Position and Orientation

KOM4520 Fundamentals of Robotic Vision

Today’s lecture

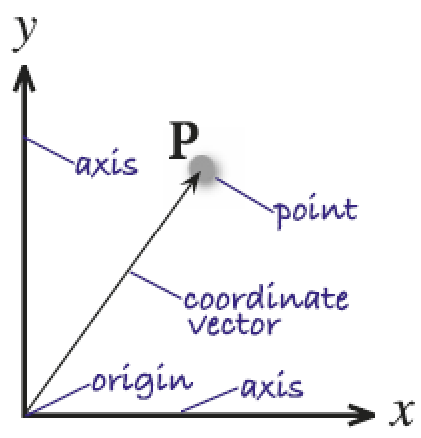
- Position & Orientation
- The relative poses
- Algebraic rules regarding the pose
- Right-handed Cartesian coordinate system
- Orientation in 2D
- Pose in 2D
- Orientation in 3D
- Three- Angle Representations
- Pose in 3D

Position & Orientation

- A point in space is a familiar concept from mathematics and can be described by a coordinate vector.
- The vector represents the displacement of the point with respect to some reference coordinate frame.

- A vector can be described in terms of a linear combination of unit vectors which are parallel to the axes of the coordinate frame.
- Note that points and vectors are different types of mathematical objects even though each can be described by a tuple of numbers.

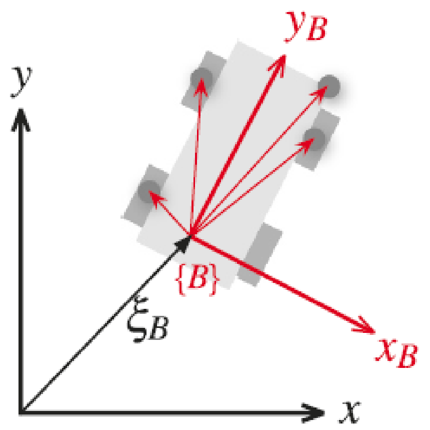
- We can add vectors but adding points makes no sense. The difference of two points is a vector, and we can add a vector to a point to obtain another point.



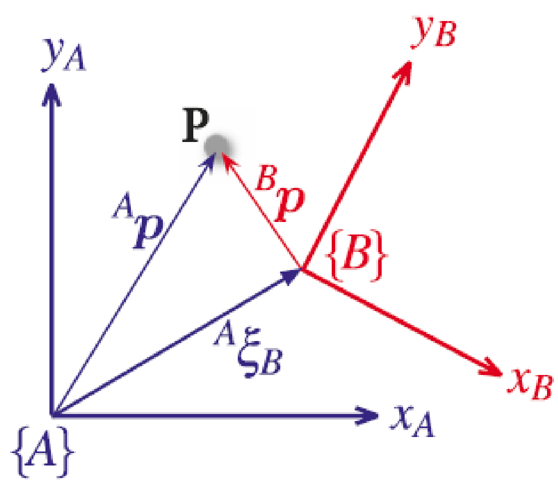
A coordinate frame, or Cartesian coordinate system, is a set of orthogonal axes which intersect at a point known as the origin.

The relative poses

- The points to be described with respect to the object's coordinate frame {B} which in turn should be described regarding a relative **pose** ξ_B
- (ξ is read as ksi)
- Object coordinate frame is labeled as $\{B\}$ and its axes are labeled x_B and y_B, adopting the frame's label as their subscript.



- Two frames $\{A\}$ and $\{B\}$ and the relative pose ${}^{A}\xi_B$ which describes $\{B\}$ with respect to $\{A\}$. The pose
- of {B} relative to {A} is ${}^{A}\xi_B$
- The leading superscript denotes the reference coordinate frame and the subscript denotes the frame being described.
- If the initial superscript is missing we assume that the change in pose is relative to the world coordinate frame which is generally denoted {O}. e.g ξ_B

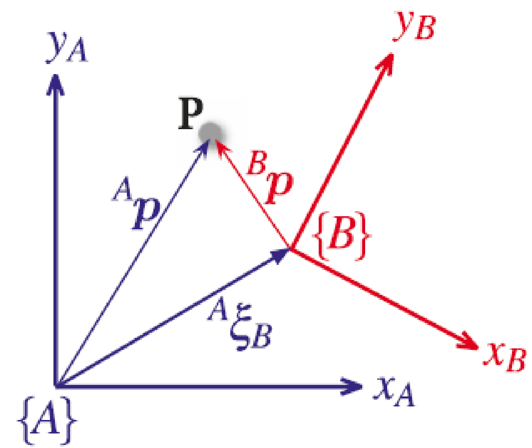


The relative poses

- The point **P** in Fig. can be described with respect to *either* coordinate frame by the vectors ${}^{A}p$ or ${}^{B}p$ respectively.

$$ {}^{A}\boldsymbol{p} = {}^{A}\xi_{B} \cdot {}^{B}\boldsymbol{p} $$

- the right-hand side expresses the motion from {*A*} to {*B*} and then to $\boldsymbol{p}$. The operator · *transforms* the vector, resulting in a new vector that describes the same point but with respect to a different coordinate frame.



