SASTRA

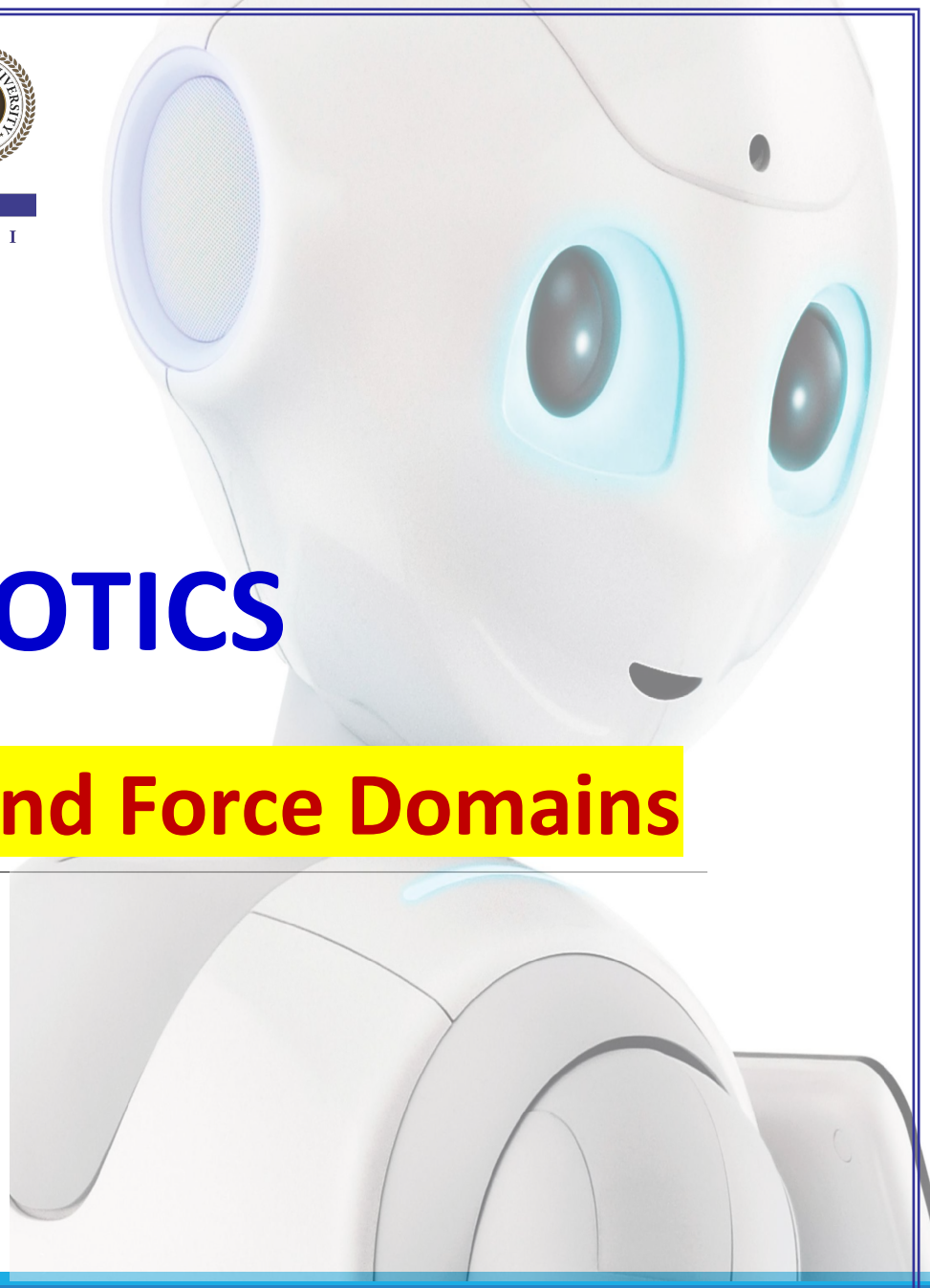
ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION

DEEMED TO BE UNIVERSITY

(U/S 3 of the UGC Act,1956)

THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

THANJAVUR | KUMBAKONAM | CHENNAI



MCT 301

INDUSTRIAL ROBOTICS

Module 3 - Jacobians in Velocity and Force Domains

DR. RM. KUPPAN CHETTY.,

M.TECH (SENSORS)., PH.D (ROBOTICS)

SMIEEE.,MIET., MIE., C.ENG.,

PROFESSOR, SCHOOL OF MECHANICAL ENGINEERING

Contents:

- **Spatial Descriptions and Mapping**
  - ✔ Workspace and Multiple Solutions
  - ✔ Derivation of Jacobian Matrix – Components
  - ✔ Singularity analysis, Dexterity, Degeneracy
  - ✔ Trajectory planning – Joint space and Task space
  - ✔ Path planning for mobile robots

Reference:

1. Saeed B Niku, Introduction to Robotics Analysis, Systems, Applications, Pearson Education
2. **R.K.Mittal and I.J.Nagrath, Robotics and control, Tata McGraw Hill Publishing Company Ltd, 2007.**
3. K.S. Fu, R.C. Gonzalez and C.S.G. Lee, Robotics, Control Sensing Vision and Intelligence, Mc Graw Hill Publications
4. https://robotacademy.net.au/masterclass/inverse-kinematics-and-robot-motion/

Functions of Jacobian Matrix:

- Helps to determine **the end-effecter velocity if we know the joint speeds**
- Helps to determine the **joint speeds to get a desired end-effecter speed**. For example, in robotic welding operation we know the desired EE speed.
- With the help of the Jacobian matrix we can determine the joints speeds to get the desired EE speed.
- Determinant of the Jacobian matrix is also an **indicator of the health/condition of the manipulator** at any configuration.
- If the determinant value is high, then the condition of the manipulator is good to carry out task; but if the determinant is close to zero we call that configuration as ill-conditioned. Such ill-conditioning happens at the boundary of the workspace.
- **Determination of singularity**. **At any point if the determinant of the Jacobian matrix becomes zero, that point is called singularity point.**

Suppose that we have a set of equations Y_i in terms of a set of variables x_j:

$$Y_i = f_i(x_1, x_2, x_3, \ldots, x_j). \tag{3.7}$$

The differential change in Y_i for a differential change in x_i is

$$\begin{cases} \delta Y_1 = \dfrac{\partial f_1}{\partial x_1}\delta x_1 + \dfrac{\partial f_1}{\partial x_2}\delta x_2 + \cdots + \dfrac{\partial f_1}{\partial x_j}\delta x_j, \\ \delta Y_2 = \dfrac{\partial f_2}{\partial x_1}\delta x_1 + \dfrac{\partial f_2}{\partial x_2}\delta x_2 + \cdots + \dfrac{\partial f_2}{\partial x_j}\delta x_j, \\ \vdots \\ \delta Y_i = \dfrac{\partial f_i}{\partial x_1}\delta x_1 + \dfrac{\partial f_i}{\partial x_2}\delta x_2 + \cdots + \dfrac{\partial f_i}{\partial x_j}\delta x_j. \end{cases} \tag{3.8}$$

$$\begin{bmatrix} \delta Y_1 \\ \delta Y_2 \\ \cdot \\ \cdot \\ \cdot \\ \delta Y_i \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_j} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_j} \\ & & \\ & & \\ & & \\ \frac{\partial f_i}{\partial x_1} & \frac{\partial f_i}{\partial x_2} & \frac{\partial f_i}{\partial x_j} \end{bmatrix} \begin{bmatrix} \delta x_1 \\ \delta x_2 \\ \\ \\ \\ \delta x_j \end{bmatrix}$$

$$\begin{bmatrix} dx \\ dy \\ dz \\ \delta x \\ \delta y \\ \delta z \end{bmatrix} = \begin{bmatrix} Robot \\ Jacobian \end{bmatrix} \begin{bmatrix} d\theta_1 \\ d\theta_2 \\ d\theta_3 \\ d\theta_4 \\ d\theta_5 \\ d\theta_6 \end{bmatrix}$$

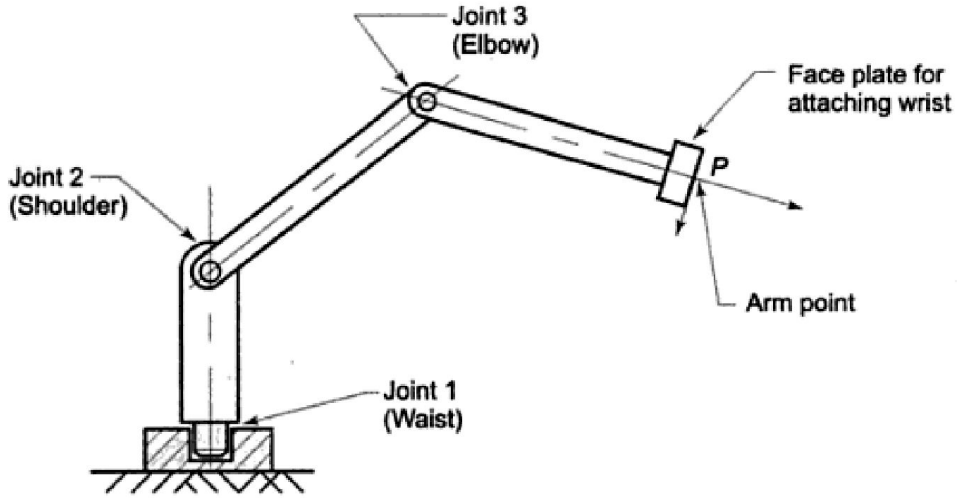
$$[D] = [J][D_\theta]$$

Jacobian Matrix of a Manipulator

Jacobian matrix of a manipulator maps joint speeds to Cartesian velocity of the end-effecter (EE). Mathematically,

$$V_e(t) = j(q)\dot{q} \text{ where} \quad \ldots\ldots\ldots\ldots(1)$$

$V_e(t)$ is the Cartesian velocity of the end-effecter
$j(q)$ is the Jacobian matrix and
$\dot{q}$ is the array of joint speeds



Picture taken from Robotics and Control by Nagrath and Mittal

For the 3-DOF articulated manipulator shown aside,

$$\underset{3x1}{V_e(t)} = \underset{3x3}{j(q)}\underset{3x1}{\dot{q}}$$

Since it is 3-DOF, the EE will have 3 DOFs; so $V_e(t)$ will be of 3x1; since there are three joints $\dot{q}$ will be an array of 3x1; hence *j(q)* will be 3x3.

Jacobian Matrix of a Manipulator

$$V_e(t) = j(q)\dot{q}$$

For 3-DOF manipulator, $$V_e(t) = \begin{bmatrix} j_1 & j_2 & j_3 \end{bmatrix} \begin{Bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{Bmatrix} = j_1\dot{q}_1 + j_2\dot{q}_2 + j_3\dot{q}_3 \quad \ldots\ldots\ldots\ldots(2)$$

3x1 [3x1 3x1 3x1] 3x1

3x3

So, for a 3-DOF manipulator, Jacobian matrix consists of 3 columns where each column is a contribution of the corresponding joint towards the end-effecter velocity. For example, j_1 is the contribution of the first joint towards the EE velocity. Similarly, j_2 is the contribution of the 2nd joint towards the EE velocity.

So, for a 6-DOF manipulator, Each j_i is 6x1 element.

$$V_e = \begin{bmatrix} V_e \\ \omega_e \end{bmatrix} = \begin{bmatrix} v_{ex} \\ v_{ey} \\ v_{ez} \\ \omega_{ex} \\ \omega_{ey} \\ \omega_{ez} \end{bmatrix} = \begin{bmatrix} j_1 & j_2 & j_3 & j_4 & j_5 & j_6 \end{bmatrix} \begin{Bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \\ \dot{q}_5 \\ \dot{q}_6 \end{Bmatrix}$$

6x1 6X6 6x1

Jacobian Matrix of a Manipulator

$$j_i = \begin{bmatrix} j_{vi} \\ j_{wi} \end{bmatrix} = \begin{bmatrix} j_{vxi} \\ j_{vyi} \\ j_{vzi} \\ j_{wxi} \\ j_{wyi} \\ j_{wzi} \end{bmatrix}$$

. In other words, for a 6-DOF manipulator

$$\underset{6x1}{\begin{bmatrix} v_{ex} \\ v_{ey} \\ v_{ez} \\ \omega_{ex} \\ \omega_{ey} \\ \omega_{ez} \end{bmatrix}} = \underset{6x6}{\begin{bmatrix} j_{vx1} & \cdot & \cdot & \cdot & \cdot & \cdot \\ j_{vy1} & \cdot & \cdot & \cdot & \cdot & \cdot \\ j_{vz1} & \cdot & \cdot & \cdot & \cdot & \cdot \\ j_{wx1} & \cdot & \cdot & \cdot & \cdot & \cdot \\ j_{wy1} & \cdot & \cdot & \cdot & \cdot & \cdot \\ j_{wz1} & \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix}} \underset{6x1}{\begin{Bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \\ \dot{q}_5 \\ \dot{q}_6 \end{Bmatrix}}$$

V_e = $J(q)$ $\dot{q}$

Derivation of Jacobian Matrix

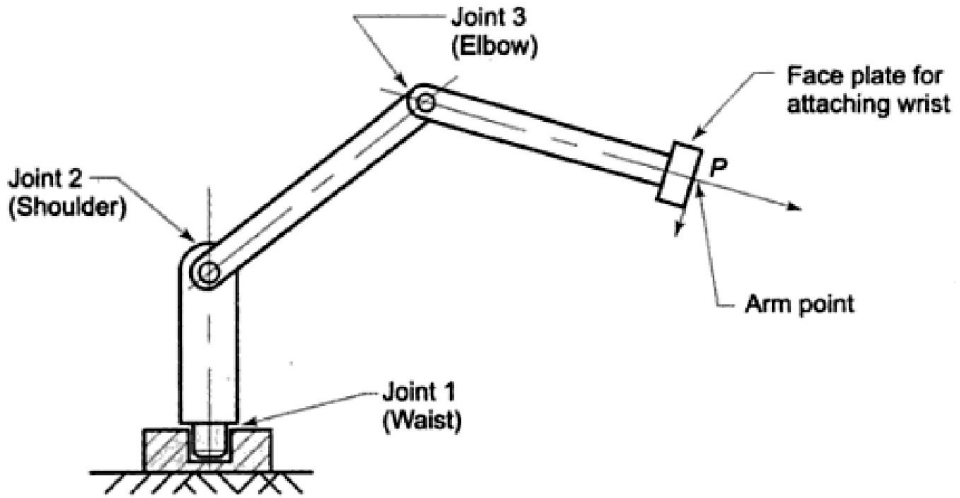
For the 3-DOF articulated manipulator shown aside,

$$V_e(t) = j(q)\dot{q}$$

~~3x1~~ ~~3x3~~ ~~3x1~~

6x1 6x3 3x1

To derive the Jacobian, we need the position of the EE which can be obtained from forward kinematics model.



Picture taken from Robotics and Control by Nagrath and Mittal

$$^0T_3 = {}^0T_1\,{}^1T_2\,{}^2T_3 = \begin{bmatrix} C_1C_{23} & -C_1S_{23} & S_1 & C_1(L_3C_{23}+L_2C_2) \\ S_1C_{23} & -S_1S_{23} & -C_1 & S_1(L_3C_{23}+L_2C_2) \\ S_{23} & C_{23} & 0 & L_3S_{23}+L_2S_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

So, position of the EE along X-axis (p_{ex}) = $C\theta_1(L_3C\theta_{23}+L_2C\theta_2)$……(1)

$p_{ey} = S\theta_1(L_3C\theta_{23}+L_2C\theta_2)$ …….(2)

$p_{ez} = L_3S\theta_{23}+L_2S\theta_2$……....(3)

$$p_{ex} = C\theta_1(L_3C\theta_{23} + L_2C\theta_2) = L_3cos\theta_1cos(\theta_2+\theta_3) + L_2cos\theta_1cos\theta_2$$

$$\Rightarrow v_{ex} = \frac{\partial p_{ex}}{\partial \theta_1} * \frac{d\theta_1}{dt} + \frac{\partial p_{ex}}{\partial \theta_2} * \frac{d\theta_2}{dt} + \frac{\partial p_{ex}}{\partial \theta_3} * \frac{d\theta_3}{dt}$$

$$= \frac{\partial p_{ex}}{\partial \theta_1} * \dot{\theta}_1 + \frac{\partial p_{ex}}{\partial \theta_2} * \dot{\theta}_2 + \frac{\partial p_{ex}}{\partial \theta_3} * \dot{\theta}_3$$

$$= (-L_3S\theta_1C\theta_{23} - L_2S\theta_1C\theta_2) * \dot{\theta}_1 +$$
$$(-L_3cos\theta_1sin(\theta_2+\theta_3) - L_2cos\theta_1sin\theta_2) * \dot{\theta}_2 - L_3cos\theta_1sin(\theta_2+\theta_3) * \dot{\theta}_3$$

$$So, \qquad v_{ex} = \begin{bmatrix} (-L_3S\theta_1C\theta_{23} - L_2S\theta_1C\theta_2) & (-L_3cos\theta_1sin(\theta_2+\theta_3) - L_2cos\theta_1sin\theta_2) & -L_3cos\theta_1sin(\theta_2+\theta_3) \end{bmatrix} \begin{Bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{Bmatrix}$$

.....(4)

$$p_{ey} = S\theta_1(L_3C\theta_{23}+L_2C\theta_2) \text{ .....(2)}$$

$$p_{ey} = L_3 sin\theta_1 cos(\theta_2+\theta_3) + L_2 sin\theta_1 cos\theta_2$$

$$\Rightarrow v_{ey} = \frac{\partial p_{ey}}{\partial \theta_1} * \frac{d\theta_1}{dt} + \frac{\partial p_{ey}}{\partial \theta_2} * \frac{d\theta_2}{dt} + \frac{\partial p_{ey}}{\partial \theta_3} * \frac{d\theta_3}{dt}$$

$$= (L_3 cos\theta_1 cos(\theta_2+\theta_3) + L_2 cos\theta_1 cos\theta_2) * \dot{\theta}_1 +$$

$$(-L_3 sin\theta_1 sin(\theta_2+\theta_3) - L_2 sin\theta_1 sin\theta_2) * \dot{\theta}_2 + (-L_3 sin\theta_1 sin(\theta_2+\theta_3) * \dot{\theta}_3$$

$$So, \quad v_{ey} = \begin{bmatrix} (L_3 cos\theta_1 cos\theta_{23} + L_2 cos\theta_1 cos\theta_2) & (-L_3 sin\theta_1 sin(\theta_2+\theta_3) - L_2 sin\theta_1 sin\theta_2) & -L_3 sin\theta_1 sin(\theta_2+\theta_3] \end{bmatrix} \begin{Bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{Bmatrix}$$

.....(5)

$$p_{ez} = L_3 sin(\theta_2 + \theta_3) + L_2 sin\theta_2$$

$$\Rightarrow v_{ez} = \frac{\partial p_{ez}}{\partial \theta_1} * \frac{d\theta_1}{dt} + \frac{\partial p_{ez}}{\partial \theta_2} * \frac{d\theta_2}{dt} + \frac{\partial p_{ez}}{\partial \theta_3} * \frac{d\theta_3}{dt}$$

$$= (0) * \dot{\theta}_1 +$$

$$(L_3 cos(\theta_2 + \theta_3) + L_2 cos\theta_2) * \dot{\theta}_2 + (L_3 cos(\theta_2 + \theta_3)) * \dot{\theta}_3$$

$$So, \qquad v_{ez} = \begin{bmatrix} 0 & (L_3 cos(\theta_2 + \theta_3) + L_2 cos\theta_2) & L_3 cos(\theta_2 + \theta_3) \end{bmatrix} \begin{Bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{Bmatrix} \qquad .....(6)$$

Combining Eqns 4,5 and 6,

$$\begin{bmatrix} v_{ex} \\ v_{ey} \\ v_{ez} \end{bmatrix} = \begin{bmatrix} -L_3 S\theta_1 C\theta_{23} - L_2 S\theta_1 C\theta_2 & -L_3 cos\theta_1 sin(\theta_2+\theta_3) - L_2 cos\theta_1 sin\theta_2 & -L_3 cos\theta_1 sin(\theta_2+\theta_3) \\ L_3 cos\theta_1 cos\theta_{23} + L_2 cos\theta_1 cos\theta_2 & -L_3 sin\theta_1 sin(\theta_2+\theta_3) - L_2 sin\theta_1 sin\theta_2 & -L_3 sin\theta_1 sin(\theta_2+\theta_3 \\ 0 & L_3 cos(\theta_2+\theta_3) + L_2 cos\theta_2 & L_3 cos(\theta_2+\theta_3) \end{bmatrix} \begin{Bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{Bmatrix}$$

In the beginning we had, $V_e(t) = j(q)\dot{q}$

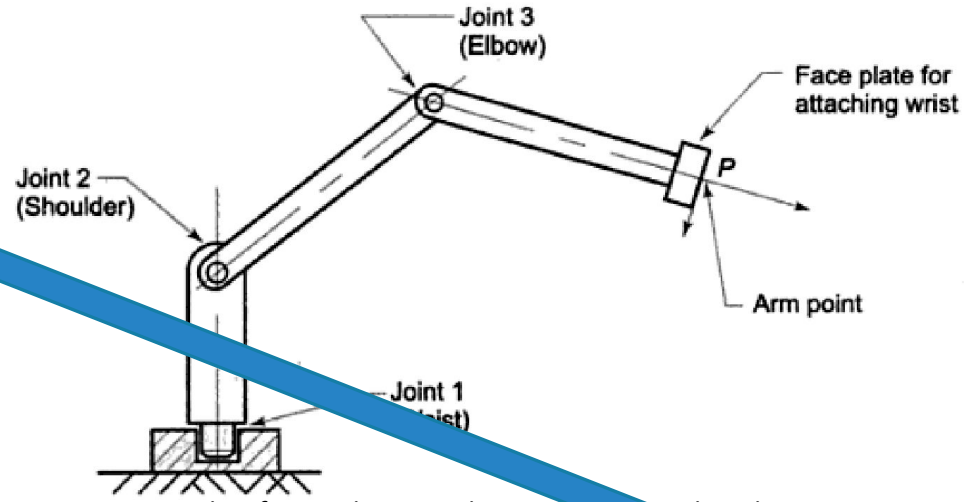
6x1 6x3 3x1

$$\begin{bmatrix} v_{ex} \\ v_{ey} \\ v_{ez} \\ \omega_{ex} \\ \omega_{ey} \\ \omega_{ez} \end{bmatrix} = \begin{bmatrix} -L_3 S\theta_1 C\theta_{23} - L_2 S\theta_1 C\theta_2 & -L_3 cos\theta_1 sin(\theta_2+\theta_3) - L_2 cos\theta_1 sin\theta_2 & -L_3 cos\theta_1 sin(\theta_2+\theta_3) \\ L_3 cos\theta_1 cos\theta_{23} + L_2 cos\theta_1 cos\theta_2 & -L_3 sin\theta_1 sin(\theta_2+\theta_3) - L_2 sin\theta_1 sin\theta_2 & -L_3 sin\theta_1 sin(\theta_2+\theta_3 \\ 0 & (L_3 cos(\theta_2+\theta_3) + L_2 cos\theta_2) & L_3 cos(\theta_2+\theta_3) \\ ? & ? & ? \\ ? & ? & ? \\ ? & ? & ? \end{bmatrix} \begin{Bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{Bmatrix}$$

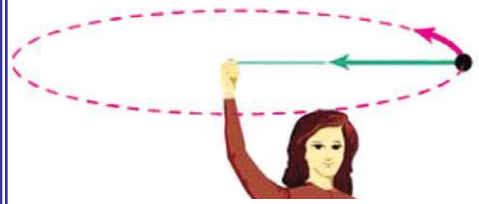
Derivation of Angular Components of Jacobian Matrix

$$\begin{bmatrix} v_{ex} \\ v_{ey} \\ v_{ez} \\ \omega_{ex} \\ \omega_{ey} \\ \omega_{ez} \end{bmatrix} = \begin{bmatrix} j_{vx1} & \cdot & \cdot & \cdot & \cdot & \cdot \\ j_{vy1} & \cdot & \cdot & \cdot & \cdot & \cdot \\ j_{vz1} & \cdot & \cdot & \cdot & \cdot & \cdot \\ j_{wx1} & \cdot & \cdot & \cdot & \cdot & \cdot \\ j_{wy1} & \cdot & \cdot & \cdot & \cdot & \cdot \\ j_{wz1} & \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix} \begin{Bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \\ \dot{q}_5 \\ \dot{q}_6 \end{Bmatrix}$$

$V_e = J(q)\,\dot{q}$

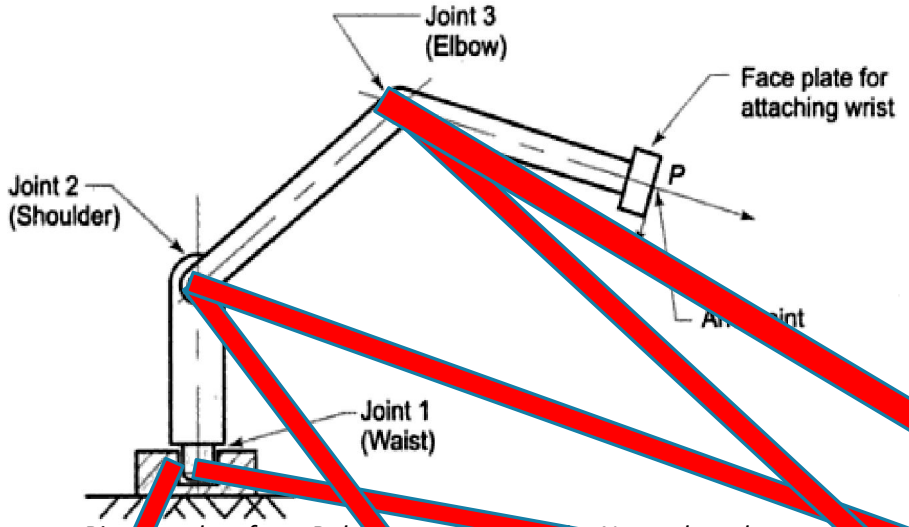


Picture taken from Robotics and Control by ...ath and ...



