PERSONAL USE ONLY

Basic elements

- Links/joints/bones/RB – EF, payload
- Actuators → Electrical actuators control excellent.
  → Hydraulic actuator
  → Pneumatic actuators poor control
- Sensors (Encoders)
- Controllers (Logics, computation)

→ Stepper ON/OFF

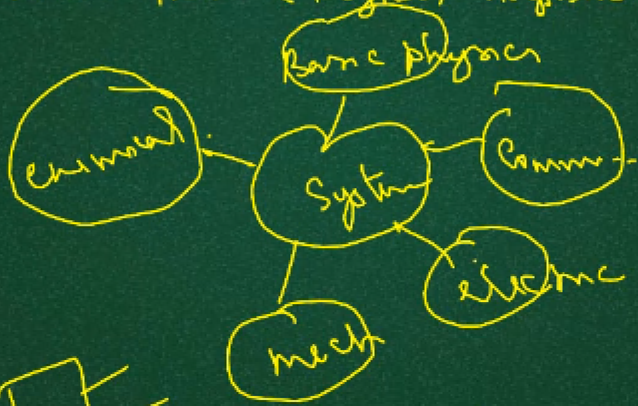
→ DC

RLC AC

MotorCAD

Flux 3D

50L

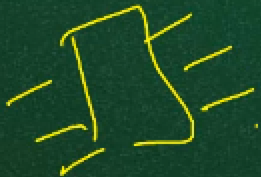


→ Matlab

→ open modelica ✓

Simulink.

System modelling



Links/joints?

→ Loads/Forces ✓ $\boxed{\text{motion/Forces.}}$

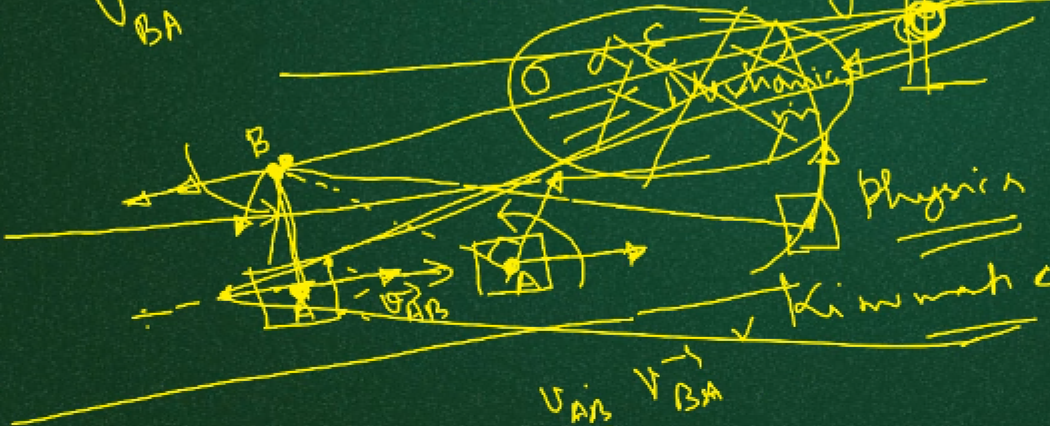
→ Translation of mechanism ✓

$\vec{v}_{AB} \neq \vec{v}_{BA}$

Power → Links → E.F — to do work.

$\vec{v}_{BA}$

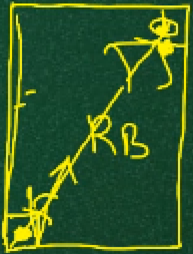
RB → Rigid body → No relative motion between the particles within the RB

