Chapter 1. Representing Position and Orientation

Phan Tran Dang Khoa
University of Science and Technology – The University of Danang

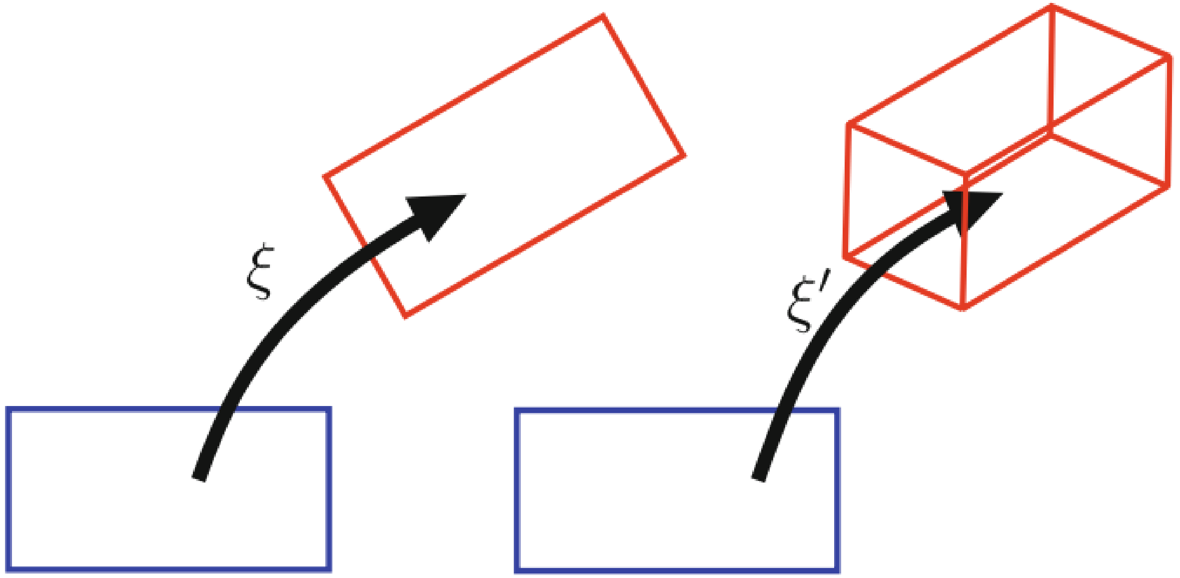
Contents

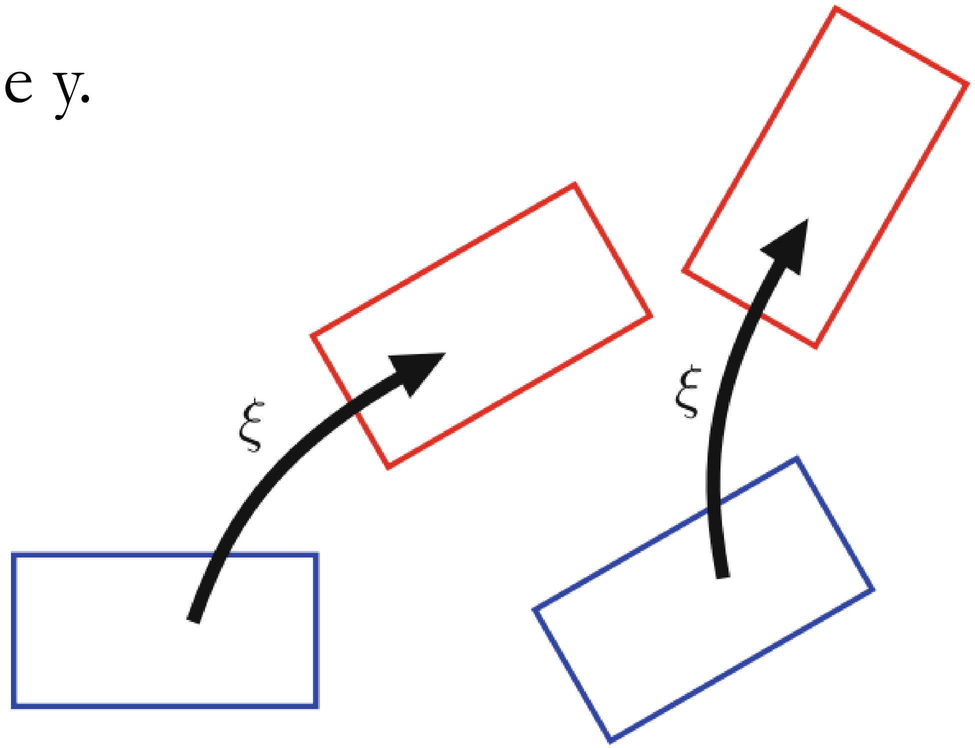
1.1 Foundations

- Important general concepts that underpin our ability to describe where objects are in the world, both graphically and algebraically.

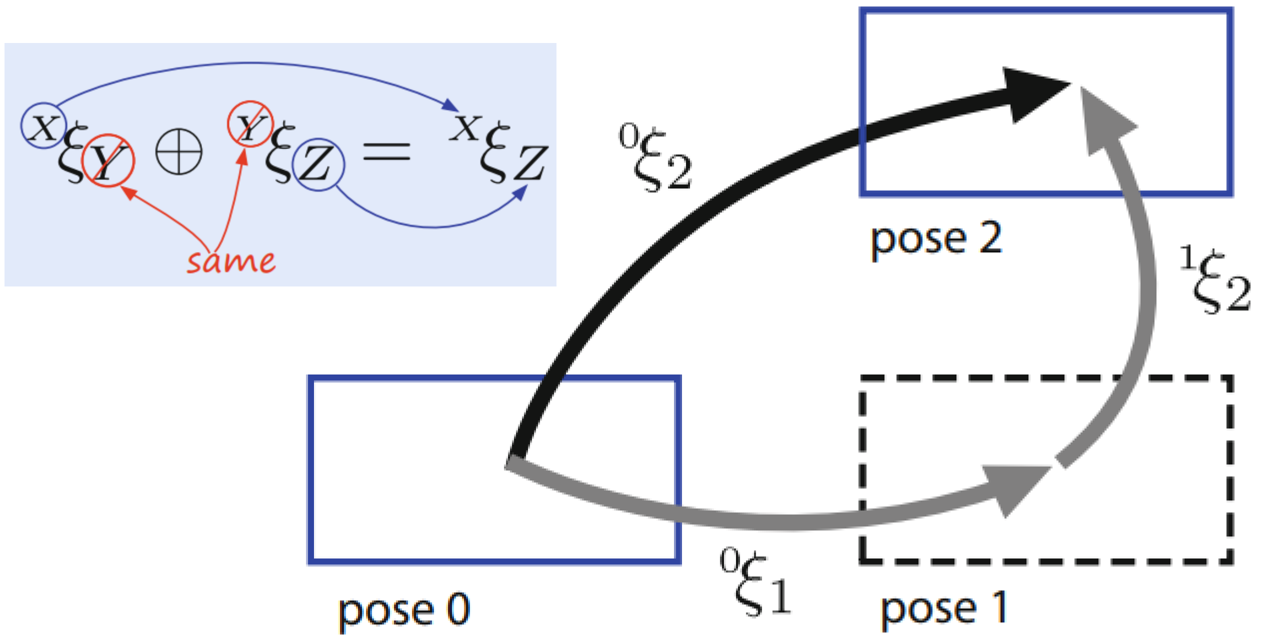
1.1.1 Relative Pose

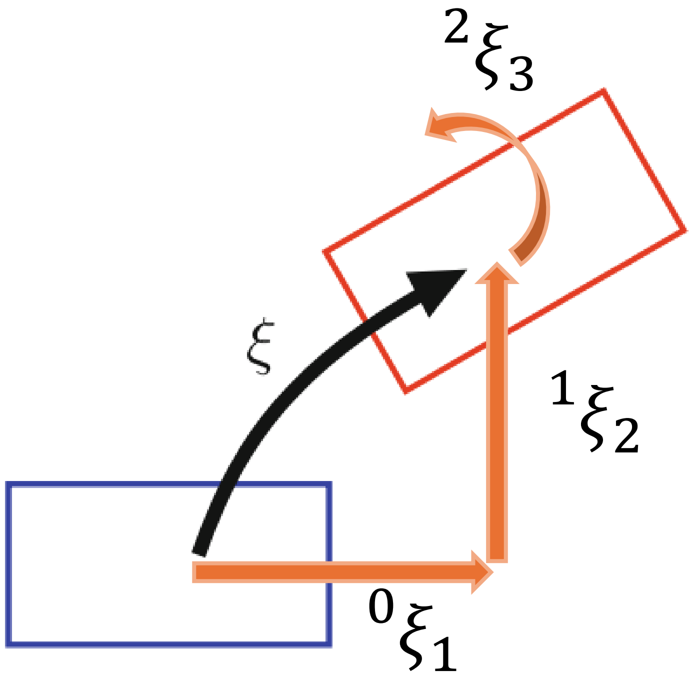
- Position + Orientation = Pose.
- The motion is defined with respect to an initial pose.
- We can only describe the pose of an object with respect to some other pose (reference pose).
- ${}^{x}\boldsymbol{\xi}_{y}$ means the motion from pose x to pose y.





- Motion composition or compounding: ${}^{0}\boldsymbol{\xi}_2 = {}^{0}\boldsymbol{\xi}_1 \oplus {}^{1}\boldsymbol{\xi}_2$
- The motions are consecutive.
- The order of the motion is important.
- Composition of motions is not commutative.
- Any motion can be decomposed into a number of smaller or simpler motions.



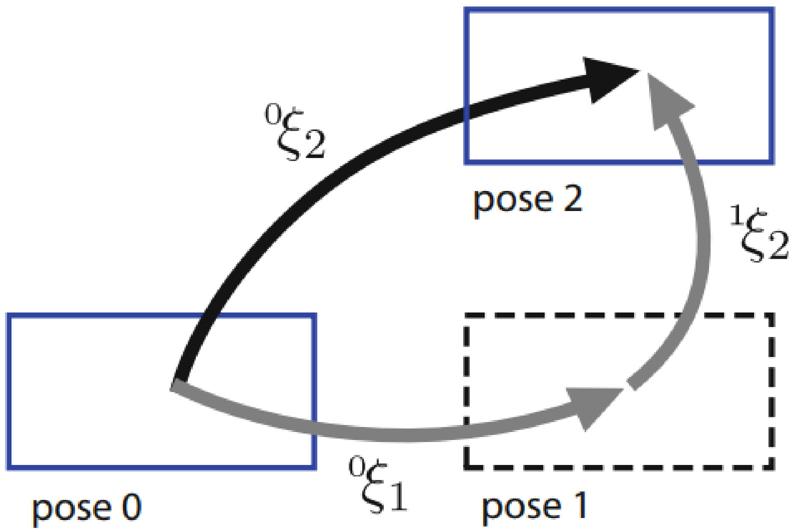


- For any motion there is an inverse or opposite motion.
- For a motion from pose X to pose Y, the inverse motion is from pose Y to pose X, that is

$$ {}^{X}\boldsymbol{\xi}_{Y} \oplus {}^{Y}\boldsymbol{\xi}_{X} = {}^{X}\boldsymbol{\xi}_{X} = \varnothing $$

where $\varnothing$ is the null motion and means no motion.

- Operator $\ominus$ that turns any motion into its inverse: ${}^{Y}\boldsymbol{\xi}_{X} \equiv \ominus\, {}^{X}\boldsymbol{\xi}_{Y}$
- Corollary: ${}^{X}\boldsymbol{\xi}_{Y} \ominus {}^{X}\boldsymbol{\xi}_{Y} = \varnothing$, ${}^{X}\boldsymbol{\xi}_{Y} \oplus \varnothing = {}^{X}\boldsymbol{\xi}_{Y}$, ${}^{X}\boldsymbol{\xi}_{Y} \ominus \varnothing = {}^{X}\boldsymbol{\xi}_{Y}$
- Example:



$$ {}^{0}\boldsymbol{\xi}_{1} = {}^{0}\boldsymbol{\xi}_{2} \oplus {}^{2}\boldsymbol{\xi}_{1} $$
$$ {}^{2}\boldsymbol{\xi}_{1} = \ominus\, {}^{1}\boldsymbol{\xi}_{2} $$
$$ {}^{0}\boldsymbol{\xi}_{1} = {}^{0}\boldsymbol{\xi}_{2} \ominus {}^{1}\boldsymbol{\xi}_{2} $$

- Example:

$$
\begin{aligned}
{}^{0}\xi_2 &= {}^{0}\xi_1 \oplus {}^{1}\xi_2 \\
{}^{0}\xi_2 \underline{\ominus\, {}^{1}\xi_2} &= {}^{0}\xi_1 \oplus {}^{1}\xi_2 \underline{\ominus\, {}^{1}\xi_2} \\
{}^{0}\xi_2 \ominus {}^{1}\xi_2 &= {}^{0}\xi_1 \oplus \cancel{{}^{1}\xi_2 \ominus {}^{1}\xi_2} \\
{}^{0}\xi_2 \ominus {}^{1}\xi_2 &= {}^{0}\xi_1 \oplus \varnothing \\
&= {}^{0}\xi_1 \ .
\end{aligned}
$$

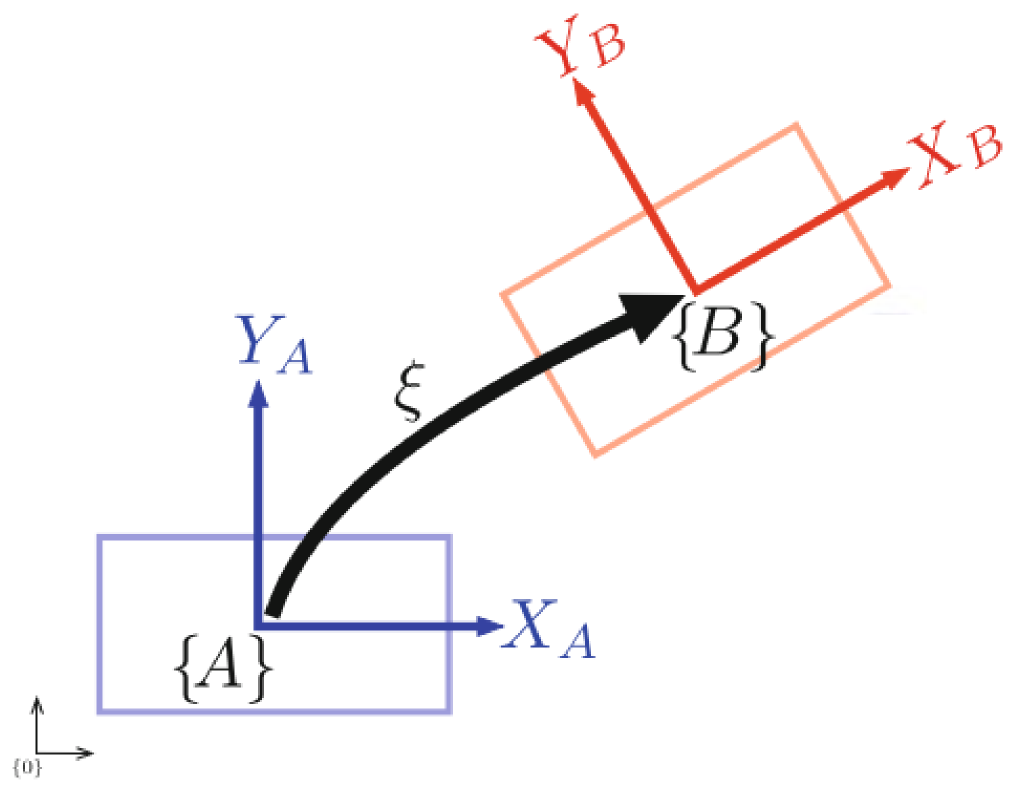
- We could do this in a single step by "taking across to the other side" and "negating" it.

$$\oplus \rightarrow \ominus$$

$${}^{0}\xi_2 = {}^{0}\xi_1 \oplus {}^{1}\xi_2$$

1.1.2 Coordinate Frames

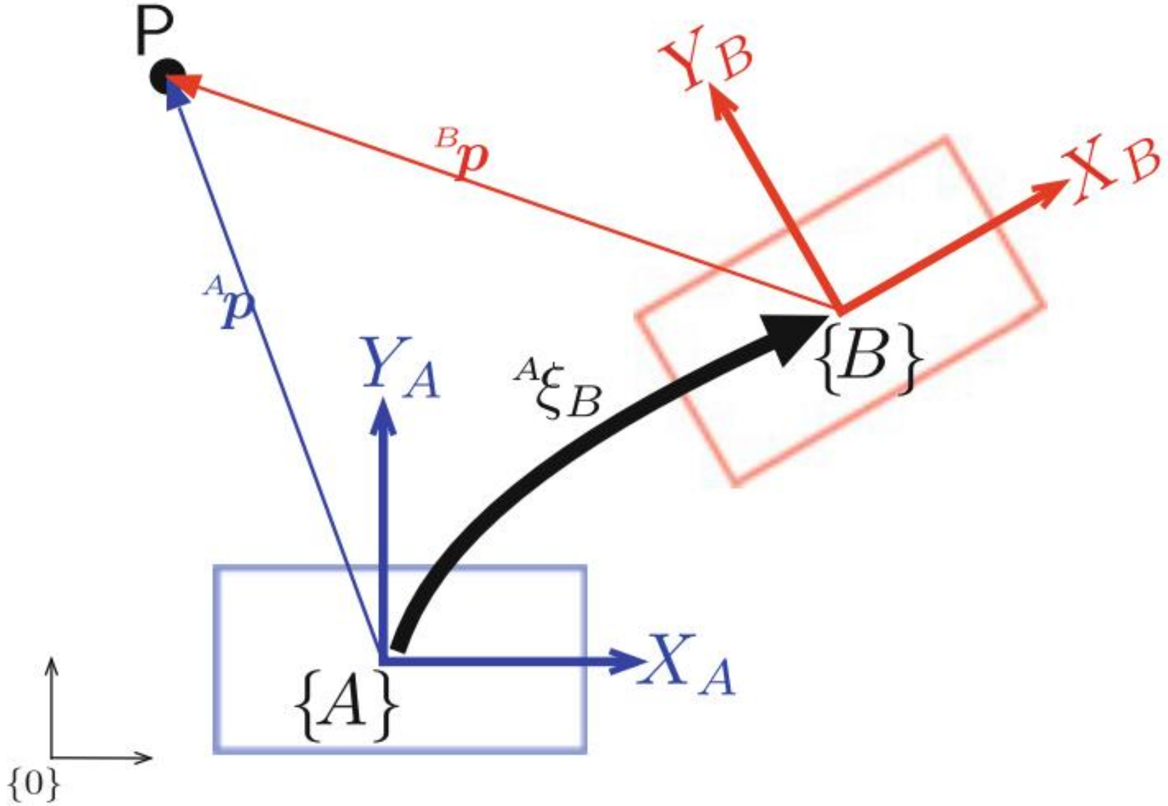
- Relative pose = translation + rotation.
- The position and orientation of the coordinate frame describe the position and orientation of the object it is attached to.
- Translation = distance; rotation = set of angles.
- Coordinate frame is denoted by $\{A\}$, {B},…
- Reference frame - $\{0\}$. Leading superscript is omitted, the reference frame is assumed.
- Body frame (associated with a moving body) - $\{B\}$.