Angular velocity of the stone is imparted from the finger.
Angular velocity is a vector having direction and amplitude.
Direction is the direction of the (joint) axis and magnitude is a scalar in degrees/sec.

$$\text{Angular velocity} = \begin{Bmatrix} 0 \\ 0 \\ 1 \end{Bmatrix} \dot{\theta}$$



Picture taken from Robotics and Control, Nagrath and Mittal

$$\begin{bmatrix} v_{ex} \\ v_{ey} \\ v_{ez} \\ \omega_{ex} \\ \omega_{ey} \\ \omega_{ez} \end{bmatrix} = \begin{bmatrix} -L_3S\theta_1C\theta_{23} - L_2S\theta_1C\theta_2 & -L_3cos\theta_1 sin(\theta_2+\theta_3) - L_2cos\theta_1 sin\theta_2 & -L_3cos\theta_1 sin(\theta_2+\theta_3) \\ L_3cos\theta_1 cos\theta_{23} + L_2cos\theta_1 cos\theta_2 & -L_3sin\theta_1 sin(\theta_2+\theta_3) - L_2sin\theta_1 sin\theta_2 & -L_3sin\theta_1 sin(\theta_2+\theta_3) \\ 0 & (L_3cos(\theta_2+\theta_3) + L_2cos\theta_2) & L_3cos(\theta_2+\theta_3) \\ ? & ? & ? \\ ? & ? & ? \\ ? & ? & ? \end{bmatrix} \begin{Bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{Bmatrix}$$

$$\begin{bmatrix} v_{ex} \\ v_{ey} \\ v_{ez} \\ \omega_{ex} \\ \omega_{ey} \\ \omega_{ez} \end{bmatrix} = \begin{bmatrix} -L_3S\theta_1C\theta_{23} - L_2S\theta_1C\theta_2 & -L_3cos\theta_1sin(\theta_2+\theta_3) - L_2cos\theta_1sin\theta_2 & -L_3cos\theta_1sin(\theta_2+\theta_3) \\ L_3cos\theta_1cos\theta_{23} + L_2cos\theta_1cos\theta_2 & -L_3sin\theta_1sin(\theta_2+\theta_3) - L_2sin\theta_1sin\theta_2 & -L_3sin\theta_1sin(\theta_2+\theta_3 \\ 0 & (L_3cos(\theta_2+\theta_3) + L_2cos\theta_2) & L_3cos(\theta_2+\theta_3) \\ ? & ? & ? \\ ? & ? & ? \\ ? & ? & ? \end{bmatrix} \begin{Bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{Bmatrix}$$

$$\begin{Bmatrix} 0 \\ 0 \\ 1 \end{Bmatrix}$$

This is the direction of the 1st joint axis, i.e. Z_0.

This is the direction of the 2nd joint axis, i.e. Z_1. Write Z_1 w.r.t base frame (frame 0). 3rd Column of ${}^0T_1$ is Z_1.

This is the direction of the 3rd joint axis, i.e. Z_2. Write Z_2 w.r.t base frame (frame 0). 3rd Column of ${}^0T_2$ is Z_2.

$${}^0T_1 = \begin{bmatrix} C1 & 0 & S1 & 0 \\ S1 & 0 & -C1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad {}^0T_2 = \begin{bmatrix} C1C2 & -C1S2 & S1 & L2C1C2 \\ S1C2 & -S1S2 & -C1 & L2S1C2 \\ S2 & C2 & 0 & L2S2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$
\begin{bmatrix} v_{ex} \\ v_{ey} \\ v_{ez} \\ \omega_{ex} \\ \omega_{ey} \\ \omega_{ez} \end{bmatrix} = \begin{bmatrix} -L_3 S\theta_1 C\theta_{23} - L_2 S\theta_1 C\theta_2 & -L_3 cos\theta_1 sin(\theta_2+\theta_3) - L_2 cos\theta_1 sin\theta_2 & -L_3 cos\theta_1 sin(\theta_2+\theta_3) \\ L_3 cos\theta_1 cos\theta_{23} + L_2 cos\theta_1 cos\theta_2 & -L_3 sin\theta_1 sin(\theta_2+\theta_3) - L_2 sin\theta_1 sin\theta_2 & -L_3 sin\theta_1 sin(\theta_2+\theta_3 \\ 0 & (L_3 cos(\theta_2+\theta_3) + L_2 cos\theta_2) & L_3 cos(\theta_2+\theta_3) \\ 0 & S1 & S1 \\ 0 & -C1 & -C1 \\ 1 & 0 & 0 \end{bmatrix} \begin{Bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{Bmatrix}
$$

Determine the Jacobian of the spherical arm given that ${}^{0}T_{1}(\theta_1)=\begin{bmatrix} C_1 & 0 & S_1 & 0 \\ S_1 & 0 & -C_1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$;

${}^{1}T_{2}(\theta_2)=\begin{bmatrix} C_2 & 0 & -S_2 & 0 \\ S_2 & 0 & C_2 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$; ${}^{2}T_{3}(d_3)=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$;

$$0T3=\begin{bmatrix} C_1C_2 & -S_1 & -C_1S_2 & -C_1S_2d_3 \\ S_1C_2 & C_1 & -S_1S_2 & -S_1S_2d_3 \\ S_2 & 0 & C_2 & C_2d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

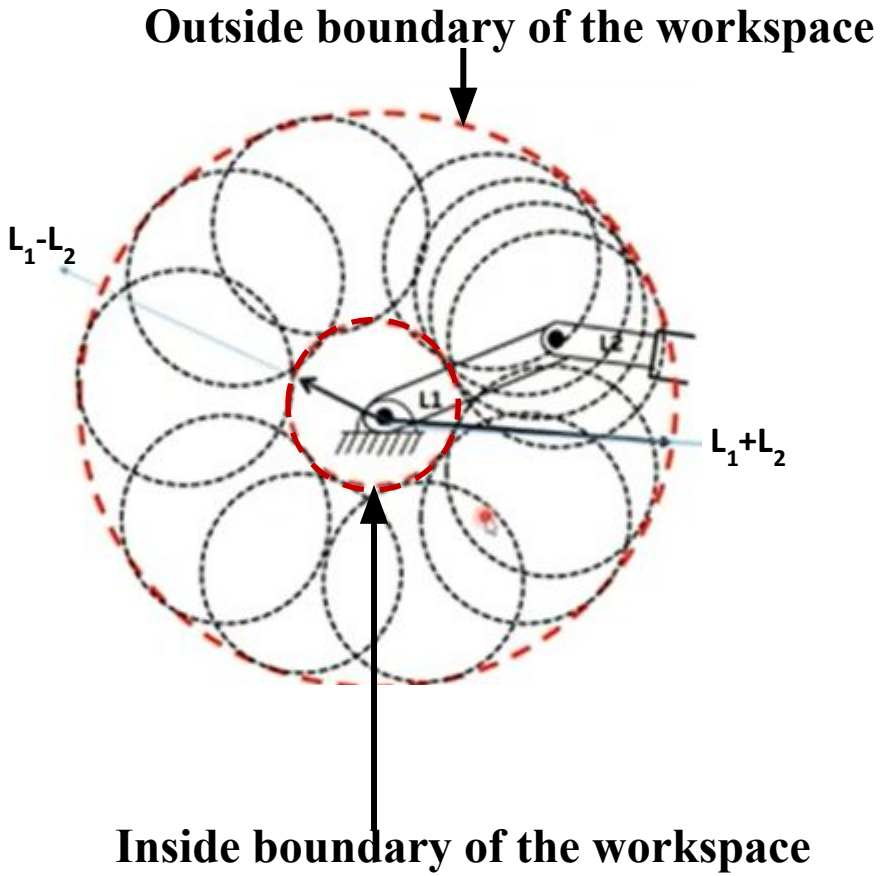
$$J_v=\begin{bmatrix} S_1S_2d_3 & -C_1C_2d_3 & -C_1S_2 \\ -C_1S_2d_3 & -S_1C_2d_3 & -S_1S_2 \\ 0 & -S_2d_3 & C_2 \end{bmatrix}$$

$$J=\begin{bmatrix} S_1S_2d_3 & -C_1C_2d_3 & -C_1S_2 \\ -C_1S_2d_3 & -S_1C_2d_3 & -S_1S_2 \\ 0 & -S_2d_3 & C_2 \\ 0 & C_1 & 0 \\ 0 & S_1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

$$J_\omega=\begin{bmatrix} 0 & C_1 & 0 \\ 0 & S_1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

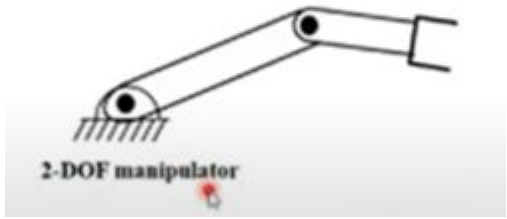
Singularity Analysis

- Singularity is a point within the workspace where the manipulator loses all controllability.
- Mathematically, singularity occurs when the Jacobian matrix (velocity jacobian) becomes singular, i.e., det |J|=0.
- At singularity the end-effecter may lose or gain an extra degree(s) of freedom which is very dangerous for the manipulator
- Since the determinant of the Jacobian matrix goes zero, considering the relationship $V_e = J\dot{q}$, for a finite end-effecter speed, the required joint speed becomes infinite ($\dot{q} = J^{-1} V_e$).
- For a serial manipulator, the singularity occurs when the manipulator is fully folded back or fully stretched out. That is, singularity for a serial manipulator occurs at the boundary of the workspace wherein they lose degree(s) of freedom.



Singularity Analysis

$$V_e = \begin{bmatrix} -s\theta_1 - s\theta_{12} & -s\theta_{12} \\ c\theta_1 + c\theta_{12} & c\theta_{12} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix} \dot{q}$$



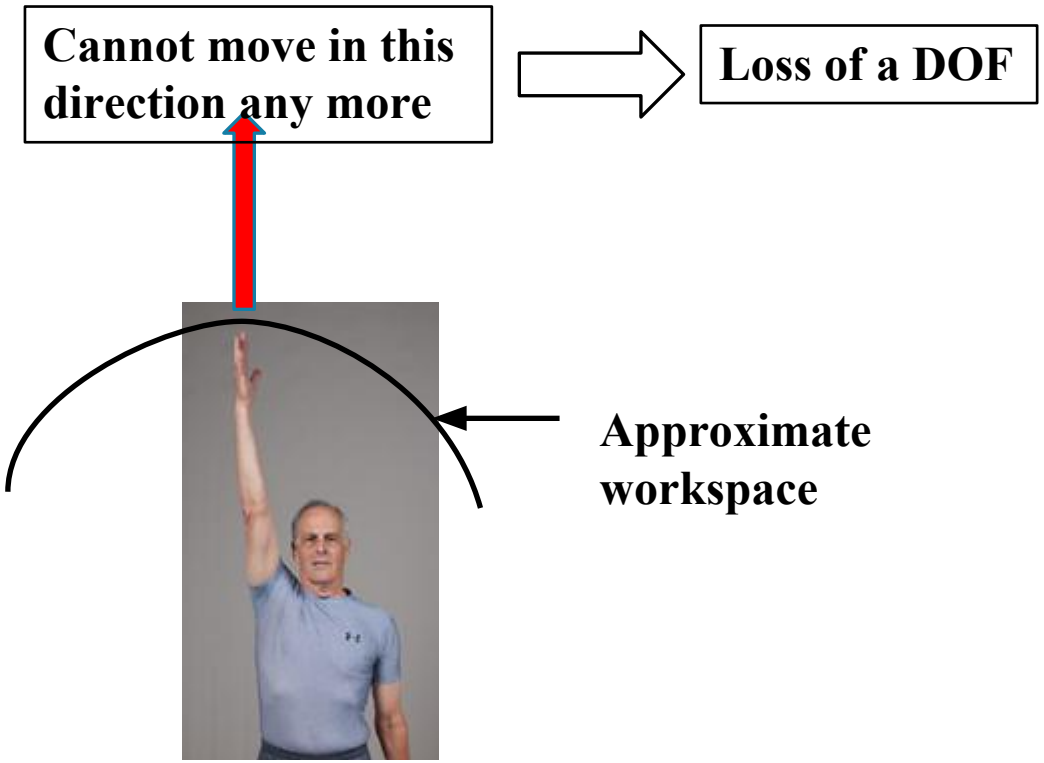
Since a 2-DOF manipulator has got two translational degrees of freedom, only top two rows of the Jacobian matrix constitute the significant Jacobian. So, the velocity relationship is as follows:

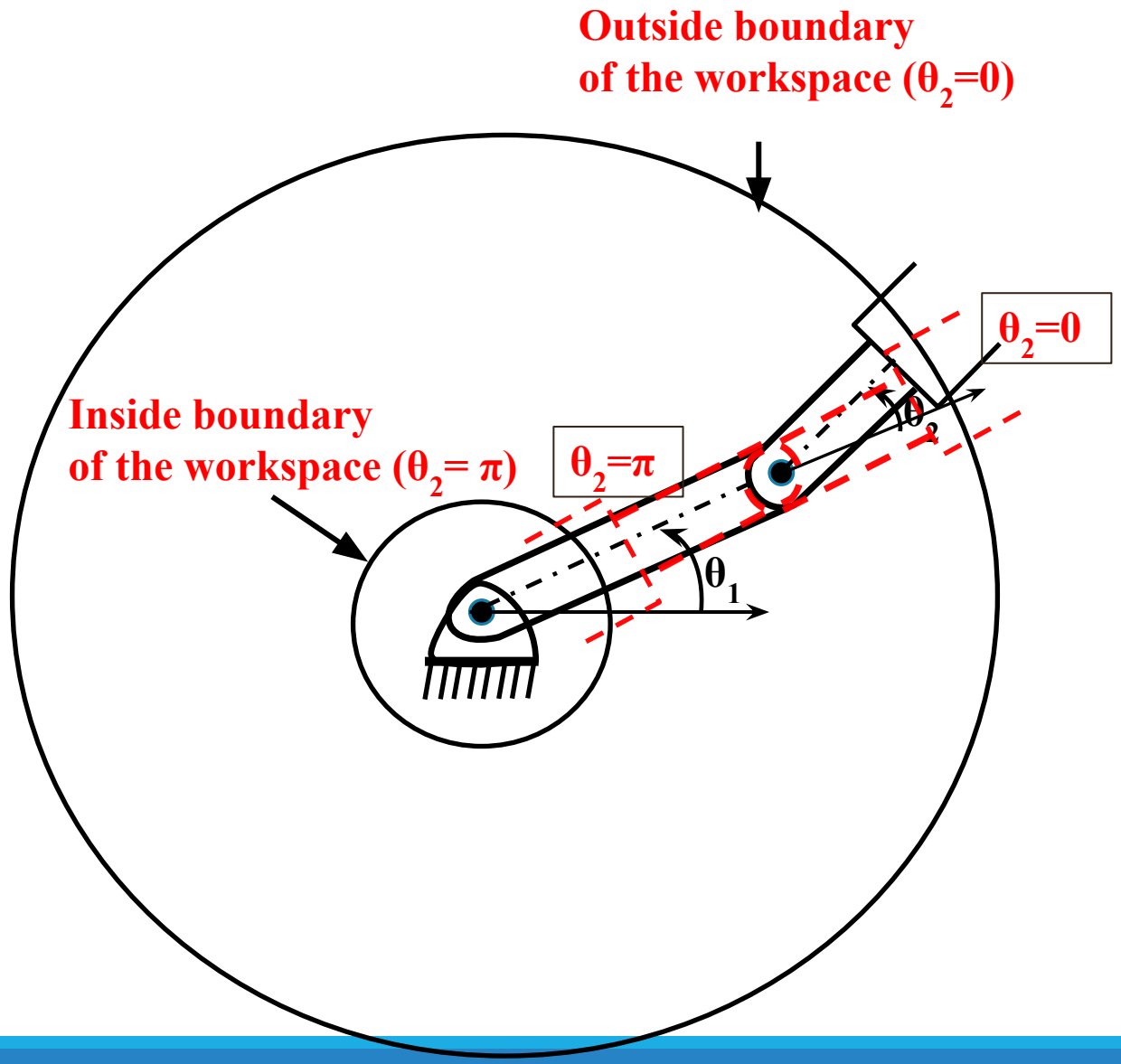
$$\begin{Bmatrix} v_{ex} \\ v_{ey} \end{Bmatrix} = \begin{bmatrix} -s\theta_1 - s\theta_{12} & -s\theta_{12} \\ c\theta_1 + c\theta_{12} & c\theta_{12} \end{bmatrix} \begin{Bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{Bmatrix} \quad \text{So} \quad J = \begin{bmatrix} -s\theta_1 - s\theta_{12} & -s\theta_{12} \\ c\theta_1 + c\theta_{12} & c\theta_{12} \end{bmatrix}$$

$$det|J| = -s\theta_1 * c\theta_{12} - s\theta_{12} * c\theta_{12} + s\theta_{12} * c\theta_1 + c\theta_{12}s\theta_{12}$$
$$= s\theta_{12} * c\theta_1 - s\theta_1 * c\theta_{12} = sin(\theta_1 + \theta_2 - \theta_1) = sin\theta_2$$

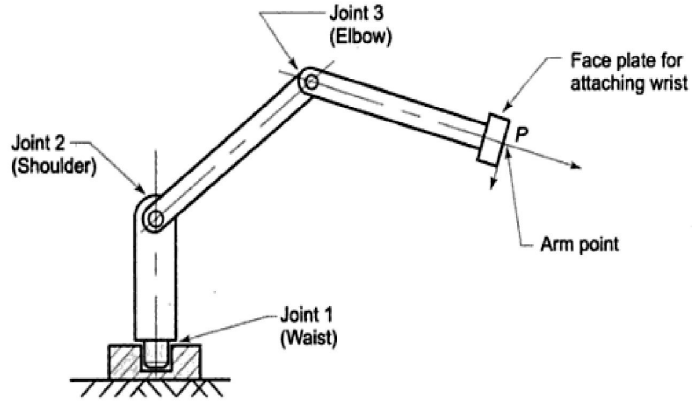
For singularity, det |J|=0.

So $\sin\theta_2=0 \Rightarrow \theta_2=0$ or $\theta_2=\pi$ for any value of θ_1





3-DOF Articulated Manipulator



$$\begin{bmatrix} v_{ex} \\ v_{ey} \\ v_{ez} \\ \omega_{ex} \\ \omega_{ey} \\ \omega_{ez} \end{bmatrix} = \begin{bmatrix} -L_3 S\theta_1 C\theta_{23} - L_2 S\theta_1 C\theta_2 & -L_3 cos\theta_1 sin(\theta_2+\theta_3) - L_2 cos\theta_1 sin\theta_2 & -L_3 cos\theta_1 sin(\theta_2+\theta_3) \\ L_3 cos\theta_1 cos\theta_{23} + L_2 cos\theta_1 cos\theta_2 & -L_3 sin\theta_1 sin(\theta_2+\theta_3) - L_2 sin\theta_1 sin\theta_2 & -L_3 sin\theta_1 sin(\theta_2+\theta_3 \\ 0 & (L_3 cos(\theta_2+\theta_3) + L_2 cos\theta_2) & L_3 cos(\theta_2+\theta_3) \\ 0 & S1 & S1 \\ 0 & -C1 & -C1 \\ 1 & 0 & 0 \end{bmatrix} \begin{Bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{Bmatrix}$$

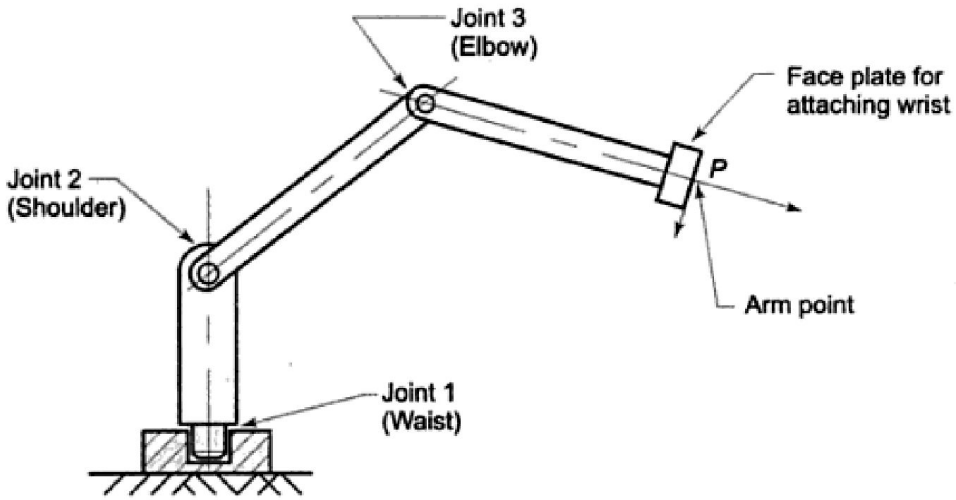
Since, it is going to have a spherical volume, any point inside that volume only needs x, y and z coordinates to define the end-effecter. In other words, it would have only three linear velocities. So,

$$\begin{bmatrix} v_{ex} \\ v_{ey} \\ v_{ez} \end{bmatrix} = \begin{bmatrix} -L_3 S\theta_1 C\theta_{23} - L_2 S\theta_1 C\theta_2 & -L_3 cos\theta_1 sin(\theta_2+\theta_3) - L_2 cos\theta_1 sin\theta_2 & -L_3 cos\theta_1 sin(\theta_2+\theta_3) \\ L_3 cos\theta_1 cos\theta_{23} + L_2 cos\theta_1 cos\theta_2 & -L_3 sin\theta_1 sin(\theta_2+\theta_3) - L_2 sin\theta_1 sin\theta_2 & -L_3 sin\theta_1 sin(\theta_2+\theta_3 \\ 0 & (L_3 cos(\theta_2+\theta_3) + L_2 cos\theta_2) & L_3 cos(\theta_2+\theta_3) \end{bmatrix} \begin{Bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{Bmatrix}$$

The Jacobian in Statics

$$V_e = J(q)\dot{q}$$

- This Jacobian matrix *(J(q))* is called Velocity Jacobian. There is another Jacobian matrix called, Jacobian in Statics
- Maps the end-effecter force and moment to the required joint torques.
- In other words, given the end-effecter force and moment, this Jacobian matrix can determine the required joint torques.