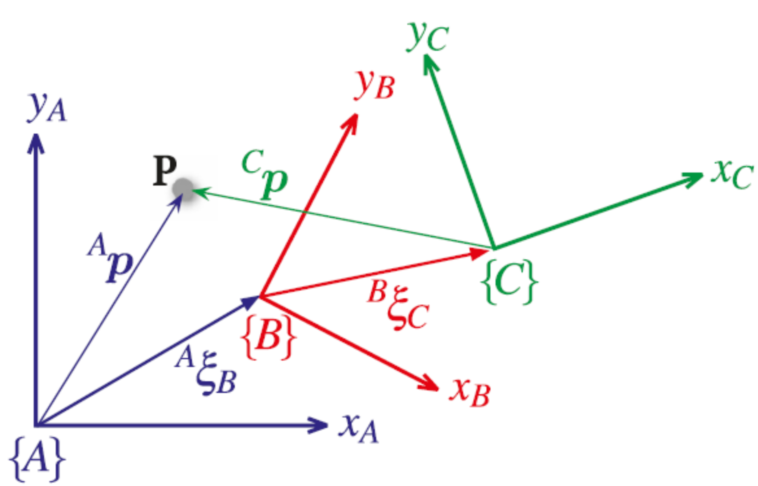
- If one frame can be described in terms of another by a relative pose then they can be applied sequentially

$$ {}^{A}\xi_{C} = {}^{A}\xi_{B} \oplus {}^{B}\xi_{C} $$
$$ {}^{A}\boldsymbol{p} = {}^{A}\xi_{C} \cdot {}^{C}\boldsymbol{p} = ({}^{A}\xi_{B} \oplus {}^{B}\xi_{C}) \cdot {}^{C}\boldsymbol{p} $$

$\oplus$ is composition of relative poses.

(there are mathematical representations for ξ, ·, and $\oplus$)

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



The relative poses

$$\xi_B = \xi_F \oplus {}^F\xi_B = \xi_R \oplus {}^R\xi_C \oplus {}^C\xi_B$$

The pose of {B} rel. to world coord.

equals to

the pose of {F} rel. to world coord. combined with the pose of {B} rel. to {F}

and also equals to

the pose of {R} rel. to world coord. combined with the pose of {C} rel. to {R} combined with the pose {B} rel. to {C}

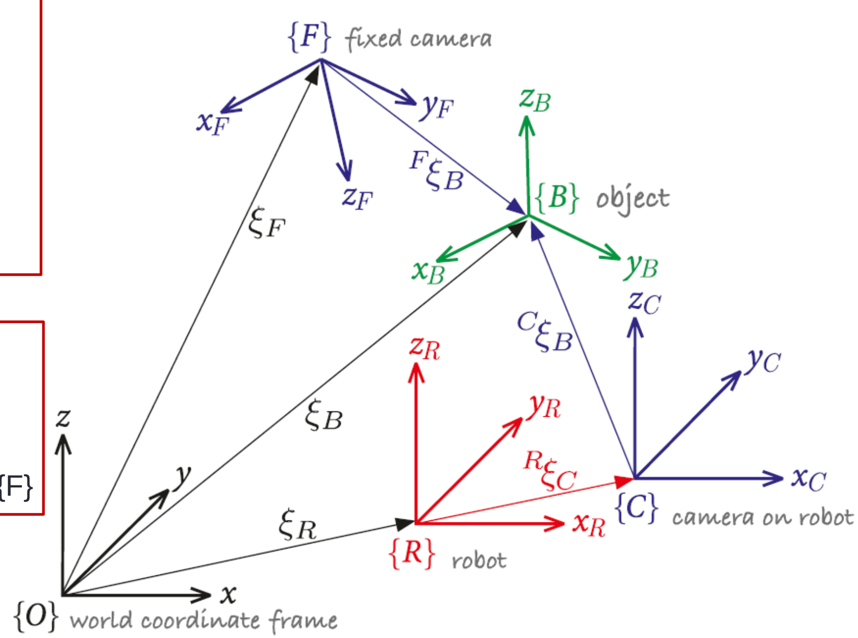
$$\xi_R = \xi_F \oplus {}^F\xi_R$$

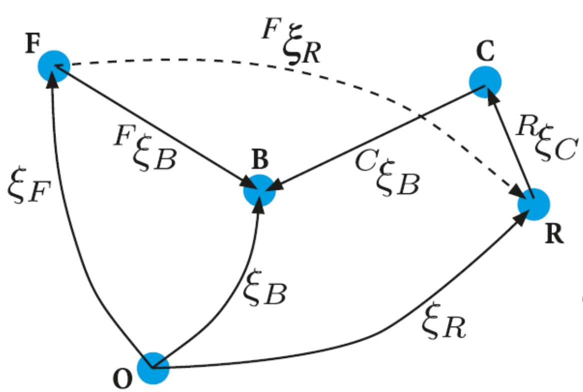
The pose of {R} rel. to world coord.

equals to

the pose of {F} rel. to world coord. combined with the pose of {R} rel. to {F}

Multiple 3-dimensional coordinate frames and relative poses





directed graph for the same scenario

Algebraic rules regarding the pose

$\oplus$ is composition of relative poses, 0 represents a zero relative pose

A pose ${}^{A}\xi_{B}$ has an inverse as $\ominus\, {}^{A}\xi_{B}$ and equals to ${}^{B}\xi_{A}$

(which represented with an arrow from $\{B\}$ to $\{A\}$)

$\ominus$ also represents inverse composition

$$\xi \oplus 0 = \xi$$
$$\xi \ominus 0 = \xi$$
$$\xi \ominus \xi = 0$$
$$\ominus\, \xi \oplus \xi = 0$$

Composition is not commutative $\quad \xi_1 \oplus \xi_2 \neq \xi_2 \oplus \xi_1$

Only exception is when $\xi_1 \oplus \xi_2 = 0$

A relative pose can transform a point expressed as a vector relative to one frame ⟶ (to) a vector relative to another frame

$${}^{A}\boldsymbol{p} = {}^{A}\xi_{B} \cdot {}^{B}\boldsymbol{p}$$

$\xi_R = \xi_F \oplus {}^{F}\xi_R$ this equation says that the pose of the robot is the same as composing two relative poses: the pose of fixed camera relative to world frame and the pose of robot relative to fixed camera. By adding inverse of ξ_F

$$\ominus\, \xi_F \oplus \xi_R = \ominus\, \xi_F \oplus \xi_F \oplus {}^{F}\xi_R$$
$$\ominus\, \xi_F \oplus \xi_R = {}^{F}\xi_R$$

Which is the pose of the robot relative to the fixed camera

The discussions on relative poses in short

- What is ξ? We will use it as a mathematical entity.

