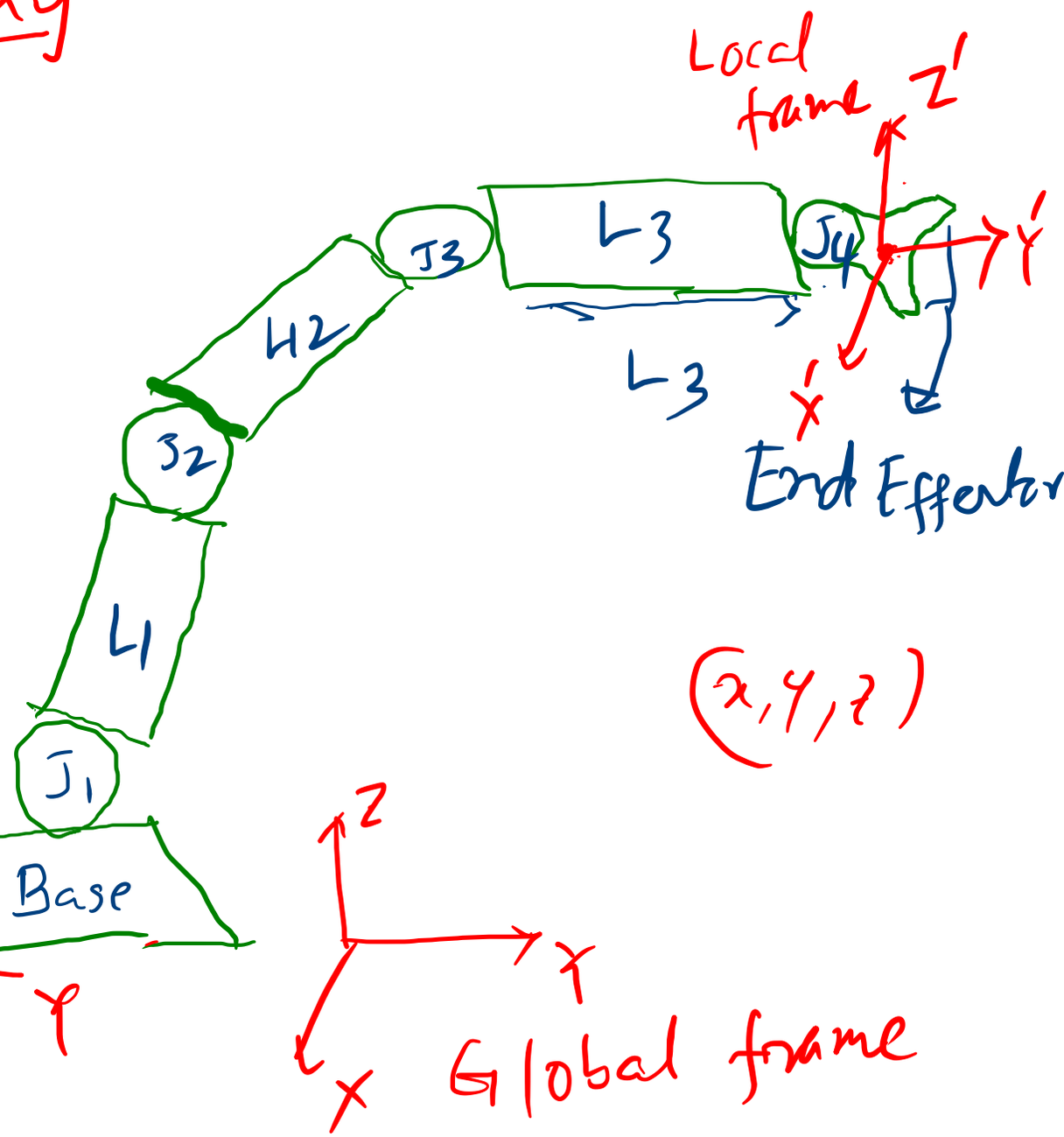
Robot Anatomy

Date 29.01.2021

1. joints (J)
2. Links (L)
3. End Effector
4. Base

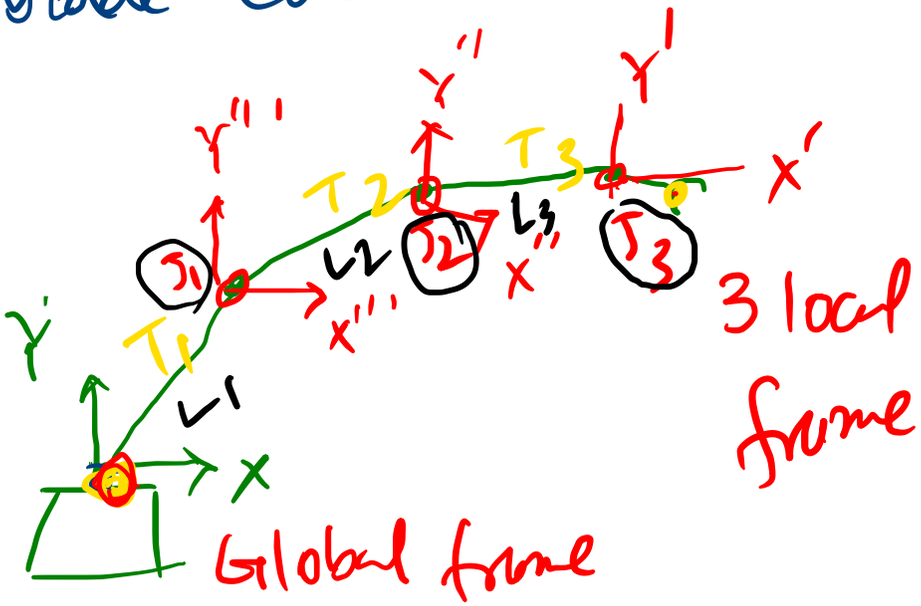


$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = T \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Co-ordinate Transformation

- Passive Transformation
- Active Transformation

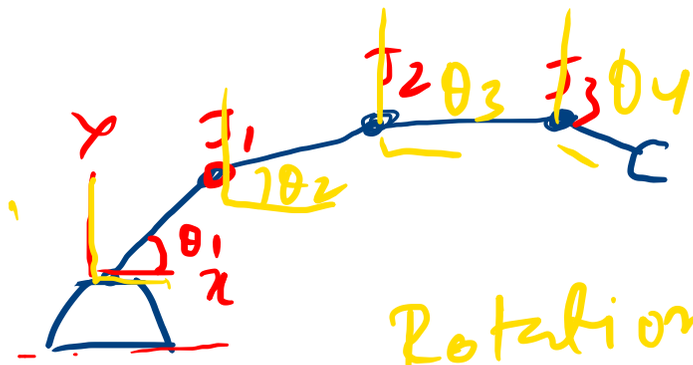
Embedde Co-ordinate System



L': Link

J': Joint

3 local frame



Rotational Transformation

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = T_1 T_2 T_3 \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = R_1 R_2 R \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Coding Python