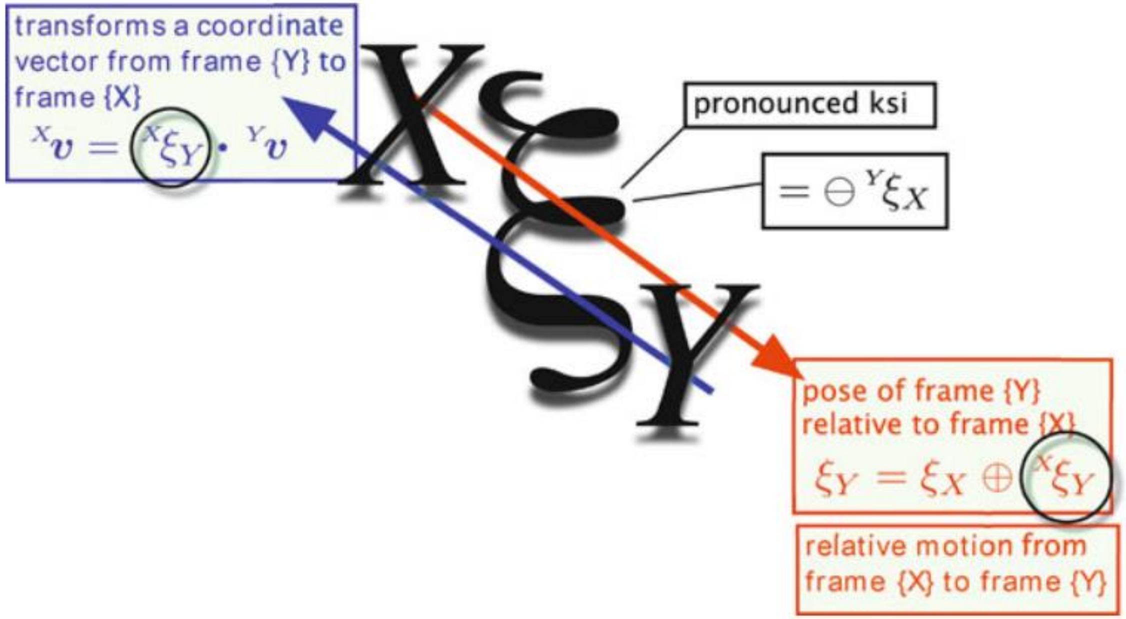


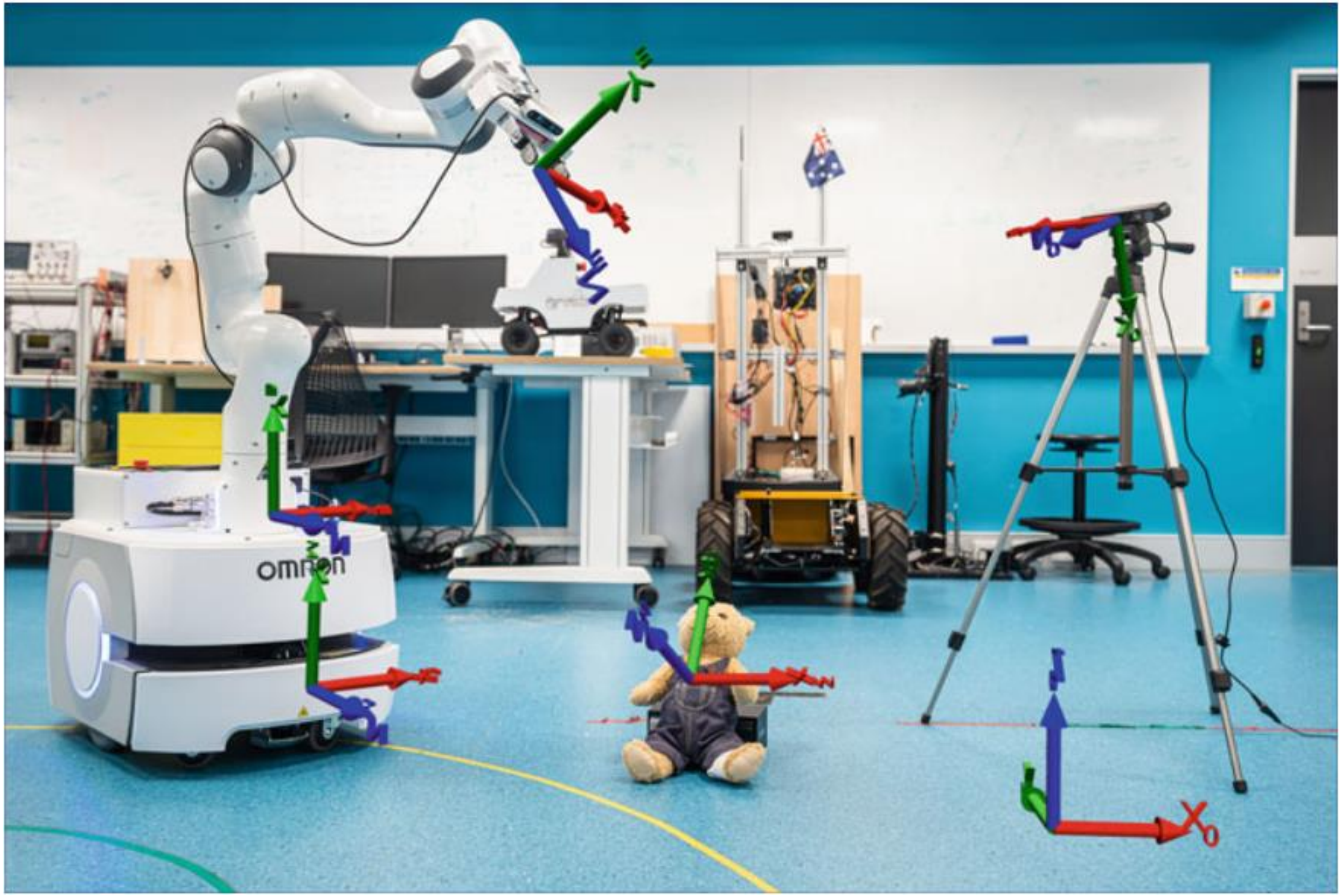
- A point P can be described by two different coordinate vectors.
- ${}^{A}\boldsymbol{p}$ with respect to frame {A}; ${}^{B}\boldsymbol{p}$ with respect to frame {B}.
- Two coordinate vectors are related by

$$ {}^{A}\boldsymbol{p} = {}^{A}\boldsymbol{\xi}_{B} \cdot {}^{B}\boldsymbol{p} $$

(motion from {A} to {B} and then to P)

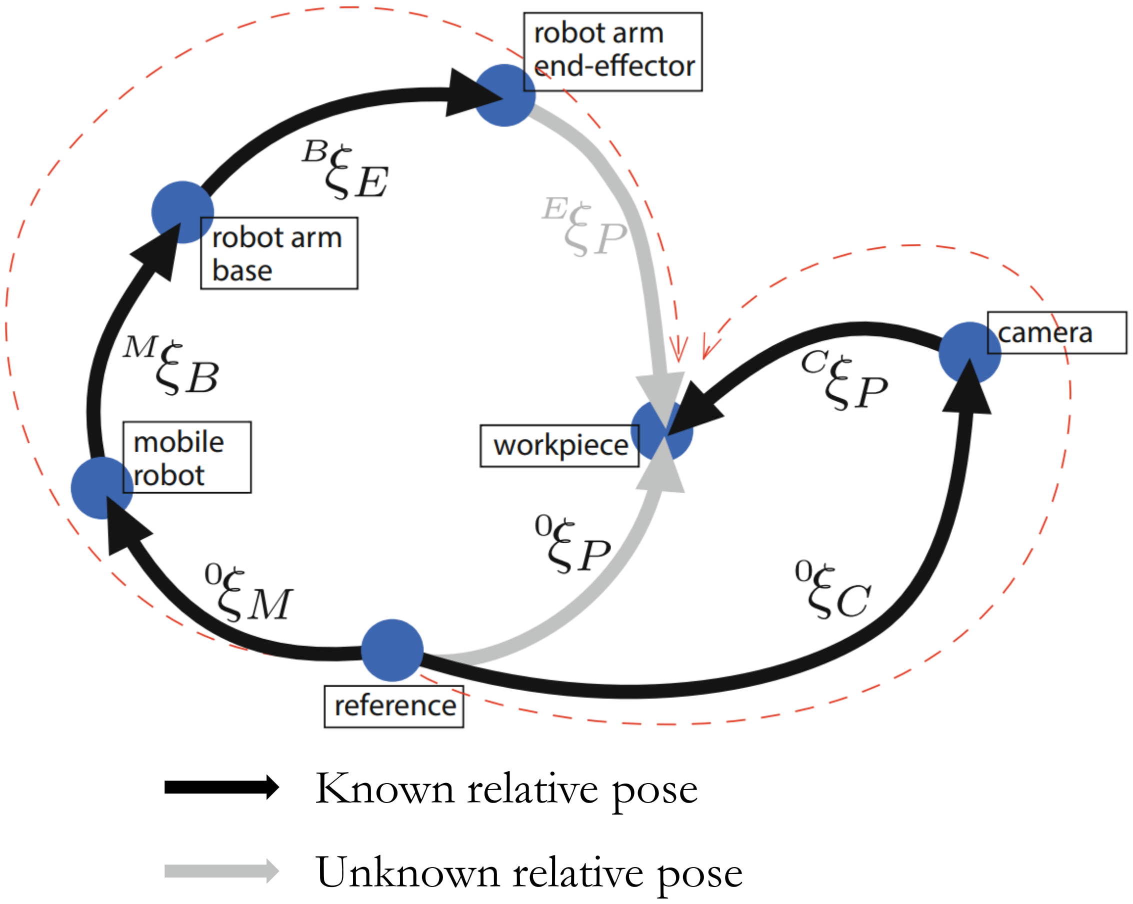





omron

1.1.3 Pose Graphs

- Pose graph = vertices + edges.
- Vertices – poses; Edges – motions (relative poses).



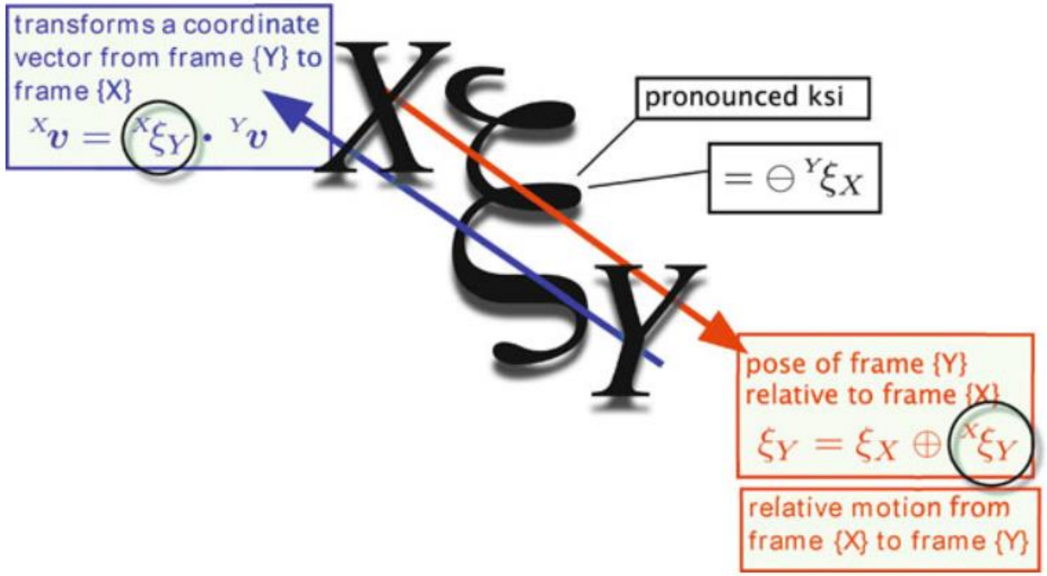
$$ {}^{0}\xi_{M} \oplus {}^{M}\xi_{B} \oplus {}^{B}\xi_{E} \oplus \underline{{}^{E}\xi_{P}} = {}^{0}\xi_{C} \oplus {}^{C}\xi_{P} $$

$$ \underline{{}^{E}\xi_{P}} = \ominus {}^{B}\xi_{E} \ominus {}^{M}\xi_{B} \ominus {}^{0}\xi_{M} \oplus {}^{0}\xi_{C} \oplus {}^{C}\xi_{P} $$

* If we traverse the edge in the direction of its arrow, precede it with the $\oplus$ operator, otherwise use $\ominus$.

1.1.4 Summary

- The position and orientation of an object is referred to as its pose.
- A motion, denoted by $\boldsymbol{\xi}$, causes a change in pose – it is a relative pose defined with respect to the initial pose.
- There is no absolute pose, a pose is always relative to a reference pose.
- We can perform algebraic manipulation of expressions written in terms of relative poses using the operators $\oplus$ and $\ominus$, and the concept of a null motion $\varnothing$.
- We can represent a set of poses, with known relative poses, as a pose graph.
- To quantify pose, we rigidly attach a coordinate frame to an object. The origin of that frame is the object's position, and the directions of the frame's axes describe its orientation.
- The constituent points of a rigid object are described by constant coordinate vectors relative to its coordinate frame.
- Any point can be described by a coordinate vector with respect to any coordinate frame. A coordinate vector can be transformed between frames by applying the relative pose of those frames to the vector using the $\cdot$ operator.



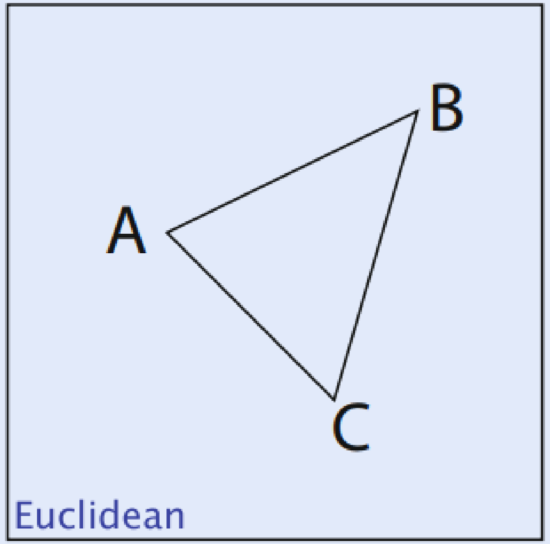
Excurse: Euclid of Alexandria

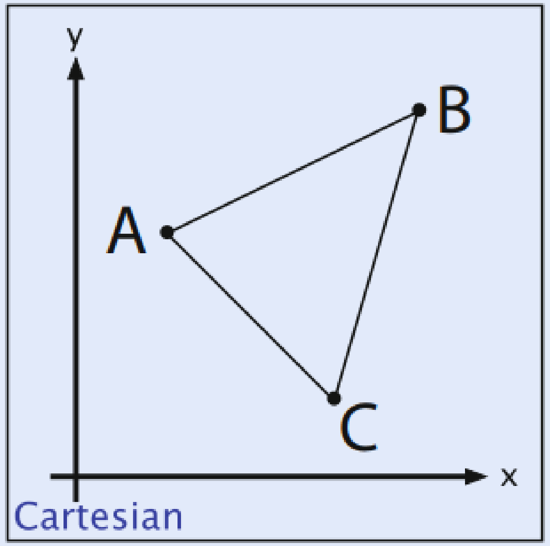
Euclid (325–265 BCE) was a Greek mathematician, who was born and lived in Alexandria, Egypt, and is considered the “father of geometry”, His great work *Elements*, comprising 13 books, captured and systematized much early knowledge about geometry and numbers. It deduces the properties of planar and solid geometric shapes from a set of 5 axioms and 5 postulates.

Elements is probably the most successful book in the history of mathematics. It describes plane geometry and is the basis for most people’s first introduction to geometry and formal proof, and is the basis of what we now call Euclidean geometry. Euclidean distance is simply the distance between two points on a plane. Euclid also wrote *Optics* which describes geometric vision and perspective.

Excurse: Euclidean versus Cartesian Geometry

Euclidean geometry is concerned with points and lines in the Euclidean plane (2D) or Euclidean space (3D). It is based entirely on a set of axioms and makes no use of arithmetic. Descartes added a coordinate system (2D or 3D) and was then able to describe points, lines and other curves in terms of algebraic equations. The study of such equations is called analytic geometry and is the basis of all modern geometry. The Cartesian plane (or space) is the Euclidean plane (or space) with all its axioms and postulates *plus* the extra facilities afforded by the added coordinate system. The term Euclidean geometry is often used to mean that Euclid's fifth postulate (parallel lines never intersect) holds, which is the case for a planar surface but not for a curved surface.





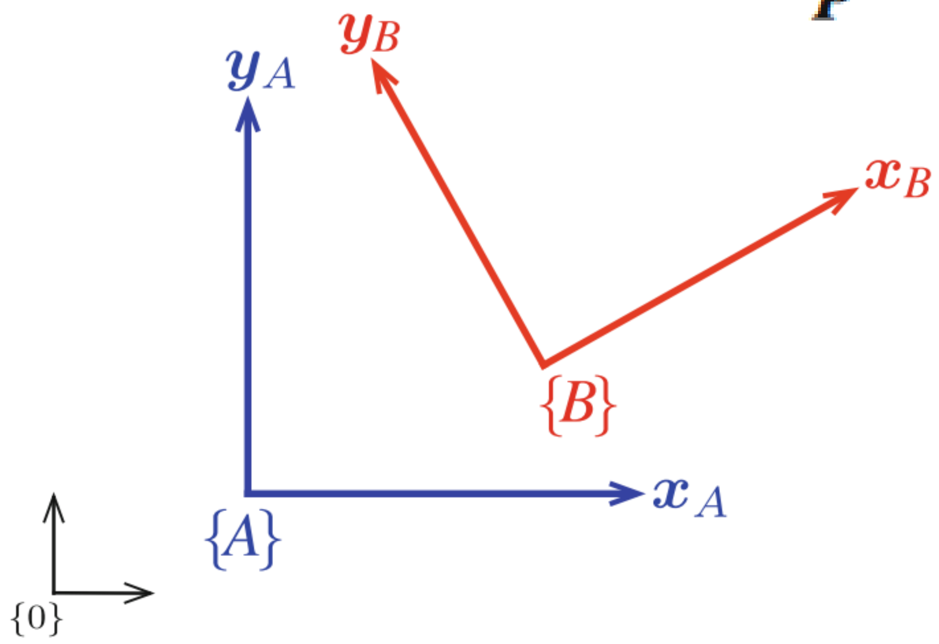
Excurse: René Descartes

Descartes (1596–1650) was a French philosopher, mathematician and part-time mercenary. He is famous for the philosophical statement "*Cogito, ergo sum*" or "*I am thinking, therefore I exist*" or "*I think, therefore I am*". He was a sickly child and developed a life-long habit of lying in bed and thinking until late morning. A possibly apocryphal story is that during one such morning he was watching a fly walk across the ceiling and realized that he could describe its position in terms of its distance from the two edges of the ceiling. This is the basis of the *Cartesian* coordinate system and modern (analytic) geometry, which he described in his 1637 book *La Géométrie*. For the first time, mathematics and geometry were connected, and modern calculus was built on this foundation by Newton and Leibniz. Living in Sweden, at the invitation of Queen Christina, he was obliged to rise at 5 A.M., breaking his lifetime habit – he caught pneumonia and died. His remains were later moved to Paris, and are now lost apart from his skull which is in the Musée de l'Homme. After his death, the Roman Catholic Church placed his works on the Index of Prohibited Books – the Index was not abolished until 1966.

1.2 Working in Two Dimensions (2D)

- Coordinate system: orthogonal axes, origin, basis vectors.
- The basis vectors are unit vectors, denoted by $\hat{x}$ and $\hat{y}$.
- A point is represented by its x- and y-coordinates (x, y) or as a coordinate vector from the origin to the point:

$$\boldsymbol{p} = x\hat{\boldsymbol{x}} + y\hat{\boldsymbol{y}}$$



Two 2D coordinate frames {A} and {B}, defined with respect to the reference frame {0}. {B} is rotated and translated with respect to {A}.

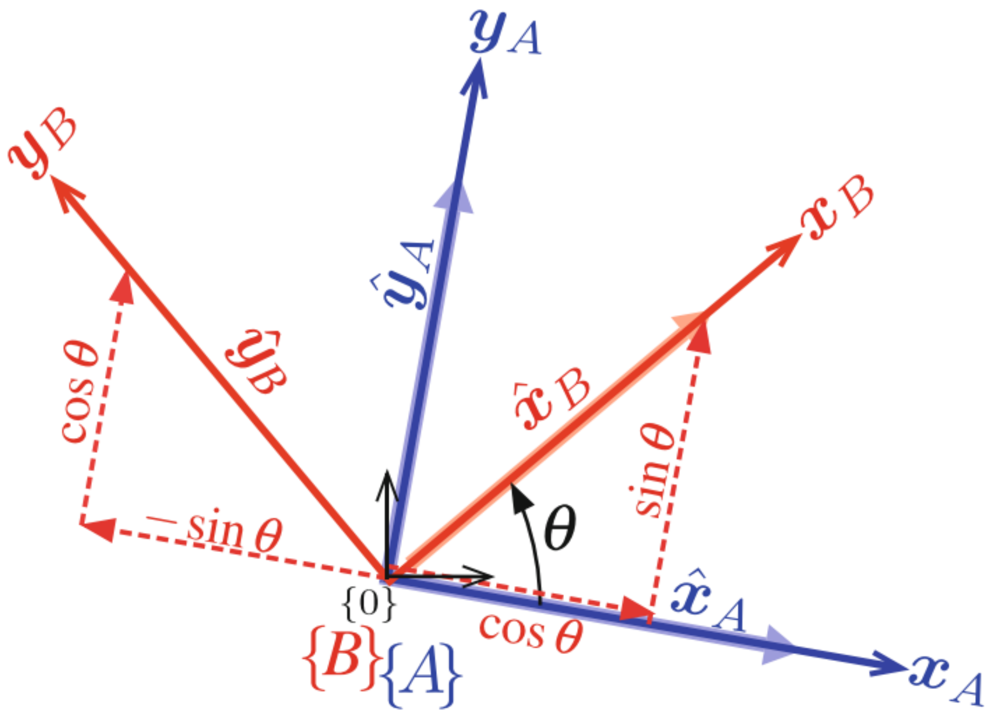
1.2.1 Orientation in Two Dimensions

1.2.1.1 2D Rotation Matrix

- Frame {B} is obtained by rotating frame {A} by θ in the positive (counter-clockwise) direction about the origin.
- Basis vectors of frame {B} can be expressed in terms of the basis vectors of frame {A}:

$$\hat{\boldsymbol{x}}_B = \hat{\boldsymbol{x}}_A \cos\theta + \hat{\boldsymbol{y}}_A \sin\theta$$
$$\hat{\boldsymbol{y}}_B = -\hat{\boldsymbol{x}}_A \sin\theta + \hat{\boldsymbol{y}}_A \cos\theta$$

$$\boxed{\begin{pmatrix}\hat{\boldsymbol{x}}_B & \hat{\boldsymbol{y}}_B\end{pmatrix} = \begin{pmatrix}\hat{\boldsymbol{x}}_A & \hat{\boldsymbol{y}}_A\end{pmatrix} \underbrace{\begin{pmatrix}\cos\theta & -\sin\theta \\ \sin\theta & \cos\theta\end{pmatrix}}_{{}^{A}\mathbf{R}_B(\theta)}}$$



- A rotation matrix transforms frame {A} into frame {B}

$${}^{A}\mathbf{R}_B(\theta) = \begin{pmatrix}\cos\theta & -\sin\theta \\ \sin\theta & \cos\theta\end{pmatrix}$$

- Rotation matrices have some special properties:
  - The columns are the basis vectors that define the axes of the rotated 2D coordinate frame, and therefore have unit length and are orthogonal.
  - It is an orthogonal (also called orthonormal) matrix ▶ and therefore its inverse is the same as its transpose, that is, $\mathbf{R}^{-1} = \mathbf{R}^\top$.
  - The matrix-vector product $\mathbf{R}\boldsymbol{v}$ preserves the length and relative orientation of vectors $\boldsymbol{v}$ and therefore its determinant is $+1$.
  - It is a member of the Special Orthogonal (SO) group of dimension 2 which we write as $\mathbf{R} \in \mathbf{SO}(2) \subset \mathbb{R}^{2\times 2}$. Being a group under the operation of matrix multiplication means that the product of any two matrices belongs to the group, as does its inverse.

