Representation in 2D Space

Dr. Divya Udayan J, Ph.D.(Konkuk University, S.Korea)
Assistant Professor(Selection Grade)
Department of CSE
Amrita School of Engineering, Amritapuri Campus,
Amrita Vishwa Vidyapeetham
Email: divyaudayanj@am.amrita.edu
Mobile: 9550797705

Representing position and orientation of robots in a global environment

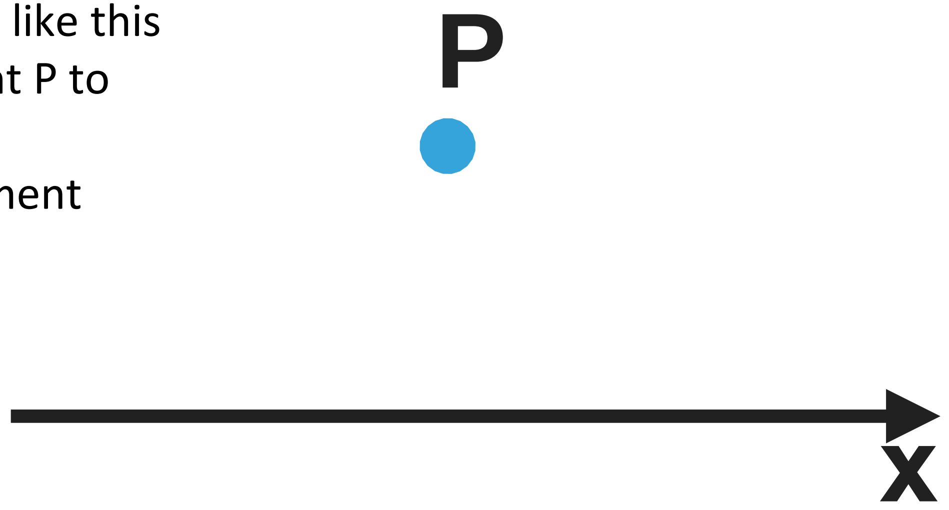
POINT in 2D space

- How a robot can identify its position in the environment
- How a robot can identify its whole existence in the environment
- After knowing this info, we need to feed this info to the robot, or program it to the robot

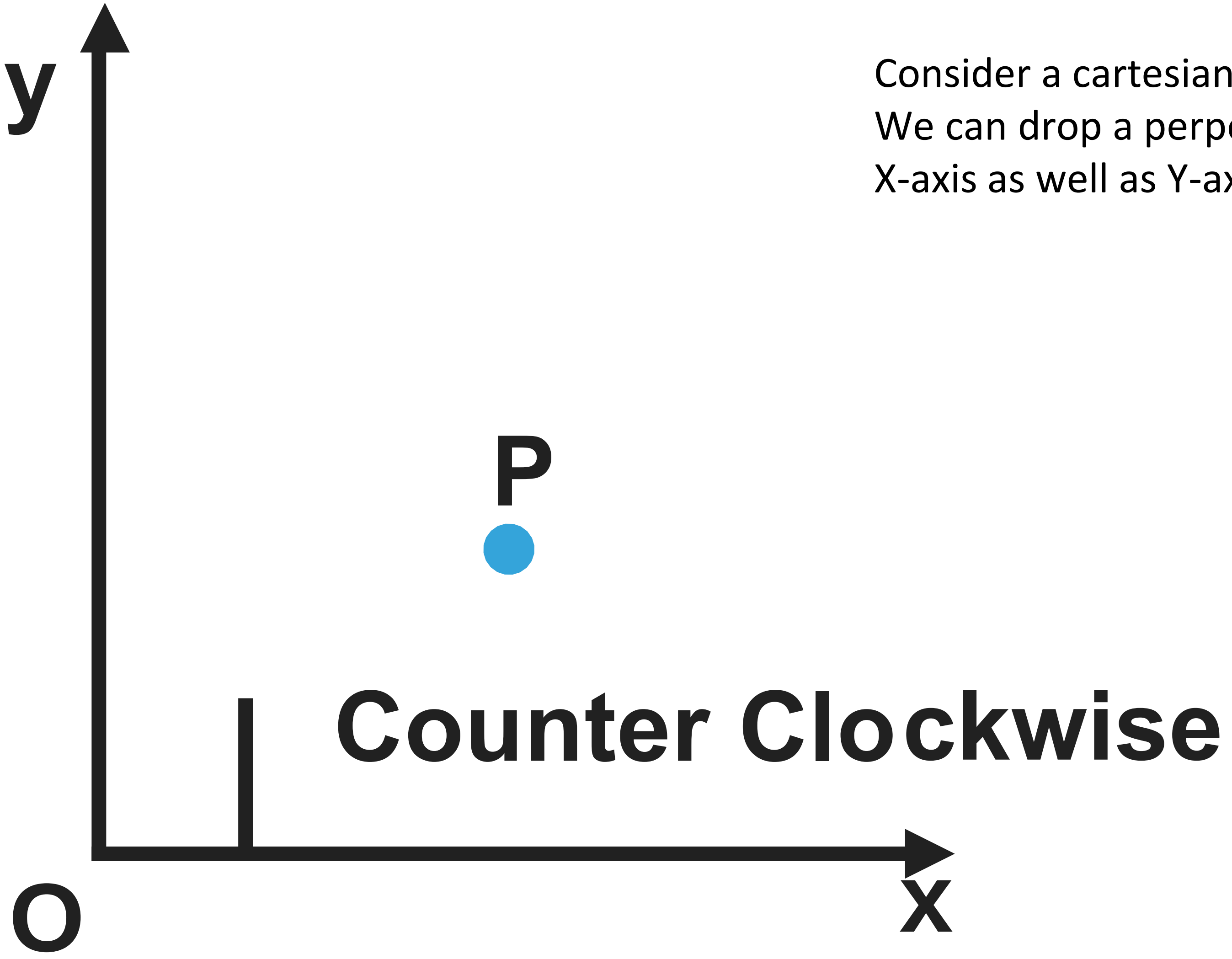
POINT in 2D space

Cartesian Coordinate System

How to identify the position of a point
In space?
Consider a cartesian coordinate system like this
We can drop a perpendicular from point P to
X-axis as well as Y-axis
By doing this, we can find the displacement
Magnitude across X-axis and Y-axis
i.e co-ordinates of the point P(a,b)

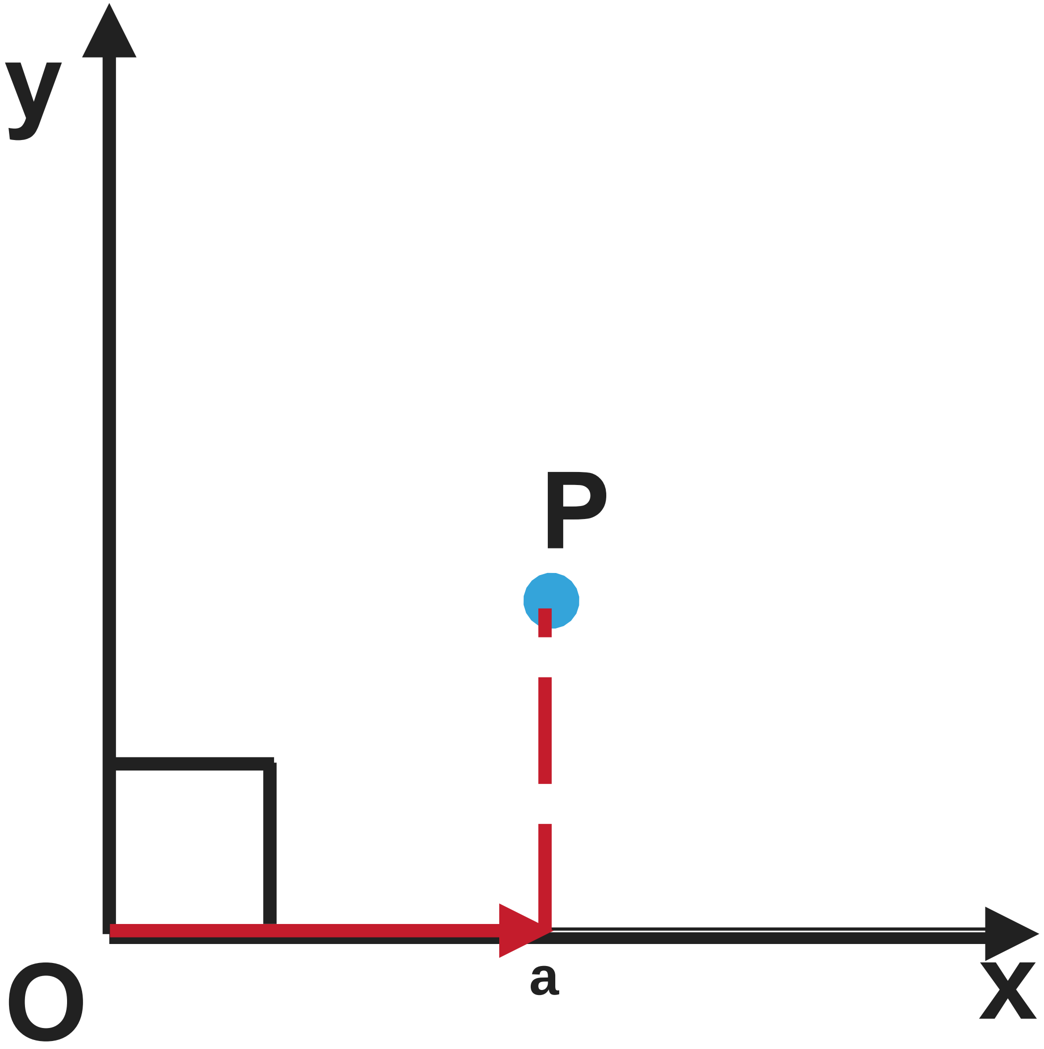


POINT in 2D space



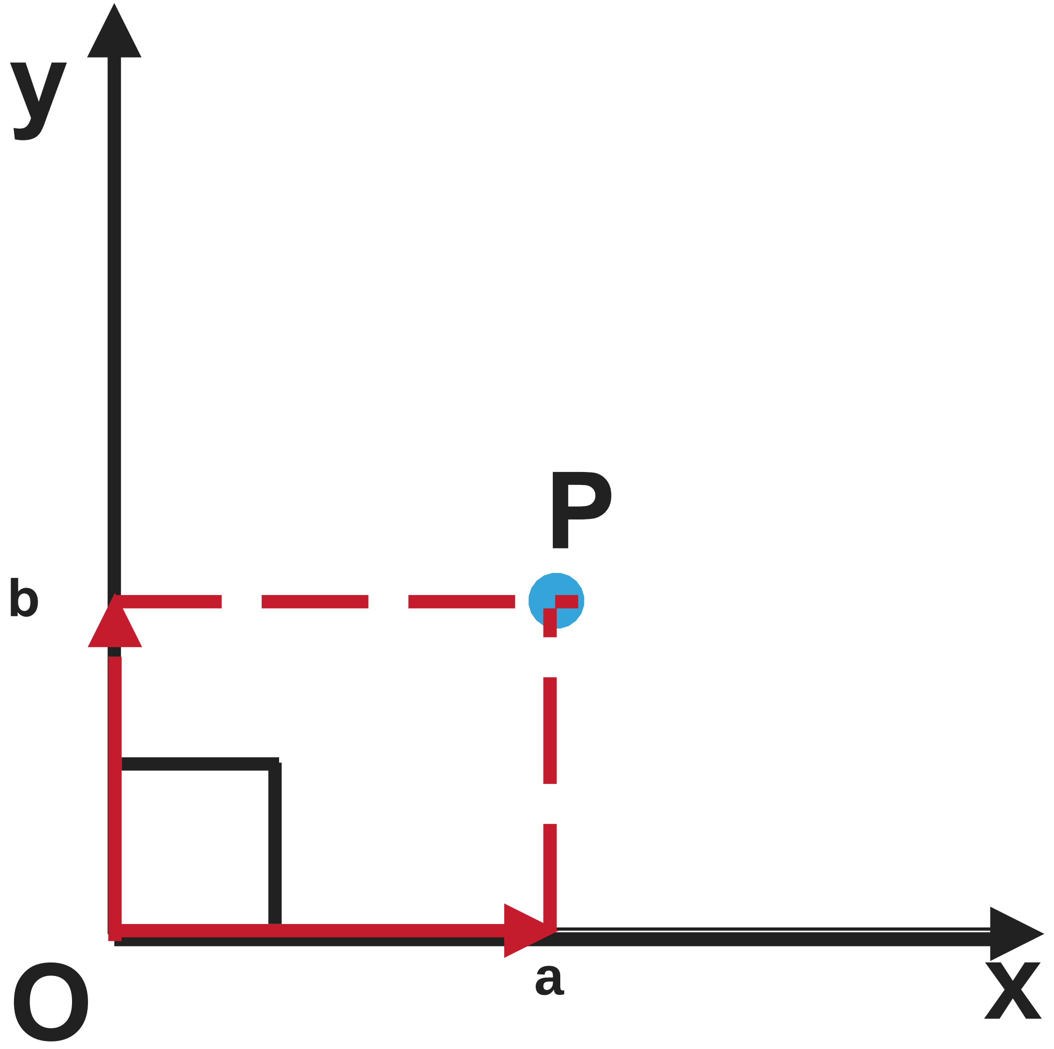
Consider a cartesian coordinate system like this
We can drop a perpendicular from point P to X-axis as well as Y-axis

POINT in 2D space

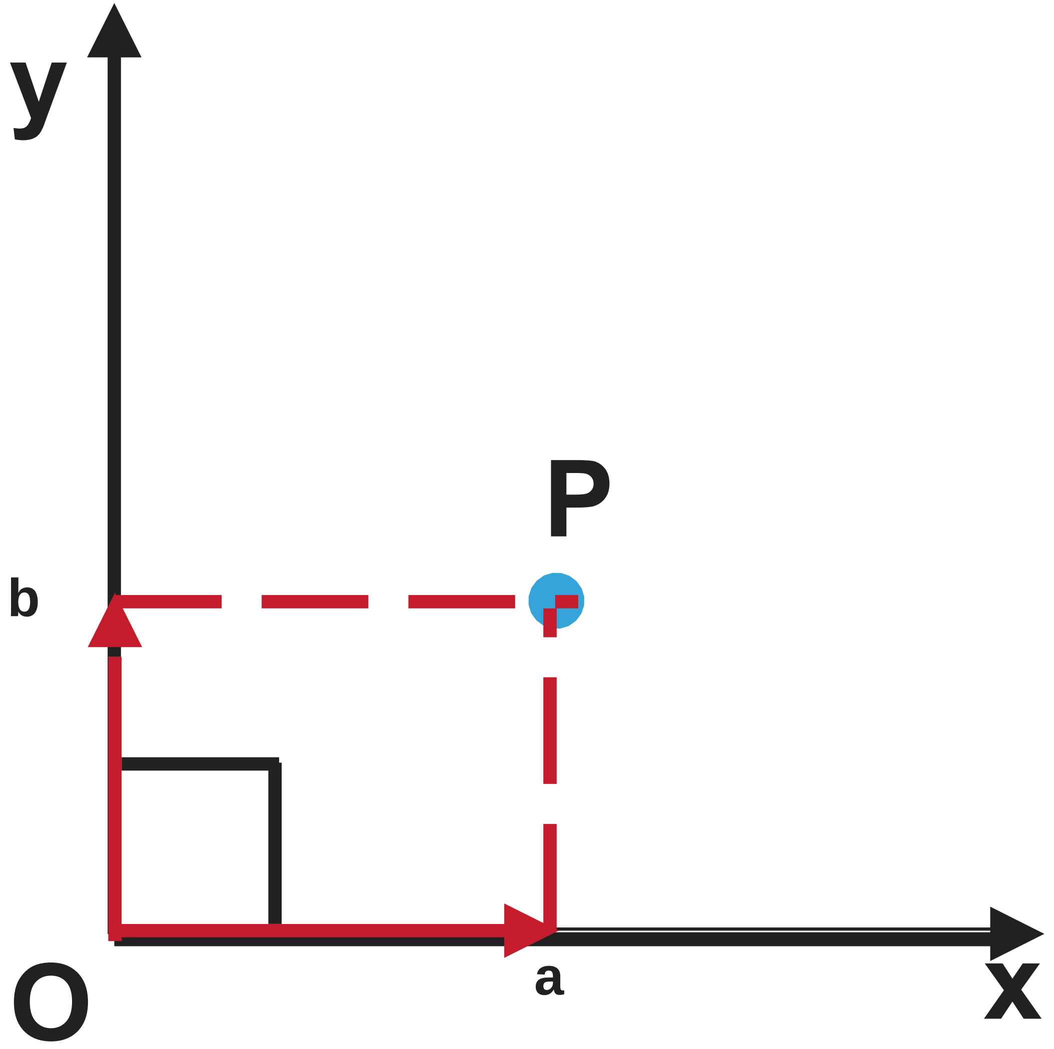


By doing this, we can find the displacement Magnitude across X-axis and Y-axis
i.e co-ordinates of the point P(a,b)