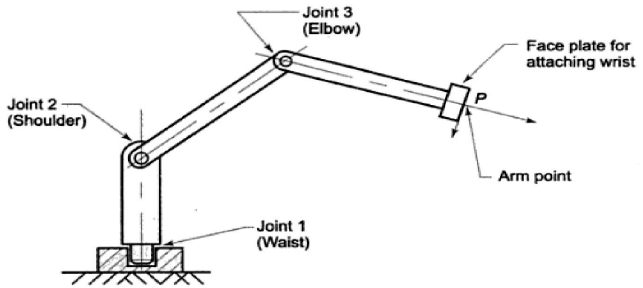


Picture taken from Robotics and Control by Nagrath and Mittal

Derivation of Jacobian in Statics

Principle of virtual work: Assume infinitesimal displacements at the joints, and end-effecter. Then the total work done is as follows.

$$\delta W = \tau_1 \delta q_1 + \tau_2 \delta q_2 + ... + \tau_n \delta q_n - (f_{n,n+1})^T \delta x_e - (\eta_{n,n+1})^T \delta \phi_e$$



Picture taken from Robotics and Control by Nagrath and Mittal

where

δW = Virtual work done

$\tau_1, \tau_2, ..., \tau_n$ = Joint torques in the 'n' joints

$\delta q_1, \delta q_2, ..., \delta q_n$ = Infinitesimal displacement in each joint

$f_{n,n+1}$ = Force exerted by the end-effecter (n^{th} link) on the environment ($n+1^{th}$ link).

So the reaction force from the environment to the end-effecter will be $-f_{n,n+1}$

$\eta_{n,n+1}$ = Moment exerted by the end-effecter (n^{th} link) on the environment ($n+1^{th}$ link).

So the reaction moment from the environment to the end-effecter will be $-\eta_{n,n+1}$

δx_e = Infintesimal end-effecter displacement

$\delta \phi e$ = Infintesimal end-effecter rotation

$$\delta W = \begin{bmatrix} \tau_1 & \tau_2 & . & . & \tau_n \end{bmatrix} \begin{bmatrix} \delta q_1 \\ \delta q_2 \\ . \\ . \\ \delta q_n \end{bmatrix} - \begin{bmatrix} f_{n,n+1} & \eta_{n,n+1} \end{bmatrix} \begin{bmatrix} \delta x_e \\ \delta \phi_e \end{bmatrix}$$

$\Rightarrow \delta W = \tau^T \delta q - F^T \delta p$

where τ : array of joint torques (a column matrix)

δq : array of infinitesimal joint displacements

F : End-effcter force and moment

δp : end-effecter displacements$(\delta X_e, \delta \phi_e)$

We know, $V_e = J(q)\dot{q}$. In terms of instantaneous displacement, this can be written as $\delta p = J(q)\delta q$ where $V_e = \lim_{\delta t \to 0} \frac{\delta p}{\delta t}$, and $\dot{q} = \lim_{\delta t \to 0} \frac{\delta q}{\delta t}$.

Substitutingδp in the above expression, we get

$\delta W = \tau^T \delta q - F^T J(q)\delta q = (\tau^T - F^T J(q))\delta q$

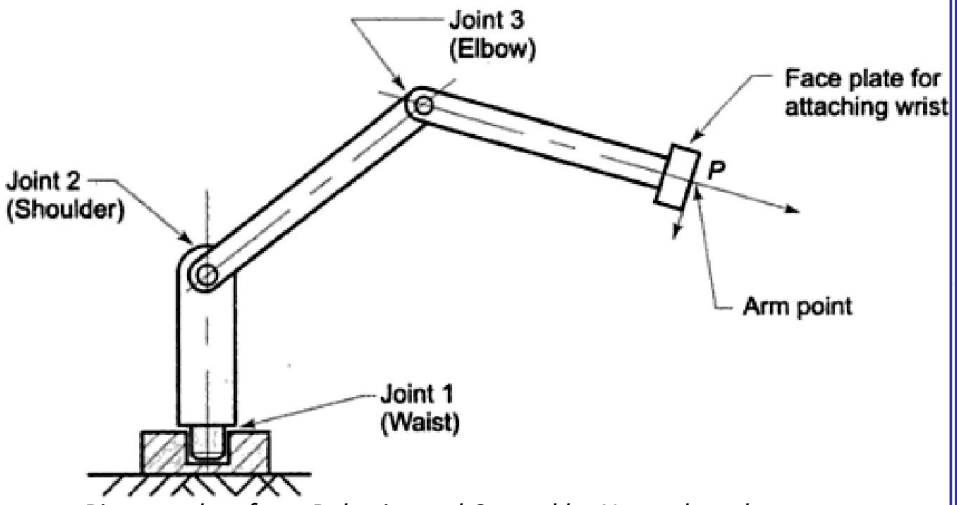
$= (\tau - J^T(q)F)^T \delta q$ since $(ab)^T = b^T a^T$

According to principle of virtual work, the manipulator is in static equilibrium.

$$\text{So } \delta W = 0.$$

$$\text{Hence } (\tau - J^T(q)F)^T \Rightarrow \boxed{\tau = J^T(q)F}$$

This $J^T(q)$ is called the Jacobian in statics.



Picture taken from Robotics and Control by Nagrath and Mittal

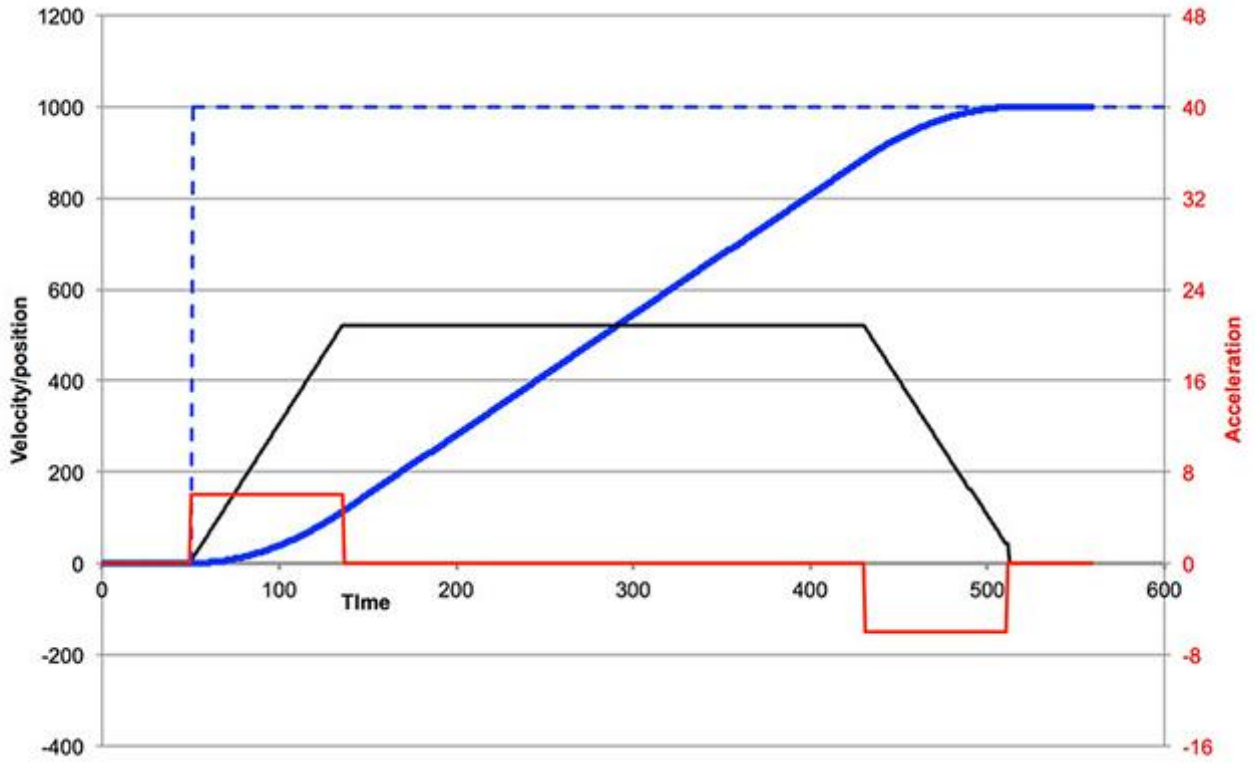
Example: *The 3-DOF articulated manipulator has to exert* $[20\ 30\ -40\ 0\ 0\ 0]^T$ *at the end-effecter. Determine the joint torqu required.*

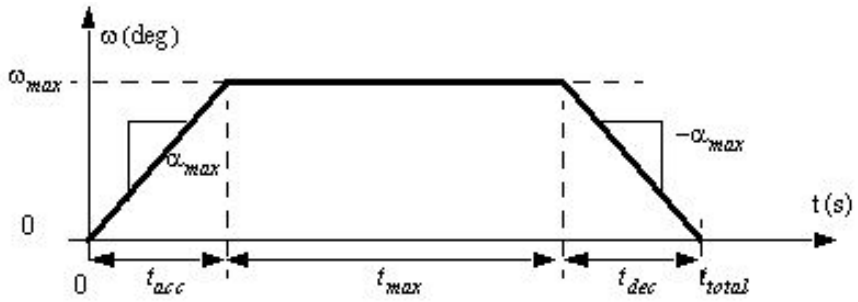
Solution:

First determine velocity Jacobian $J(q)$. It will be 6x3 matrix; $J^T(q)$ will be 3x6 matrix. Now 'F' is 6x1. So 3x6 and 6x1, will give you 3x1 matrix of joint torques. Since there are 3 joints, we need 3 joint torques.

Motion Profiles

- Defines how the motor should behave – in terms of position, velocity, and acceleration – during the move w.r.t time.
- Motion:
  - Moving along a tool from point A to B in a straight line or Circular or complex paths.
  - Complex paths are called Trajectories or Motion Profile





where,

ω_{max} = the maximum velocity

α_{max} = the maximum acceleration

t_{acc}, t_{dec} = the acceleration and deceleration times

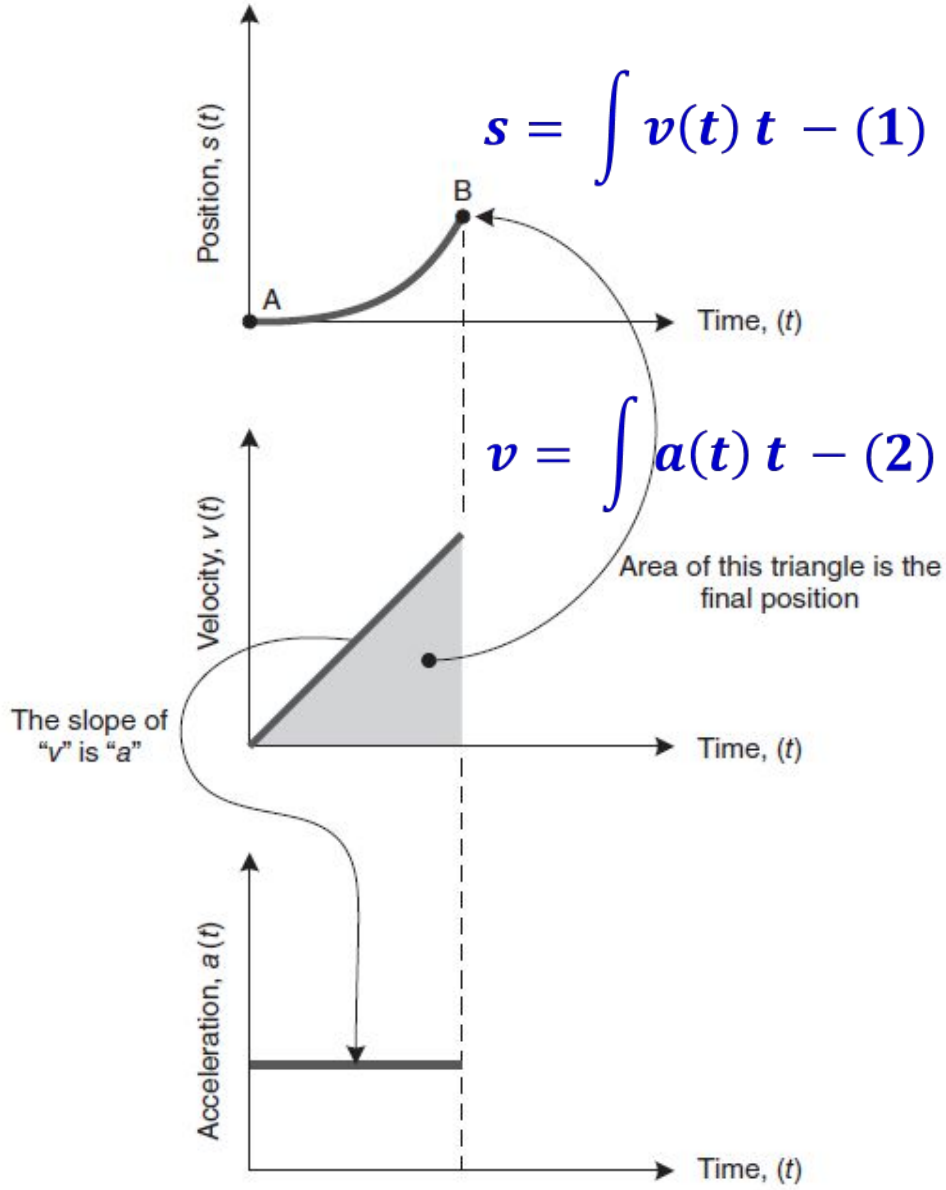
t_{max} = the times at the maximum velocity

t_{total} = the total motion time

Motion controller generates the motion profile at regular intervals to create velocity and position commands for the servo control system of each motor.

Motion Profiles - Concepts

- Common Motion Profiles are
  - Trapezoidal
  - Triangular
  - S- Shaped – Sigmoidal
- Kinematics of the system is used for motion profile calculations
  - Governs the relationship between time, position, velocity and acceleration
- Consider and Axis moving from "A" to "B"
  - $S(t) = Position\ along\ the\ trajectory$
  - $v(t) = \frac{ds}{dt} = Velocity$ = Change of position $S(t)$ at a given time interval
  - $a(t) = \frac{dv}{dt}$ = Acceleration = Change of Velocity $v(t)$ at a given time interval



Basic relationships between position, velocity, and acceleration

Motion Profiles - Concepts

Geometric Rules for Motion Profile

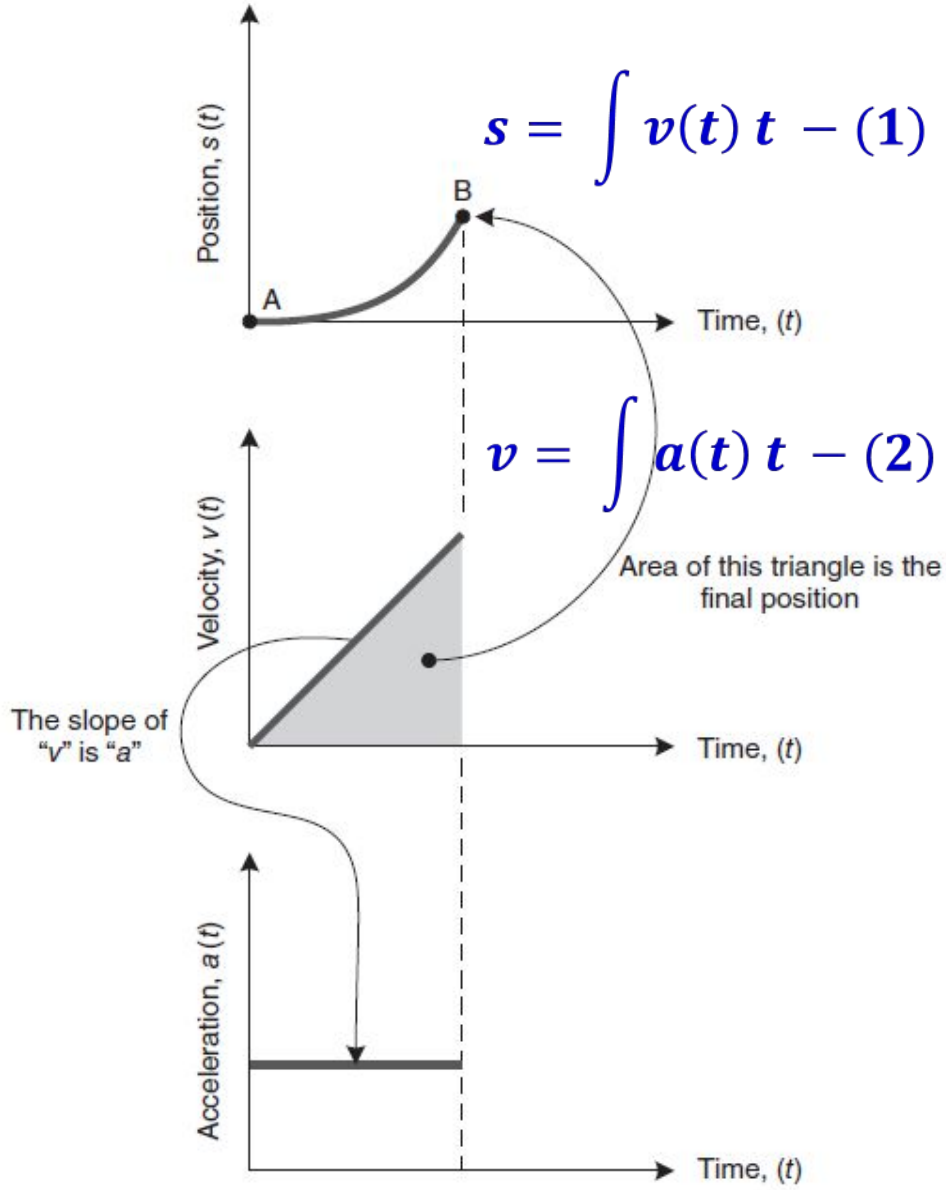
- **Position at time *t* is equal to the area under the velocity curve up to time *t***
- **Acceleration is the slope of the velocity curve**

From equation (1) and (2)

$$v = v_0 + a(t - t_0)$$

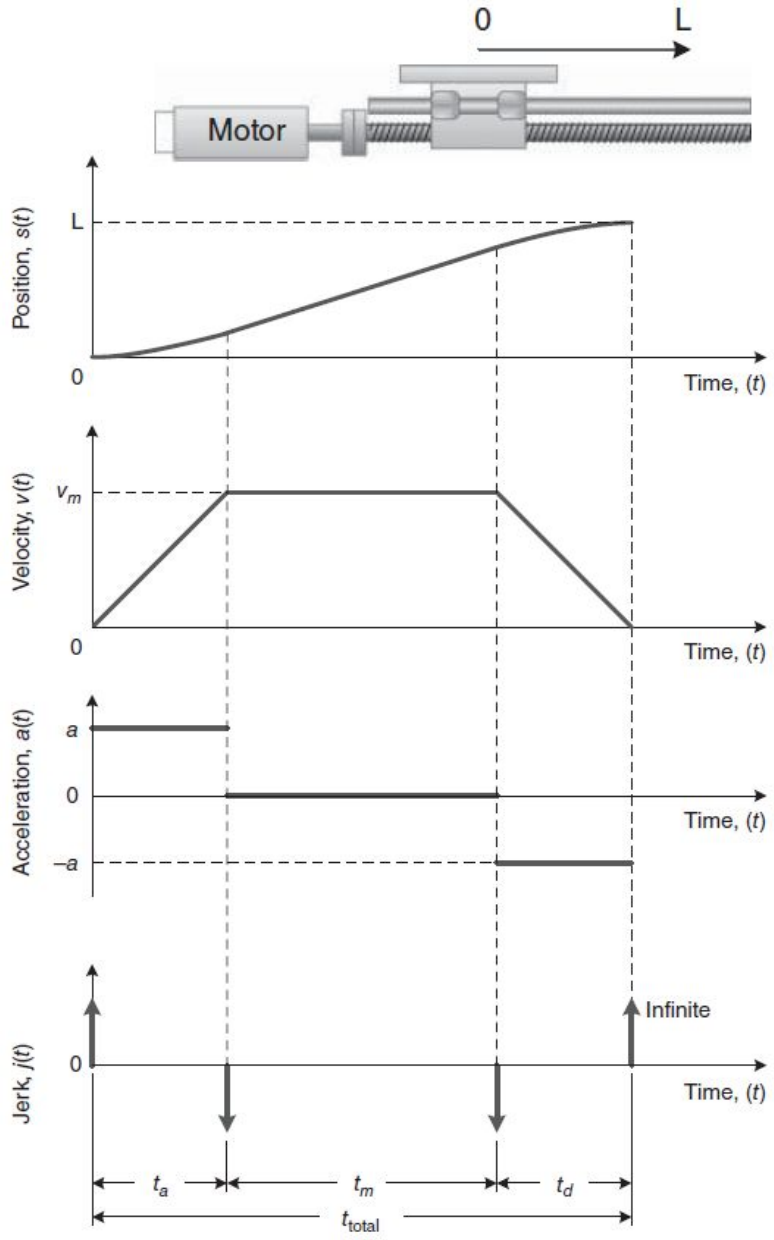
$$s = s_0 + v_0(t - t_0) + \frac{1}{2}a(t - t_0)^2$$

Where, t_0 is the initial time, v_0 is the initial velocity, and S_0 is the initial position. The acceleration "a" is constant.