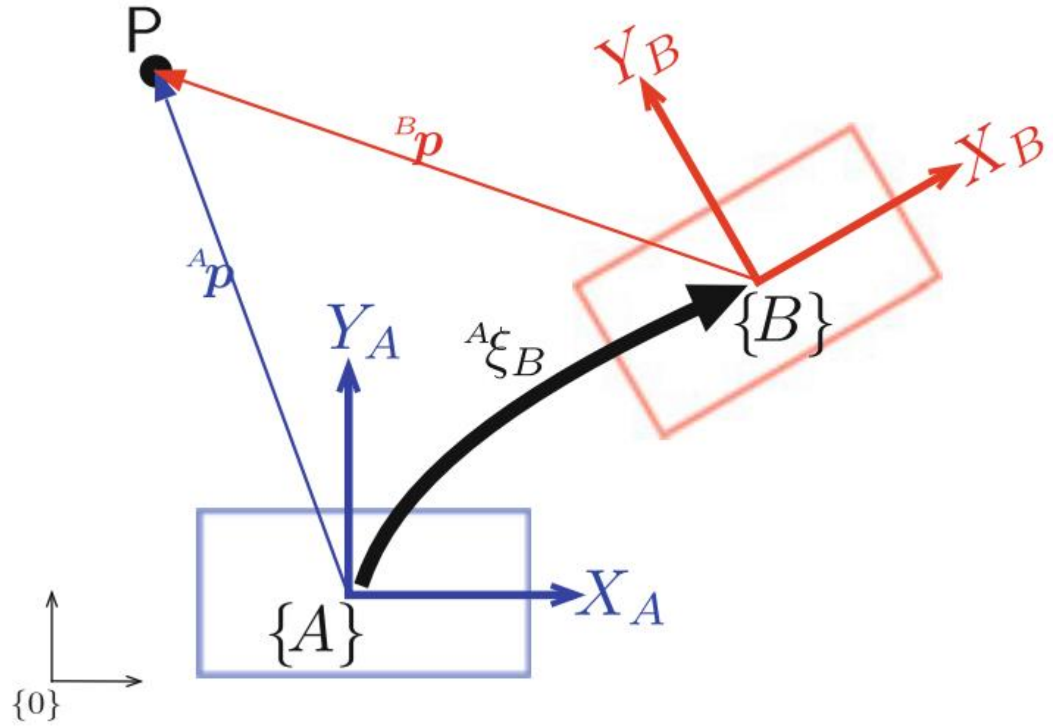
- The point can be described by the coordinate vector $({}^{A}p_x, {}^{A}p_y)^T$ with respect to frame {A} or the coordinate vector $({}^{B}p_x, {}^{B}p_y)^T$ with respect to frame {B}.

$$ {}^{A}\boldsymbol{p} = (\hat{\boldsymbol{x}}_A \quad \hat{\boldsymbol{y}}_A)\begin{pmatrix} {}^{A}p_x \\ {}^{A}p_y \end{pmatrix}, \quad {}^{B}\boldsymbol{p} = (\hat{\boldsymbol{x}}_B \quad \hat{\boldsymbol{y}}_B)\begin{pmatrix} {}^{B}p_x \\ {}^{B}p_y \end{pmatrix}. $$

$$ {}^{A}\boldsymbol{p} = (\hat{\boldsymbol{x}}_A \quad \hat{\boldsymbol{y}}_A)\, {}^{A}\mathbf{R}_B(\theta)\begin{pmatrix} {}^{B}p_x \\ {}^{B}p_y \end{pmatrix} $$

$$ (\hat{\boldsymbol{x}}_A \quad \hat{\boldsymbol{y}}_A)\underline{\begin{pmatrix} {}^{A}p_x \\ {}^{A}p_y \end{pmatrix}} = (\hat{\boldsymbol{x}}_A \quad \hat{\boldsymbol{y}}_A)\, \underline{{}^{A}\mathbf{R}_B(\theta)\begin{pmatrix} {}^{B}p_x \\ {}^{B}p_y \end{pmatrix}} $$

$$ \boxed{\begin{pmatrix} {}^{A}p_x \\ {}^{A}p_y \end{pmatrix} = {}^{A}\mathbf{R}_B(\theta)\begin{pmatrix} {}^{B}p_x \\ {}^{B}p_y \end{pmatrix}} $$



- A rotation matrix has all the characteristics of relative pose.

ξ as an SO(2) Matrix

For the case of pure rotation in 2D, $\boldsymbol{\xi}$ can be implemented by a rotation matrix $\mathbf{R} \in \mathbf{SO}(2)$. Its implementation is:

composition	$\boldsymbol{\xi}_1 \oplus \boldsymbol{\xi}_2$	$\mapsto \mathbf{R}_1\mathbf{R}_2$, matrix multiplication
inverse	$\ominus\boldsymbol{\xi}$	$\mapsto \mathbf{R}^{-1} = \mathbf{R}^\top$, matrix transpose
identity	$\varnothing$	$\mapsto \mathbf{R}(0) = \mathbf{1}_{2\times2}$, identity matrix
vector-transform	$\boldsymbol{\xi} \cdot \boldsymbol{v}$	$\mapsto \mathbf{R}\boldsymbol{v}$, matrix-vector product

Composition is commutative, that is, $\mathbf{R}_1\mathbf{R}_2 = \mathbf{R}_2\mathbf{R}_1$, and $\mathbf{R}(-\theta) = \mathbf{R}^\top(\theta)$.

- **Example:**

```
>> R = rotm2d(0.3)
R =
     0.9553   -0.2955
     0.2955    0.9553
>> plottform2d(R);
>> det(R)
ans =
     1.0000
>> det(R*R)
ans =
     1.0000
```

```
>> syms theta real
>> R = rotm2d(theta)
R =
[cos(theta), -sin(theta)]
[sin(theta),  cos(theta)]
>> simplify(R * R)
ans =
[cos(2*theta), -sin(2*theta)]
[sin(2*theta),  cos(2*theta)]
>> det(R)
ans =
cos(theta)^2 + sin(theta)^2
>> simplify(ans)
ans =
1
```

Excurse: 2D Skew-Symmetric Matrix

In 2 dimensions, the skew- or anti-symmetric matrix is

$$[\omega]_\times = \begin{pmatrix} 0 & -\omega \\ \omega & 0 \end{pmatrix} \in \mathbf{so}(2) \tag{2.14}$$

which has a distinct structure with a zero diagonal and only one unique value $\omega \in \mathbb{R}$, and $[\omega]_\times^\top = -[\omega]_\times$. The vector space of 2D skew-symmetric matrices is denoted $\mathbf{so}(2)$ and is the Lie algebra of $\mathbf{SO}(2)$. The $[\cdot]_\times$ operator is implemented by

```
>> X = vec2skew(2)
X =
     0    -2
     2     0
```

and the inverse operator $\vee_\times(\cdot)$ by

```
>> skew2vec(X)
ans =
     2
```

1.2.1.2 Matrix Exponential for Rotation

- There is a fascinating, and very useful, connection between a rotation matrix and the exponential of a skew-symmetric matrix.
- Considering a pure rotation of 0.3 radians expressed as a rotation

```
>> R = rotm2d(0.3)
R =
    0.9553   -0.2955
    0.2955    0.9553
```

- We can take the matrix logarithm using the function `logm`

```
>> L = logm(R)
L =
         0   -0.3000
    0.3000         0
```

- The result is a skew-symmetric matrix with the unique element of 0.3 which is the rotation angle.

```
>> S = skew2vec(L)
S =
    0.3000
```

- Exponentiating the logarithm of the rotation matrix yields the original rotation matrix:

```
>> expm(L)
ans =
    0.9553   -0.2955
    0.2955    0.9553
```

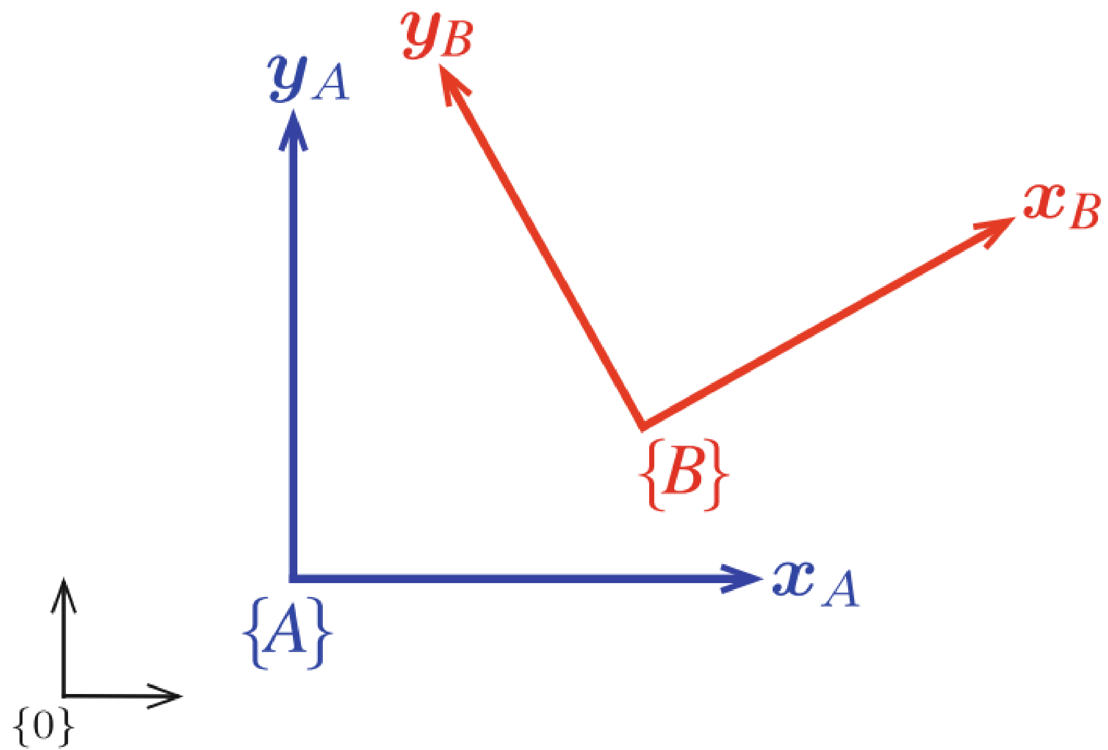
- In general, we can write

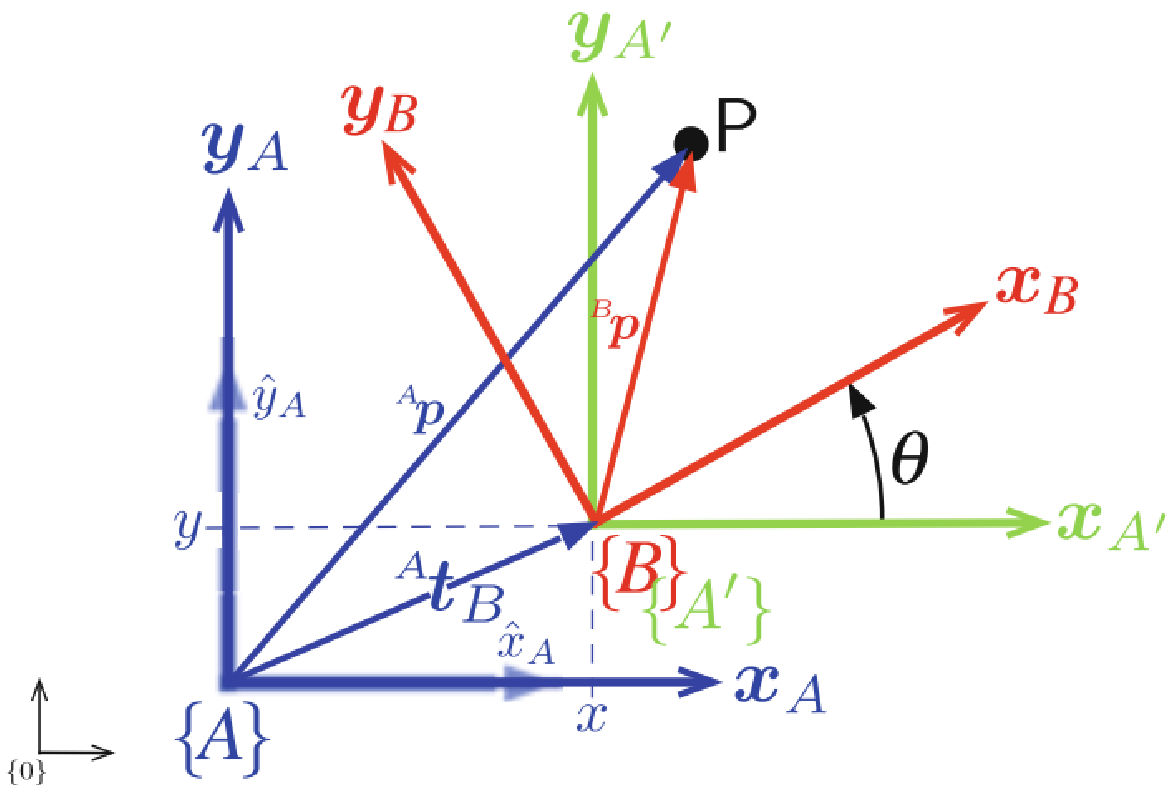
$$\mathbf{R} = \mathrm{e}^{[\theta]_\times} \in \mathbf{SO}(2)$$

where θ is the rotation angle, and $[\cdot]_\times : \mathbb{R} \mapsto \mathbf{SO}(2) \subset \mathbb{R}^{2\times 2}$ is a mapping from a scalar to a skew-symmetric matrix.

1.2.2 Pose in Two Dimensions

- To describe the relative pose of the frames, we need to account for the translation between the origins of the frames as well as the rotation.





1.2.2.1 2D Homogeneous Transformation Matrix

- First step: {B} → {A'} using the rotation matrix ${}^{A}\boldsymbol{R}_{B}(\theta)$.
- Second step: {A'} → {A} using the translation ${}^{A}\boldsymbol{t}_{B}$.

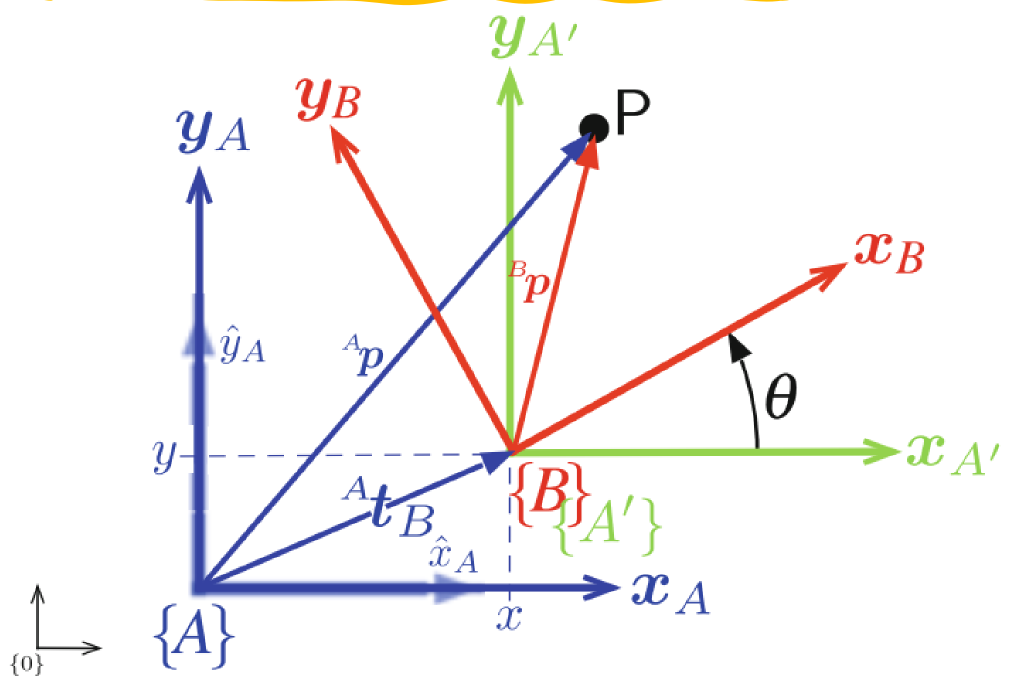
$$\begin{pmatrix} {}^{A}x \\ {}^{A}y \end{pmatrix} = \begin{pmatrix} {}^{A'}x \\ {}^{A'}y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

$$= \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} {}^{B}x \\ {}^{B}y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

$$= \begin{pmatrix} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \end{pmatrix} \begin{pmatrix} {}^{B}x \\ {}^{B}y \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} {}^{A}x \\ {}^{A}y \\ 1 \end{pmatrix} = \begin{pmatrix} {}^{A}\mathbf{R}_{B}(\theta) & {}^{A}\boldsymbol{t}_{B} \\ \mathbf{0}_{1\times 2} & 1 \end{pmatrix} \begin{pmatrix} {}^{B}x \\ {}^{B}y \\ 1 \end{pmatrix}$$

$$\boxed{\begin{pmatrix} {}^{A}x \\ {}^{A}y \\ 1 \end{pmatrix} = \begin{pmatrix} {}^{A}\mathbf{R}_{B}(\theta) & {}^{A}\boldsymbol{t}_{B} \\ \mathbf{0}_{1\times 2} & 1 \end{pmatrix} \begin{pmatrix} {}^{B}x \\ {}^{B}y \\ 1 \end{pmatrix}}$$



Excurse: Homogeneous Vectors

A vector $\boldsymbol{p} = (x, y)$ is written in homogeneous form as $\tilde{\boldsymbol{p}} = (x_1, x_2, x_3) \in \mathbb{P}^2$ where $\mathbb{P}^2$ is the 2-dimensional projective space, and the tilde indicates the vector is homogeneous.

Homogeneous vectors have the important property that $\tilde{\boldsymbol{p}}$ is equivalent to $\lambda\tilde{\boldsymbol{p}}$ for all $\lambda \neq 0$ which we write as $\tilde{\boldsymbol{p}} \simeq \lambda\tilde{\boldsymbol{p}}$. That is, $\tilde{\boldsymbol{p}}$ represents the same point in the plane irrespective of the overall scaling factor. Homogeneous representation is also used in computer vision which we discuss in Part IV. Additional details are provided in ▶ App. C.2.

To convert a point to homogeneous form we typically append an element equal to one, for the 2D case this is $\tilde{\boldsymbol{p}} = (x, y, 1)$. The dimension of the vector has been increased by one, and a point on a plane is now represented by a 3-vector. The Euclidean or nonhomogeneous coordinates are related by $x = x_1/x_3$, $y = x_2/x_3$ and $x_3 \neq 0$.