- A point is described by a bound coordinate vector that represents its displacement from the origin of a reference coordinate system.
- Points and vectors are different things.
- A set of points that represent a rigid object can be described by a single coordinate frame, and its principal points are described by constant vectors relative to that coordinate frame.

- The position and orientation of an object's coordinate frame is referred to as its pose.
- A relative pose describes the pose of one coordinate frame with respect to another and is denoted by an algebraic variable ξ .
- A coordinate vector describing a point can be represented with respect to a different coordinate frame by applying the relative pose to the vector using the $\cdot$ operator.
- We can perform algebraic manipulation of expressions written in terms of relative poses and the operators $\oplus$ and $\ominus$.

After this point different representations of ξ will be discussed

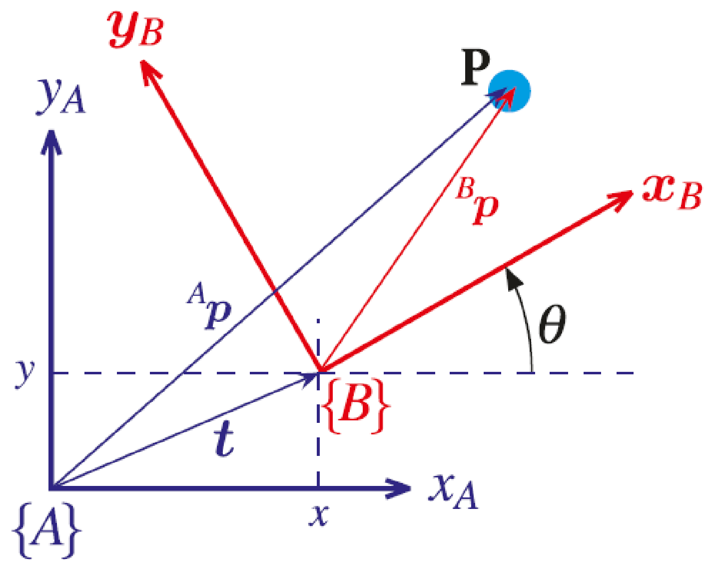
Right-handed Cartesian coordinate system

- The right-handed Cartesian coordinate system or coordinate frame with orthogonal axes denoted *x* and *y* and typically drawn with the x-axis horizontal and the y-axis vertical.
- The point of intersection is called the origin. Unit-vectors parallel to the axes are denoted $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{y}}$.

- A point is described with a bound vector $\boldsymbol{p}$

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

- The coordinate frame $\{B\}$ can be described with respect to the reference frame $\{A\}$.
- The origin of $\{B\}$ has been displaced by the vector $\boldsymbol{t} = (x, y)$ and then rotated counter-clockwise by an angle θ



A concrete representation of pose is therefore associated with 3-vectors $^A\xi_B \sim (x, y, \theta)$

Just using (x, y, θ) is not convenient for $\oplus$ operation. We should find a different ways to represent pose.

Orientation in 2D

- Consider a frame $\{V\}$ whose axes are parallel to those of the reference frame $\{A\}$ but whose origin is the same as $\{B\}$
- We can express a point P with respect to $\{V\}$ as

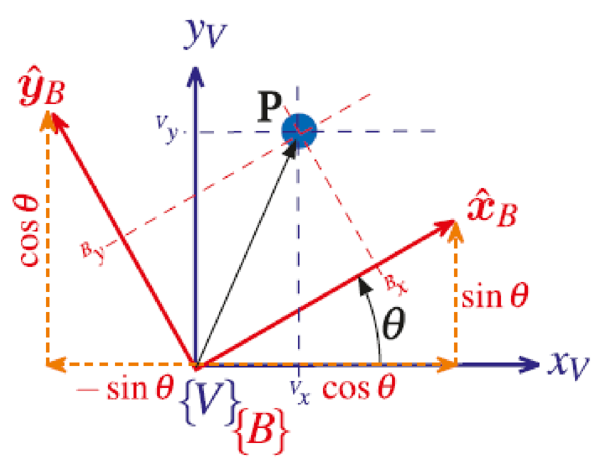
$$^{V}\boldsymbol{p} = {}^{V}x\,\hat{\boldsymbol{x}}_V + {}^{V}y\,\hat{\boldsymbol{y}}_V$$

$$^{V}\boldsymbol{p} = \begin{bmatrix}\hat{\boldsymbol{x}}_V & \hat{\boldsymbol{y}}_V\end{bmatrix}\begin{bmatrix}^{V}x \\ ^{V}y\end{bmatrix}$$