Python/C++ CUDA { CPU, GPU } < ms → Single tas

Verilog → FPGA < µs for Single task

DRDO, ISRO, Self-Driving car

GPU → High-speed Robotic Computing

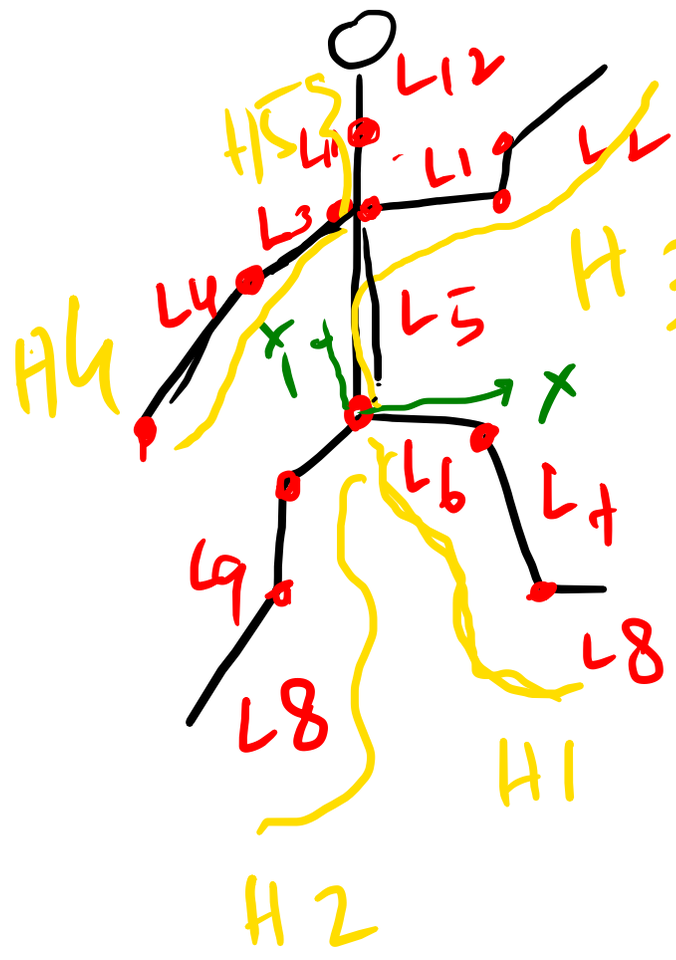
H: Robot Hand

H1:

H2:

H4:

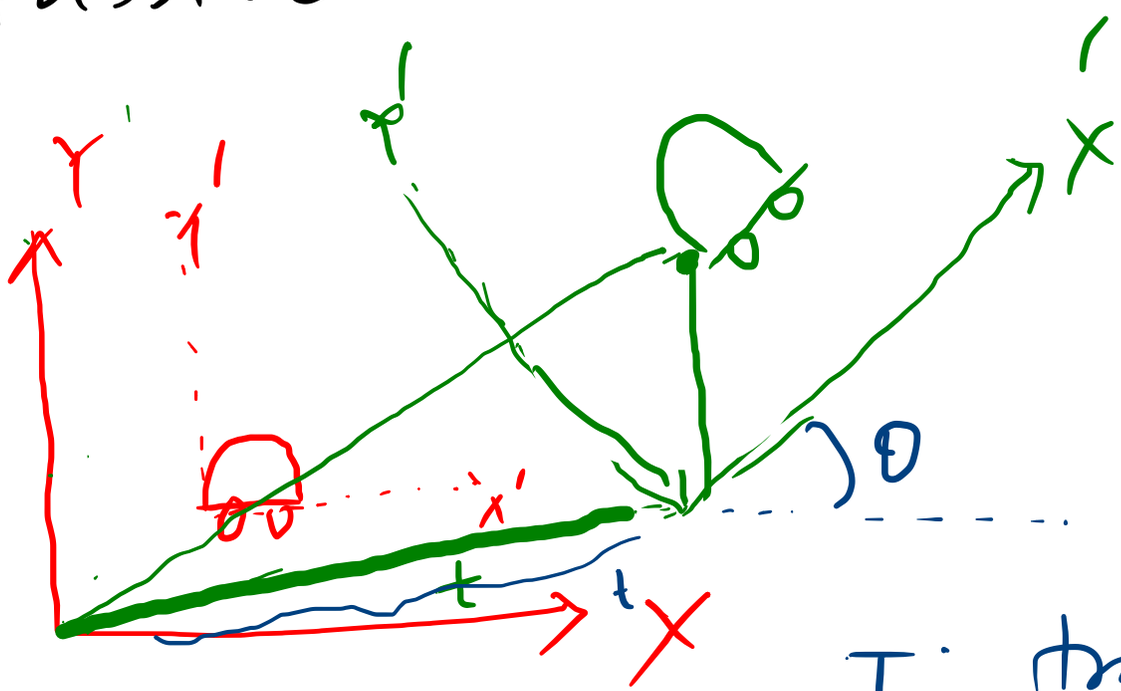
H5:



Co-ordinate Transformation

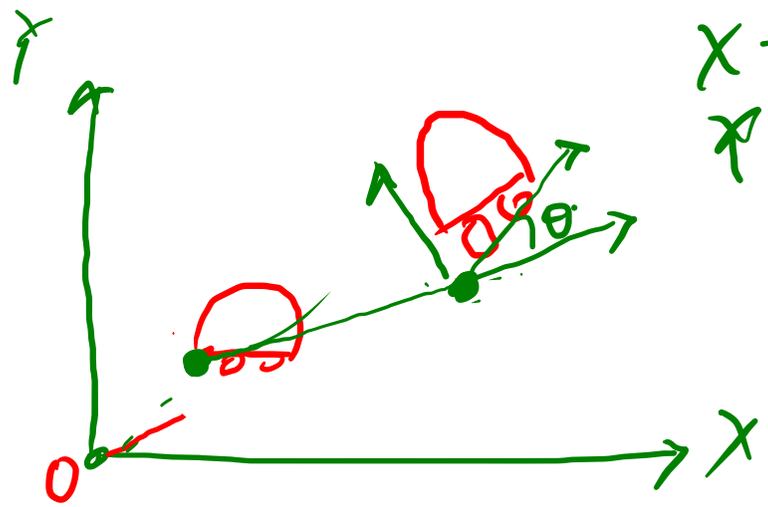
2D or 3D

Passive



Global

Active



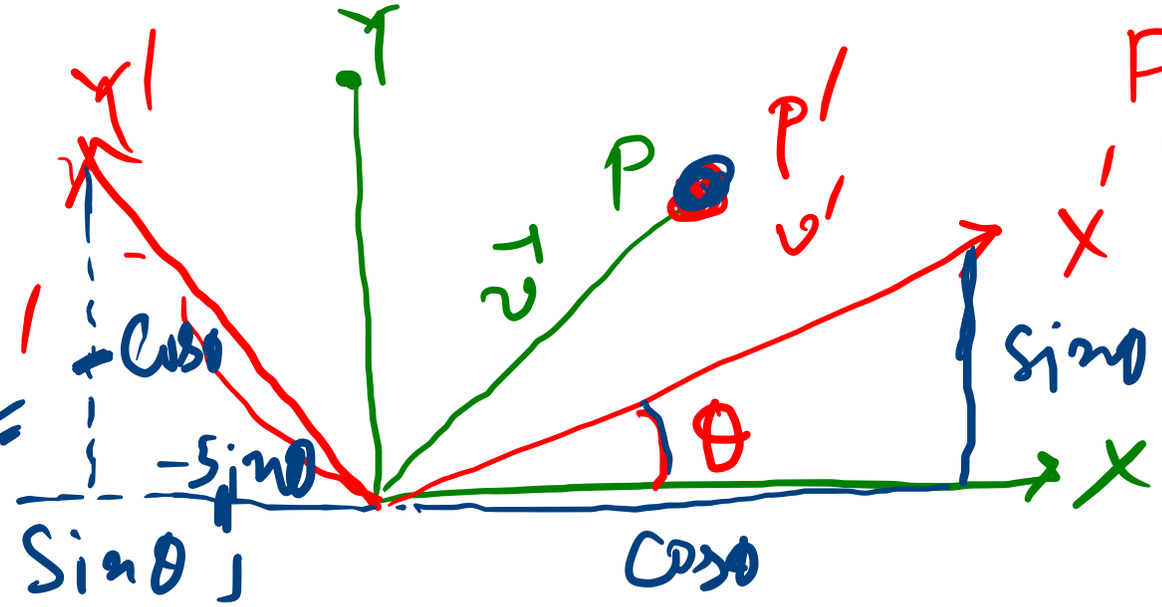
T: Translation

R: Rotation

General Transformation

G = RtT

Passive



$\vec{v} = v_x \hat{i} + v_y \hat{j}$

$\vec{v}' = v_x' \hat{i}' + v_y' \hat{j}'$

$\hat{i}' = \cos\theta\, \hat{i} + \sin\theta\, \hat{j}$

$\hat{j}' = -\sin\theta\, \hat{i} + \cos\theta\, \hat{j}$

$P' = P$

$v' = v$

$\vec{v}' = v_x' (\cos\theta\, \hat{i} + \sin\theta\, \hat{j}) + v_y' (-\sin\theta\, \hat{i} + \cos\theta\, \hat{j})$

$\vec{v}' = (v_x' \cos\theta - v_y' \sin\theta)\, \hat{i} + (v_x' \sin\theta + v_y' \cos\theta)\, \hat{j}$