- The coordinate vectors for point P are now expressed in homogeneous form:

$$
\begin{aligned}
{}^{A}\tilde{\boldsymbol{p}} &= \begin{pmatrix} {}^{A}\mathbf{R}_{B}(\theta) & {}^{A}\boldsymbol{t}_{B} \\ \mathbf{0}_{1\times 2} & 1 \end{pmatrix} {}^{B}\tilde{\boldsymbol{p}} \\
&= {}^{A}\mathbf{T}_{B}\, {}^{B}\tilde{\boldsymbol{p}}
\end{aligned}
$$

and ${}^{A}\boldsymbol{T}_{B}$ is referred to as a homogeneous transformation – it transforms homogeneous vectors.

- The matrix has a very specific structure and belongs to the Special Euclidean (SE) group of dimension 2: ${}^{A}\boldsymbol{T}_{B} \in \mathbf{se}(2) \subset \mathbb{R}^{3\times 3}$.

- The matrix ${}^{A}\boldsymbol{T}_{B}$ represents translation and rotation.

ξ as an $\mathbf{SE}(2)$ Matrix

For the case of rotation and translation in 2D, $\boldsymbol{\xi}$ can be implemented by a homogeneous transformation matrix $\mathbf{T} \in \mathbf{SE}(2)$ which is sometimes written as an ordered pair $(\mathbf{R}, \boldsymbol{t}) \in \mathbf{SO}(2) \times \mathbb{R}^2$. The implementation is:

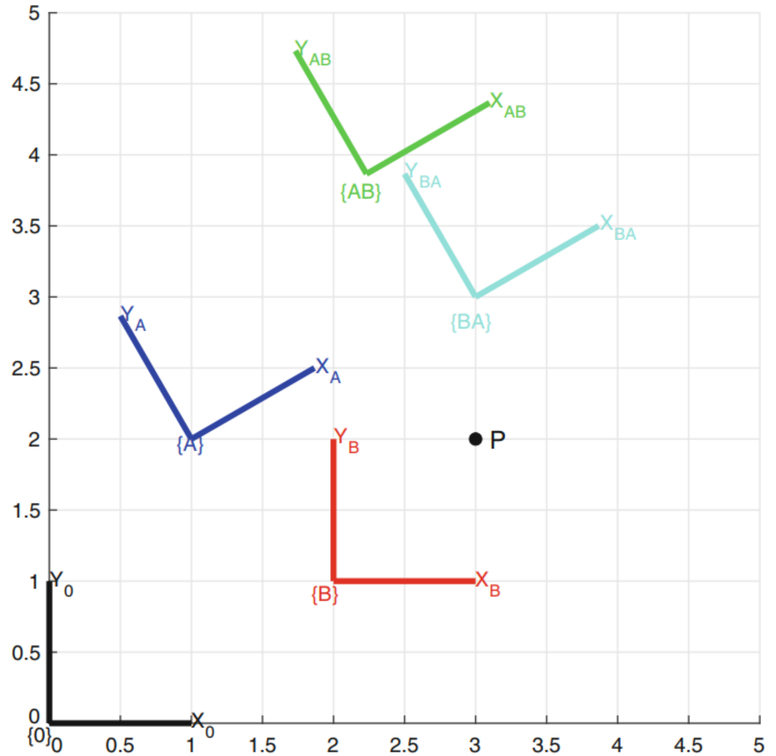
composition	$\boldsymbol{\xi}_1 \oplus \boldsymbol{\xi}_2$	$\mapsto \mathbf{T}_1\mathbf{T}_2 = \begin{pmatrix} \mathbf{R}_1\mathbf{R}_2 & \boldsymbol{t}_1 + \mathbf{R}_1\boldsymbol{t}_2 \\ \mathbf{0}_{1\times 2} & 1 \end{pmatrix}$, matrix multiplication
inverse	$\ominus\boldsymbol{\xi}$	$\mapsto \mathbf{T}^{-1} = \begin{pmatrix} \mathbf{R}^\top & -\mathbf{R}^\top\boldsymbol{t} \\ \mathbf{0}_{1\times 2} & 1 \end{pmatrix}$, matrix inverse
identity	$\varnothing$	$\mapsto \mathbf{1}_{3\times 3}$, identity matrix
vector-transform	$\boldsymbol{\xi} \cdot \boldsymbol{v}$	$\mapsto \epsilon(\mathbf{T}\tilde{\boldsymbol{v}})$, matrix-vector product

where $\tilde{\cdot} : \mathbb{R}^2 \mapsto \mathbb{P}^2$ and $\epsilon(\cdot) : \mathbb{P}^2 \mapsto \mathbb{R}^2$. Composition is not commutative, that is, $\mathbf{T}_1\mathbf{T}_2 \neq \mathbf{T}_2\mathbf{T}_1$.

- **Example:**

```
>> rotm2d(0.3)
ans =
    0.9553   -0.2955
    0.2955    0.9553
>> tformr2d(0.3)
ans =
    0.9553   -0.2955         0
    0.2955    0.9553         0
         0         0    1.0000
>> TA = trvec2tform([1 2])*tformr2d(deg2rad(30))
TA =
    0.8660   -0.5000    1.0000
    0.5000    0.8660    2.0000
         0         0    1.0000
>> axis([0 5 0 5]);
>> plottform2d(TA,frame="A",color="b");
>> T0 = trvec2tform([0 0]);
>> plottform2d(T0, frame="0",color="k");
```

```
>> TB = trvec2tform([2 1])
>> plottform2d(TB,frame="B",color="r");
>> TAB = TA*TB
>> plottform2d(TAB,frame="AB",color="g");
>> TBA = TB*TA;
>> plottform2d(TBA,frame="BA",color="c");
>> P = [3;2];
>> plotpoint(P',"ko",label="P");
```



Determine the coordinate of the point with respect to {A}:

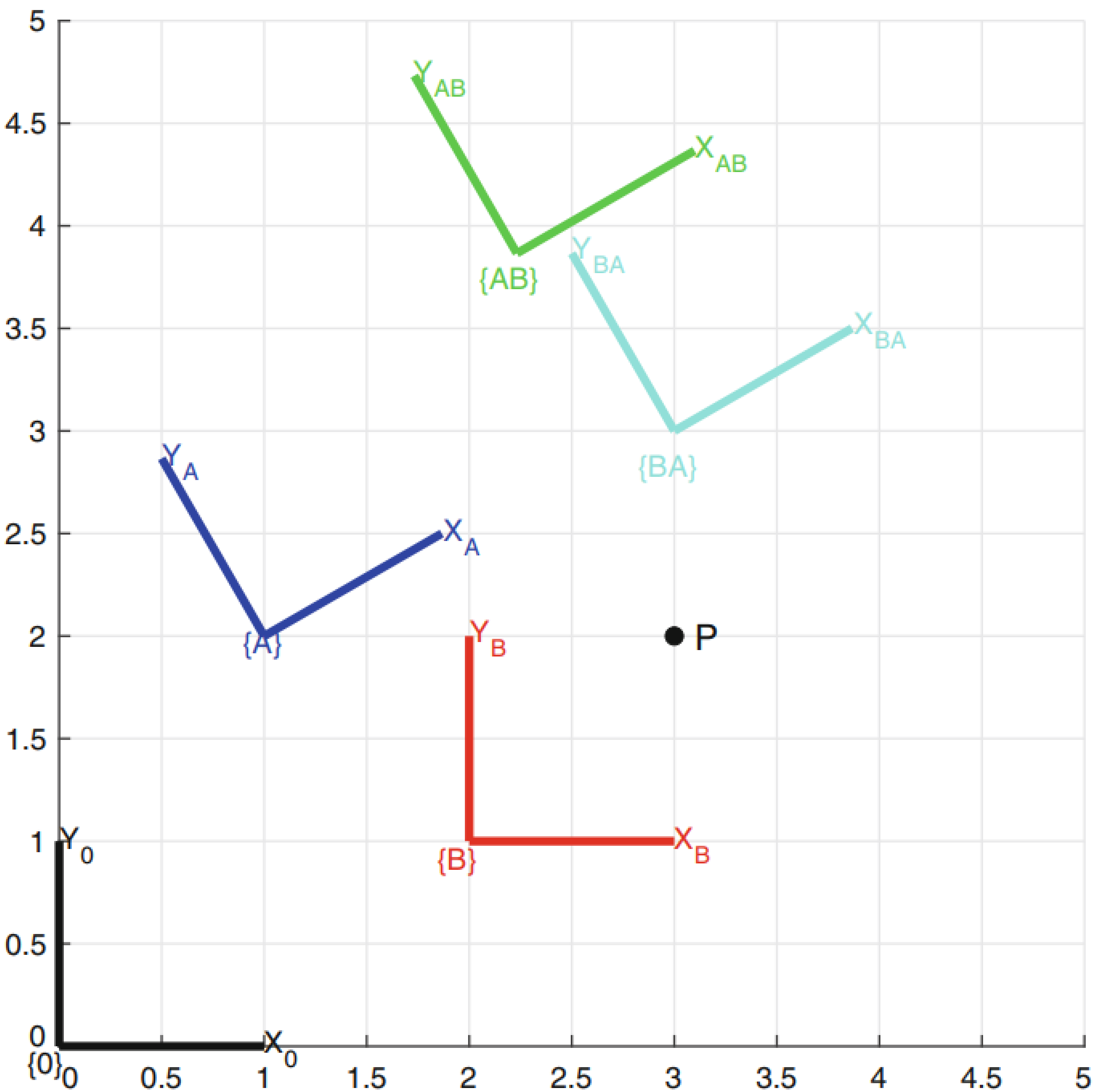
$$^{0}\boldsymbol{p} = {}^{0}\boldsymbol{\xi}_{A} \cdot {}^{A}\boldsymbol{p}$$

$$^{A}\boldsymbol{p} = (\ominus {}^{0}\boldsymbol{\xi}_{A}) \cdot {}^{0}\boldsymbol{p}$$

```
>> inv(TA)*[P;1]
ans =
     1.7321
    -1.0000
     1.0000

>> h2e(ans')
ans =
     1.7321   -1.0000

>> homtrans(inv(TA),P')
ans =
     1.7321   -1.0000
```

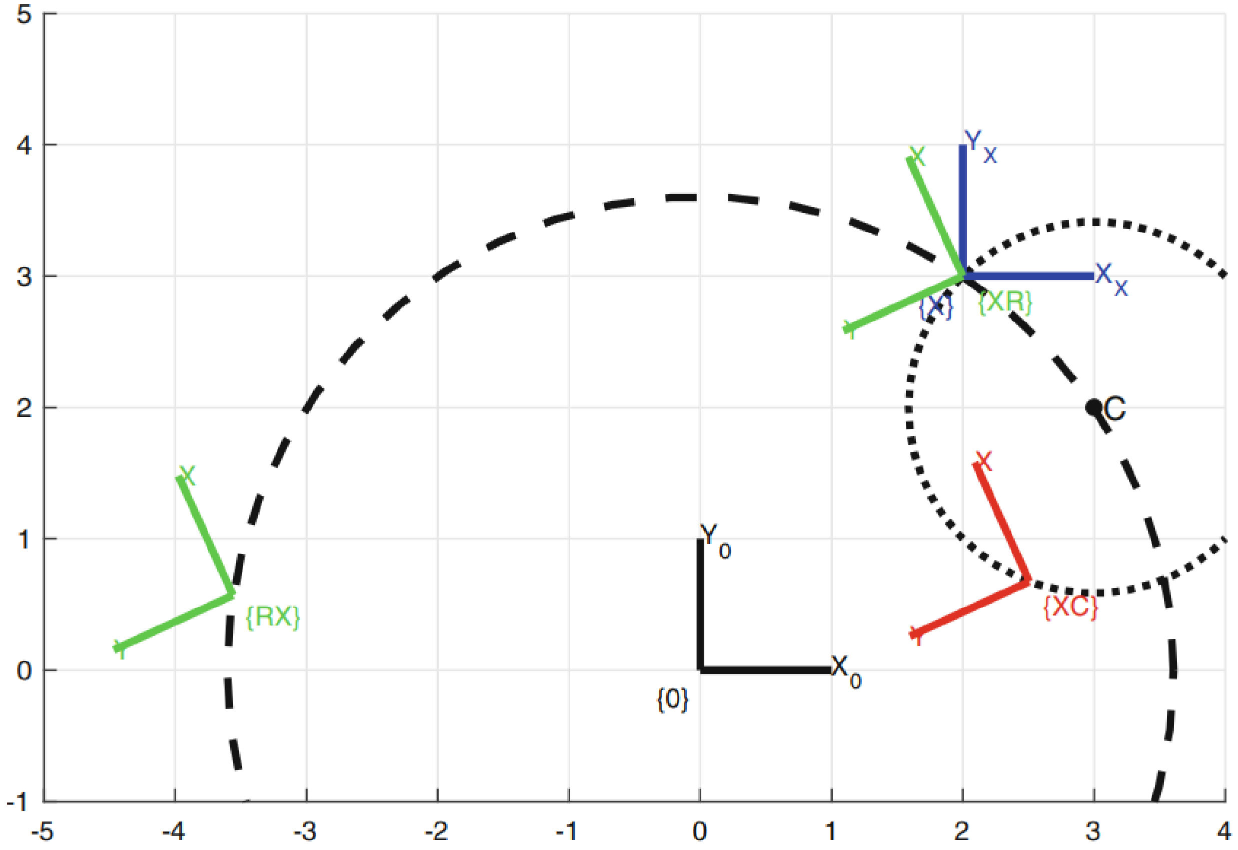


1.2.2.2 Rotating a Coordinate Frame

- We will explore rotation of coordinate frames.

```
>> axis([-5 4 -1 5]);
>> T0 = trvec2tform([0 0]);
>> plottform2d(T0,frame="0",color="k");
>> TX = trvec2tform([2 3]);
>> plottform2d(TX,frame="X",color="b");
>> TR = tformr2d(2);
>> plottform2d(TR*TX,framelabel="RX",color="g");
>> plottform2d(TX*TR,framelabel="XR",color="g");
```

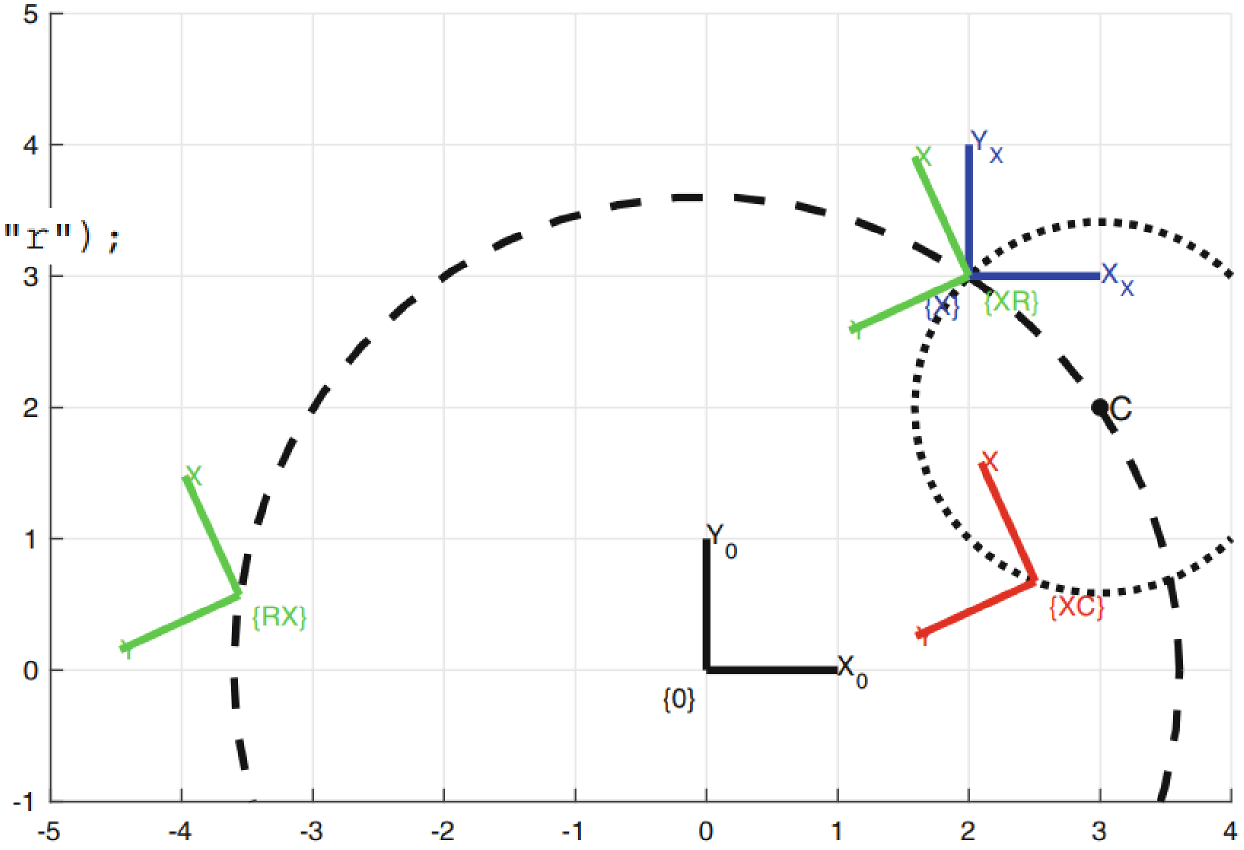
- The frame {X} is rotated by 2 radians about {0} to give frame {RX}, about {X} to give {XR}.



- Rotate a coordinate frame about some arbitrary point C.

```
>> C = [3 2];
>> plotpoint(C,"ko",label="C");

>> TC = trvec2tform(C)*TR*trvec2tform(-C)
TC =
   -0.4161   -0.9093    6.0670
    0.9093   -0.4161    0.1044
         0         0    1.0000
>> plottform2d(TC*TX,framelabel="XC",color="r");
```



S1. Place C at the origin of the reference frame.
S2. Rotate the shifted version of {X} about the origin, which is where C now is.
S3. Rotate the shifted version of {X} about the origin, which is where C now is.

1.2.2.3 Matrix Exponential for Pose

- We can take the logarithm of the **SE**(2) matrix TC from the previous example

```
>> L = logm(TC)
L =
         0   -2.0000    4.0000
    2.0000         0   -6.0000
         0         0         0
```

and the result is an augmented skew-symmetric matrix:

- the upper-left corner is a 2x2 skew-symmetric matrix;
- the upper right column is a 2-vector;
- the bottom row is zero.

- The three unique elements can be unpacked

```
>> S = skewa2vec(L)
S =
    2.0000    4.0000   -6.0000
```

and the first element is the rotation angle.

- Exponentiating L yields the original **SE**(2) matrix:

```
>> expm(vec2skewa(S))
ans =
   -0.4161   -0.9093    6.0670
    0.9093   -0.4161    0.1044
         0         0    1.0000
```

- In general, we can write: $\mathbf{T} = e^{[S]} \in \mathbf{SE}(2)$, where $\boldsymbol{S} \in \mathbb{R}^3$ and $[\cdot]: \mathbb{R}^3 \longmapsto \mathbf{se}(2) \subset \mathbb{R}^{3\times 3}$.
- The vector S is a twist vector.