The two unit vectors of coordinate frame $\{B\}$ can be described by its two orthogonal axes

$$\hat{\boldsymbol{x}}_B = cos\theta\hat{\boldsymbol{x}}_V + sin\theta\hat{\boldsymbol{y}}_V$$
$$\hat{\boldsymbol{y}}_B = -sin\theta\hat{\boldsymbol{x}}_V + cos\theta\hat{\boldsymbol{y}}_V$$
$$\begin{bmatrix}\hat{\boldsymbol{x}}_B & \hat{\boldsymbol{y}}_B\end{bmatrix} = \begin{bmatrix}\hat{\boldsymbol{x}}_V & \hat{\boldsymbol{y}}_V\end{bmatrix}\begin{bmatrix}cos\theta & -sin\theta \\ sin\theta & cos\theta\end{bmatrix}$$

$$^{B}\boldsymbol{p} = {}^{B}x\,\hat{\boldsymbol{x}}_B + {}^{B}y\,\hat{\boldsymbol{y}}_B$$
$$^{B}\boldsymbol{p} = \begin{bmatrix}\hat{\boldsymbol{x}}_B & \hat{\boldsymbol{y}}_B\end{bmatrix}\begin{bmatrix}^{B}x \\ ^{B}y\end{bmatrix}$$
$$^{B}\boldsymbol{p} = \begin{bmatrix}\hat{\boldsymbol{x}}_V & \hat{\boldsymbol{y}}_V\end{bmatrix}\begin{bmatrix}cos\theta & -sin\theta \\ sin\theta & cos\theta\end{bmatrix}\begin{bmatrix}^{B}x \\ ^{B}y\end{bmatrix}$$



Equating $^{V}\boldsymbol{p}$ and $^{B}\boldsymbol{p}$

$$\begin{bmatrix}\hat{\boldsymbol{x}}_V & \hat{\boldsymbol{y}}_V\end{bmatrix}\boxed{\begin{bmatrix}^{V}x \\ ^{V}y\end{bmatrix}} = \begin{bmatrix}\hat{\boldsymbol{x}}_V & \hat{\boldsymbol{y}}_V\end{bmatrix}\boxed{\begin{bmatrix}cos\theta & -sin\theta \\ sin\theta & cos\theta\end{bmatrix}\begin{bmatrix}^{B}x \\ ^{B}y\end{bmatrix}}$$

$$\begin{bmatrix}^{V}x \\ ^{V}y\end{bmatrix} = {}^{V}\mathbf{R}_B\begin{bmatrix}^{B}x \\ ^{B}y\end{bmatrix}$$

Orientation in 2D

$$ {}^{V}\boldsymbol{R}_{B}(\theta) = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} $$

is a 2D rotation matrix with some special properties

- it is *orthonormal* since each of its columns is a unit vector and the columns are orthogonal.
- the columns are the unit vectors that define the axes of the rotated frame $\{B\}$ *with respect to* $\{V\}$ are by definition both unit-length and orthogonal.

- its *determinant* is +1, which means that the length of a vector is unchanged after transformation,.
- the inverse is the same as the transpose, that is, $\boldsymbol{R}^{-1} = \boldsymbol{R}^{T}$.

Using the equation

$$ \begin{bmatrix} {}^{V}x \\ {}^{V}y \end{bmatrix} = {}^{V}\boldsymbol{R}_{B} \begin{bmatrix} {}^{B}x \\ {}^{B}y \end{bmatrix} $$

We can get

$$ \begin{bmatrix} {}^{B}x \\ {}^{B}y \end{bmatrix} = ({}^{V}\boldsymbol{R}_{B})^{-1} \begin{bmatrix} {}^{V}x \\ {}^{V}y \end{bmatrix} = ({}^{V}\boldsymbol{R}_{B})^{T} \begin{bmatrix} {}^{V}x \\ {}^{V}y \end{bmatrix} = {}^{B}\boldsymbol{R}_{V} \begin{bmatrix} {}^{V}x \\ {}^{V}y \end{bmatrix} $$

Orthonormal Matrix

Suppose $\boldsymbol{A} = [\boldsymbol{u}_1|\boldsymbol{u}_2|\boldsymbol{u}_3| \dots |\boldsymbol{u}_n]$ is a matrix or a set of orthogonal vectors ($\langle \boldsymbol{u}_i, \boldsymbol{u}_j \rangle = 0$ whenever $i \neq j$) such that $\| \boldsymbol{u}_i \| = 1$ for all $1 \leq i \leq n$).
Then $\boldsymbol{A}$ is an orthonormal matrix

Orientation in 2D

R = rot2(0.2) R = 0.9801 -0.1987 0.1987 0.9801	det(R) ans = 1	>> syms theta >> R = rot2(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)] >> simplify(det(R)) ans = 1

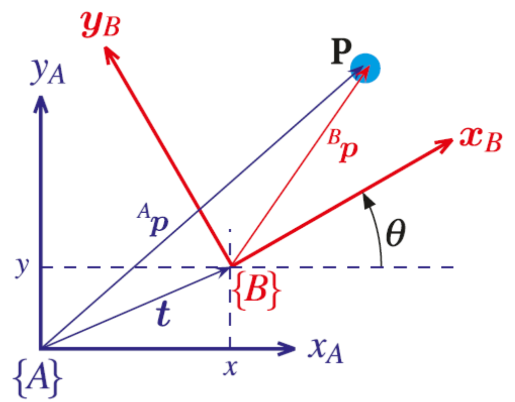
Operations involved with 2D rotation matrices in MATLAB

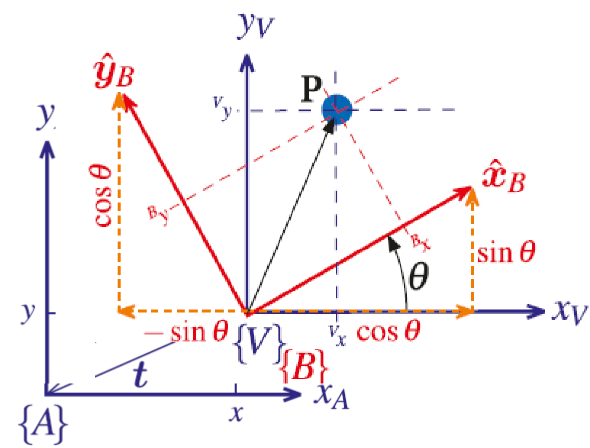
Pose in 2D

The translation should be regarded as well.