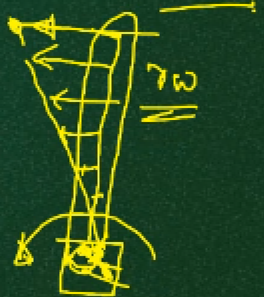


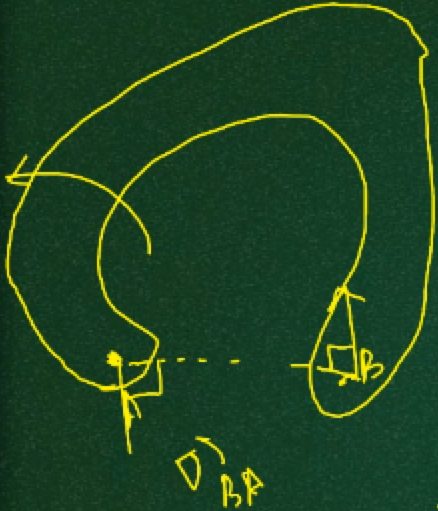
Physics

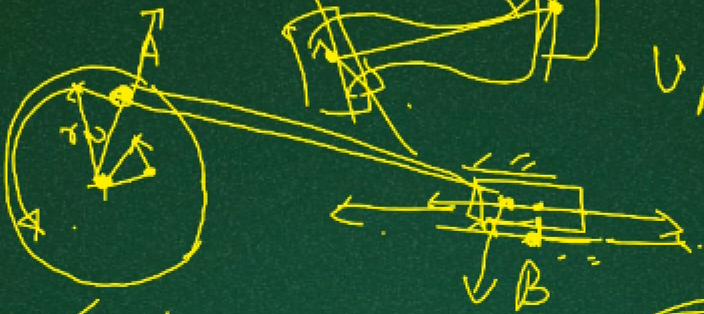
Kinematic

v_{AB} $\vec{v}_{BA}$

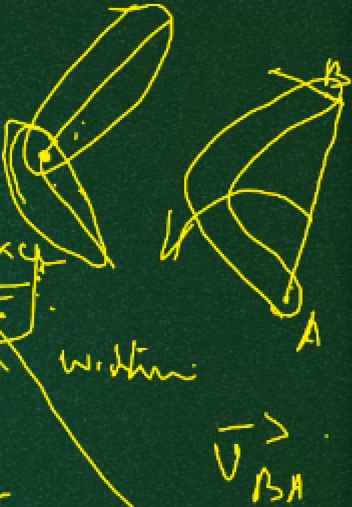








V_{AB}



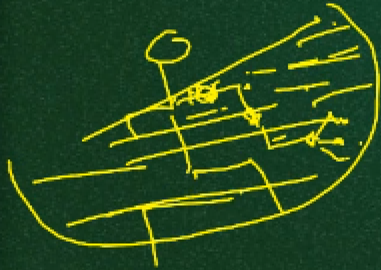
Task space

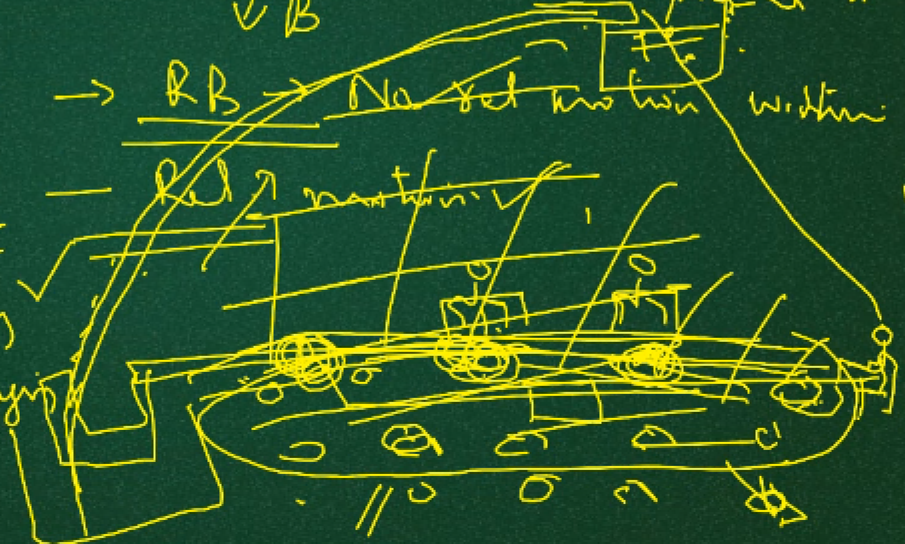
✓ Links → RB → No rel motion within

joints — Rel motion

✓ flexibility

Load carrying





Robotics

DOF

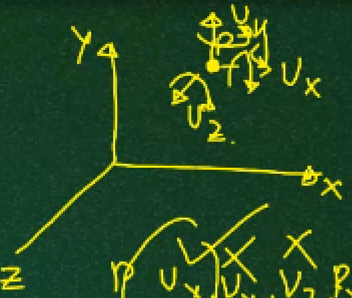
Engineering

controllable

EF

more ✓

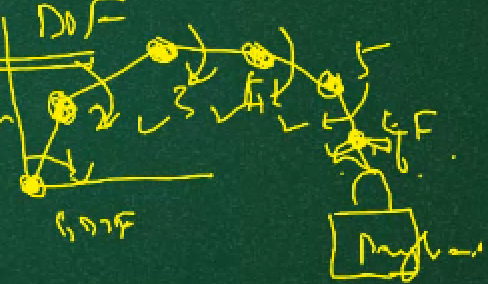
6DOF



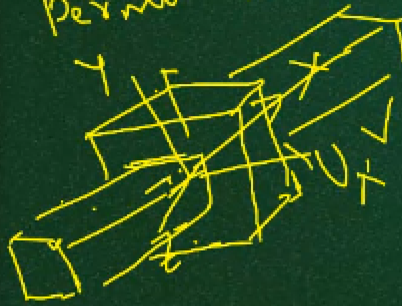
$V_x, V_y, V_z, R_x, R_y, R_z$

6 DOF.

Controllable DOF ⟺ Robot DOF



Permit a linear translation rel motion



Primitive

6DOF

① Prismatic joint

② Revolute joint

1 DOF — easiest for control.

R ✓ T ✓

Stroke robot

Door hinge

Flexibility ✓ Dexterity ✓

U_x, U_y, V_z, R_x, R_y, R_z

Revolute joint

CMU

1 DOF.

1 DOF   Rev

A, B, C, S