Excurse: 2D Augmented Skew-Symmetric Matrix

In 2 dimensions, the augmented skew-symmetric matrix corresponding to the vector $\boldsymbol{S} = (\omega, v_x, v_y)$ is

$$[\boldsymbol{S}] = \left(\begin{array}{cc|c} 0 & -\omega & v_x \\ \omega & 0 & v_y \\ \hline 0 & 0 & 0 \end{array}\right) \in \mathbf{se}(2) \tag{2.19}$$

which has a distinct structure with a zero diagonal and bottom row, and a skew-symmetric matrix in the top-left corner. The vector space of 2D augmented skew-symmetric matrices is denoted $\mathbf{se}(2)$ and is the Lie algebra of $\mathbf{SE}(2)$. The $[\cdot]$ operator is implemented by

```
>> X = vec2skewa([1 2 3])
X =
     0    -1     2
     1     0     3
     0     0     0
```

and the inverse operator $\vee(\cdot)$ by

```
>> skewa2vec(X)
ans =
     1     2     3
```

1.2.2.4 2D Twists

- A twist = a rotational center and rotation angle that will rotate the first frame into the second.
- 2D twist vector $(\omega, \boldsymbol{v}) \in \mathbb{R}^3$ comprising a scalar ω and a moment $\boldsymbol{v} \in \mathbb{R}^2$.

```
>> S = Twist2d.UnitRevolute(C)
S =
( 1; 2  -3 )
```

- This particular twist is a unit twist that describes a rotation 1 rad about the point C. We require a rotation of 2 rad about the point C:

```
>> expm(vec2skewa(2*S.compact))
ans =
   -0.4161   -0.9093    6.0670
    0.9093   -0.4161    0.1044
         0         0    1.0000
```

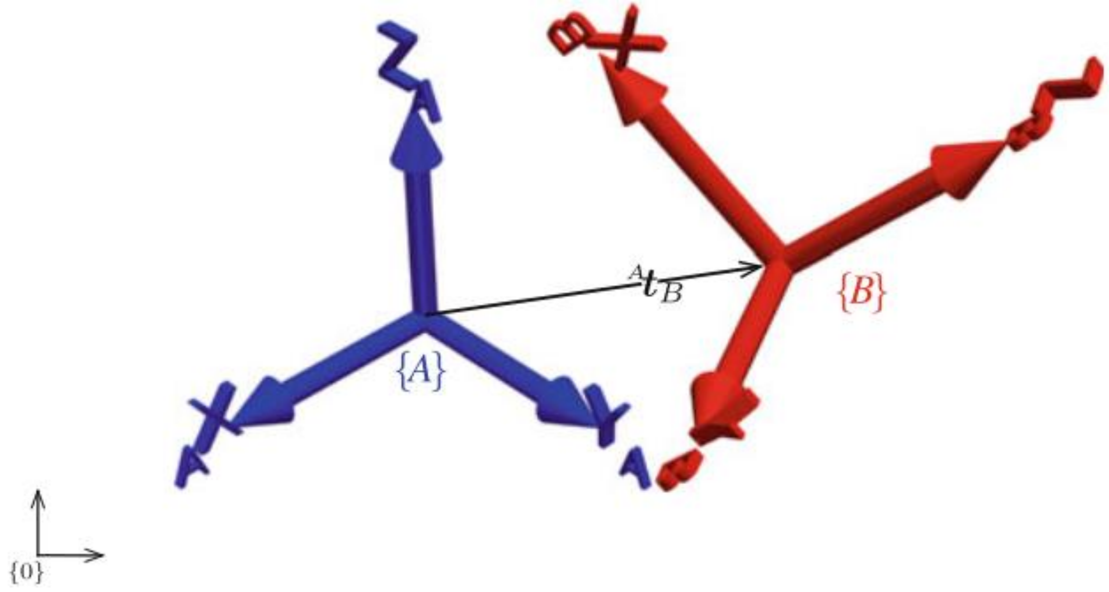
```
>> S.tform(2)
ans =
   -0.4161   -0.9093    6.0670
    0.9093   -0.4161    0.1044
         0         0    1.0000
```

1.3 Working in Three Dimensions (3D)

- The direction of the z-axis obeys the right-hand rule and forms a right-handed coordinate frame.
- Basis vectors: $\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}$ and $\hat{\boldsymbol{z}}$.

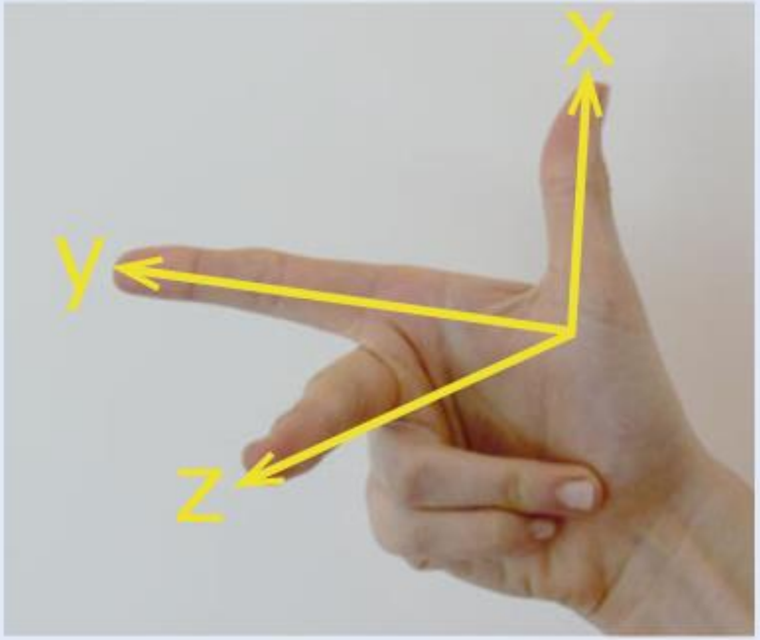
$$\hat{\boldsymbol{z}} = \hat{\boldsymbol{x}} \times \hat{\boldsymbol{y}}, \quad \hat{\boldsymbol{x}} = \hat{\boldsymbol{y}} \times \hat{\boldsymbol{z}}, \quad \hat{\boldsymbol{y}} = \hat{\boldsymbol{z}} \times \hat{\boldsymbol{x}}$$

- A point P is represented by: $\boldsymbol{p} = x\hat{\boldsymbol{x}} + y\hat{\boldsymbol{y}} + z\hat{\boldsymbol{z}}$



Excurse: Right-Hand Rule

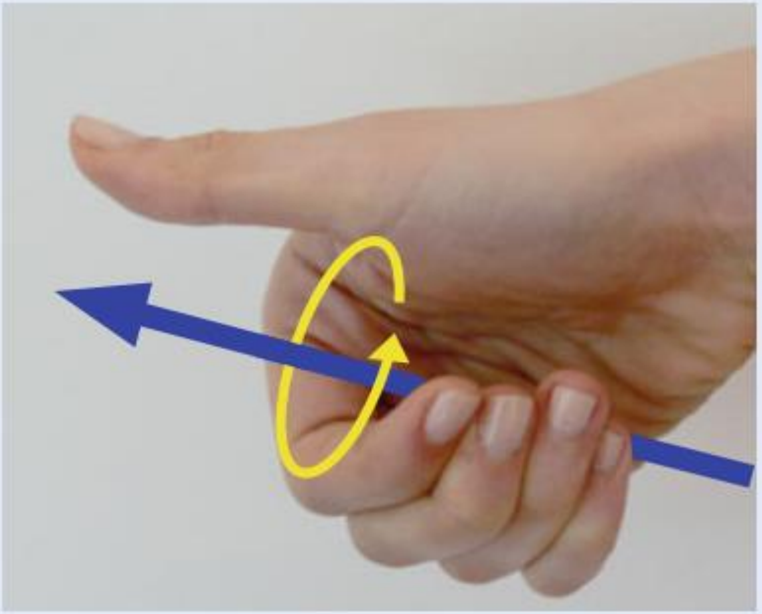
A right-handed coordinate frame is defined by the first three fingers of your right hand which indicate the relative directions of the x-, y- and z-axes respectively.



1.3.1 Orientation in Three Dimensions

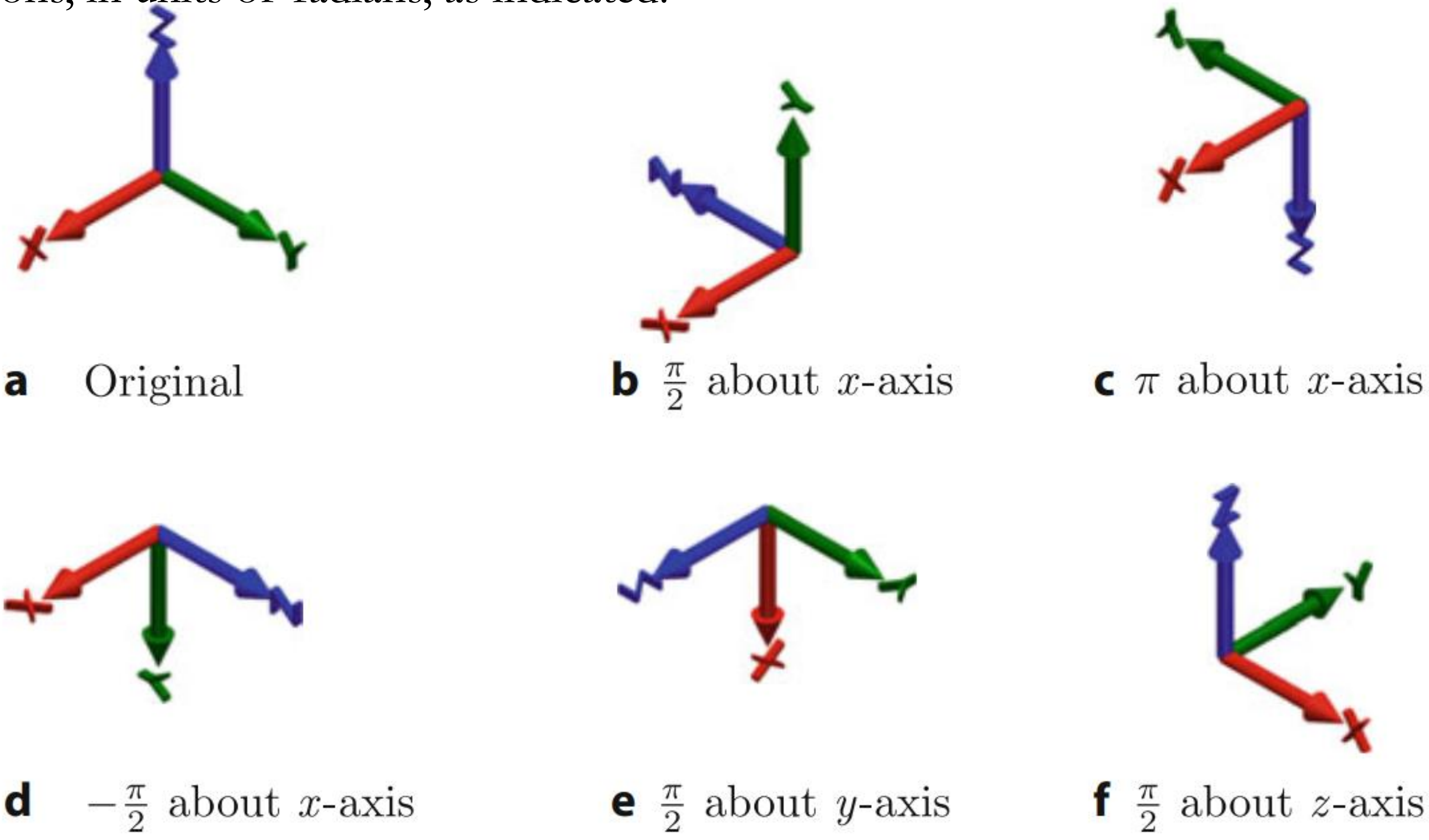
- We start by considering rotation about a single coordinate frame axis.
- Excurse: Rotation About a Vector

Wrap your right hand around the vector with your thumb (your x-finger) in the direction of the arrow. The curl of your fingers indicates the direction of increasing angle.

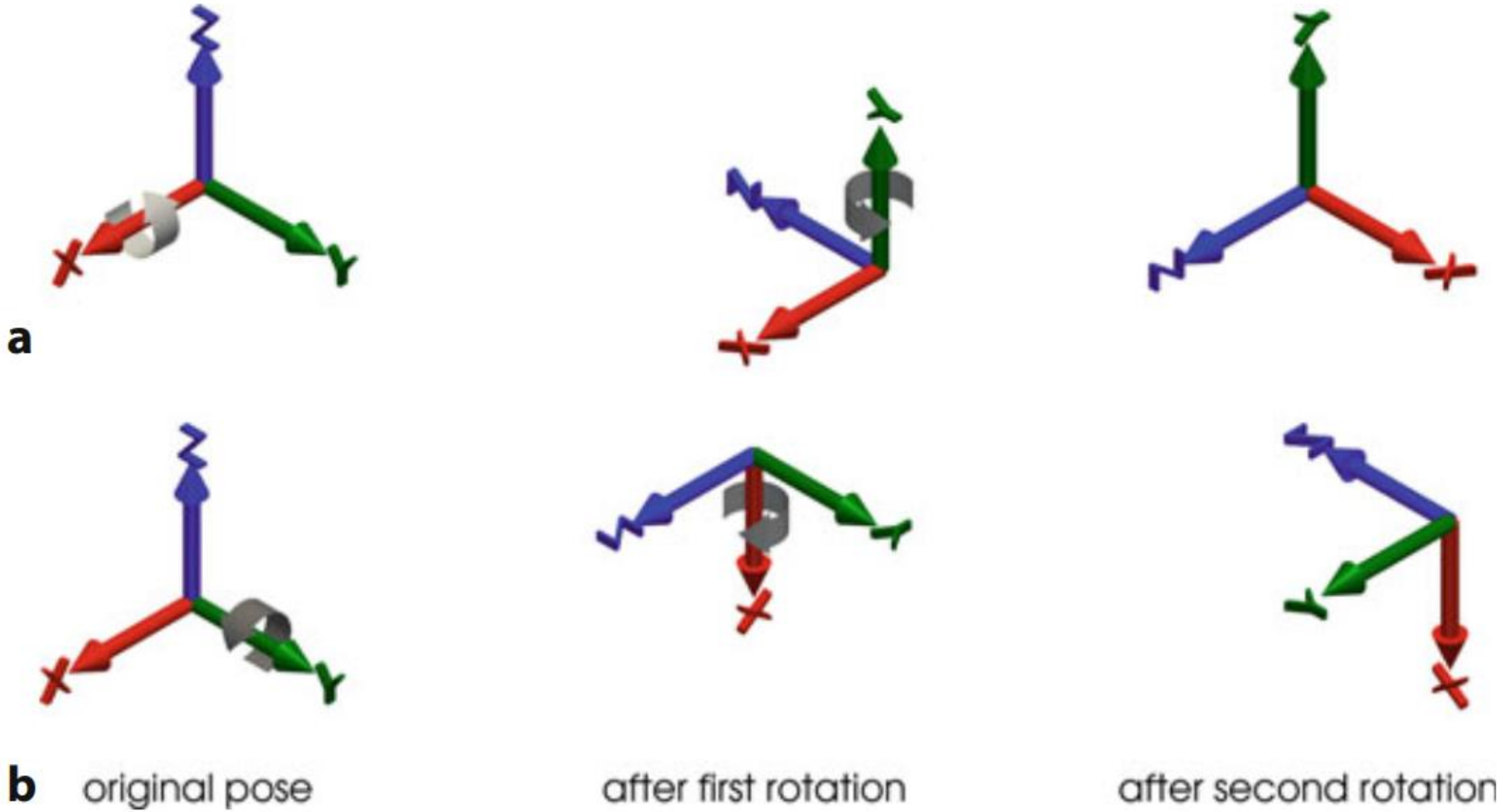


Rotation of a 3D coordinate frame. **a** The original coordinate frame, **b–f** frame a after various rotations, in units of radians, as indicated.

https://sn.pub/j7PTLU



In three dimensions, rotation is not commutative – the result depends on the order in which rotations are applied.



- Many ways to represent rotation:
  - rotation matrices;
  - Euler and Cardan angles;
  - rotation axis and angle;
  - exponential coordinates;
  - unit quaternions.

1.3.1.1 3D Rotation Matrix

- We can represent the orientation of a coordinate frame by its basis vectors expressed in terms of the reference coordinate frame.
- Each basis vector has three elements and they form the columns of a 3x3 orthogonal matrix ${}^{A}\mathbf{R}_{B}$

$$\begin{pmatrix} {}^{A}x \\ {}^{A}y \\ {}^{A}z \end{pmatrix} = {}^{A}\mathbf{R}_{B} \begin{pmatrix} {}^{B}x \\ {}^{B}y \\ {}^{B}z \end{pmatrix}$$

which transforms a coordinate vector defined with respect to frame {B} to a coordinate vector with respect to frame {A}.

- A 3-dimensional rotation matrix R has the same special properties as its 2D counterpart:
  - The columns are the basis vectors that define the axes of the rotated 3D coordinate frame, and therefore have unit length and are orthogonal.
  - It is orthogonal (also called orthonormal) ▶ and therefore its inverse is the same as its transpose, that is, $\mathbf{R}^{-1} = \mathbf{R}^\top$.
  - The matrix-vector product $\mathbf{R}\boldsymbol{v}$ preserves the length and relative orientation of vectors $\boldsymbol{v}$ and therefore its determinant is $+1$.
  - It is a member of the Special Orthogonal (SO) group of dimension 3 which we write as $\mathbf{R} \in \mathbf{SO}(3) \subset \mathbb{R}^{3\times3}$. Being a group under the operation of matrix multiplication means that the product of any two matrices belongs to the group, as does its inverse.

- The rotation matrices that correspond to a coordinate frame rotation of about the x-, y- and z-axes are

$$\mathbf{R}_x(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix}$$

$$\mathbf{R}_y(\theta) = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix}$$

$$\mathbf{R}_z(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

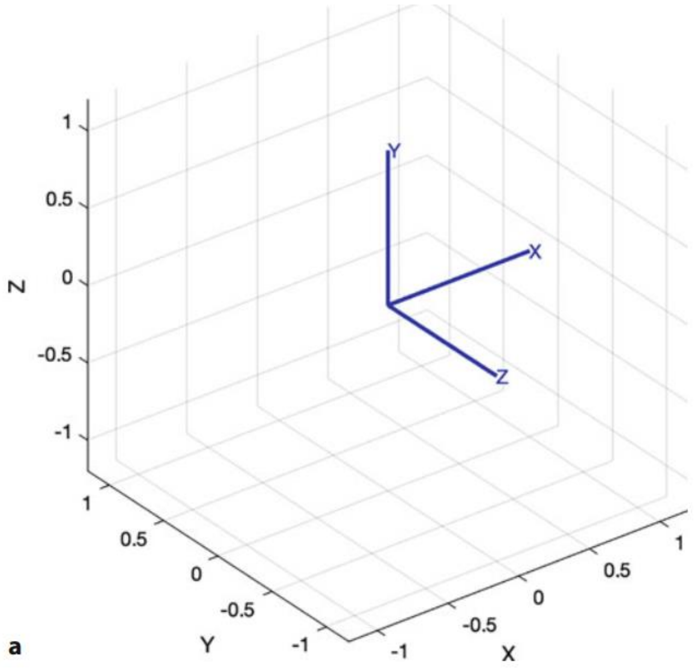
- **Example:**

```
>> R = rotmx(pi/2)
R =
     1     0     0
     0     0    -1
     0     1     0
>> plottform(R);
>> animtform(R)
```

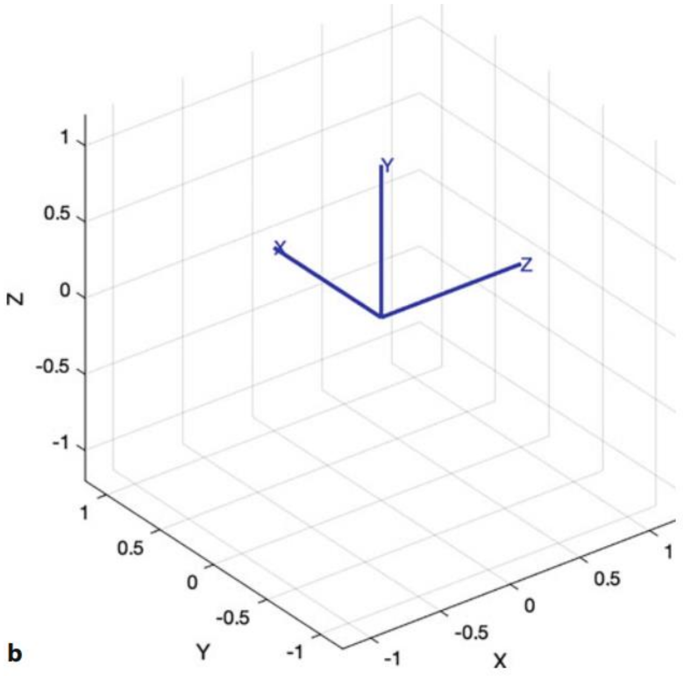
```
>> R = rotmx(pi/2)*rotmy(pi/2)
R =
     0     0     1
     1     0     0
     0     1     0
>> plottform(R)
```

```
>> rotmy(pi/2)*rotmx(pi/2)
ans =
     0     1     0
     0     0    -1
    -1     0     0
```

(The noncommutativity of rotation is clearly shown by reversing the order of the rotations)



a



b

ξ as an SO(3) Matrix

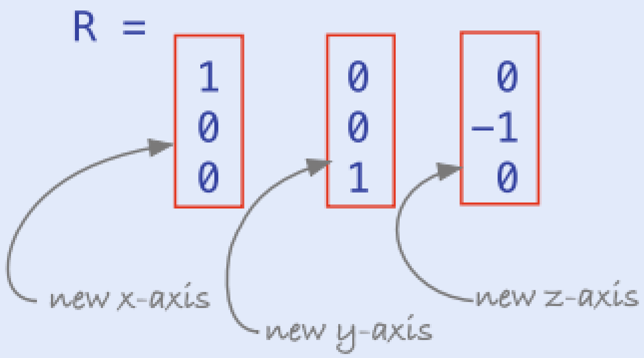
For the case of pure rotation in 3D, ξ can be implemented by a rotation matrix $\mathbf{R} \in \mathbf{SO}(3)$. The implementation is:

composition	$\boldsymbol{\xi}_1 \oplus \boldsymbol{\xi}_2$	$\mapsto \mathbf{R}_1\mathbf{R}_2$, matrix multiplication
inverse	$\ominus\boldsymbol{\xi}$	$\mapsto \mathbf{R}^{-1} = \mathbf{R}^\top$, matrix transpose
identity	$\varnothing$	$\mapsto \mathbf{1}_{3\times3}$, identity matrix
vector-transform	$\boldsymbol{\xi} \cdot \boldsymbol{v}$	$\mapsto \mathbf{R}\boldsymbol{v}$, matrix-vector product

Composition is not commutative, that is, $\mathbf{R}_1\mathbf{R}_2 \neq \mathbf{R}_2\mathbf{R}_1$.