Trapezoidal velocity profile

- Area under the curve is the total distance moved
- Slope of the initial and final ramp are the maximum acceleration and deceleration
- Top level of the trapezoid is the maximum velocity
- Continuous acceleration, but there will be a jerk (third order derivative) at the four sharp corners
- Three distinct phases in the motion
  - Acceleration
  - Constant Velocity (Zero Acceleration)
  - Deceleration
- Desired motion parameters to move an axis of the machine are
  - **Move velocity v_m**
  - **Acceleration a**
  - **Distance s to be travelled by the axis**



Trapezoidal velocity profile and associated position, acceleration, and jerk profiles to move an axis from 0 to position L

Geometric Approach

- Goal is to determine move time t_m
- W.k.t acceleration a is the slope of the velocity curve

$$\therefore a = \frac{v_m}{t_a} \qquad (1)$$

- $t_a = t_d = \frac{v_m}{a}$ (2)
- The total motion time is as follows

$$t_{total} = t_a + t_m + t_d \qquad (3)$$

- **Total distance travelled by the axis is found by**

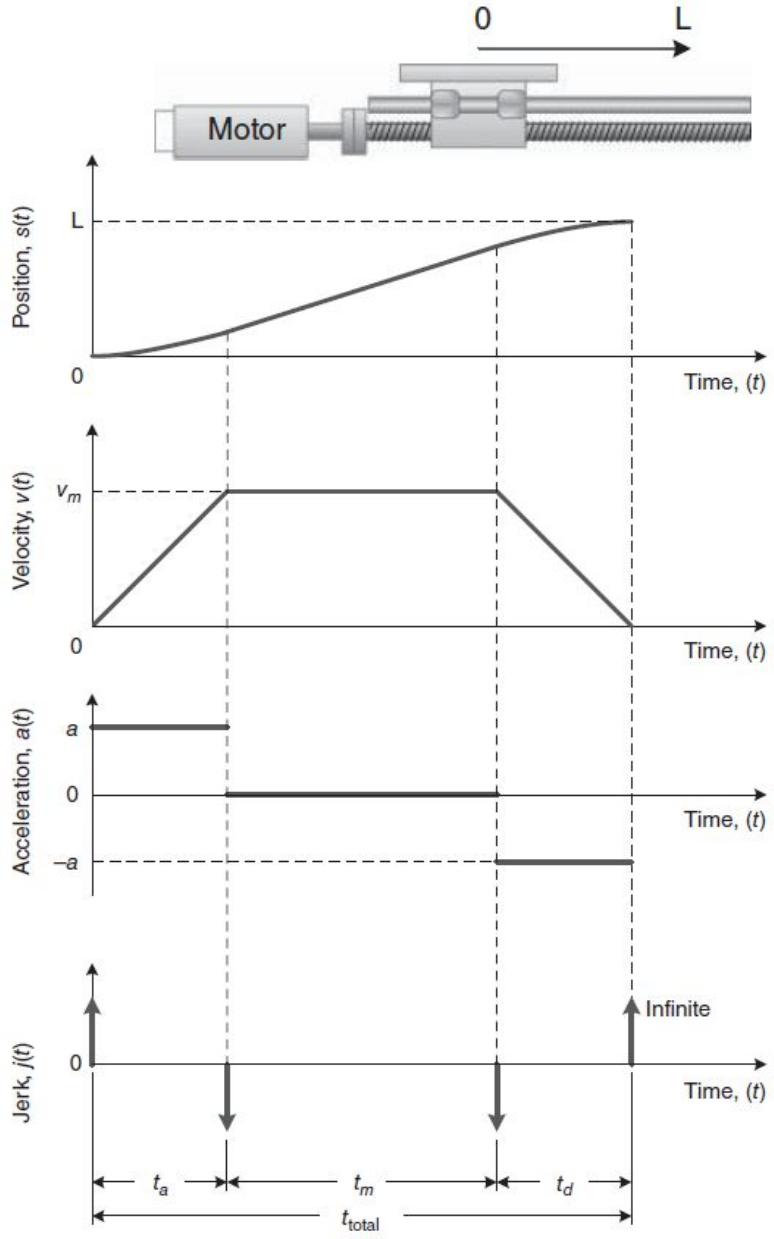
$$L = \frac{t_a v_m}{2} + t_m v_m + \frac{t_d v_m}{2} \qquad (4)$$

When $t_a = t_d$

$$L = v_m(t_a + t_m)$$

$$t_m = \frac{L}{v_m} - t_a$$

Analytical Approach ?



Trapezoidal velocity profile and associated position, acceleration, and jerk profiles to move an axis from 0 to position L

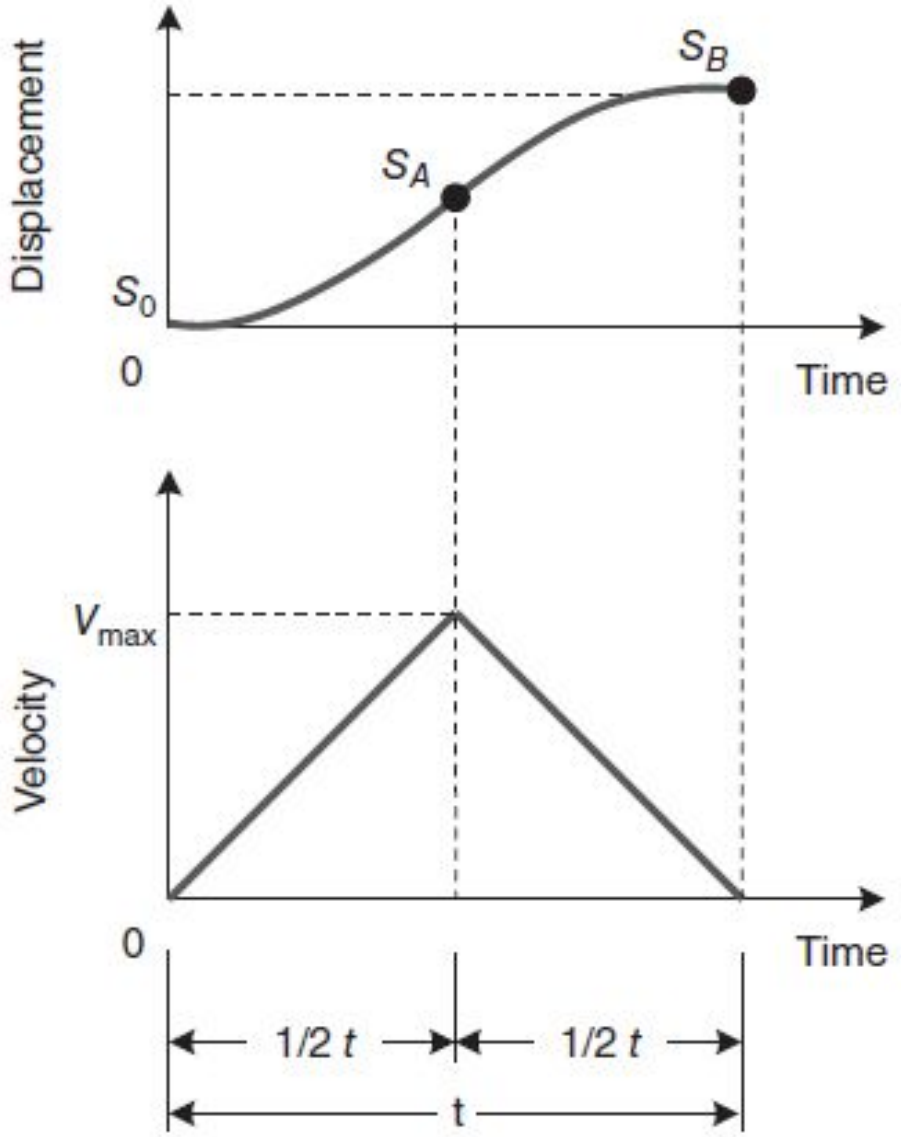
Triangular velocity profile

- Characterized by equal acceleration and deceleration times (distance travelled)
- Acceleration period
- Deceleration period
- Provides fastest movements between two points
- Applications that don't require a period of constant velocity
- Height of the triangle represents the maximum velocity
- Acceleration is half the move time

$$s_B = \frac{1}{2} v_{\max} \frac{t}{2} + \frac{1}{2} v_{\max} \frac{t}{2}$$
$$= \frac{1}{2} v_{\max} t$$

$$v_{\max} = \frac{2 s_B}{t}$$

$$a = \frac{2 v_{\max}}{t}$$



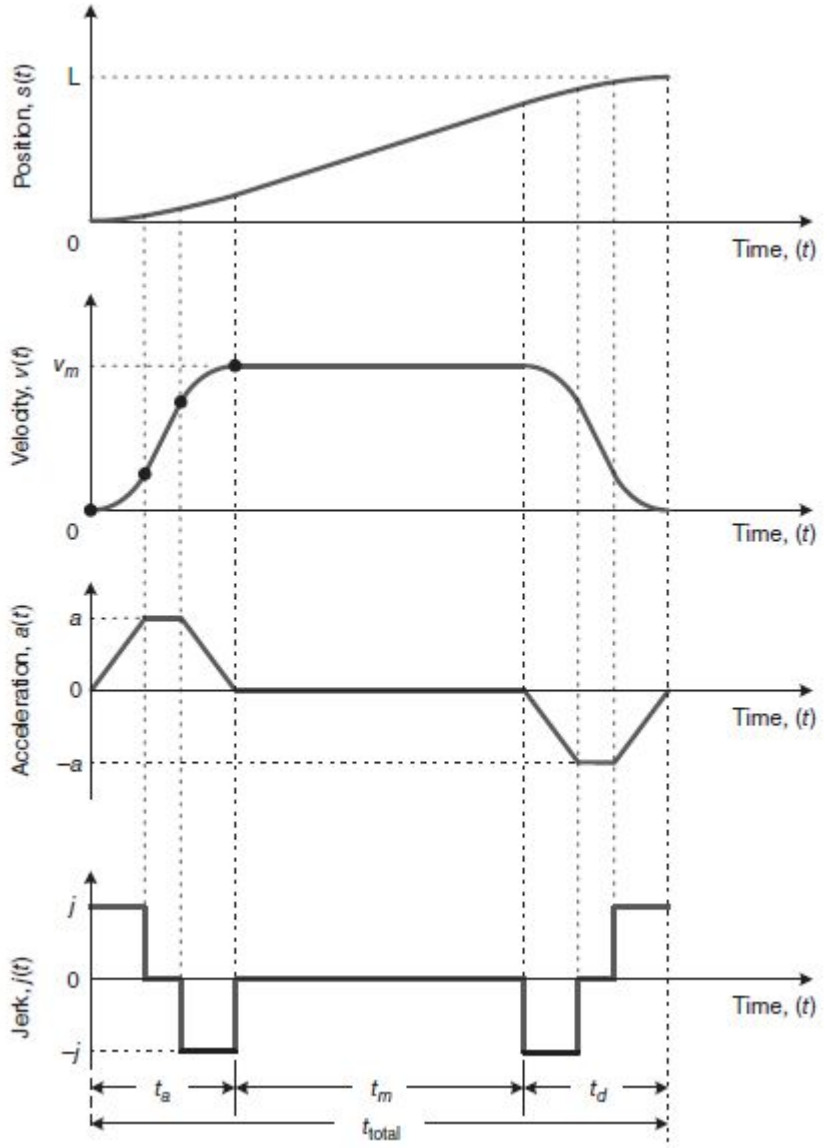
S - Curve velocity profile

- **Smoothing the beginning and end of the acceleration and deceleration phases of motion**
- **limits the rate of change of acceleration and deceleration (jerk)**
- **Second order Polynomial functions are used to generate the rounded corners**
- **two pieces of quadratic curves designated as “A” and “B”**

$$v(t) = c_o + c_1 t + c_2 t^2 \qquad a(t) = c_1 + 2\,c_2\,t$$

$$j(t) = 2\,c_2$$

where C_0, C_1, and C_2 are coefficients



S-curve velocity profile and the associated position, acceleration, and jerk profiles

S - Curve velocity profile

S-CURVE VELOCITY PROFILE (FIGURE 2.9)

$$t_a = \frac{2v_m}{a}$$

$$C_1 = \frac{a^2}{2v_m}$$

Curve A

$0 \le t \le \frac{t_a}{2}$

$$s_A(t) = C_1 \frac{t^3}{3}$$

$$v_A(t) = C_1 t^2$$

$$a_A(t) = 2C_1 t$$

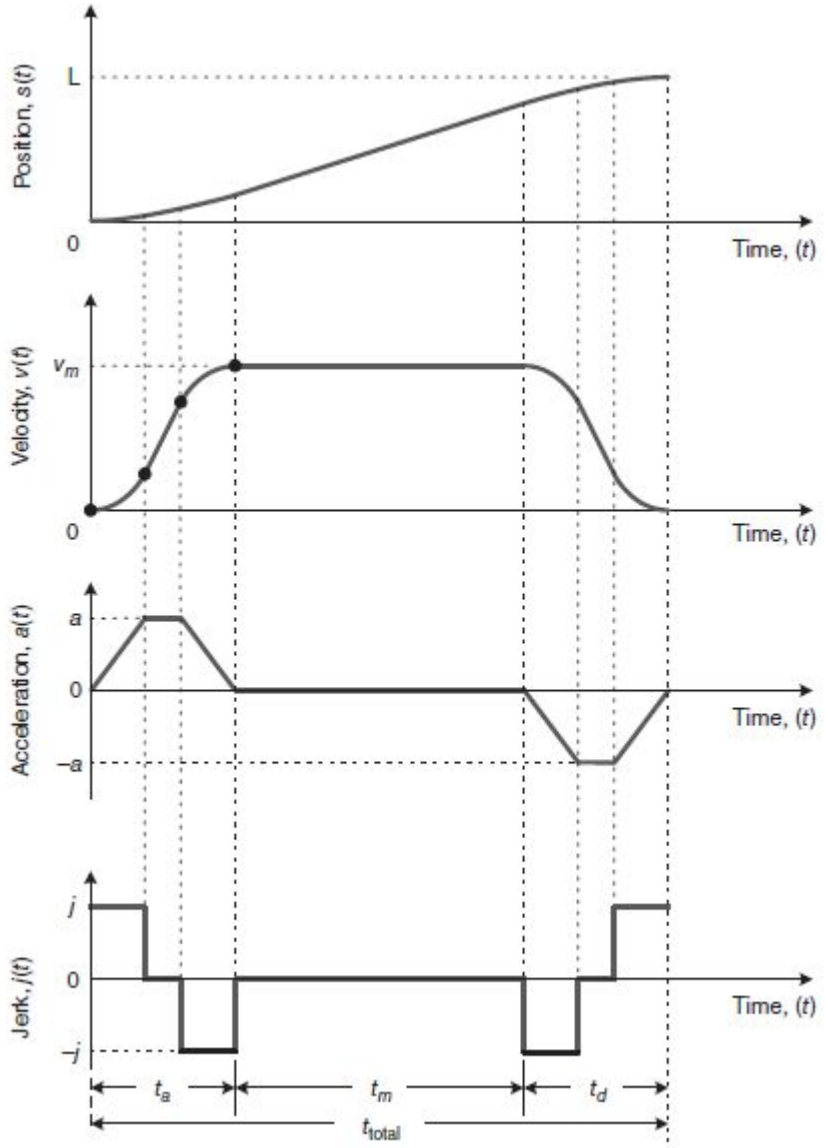
Curve B

$\frac{t_a}{2} < t \le t_a$

$$s_B(t) = C_1 \frac{t_a^3}{24} + v_m \left(t - \frac{t_a}{2}\right) - C_1 \cdot \left\{ t_a^2 \left(t - \frac{t_a}{2}\right) - t_a \left(t^2 - \left(\frac{t_a}{2}\right)^2\right) + \frac{1}{3}\left(t^3 - \left(\frac{t_a}{2}\right)^3\right) \right\}$$

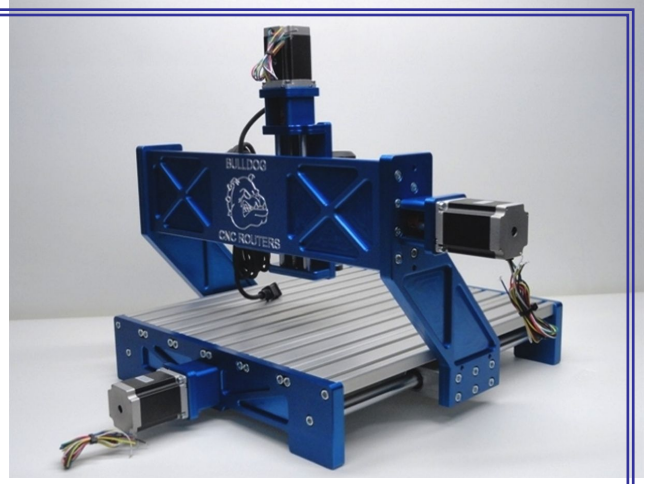
$$v_B(t) = v_m - C_1 (t_a - t)^2$$

$$a_B(t) = 2C_1 (t_a - t)$$



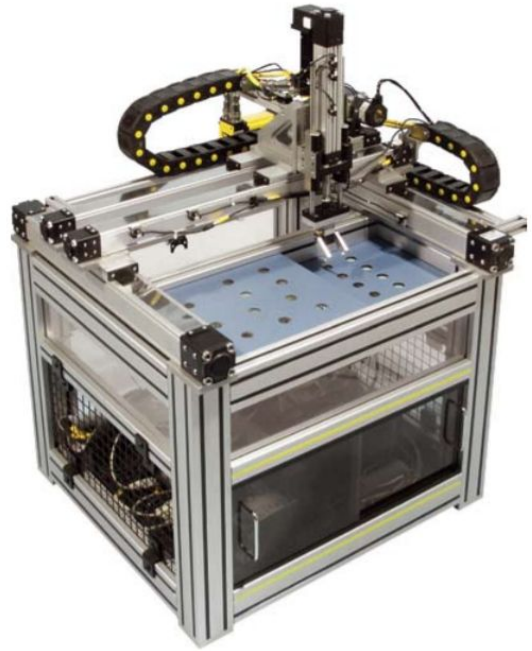
S-curve velocity profile and the associated position, acceleration, and jerk profiles

Multi Axes Motion



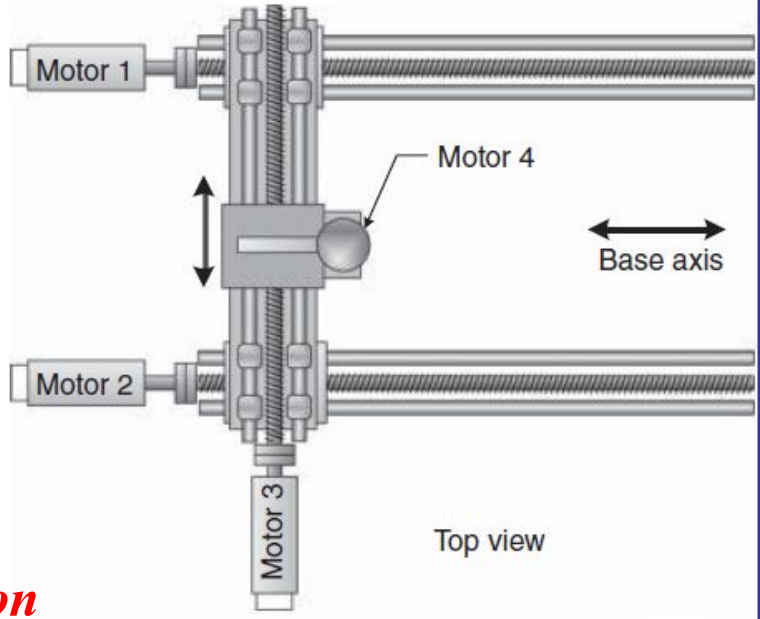
- Systems having motion in more than one axis
- Requires coordination of the individual axis to complete the task
- Example – CNC milling machines, Gantry Robots

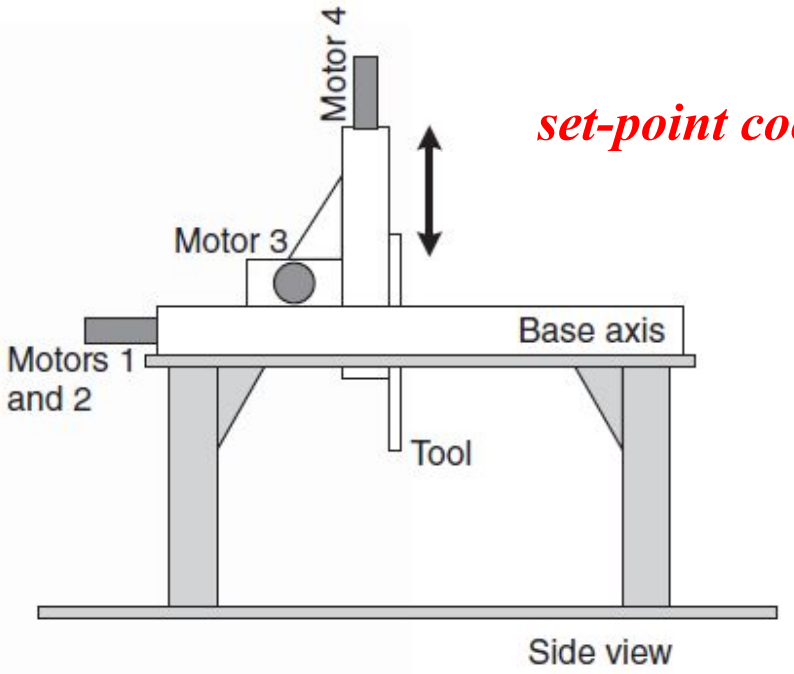
- **Coordination approaches in moving the axis of the machines are**
  - **Move one axis at a time**
  - **Start moving all axis at the same time – Slew Motion**
  - **Adjust the motion of the axes so that they all start and finish at the same time – Interpolated Motion**



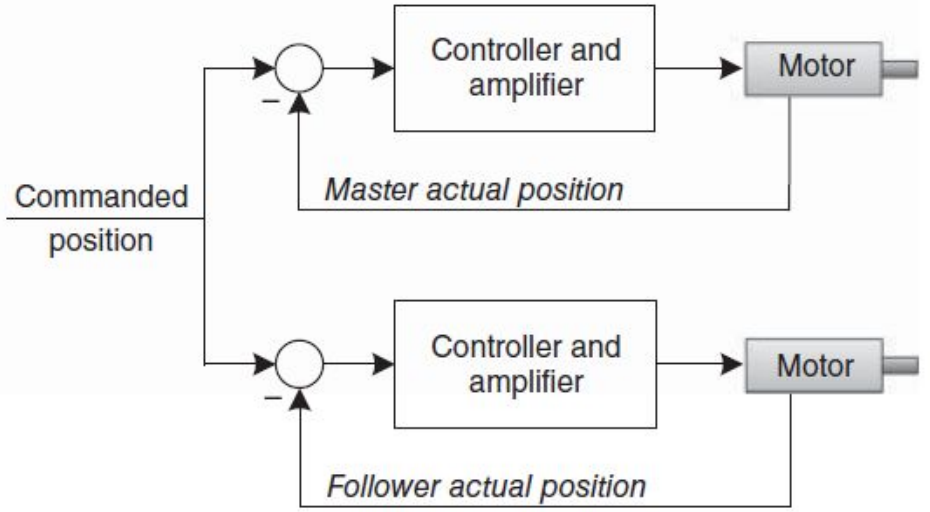
Multi-Axis Motion

- Multi-axis motion may involve
  - Multiple motors driving one axis
  - Coordinated motion of two or more axes
  - Following using master/follower synchronization
  - Tension control, or
  - Kinematics.