POINT in 2D space

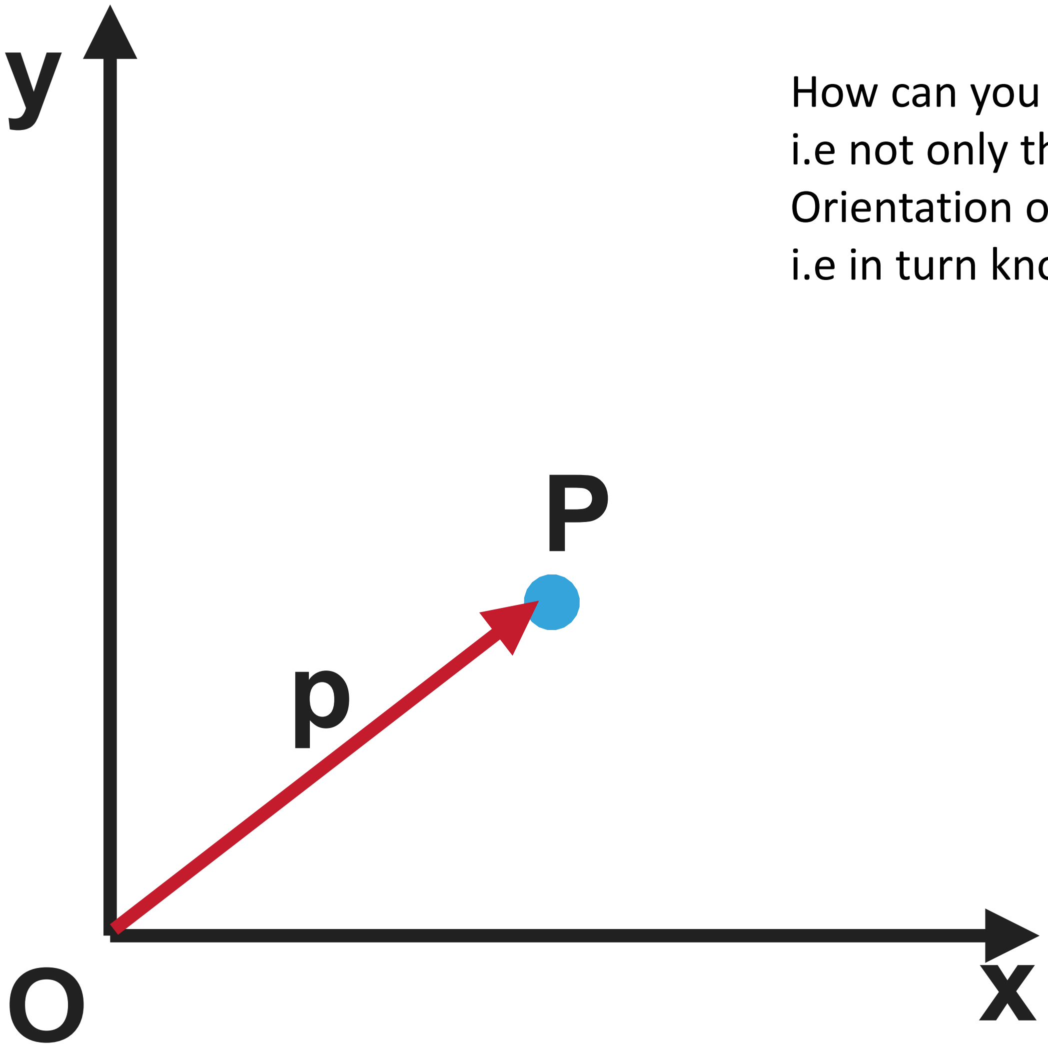


POINT in 2D space



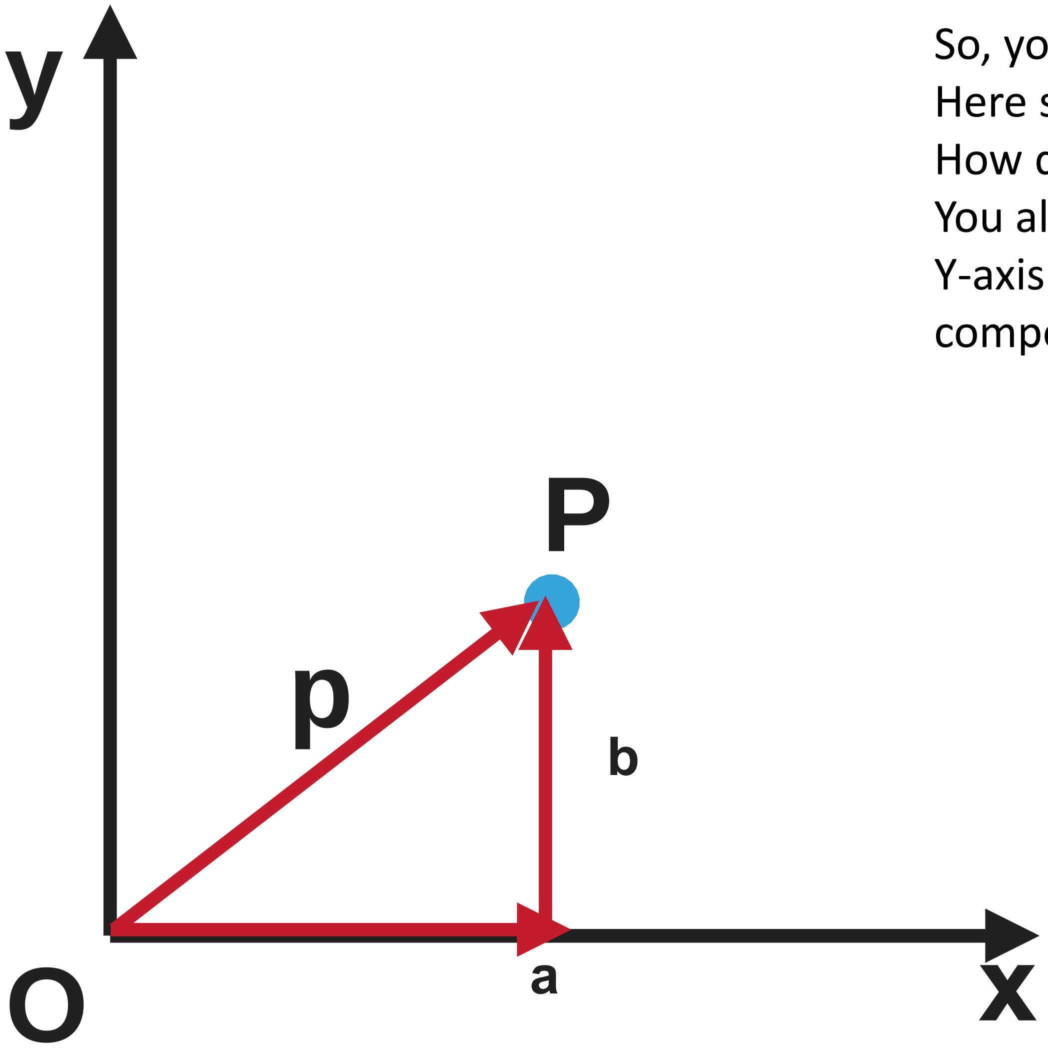
$$p = (a, b)$$

POINT in 2D space



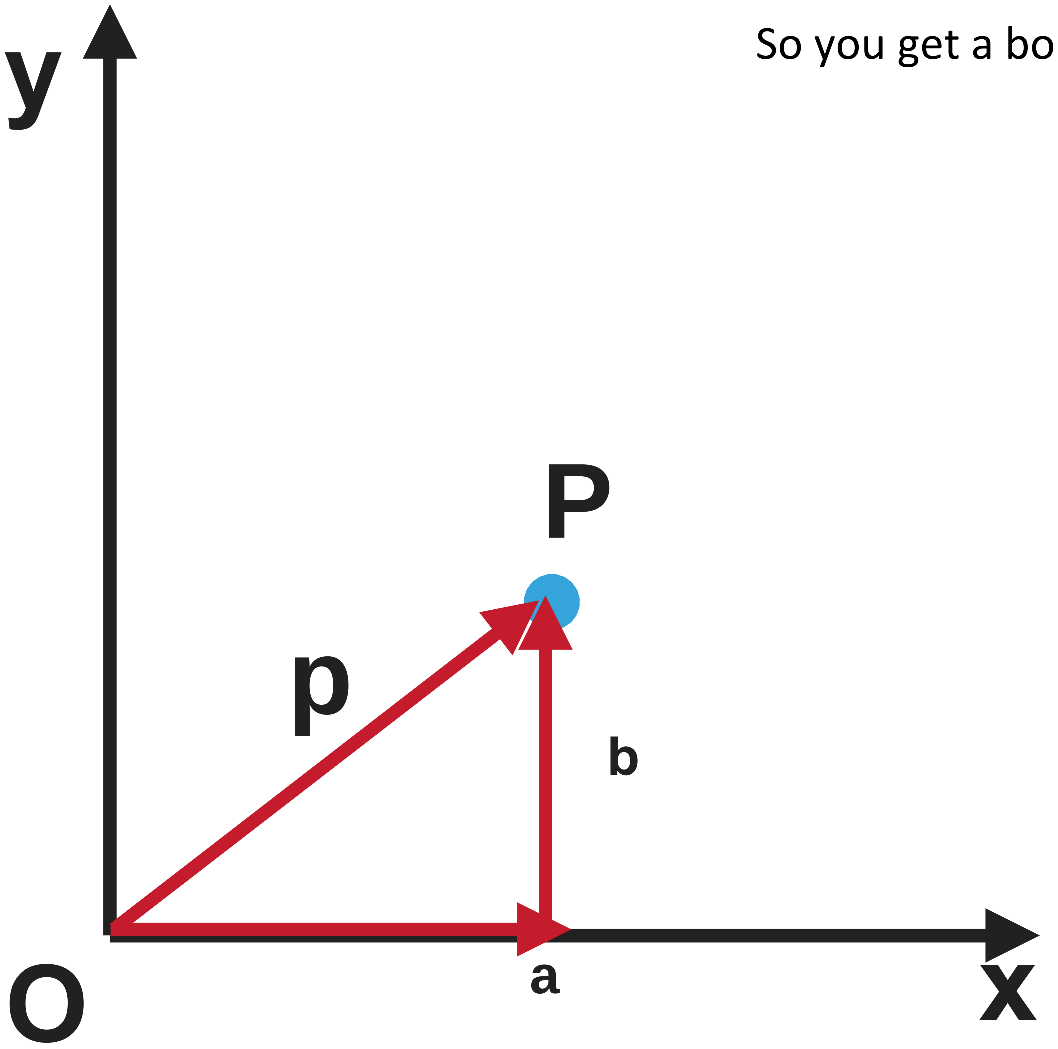
How can you represent in terms of vector?
i.e not only the position, but you also need to know the Orientation or what direction it is actually
i.e in turn knowing the vector representation of the point.

POINT in 2D space



So, you drop a vector from the origin to the point
Here small ‘p’.
How do you find ‘p’
You already dropped a perpendicular to X-axis and Y-axis and taking that magnitude as the parallel component across the X-axis.

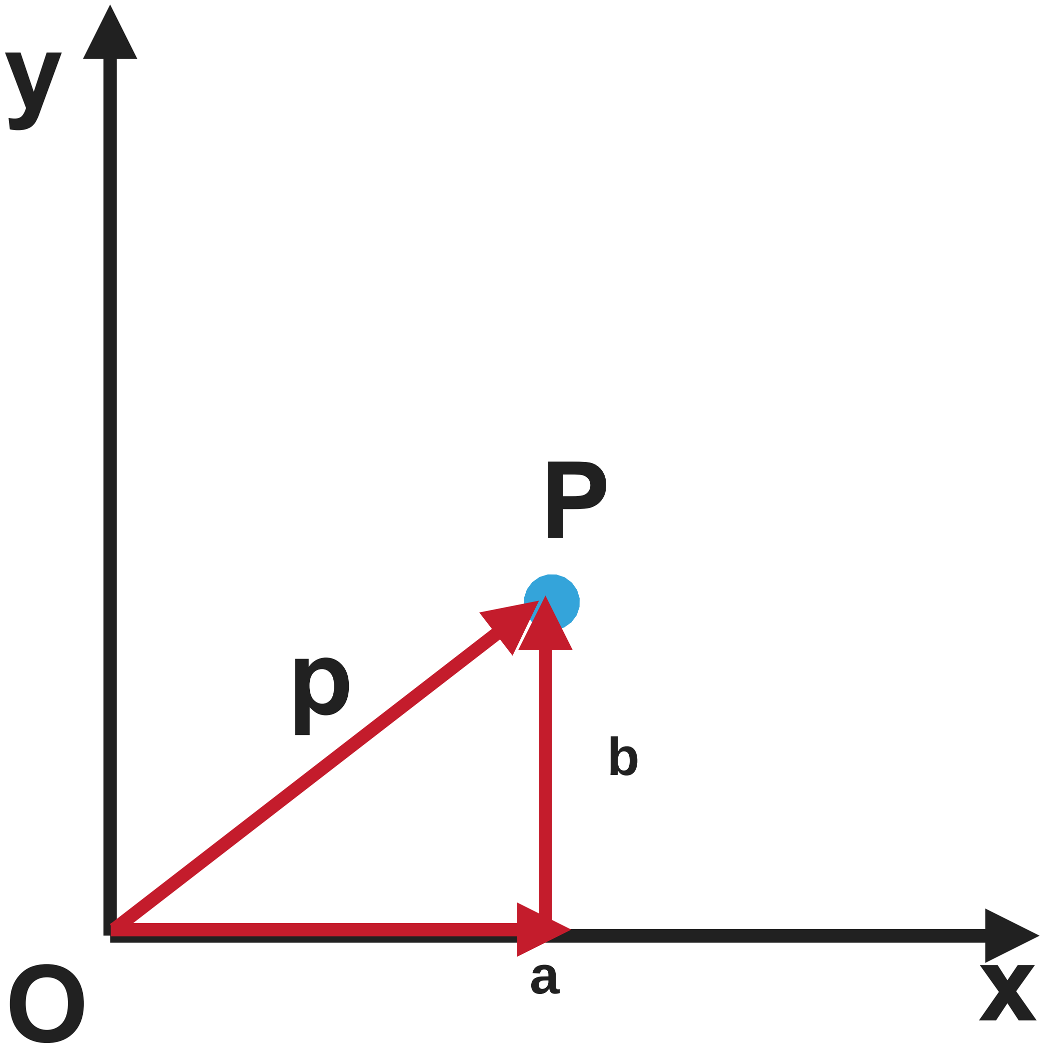
POINT in 2D space



So you get a bound vector

$P = \begin{bmatrix} a \\ b \end{bmatrix}$

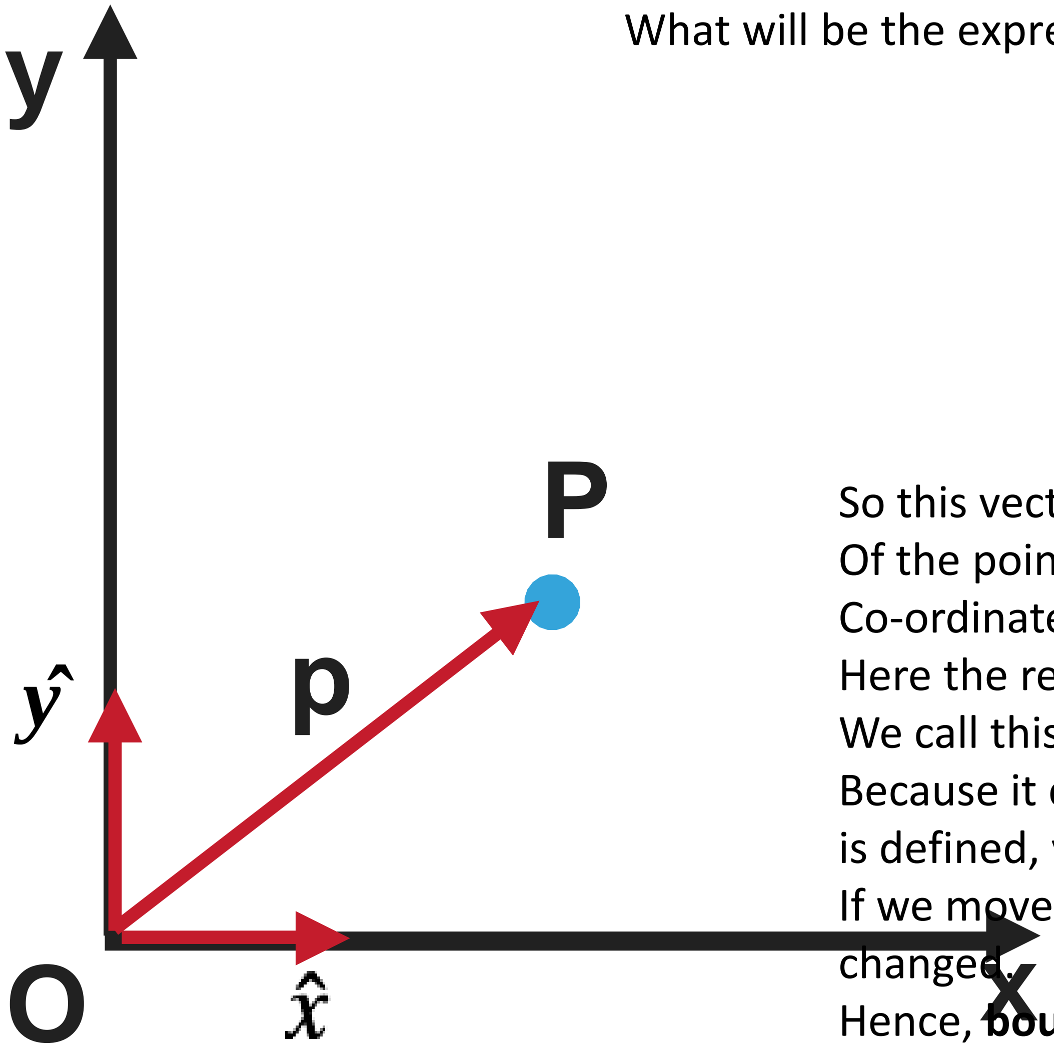
POINT in 2D space



$$P = \begin{bmatrix} a \\ b \end{bmatrix}$$

Bound Vector

POINT in 2D space



What will be the expression of vector ‘p’

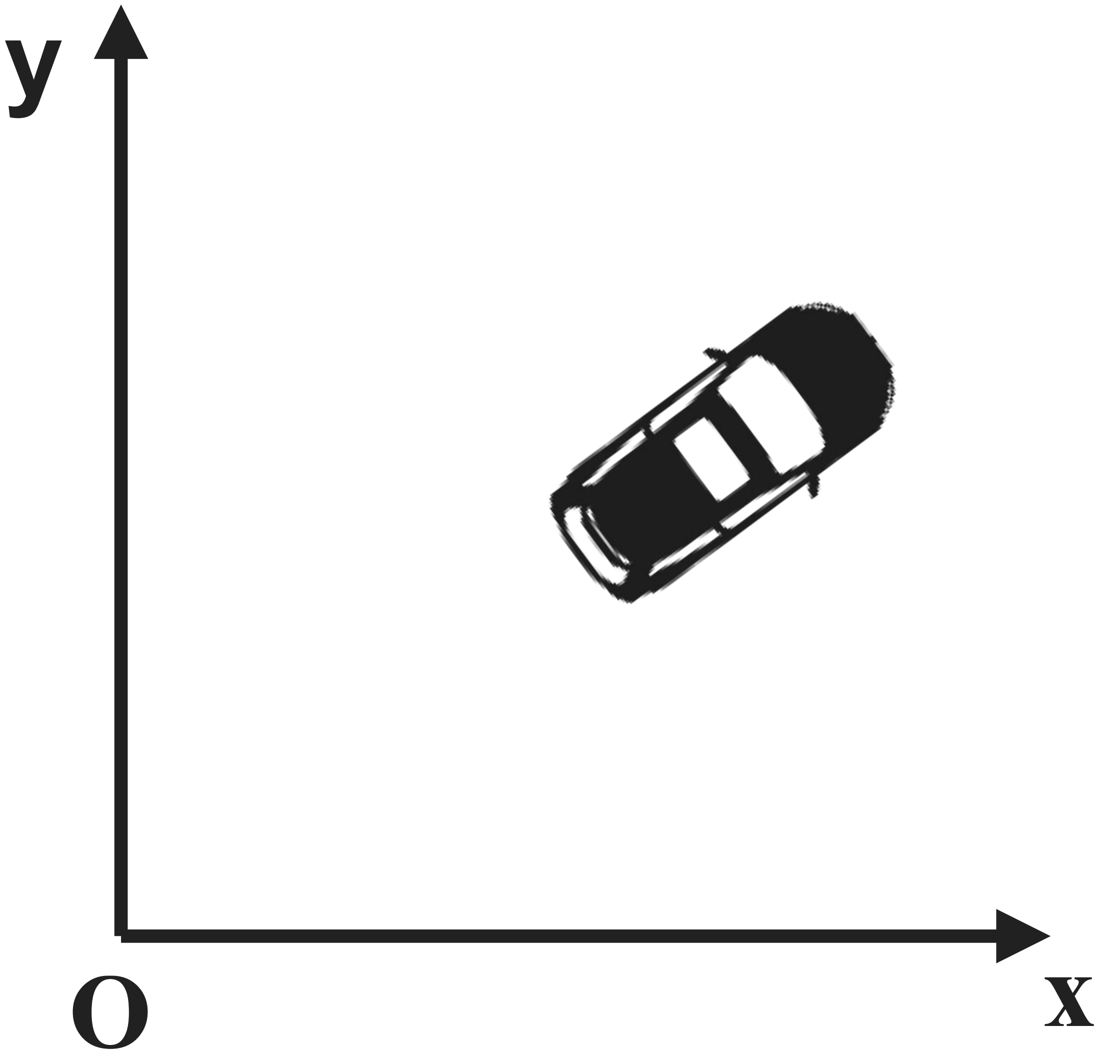
$$\vec{p} = \begin{bmatrix} a \\ b \end{bmatrix}$$

