Robot Kinematic:~

$\bar{P} = \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$ where, $a_x = \frac{x}{w}$, $b_y = \frac{y}{w}$, $c_z = \frac{z}{w}$.

If.
- $w > 1$, vector components enlarge.
- $w < 1$, vector components are smaller.
- $w = 1$, no scaling.
- $w = 0$, length ∞, and has direction.

eg. $P = \begin{bmatrix} 6 \\ 10 \\ 4 \\ 2 \end{bmatrix} \longleftrightarrow \bar{P} = 3\hat{i} + 5\hat{j} + 2\hat{k}$.

- **Fixed Frame:-**
  - $\bar{n}$ for normal vector.
  - $\bar{a}$ for approach vector.
  - $\bar{o}$ for orientation vector.

$\boxed{\bar{n}, \bar{o}, \bar{a}}$

Rotation Matrix:-

About x-axis

$$\begin{array}{c|cccc} & x & y & z & \\ \hline x & 1 & 0 & 0 & P_1 \\ y & 0 & \cos\theta & -\sin\theta & P_2 \\ z & 0 & \sin\theta & \cos\theta & P_3 \\ \hline & 0 & 0 & 0 & w \end{array}$$

About y-axis.

$$\begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

About z-axis

$$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$F = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$\bar{n}, \bar{o}, \bar{a}$ are mutually $\perp^{lar}$ to each other.

$\therefore \bar{n}.\bar{o} = \bar{n}.\bar{a} = \bar{o}.\bar{a} = 0$.
Also, $|\bar{n}| = |\bar{a}| = |\bar{o}| = 1$.
$\}\rightarrow$ Use these two formulas.

Homogeneous Transformation:-

- **Pure Translation:-**

$$T = \begin{bmatrix} 1 & 0 & 0 & dx \\ 0 & 1 & 0 & dy \\ 0 & 0 & 1 & dz \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Pre-Multiplication.

$F_{new} = Trans(dx\ dy\ dz) \times F_{old}$.

- **Rotational ~~Translation~~ Transformation:-**

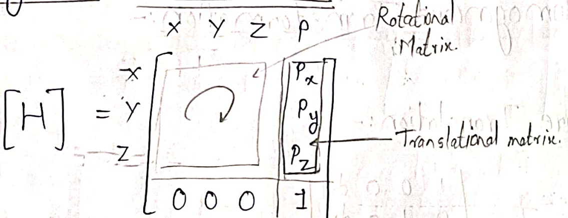
- **Combinational Translation:-**
  Combination of Translation and Rotation.

[*] Trick for Inverse:-

$$M = \begin{bmatrix} n_x & o_x & a_x & P_x \\ n_y & o_y & a_y & P_y \\ n_z & o_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad M^{-1} = \begin{bmatrix} n_x & n_y & n_z & -\bar{P}.\bar{n} \\ o_x & o_y & o_z & -\bar{P}.\bar{o} \\ a_x & a_y & a_z & -\bar{P}.\bar{a} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- **Numericals:-**

I. Homogeneous Transformation:

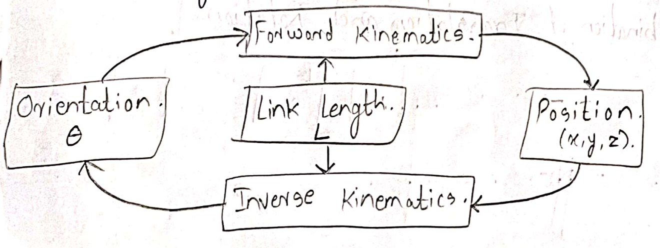


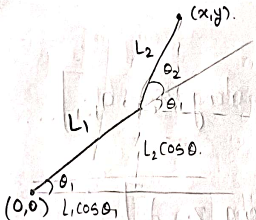
- **Forward Kinematics:-**

If we know link lengths and Angles, then we can obtain end effector position and orientation. This is forward kinematics.