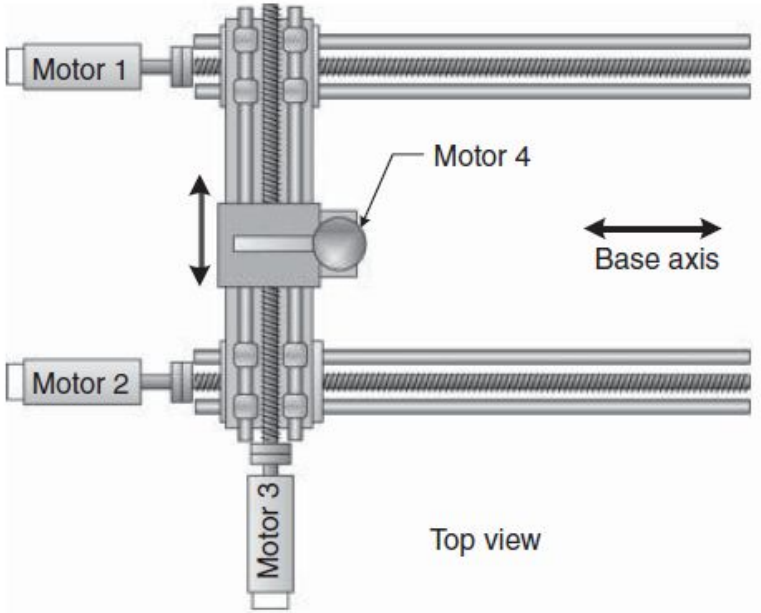
set-point coordinated motion

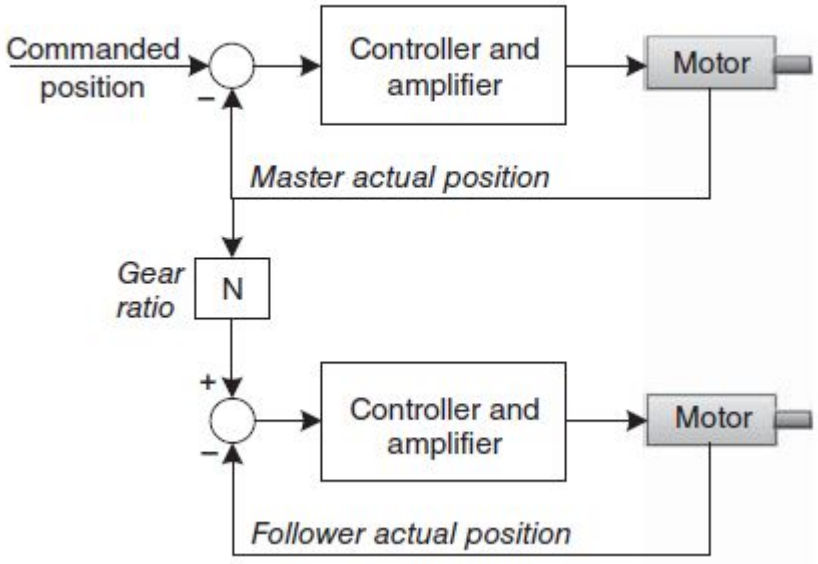


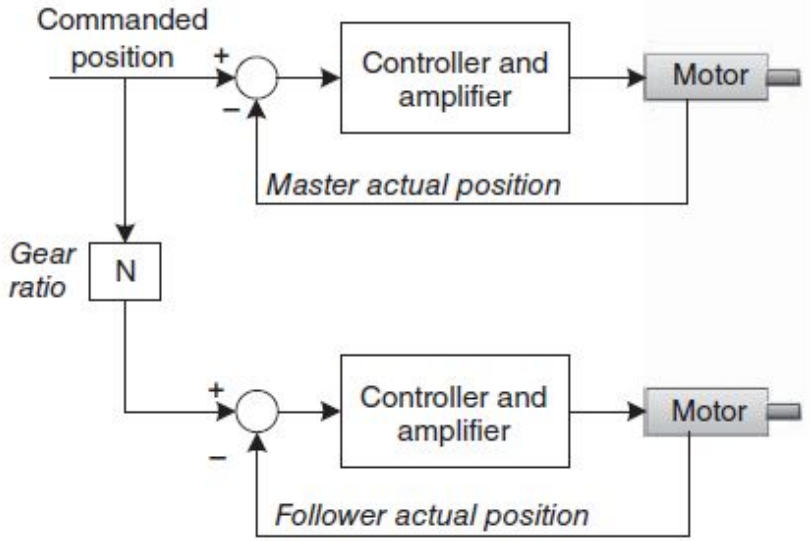
Gantry machine with set-point coordinated base axis. (a) Gantry machine with two motors driving the base axis. (b) Same set-point commanded to both motors of the base axis

Multi-Axis Motion

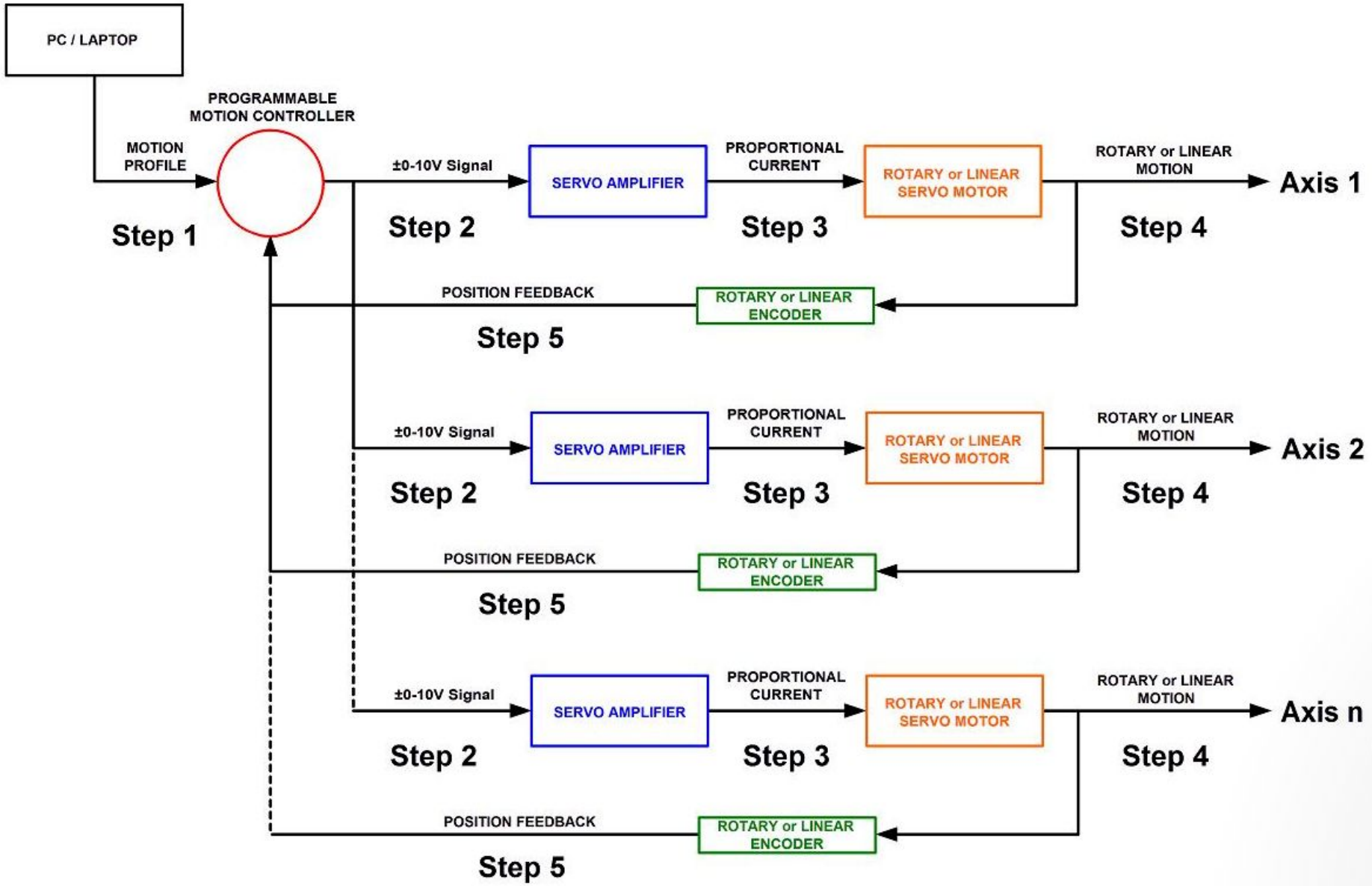
- **Coordinated motion of two or more axes**
  - Each motor is driving a single axis, but all axes move in coordination
  - All axis are assigned same coordinate system
- **Master/Slave Synchronization**
  - Master axis and slave axes
  - Constant gear ratio called electronic gearing







MULTI AXIS: LINEAR MOTOR DRIVEN POSITIONING STAGE



Motion profiles

(speed, acceleration, deceleration, position, PID etc.)

Voltage loop (convert the parameters into voltages based on error signals)

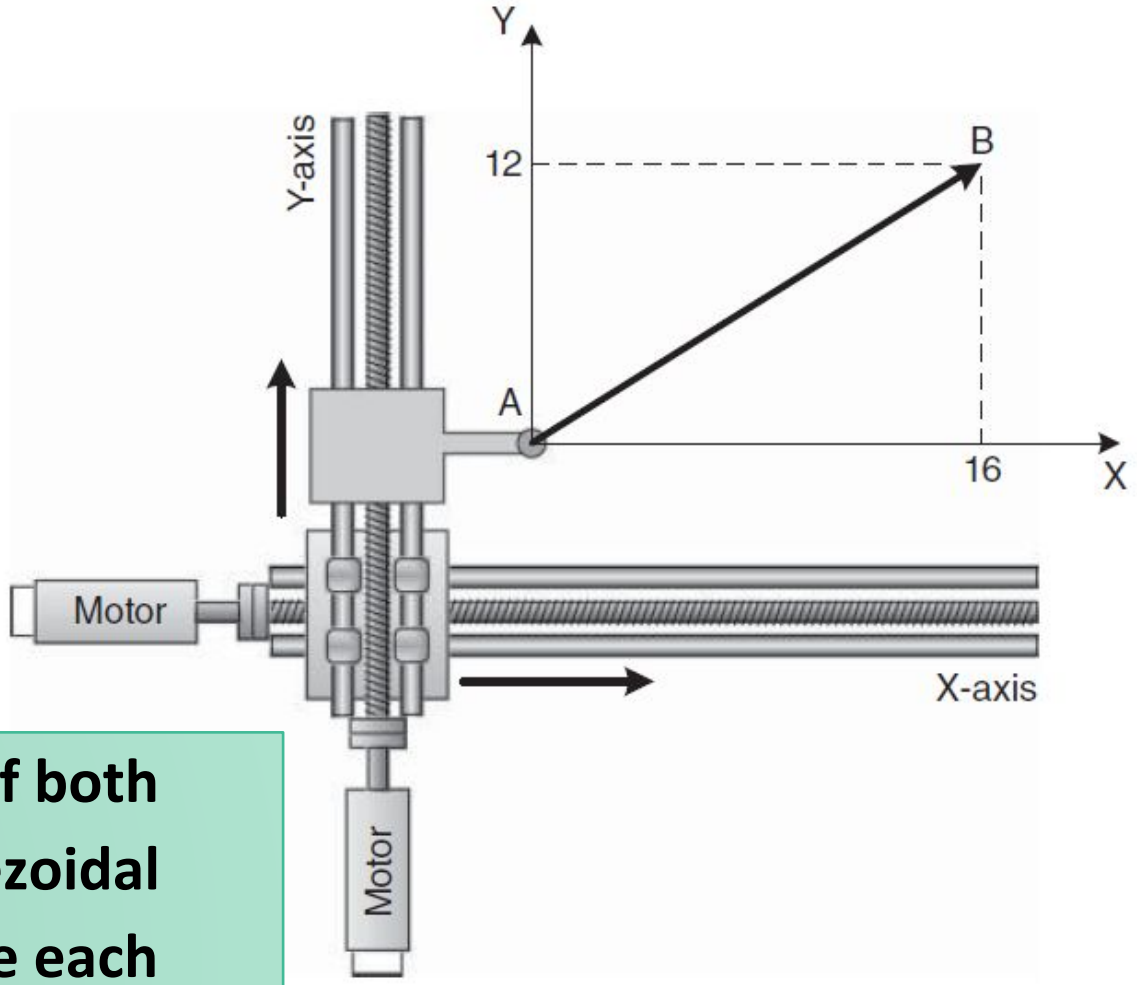
Current loop

Desired Motion

Position feedback

Multi Axes Motion – Slew Motion

- All axes start moving with the same speed and at the same time, but each axis finishes its motion at a different time.
- Each axis travels at different speeds but start at the same time
- Usually results in unnecessary wear on the axis and often leads to unanticipated results in the path taken by the system.



Consider the machine shown in the above figure. If both axes are moving at the speed of 4 in/s using trapezoidal velocity profile with ta = 0.2 s, how long will it take each axis to complete its move? Will the tool tip follow a straight line as shown in figure?

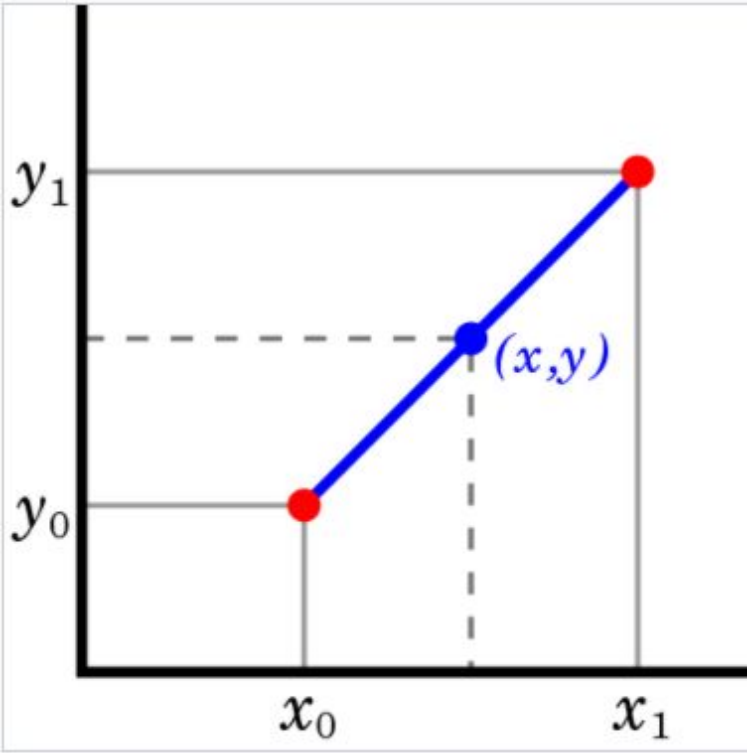
Multi Axes Motion – Interpolated Motion

- All joints start, and stop at the same time
- Faster axis are slowed down by the controller such that all axis finish at the same time.
- All the axes are coordinated by the controller
- Two approaches to achieve motions are
  - ✔ **Slow down the faster axes while keeping the acceleration time, t_a, the same as the axis that takes the longest time to complete its motion**
  - ✔ **Slow down the faster axes while keeping the acceleration, a, the same as the axis that takes the longest time to complete its motion.**
- To make the tool tip to follow a straight line in the previous example, the controller will adjust the velocity of the Y-axis to 3 in/s making both the axis have similar time to complete its motion

$$u_y = \frac{L}{(t_m + t_a)} = \frac{12}{2.8 + 0.2} = 3\ in/s$$

Multi Axes Motion – Interpolated Motion

- Types of Interpolation are
  - Linear Interpolation
  - Circular Interpolation – move consisting of a straight-line chord segments
    - Translating the linear axis positions into the curved tool motions
    - $\boldsymbol{R = \sqrt{(x^2+y^2)}}$
  - Contouring - a series of points is provided during programming, and the **motion controller** extrapolates a smooth line from these points.



$$y = y_0 + (x - x_0)\frac{y_1 - y_0}{x_1 - x_0} = \frac{y_0(x_1 - x) + y_1(x - x_0)}{x_1 - x_0}$$

Motion (Move) Modes

Motion Controller:

- Calculates and controls the mechanical trajectories an actuator must follow
- Ensure the axis position and **move** between two points
  - Linear
  - Circular
- Move Constraints are
  - Position / distance
  - Velocity Limits
  - Acceleration / Deceleration
- Move constraints are used to create the trajectory profile by the controller

Motion (Move) Modes

Linear Move:

- Rectilinear motion, is one-dimensional motion along a straight line
- Start and End points along with constraints are necessary
- Target positions are specified in two ways
  - Absolute – Axis starts at origin as reference
  - Incremental – Axis starts with the previous position as reference

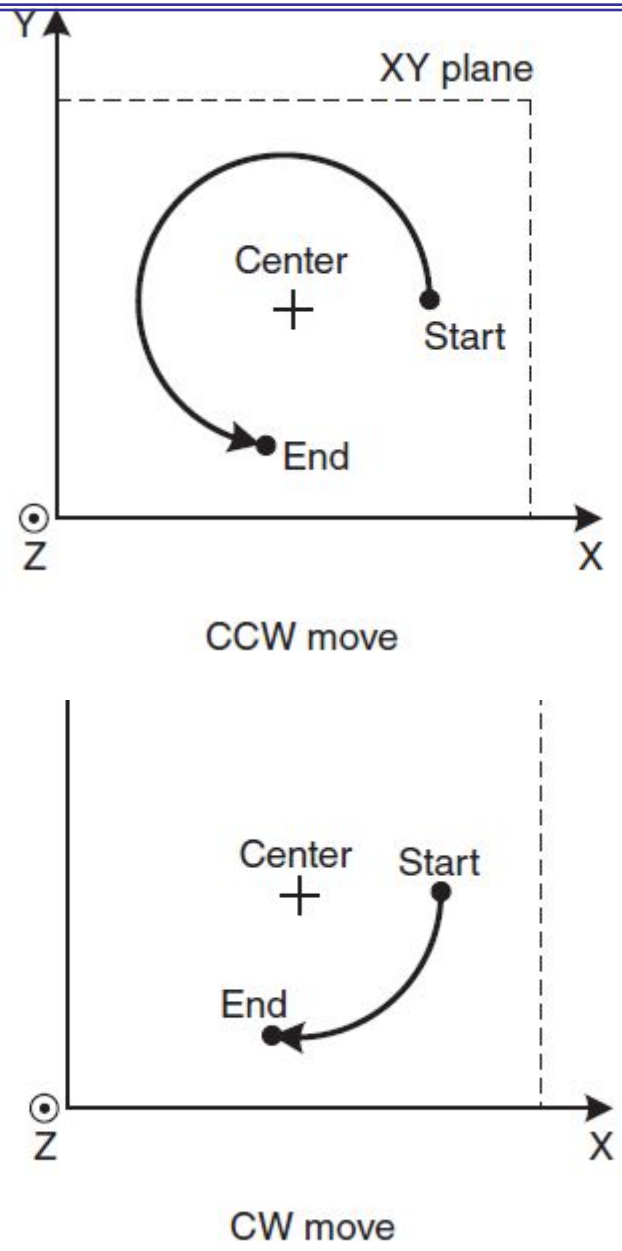
- $v_{avg} = \frac{\Delta x}{\Delta t} = \left(\frac{X_2 - X_1}{t_2 - t_1}\right)$
- Example – given trapezoidal velocity profile with move velocity (v_m),
- Controller computes the move time (t_m) and acceleration and deceleration time (t_d)

Motion (Move) Modes

Circular Moves

- Circular path interms of arc is followed between two points A and B
- 2 D as an arc or 3D as an helix
- Requires multiple axis to be coordinated
  - Start Point
  - End Point
  - Center point of the arc
  - Direction of the move I.e. CW or CCW

$$R = \sqrt{(x^2+y^2)}$$

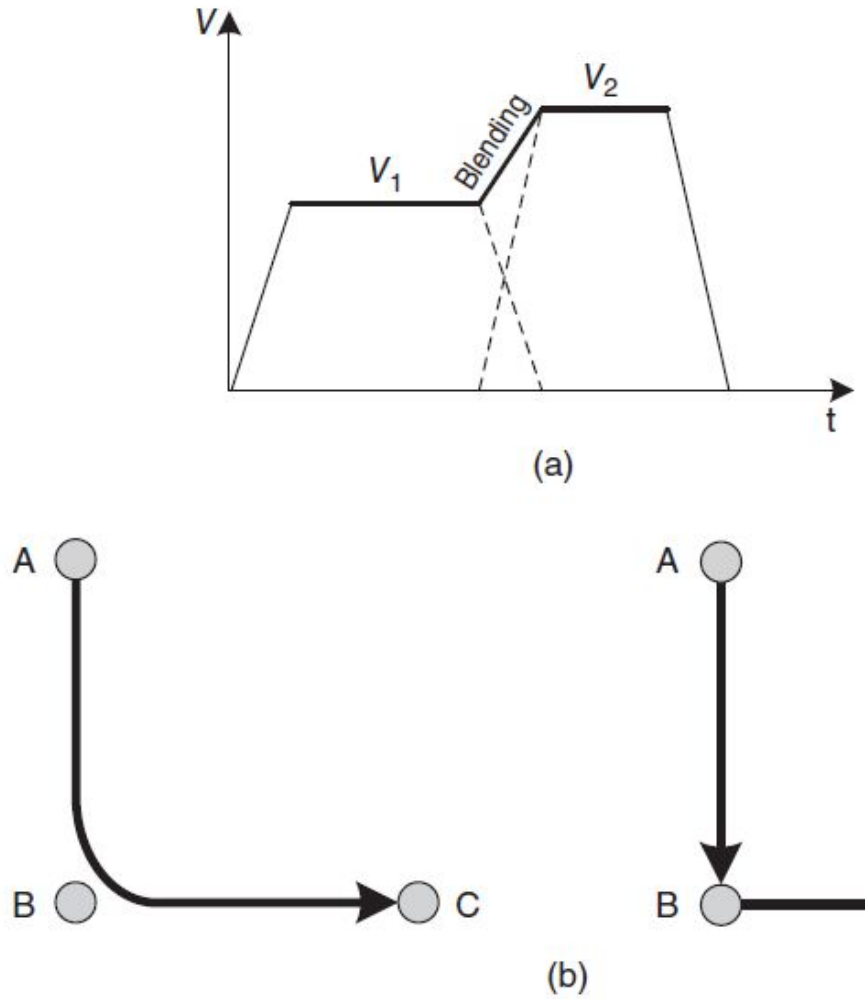


Circular moves in the XY plane

Motion (Move) Modes

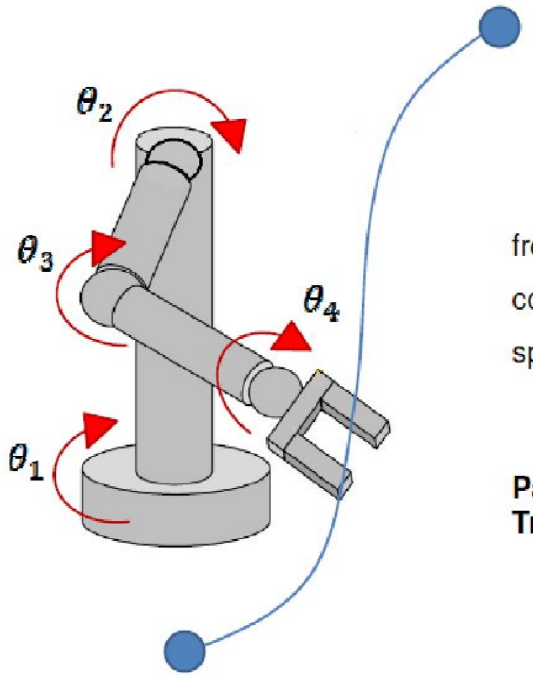
Contour Moves

- Mix of various moves and must when a trajectory cannot be construction with straight line or an arc
- Sequence of points are connected by means of splines – cubic splines to generate trajectory
- Smooth motion is ensured by means of blending between two linear, or two circular or between a linear and circular segments
- Smooth transition from one velocity to another by blending the velocity of the first move to the velocity of the next move
- End position of each segment may or may not be reached.



