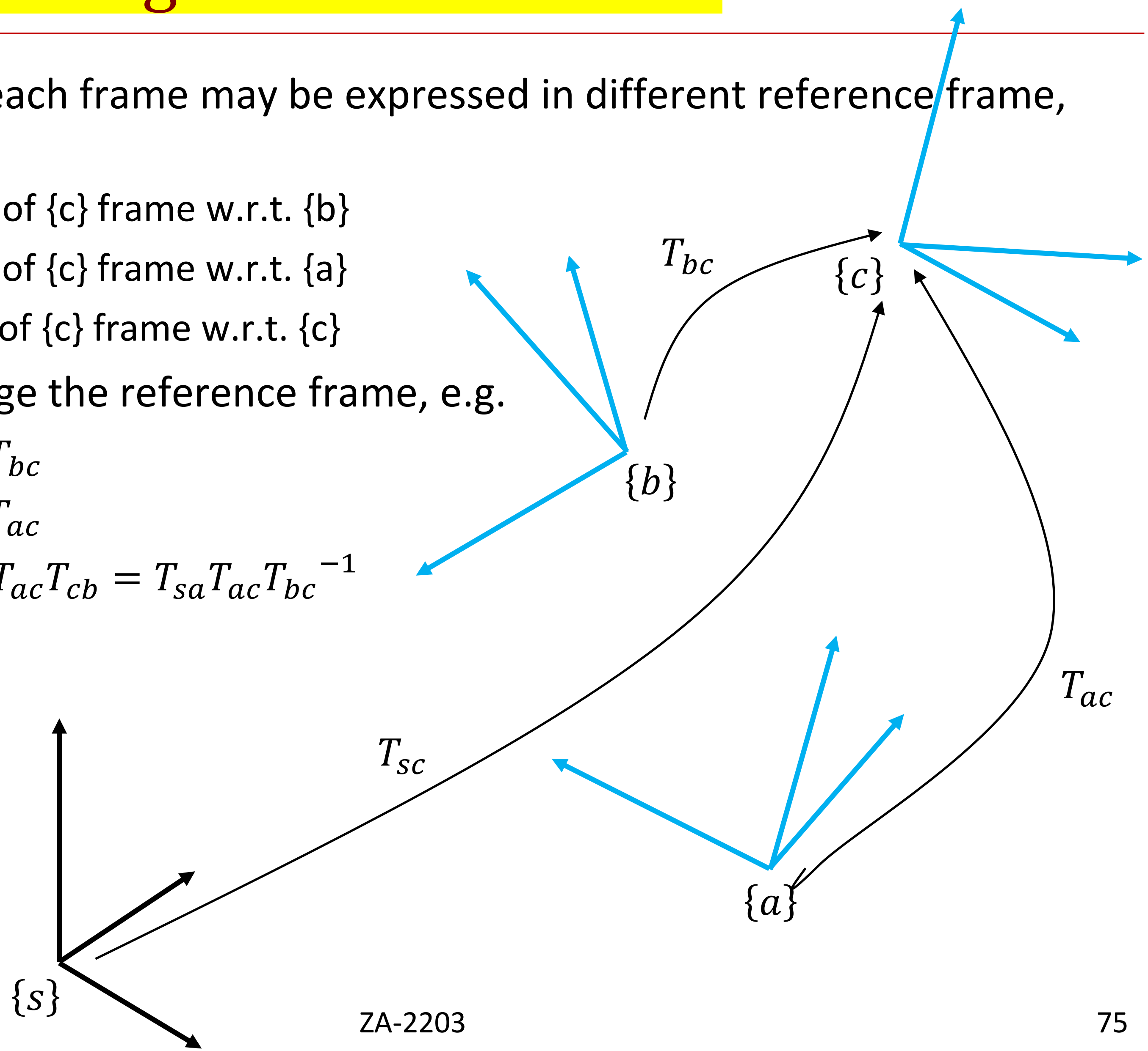
(repeat) T Matrix: displace a vector or frame

- When displacing a vector with a T matrix, the vector needs to be represented in transformation coordinates, i.e. add one at the bottom to form an $\mathbb{R}^4$ vector.
- Here we show displacing a frame attached to a rigid body.