- **Inverse Kinematics:-**

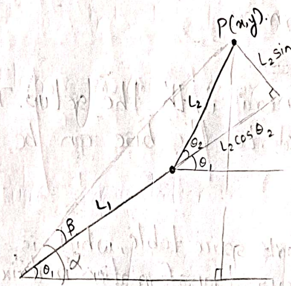
Calculating Orientation from End effector Position and link lengths is Inverse Kinematics.





$$\left.\begin{aligned} x &= L_1\cos\theta_1 + L_2\cos(\theta_1+\theta_2) \\ y &= L_1\sin\theta_1 + L_2\sin(\theta_1+\theta_2) \end{aligned}\right\} \text{(I)}$$

Forward Kinematics.



$\tan\alpha = \dfrac{y}{x}$.

Reverse Kinematics.

$$\tan\beta = \frac{L_2\sin\theta_2}{L_1 + L_2\cos\theta_2}$$

Squaring and adding eq^{ns} I,

$\therefore x^2+y^2 = L_1^2+L_2^2+2L_1L_2\cos\theta_2$.

$\therefore L_2\cos\theta_2 = \dfrac{x^2+y^2-L_1^2-L_2^2}{2L_1}$

Find θ_2 from here.

Then find β, α.

Then we'll get θ_1.

- A Point is Approachable Iff

$$\boxed{L_1+L_2 \geq \sqrt{x^2+y^2}.}$$

- Robot reach:-

$$\boxed{|L_1-L_2| \leq \text{Reach} \leq L_1+L_2}$$

$$^{i-1}_{\;\;i}[T] = [Rot\ @_{x_{i-1}}, \alpha_{i-1}]\,[\text{Trans along } x_{i-1}, a_{i-1}] \ldots$$

$$[Rot\ @_{z_i}, \theta_i]\,[\text{Trans along } z_i, d_i]$$

[D-H] parameter Transformer.

Q. A robot work cell as a camera with the setup. The origin of six joint robot fixed to the base can be seen by Camera.

A cube is placed on the work space table, which is seen by camera. The homogeneous transformation matrix T_1 maps the camera with the cube (cube w.r.t. camera). The origin of the base co-ordinate system w.r.t. camera is given by T_2 (i) Find the position and orientation of cube w.r.t. base co-ordinate system.
(ii) After system has been setup, someone rotates the camera $90°$ about z-axis. What is the position and orientation of camera w.r.t. base co-ordinate system.

$$^{Camera}_{Cube}[T_1] = \begin{bmatrix} 0 & 1 & 0 & 2 \\ 1 & 0 & 0 & 8 \\ 0 & 0 & -1 & 7 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad ^{Camera}_{base}[T_2] = \begin{bmatrix} 1 & 0 & 0 & -8 \\ 0 & -1 & 0 & 15 \\ 0 & 0 & -1 & 6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(i)
$$^{Base}_{Cube}[T_3] = {}^{Base}_{Camera}[T_2^{-1}] \cdot {}^{Camera}_{Cube}[T_1] = {}^{Base}_{Camera}$$

(i)
$$^{Base}_{Cube}[T_3] = \begin{bmatrix} 1 & 0 & 0 & 8 \\ 0 & -1 & 0 & 15 \\ 0 & 0 & -1 & 6 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 2 \\ 1 & 0 & 0 & 8 \\ 0 & 0 & -1 & 7 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$^{Base.}_{Cube}[T_3] = \begin{bmatrix} 0 & 1 & 0 & 10 \\ -1 & 0 & 0 & 7 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

→ Orientation.
→ Position.

(ii)
$$^{Base}_{Camera}[T_4] = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 8 \\ 0 & -1 & 0 & 15 \\ 0 & 0 & -1 & 6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Check!!