$$p = a\hat{x} + b\hat{y}$$

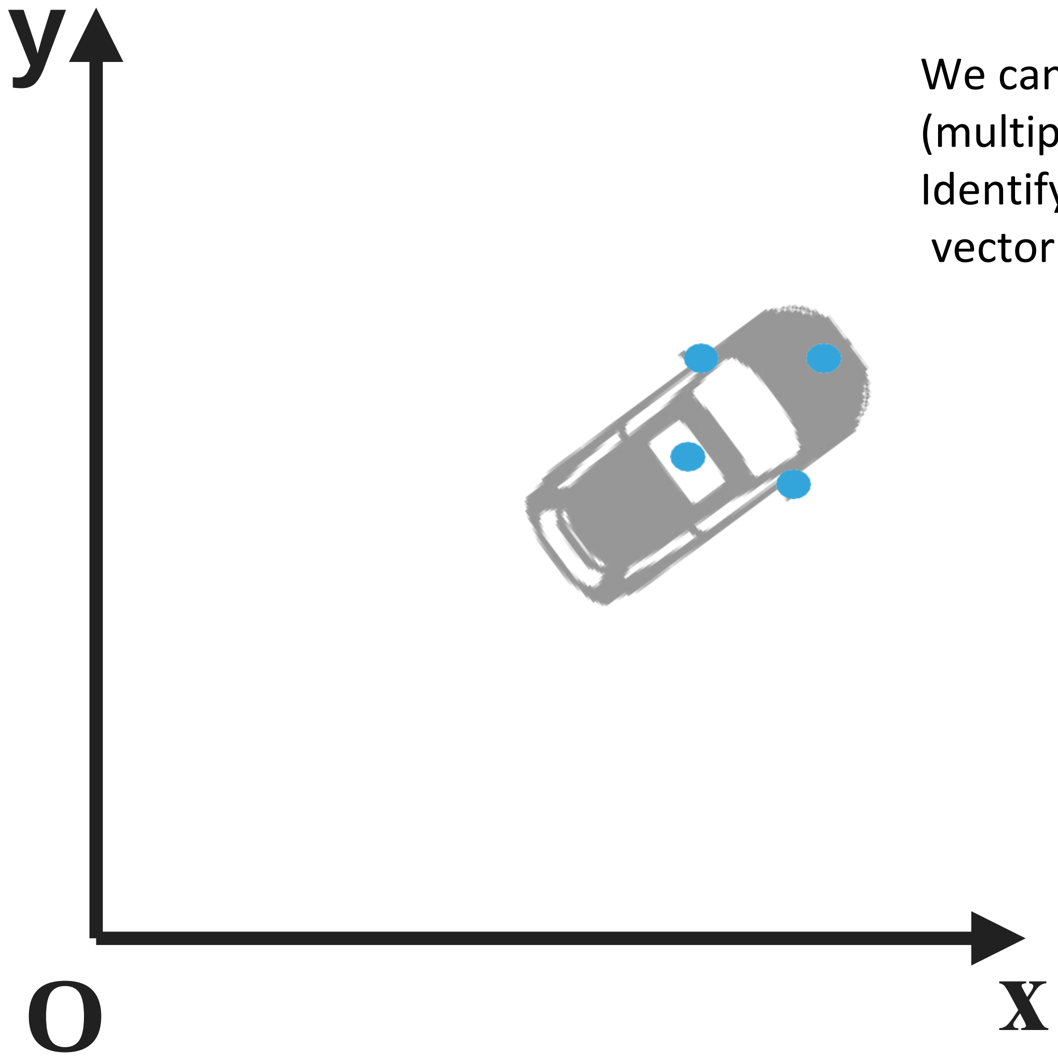
So this vector ‘p’ represents the displacement
Of the point with respect to some reference
Co-ordinate frame.
Here the reference co-ordinate frame is ‘O’
We call this ‘p’ as bound vector
Because it cannot be really moved. Once the vector
is defined, you cannot really move the point.
If we move the point, the vector representation also
changed.
Hence, **bound vector.**

But something is missing here.
We cannot call a robot as a point .

OBJECT in 2D space

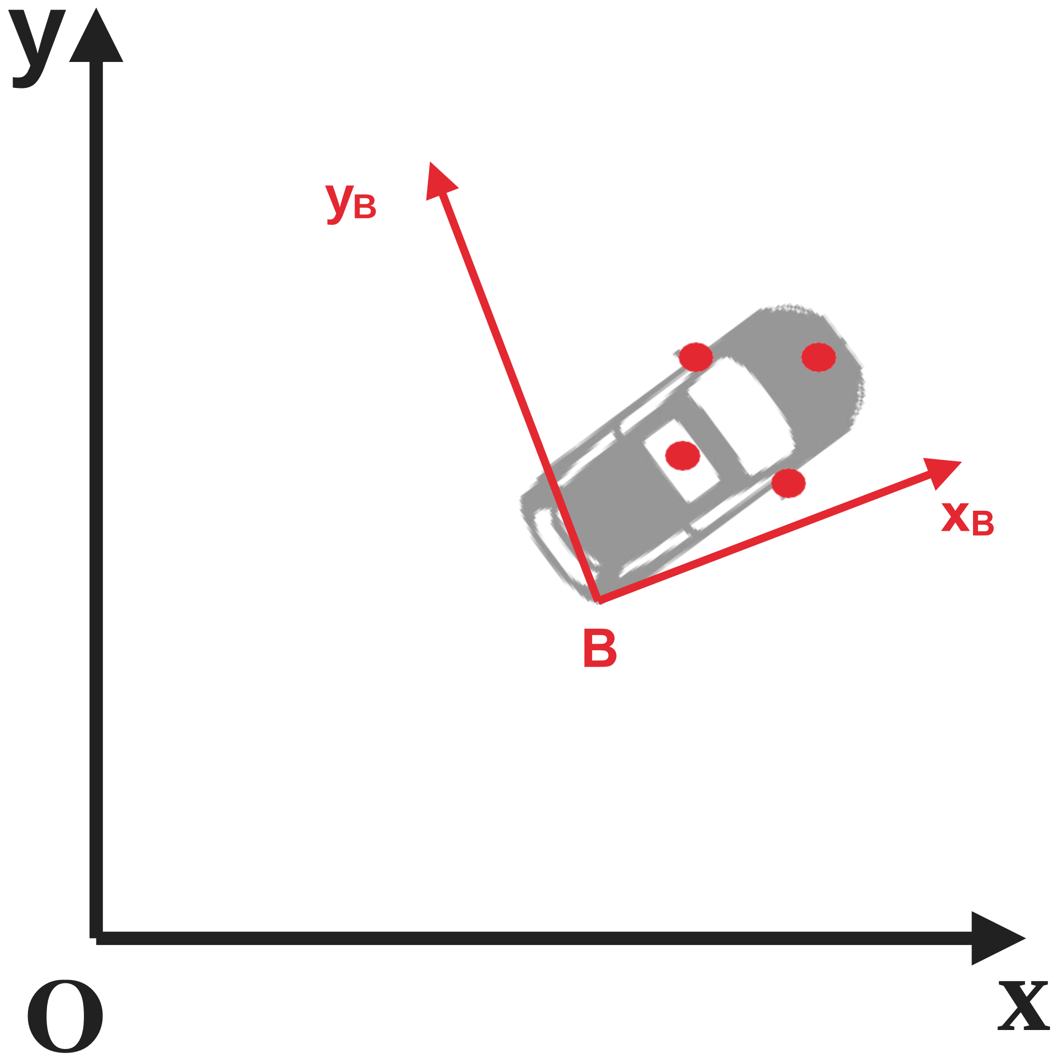


OBJECT in 2D space



We can represent an object as points.
(multiple points), then
Identify the position of those points using
vector analysis.

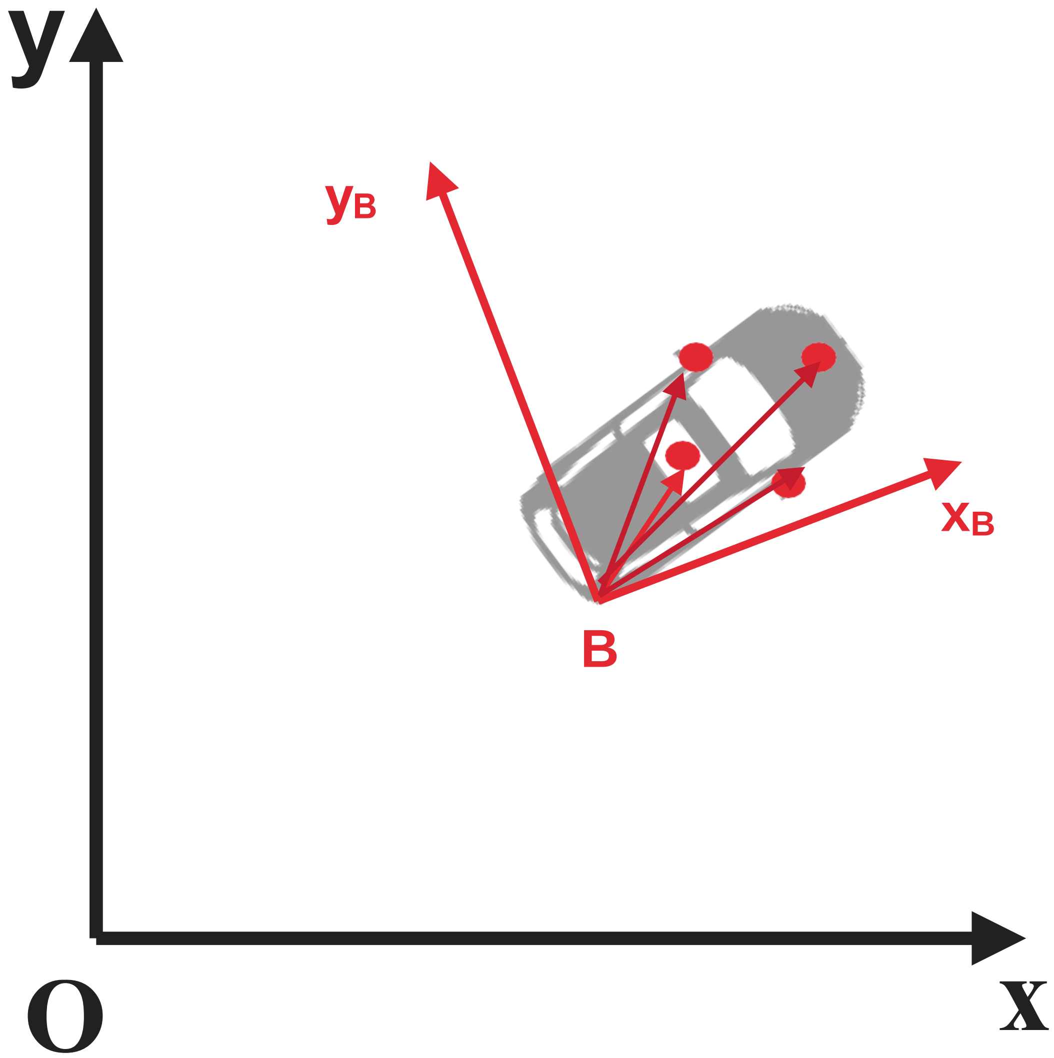
OBJECT in 2D space



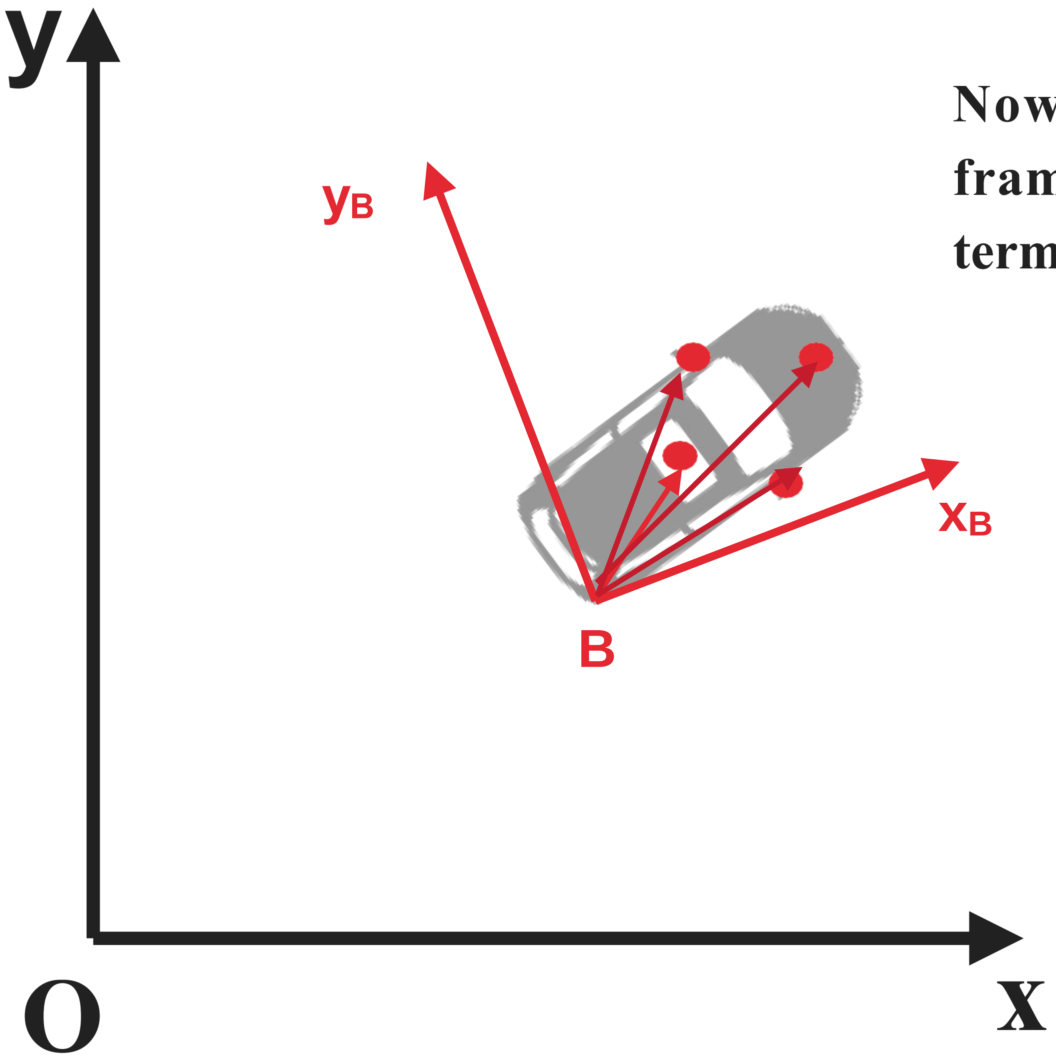
Here also, there is one problem.

The object itself has its own co-ordinate system, called Object coordinate system.

OBJECT in 2D space

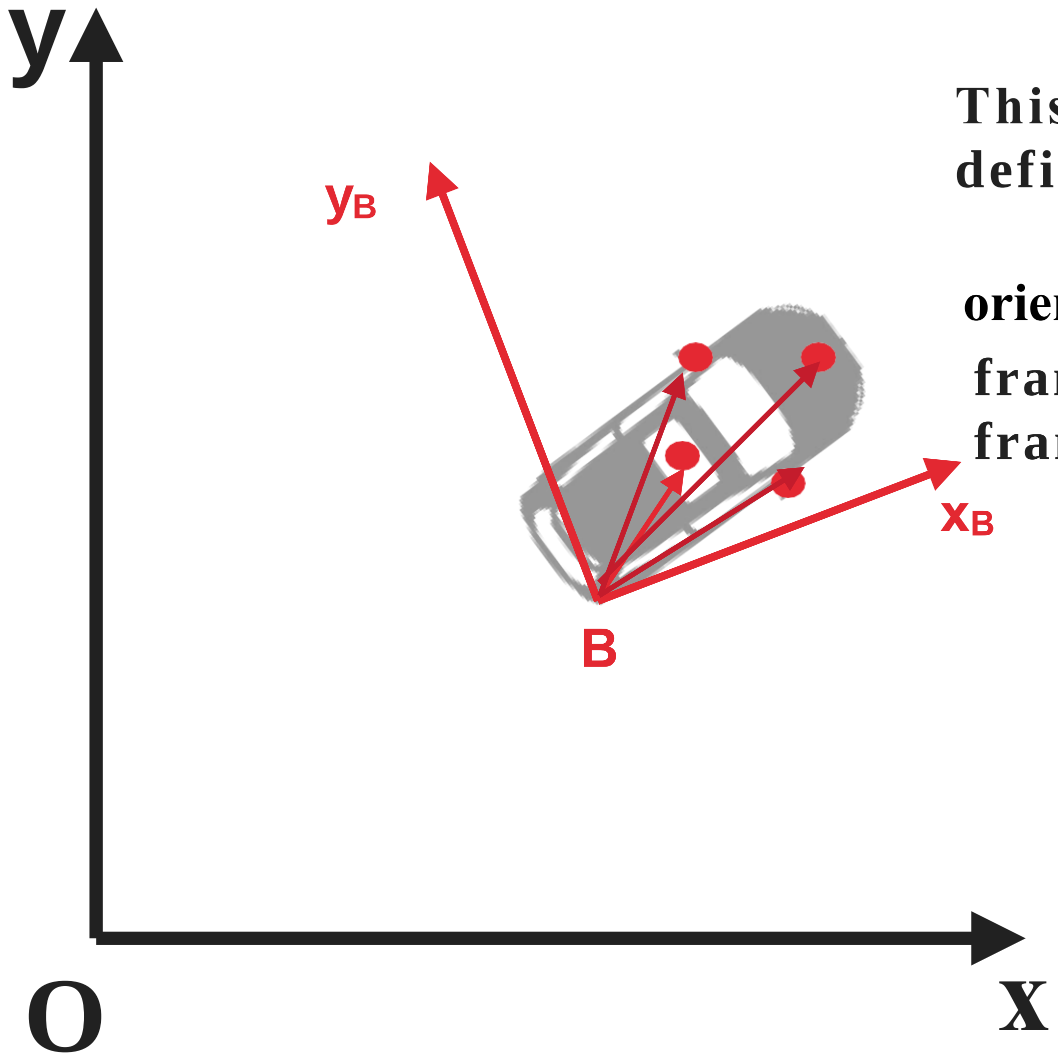


OBJECT in 2D space



Now, the object coordinate frame can be described in terms of world coordinate frame.