T Matrix: change reference frame

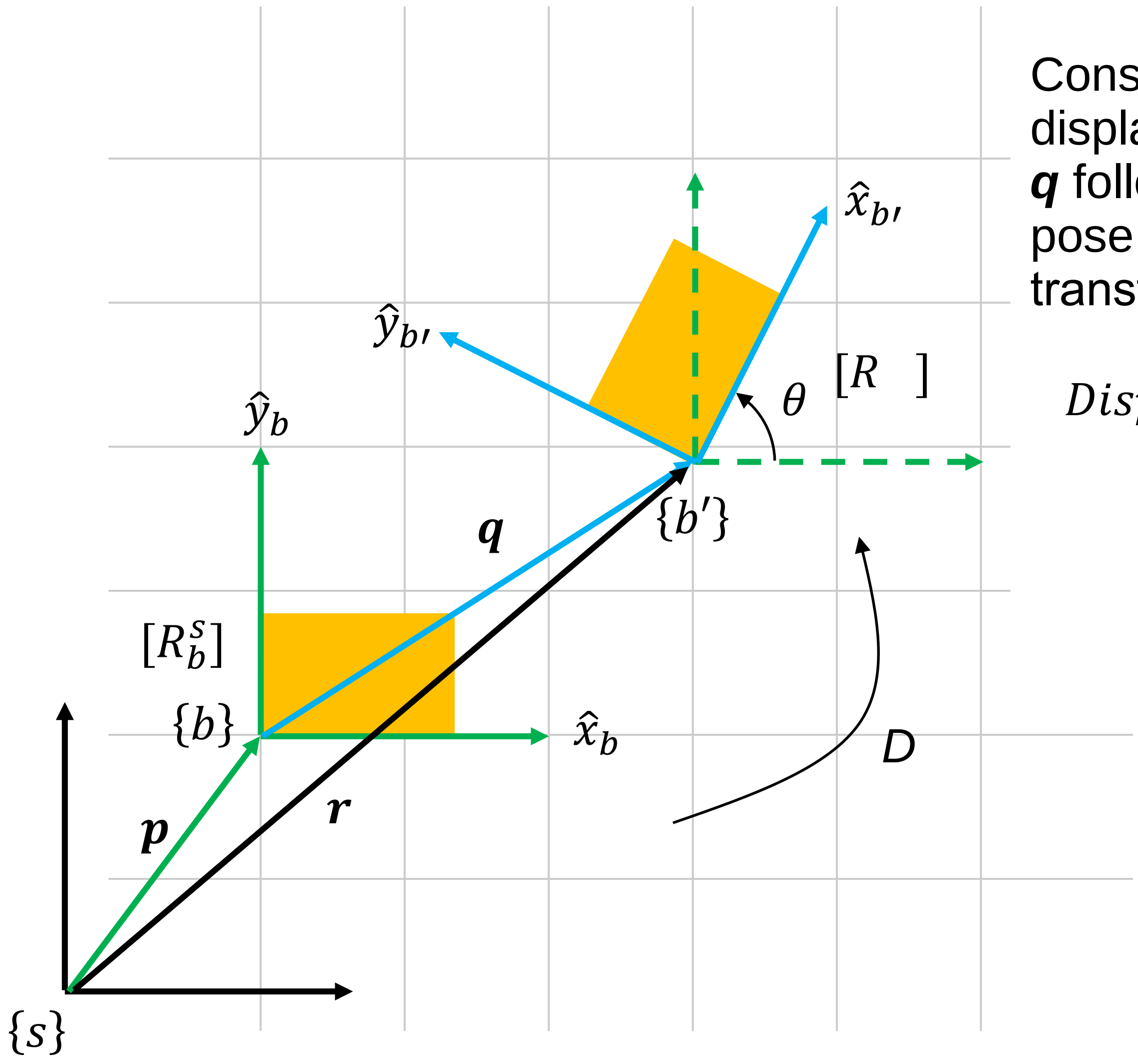
- The pose of each frame may be expressed in different reference frame, e.g.
  - T_{bc}: pose of {c} frame w.r.t. {b}
  - T_{ac}: pose of {c} frame w.r.t. {a}
  - T_{sc}: pose of {c} frame w.r.t. {c}
- We can change the reference frame, e.g.
  - $T_{sc} = T_{sb}T_{bc}$
  - $T_{sc} = T_{sa}T_{ac}$
  - $T_{sb} = T_{sa}T_{ac}T_{cb} = T_{sa}T_{ac}{T_{bc}}^{-1}$