=

Base 0

0. Camera.

Q. A 6 jointed robotic manipulator, has a camera.
Pos. and orien. of obj. w.r.t. camera is T_1.
Base w.r.t. Camera is T_2.
Pos. and orien. of Gripper w.r.t. base is T_3.
(i) Determine Pos. and orient. of obj. w.r.t. base.
(ii) ———————— " ———————— gripper.

$$\underset{\text{Obj.}}{\overset{\text{Camera.}}{[T_1]}} = \begin{bmatrix} 0 & 1 & 0 & 5 \\ 1 & 0 & 0 & 6 \\ 0 & 0 & -1 & 10 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \underset{\text{Base.}}{\overset{\text{Camera.}}{[T_2]}} = \begin{bmatrix} 1 & 0 & 0 & -20 \\ 0 & -1 & 0 & 10 \\ 0 & 0 & -1 & 12 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\cancel{8 \quad 6 \quad 6 \quad 1}$$

$$\underset{\text{Gripper}}{\overset{\text{Base.}}{[T_3]}} = \begin{bmatrix} 1 & 0 & 0 & 8 \\ 0 & 1 & 0 & 6 \\ 0 & 0 & 1 & 6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{(i)} \quad \underset{\text{Obj.}}{\overset{\text{Base}}{[T_4]}} = \underset{\text{Camera.}}{\overset{\text{Base.}}{[T_2^{-1}]}} \underset{\text{Obj.}}{\overset{\text{Camera.}}{[T_1]}} = \begin{bmatrix} 1 & 0 & 0 & 20 \\ 0 & -1 & 0 & 10 \\ 0 & 0 & -1 & 12 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 5 \\ 1 & 0 & 0 & 6 \\ 0 & 0 & -1 & 10 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\underset{\text{Obj.}}{\overset{\text{Base.}}{[T_4]}} = \begin{bmatrix} 0 & 1 & 0 & 25 \\ -1 & 0 & 0 & 4 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{(ii)} \quad \underset{\text{Obj}}{\overset{\text{Gripper.}}{[T_5]}} = \underset{\text{Base}}{\overset{\text{Gripper.}}{[T_3^{-1}]}} \underset{\text{Obj}}{\overset{\text{Base}}{[T_4]}} = \begin{bmatrix} 1 & 0 & 0 & -8 \\ 0 & 1 & 0 & -6 \\ 0 & 0 & 1 & -6 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 25 \\ -1 & 0 & 0 & 4 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

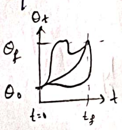
$$\text{(ii)} \quad \underset{\text{Obj.}}{\overset{\text{Gripper.}}{[T_5]}} = \begin{bmatrix} 0 & 1 & 0 & 17 \\ -1 & 0 & 0 & -2 \\ 0 & 0 & 1 & -4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Trajectory Planning in Robotics :-

Trajectory refers to a time history of position, velocity, and acceleration for each degree of freedom.

Joint Space Schemes :-

Trajectory is usually denoted by Cubic-polynomial fit eq^ns.



$\theta(0) = \theta_0$
$\theta(t_f) = \theta_f$.

$$\theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3.$$
$$\theta'(t) = a_1 + 2a_2 t + 3a_3 t^2$$
$$\theta''(t) = 2a_2 + 6a_3 t.$$

Usually, initial velocity is 0, and final velocity is Zero.

$\theta'(0) = 0$; $\theta'(t_f) = 0$.

$\Rightarrow a_1 = 0$ $\qquad \Rightarrow 2a_2 + 3a_3 t_f = 0.$ —(I).

$\theta(0) = a_0 = \theta_0.$ —(Initial Angle).

$\theta(t_f) = \theta_f = \theta_0 + a_2 t_f^2 + a_3 t_f^3$ —(II).

$$\therefore \boxed{a_2 = \frac{3}{t_f^2}(\theta_f - \theta_0)} \; ; \; \boxed{a_3 = -\frac{2}{t_f^3}(\theta_f - \theta_0)} \text{ —(From I and II).}$$

eg. $\theta_0 = 15^\circ$, $\theta_f = 75^\circ$, $t_f = 3$ sec. Find coeffs. of Cubic polynomial fit.