Since the axes of $\{V\}$ and $\{A\}$ are parallel this will correspond to a vectoral addition

- $\begin{bmatrix} {}^{A}x \\ {}^{A}y \end{bmatrix} = \begin{bmatrix} {}^{V}x \\ {}^{V}y \end{bmatrix} + \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} \begin{bmatrix} {}^{B}x \\ {}^{B}y \end{bmatrix} + \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} cos\theta & -sin\theta & x \\ sin\theta & cos\theta & y \end{bmatrix} \begin{bmatrix} {}^{B}x \\ {}^{B}y \\ 1 \end{bmatrix}$
- Or more compactly as
- $\begin{bmatrix} {}^{A}x \\ {}^{A}y \\ 1 \end{bmatrix} = \begin{bmatrix} cos\theta & -sin\theta & x \\ sin\theta & cos\theta & y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} {}^{B}x \\ {}^{B}y \\ 1 \end{bmatrix} = \begin{bmatrix} {}^{A}\boldsymbol{R}_{B} & \boldsymbol{t} \\ \boldsymbol{0}_{1x2} & 1 \end{bmatrix} \begin{bmatrix} {}^{B}x \\ {}^{B}y \\ 1 \end{bmatrix}$





$\boldsymbol{t} = (x, y)$ is the translation of the frame and the orientation is the ${}^{A}\boldsymbol{R}_{B}$

${}^{A}\boldsymbol{R}_{B}$ and ${}^{V}\boldsymbol{R}_{B}$ are the same since frames {A} and {B} are parralel.

Pose in 2D

- The coordinate vectors for point **P** are now expressed in homogeneous form and we write

$$^{A}\tilde{\boldsymbol{p}} = \begin{bmatrix} ^{A}\boldsymbol{R}_{B} & \boldsymbol{t} \\ \boldsymbol{0}_{1x2} & 1 \end{bmatrix} {}^{B}\tilde{\boldsymbol{p}} = {}^{A}\boldsymbol{T}_{B}\, {}^{B}\tilde{\boldsymbol{p}}$$

$^{A}\boldsymbol{T}_{B}$ is referred as homogeneous transformation and represents translation & orientation or relative pose. This is often referred as rigid body motion.

$$\boldsymbol{T} = \begin{bmatrix} cos\theta & -sin\theta & x \\ sin\theta & cos\theta & y \\ 0 & 0 & 1 \end{bmatrix}$$

Is a concrete representation of ξ

These are several mathematical resemblances of ξ related operations

$\boldsymbol{T}_1 \oplus \boldsymbol{T}_2 \rightarrow \boldsymbol{T}_1\boldsymbol{T}_2$ as matrix multiplication

$$\boldsymbol{T}_1\boldsymbol{T}_2 = \begin{bmatrix} \boldsymbol{R}_1\boldsymbol{R}_2 & \boldsymbol{t}_1 + \boldsymbol{R}_1\boldsymbol{t}_2 \\ \boldsymbol{0}_{1x2} & 1 \end{bmatrix}$$

The pose 0 is the identity matrix

$\xi \oplus 0 \rightarrow \boldsymbol{T}\boldsymbol{I} = \boldsymbol{T}$

$\ominus \boldsymbol{\xi} \rightarrow \boldsymbol{T}^{-1}$

$\boldsymbol{\xi} \ominus \boldsymbol{\xi} \rightarrow \boldsymbol{T}^{-1}\boldsymbol{T} = \boldsymbol{I}$

$$\boldsymbol{T}^{-1} = \begin{bmatrix} \boldsymbol{R}^{T} & -\boldsymbol{R}^{T}\boldsymbol{t} \\ \boldsymbol{0}_{1x2} & 1 \end{bmatrix}$$

$\boldsymbol{T} \cdot \tilde{\boldsymbol{p}} \rightarrow \boldsymbol{T}\tilde{\boldsymbol{p}}$

Pose in 2D

Operations involved with 2D pose in MATLAB

>> T1 = transl2(1, 2) * trot2(30, 'deg') T1 = 0.8660 -0.5000 1.0000 0.5000 0.8660 2.0000 0 0 1.0000 plotvol([0 5 0 5]); >> trplot2(T1, 'frame', '1', 'color', 'b')	>> T2 = transl2(2, 1) T2 = 1 0 2 0 1 1 0 0 1 >>trplot2(T2, 'frame', '2', 'color', 'r');
>> T3 = T1*T2 T3 = 0.8660 -0.5000 2.2321 0.5000 0.8660 3.8660 0 0 1.0000 >> trplot2(T3, 'frame', '3', 'color', 'g');	T4 = T2*T1; >> trplot2(T4, 'frame', '4', 'color', 'c'); >> P = [3 ; 2]; >> plot_point(P, 'label', 'P', 'solid', 'ko');

The function transl2 creates a relative pose with a finite translation but zero rotation, while trot2 creates a relative pose with a finite rotation but zero translation