Screw theory

Describing motion and velocities

Screw motion

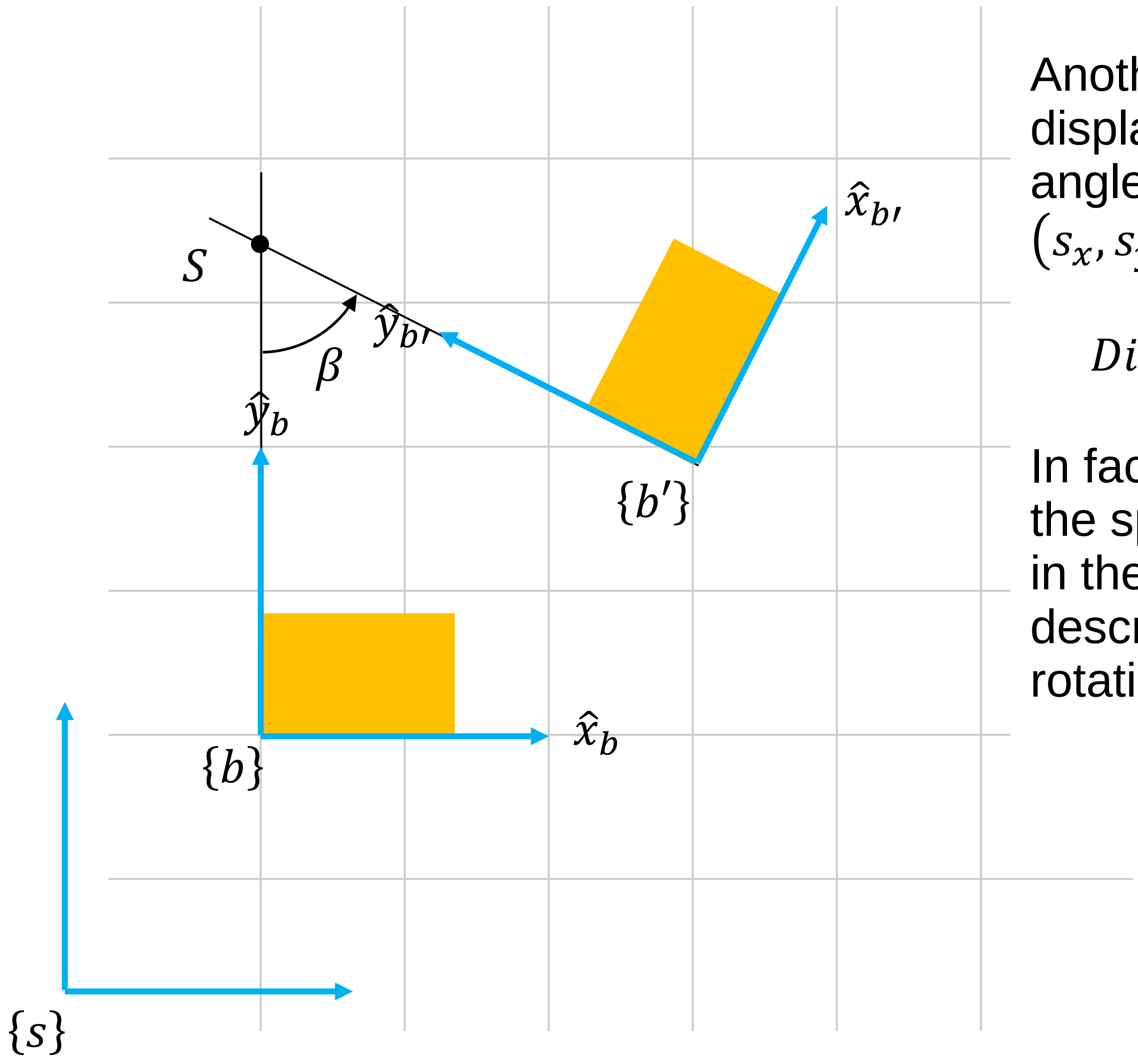


Consider a rigid body being displaced by first a translation of ***q*** followed by a rotation of ***R***. The pose of the rigid body is transformed from frame {b} to {b'}.

$$Displacement\ D = \left(R = R_{b'}^{b}, \boldsymbol{q}\right)$$

$$Pose\ \xi_{b'}^{s} = (R_{b'}^{s}, \boldsymbol{r})$$
$$R_{b'}^{s} = R_b^s R$$
$$r = R_b^s q + p$$