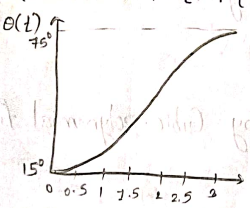
$\rightarrow$ $a_0 = \theta_0 = \underline{15^\circ}$ $\qquad a_2 = \frac{3}{(3)^2}(60) = \underline{20^\circ}$

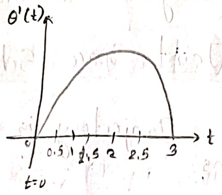
$a_1 = \underline{0^\circ}$

$a_3 = -\frac{2}{\cancel{27}\,9}(\overset{20}{\cancel{60}}) = \underline{-4.44^\circ}$

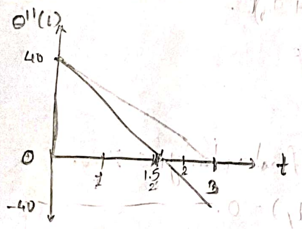
Plot Position, Velocity and acceleration.

$\theta(t) = 15 + 20t^2 + (-4.44)t^3$. $\qquad \theta'(t) = 40t - 13.33t^2$.





$\theta''(t) = 40 - 26.66 \times t$.

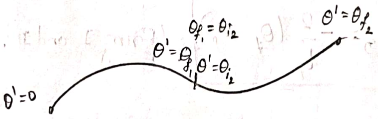


Plot the graph properly, at proper scale.

θ \ t	0	0.5	1	1.5	2.0	2.5	3.0
θ	15°						75°
θ'	0°						0°
θ''	40						-40

☐ Via Point Method :- IMP

Here, initial and final velocities won't be zero.

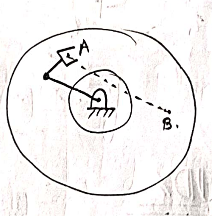


The final velocity of first curve will continue as the initial velocity of second curve.

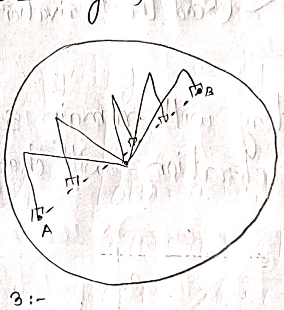
It's a midpoint w.r.t. time (half of total time).

◇ Geometric Problems with Cartesian Paths:-

· Type 1 :- Intermediate pt. unreachable.



· Type 2 :- High joint rate near singularity.

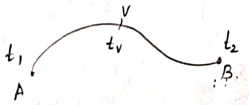


· Type 3 :-

Via Point Method :-

$\theta'_A(t_1)=0$ $\quad$ $\theta'_B(t_2)=0$

$\theta'_A(t_v)\neq 0.$ $\quad$ $\theta'_B(t_v)\neq 0.$

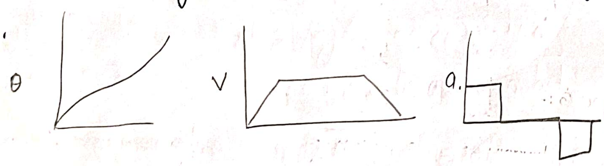


Blending Scheme :- STUDY

$$\boxed{a \geq \frac{4(\theta_f-\theta_0)}{T^2}}$$ (Read the Derivation)

Here, the position/trajectory will remain same. But, the velocity and acceleration's will change.

eg.



Linear Trajectory :-

$$\theta_t = \theta_0 + (\theta_f-\theta_0)\left(\frac{t-t_i}{t_f-t_i}\right)$$

$$\boxed{\frac{\theta_t-\theta_0}{\theta_f-\theta_0}=\frac{t-t_0}{t_f-t_0}} \qquad \frac{x-x_1}{x_2-x_1}=\frac{y-y_1}{y_2-y_1}$$

Differential Relationships :-

· Langragian :- $\rightarrow L_2 - L_1$

Investment and Evaluation Strategies :-

1. Method of pay-back period.
2. Return on investment method.
3. Discounted ~~of~~ cash flow method.
4. Equivalent uniform annual cost method.

1. Method of Payback period :-

$$\boxed{P = \frac{C}{L+V-R}}$$

P = Payback period in Years.
C = total capital (investment).
L = Labour saving.
V = added value
R = annual running cost.

If there are 'n' shifts, then multiply 'L' and 'V' by 'n'.
Take 'R' as mentioned.

eg. $C = 500{,}000$.
$V = 120000$ $R = 100000$.
$L = 80 \times 2600$

$$\therefore P = \frac{C}{L+V-R} = 2.77.$$

If different factors are mentioned, put them in the given fields as per your logic.