Excurse: Reading a Rotation Matrix

The columns from left to right tell us the directions of the new frame's axes in terms of the current coordinate frame.



In this case, the new frame has its x-axis in the old x-direction $(1, 0, 0)$, its y-axis in the old z-direction $(0, 0, 1)$, and the new z-axis in the old negative y-direction $(0, -1, 0)$. The x-axis was unchanged, since this is the axis around which the rotation occurred. The rows are the converse and represent the current frame axes in terms of the new frame's axes.

1.3.1.2 Three-Angle Representations

- Euler's rotation theorem: *Any two independent orthonormal coordinate frames can be related by a sequence of rotations (not more than three) about coordinate axes, where no two successive rotations may be about the same axis.* (Kuipers, 1999).
- There are a total of twelve unique rotation sequences:

  - XYX, XZX, YXY, YZY, ZXZ, or ZYZ.

  - XYZ, XZY, YZX, YXZ, ZXY, or ZYX.
- In mechanical dynamics, the ZYZ sequence is commonly used

$$\mathbf{R}(\phi, \theta, \psi) = \mathbf{R}_z(\phi)\,\mathbf{R}_y(\theta)\,\mathbf{R}_z(\psi)$$

and the Euler angles are written as the 3-vector $\Gamma = (\phi, \theta, \psi) \in (S^1)^3$.

- **Example:**

```
>> R = rotmz(0.1)*rotmy(-0.2)*rotmz(0.3)
R =
     0.9021   -0.3836   -0.1977
     0.3875    0.9216   -0.0198
     0.1898   -0.0587    0.9801

>> R = eul2rotm([0.1 -0.2 0.3],"ZYZ")
R =
     0.9021   -0.3836   -0.1977
     0.3875    0.9216   -0.0198
     0.1898   -0.0587    0.9801

>> gamma = rotm2eul(R,"ZYZ")
gamma =
     0.1000   -0.2000    0.3000
```

```
>> R = eul2rotm([0.1 0.2 0.3],"ZYZ")
R =
     0.9021   -0.3836    0.1977
     0.3875    0.9216    0.0198
    -0.1898    0.0587    0.9801

>> gamma = rotm2eul(R,"ZYZ")
gamma =
    -3.0416   -0.2000   -2.8416

>> eul2rotm(gamma,"ZYZ")
ans =
     0.9021   -0.3836    0.1977
     0.3875    0.9216    0.0198
    -0.1898    0.0587    0.9801
```

The mapping from a rotation matrix to ZYZ-Euler angles is not unique.

Excurse: Leonhard Euler

Euler (1707–1783) was a Swiss mathematician and physicist who dominated eighteenth century mathematics. He was a student of Johann Bernoulli and applied new mathematical techniques such as calculus to many problems in mechanics and optics. He developed the functional notation, $y = f(x)$, and in robotics we use his rotation theorem and his equations of motion in rotational dynamics.

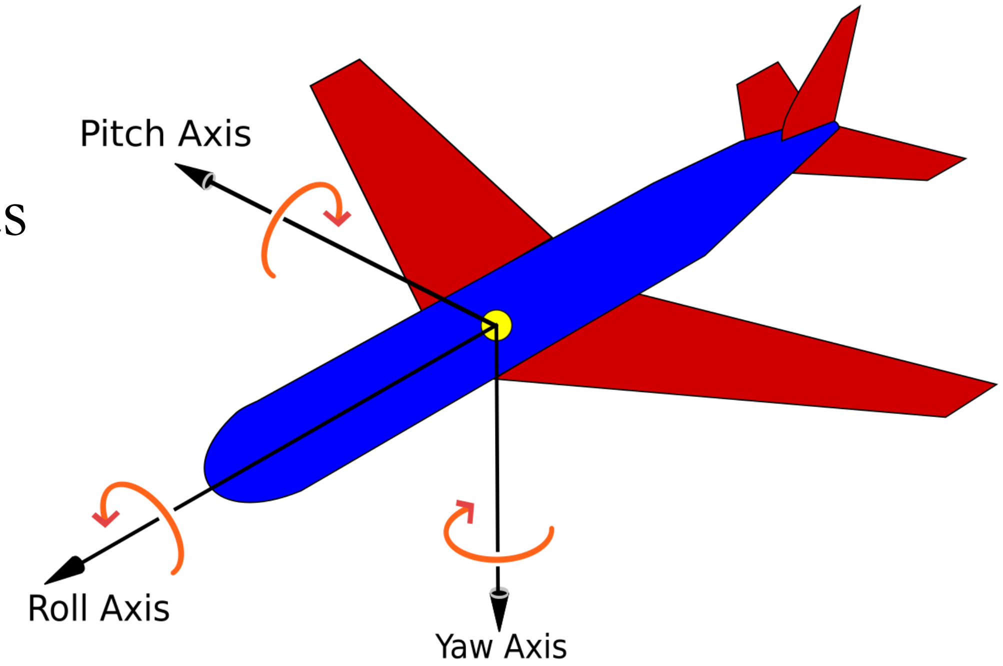
He was prolific and his collected works fill 75 volumes. Almost half of this was produced during the last seventeen years of his life when he was completely blind.

- The other widely used convention are the Cardan angles: roll, pitch and yaw which we denote as α, β and γ respectively.
- When describing the attitude of vehicles such as ships, aircraft and cars, the convention is that the x-axis of the body frame points in the forward direction and its z-axis points either up or down.
- The ZYX angle sequence:

$$\mathbf{R}(\alpha, \beta, \gamma) = \mathbf{R}_z(\gamma)\,\mathbf{R}_y(\beta)\,\mathbf{R}_x(\alpha)$$

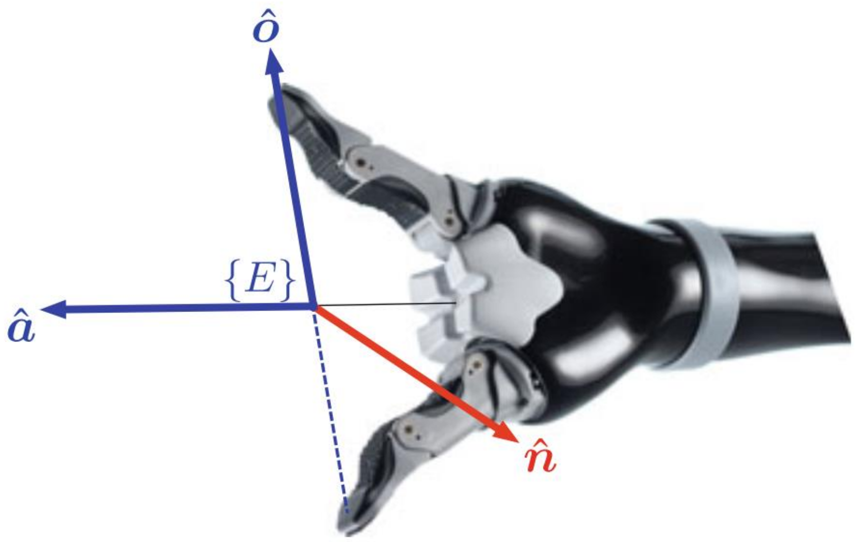
- The roll, pitch and yaw angles are written as

the 3-vector $\Gamma = (\gamma, \beta, \alpha) \in (S^1)^3$.




```
>> R = eul2rotm([0.3 0.2 0.1],"ZYX")
R =
    0.9363   -0.2751    0.2184
    0.2896    0.9564   -0.0370
   -0.1987    0.0978    0.9752
>> gamma = rotm2eul(R,"ZYX")
gamma =
    0.3000    0.2000    0.1000
```

- When describing the orientation of a robot gripper the convention is that its coordinate frame has the z-axis pointing forward and the y-axis is parallel to a line between the finger tips.

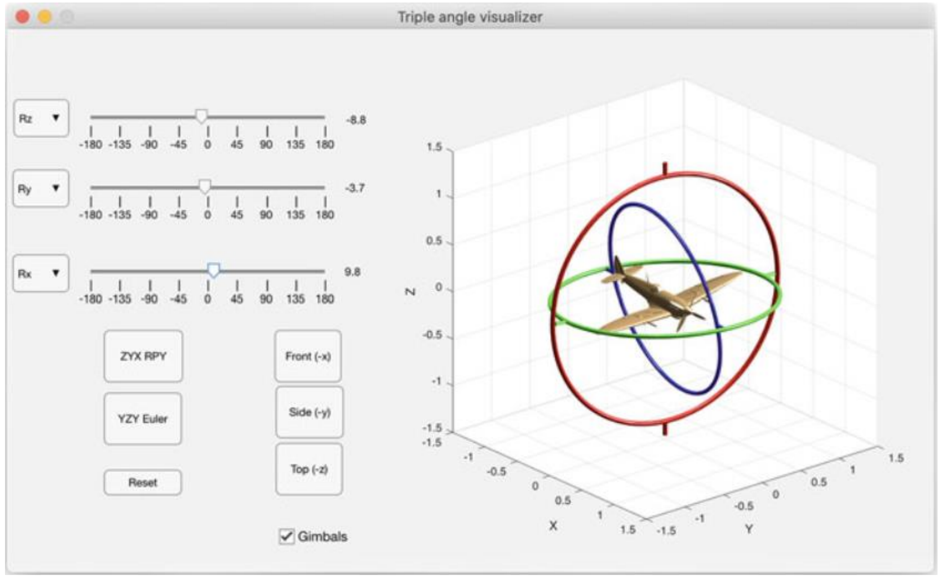


- The XYZ angle sequence:

$$\mathbf{R}(\alpha, \beta, \gamma) = \mathbf{R}_x(\gamma)\,\mathbf{R}_y(\beta)\,\mathbf{R}_z(\alpha)$$

```
>> R = eul2rotm([0.3 0.2 0.1],"XYZ")
R =
     0.9752   -0.0978    0.1987
     0.1538    0.9447   -0.2896
    -0.1593    0.3130    0.9363
>> gamma = rotm2eul(R,"XYZ")
gamma =
     0.3000    0.2000    0.1000

>> tripleangleApp
```



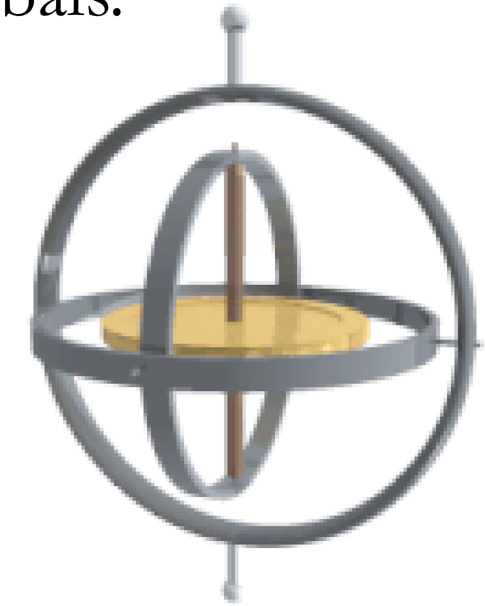
Excurse: Gerolamo Cardano

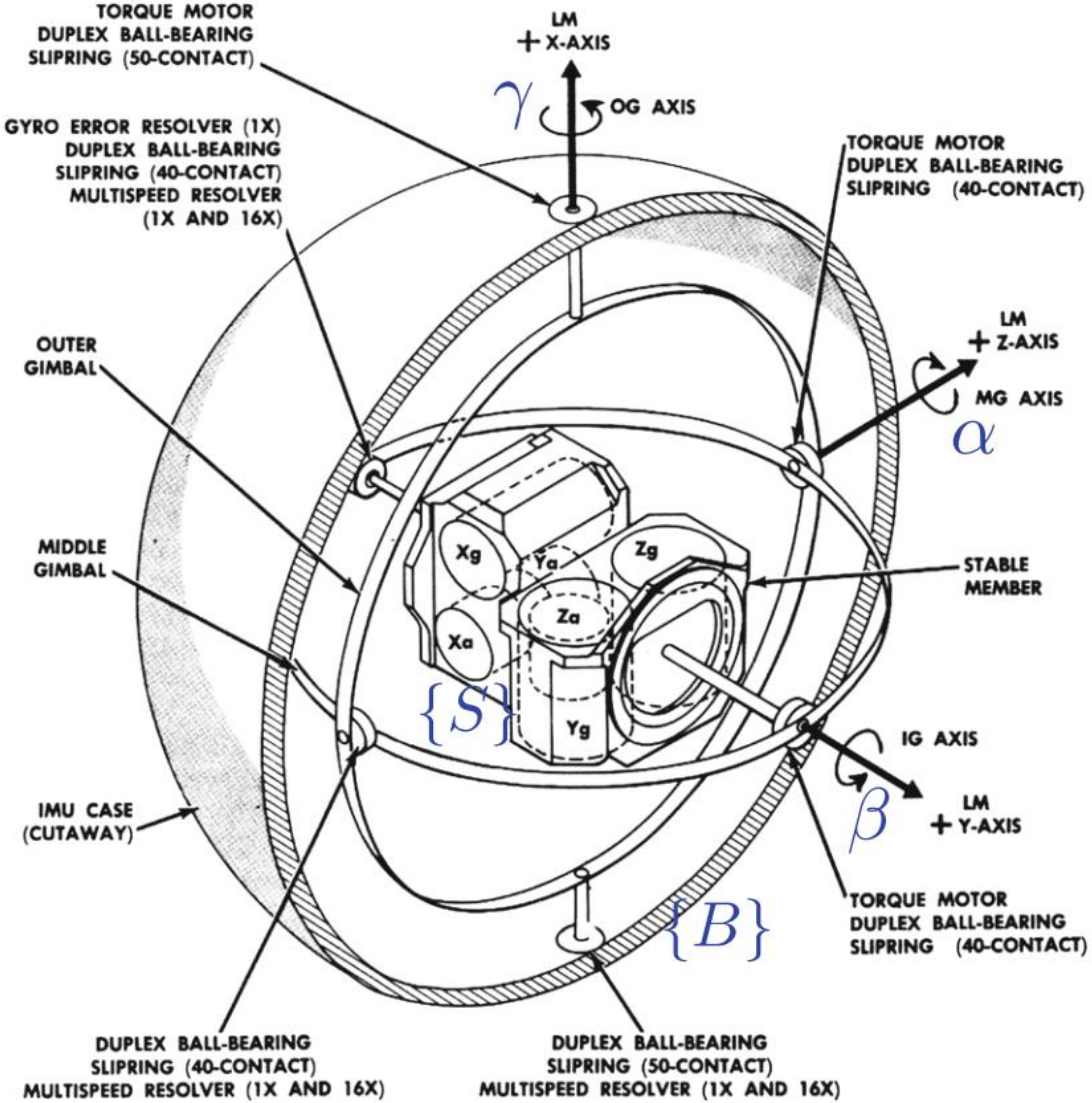
Cardano (1501–1576) was an Italian Renaissance mathematician, physician, astrologer, and gambler. He was born in Pavia, Italy, the illegitimate child of a mathematically gifted lawyer. He studied medicine at the University of Padua and later was the first to describe typhoid fever. He partly supported himself through gambling and his book about games of chance *Liber de ludo aleae* contains the first systematic treatment of probability as well as effective cheating methods. His family life was problematic: his eldest son was executed for poisoning his wife, and his daughter was a prostitute who died from syphilis (about which he wrote a treatise). He computed and published the horoscope of Jesus, was accused of heresy, and spent time in prison until he abjured and gave up his professorship.

He published the solutions to the cubic and quartic equations in his book *Ars magna* in 1545, and also invented the combination lock, the gimbal consisting of three concentric rings allowing a compass or gyroscope to rotate freely (see Fig. 2.19), and the Cardan shaft with universal joints – the drive shaft used in motor vehicles today.

1.3.1.3 Singularities and Gimbal Lock

- Structure of a mechanical gyroscope:
  - Stable member hold by three orthogonal gyroscope;
  - Gimbals.





- In this design, the gimbals form a Cardanian YZX sequence:

$$^{S}\mathbf{R}_B = \mathbf{R}_y(\beta)\boxed{\mathbf{R}_z(\alpha)\mathbf{R}_x(\gamma)}$$

- When the rotation angle α is 90^o, the axes of the inner (β) and the outer (γ) gimbals are aligned and they share the same rotation axis. This is known as gimbal lock.
- There are now only two effective rotational axes:

$$\mathbf{R}_y(\theta)\,\mathbf{R}_z\left(\tfrac{\pi}{2}\right) \equiv \mathbf{R}_z\left(\tfrac{\pi}{2}\right)\mathbf{R}_x(\theta)$$

$$^{S}\mathbf{R}_B = \mathbf{R}_z\left(\tfrac{\pi}{2}\right)\mathbf{R}_x(\beta)\,\mathbf{R}_x(\gamma) = \mathbf{R}_z\left(\tfrac{\pi}{2}\right)\mathbf{R}_x(\beta+\gamma)$$

- The loss of a degree of freedom means that mathematically we cannot invert the transformation. The best we can do is carefully choose the angle sequence and coordinate system.
- Singularities are a consequence of using just three parameters to represent orientation. To eliminate this problem, we need to adopt different representations of orientation.

Excurse: Apollo 13

Apollo 13 mission clock: 02 08 12 47

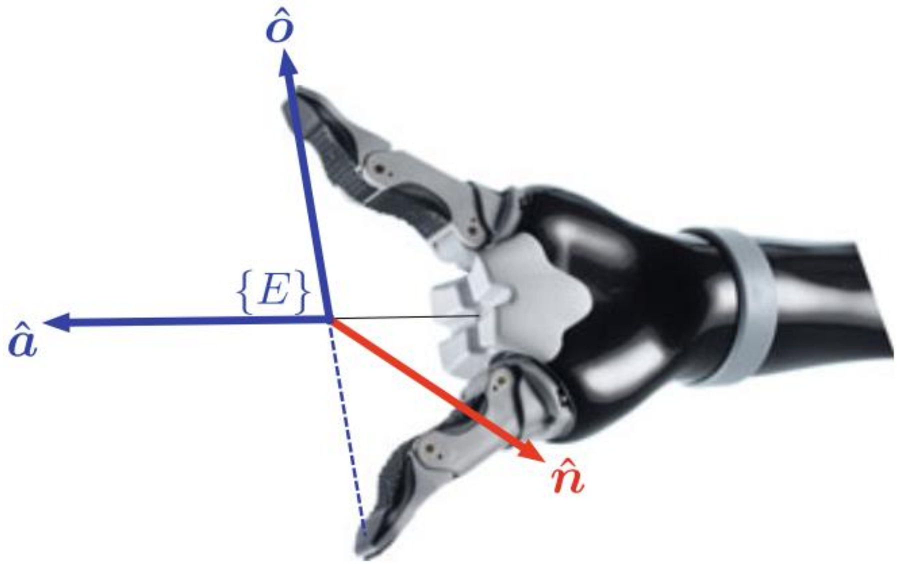
- **Flight:** "Go, Guidance."
- **Guido:** "He's getting close to gimbal lock there."
- **Flight:** "Roger. CapCom, recommend he bring up C3, C4, B3, B4, C1 and C2 thrusters, and advise he's getting close to gimbal lock."
- **CapCom:** "Roger."

Apollo 13, mission control communications loop (1970) (Lovell and Kluger 1994, p 131; NASA 1970).

1.3.1.4 Two-Vector Representation

- For arm-type robots, it is useful to consider a coordinate frame {E} attached to the end effector.
  - approach vector: $\hat{\boldsymbol{a}} = (a_x, a_y, a_z)$;
  - orientation vector: $\hat{\boldsymbol{o}} = (o_x, o_y, o_z)$;
  - normal vector: $\hat{\boldsymbol{n}} = \hat{\boldsymbol{o}} \times \hat{\boldsymbol{a}}$.

$$\mathbf{R} = \begin{pmatrix} n_x & o_x & a_x \\ n_y & o_y & a_y \\ n_z & o_z & a_z \end{pmatrix}$$




```
>> a = [0 0 -1];

>> o = [1 1 0];

>> R = oa2rotm(o,a)
R =
   -0.7071    0.7071         0
    0.7071    0.7071         0
         0         0   -1.0000
```

1.3.1.5 Rotation About an Arbitrary Vector

- Two coordinate frames of arbitrary orientation are related by a single rotation about some axis in space.
- For the example rotation used earlier:

```
>> R = eul2rotm([0.1 0.2 0.3]);
```

we can determine such an angle and axis by

```
>> rotm2axang(R)
ans =
    0.7900    0.5834    0.1886    0.3655
```

where the first three elements are a unit vector $\hat{\boldsymbol{v}}$ the first three elements are a unit vector $\boldsymbol{\theta}$ about that axis.

- Note that this transformation is not unique.