Screw motion

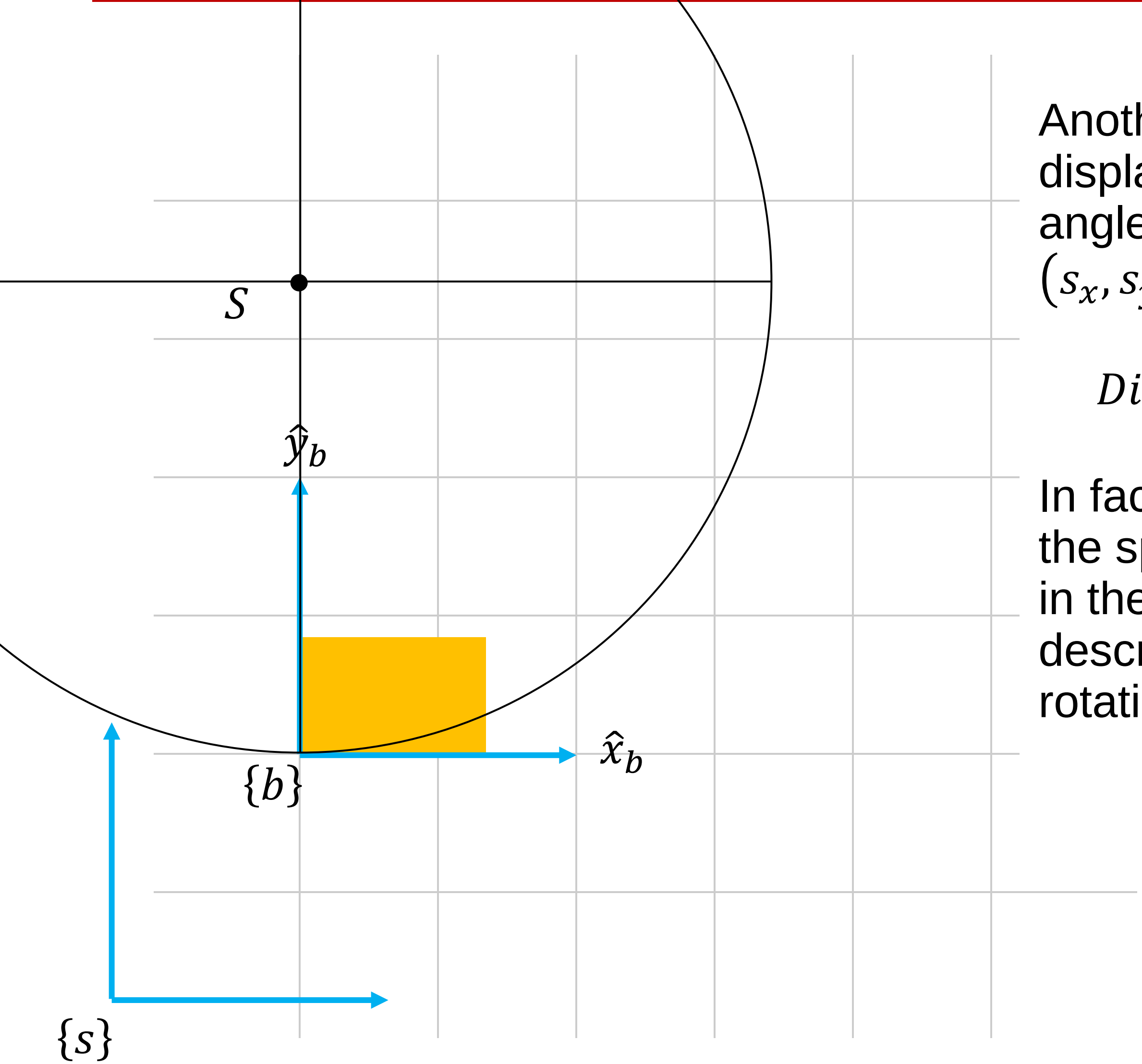


Another way of thinking of the displacement is a rotation of an angle β about a fixed point $S = (s_x, s_y)$ in the space $\{s\}$.

$$Displacement\ D = (\beta, s_x, s_y)$$

In fact, for any displacement in the space, there is such a point S in the space that can be used to describe such displacement by a rotation around the point S.

Screw motion

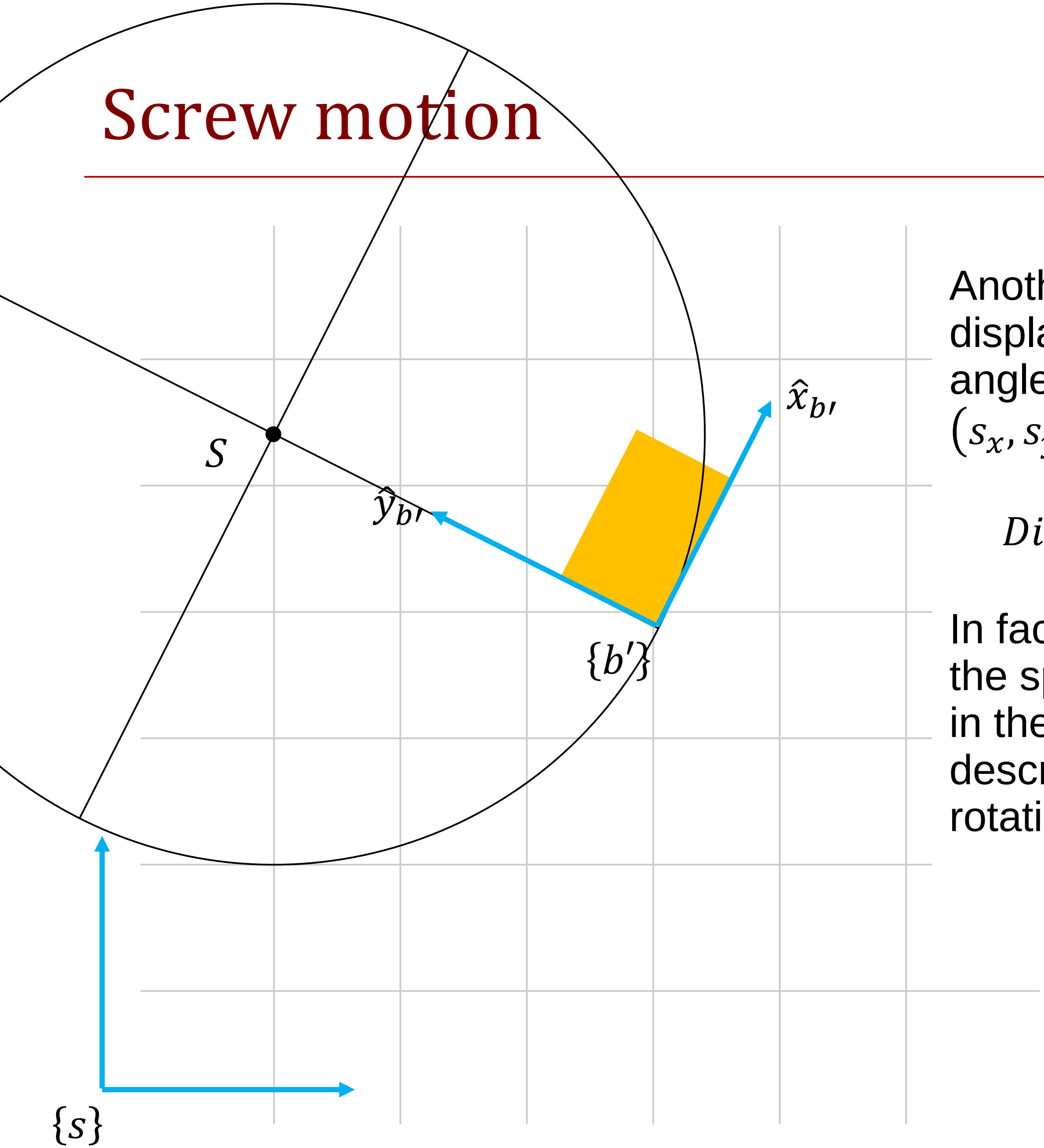


Another way of thinking of the displacement is a rotation of an angle β about a fixed point $S = (s_x, s_y)$ in the space $\{s\}$.

$$Displacement\ D = (\beta, s_x, s_y)$$

In fact, for any displacement in the space, there is such a point S in the space that can be used to describe such displacement by a rotation around the point S.