Blended moves. (a) Blending of velocities between two consecutive moves. (b) Two linear moves (Left) with and (Right) without blending

Path & Trajectory planning

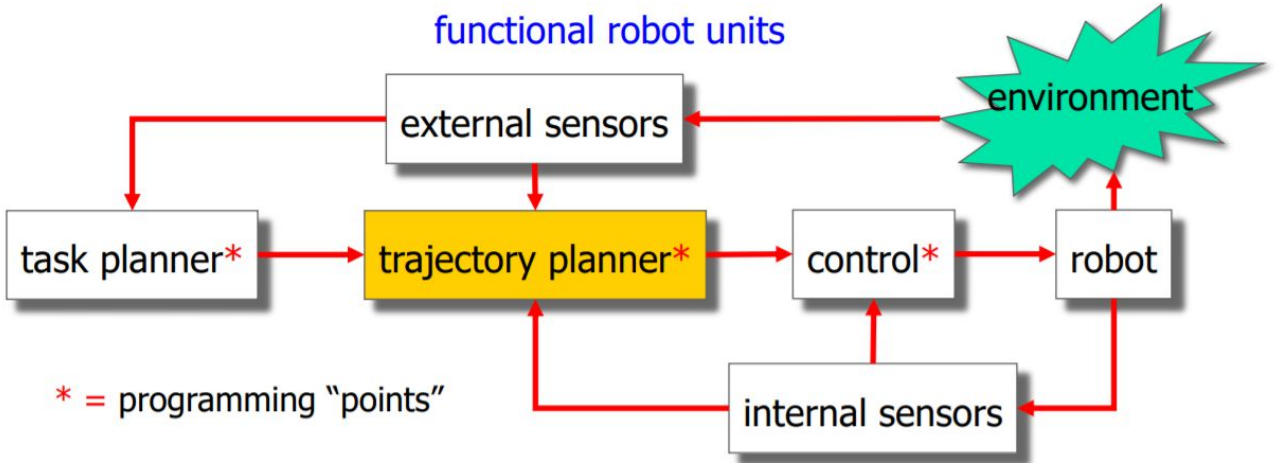


freedc
compl
space

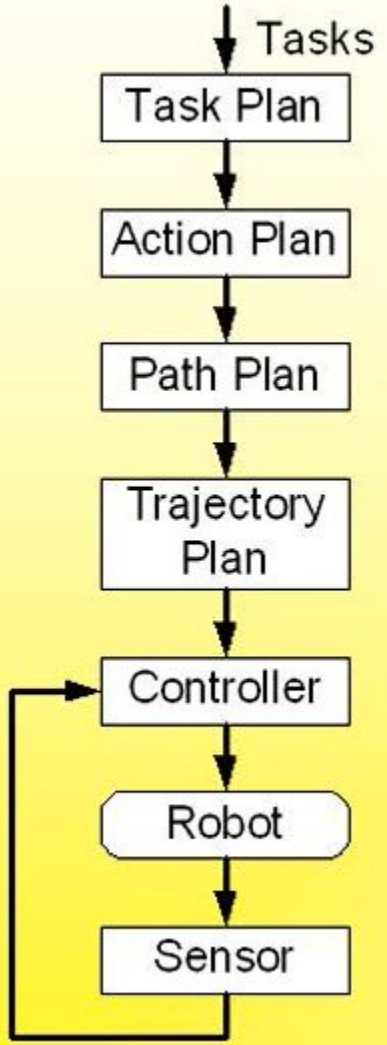
Path:
Traje

Manipulators with multi degree of freedom for accomplishing various complex manipulation in the work space

- Path: Only Geometric Description
- Trajectory: Timing Included



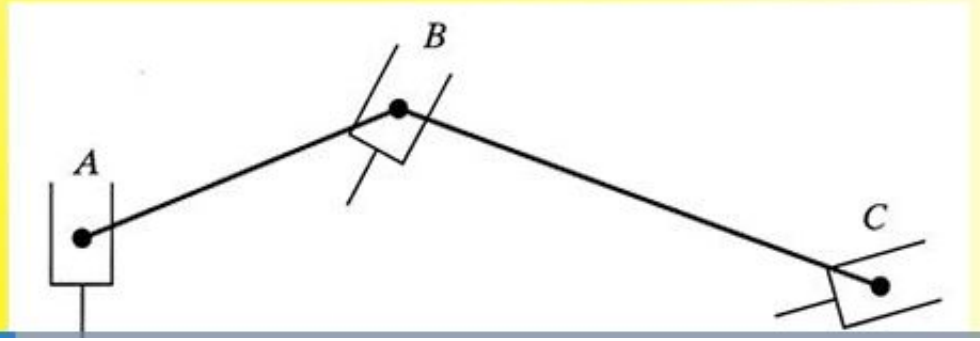
Path Planning and Trajectory Tracking



- Path planning
  - Geometric path
  - Issues: obstacle avoidance, shortest path
- Trajectory planning,
  - "interpolate" or "approximate" the desired path by a class of polynomial functions
  - Generate a sequence of time-based "control set points" for the control of manipulator from the initial configuration to its destination.

Path vs Trajectory

- Path: A sequence of robot configurations in a particular order without regard to the timing of these configurations.
- Trajectory: It concerned about when each part of the path must be attained, thus specifying timing.



Sequential robot movements in a path

Path Planning Problem Definition

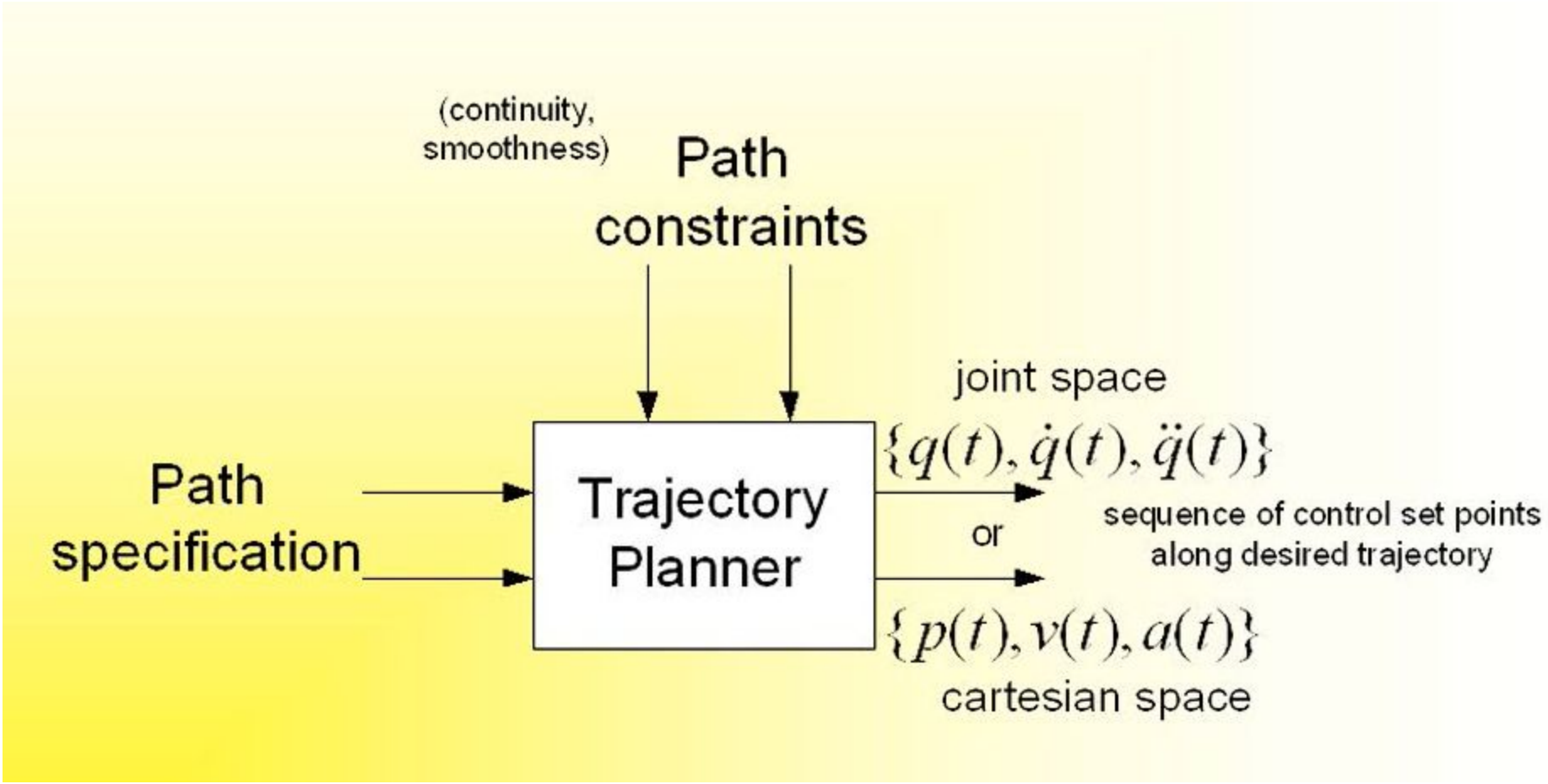
- Problem statement: Compute a collision-free path for a rigid or articulated moving object among static obstacles.
- Input
  - Geometry of a moving object (a robot, a digital actor, or a molecule) and obstacles
  - How does the robot move?
  - Kinematics of the robot (degrees of freedom)
  - Initial and goal robot configurations (positions & orientations)
- Output

  Continuous sequence of collision-free robot configurations connecting the initial and goal configurations

Trajectory Planning Problem Definition

- Problem statement Turn a specified Cartesian-space trajectory of P_e into appropriate joint position reference values
- Input
  - Cartesian space path
  - Path constraints including velocity and acceleration limits and singularity analysis.
- Output
  - a series of joint position/velocity reference values to send to the controller

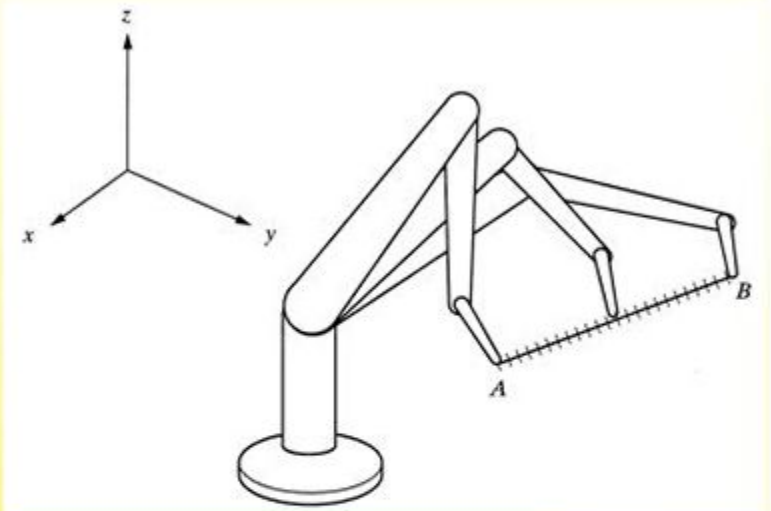
Typical Trajectory Planner



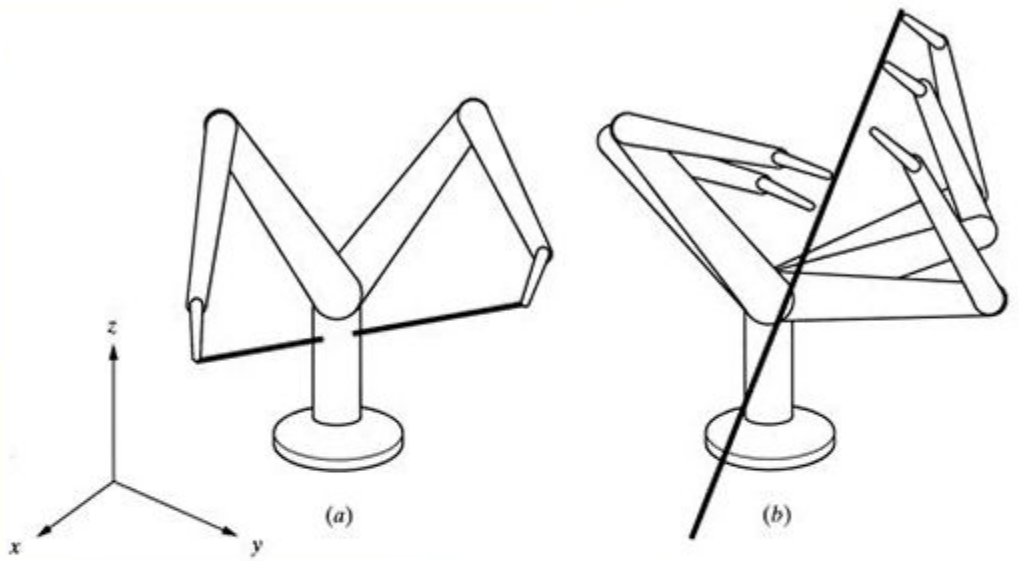
Joint Space vs Task space

- Joint-space description:
  - The description of the motion to be made by the robot by its joint values.
  - The motion between the two points in joint space is not predicted.
- Task space description:
  - The motion between the two points is known at all times and controllable.
  - It is easy to visualize the trajectory, but is difficult to ensure singularity robustness.

Joint Space vs Task space

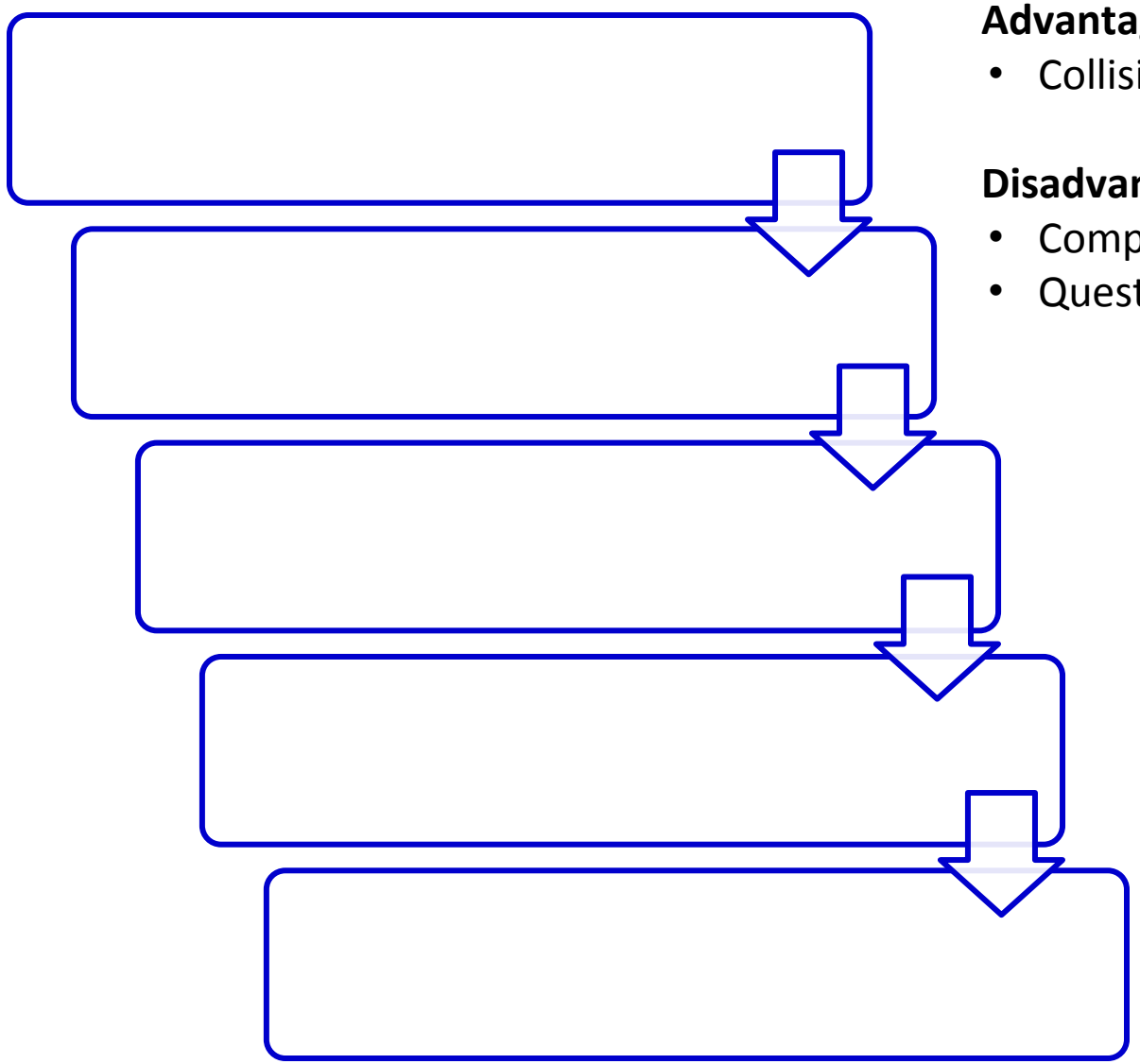


Sequential motions of a robot to follow a straight line.



Cartesian-space trajectory (a) The trajectory specified in Cartesian coordinates may force the robot to run into itself, and (b) the trajectory may requires a sudden change in the joint angles.

Planning in Task space



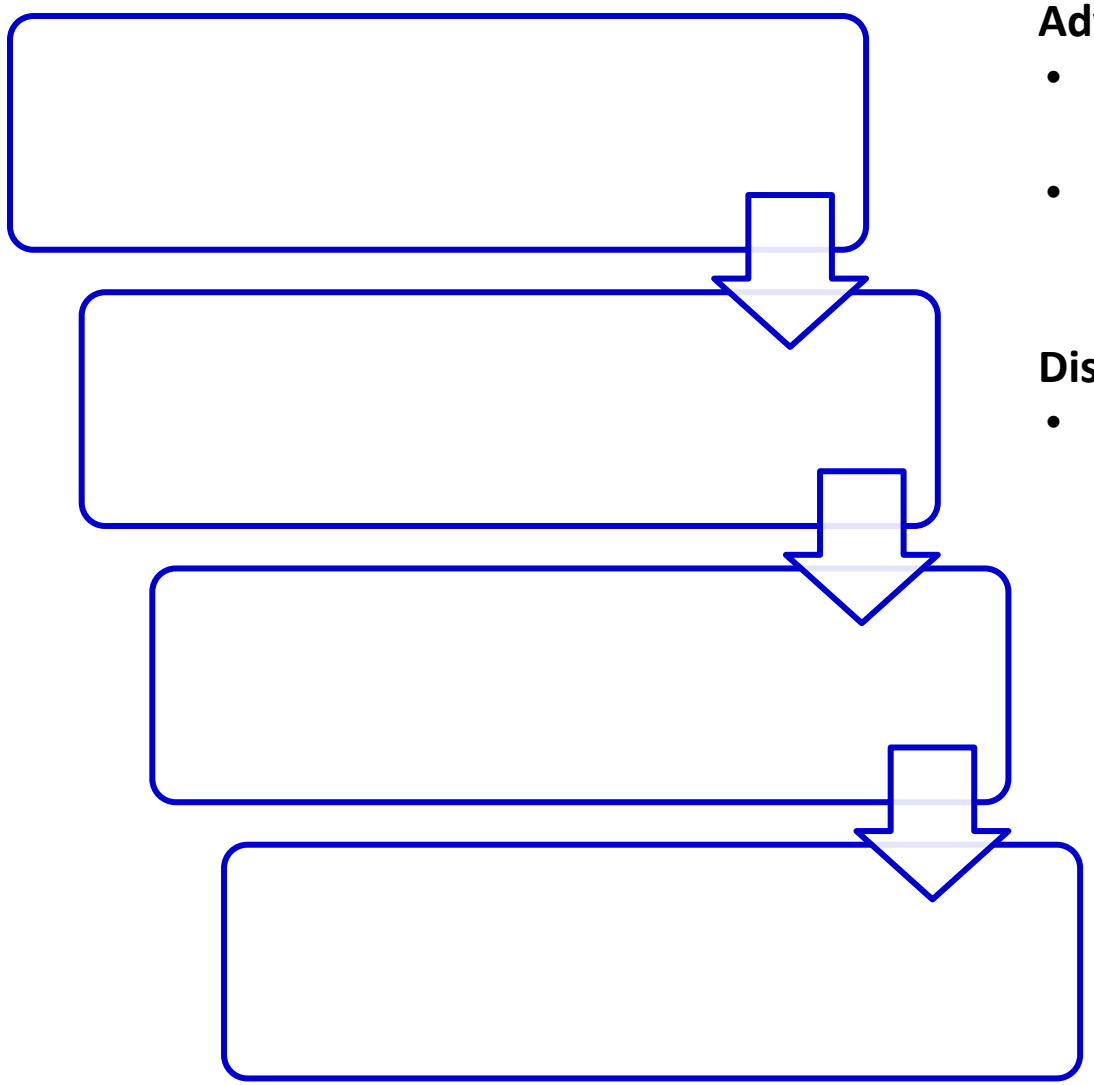
Advantages:

- Collision free path

Disadvantages:

- Computationally inexpensive
- Quest is to set the total time Tpath

Planning in Joint Space



Advantages:

- Inverse Kinematics is computed only once
- Can easily take into account joint angle and velocity constraints

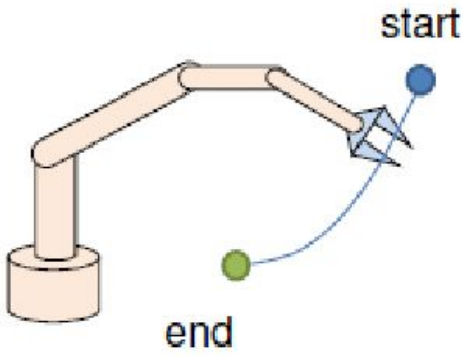
Disadvantages:

- Cannot deal with operational space obstacles

Types of Motion

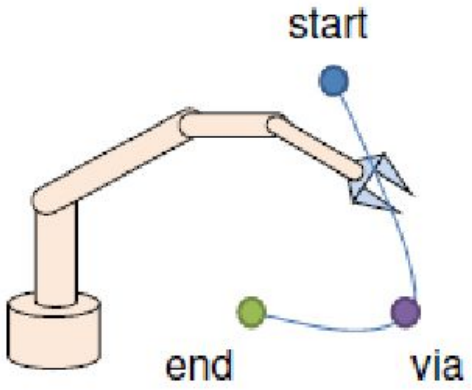
1. Point to point motion:

- End effector moves from a ***start point*** to ***end point*** in work space
- All joints' movements are coordinated for the point-to-point motion
- End effector travels in an arbitrary path



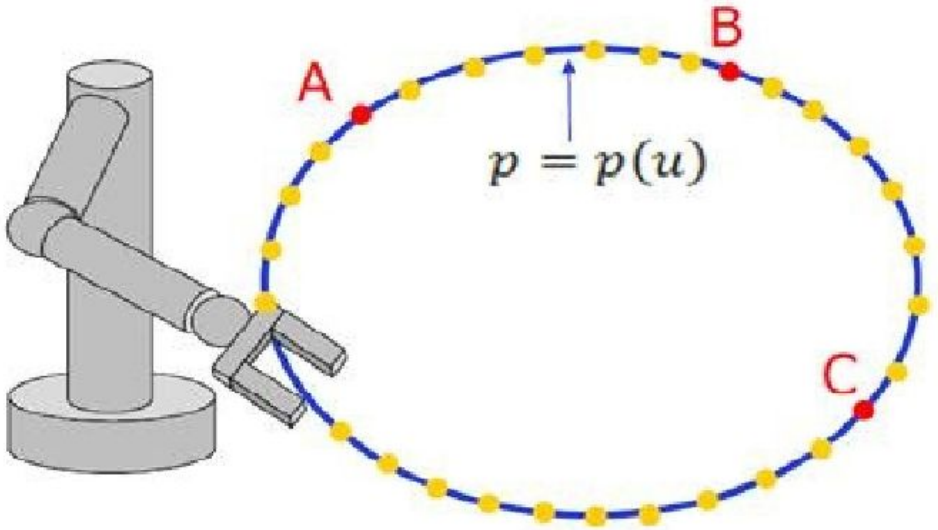
2. Motion with Via Points

- End effector moves through an intermediate point between start and end
- End effector moves through a via point without stopping



Path Definition

"Expressing the desired positions of a manipulator in the space, as a parametric function of time"



- B. Siciliano et al. Robotics (Modelling, planning and control), Springer, Berlin, 2009, chapter 4: Trajectory planning, pages 161-189.