Rigidity x   T

Unstructured R_y 1 DOF

Cluttered

environment R_z

R_x, R_y, R_z

U_x 1 DOF U_y

R_z P, R

2 DOF

3 DOF

control DOF < Robot DOF → Non Holonomic

Controllable DOF = Robot DOF   Holonomic System

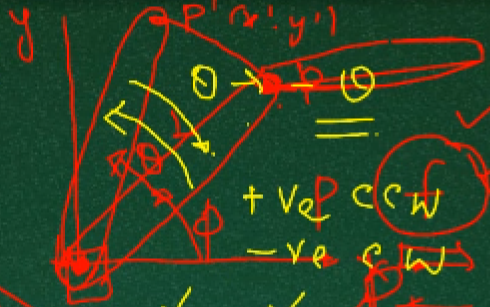
cml DOF >>> Robot DOF → Redundant

Hyper redundant

$P = (x, y)$ Orthogonal

$x = L\cos\phi$ ← P

$y = L\sin\phi$



+ve CCW

−ve CW ⇒ P'

$x' = L\cos(\theta+\phi)$

$y' = L\sin(\theta+\phi)$

$$\frac{1}{|R|}\begin{bmatrix}\cos\theta & \sin\theta \\ -\sin\theta & \cos\theta\end{bmatrix}$$

$x' = L\cos\theta\cos\phi - L\sin\theta\sin\phi$

$y' = L\sin\theta\cos\phi + L\cos\theta\sin\phi$

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

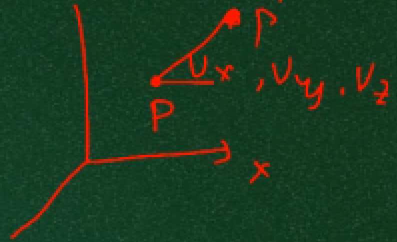
$y' = x\sin\theta + y\cos\theta$

$$\Rightarrow \begin{bmatrix}x' \\ y'\end{bmatrix} = \begin{bmatrix}\cos\theta & -\sin\theta \\ \sin\theta & \cos\theta\end{bmatrix}\begin{bmatrix}x \\ y\end{bmatrix}$$