OBJECT in 2D space

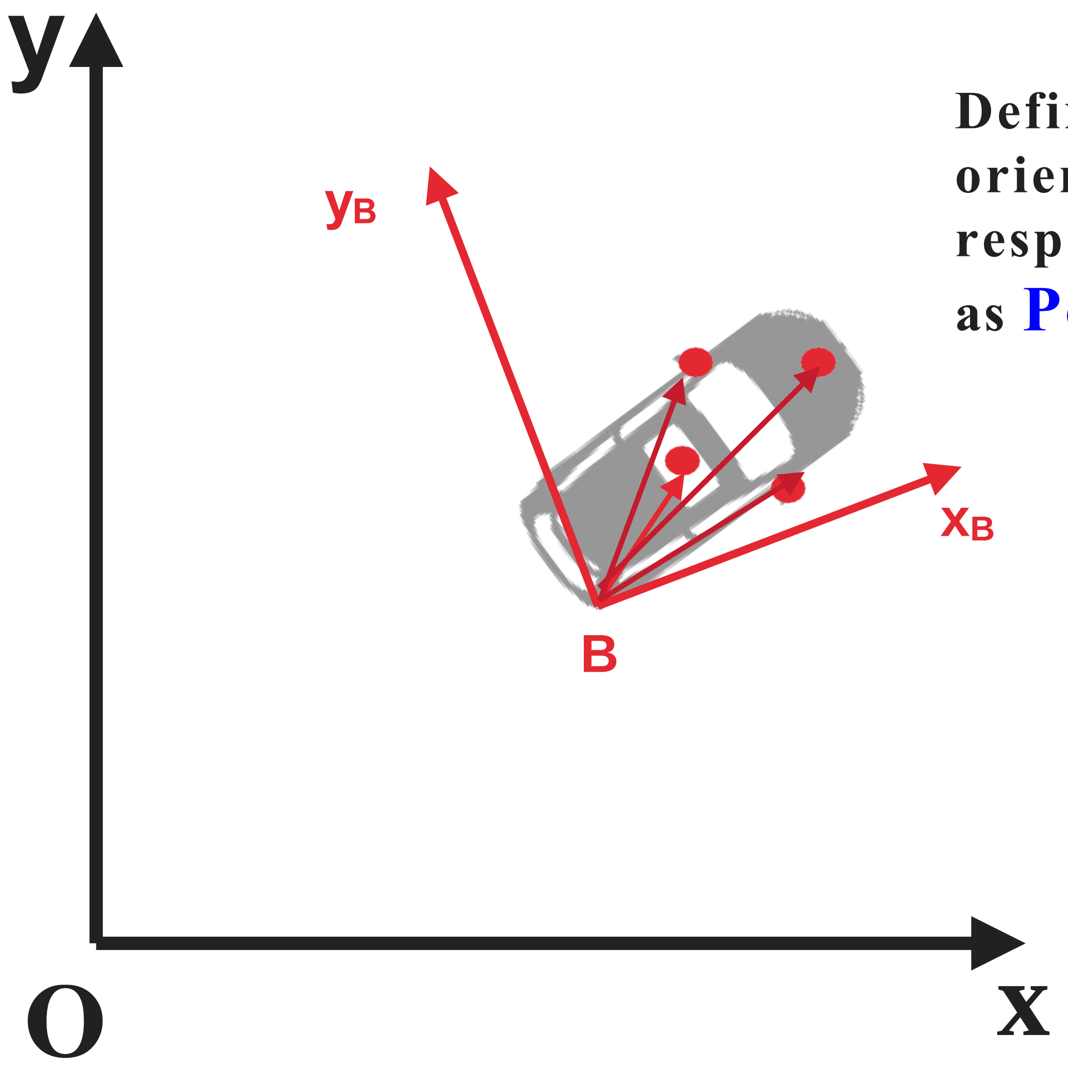


This can be achieved by defining the position and

orientation of the object

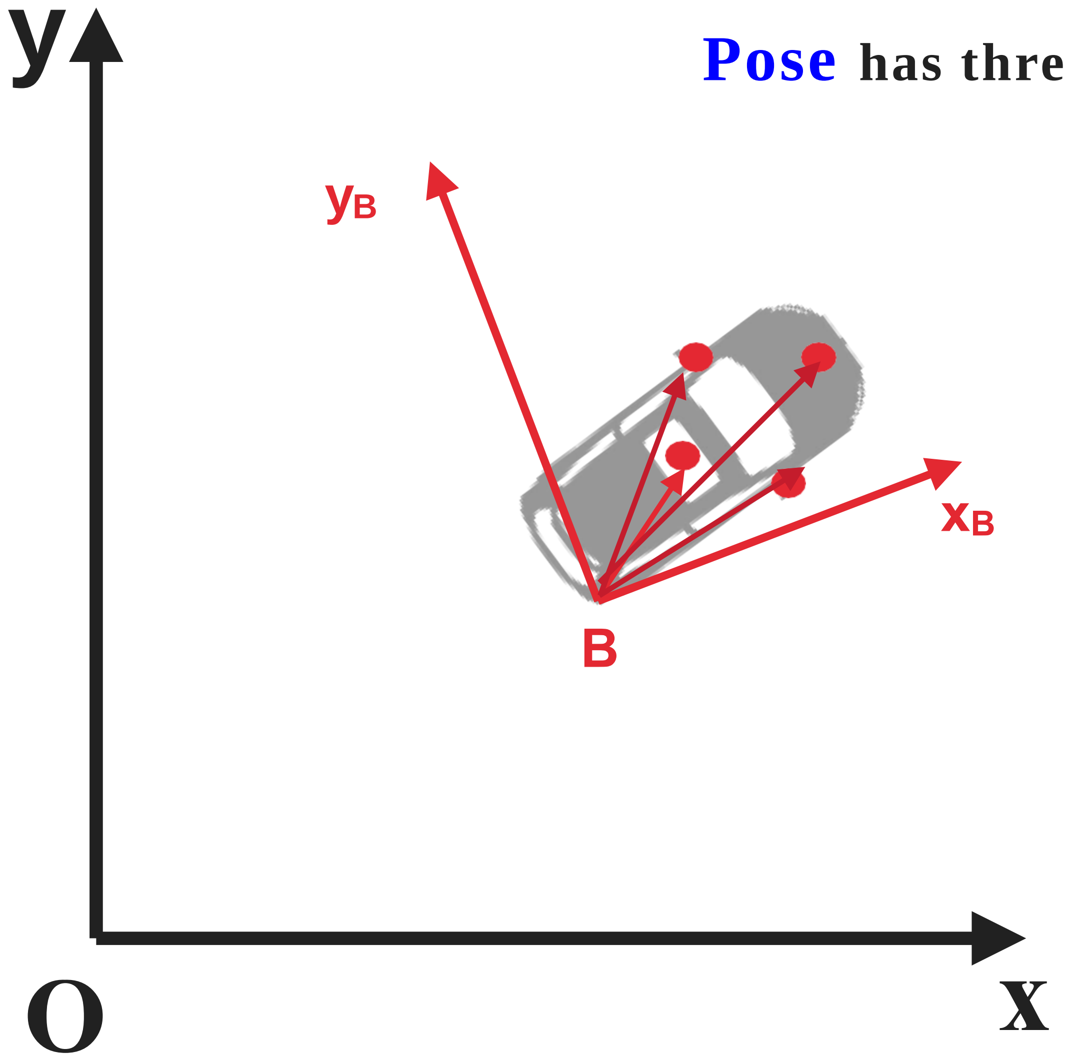
frame in terms of world frame.

POSE in 2D space



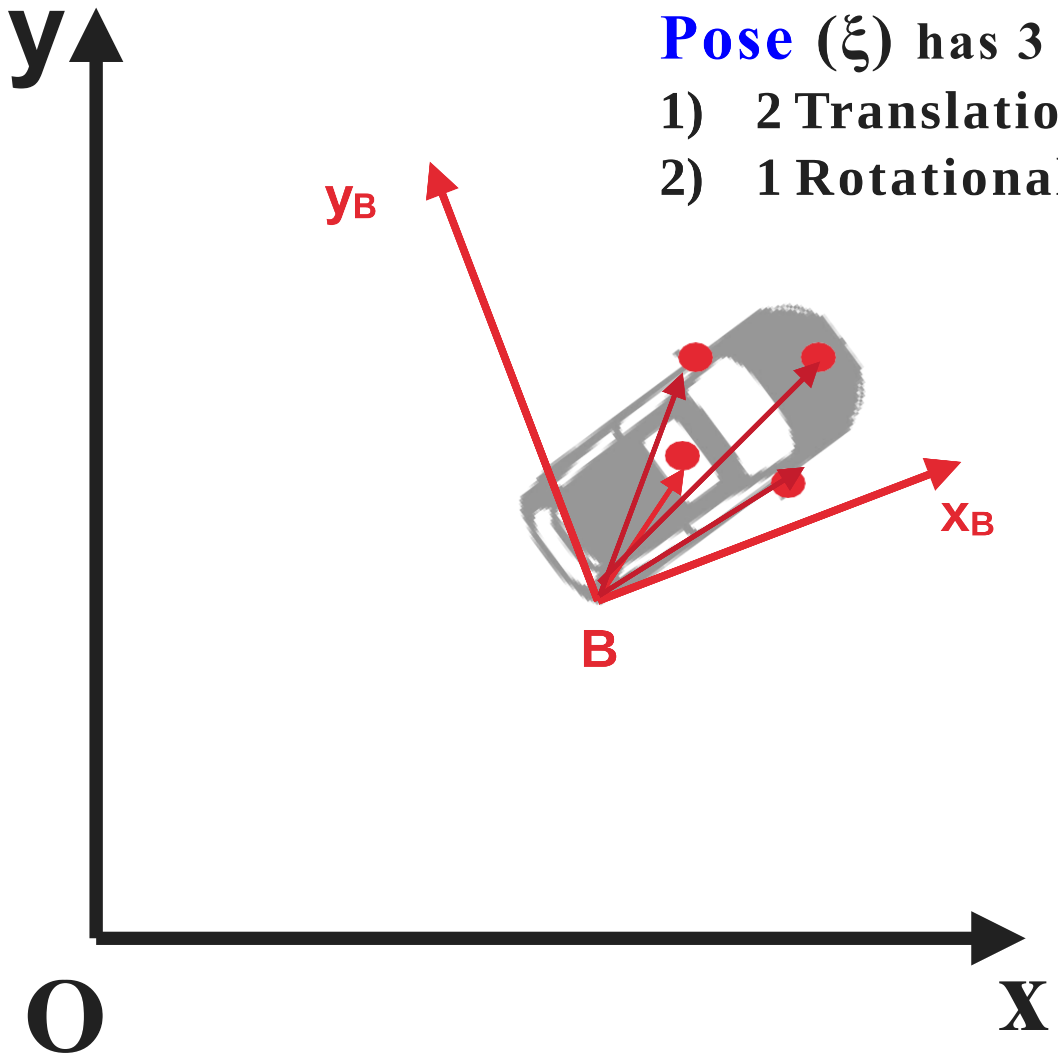
Defining position and orientation of an object with respect to a frame is called as **Pose**.

POSE in 2D space



Pose has three parameters:

POSE in 2D space

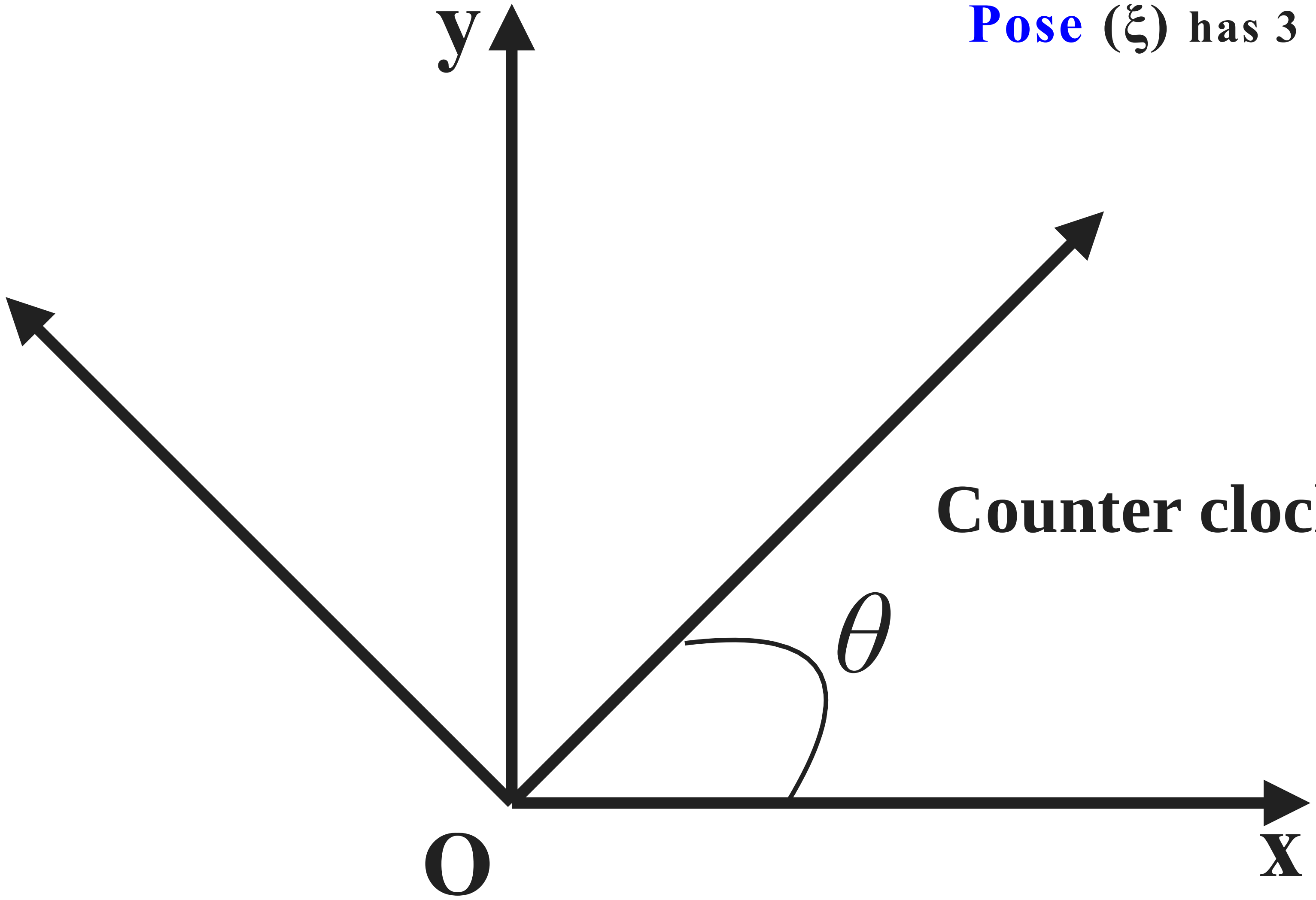


Pose (ξ) **has 3 parameters:**

1) **2 Translational Components** $\rightarrow x, y$
2) **1 Rotational Component** $\rightarrow \theta$

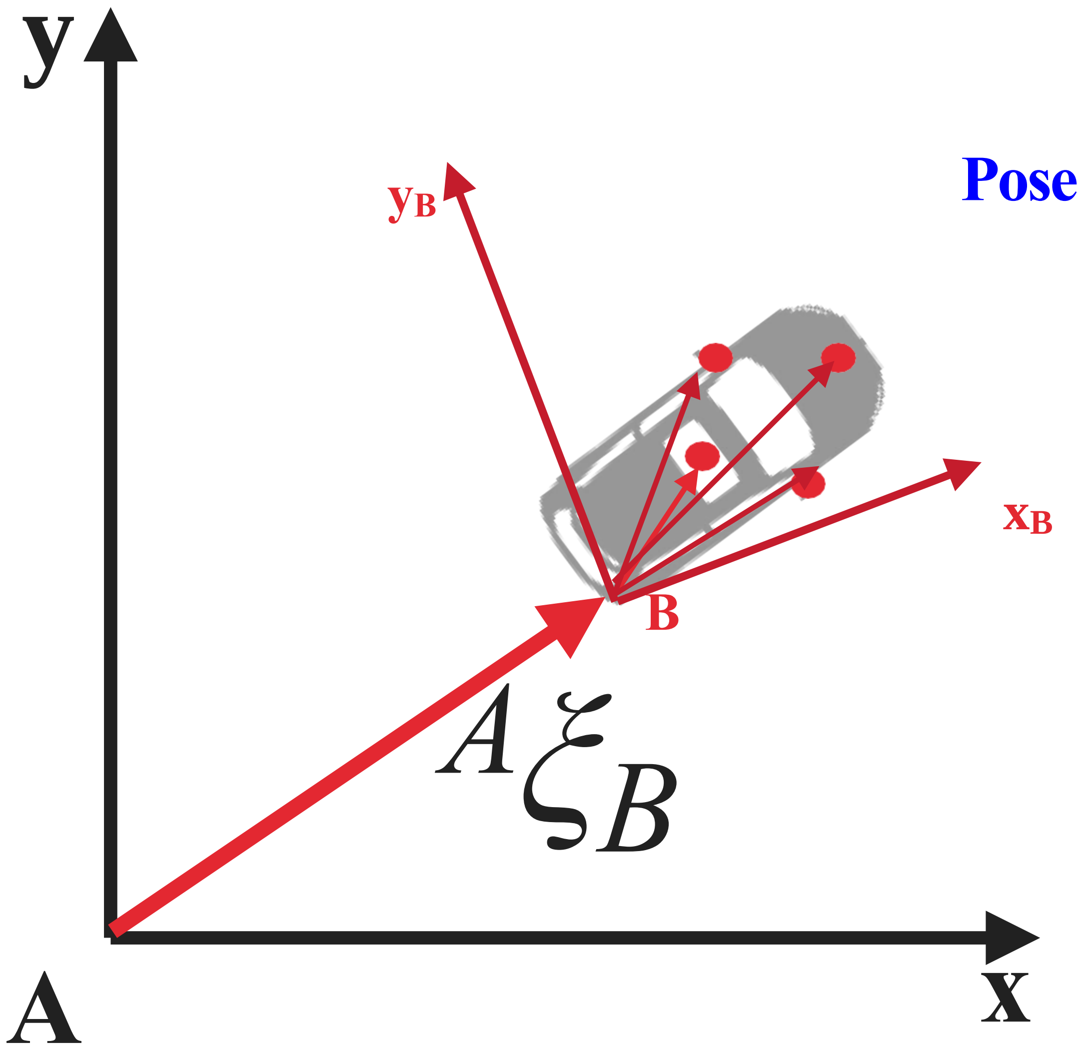
POSE in 2D space

Pose **(ξ) has 3 parameters: (x, y, θ)**



Counter clockwise rotation + θ

Relative POSE in 2D space



Pose (${}^{A}\xi_{B}$) **of frame{B} with respect to frame{A}.**

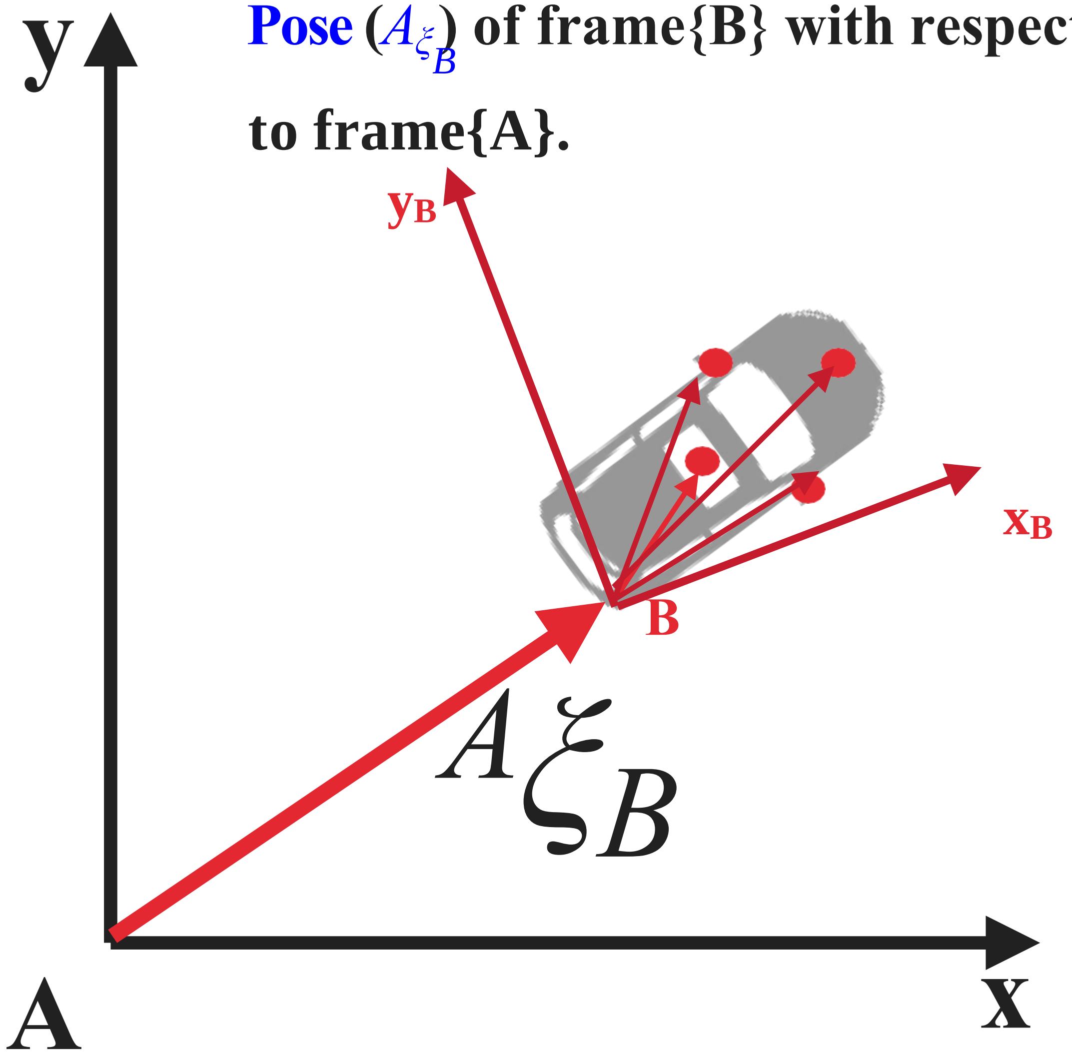
What is the pose of frame 'B' wrt frame 'A'