2. Return on Investment :-

C = Cost / Initial investment.
F = Salvage Value.
C/n = Yearly depreciation. (Straight line Method of depriciation).

$$\text{Annual rate of return} = \frac{\text{Net saving (or income)} \times 100}{\text{Total investment on robot.}}$$

$$= \frac{[(\text{Increased output}) + (\text{Labour Saving}) - (\text{Annual depriciation}) - (\text{Annual maintenance Cost})] \times 100}{\text{Total Capital.}}$$

3. Discounted Cash flow Method :-

$$R = \frac{1}{(1+r)^n}$$

r = yearly interest rate.
n = no. of years.

$$\boxed{PV = \sum_{K=0}^{n} \frac{(-C_K + S_K - R_K)}{(1+i)^K} + \frac{L}{(1+i)^n}}$$

PV = Present Value.
C_K = Investment Cost in k^{th} year.
S_K = Saving in year k.
R_K = Running cost in year k.
L = Forecasting of salvage value.
i = interest rate.
n = life in years.

4. Equivalent uniform annual cost :-

Equivalent Uniform Cost = − Cost × Risk factor + Annual revenue − annual running cost.

If it is +ve, it is a good investment.

$$\boxed{Y = \frac{(P+A+I) - C}{(L+M-O)\,H\,(1-TR) + D\,(TR)}}$$

Y = no. of years to breakeven.

Robot Languages and Programming:-

- **Wave and AL** :-
  AL ⟶ Arm Language.
  Wave ⟶ Hand Eye Co-ordination.
- **VAL**:- Victor Assembly Language for PUMA robot.
- **AML**:- A manufacturing Language. by IBM.
- **MCL**:- MCL by McDonnel-Douglas.
  Modified from APT (Automatically Programmed Tooling).
- **RAIL**:- RAIL by Automatix.
- **HELP**:- by General Electric Company.
- **JARs**:- by NASA's JPL.
- **RPL**:- by SRI International.
  Idea of LISP, FORTRAN like syntax
- **AUTOPASS**:- by IBM.

Classification of Robot Languages:-

1. First Generation Language.
2. Second Generation Language.
3. World modelling and task-oriented object level language.

Modes of Operation:-

1. Monitor Mode.
2. Run mode. or Execute mode.
3. Editor mode

VAL:-

- Locations:-
- Trajectory Control:-
- Monitor Commands:-
- Defining and Determining Locations:-

eg. HERE PART or HERE P1.

POINT PART = P1.

- Motion Control:-

eg. MOVE. DRAW. DEPART. CIRCLE.

MOVES. APPRO APPROS/DEPARTS

Suffix (s) for straight line.

eg. MOVE P1.

MOVES P1.

APPRO P1,50 [Approach near P1 keeping distance 50 units].

MOVE P1.

DEPART 50. [Move 50 units away].

SPEED 60 [60 % of Rated Speed].

SPEED. 15 IPS [Inch Per Second / Any specific Unit]

DRIVE -4, 75

GRASP 20, 15

OPEN.

CLOSE

OPENI

CLOSEI

} I ~ Immediately.

STORE SIGNAL

LOAD. WAIT.

Robot Programming Methods:-

1. Manual Method.
2. Walk-through Method. (Actually walk and show locations).
3. Power Lead-through Method. (Using Teach Pendent device). (or joysticks)
4. Off-line programming.

Types of Programming:-

- Joint level programming. (Very Basic Actions).
- Robot level programming. (Positions and Orientations)
- High level programming. (Object level / Task level).

Online Programming:

- teach pendant.
- lead through programming.

Offline Programming:

- Robot programming language
- Task level programming

Robot Safety:-

- Safety in Robots.

OSHA: Occupational Safety and Health Act.

Three levels of Robot safety:-

- Level 1: Workstation Perimeter.
- Level 2: Within the workstation.
- Level 3: Adjacent to robot arm.

[*] **Remotely Operated Robots:-**

[*] **AI in Robots:-**

[*] **Tele-operated robots.**