Pose in 2D

$$^{0}\boldsymbol{p} = {}^{0}\xi_{1}\, {}^{1}\boldsymbol{p}$$

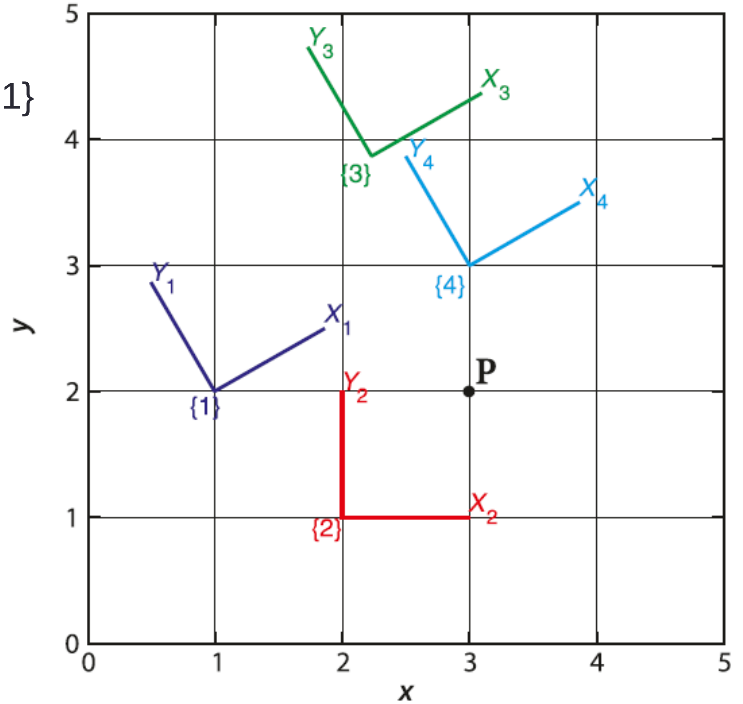
To determine the coordinate of the point with respect to {1}

$$^{1}\boldsymbol{p} = \left({}^{0}\xi_{1}\right)^{-1} \cdot {}^{0}\boldsymbol{p}$$

```
>> P1 = inv(T1) * [P; 1]
P1 =
1.7321
-1.0000
1.0000
```

homogeneous form of the point

```
>> h2e( inv(T1) * e2h(P) )
ans =
1.7321
-1.0000
```



- The function e2h converts Euclidean form to homogeneous forom and h2e performs the inverse conversion.

Pose in 2D – centers of rotation

Operations involved with 2D pose in MATLAB

```
>> plotvol([-5 4 -1 5]);
>> T0 = eye(3,3);
>> trplot2(T0, 'frame', '0');
>> X = transl2(2, 3);
>> trplot2(X, 'frame', 'X');
```

```
>> R = trot2(2);
```

and plot the effect of the two possible orders of composition

```
>> trplot2(R*X, 'framelabel', 'RX', 'color', 'r');
>> trplot2(X*R, 'framelabel', 'XR', 'color', 'r');
```

the frame {*RX*} has been rotated about the origin, while frame {*XR*} has been rotated about the origin of {*X*}.

```
>> C = [1 2]';
>> plot_point(C, 'label', ' C', 'solid', 'ko')
```

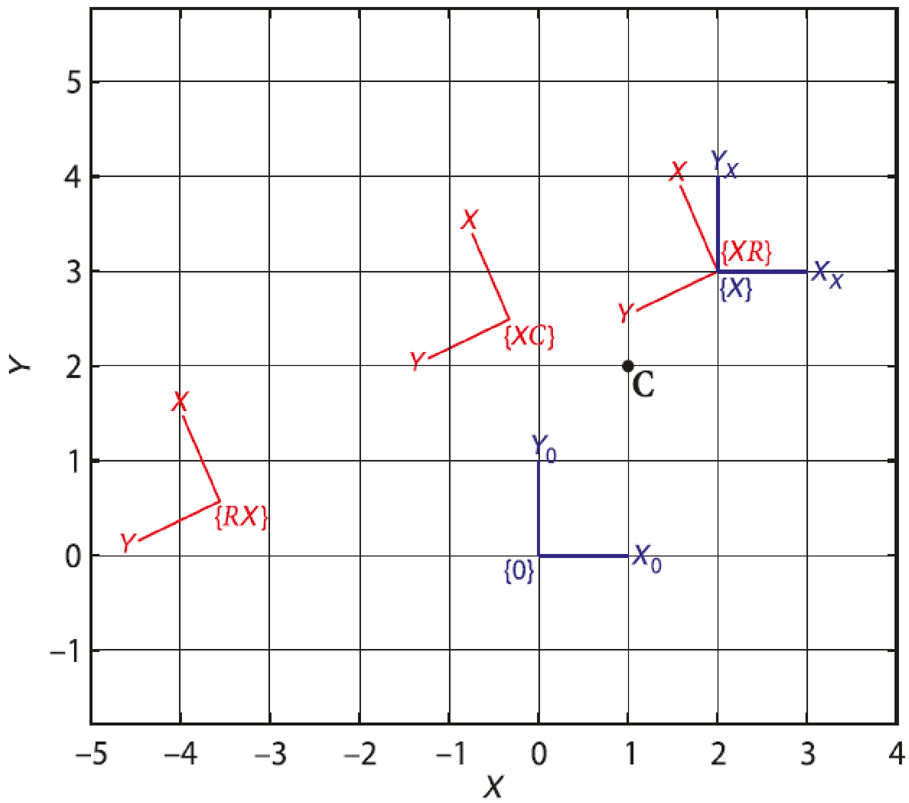
and then compute a transform to rotate about point **C**

```
>> RC = transl2(C) * R * transl2(-C)
RC =
-0.4161 -0.9093 3.2347
0.9093 -0.4161 1.9230
0 0 1.0000
```