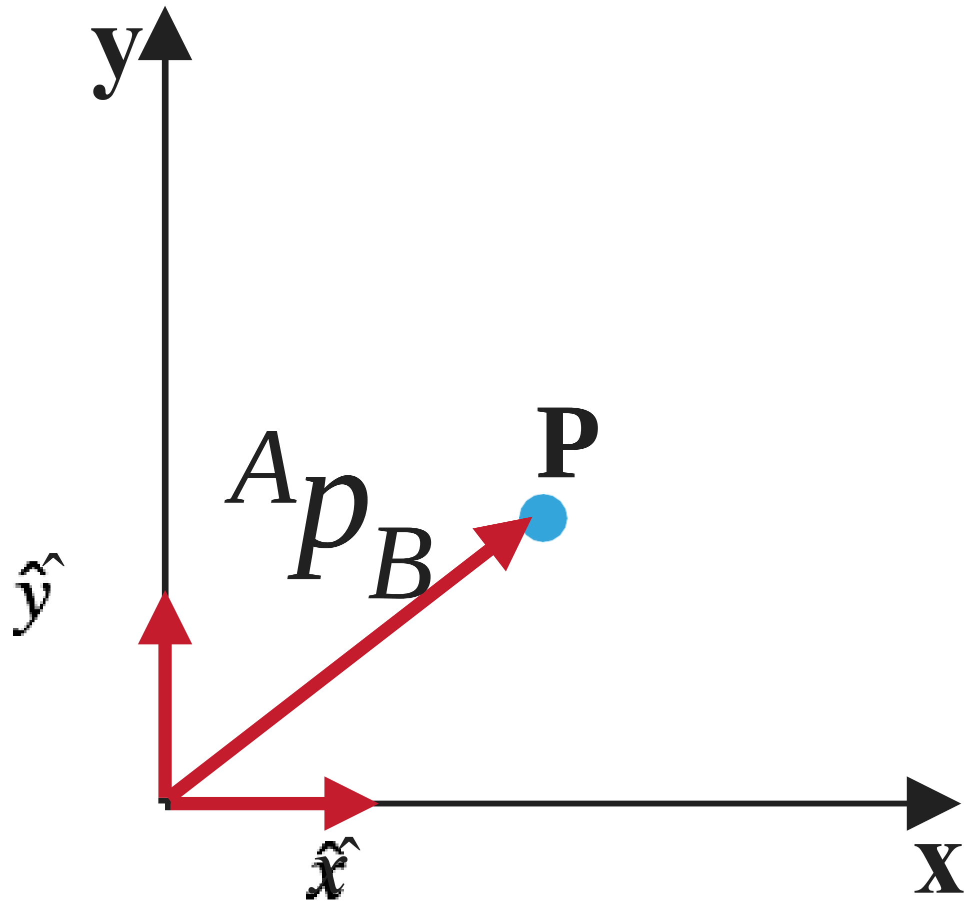
Find translational component and rotational component individually

Relative POSE in 2D space

$\{A\}$ $\{B\}$
ξ x_i

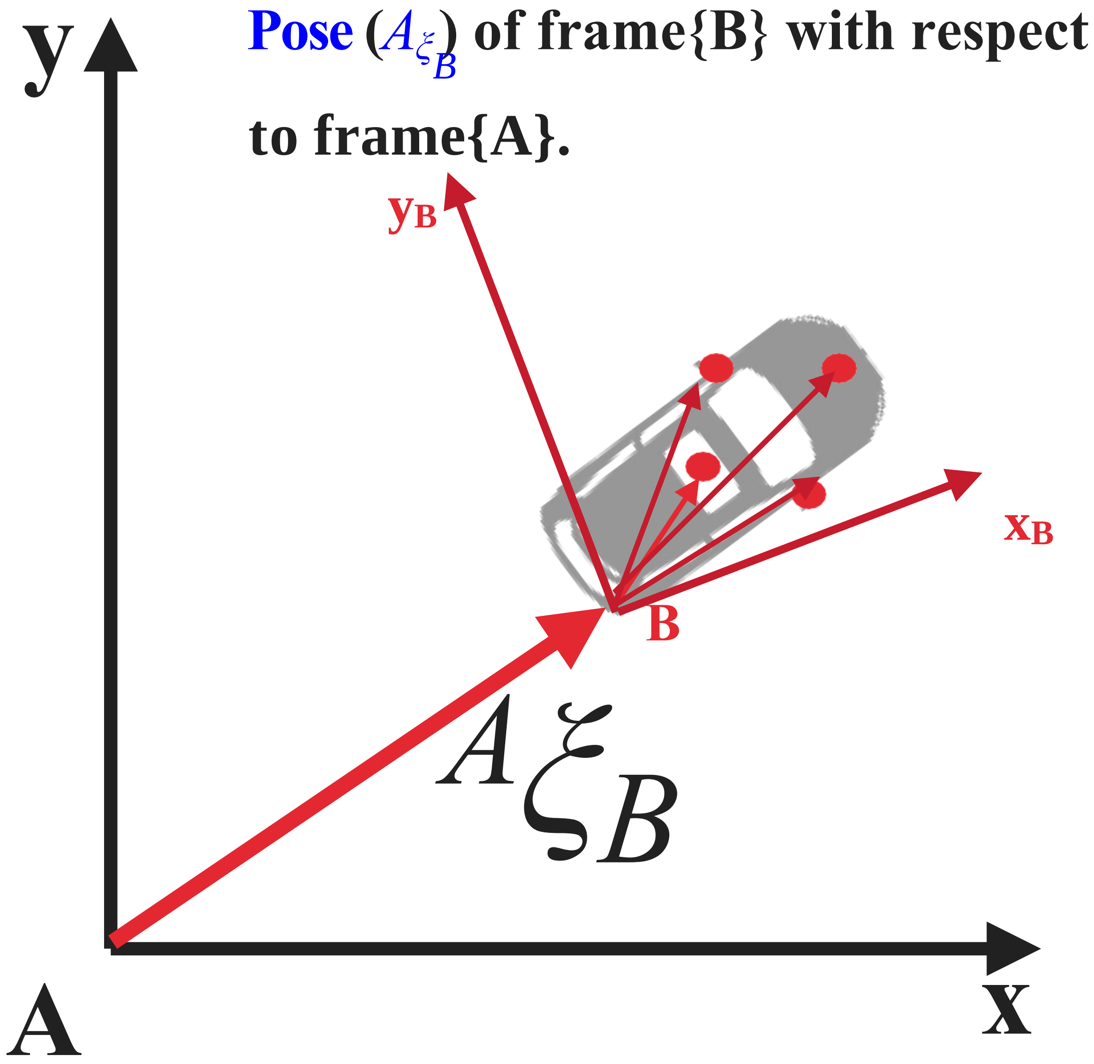
Pose ($^{A}\xi_{B}$) **of frame{B} with respect to frame{A}.**

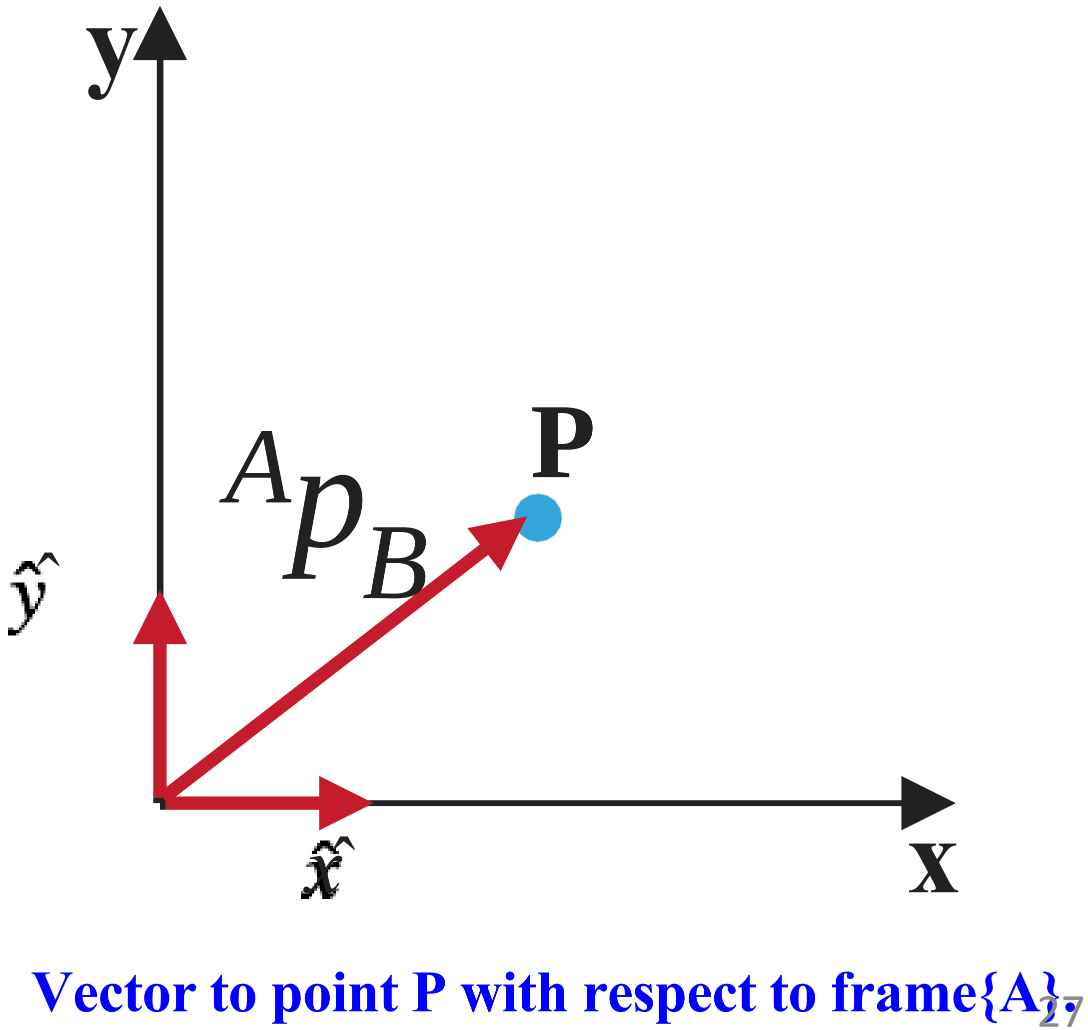




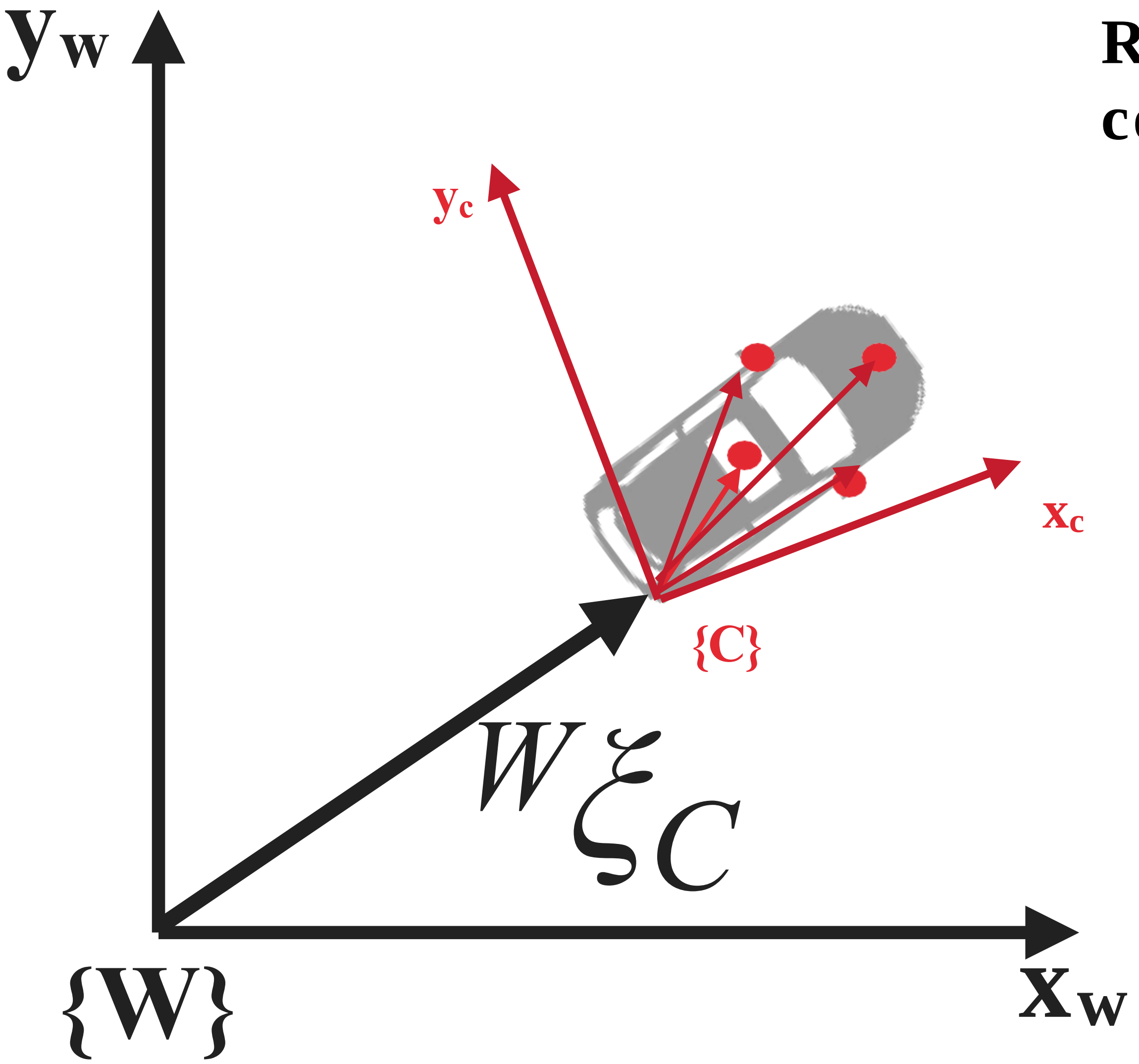
Vector to point P with respect to frame{A}.