1.3.1.6 Matrix Exponential for Rotation

- Consider an x-axis rotation expressed as a rotation matrix

```
>> R = rotmx(0.3)
R =
    1.0000         0         0
         0    0.9553   -0.2955
         0    0.2955    0.9553
```

- We can take the logarithm using the `logm`

```
>> L = logm(R)
L =
         0         0         0
         0         0   -0.3000
         0    0.3000         0
```

and the result is a sparse matrix with two elements that have a magnitude of 0.3, which is the rotation angle. This is a skew-symmetric matrix.

- Unpacking the skew-symmetric matrix gives

```
>> S = skew2vec(L)
S =
    0.3000         0         0
```

Excurse: 3D Skew-Symmetric Matrix

In three dimensions, the skew- or anti-symmetric matrix has the form

$$[\boldsymbol{\omega}]_\times = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix} \in \mathbf{so}(3) \tag{2.28}$$

which has a distinct structure with a zero diagonal and only three unique elements $\boldsymbol{\omega} \in \mathbb{R}^3$, and $[\boldsymbol{\omega}]_\times^\top = -[\boldsymbol{\omega}]_\times$. The vector space of 3D skew-symmetric matrices is denoted $\mathbf{so}(3)$ and is the Lie algebra of $\mathbf{SO}(3)$. The vector cross product can be written as a matrix-vector product $\boldsymbol{v}_1 \times \boldsymbol{v}_2 = [\boldsymbol{v}_1]_\times \boldsymbol{v}_2$. The $[\cdot]_\times$ operator is implemented by

```
>> X = vec2skew([1 2 3])
X =
     0    -3     2
     3     0    -1
    -2     1     0
```

and the inverse operation, the $\vee_\times(\cdot)$ operator by

```
>> skew2vec(X)
ans =
     1     2     3
```

Both functions work for the 3D case, shown here, and the 2D case where the vector is a 1-vector.

- Exponentiating the logarithm of the rotation matrix using the matrix exponential function yields the original rotation matrix:

```
>> expm(L)
ans =
    1.0000         0         0
         0    0.9553   -0.2955
         0    0.2955    0.9553
```

- In general, we can write:

$$\mathbf{R} = e^{\theta[\hat{\omega}]_\times} \in \mathbf{SO}(3)$$

where θ is the rotation angle, $\hat{\boldsymbol{\omega}}$ is a unit vector parallel to the rotation axis, and the notation $[\cdot]_\times : \mathbb{R}^3 \longmapsto \mathbf{so}(3) \subset \mathbb{R}^{3\times3}$ indicates a mapping from a vector to a skewsymmetric matrix.

1.3.1.7 Unit Quaternions

- A quaternion is a hypercomplex number which has a real part and three complex parts

$$\breve{q} = s + ui + vj + wk \in \mathbb{H}$$

where the orthogonal complex numbers *i*, *j* , and *k* are defined such that

$$i^2 = j^2 = k^2 = ijk = -1$$

- It is more common to consider a quaternion as an ordered pair $\breve{q} = (s, \boldsymbol{v})$, where $s \in \mathbb{R}$ is the scalar part and $\boldsymbol{v} = (u, v, w) \in \mathbb{R}^3$ is the vector part.
- Quatenions form a vector space and therefore allow operations such as addition and subtraction, performed element-wise, and multiplication by a scalar.

- Quaternions also allow conjugation

$$\breve{q}^* = s - ui - vj - wk \in \mathbb{H}$$

- Quternion multiplication:

$$\breve{q}_q \circ \breve{q}_2 = \underbrace{(s_1 s_2 - \boldsymbol{v}_1 \cdot \boldsymbol{v}_2)}_{\text{real}} + \underbrace{(s_1 \boldsymbol{v}_2 + s_2 \boldsymbol{v}_1 + \boldsymbol{v}_1 \times \boldsymbol{v}_2)}_{\text{vector}} \in \mathbb{H}$$

- The inner product is a scalar

$$\breve{q}_1 \cdot \breve{q}_2 = s_1 s_2 + u_1 u_2 + v_1 v_2 + w_1 w_2 \in \mathbb{R}$$

- The magnitude of a quaternion:

$$\|\breve{q}\| = \sqrt{\breve{q} \cdot \breve{q}} = \sqrt{s^2 + u^2 + v^2 + w^2} \in \mathbb{R}$$

- A pure quaternion is one whose scalar part is zero.

- To represent rotations, we use unit quaternions which has $\|\breve{q}\| = 1$ and is denoted by $\mathring{q}$. They can be considered as a rotation of θ about the unit vector $\hat{\boldsymbol{v}}$.

$$\mathring{q} = \underbrace{\cos\frac{\theta}{2}}_{\text{real}} + \underbrace{\sin\frac{\theta}{2}(\hat{v}_x i + \hat{v}_y j + \hat{v}_z k)}_{\text{vector}} \in S^3$$

```
>> q = quaternion(rotmx(0.3),"rotmat","point")
q =
  quaternion
     0.98877 + 0.14944i +       0j +       0k
```

- It is possible to store only the vector part of a unit quaternion, since the scalar part can be recovered by $s = \pm(1 - u^2 - v^2 - w^2)^{1/2}$.
- This 3-vector form of a unit quaternion is frequently used for optimization problems such as posegraph relaxation or bundle adjustment – it has minimum dimensionality and is singularity free.

ξ as a Unit Quaternion

For the case of pure rotation in 3D, $\boldsymbol{\xi}$ can be implemented with a unit quaternion $\mathring{q} \in S^3$. The implementation is:

composition	$\boldsymbol{\xi}_1 \oplus \boldsymbol{\xi}_2$	$\mapsto \mathring{q}_1 \circ \mathring{q}_2$, Hamilton product
inverse	$\ominus\boldsymbol{\xi}$	$\mapsto \mathring{q}^* = s - \boldsymbol{v}$, quaternion conjugation
identity	$\varnothing$	$\mapsto 1 + \mathbf{0}$
vector-transform	$\boldsymbol{\xi} \cdot \boldsymbol{v}$	$\mapsto \mathring{q} \circ \breve{v} \circ \mathring{q}^*$, where $\breve{v} = 0 + \boldsymbol{v}$ is a pure quaternion

Composition is not commutative, that is, $\mathring{q}_1 \circ \mathring{q}_2 \neq \mathring{q}_2 \circ \mathring{q}_1$.

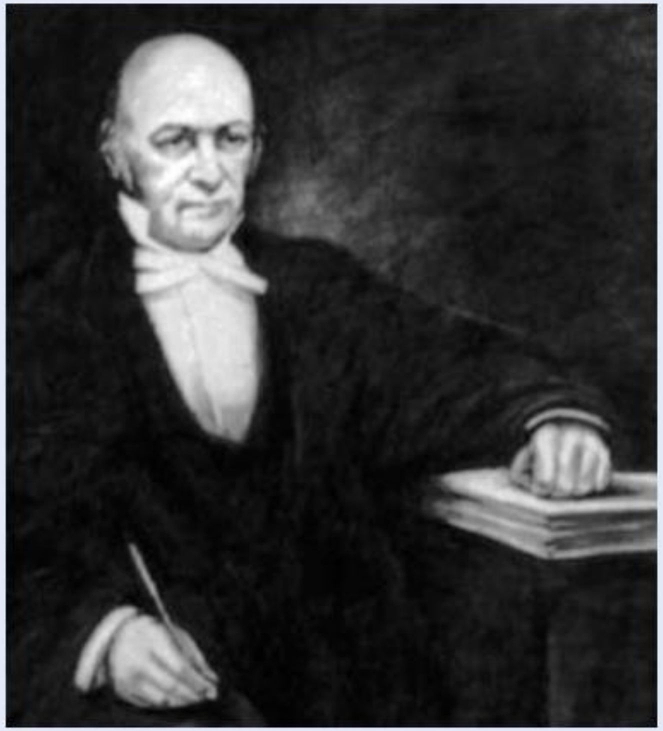
Excurse: Sir William Rowan Hamilton

Hamilton (1805–1865) was an Irish mathematician, physicist, and astronomer. He was a child prodigy with a gift for languages and by age thirteen knew classical and modern European languages as well as Persian, Arabic, Hindustani, Sanskrit, and Malay. Hamilton taught himself mathematics at age 17, and discovered an error in Laplace's Celestial Mechanics. He spent his life at Trinity College, Dublin, and was appointed Professor of Astronomy and Royal Astronomer of Ireland while still an undergraduate. In addition to quaternions, he contributed to the development of optics, dynamics, and algebra. He also wrote poetry and corresponded with Wordsworth who advised him to devote his energy to mathematics.

According to legend, the key quaternion equation, (2.30), occurred to Hamilton in 1843 while walking along the Royal Canal in Dublin with his wife, and this is commemorated by a plaque on Broome bridge:

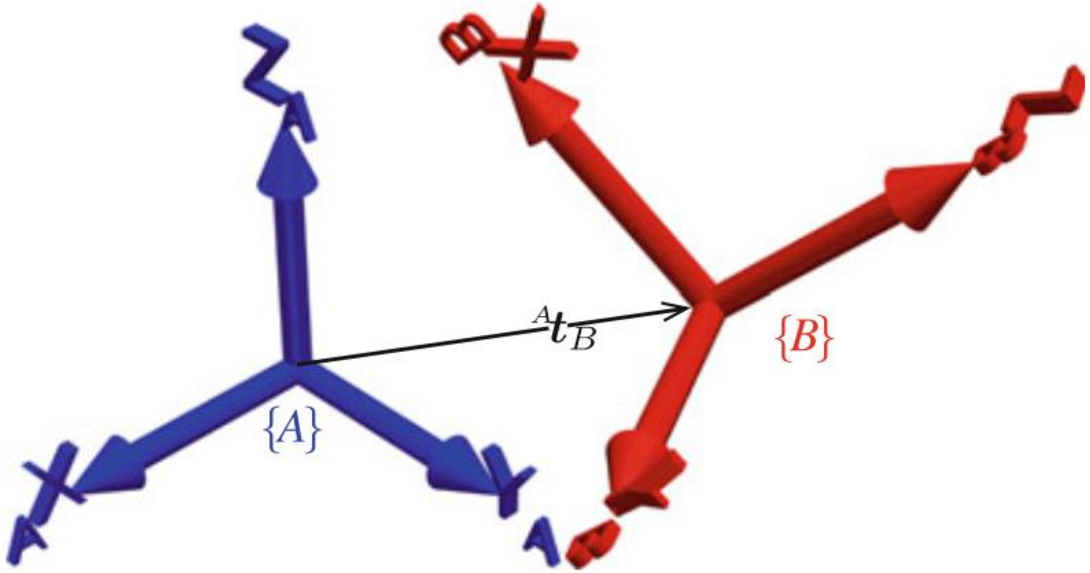
» Here as he walked by on the 16th of October 1843 Sir William Rowan Hamilton in a flash of genius discovered the fundamental formula for quaternion multiplication $i^2 = j^2 = k^2 = ijk = -1$ & cut it on a stone of this bridge.

His original carving is no longer visible, but the bridge is a pilgrimage site for mathematicians and physicists.



1.3.2 Pose in Three Dimensions

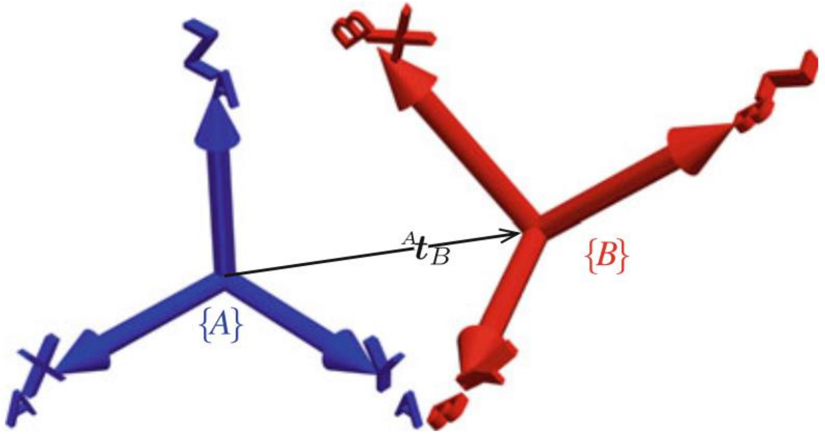
- To describe the relative pose of the frames, we need to account for the translation between the origins of the frames, as well as the rotation.
- We will combine particular methods of representing rotation with representations of translation, to create tangible representations of relative pose in 3D.



1.3.2.1 Homogeneous Transformation Matrix

- The 3D homogeneous transformation matrix

$$\begin{pmatrix} {}^{A}x \\ {}^{A}y \\ {}^{A}z \\ 1 \end{pmatrix} = \begin{pmatrix} {}^{A}\mathbf{R}_B & {}^{A}\boldsymbol{t}_B \\ \mathbf{0}_{1\times 3} & 1 \end{pmatrix} \begin{pmatrix} {}^{B}x \\ {}^{B}y \\ {}^{B}z \\ 1 \end{pmatrix}$$



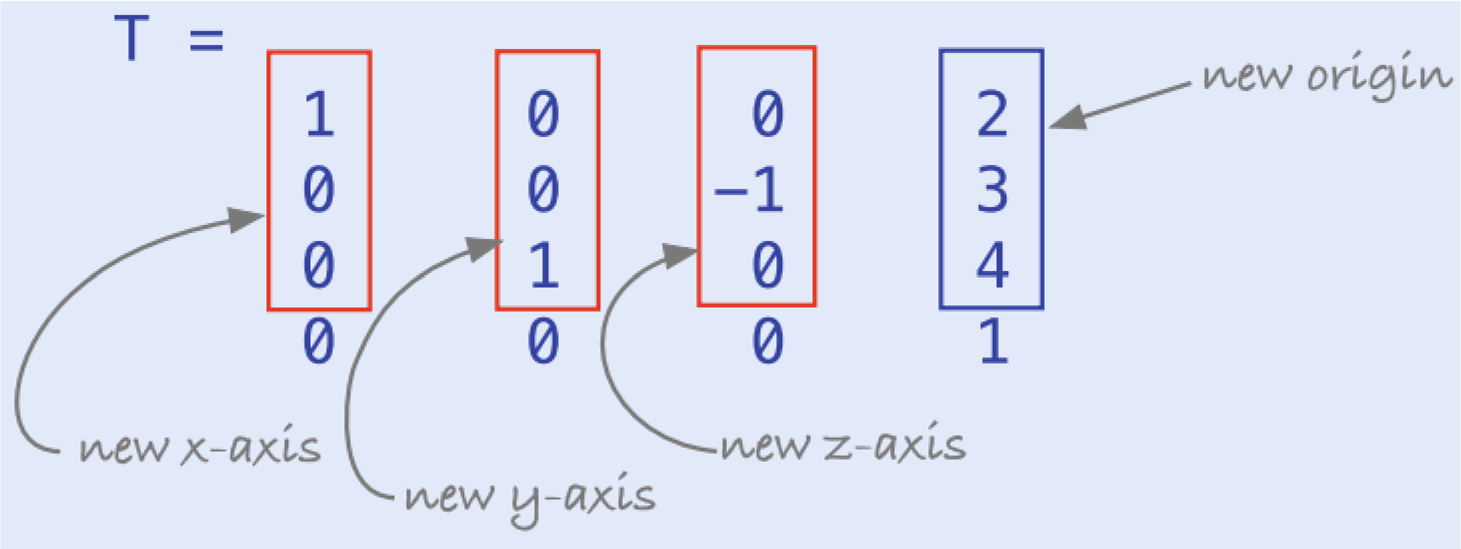
where ${}^{A}\boldsymbol{t}_B \in \mathbb{R}^3$ - translation vector; and ${}^{A}\boldsymbol{R}_B \in \boldsymbol{SO}(3)$ – rotation matrix.

- If points arerepresented by homogeneous coordinate vectors, then

$$\begin{aligned} {}^{A}\tilde{\boldsymbol{p}} &= \begin{pmatrix} {}^{A}\mathbf{R}_B & {}^{A}\boldsymbol{t}_B \\ \mathbf{0}_{1\times 3} & 1 \end{pmatrix} {}^{B}\tilde{\boldsymbol{p}} \\ &= {}^{A}\mathbf{T}_B \, {}^{B}\tilde{\boldsymbol{p}} \end{aligned}$$

where ${}^{A}\boldsymbol{T}_B \in \mathbf{SE}(3) \subset \mathbb{R}^{4\times 4}$ is a homogeneous transformation matrix.

- A homogeneous transformation matrix is a representation of a relative pose or a motion. It defines the new coordinate frame in terms of the previous coordinate frame and we can easily read it as follows




```
>> T = trvec2tform([2 0 0])*tformrx(pi/2)*trvec2tform([0 1 0])
T =
     1     0     0     2
     0     0    -1     0
     0     1     0     1
     0     0     0     1
```

ξ as an $\mathbf{SE}(3)$ Matrix

For the case of rotation and translation in 3D, $\boldsymbol{\xi}$ can be implemented by a homogeneous transformation matrix $\mathbf{T} \in \mathbf{SE}(3)$ which is sometimes written as an ordered pair $(\mathbf{R}, \boldsymbol{t}) \in \mathbf{SO}(3) \times \mathbb{R}^3$. The implementation is:

composition	$\boldsymbol{\xi}_1 \oplus \boldsymbol{\xi}_2$	$\mapsto \mathbf{T}_1\mathbf{T}_2 = \begin{pmatrix} \mathbf{R}_1\mathbf{R}_2 & \boldsymbol{t}_1 + \mathbf{R}_1\boldsymbol{t}_2 \\ \mathbf{0}_{1\times3} & 1 \end{pmatrix}$, matrix multiplication
inverse	$\ominus\boldsymbol{\xi}$	$\mapsto \mathbf{T}^{-1} = \begin{pmatrix} \mathbf{R}^\top & -\mathbf{R}^\top\boldsymbol{t} \\ \mathbf{0}_{1\times3} & 1 \end{pmatrix}$, matrix inverse
identity	$\varnothing$	$\mapsto \mathbf{1}_{4\times4}$, identity matrix
vector-transform	$\boldsymbol{\xi} \cdot \boldsymbol{v}$	$\mapsto \epsilon(\mathbf{T}\tilde{\boldsymbol{v}})$, matrix-vector product using homogeneous vectors

where $\tilde{\cdot} : \mathbb{R}^3 \mapsto \mathbb{P}^3$ and $\epsilon(\cdot) : \mathbb{P}^3 \mapsto \mathbb{R}^3$. Composition is not commutative, that is, $\mathbf{T}_1\mathbf{T}_2 \neq \mathbf{T}_2\mathbf{T}_1$.

1.3.2.2 Matrix Exponential for Pose

- Consider the **SE**(3) matrix:

```
>> T = trvec2tform([2 3 4])*tformrx(0.3)
T =
     1.0000         0         0    2.0000
          0    0.9553   -0.2955    3.0000
          0    0.2955    0.9553    4.0000
          0         0         0    1.0000
```

and its logarithm

```
>> L = logm(T)
L =
          0         0         0    2.0000
          0         0   -0.3000    3.5775
          0    0.3000         0    3.5200
          0         0         0         0
```

is a sparse matrix and the rotation magnitude of 0.3 is clearly evident.

- The structure of this matrix:
  - upper-left corner: a 3x3 skew-symmetric matrix;
  - upper right column: a 3-vector;
  - bottom row is zero.
- We can unpack the six unique elements by

```
>> S = skewa2vec(L)

S =
    0.3000   0   0   2.0000   3.5775   3.5200
```

Rotation about x-axis

- Exponentiating the logarithm matrix yields the original SE(3) matrix, and the logarithm can be reconstructed from just the six elements

```
>> expm(vec2skewa(S))
ans =
    1.0000         0         0    2.0000
         0    0.9553   -0.2955    3.0000
         0    0.2955    0.9553    4.0000
         0         0         0    1.0000
```

- In general, we can write

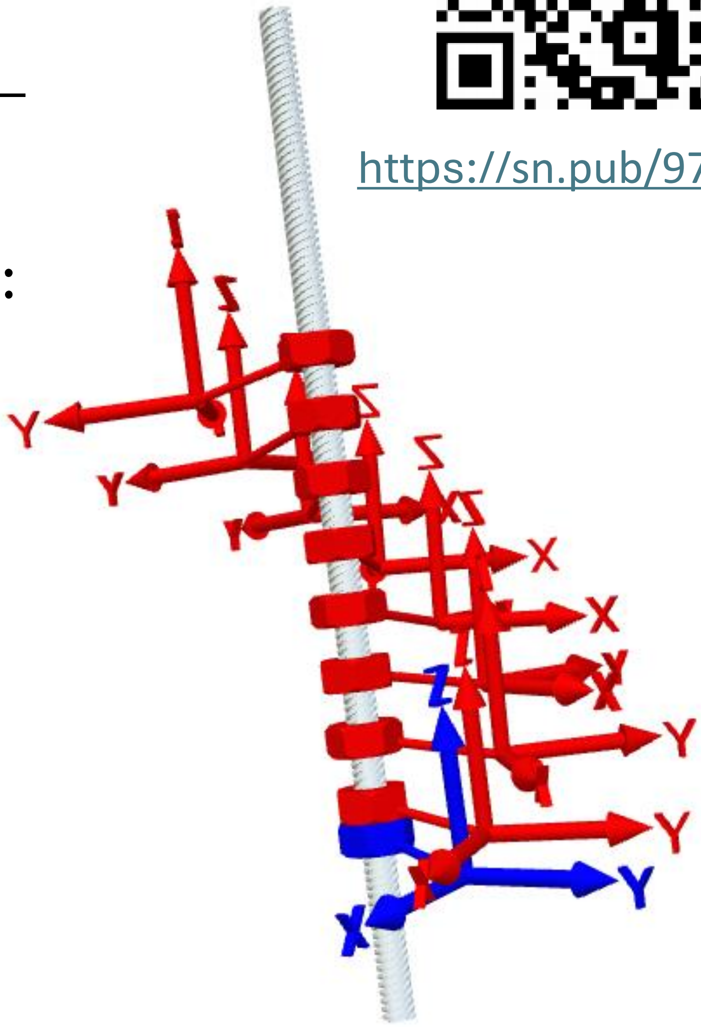
$$\mathbf{T} = e^{[\boldsymbol{S}]} \in \mathbf{SE}(3)$$

where $\boldsymbol{S} \in \mathbb{R}^6$ **and** $[\cdot]: \mathbb{R}^{\mathbf{6}} \longmapsto \mathbf{se}(3) \subset \mathbb{R}^{4\times 4}$. The vector S is a twist vector.