and applying this

```
>> trplot2(RC*X, 'framelabel', 'XC', 'color', 'r');
```

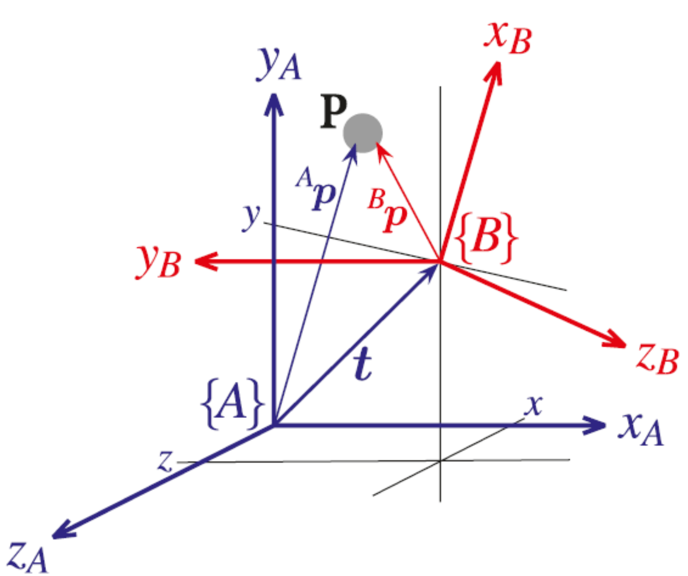
Pose in 2D – centers of rotation



The frame {*X*} is rotated by 2 radians about {0} to give frame {*RX*}, about {*X*} to give {*XR*}, and about point **C** to give frame {*XC*}

Orientation in 3D

point **P** is represented by its *x*-, *y*- and *z*-coordinates (*x*, *y*, *z*) or as a bound vector



Two 3D coordinate frames {*A*} and {*B*}. {*B*} is rotated and translated with respect to {*A*}

Rotation matrix in 3D

$$\begin{bmatrix} {}^A x \\ {}^A y \\ {}^A z \end{bmatrix} = {}^A\boldsymbol{R}_B \begin{bmatrix} {}^B x \\ {}^B y \\ {}^B z \end{bmatrix}$$

which transforms the point defined with respect to frame {*B*} to a vector with respect to {*A*}.

- The orthonormal rotation matrix has nine elements but they are not independent.
- The columns have unit magnitude which provides three constraints.
- The columns are orthogonal to each other which provides another three constraints.
- Nine elements and six constraints is effectively three independent values.

Orientation in 3D

3-dimensional rotation matrix ${}^{A}\boldsymbol{R}_{B}$ has some special properties:

- it is *orthonormal* since each of its columns is a unit vector and the columns are orthogonal.
- the columns are the unit vectors that define the axes of the rotated frame *{B}* with respect to *{A}* and are by definition both unit-length and orthogonal.
- its determinant is +1, which means that the length of a vector is unchanged
- the inverse is the same as the transpose, that is, $\boldsymbol{R}^{-1} = \boldsymbol{R}^{T}$

- The orthonormal rotation matrices for rotation of θ about the *x*-, *y*- and *z*-axes are

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

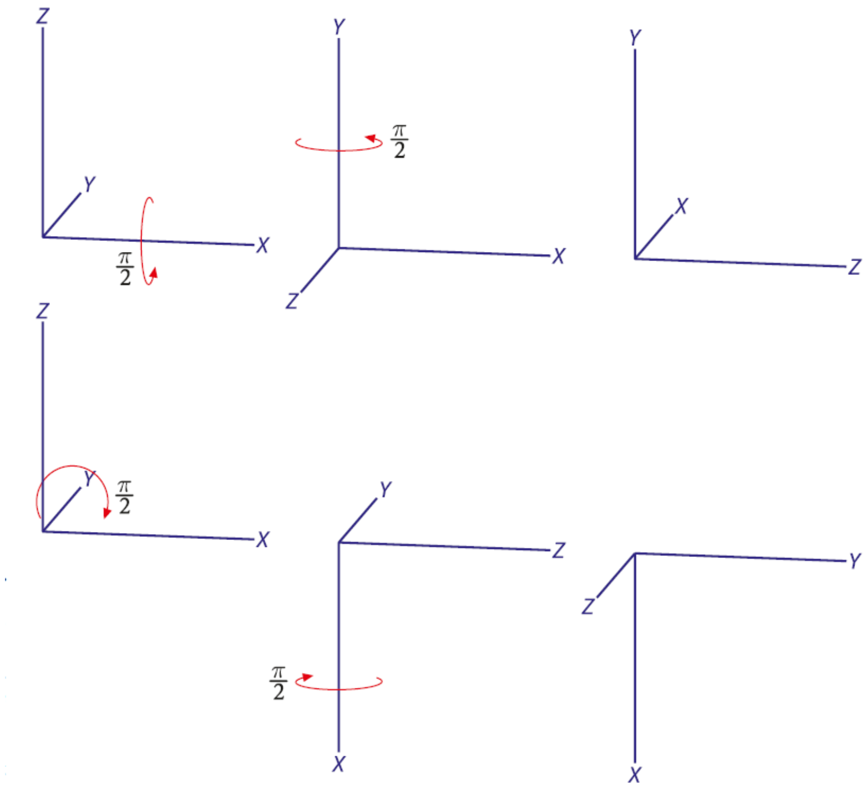
$$\boldsymbol{R}_y(\theta) = \begin{bmatrix} cos\theta & 0 & sin\theta \\ 0 & 1 & 0 \\ -sin\theta & 0 & cos\theta \end{bmatrix}$$

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

```
>> R = rotx(pi/2)
R =
1.0000 0 0
0 0.0000 -1.0000
0 1.0000 0.0000
```

Orientation in 3D

Rotation about a vector. Wrap your right hand around the vector with your thumb (your *x*-fi nger) in the direction of the arrow. The curl of your fingers indicates the direction of increasing angle.