Screw motion

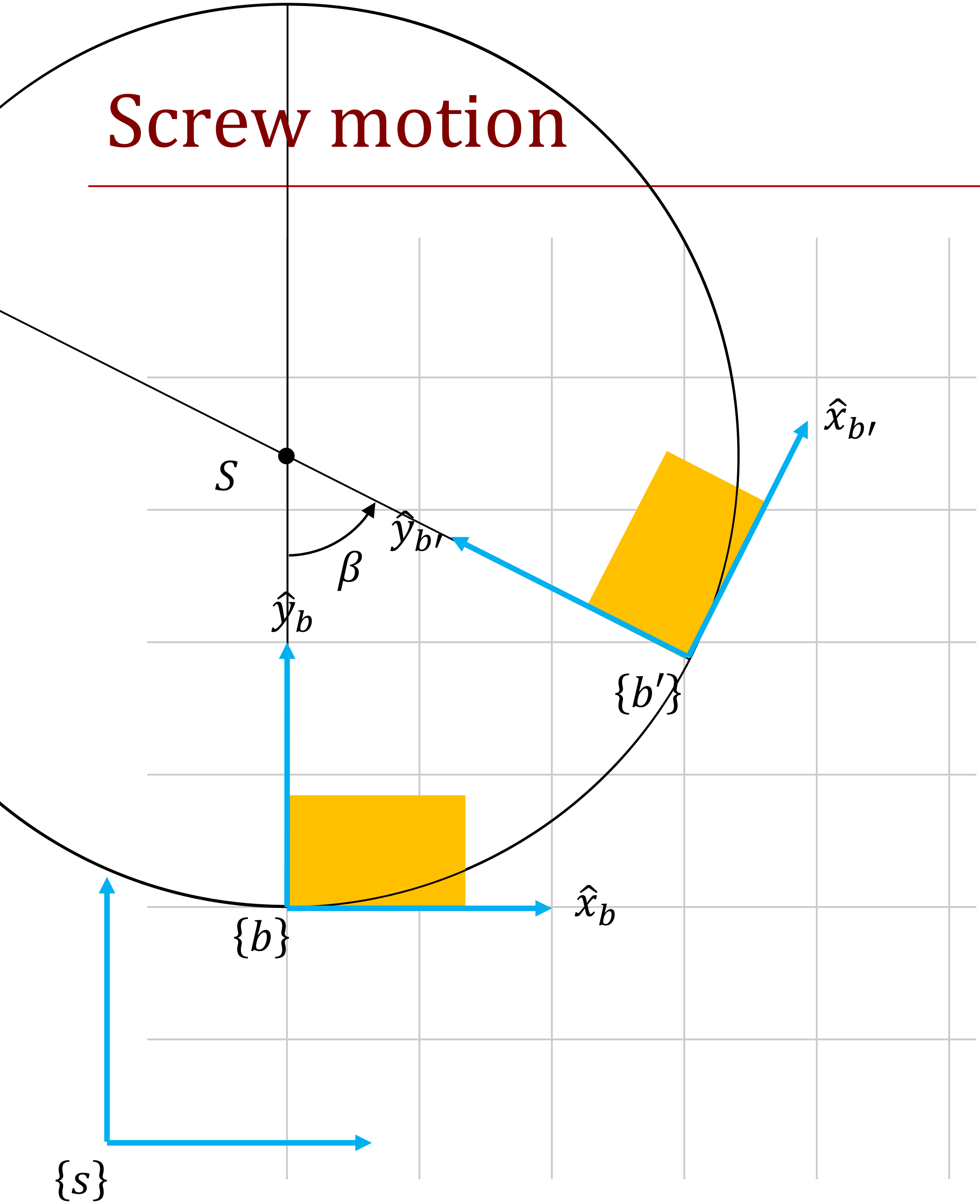


Another way of thinking of the displacement is a rotation of an angle β about a fixed point $S = (s_x, s_y)$ in the space $\{s\}$.

$$Displacement\ D = (\beta, s_x, s_y)$$

In fact, for any displacement in the space, there is such a point S in the space that can be used to describe such displacement by a rotation around the point S.