Relative POSE in 2D space



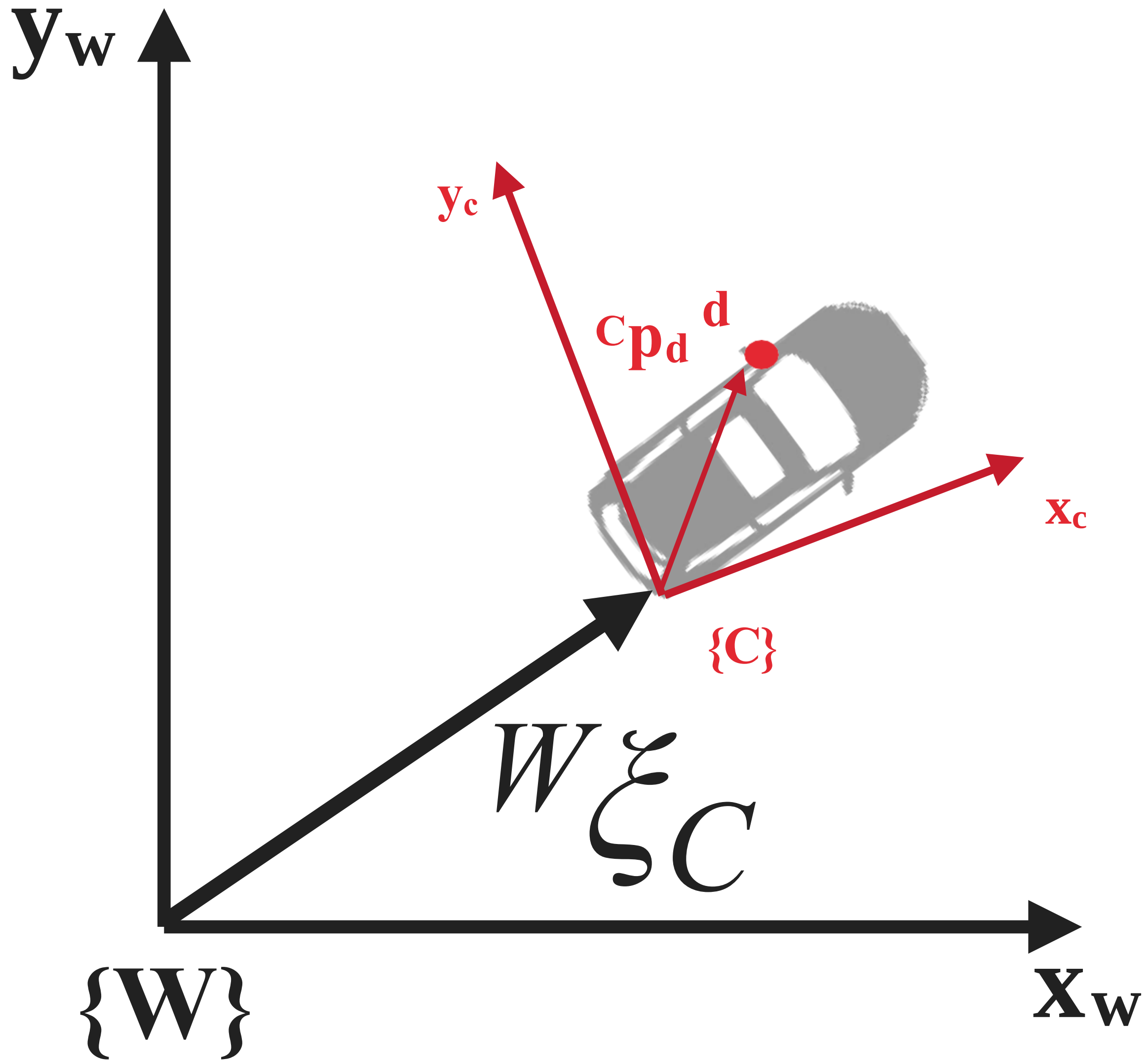


Relative POSE in 2D space

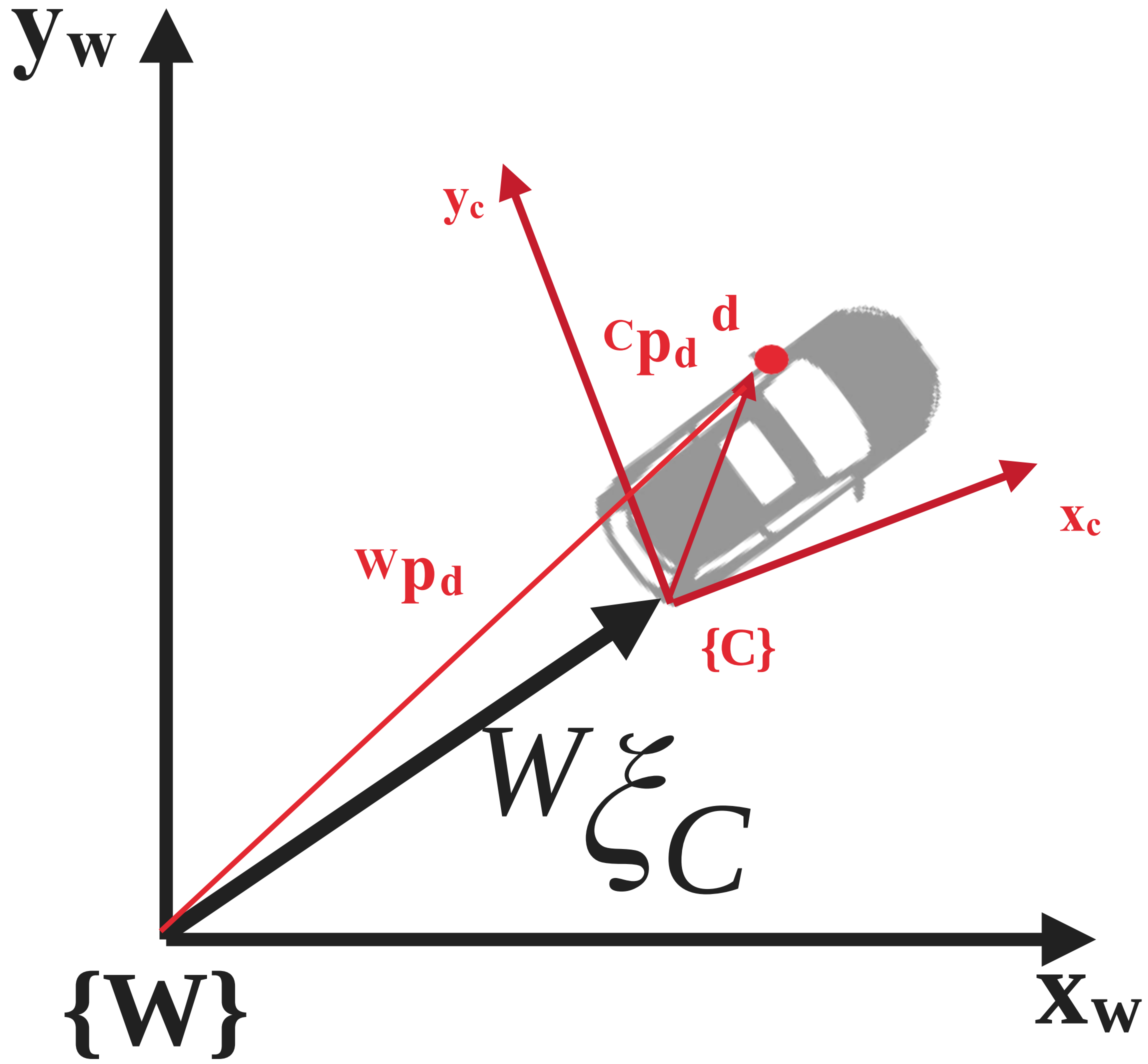


Renaming frame{A} as frame{W}, world coordinate frame{B} to frame{C}.

Relative POSE in 2D space

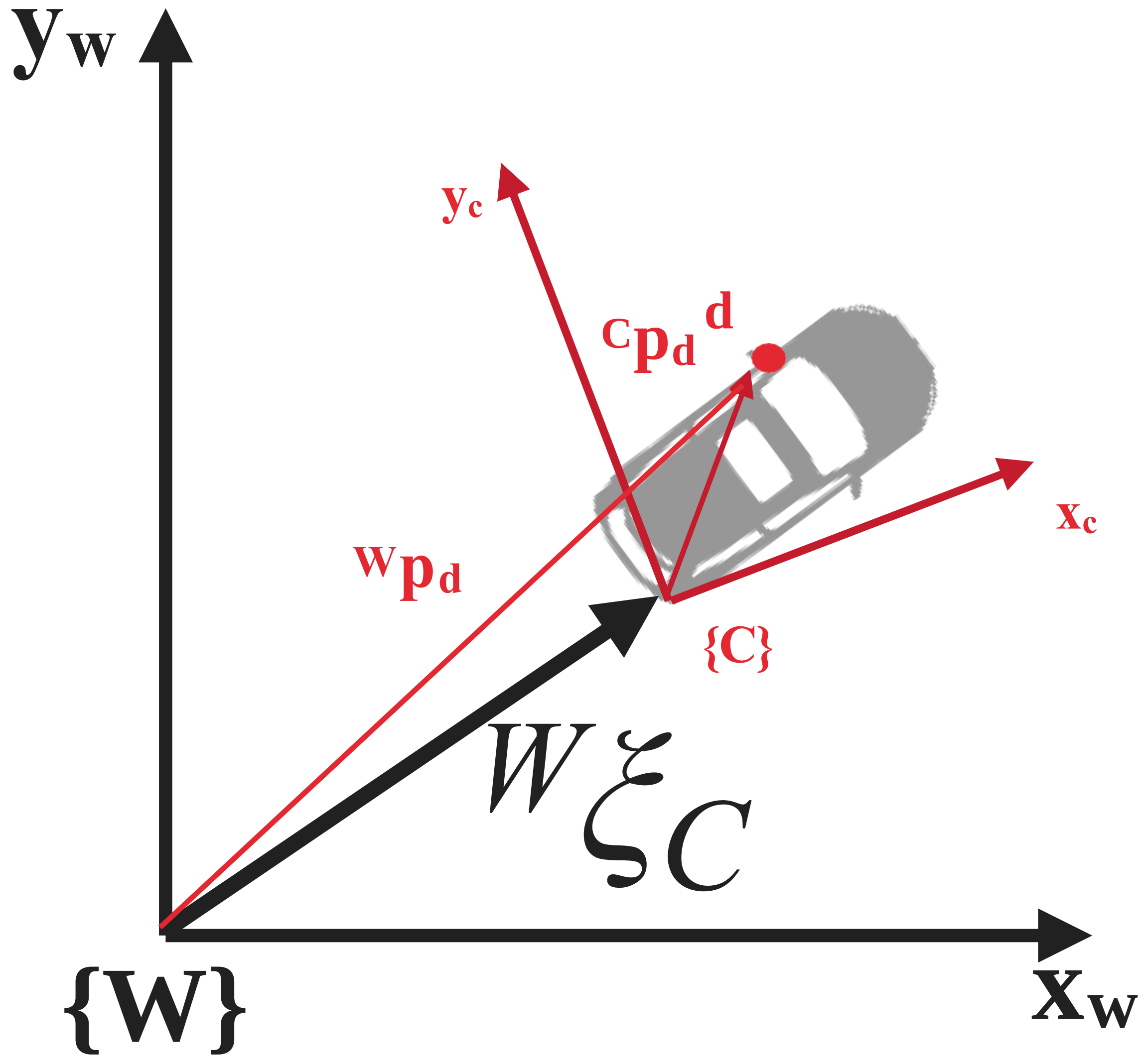


Relative POSE in 2D space



Can we add a vector and a pose ??

Relative POSE in 2D space

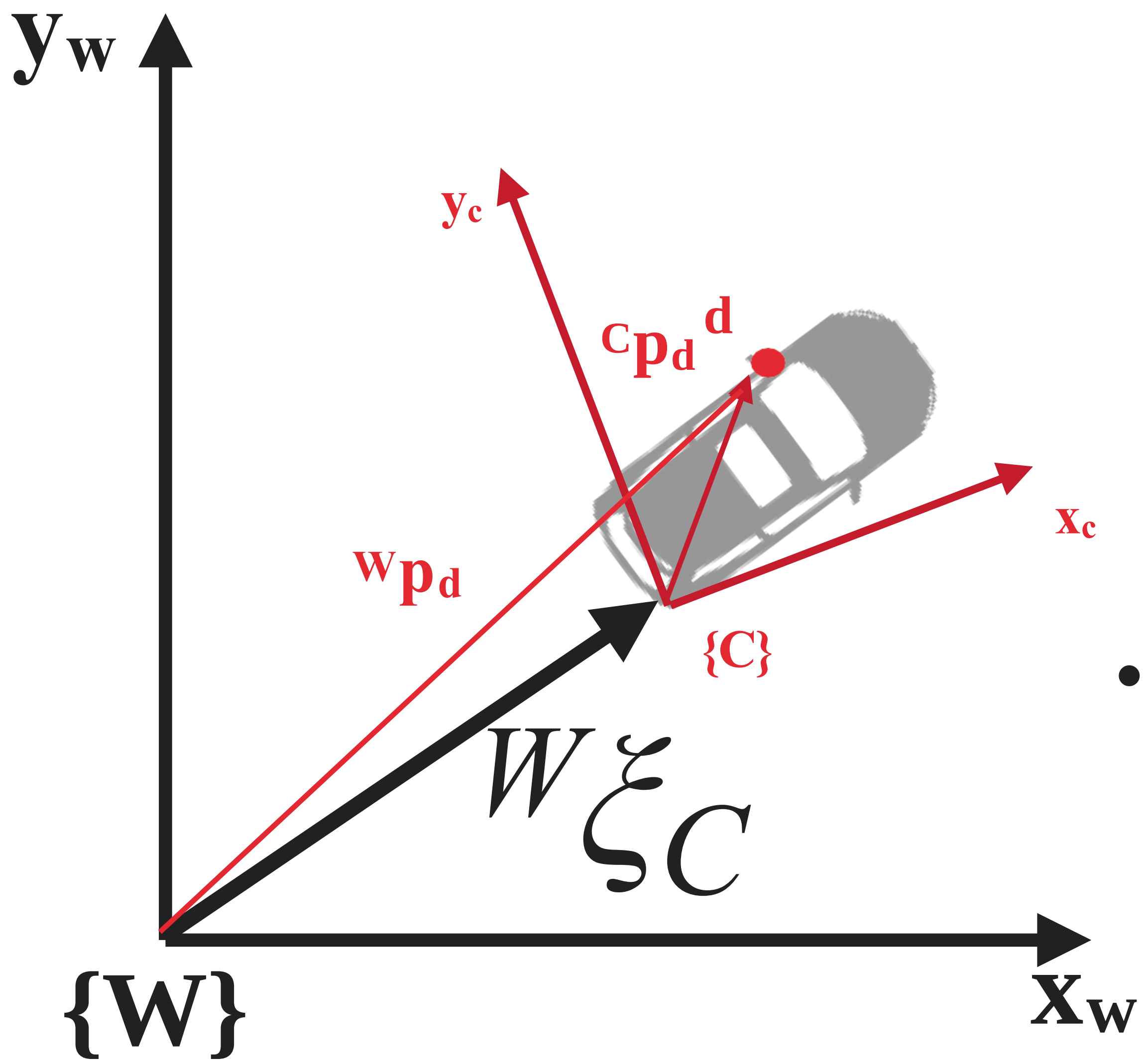


Can we add a vector and a pose ??

$$ {}^w p_d = {}^W \xi_C \bullet {}^C p_d $$

$\cancel{{}^w \xi}$

Relative POSE in 2D space



Can we add a vector and a pose ??

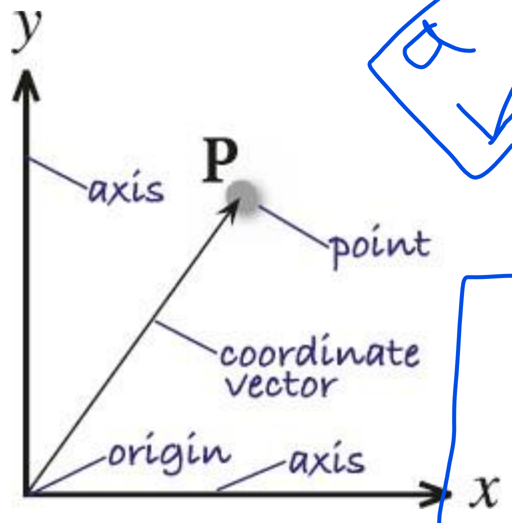
$${}^{w}p_d = {}^{w}\xi_C \bullet {}^{C}p_d$$

$\cancel{{}^{w}\xi}$

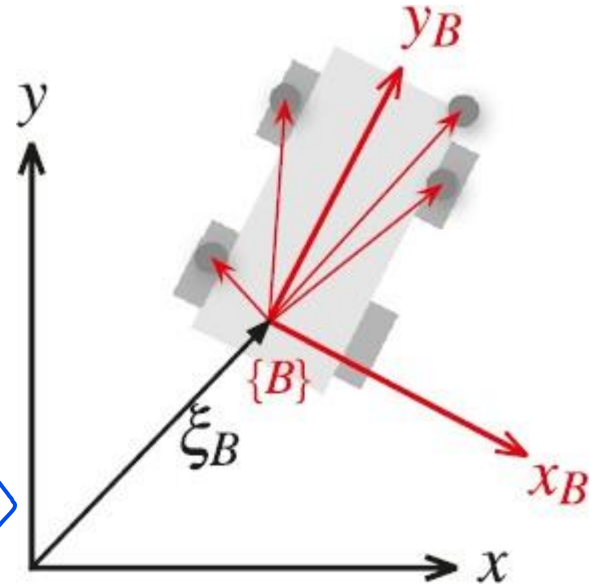
- → **transforms vector from one frame to another.**

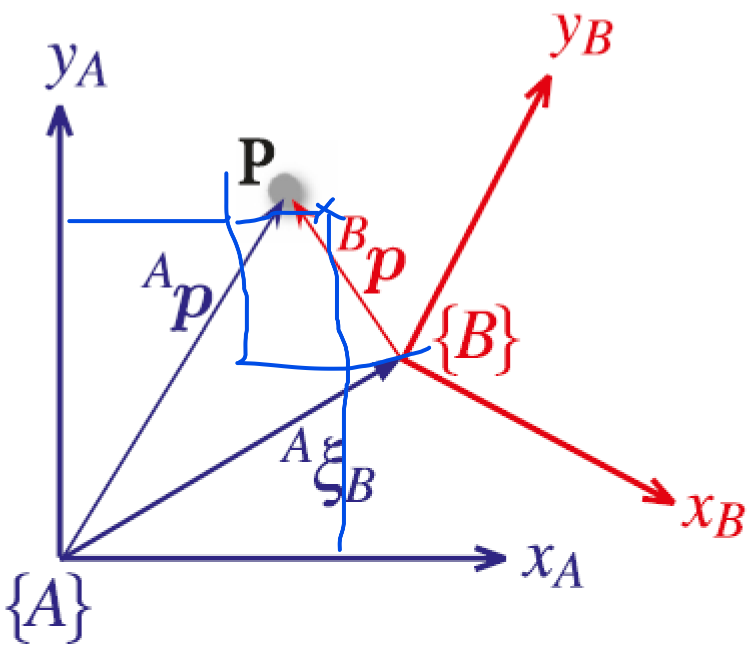
Quick Summary

- Object.
- Scalar & Vector.
- Pose.
- Point.
- Co-ordinate frame.
- Origin.



Point vs. Object





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

Working in Two Dimensions (2D)

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

- We can use a column vector (a 2x1 matrix) to represent a 2D point $\begin{vmatrix} x \\ y \end{vmatrix}$
- A general form of *linear* transformation can be written as:

$$x' = ax + by + c$$

OR

$$\begin{vmatrix} X' \\ Y' \\ 1 \end{vmatrix} = \begin{vmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{vmatrix} * \begin{vmatrix} x \\ y \\ 1 \end{vmatrix}$$

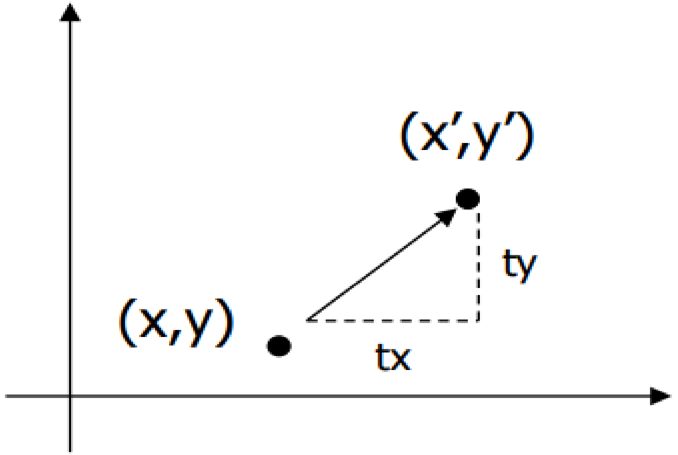
$$y' = dx + ey + f$$

Translation

- Re-position a point along a straight line
- Given a point (x,y), and the translation distance (tx,ty)

The new point: (x′, y′)

$$x' = x + tx$$
$$y' = y + ty$$



OR $P' = P + T$ where $P' = \begin{vmatrix} x' \\ y' \end{vmatrix}$ $p = \begin{vmatrix} x \\ y \end{vmatrix}$ $T = \begin{vmatrix} tx \\ ty \end{vmatrix}$

3 X 3 2D Translation Matrix

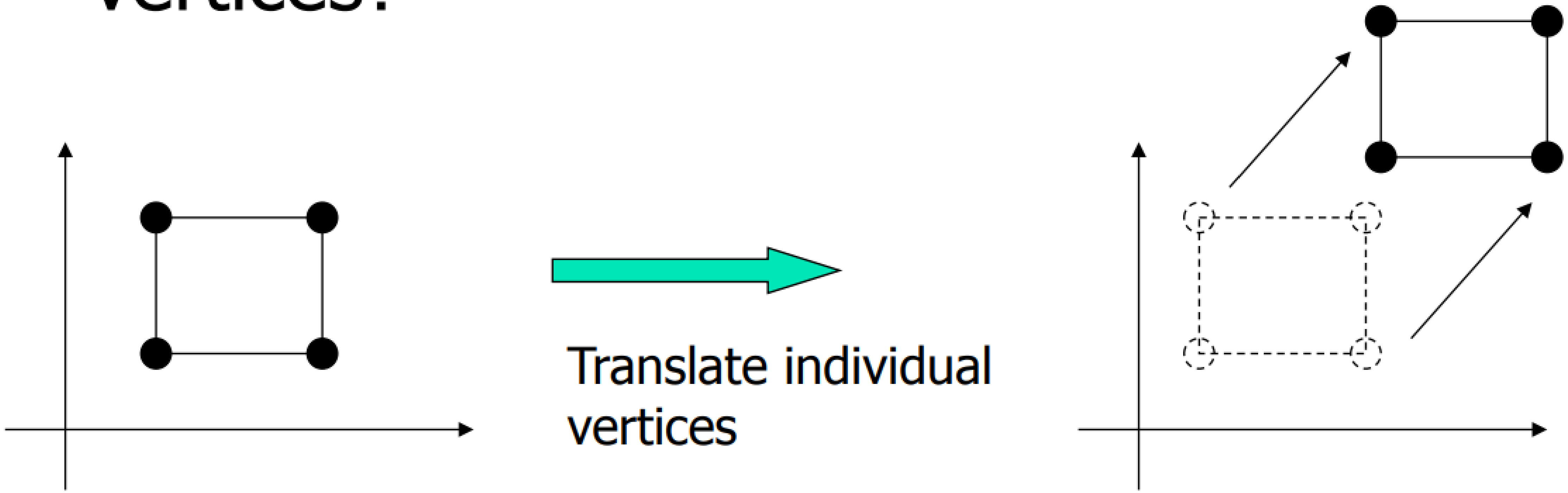
$$\begin{vmatrix} x' \\ y' \end{vmatrix} = \begin{vmatrix} x \\ y \end{vmatrix} + \begin{vmatrix} tx \\ ty \end{vmatrix}$$

Use 3 x 1 vector

$$\begin{vmatrix} x' \\ y' \\ 1 \end{vmatrix} = \begin{vmatrix} 1 & 0 & tx \\ 0 & 1 & ty \\ 0 & 0 & 1 \end{vmatrix} * \begin{vmatrix} x \\ y \\ 1 \end{vmatrix}$$

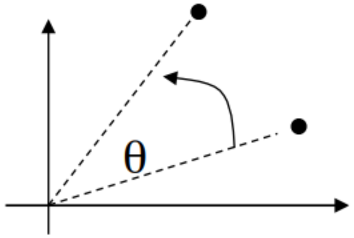
- Note that now it becomes a matrix-vector multiplication

- How to translate an object with multiple vertices?