$\vec{v} = v_x \hat{i} + v_y \hat{j}$

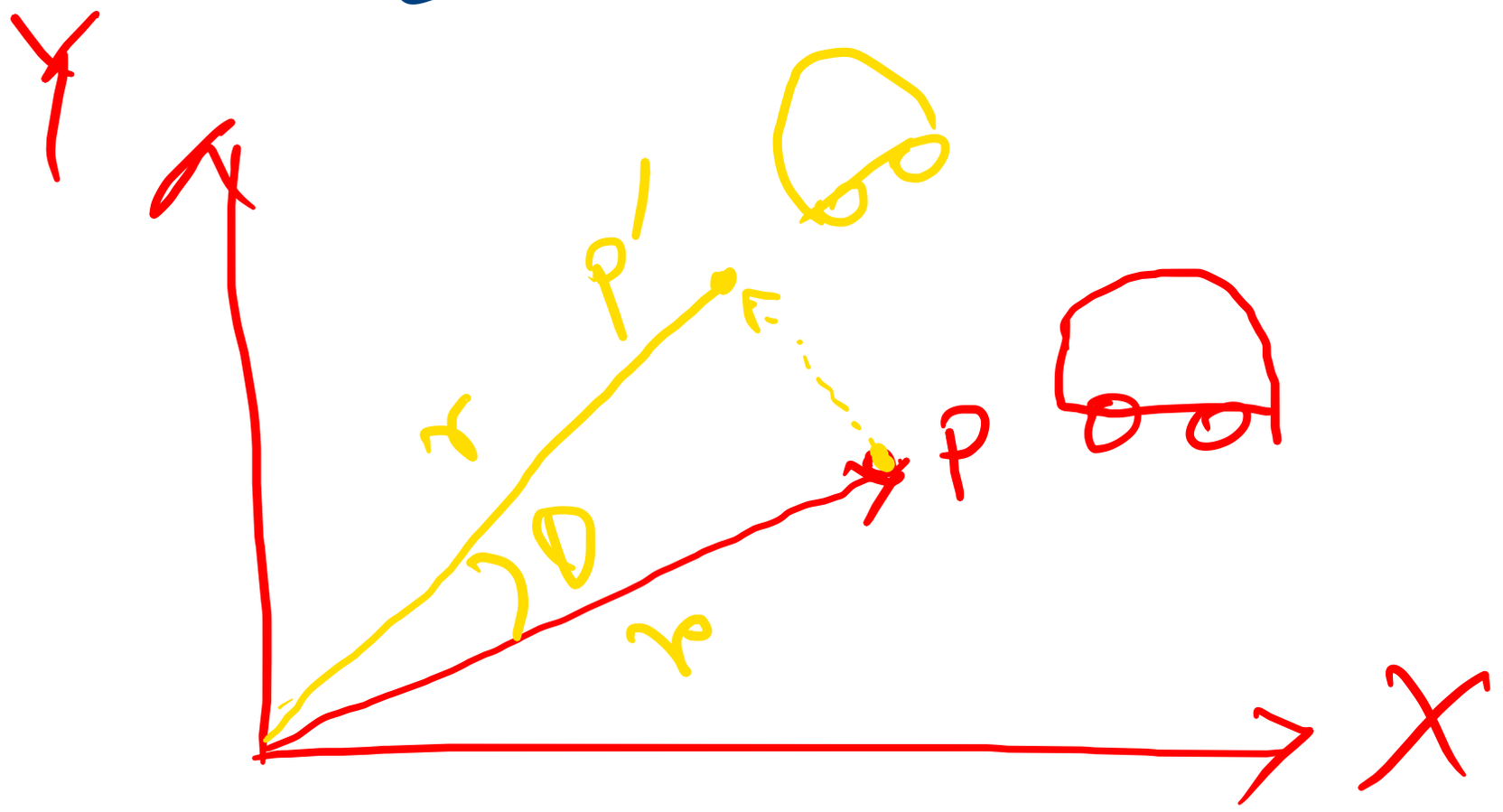
$$v_x = v_x' \cos\theta - v_y' \sin\theta$$

$$v_y = v_x' \sin\theta + v_y' \cos\theta$$

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} v_x' \\ v_y' \end{bmatrix}$$

or $$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = R \begin{bmatrix} v_x' \\ v_y' \end{bmatrix}$$

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



Active transformation

Rotational matrix is more complex
than translational matrix

For Simplicity: Step1: Rotational matrix

Step2: Translational matrix