Screw motion



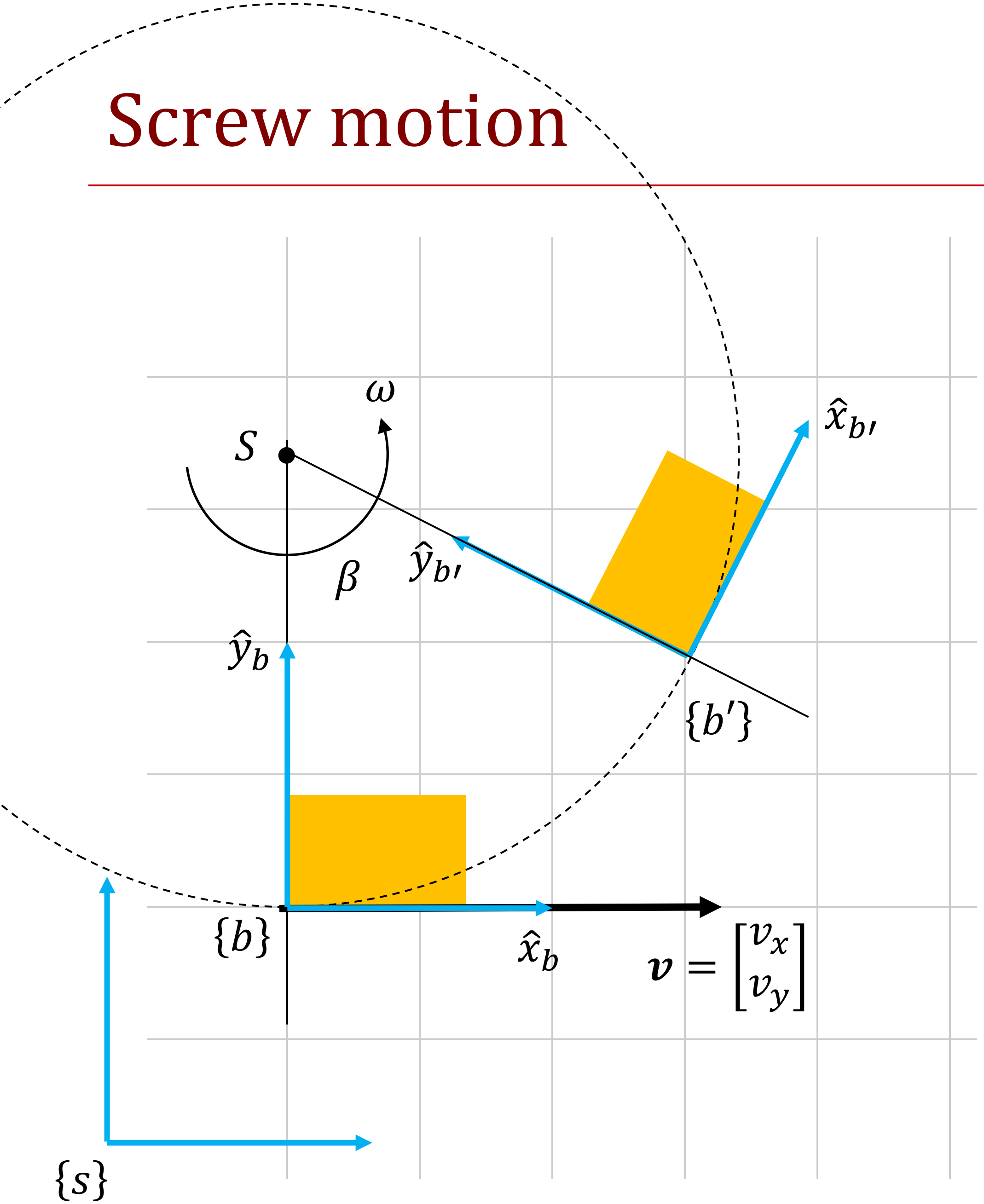
Another way of thinking of the displacement is a rotation of an angle β about a fixed point $S = (s_x, s_y)$ in the space $\{s\}$.

$$Displacement\ D = (\beta, s_x, s_y)$$

In fact, for any displacement in the space, there is such a point S in the space that can be used to describe such displacement by a rotation around the point S.

This is a planar example of a **screw motion**. The screw axis is out of the slide in the fixed frame $\{s\}$.

Screw motion



Another way of viewing screw motion is to think of it as a combination of angular ω and linear $\boldsymbol{v}$ velocities, called **screw axis**.

$$Screw\ axis\ \boldsymbol{S} = (\omega, v_x, v_y)$$

The angular velocity can be fixed at $\omega = 1\ rad/s$ and $\boldsymbol{v}$ will be determined such that the **displacement** is correctly defined at an angle $\theta = \beta$.

$$Displacement\ D = \boldsymbol{S}\theta$$

(exponential coordinates)

Similar concept is applicable in 3D space.

Angular velocity

- Base on the concepts of screw motion and axis-angle representation, **angular velocity** $\boldsymbol{\omega}$ of a rotating body (frame {b}) can be represented by determining the suitable rotation axis $\widehat{\boldsymbol{k}}$ through the origin of {b} and the rotation speed $\dot{\theta}$ about this axis. All vectors are with reference to space frame {s}.