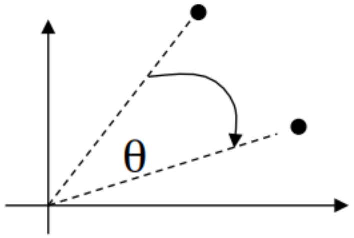
2D Rotation

- Default rotation center: Origin (0,0)



$\theta > 0$: Rotate counter clockwise

positive

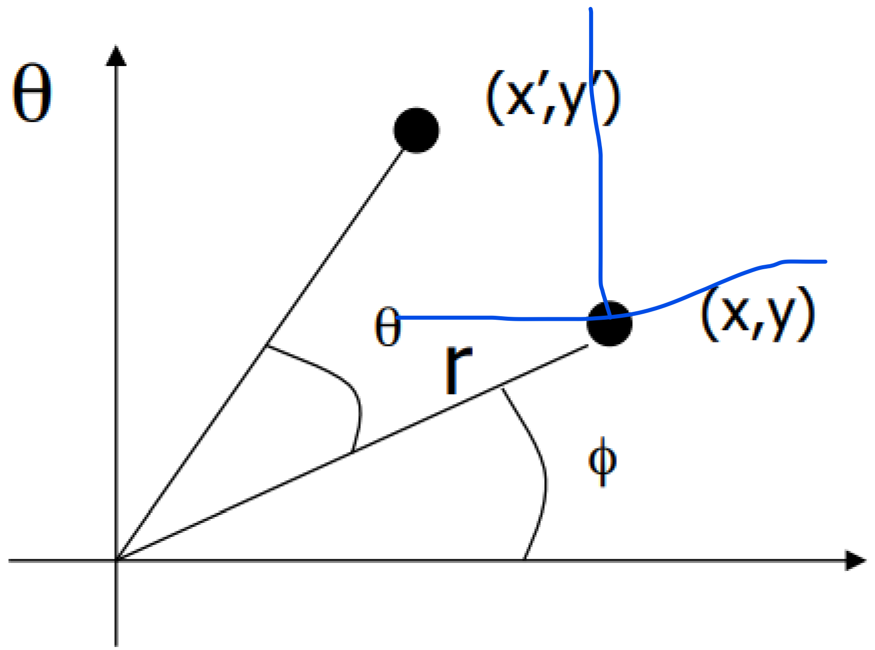


$\theta < 0$: Rotate clockwise

negative

(x,y) -> Rotate *about the origin* by θ

$\longrightarrow$ (x', y')



How to compute (x', y') ?

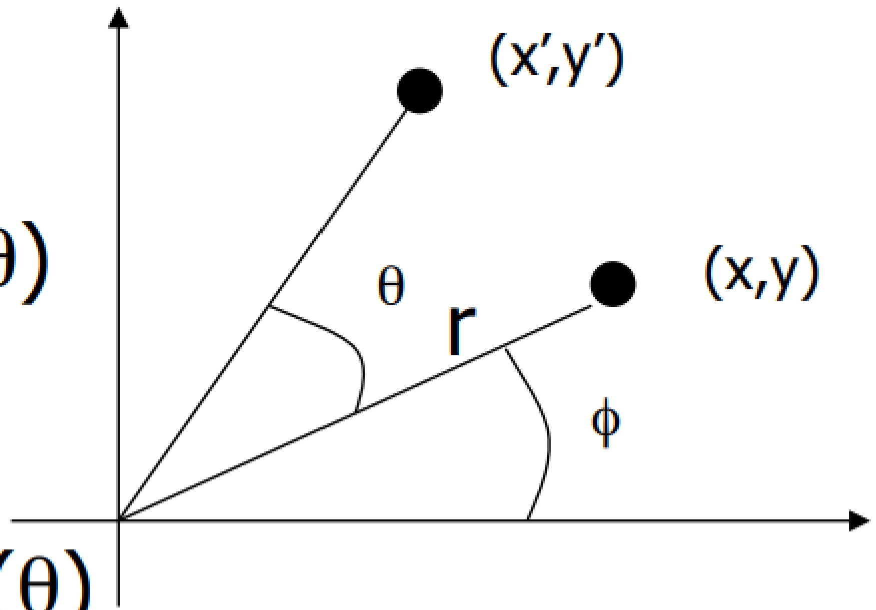
$x = r \cos(\phi) \quad y = r \sin(\phi)$

$x' = r \cos(\phi + \theta) \quad y = r \sin(\phi + \theta)$

polar form

$$x = r\cos(\phi) \quad y = r\sin(\phi)$$

$$x' = r\cos(\phi + \theta) \quad y = r\sin(\phi + \theta)$$

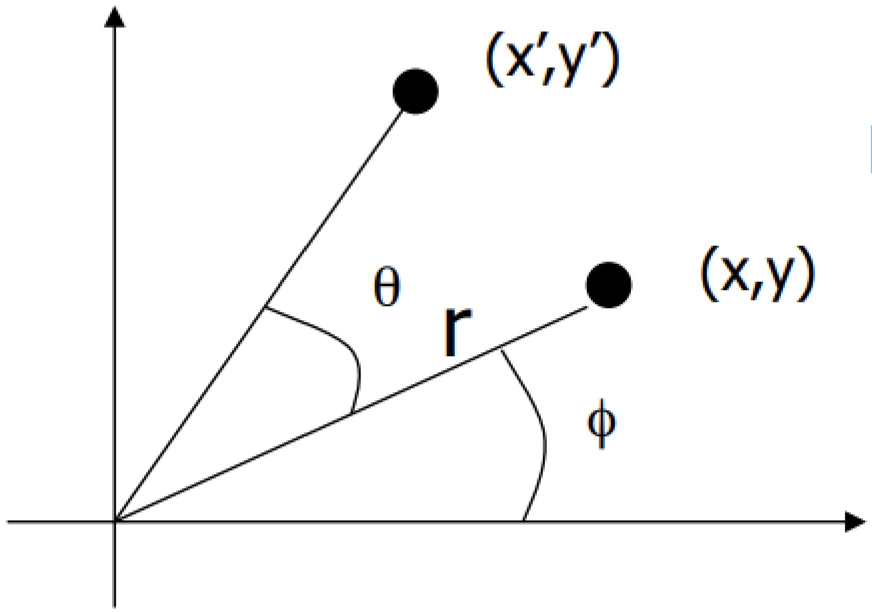


$$\begin{aligned} x' &= r\cos(\phi + \theta) \\ &= r\cos(\phi)\cos(\theta) - r\sin(\phi)\sin(\theta) \\ &= x\cos(\theta) - y\sin(\theta) \end{aligned}$$

$$\begin{aligned} y' &= r\sin(\phi + \theta) \\ &= r\sin(\phi)\cos(\theta) + r\cos(\phi)\sin(\theta) \\ &= y\cos(\theta) + x\sin(\theta) \end{aligned}$$

$$x' = x\cos(\theta) - y\sin(\theta)$$

$$y' = y\cos(\theta) + x\sin(\theta)$$



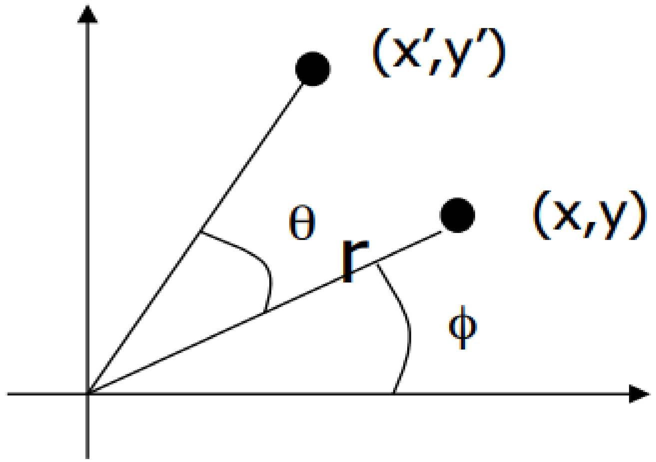
Matrix form?

$$\begin{vmatrix} x' \\ y' \end{vmatrix} = \begin{vmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{vmatrix} \begin{vmatrix} x \\ y \end{vmatrix}$$

3 x 3?

3 X 3 2D Rotation matrix

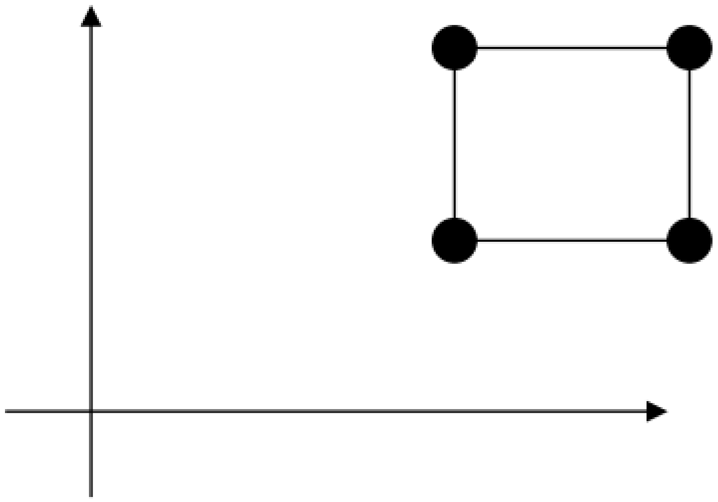
$$\begin{vmatrix} x' \\ y' \end{vmatrix} = \begin{vmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{vmatrix} \begin{vmatrix} x \\ y \end{vmatrix}$$

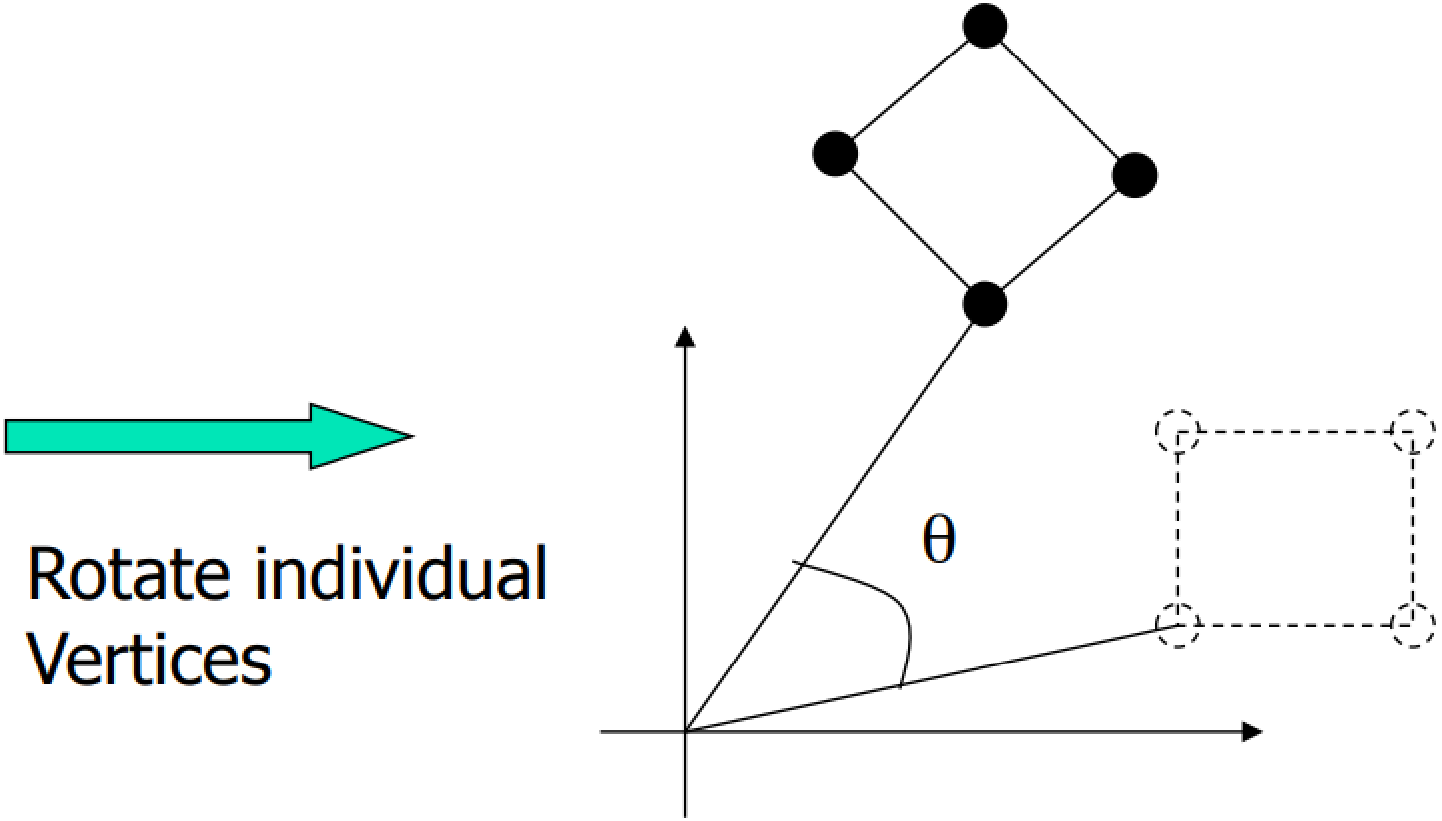


$$\Downarrow$$

$$\begin{vmatrix} x' \\ y' \\ 1 \end{vmatrix} = \begin{vmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} x \\ y \\ 1 \end{vmatrix}$$

- How to rotate an object with multiple vertices?





2D- Rotation Example

- A triangle (object) is given with coordinates (2,4), (8,4) and (5,10) rotate the triangle at 90 degrees.

➢ **For P1'**

- $X' = X \cos\theta - Y \sin\theta$

  $= 2 \cos 90 - 4 \sin 90$

  $= 0 - 4\,(1)$

  $\boxed{= -4}$

$Y' = Y \cos\theta + X \sin\theta$

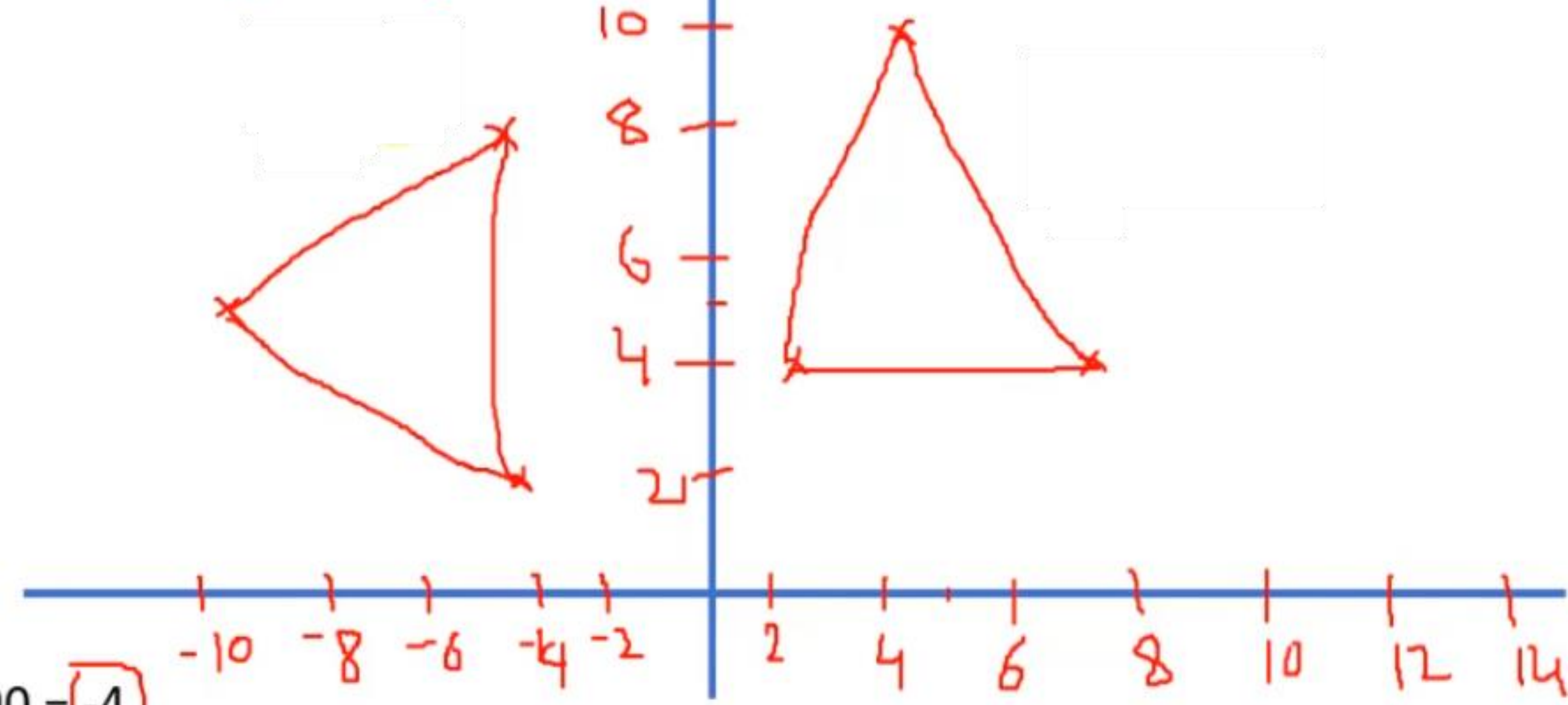
  $= 4 \cos 90 + 2 \sin 90$

  $= 0 + 2(1)$

  $\boxed{= 2}$

➢ **For P2'**

- $X' = X \cos\theta - Y \sin\theta = 8 \cos 90 - 4 \sin 90 = \boxed{-4}$
- $Y' = Y \cos\theta + X \sin\theta = 4 \cos 90 + 8 \sin 90 = \boxed{8}$



Matlab

transl2 - > translation function
trot2-> rotation function

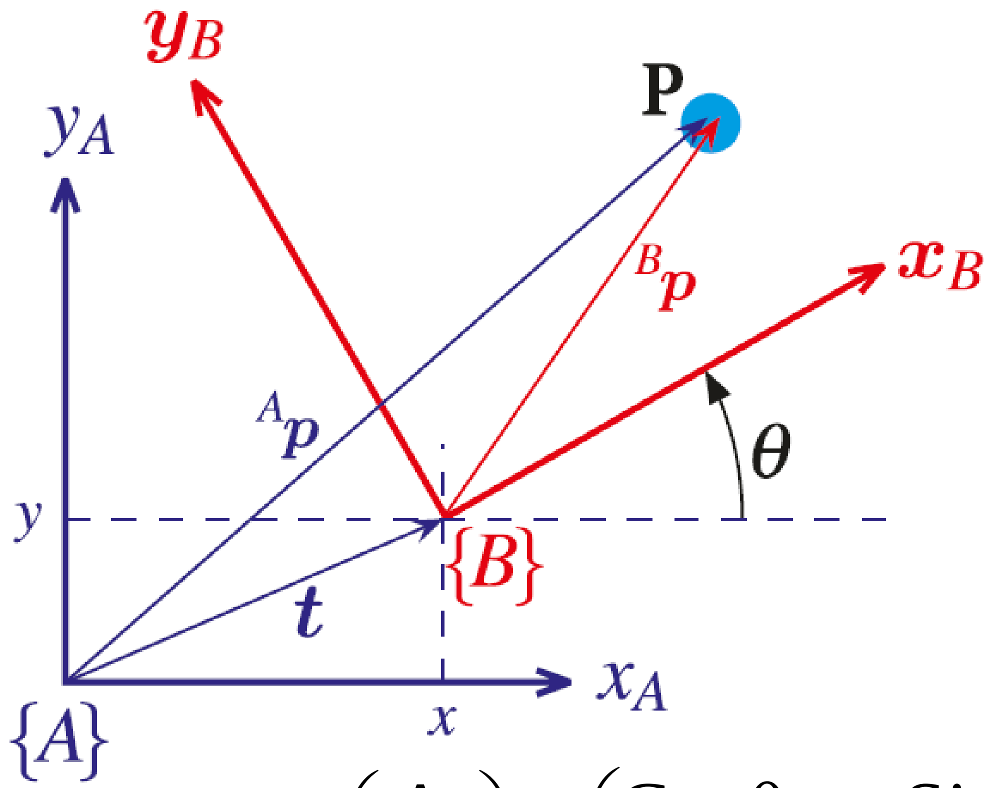
POSE = transl2 * trot2

https://petercorke.github.io/2d/trot2.html

Q)Rotate a triangle placed at A(0,0), B(1,1) and C(5,2) by an angle 45 with respect to point P(-1,-1). Plot the points.