Example showing the noncommutativity of rotation.
In the top row the coordinate frame is rotated by $\pi/2$ about the *x*-axis and then $\pi/2$ about the *y*-axis.
In the bottom row the order of rotations has been reversed.
The results are clearly different

Three- Angle Representations

- We can imagine picking up frame {*A*} in our hand and rotating it until it looked just like frame {*B*}. *Euler's rotation theorem* states that any rotation can be considered as a sequence of rotations about different coordinate axes. This means that in general an arbitrary rotation between frames can be decomposed into a sequence of three rotation angles and associated rotation axes.

- There are two classes of rotation sequence: Eulerian and Cardanian, named after Euler and Cardano respectively.

- The Eulerian type involves repetition, but not successive, of rotations about one
- particular axis: XYX, XZX, YXY, YZY, ZXZ, or ZYZ.
- The Cardanian type is characterized by rotations about all three axes: XYZ, XZY, YZX, YXZ, ZXY, or ZYX.

Three- Angle Representations

- The ZYZ sequence is commonly used in aeronautics and mechanical dynamics
- $R = R_z(\phi)R_y(\theta)R_z(\psi)$

- $\Gamma(\phi, \theta, \psi)$ - read as gamma of (phi, theta, psi)
- The Euler angles as a vector $\Gamma(\phi, \theta, \psi)$
- For example, to compute the equivalent rotation matrix for $\Gamma(\phi, \theta, \psi) = (0.1, 0.2, 0.3)$ we write as

```
>> R = rotz(0.1) * roty(0.2) *
rotz(0.3);
```

or conveniently

```
>> R = eul2r(0.1, 0.2, 0.3)
R =
0.9021 -0.3836 0.1977
0.3875 0.9216 0.0198
-0.1898 0.0587 0.9801
```

The inverse problem is finding the Euler angles that correspond to a given rotation matrix

```
>> gamma = tr2eul(R)
gamma =
0.1000 0.2000 0.3000
```

However if θ is negative

```
>> R = eul2r(0.1 , -0.2, 0.3)
R =
0.9021 -0.3836 -0.1977
0.3875 0.9216 -0.0198
0.1898 -0.0587 0.9801
```

the inverse function

```
>> tr2eul(R)
ans =
-3.0416 0.2000 -2.8416
```

returns a positive value for θ and quite different values for ϕ and ψ .
The mapping from a rotation matrix to Euler angles is not unique and the Toolbox *always* returns a positive angle for θ .

Three- Angle Representations

- Another widely used convention are the Cardan angles: roll, pitch and yaw.
- there are two different versions in common use. Text books seem to define the roll-pitch-yaw sequence as ZYX or XYZ.

- It is intuitive to apply the rotations in the sequence:
- yaw (direction of travel),
- pitch (elevation of the front with respect to horizontal)
- and then finally roll (rotation about the forward axis of the vehicle).

- This leads to the ZYX angle sequence
- $R = R_z(\theta_y)R_y(\theta_p)R_z(\theta_r)$
- Roll-pitch-yaw angles are also known as Tait-Bryan angles and for aeronautical applications they can be called bank, attitude and heading angles respectively.

```
>> R = rpy2r(0.1, 0.2, 0.3)
R =
0.9363 -0.2751 0.2184
0.2896 0.9564 -0.0370
-0.1987 0.0978 0.9752
and the inverse is
>> gamma = tr2rpy(R)
gamma =
0.1000 0.2000 0.3000
```

Pose in 3D

- The derivation for the homogeneous transformation matrix is similar to the 2D case

$$\begin{bmatrix} {}^{A}x \\ {}^{A}y \\ {}^{A}z \\ 1 \end{bmatrix} = \begin{bmatrix} {}^{A}\boldsymbol{R}_{B} & \boldsymbol{t} \\ \boldsymbol{0}_{1\text{x}3} & 1 \end{bmatrix} \begin{bmatrix} {}^{B}x \\ {}^{B}y \\ {}^{B}z \\ 1 \end{bmatrix}$$

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

- $t \in R^3$ is a vector defining the origin of frame {*B*} with respect to frame {*A*}, and $\boldsymbol{R}$ is the 3×3 orthonormal matrix which describes the orientation of the axes of frame {*B*} with respect to frame {*A*}.

${}^{A}\boldsymbol{T}_{B}$ is the 4x4 homogeneous transformation matrix

$$\boldsymbol{T}_1\boldsymbol{T}_2 = \begin{bmatrix} \boldsymbol{R}_1\boldsymbol{R}_2 & \boldsymbol{t}_1 + \boldsymbol{R}_1\boldsymbol{t}_2 \\ \boldsymbol{0}_{1\times3} & 1 \end{bmatrix}$$

$$\boldsymbol{T}^{-1} = \begin{bmatrix} \boldsymbol{R}^{T} & -\boldsymbol{R}^{T}\boldsymbol{t} \\ \boldsymbol{0}_{1\times3} & 1 \end{bmatrix}$$

Pose in 3D

```
>> T = transl(1, 0, 0) * trotx(pi/2) * transl(0, 1, 0)
T =
1.0000 0 0 1.0000
0 0.0000 -1.0000 0.0000
0 1.0000 0.0000 1.0000
0 0 0 1.0000
```

The rotation matrix component of T is

```
>> t2r(T)
ans =
1.0000 0 0
0 0.0000 -1.0000
0 1.0000 0.0000
```

and the translation component is a column vector

```
>> transl(T)'
ans =
1.0000 0.0000 1.0000
```