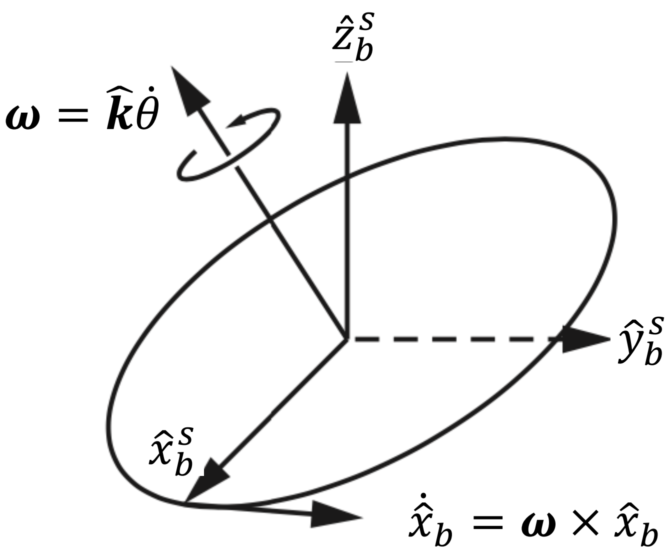
$$\boldsymbol{\omega} = \widehat{\boldsymbol{k}}\dot{\theta}$$

Given the orientation of the body

$$R = [\hat{x}_b \quad \hat{y}_b \quad \hat{z}_b]$$

Then $\dot{\hat{x}}_b = \boldsymbol{\omega} \times \hat{x}_b$, $\dot{\hat{y}}_b = \boldsymbol{\omega} \times \hat{y}_b$, $\dot{\hat{\boldsymbol{z}}}_{\boldsymbol{b}} = \boldsymbol{\omega} \times \hat{z}_b$

$$\dot{R} = [\omega \times \hat{x}_b \quad \omega \times \hat{x}_b \quad \omega \times \hat{x}_b] = \omega \times R$$



Source: Modern Robotics

We can replace the cross product with skew-symmetric matrix multiplication.

Fixed frame (pre-multiply): $\dot{R} = S(\omega_s)R$

Body frame (post-multiply): $\dot{R} = RS(\omega_b)$

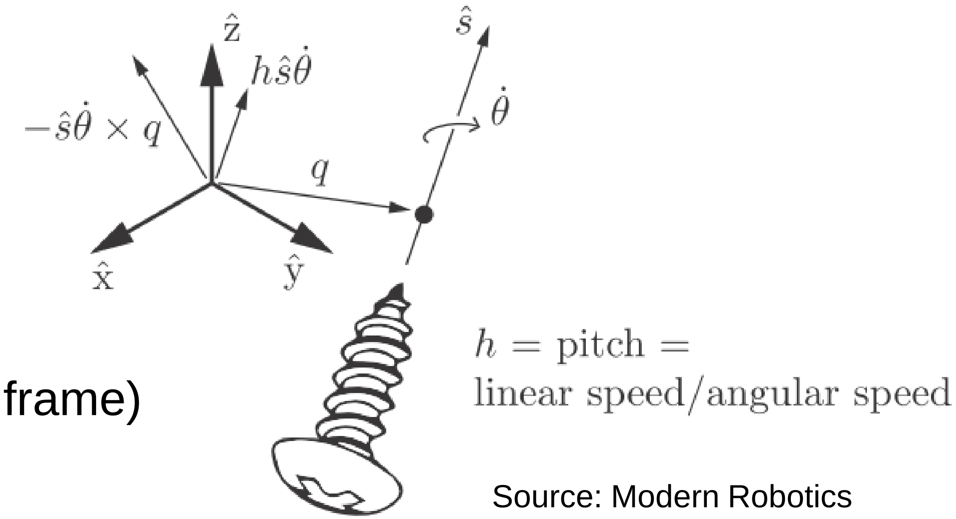
Note $S(\omega_s)$ is the skew-symmetric matrix of vectorω_s.

Twist: linear and angular velocities

Any **rigid-body (spatial) velocity** can be represented as a **twist** comprising of a **screw axis** $\boldsymbol{S} = \{q, \hat{s}, h\}$ (a direction $\hat{s} \in S^2$, a point $q \in \mathbb{R}^3$ on the screw, and the **pitch** (linear speed/angular speed) of the screw h), plus the **speed** along the screw $\dot{\theta}$.

Twist is given by

$$\mathcal{V} = \mathcal{S}\dot{\theta} = \begin{bmatrix}\omega \\ v\end{bmatrix} = \begin{bmatrix}\hat{s}\dot{\theta} \\ -\hat{s}\dot{\theta} \times q + h\hat{s}\dot{\theta}\end{bmatrix}$$



Source: Modern Robotics

Matrix representation:

Body twist (spatial velocity in the body frame)

$$T_{sb}^{-1}\dot{T}_{sb} = [\mathcal{V}_b] = \begin{bmatrix}[\omega_b] & v_b \\ 0 & 0\end{bmatrix}$$

Spatial twist (spatial velocity in the space frame)

$$\dot{T}_{sb}T_{sb}^{-1} = [\mathcal{V}_s] = \begin{bmatrix}[\omega_s] & v_s \\ 0 & 0\end{bmatrix}$$

(this is called Matrix Logarithm)

Note $[\omega_b]$ is the skew-symmetric matrix of the vectorω_b.

If h is infinite, $\dot{\theta}$ is the linear speed. Otherwise, it is the angular speed.

Twist: linear and angular velocities

Any **rigid-body (spatial) velocity** can be represented as a **twist** comprising of a **screw axis** $\boldsymbol{S} = \{q, \hat{s}, h\}$ (a direction $\hat{s} \in S^2$, a point $q \in \mathbb{R}^3$ on the screw, and the **pitch** (linear speed/angular speed) of the screw h), plus the **speed** along the screw $\dot{\theta}$.