Q)Rotate a triangle placed at A(0,0), B(1,1) and C(5,2) by an angle 45 with respect to origin. Plot the points.

Homogenous Matrix



← Consider this figure

$$A_p = {}^{A}t_B + {}^{A}R_B \bullet B_p$$

$$\begin{pmatrix} A_x \\ A_y \end{pmatrix} = \begin{pmatrix} \text{Cos}\theta & -\text{Sin}\,\theta \\ \text{Sin}\theta & \text{Cos}\theta \end{pmatrix} \begin{pmatrix} B_x \\ B_y \end{pmatrix} + \begin{pmatrix} {}^{A}x_B \\ {}^{A}y_B \end{pmatrix}$$

$$\begin{pmatrix} A_x \\ A_y \end{pmatrix} = \begin{pmatrix} \text{Cos}\theta & -\text{Sin} & {}^{A}x_B \\ \text{Sin} & \theta & {}^{A}y_B \end{pmatrix} \begin{pmatrix} B_x \\ B_y \\ 1 \end{pmatrix}$$

$$\theta \qquad \text{Cos}\theta$$

$$\begin{pmatrix} A_x \\ A_y \\ 1 \end{pmatrix} = \begin{pmatrix} \text{Cos}\theta & -\text{Sin}\,\theta & {}^{A}x_B \\ \text{Sin}\,\theta & \text{Cos}\,\theta & {}^{A}y_B \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} B_x \\ B_y \\ 1 \end{pmatrix}$$

$$A_p = \begin{pmatrix} {}^{A}R_B & {}^{A}t_B \\ 0,0 & 1 \end{pmatrix} \cdot B_p$$

Homogeneous Matrix

$$ {}^{A}p = {}^{A}t_{B} + {}^{A}R_{B} \cdot {}^{B}p \quad \text{—— (1)} $$

Column matrix

$$ \begin{bmatrix} A_x \\ A_y \end{bmatrix} = \begin{bmatrix} A_{x_B} \\ A_{y_B} \end{bmatrix} + \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} B_x \\ B_y \end{bmatrix} \quad \text{—— (2)} $$

$$ {}^{A}\tilde{p} = \begin{bmatrix} {}^{A}R_{B} & {}^{A}t_{B} \\ 0,\ 0 & 1 \end{bmatrix} \cdot {}^{B}\tilde{p} \quad \text{—— (3)} $$

Homogeneous vector

Homogeneous transform

Homogeneous vector

$$\tilde{A}_p = {}^{A}T_B \cdot \tilde{B}_p \text{ ——— } (4)$$

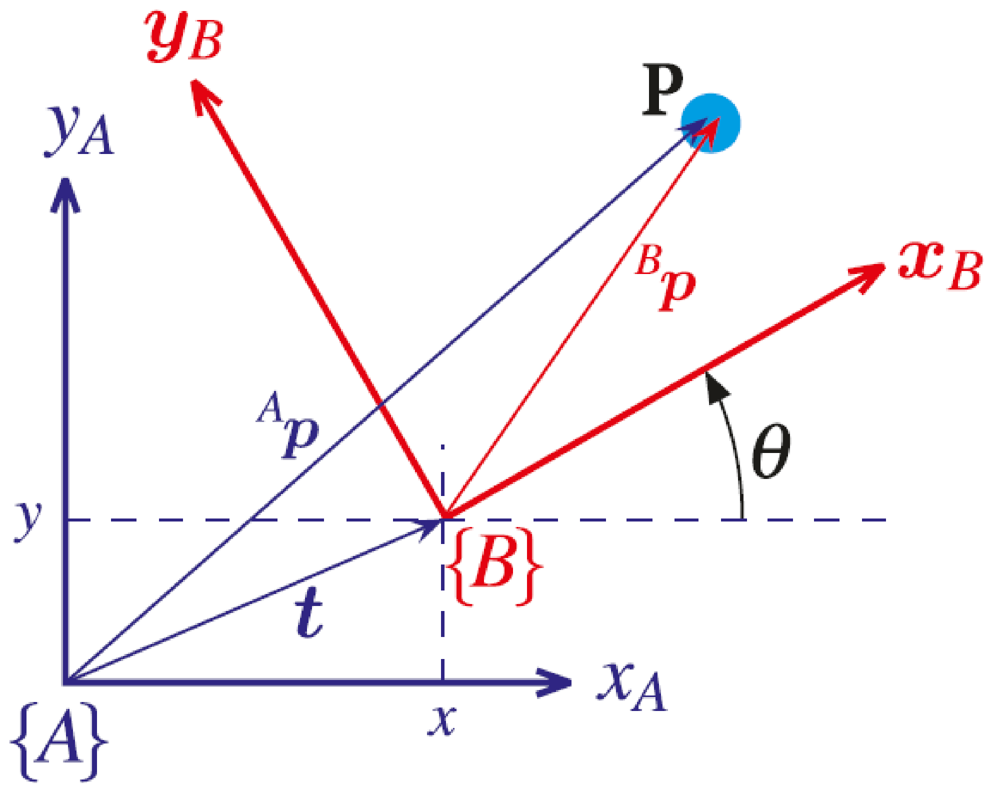
describes relative POSE as 3x3 matrix

$${}^{A}T_B = \begin{bmatrix} {}^{A}R_B & {}^{A}t_B \\ 0, 0 & 1 \end{bmatrix}$$

$${}^{A}T_B = {}^{A}\xi_B = \begin{bmatrix} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

mathematical representation of POSE

Homogenous Matrix



← Consider this figure

Homogenous Transform

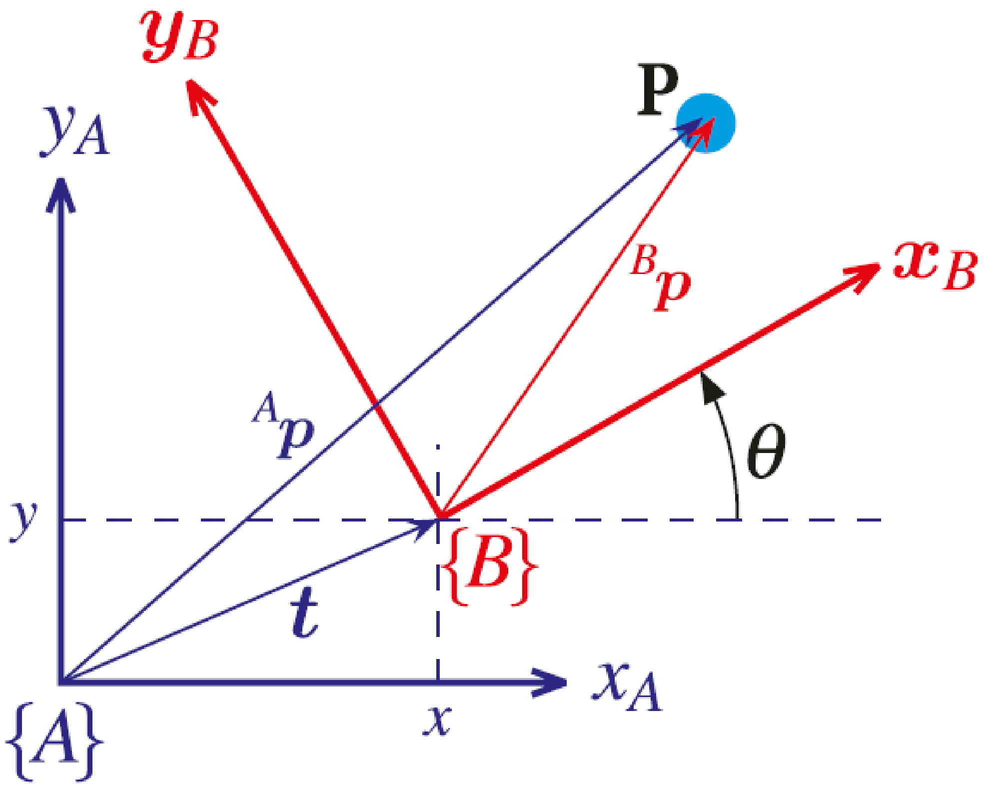
$$A_p = \begin{pmatrix} {}^A R_B & {}^A t_B \\ 0,0 & 1 \end{pmatrix} \cdot B_p$$

Homogenous Vectors

$$A_p = {}^A T_B \cdot B_p$$

Describes a relative POSE as 3×3 matrix. ← $${}^A T_B = \begin{pmatrix} {}^A R_B & {}^A t_B \\ 0,0 & 1 \end{pmatrix}$$

Homogenous Matrix



← Consider this figure

Homogenous Transform

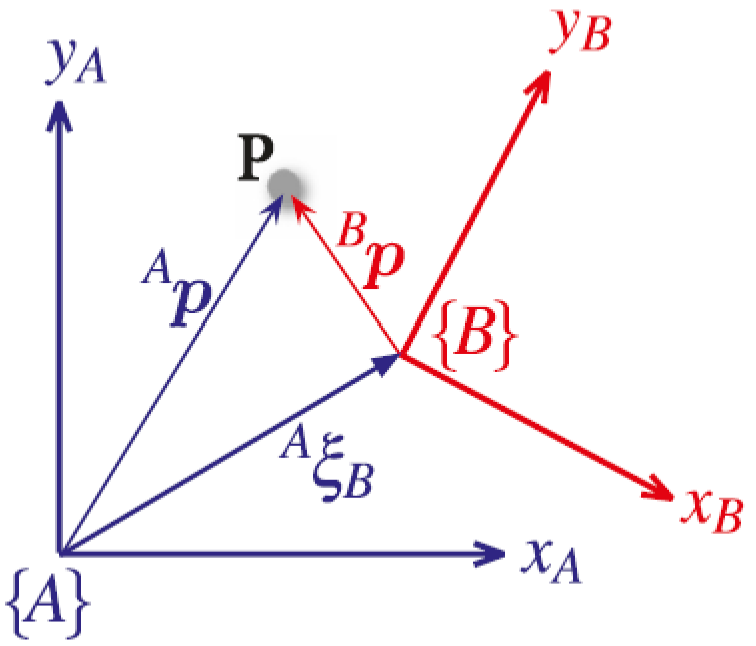
$$A_p = \begin{pmatrix} {}^A R_B & {}^A t_B \\ 0,0 & 1 \end{pmatrix} \cdot B_p$$

Homogenous Vectors

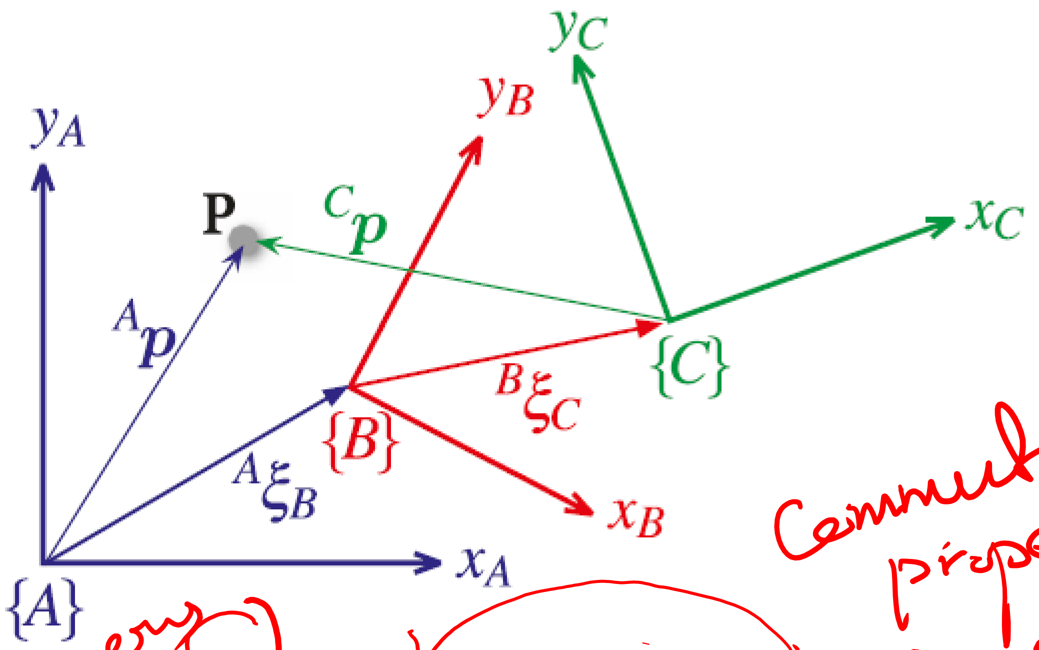
$$A_p = {}^A T_B \cdot B_p$$

Describes a relative POSE as 3×3 matrix. $\leftarrow {}^A\xi_B = {}^A T_B = \begin{pmatrix} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{pmatrix}$

Multiple Frames



$$^{A}p = {}^{A}T_B \cdot {}^{B}p$$



Homogeneous transform

$$^{A}T_C = {}^{A}T_B \cdot {}^{B}T_C$$

Commutative property

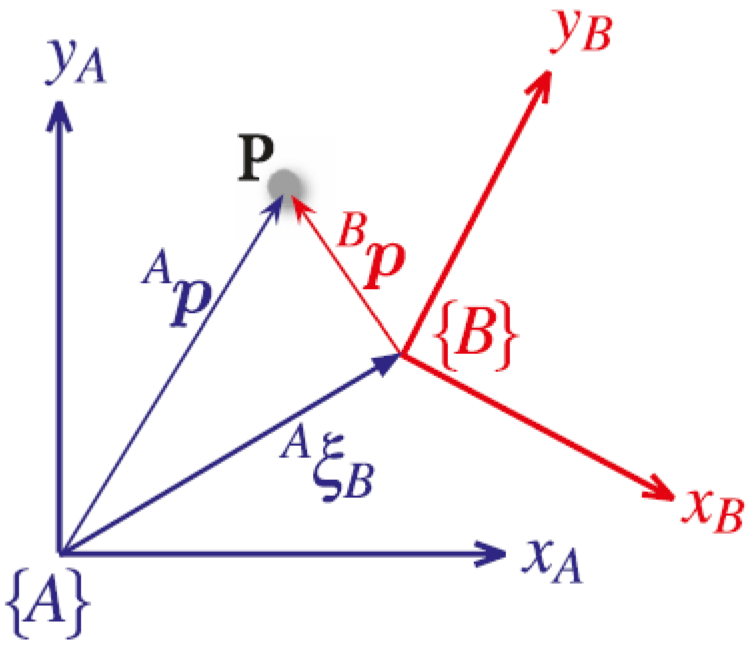
A = B

B = C

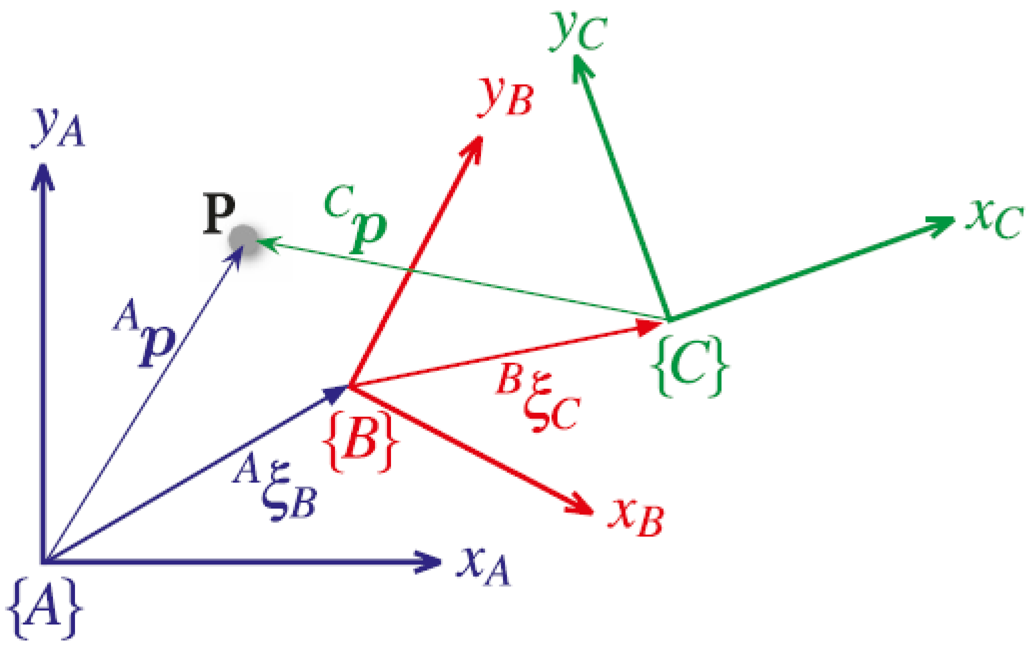
then A = C

$$^{A}p = {}^{A}T_B \cdot {}^{B}T_C \cdot {}^{C}p$$

Multiple Frames



$$^A\boldsymbol{p} = {^A\xi_B} \bullet {^B\boldsymbol{p}}$$



$$^A\xi_C = {^A\xi_B} \oplus {^B\xi_C}$$

$$^A\boldsymbol{p} = \left({^A\xi_B} \oplus {^B\xi_C} \right) \bullet {^C\boldsymbol{p}}$$

Certain things to remember

1) A **point** is described by a bound coordinate vector.
2) Points and Vectors are two different things: **(a)** we can add vectors, but not points; **(b)** difference of two points → vector.
3) A rigid object can be represented by set of points.
4) Position + Orientation of object's coordinate frame → **Pose**.
5) Relative pose → ξ.
6) The $\bullet$ operator.
7) The $\oplus$ operator.

https://petercorke.github.io/2d/trot2.html