Point to Point motion

- Simple P-T-P smooth trajectory with a few constraints on the lift- off and set-down positions
- 3 Ways a manipulator can move from P-T-P

  - Slew Motion:

    - all Joints move to their required new position as quickly as possible

    - all axes begin motion at the same time but arrive at their destination at different times.

  - Joint Interpolated motion:

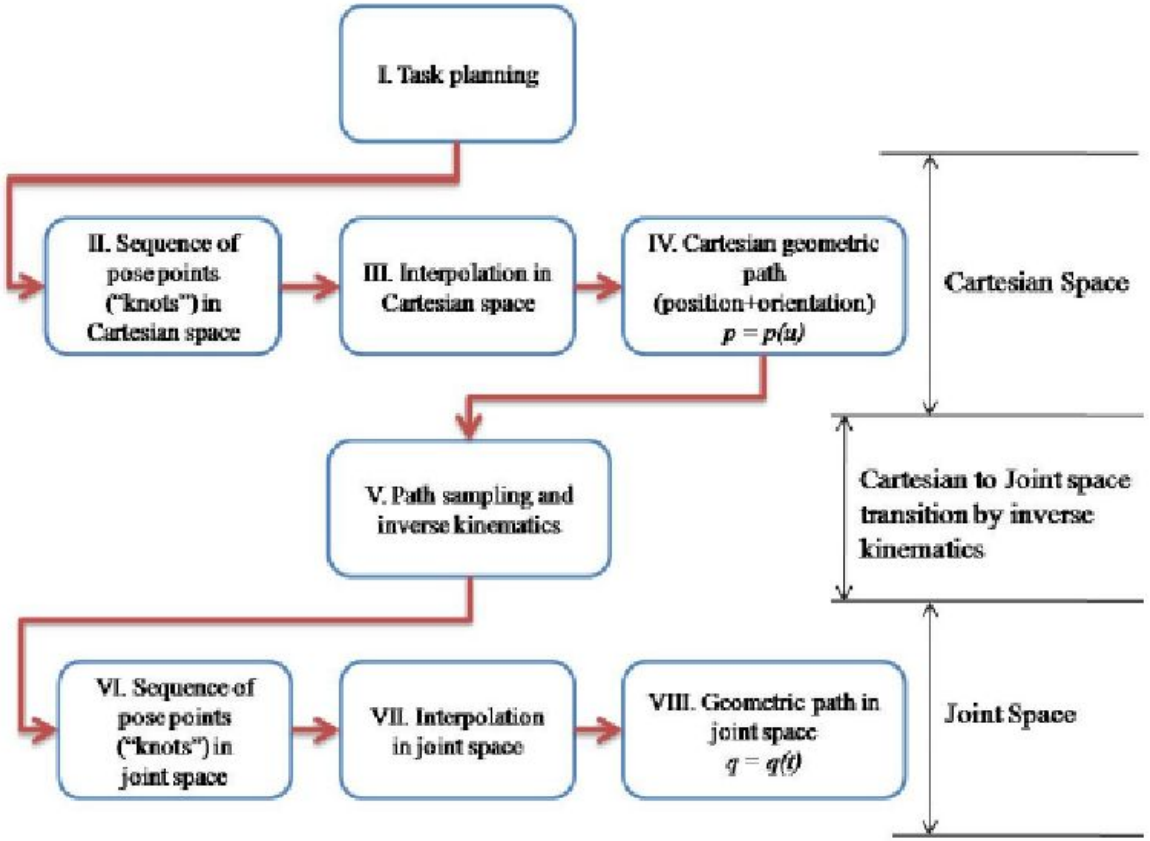
    - requires controller to calculate which joint will take the longest to arrive at its destination and slow other joints down accordingly

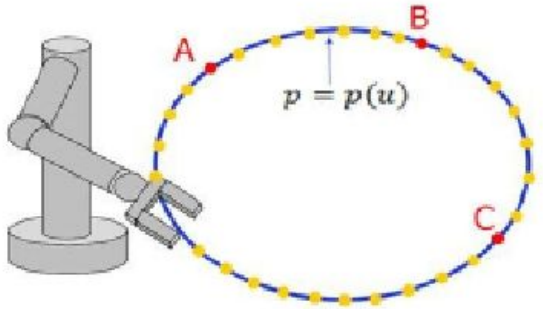
    - Separate velocity is calculated for each axis

    - motion is generally smooth

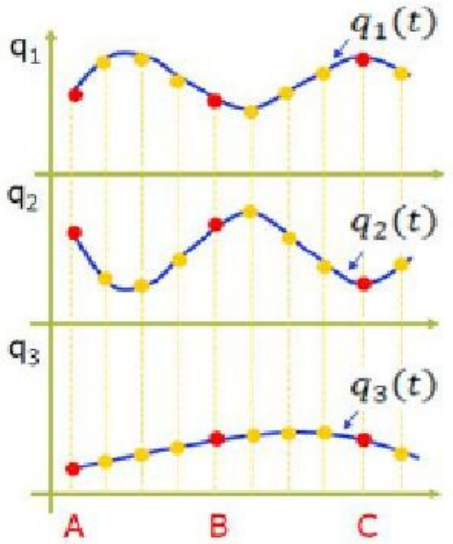
  - Continuous path motion: Straight line motion

Task to Trajectory





Cartesian space

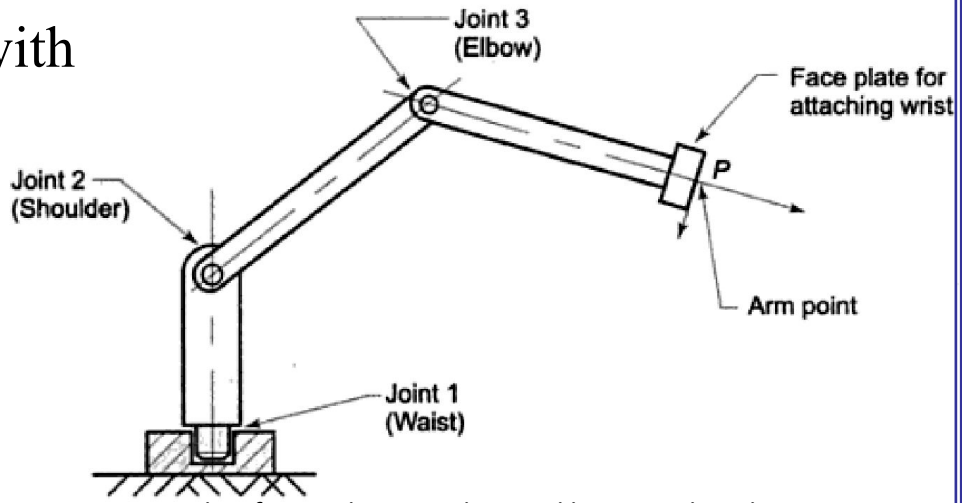


Joint space

Trajectory Planning

Generation of time sequence of position, velocity and acceleration in terms of joint or Cartesian coordinates is called trajectory planning. When the time sequence is in terms of joint coordinates it is called joint space trajectory planning and when it is in terms of Cartesian Coordinates it is called Cartesian space trajectory planning.

Path planning vs Trajectory planning: Trajectory planning is with time history.



Picture taken from Robotics and Control by Nagrath and Mittal

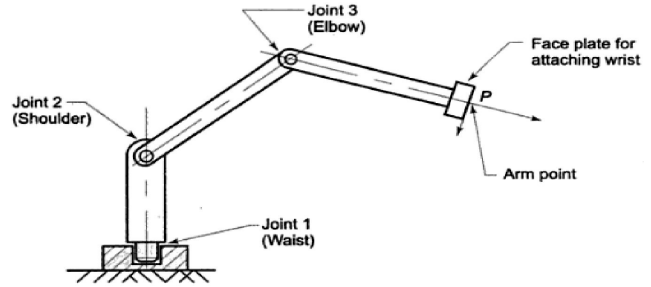
Joint Space Trajectory Planning:

Given for a task: Beginning pose, End pose and time required:

- Do IK at beginning pose, determine the initial joint angles
- Do IK at the end-pose, determine the final joint angles
- Interpolate the intermediate joint angles with time history.

Example:

The base joint of a 3-DOF articulated robot has an initial joint angle of 30º and a final joint angle of 120º for a specific task which takes 10 sec. Perform Joint space trajectory planning.



Picture taken from Robotics and Control by Nagrath and Mittal

t=0 sec	t=10 sec
30º	120º

If we give these two commands using an Arduino, it will be executed immediately. We cannot ensure the time duration. **How to ensure time duration? Answer is interpolation.**

A possible solution

t=0 sec	t=1s	t=2s	t=3s	t=4s	t=5s	t=6s	t=7s	t=8s	t=9s	t=10 sec
30º	39º	48º	57º	66º	75º	84º	93º	102º	111º	120º

This is called a linear interpolation. If joint displacement is linear, speed will be constant and zero acceleration. This is not possible. For quadratic joint displacement, speed will be linear and acceleration is constant. A preferred interpolation technique **is cubic spline trajectory**.

Joint space trajectory planning is used for point-to-point tasks. Ex: Pick and place, peg in a hole etc

Cartesian space path planning is used to continuous tasks. Ex: welding, painting etc

Cubic Spline Trajectory Planning:

A 1 DOF manipulator with rotary joint is to move from 113° to 210° in 7 sec. Find the coefficients of the cubic polynomial to interpolate a smooth trajectory. Plot the position, velocity and acceleration variation as a function of time.

Solution:

$q(t)=C_o+C_1t+c_2t^2+c_3t^3$ .....(1)

Given q(0)=113°, q(7)=210°. For a smooth trajectory $\dot{q}(0) = 0$ and $\dot{q}(7) = 0$.

Application of the constraints:

Applying q(0)=0, we get C_o=113;

Applying q(7)=210, we get $210=113+7C_1+49C_2+343C_3$ .....(2)

Applying $\dot{q}(0) = 0$: $\dot{q}(t) = C_1 + 2 * C_2t + 3C_3t^2 \Rightarrow 0 = C_1$ .....(3)

Applying $\dot{q}(7) = 0$: $0 = 14C_2 + 147C_3$ .....(4)

Eqn.(2) can be rewritten as $49C_2 + 343C_3 = 97$ .....(5)

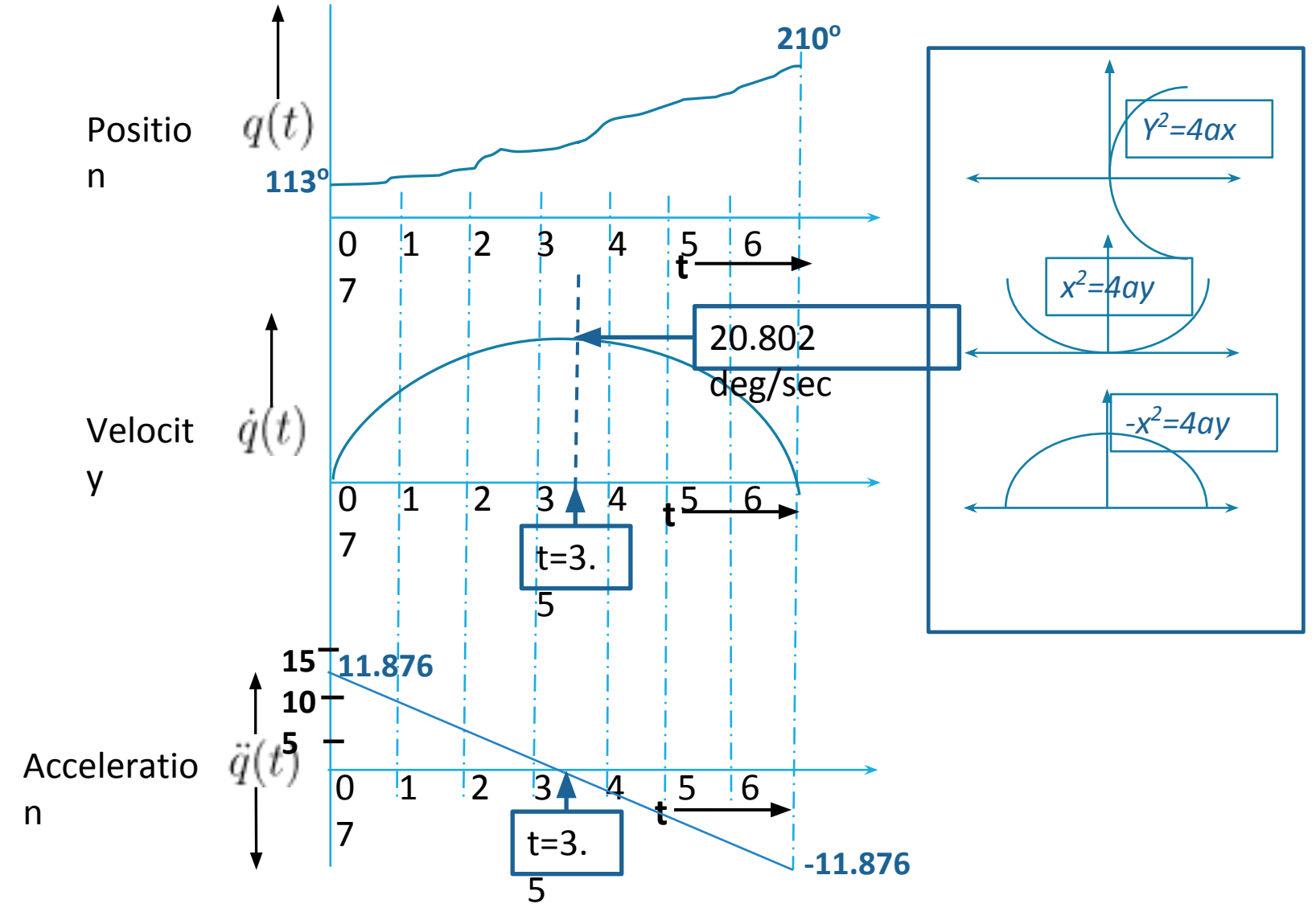
Solving Eqn. 4 and 5, we get C_2=*5.938* and C_3=-*.565*. So Eqn.(1) is $q(t)=113+5.938\ t^2-.565\ t^3$.

$$q(t) = 113 + 5.938t^2 - .565t^3$$
$$\dot{q}(t) = 11.876t - 1.695t^2$$
$$\ddot{q}(t) = 11.876 - 3.39t$$

$$\ddot{q}(t) = 11.876 - 3.39t$$
at t=0, $\ddot{q}(t) = 11.876$
at t=7, $\ddot{q}(t) = -11.876$

11.876-3.39t=
0
t=3.5

$$\dot{q}(t) = 11.876t - 1.695t^2$$
at t=3.5, $\dot{q}(t) = 20.802\ deg/sec$



Cubic Spline Trajectory Planning: - Example

1. Using the third order cubic polynomial, calculate the first joint angle of a 6-axis robot at 1, 2, 3, and 4 seconds.
   a. By assuming the robot first joint reaches to an angle of 75° in 5 seconds from its initial angle of 30°.
   b. Plot the position, velocity and acceleration curves for the motion.

Answer: $\theta_1 = 35°$, $\theta_2 = 46°$, $\theta_3 = 59°$, $\theta_4 = 71°$,

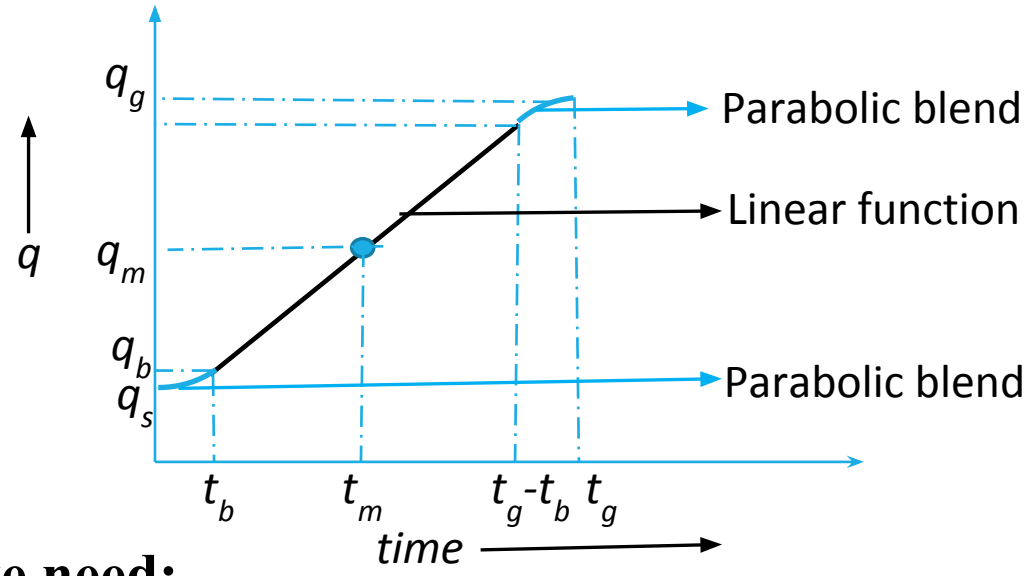
16. Determine the time history of the position, velocity, and acceleration of the end-effector of a manipulator arm which moves from start to goal point via 2 intermediate points. The desired end-effector motion is given in table below. Assume a cubic spline trajectory for each segment and velocity continuity is enforced at the via points.

End-effector	Path point (j)			
	1	2	3	4
Position (deg)	0	270	-90	360
Velocity (deg/s)	0	-90	180	0
Traversal time (s)	0	3	5	2

Linear Function with Parabolic Blends

Given:

- Initial joint displacement (q_s)
- Final joint displacement (q_g)
- Total time (t_g)
- Constant blend acceleration ($\ddot{q}_c$)



To fit a linear function and two parabolic blends, what we need:

- Blending time (t_b)
- Velocity at the end of first blend (or at the beginning of the 2nd blend) must be equal to the velocity of linear segment

Determine position, velocity and acceleration profile

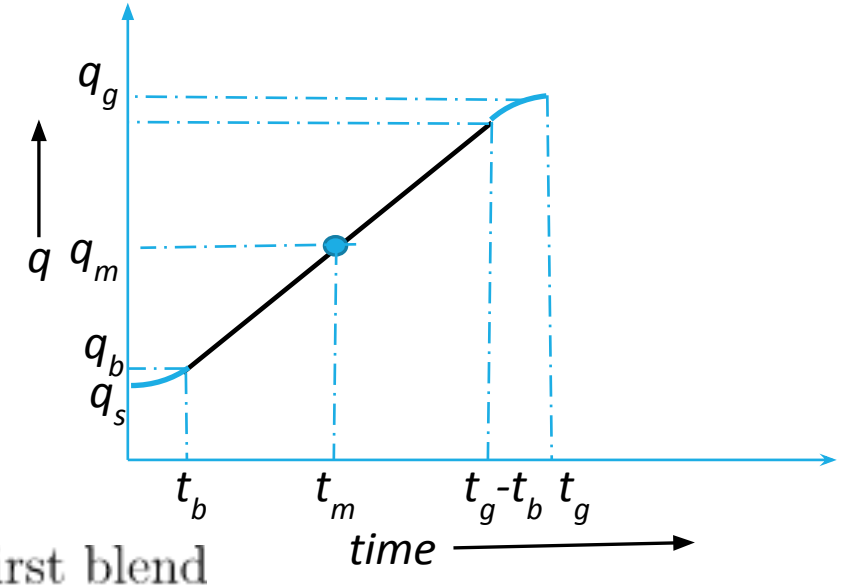
As can be seen, $q_m = \frac{q_s + q_g}{2}$ and $t_m = t_g/2$.....(1)

Velocity at the end of first blend: $\ddot{q}_c t_b$ (v=u+at; u=0; a=$\ddot{q}_c$)

Velocity of the linear segment: $\frac{q_m - q_b}{t_m - t_b}$

Since velocity at the end of first blend is same as that of the linear segment,

$\ddot{q}_c t_b = \frac{q_m - q_b}{t_m - t_b}$.....(2)



For constant acceleration $\ddot{q}_c$, the joint displacement at the end of first blend

$q_b = q_s + \frac{1}{2}\ddot{q}_c t_b^2$.....(3)

$\Rightarrow q_m + \ddot{q}_c t_b(t_m - t_b) = q_s + \frac{1}{2}\ddot{q}_c t_b^2$

$\Rightarrow \frac{(q_s + q_g)}{2} + \ddot{q}_c t_b(t_g/2 - t_b) = q_s + \frac{1}{2}\ddot{q}_c t_b^2$

$\Rightarrow \ddot{q}_c t_b^2 - \ddot{q}_c t_g t_b + (q_g - q_s) = 0$.....(4)