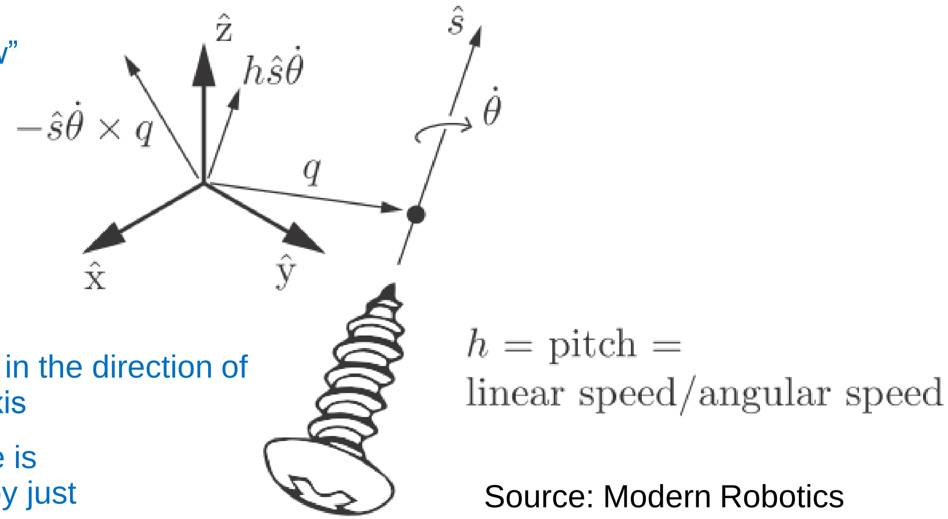
Twist is given by

angular velocity about the "screw" axis $\hat{s}$

$$\mathcal{V} = \mathcal{S}\dot{\theta} = \begin{bmatrix} \omega \\ v \end{bmatrix} = \begin{bmatrix} \hat{s}\dot{\theta} \\ -\hat{s}\dot{\theta} \times q + h\hat{s}\dot{\theta} \end{bmatrix}$$

linear velocity on the plane orthogonal to the "screw" axis

linear velocity in the direction of the "screw" axis



Source: Modern Robotics

Note screw axis $\mathcal{S}$ is not shown the diagram. Imagine there is one that describes both spatial angular and linear motion by just making a rotation around it ($\dot{\theta}$) without the pitch.

The "screw" axis $\hat{s}$ shown in the diagram is the one in the direction of angular motion ω, which in turn causes 2D linear motion on the plane orthogonal to the "screw" as well as a 1D linear motion in the direction of the "screw" resulting in 3D linear motion.

If h is infinite, $\dot{\theta}$ is the linear speed. Otherwise, it is the angular speed.

$\hat{s}$ can be conveniently determined (in contrast to the actual screw $\mathcal{S}$), e.g. as the joint axis, giving us a way to determine the angular velocity ω and spatial linear velocity v. We can then determine the screw axis $\mathcal{S} = \frac{\mathcal{V}}{\dot{\theta}}$.

Summary (1/4)

- We attach **coordinate frame** in the space and bodies in order to give "values" to the configurations (points) of the bodies.
- We use **right hand rule** for the coordinate system.
- The **position** of point can be represented by cartesian coordinates or **vector**.
- We can perform operations (e.g. addition) on vectors but not points.
- We attach coordinate frame to a rigid body and uses the **origin** to represent its **position**.
- The orientation of the coordinate frame fixed on the body is used to represent the orientation of the body.

Summary (2/4)

- **Rotation matrix** represents the **orientation** by specifying the position of the axes of the body coordinate frame in the space.
- Rotation matrix can be used to:
  - Represent **orientation** of a vector or frame
  - **Rotate** (as operator) a vector or frame
  - Change the **reference frame** of a vector or frame
- Rotation matrix does not suffer from the problems of singularity, however is complicated to interpolate the parameters due to having to maintain the constraints.
- Three **angles representations** are easy to interpolate, however suffer from singularity problem.

Summary (3/4)

- Three angles representations include:
  - **Euler angles** (relative, current frame)
  - Fixed **yaw-pitch roll** (fix frame)
  - **Axis-angle**
- Angles representations can be used to parameterize rotation matrices with its forward problem formulation.
- The inverse problem formulation determine the angles from the parameters of a rotation matrix.
- **Quaternion** represents axis-angle in four-dimension space. It avoids singularity.
- Axis-angle representation is also called **exponential coordinates**. This representation is useful to describe velocities.

Summary (4/4)

- Describing the configuration or **pose** put together position (vector) and orientation (rotation matrix).
- Vector is also used to represent **translation motion**.
- Rotation matrix is also used to represent **rotation motion**.
- **Homogeneous transformation matrix** put together the translation and rotation motion in one matrix. This allow a single operation (multiplication) when computing new pose resulting from the motion.
- **Screw theory** is used to describe velocities.
- Linear and rotational (angular) velocities are represented by the rotational and linear motion of the screw.
- **Twist** is the representation of linear and angular velocities put in one.

Reading List

- Read Chapter 3 of Modern Robotics

To Do List

- Watch Chapter 3 videos of Modern Robotics on Coursera, or on YouTube https://www.youtube.com/playlist?list=PLggLP4f-rq02vX0OQQ5vrCxbJrzamYDfx