$P \to P'$ $[R]$

$P' \to P$ $[R^{-1}]$

$f \Rightarrow R_z$



V_x, V_y, V_z

$x + V_x$

$y + V_y$

$z + V_z$

$[T]$ P

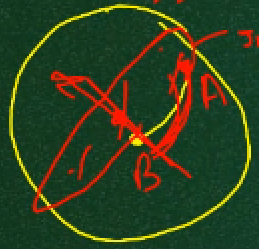
Combination rotation 3×3 3×1 vector

(a, b) 90°

(a, ib) i

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad P$$

(a, −b)



$$\boxed{R_y R_x \neq R_x R_y}$$

Non-commutative

$R_z R_y R_x$

$P \Rightarrow p' \Rightarrow p''$

Great circle

Hyper sphere

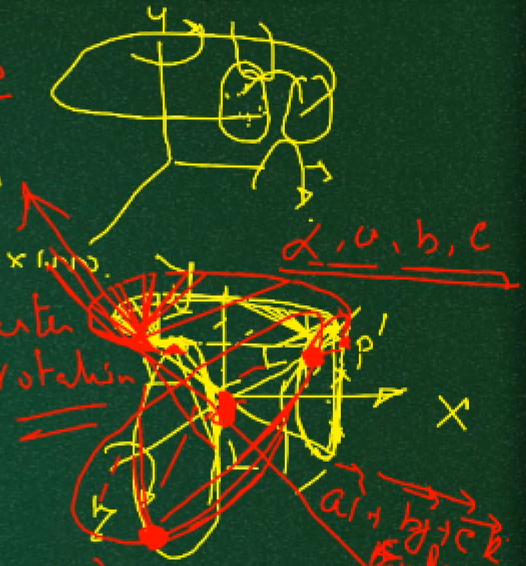
a, ib

p' p

3×1 row 3×1 row

i Quarter rotation

90°



α, a, b, c

ai + bj + ck

Euler axis

a, b

(a + ib) p''

Quarternions VG

PG

Q [a, b, c, d]

$[R]^{3D}$ $[T]$

$\underline{4 \times 4}$

$$\overset{T}{\underset{4\times 4}{\begin{bmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix}}} \times \underset{4\times 1}{\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}} = \begin{bmatrix} x + T_x \\ y + T_y \\ z + T_z \\ 1 \end{bmatrix}$$

$$M = T\,R = \left[\begin{array}{c|c} R & T \\ \hline 0 & 1 \end{array}\right]$$

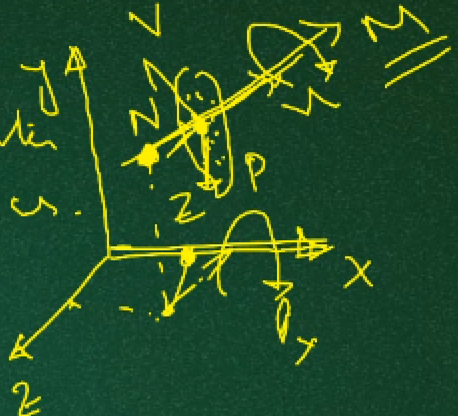
$[T] \times [R]$

$$T \times R_x \Rightarrow \left[\begin{array}{ccc|c} 1 & 0 & 0 & T_x \\ 0 & \cos\theta & -s\theta & T_y \\ 0 & s\theta & c\theta & T_z \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

① Mathematical elements for computer Graphics.

- Rogers
- Adams.

$\doteq T^1 T^2 T^3 [P]$



R_x

$R_y \neq R_V$

$T_1^{-1} T_2^{-1} T_3^{-1} P, \; T^3 T^2 T^1$

② Computer Graphics.

Hearn & Baker

Basic Transformation

$$M^{R,T} = \left[\begin{array}{c|c} \underline{3\times 3} & 3\times 1 \\ R & T \\ \hline 0^{1\times 3} & 1^{1\times 1} \end{array}\right]$$

$$M^{-1} = \left[\begin{array}{c|c} R^{T\;3\times 3} & -R^T T^{\;3\times 1} \\ \hline 0 & 1 \end{array}\right]$$

$$-R^T T \quad (3\times 3 \;\; 3\times 1)$$

$$M^{-1} \neq M^T$$

$\underline{4\times 4}$

$$M^{-1} = \left[\begin{array}{c|c} R^T & -R^T T \\ \hline 0 & 1 \end{array}\right]$$

Numpy

Scipy

Matplotlib

Python code

$f(M^{-1})$

$f(R)$

$f(T)$

$f(M)$