1.3.2.4 3D Twists

https://sn.pub/97Lsi6

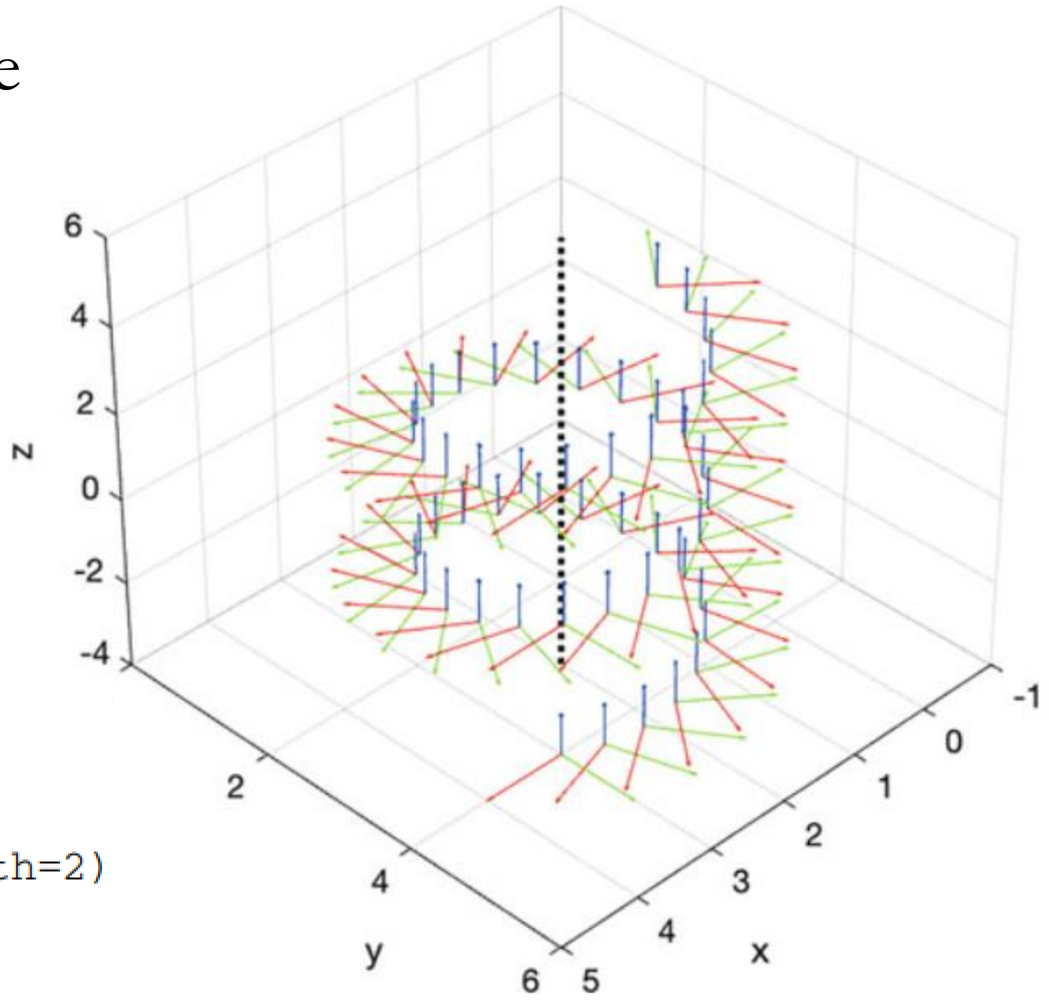
- A twist is equivalent to screw or helicoidal motion – motion about and along some line in space.
- We represent a screw as a twist vector $(\omega, v) \in \mathbb{R}^6$:
  - a vector $\boldsymbol{\omega} \in \mathbb{R}^3$: parallel to the screw axis;
  - a moment $\boldsymbol{v} \in \mathbb{R}^3$: encodes a point lying on the twist axis, as well as the screw pitch which is the ratio of the distance along the screw axis to the rotation about the screw axis.



- To illustrate the underlying screw motion, we will progressively rotate a coordinate frame about a screw. We define a screw parallel to the z-axis that passes through the point (2, 3, 2) and has a pitch of 0.5

```
>> S = Twist.UnitRevolute([0 0 1],[2 3 2],0.5);
>> X = trvec2tform([3 4 -4]);
```

- For values of θ in the range 0 to 15 rad, we evaluate the twist for each value of θ, apply it to the frame {X} and plot the result




```
>> clf; hold on
>> view(3)
>> for theta = [0:0.3:15]
>>   plottform(S.tform(theta)*X,style="rgb",LineWidth=2)
>> end
>> L = S.line();
>> L.plot("k:",LineWidth=2);
```

ξ as a 3D Twist

For the case of rotation and translation in 3D, $\boldsymbol{\xi}$ can be implemented by a twist $\boldsymbol{S} \in \mathbb{R}^6$. The implementation is:

composition	$\boldsymbol{\xi}_1 \oplus \boldsymbol{\xi}_2$	$\mapsto \log\left(e^{[\boldsymbol{S}_1]} e^{[\boldsymbol{S}_2]}\right)$, product of exponential
inverse	$\ominus \boldsymbol{\xi}$	$\mapsto -\boldsymbol{S}$, negation
identity	$\varnothing$	$\mapsto \mathbf{0}_{1\times 6}$, zero vector
vector-transform	$\boldsymbol{\xi} \cdot \boldsymbol{v}$	$\mapsto \epsilon(e^{[\boldsymbol{S}]} \tilde{\boldsymbol{v}})$, matrix-vector product using homogeneous vectors

where $\tilde{\cdot} : \mathbb{R}^3 \mapsto \mathbb{P}^3$ and $\epsilon(\cdot) : \mathbb{P}^3 \mapsto \mathbb{R}^3$. Composition is not commutative, that is, $e^{[\boldsymbol{S}_1]} e^{[\boldsymbol{S}_2]} \neq e^{[\boldsymbol{S}_2]} e^{[\boldsymbol{S}_1]}$. Note that log and exp have efficient closed form, rather than transcendental, solutions which makes composition relatively inexpensive.

1.3.2.4 Vector-Quaternion Pair

- Another way to represent pose in 3D is to combine a translation vector and a unit quaternion. It represents pose using just 7 numbers, is easy to compound, and singularity free.

$\boldsymbol{\xi}$ as a Vector-Quaternion Pair

For the case of rotation and translation in 3D, $\boldsymbol{\xi}$ can be implemented by a vector and a unit quaternion written as an ordered pair $(\boldsymbol{t}, \mathring{q}) \in \mathbb{R}^3 \times S^3$. The implementation is:

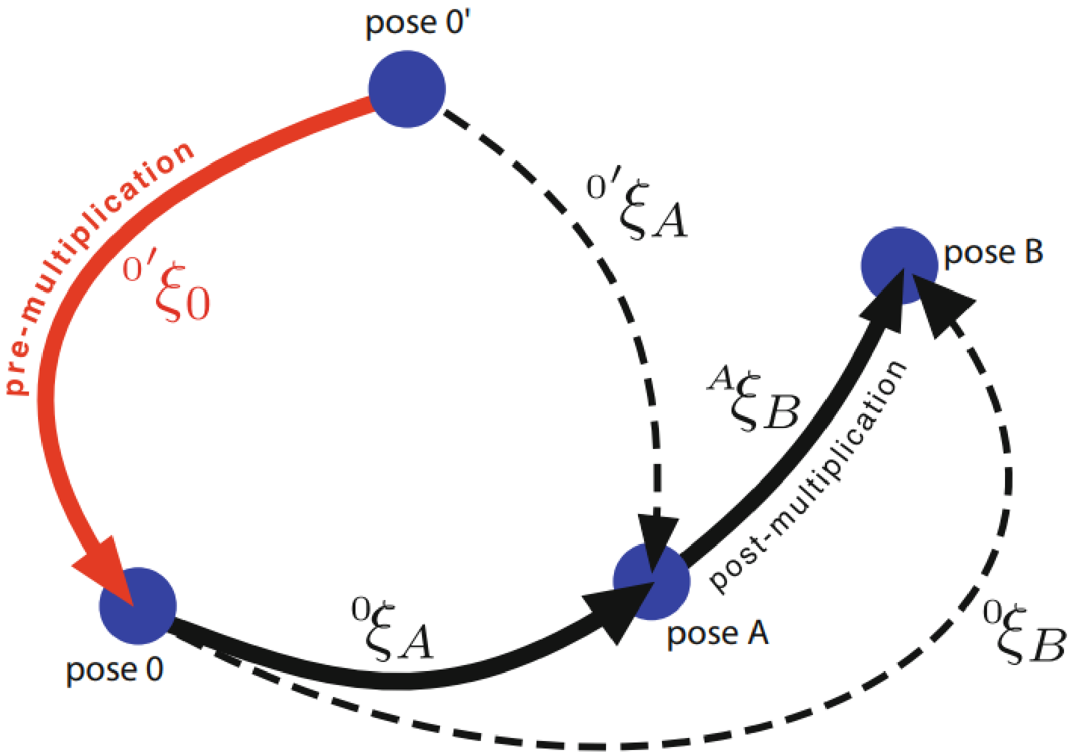
$$
\begin{array}{lll}
\text{composition} & \boldsymbol{\xi}_1 \oplus \boldsymbol{\xi}_2 & \mapsto (\boldsymbol{t}_1 + \mathring{q}_1 \cdot \boldsymbol{t}_2,\ \mathring{q}_1 \circ \mathring{q}_2) \\
\text{inverse} & \ominus \boldsymbol{\xi} & \mapsto (-\mathring{q}^* \cdot \boldsymbol{t},\ \mathring{q}^*) \\
\text{identity} & \varnothing & \mapsto (\mathbf{0}_3,\ 1 + \mathbf{0}) \\
\text{vector-transform} & \boldsymbol{\xi} \cdot \boldsymbol{v} & \mapsto \mathring{q} \cdot \boldsymbol{v} + \boldsymbol{t}
\end{array}
$$

Composition is not commutative, that is, $\boldsymbol{\xi}_1 \boldsymbol{\xi}_2 \neq \boldsymbol{\xi}_2 \boldsymbol{\xi}_1$. The vector transformation operator $\cdot : S^3 \times \mathbb{R}^3 \mapsto \mathbb{R}^3$ can be found in ► Sect. 2.3.1.7.

1.4 Advanced Topics

1.4.1 Pre- and Post-Multiplication of Transforms

- Many texts make a distinction between pre-multiplying and post-multiplying relative transformations.
- For the case of a series of rotation matrices, the former rotates with respect to the world reference frame, whereas the latter rotates with respect to the current or rotated frame.
- Post-multiplication should be familiar by now, and is the motion from {A} to {B} expressed in frame {A} – the rotated frame.

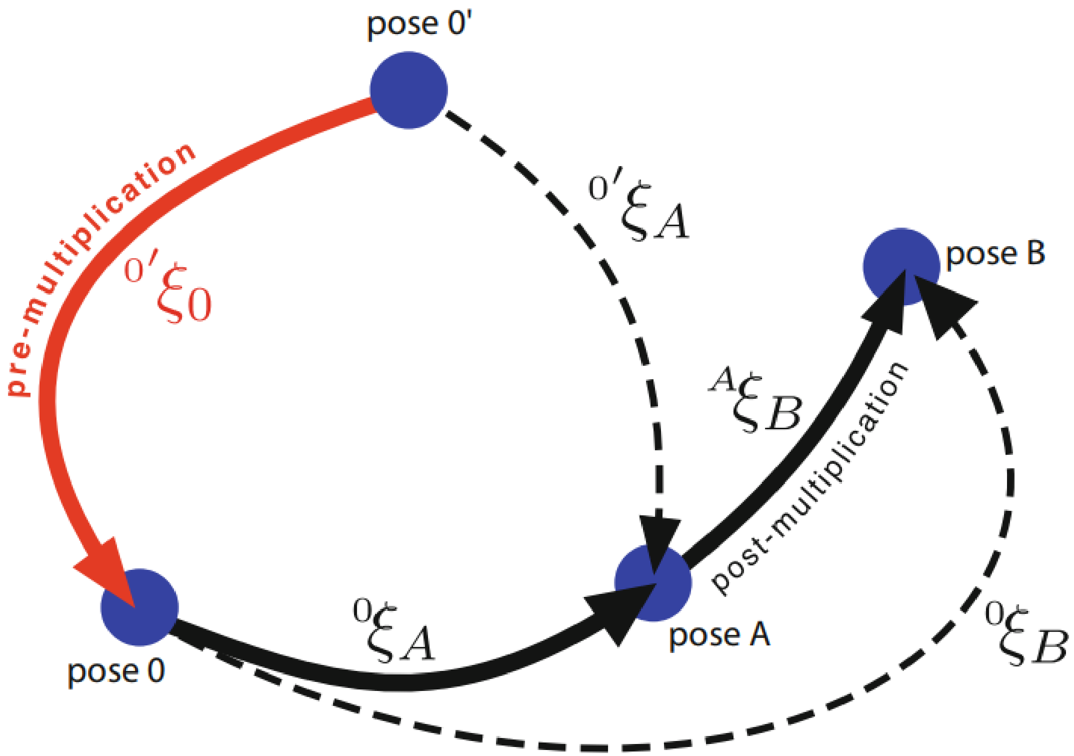


- Pre-multiplication requires that we introduce a new pose which we will call {0'}. Effectively this is a change in the reference frame – from {0} to {0'}. The transformation is therefore with respect to the reference frame, albeit a new one.
- A three-angle rotation sequence such as ZYX roll-pitch-yaw angles

$$ {}^{0}\mathbf{R}_B = \mathbf{R}_z(\gamma)\,\mathbf{R}_y(\beta)\,\mathbf{R}_x(\alpha) $$

can be interpreted in two different, but equivalent, ways:

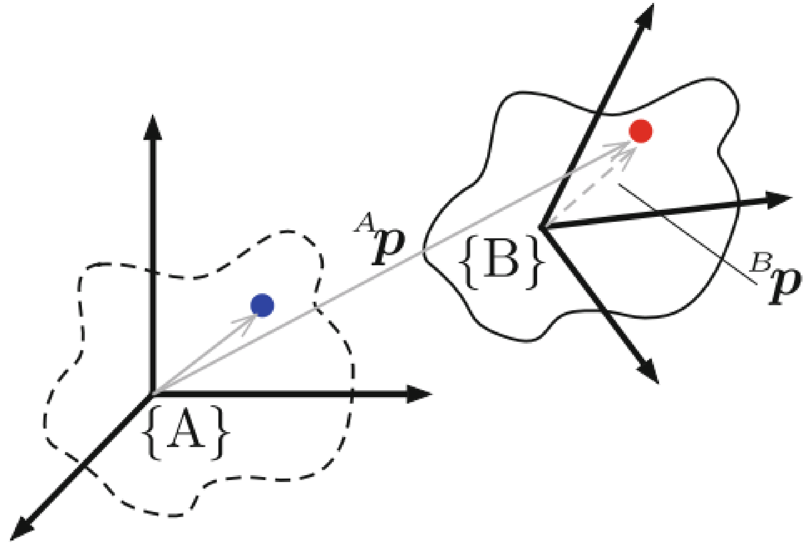
1. From right to left as consecutive rotations (extrinsic rotations).
1. From left to right as sequential rotations (intrinsic rotations).



1.4.2 Active and Passive Transformations

- Active transformation: The point has the same relative position with respect to the frame. The coordinate vector of the moved point with respect to the original coordinate frame is

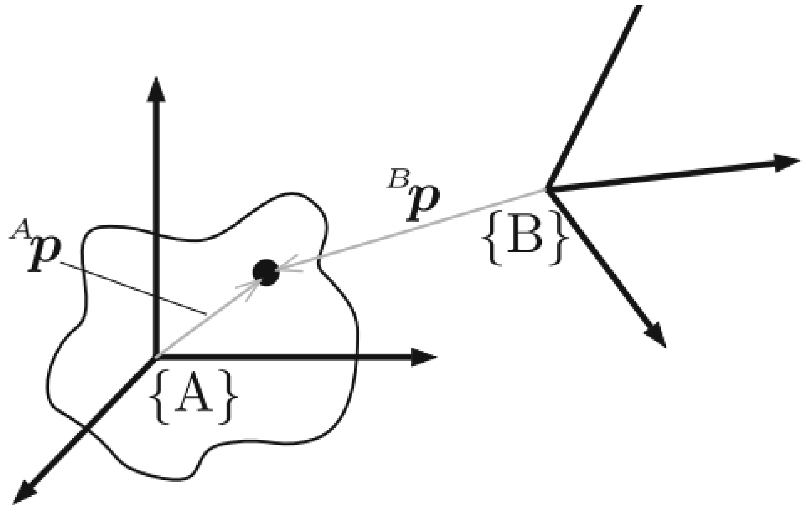
$$ {}^{A}\tilde{\boldsymbol{p}} = {}^{A}\mathbf{T}_{B}\, {}^{B}\tilde{\boldsymbol{p}} $$



a

- Passive transformation: The point does not move but the coordinate frame does. The coordinate vector of the point with respect to the moved coordinate frame is

$$ {}^{B}\tilde{\boldsymbol{p}} = {}^{B}\mathbf{T}_{A}\, {}^{A}\tilde{\boldsymbol{p}} $$



b

1.4.3 Direction Cosine Matrix

- The rotation matrix can be written as

$$ {}^{A}\mathbf{R}_{B} = \begin{pmatrix} \cos(\hat{\boldsymbol{x}}_A, \hat{\boldsymbol{x}}_B) & \cos(\hat{\boldsymbol{x}}_A, \hat{\boldsymbol{y}}_B) & \cos(\hat{\boldsymbol{x}}_A, \hat{\boldsymbol{z}}_B) \\ \cos(\hat{\boldsymbol{y}}_A, \hat{\boldsymbol{x}}_B) & \cos(\hat{\boldsymbol{y}}_A, \hat{\boldsymbol{y}}_B) & \cos(\hat{\boldsymbol{y}}_A, \hat{\boldsymbol{z}}_B) \\ \cos(\hat{\boldsymbol{z}}_A, \hat{\boldsymbol{x}}_B) & \cos(\hat{\boldsymbol{z}}_A, \hat{\boldsymbol{y}}_B) & \cos(\hat{\boldsymbol{z}}_A, \hat{\boldsymbol{z}}_B) \end{pmatrix} $$

where $\cos(\widehat{\boldsymbol{u}}_A, \widehat{\boldsymbol{u}}_B)$ is the cosin of the angle between the unit vector $\widehat{\boldsymbol{u}}_A$ in {A} and the unit vector $\widehat{\boldsymbol{u}}_B$ in {B}.

- This is called the direction cosin matrix.

1.4.4 Efciency of Representation

Type	N	$N_{\mathbf{min}}$	$N_{\mathbf{min}}/N$
$\mathbf{SO}(2)$ matrix	4	1	25%
$\mathbf{SE}(2)$ matrix	9	3	33%
$\mathbf{SE}(2)$ as 2×3 matrix †	6	3	50%
2D twist vector	3	3	**100%**
$\mathbf{SO}(3)$ matrix	9	3	33%
Unit quaternion	4	3	75%
Unit quaternion vector part	3	3	**100%**
Euler parameters	3	3	**100%**
$\mathbf{SE}(3)$ matrix	16	6	38%
$\mathbf{SE}(3)$ as 3×4 matrix †	12	6	50%
3D twist vector	6	6	**100%**
Vector + unit quaternion	7	6	86%
Unit dual quaternion	8	6	75%