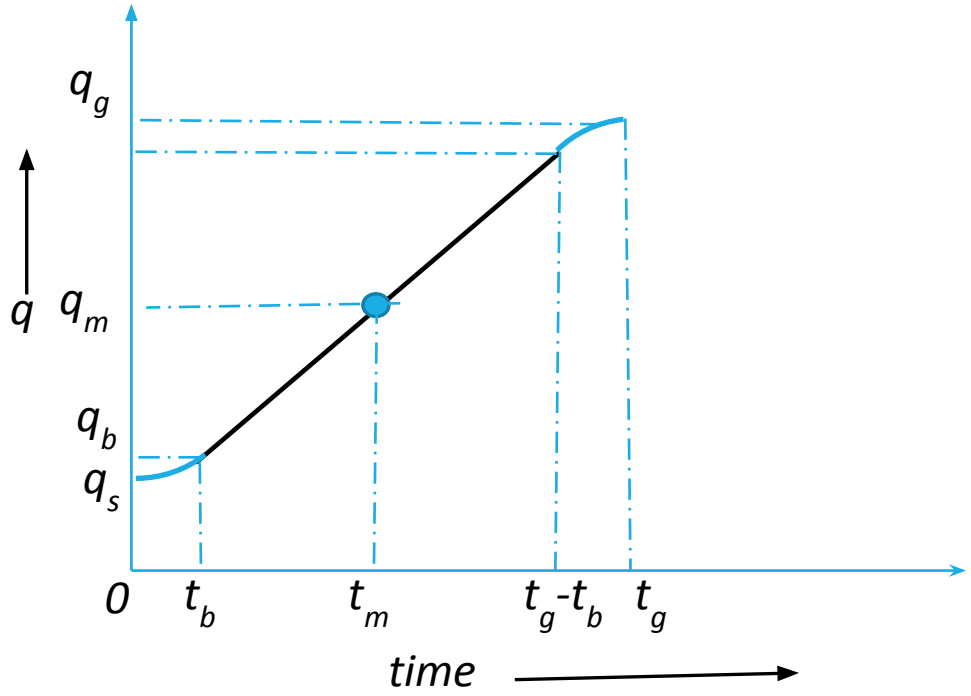
So, $t_b = \frac{t_g}{2} - \frac{1}{2}\sqrt{\frac{t_g^2\ddot{q}_c - 4(q_g - q_s)}{\ddot{q}_c}}$ and it should satisfy a consraint

$t_g^2\ddot{q}_c - 4(q_g - q_s) \geq 0 \Rightarrow |\ddot{q}_c| \geq \frac{4|q_g - q_s|}{t_g^2}$.....(5)

$$q(t) = \begin{cases} q^s + \frac{1}{2}\ddot{q}^c t^2 & 0 \le t \le t_b \\ q^s - \frac{1}{2}\ddot{q}^c t_b^2 + \ddot{q}^c t_b t & t_b < t \le t_g - t_b \\ q^g - \frac{1}{2}\ddot{q}^c (t_g \quad t)^2 & t_g - t_b < t \le t_g \end{cases}$$

$$\dot{q}(t) = \begin{cases} \ddot{q}^c t & 0 \le t \le t_b \\ \ddot{q}^c t_b & t_b \quad \le t_g - t_b \\ \ddot{q}^c (t_g - t) & t_g - t_b < t \end{cases}$$

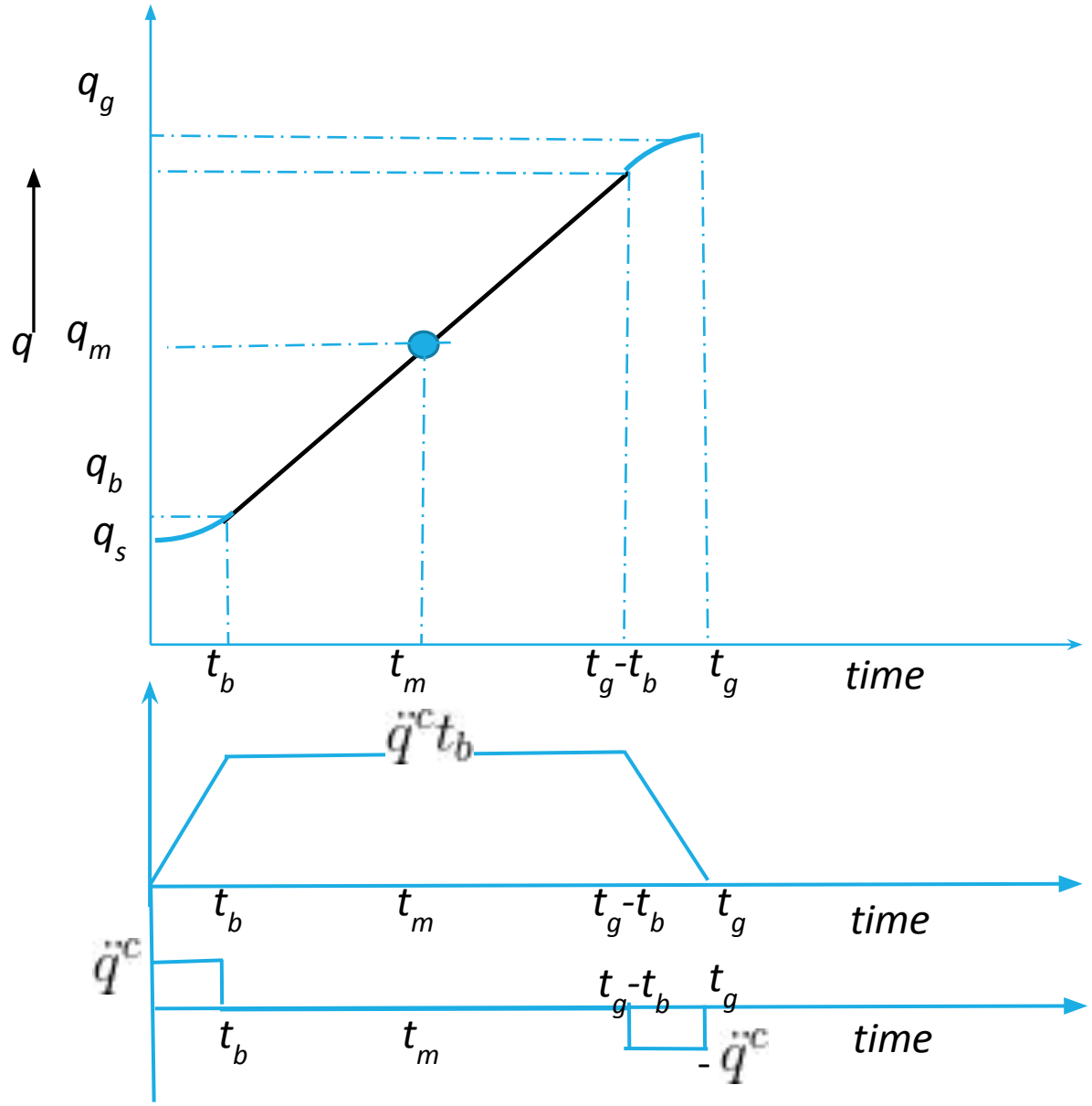
$$\ddot{q}(t) = \begin{cases} \ddot{q}^c & 0 \le t \le t_b \\ 0 & t_b < t \le t_g - t_b \\ -\ddot{q}^c & t_g - t_b < t \le t_g \end{cases}$$



$$q_s + \frac{1}{2}\ddot{q}^c t_b^2 + \ddot{q}^c t_b (t - t_b)$$

$$q_s + \frac{1}{2}\ddot{q}^c t_b^2 + \ddot{q}^c t_b t - \ddot{q}^c t_b^2$$

$$q_s - \frac{1}{2}\ddot{q}^c t_b^2 + \ddot{q}^c t_b t$$


q_g
q
q_m
q_b
q_s
t_b
t_m
t_g-t_b
t_g
time
$\ddot{q}^c t_b$
t_b
t_m
t_g-t_b
t_g
time
$\ddot{q}^c$
t_b
t_m
t_g-t_b
t_g
$-\ddot{q}^c$
time

A one DOF manipulator with rotary joint is to move from 113° to 210° in 7 seconds. Assuming that a linear trajectory with parabolic blends is used and a constant acceleration of 30°/sec^2, determine the parameters for the trajectory and plot displacement, velocity and acceleration of the joint.

Given, $q_s = 113^o$, $q_g = 210^o$, $t_g = 7$, $\ddot{q}^c = 30^o/sec^2$

The condition to be satisfied to have a linear function with parabolic blend is that

$|\ddot{q}^c| \geq \frac{4\,|q_g - q_s|}{t_g^2}$. Here, $30 \geq \frac{4\,|210 - 113|}{7^2} = 1.97$. So, in this case, this trajectory can be followed.

$$\text{So, } t_b = \frac{t_g}{2} - \frac{1}{2}\sqrt{\frac{t_g^2\ddot{q}_c - 4(q_g - q_s)}{\ddot{q}_c}}$$

$$= 7/2 - \frac{1}{2}\sqrt{\frac{7^2 * 30 - 4(210 - 113)}{30}}$$

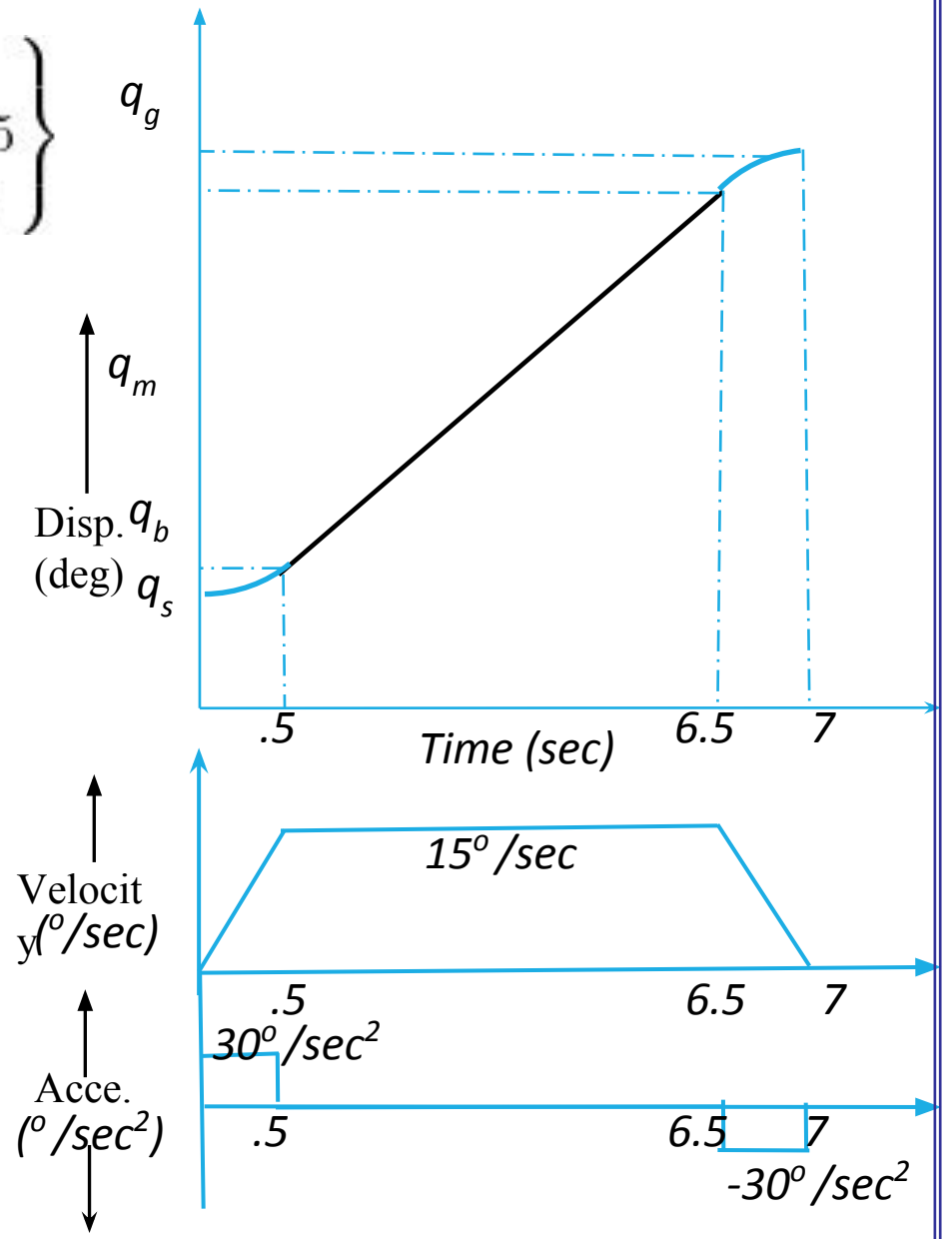
$$= .5sec$$

Now with this information of parameters, we can write the equations for displacement, velocity and acceleration and plot them also.

$$q(t) = \left\{ \begin{array}{c} 113 + \frac{1}{2}30t^2 = 113 + 15t^2 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 0 \le t \le .5 \\ 113 - \frac{1}{2} * 30 * .5^2 + 30 * .5 * t = 109.25 + 15t \ldots \ .5 \le t \le 6.5 \\ 210 - \frac{1}{2} * 30(7-t)^2 = 210 - 15 * 30(7-t)^2 \ldots \ 6.5 \le t \le 7 \end{array} \right\}$$

$$\dot{q}(t) = \left\{ \begin{array}{c} 30 * t \ldots\ldots\ldots\ldots 0 \le t \le .5 \\ 30 * .5 = 15 \ldots\ldots \ .5 \le t \le 6.5 \\ 30 * (7-t) \ldots \ 6.5 \le t \le 7 \end{array} \right\}$$

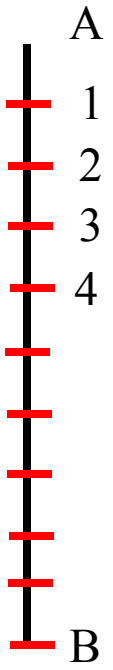
$$\ddot{q}(t) = \left\{ \begin{array}{c} 30 \ldots\ldots\ldots 0 \le t \le .5 \\ 0 \ldots\ldots \ .5 \le t \le 6.5 \\ -30 \ldots \ 6.5 \le t \le 7 \end{array} \right\}$$



Cartesian space trajectory planning:

Procedure:

Step 1: Do inverse kinematics at A; determine the joints angles
Step 2: Dispatch the joint displacement commands
Step 3: Do inverse kinematics at ‘1’, determine the joint angles
Step 4: Dispatch the motion commands
Repeat the procedure for all other sub-divisions



So Cartesian space path planning is more computationally tasking; online trajectory planning is difficult.

A straight-line path

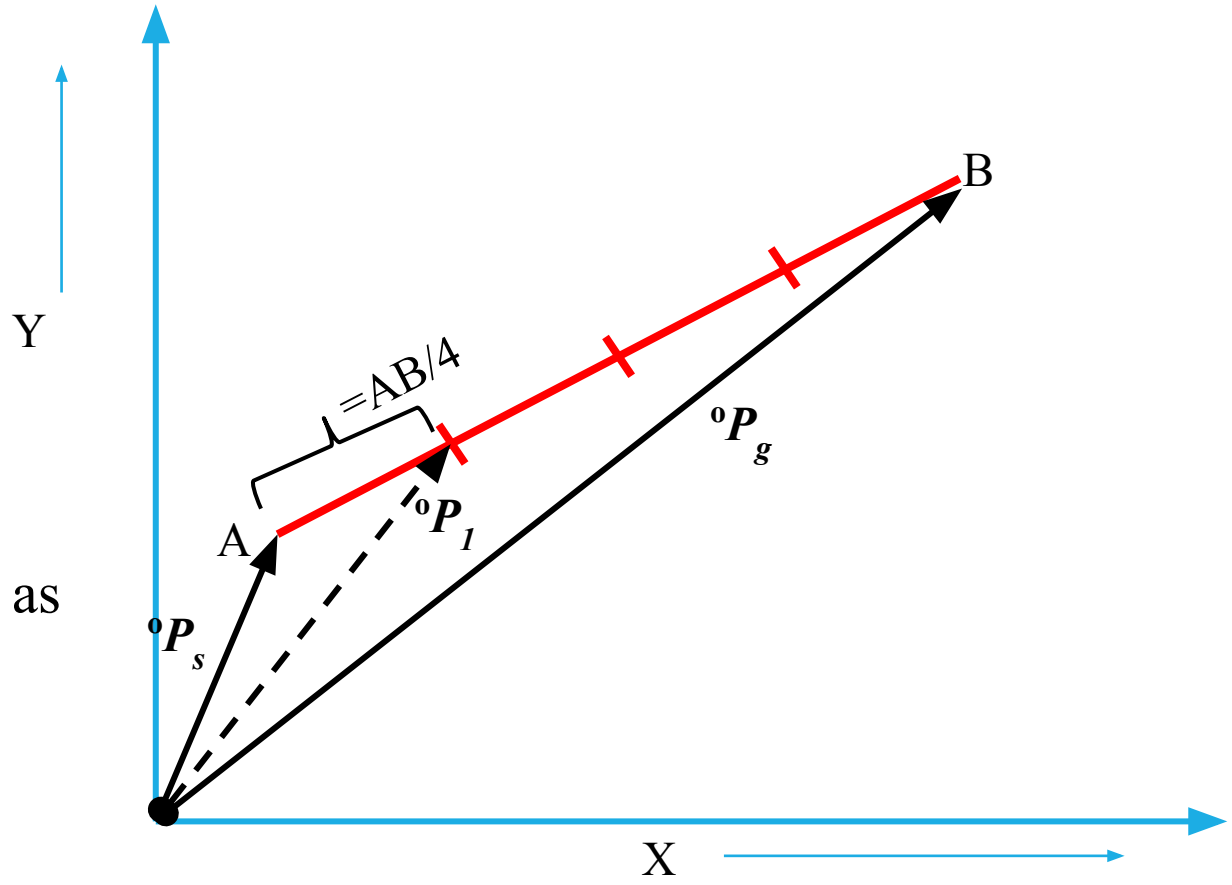
Given:

- Start point (${}^{o}P_{s}$)
- Goal point (${}^{o}P_{g}$)
- Path parameter (c)

Position Interpolation

$$ {}^{0}P_{1} = {}^{0}P_{s} + \frac{{}^{0}P_{g} - {}^{0}P_{s}}{\|{}^{0}P_{g} - {}^{0}P_{s}\|} * c $$

Here *c* is the path parameter; it can be as simple as *c=AB/4*

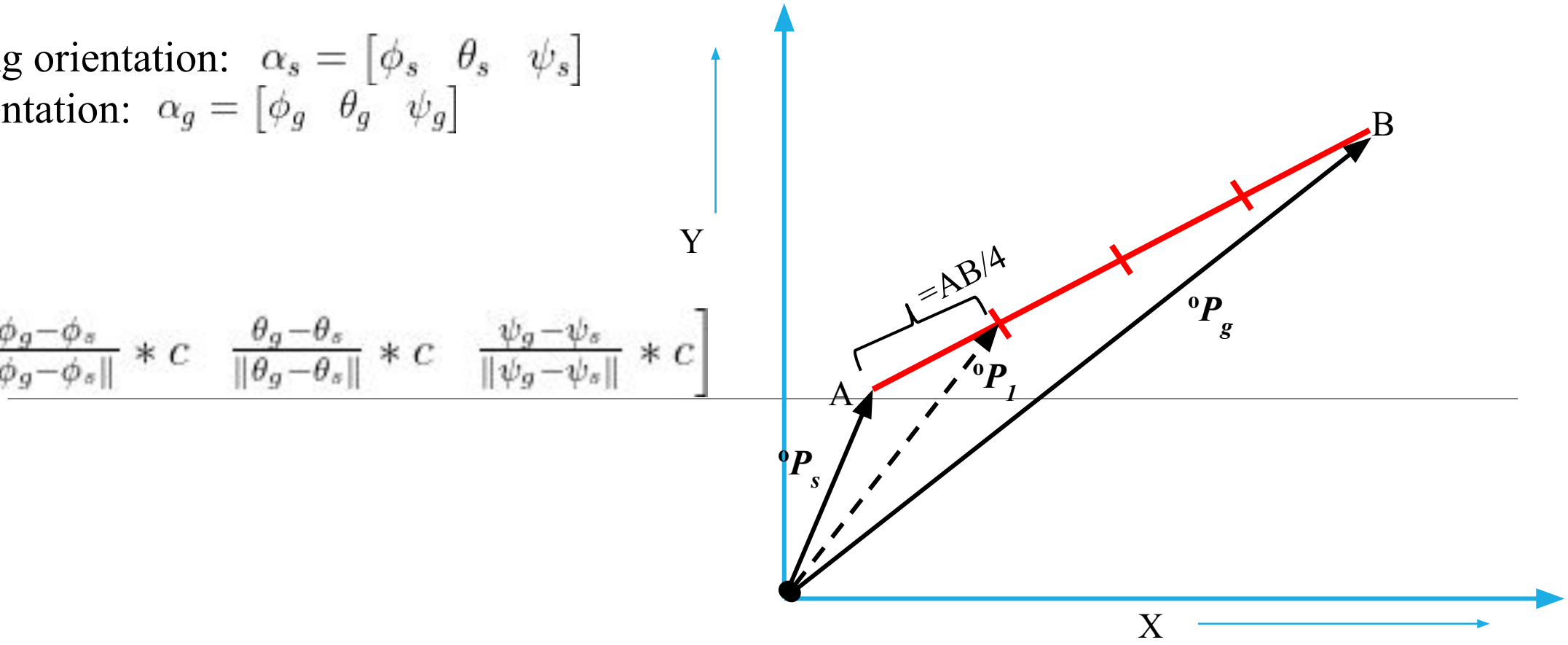


Orientation Interpolation

Beginning orientation: $\alpha_s = [\phi_s \quad \theta_s \quad \psi_s]$

End orientation: $\alpha_g = [\phi_g \quad \theta_g \quad \psi_g]$

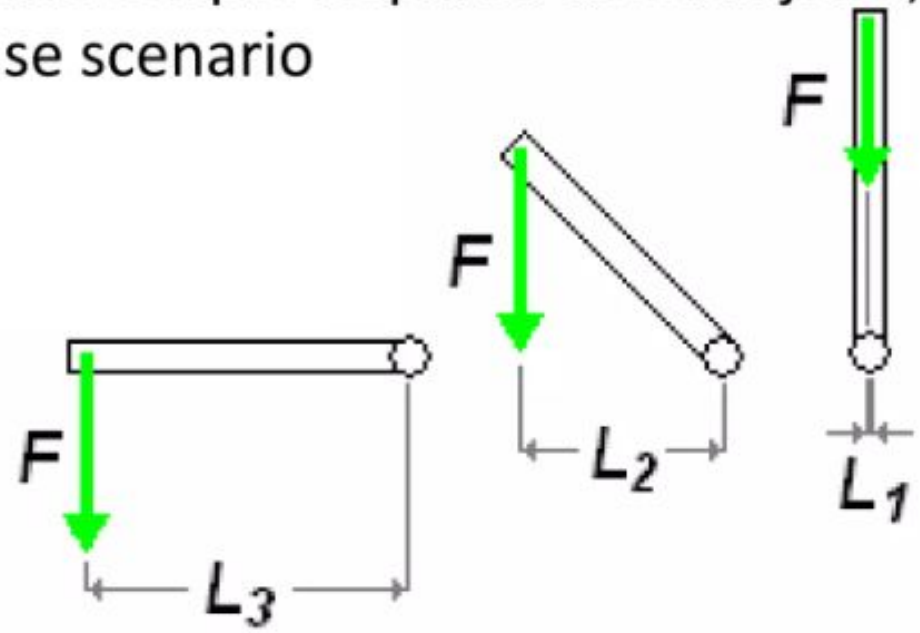
$$\alpha_t = \left[\frac{\phi_g - \phi_s}{\|\phi_g - \phi_s\|} * c \quad \frac{\theta_g - \theta_s}{\|\theta_g - \theta_s\|} * c \quad \frac{\psi_g - \psi_s}{\|\psi_g - \psi_s\|} * c \right]$$



Robot Arm Torque

To estimate the torque required at each joint, we must choose the worst case scenario

Greatest torque

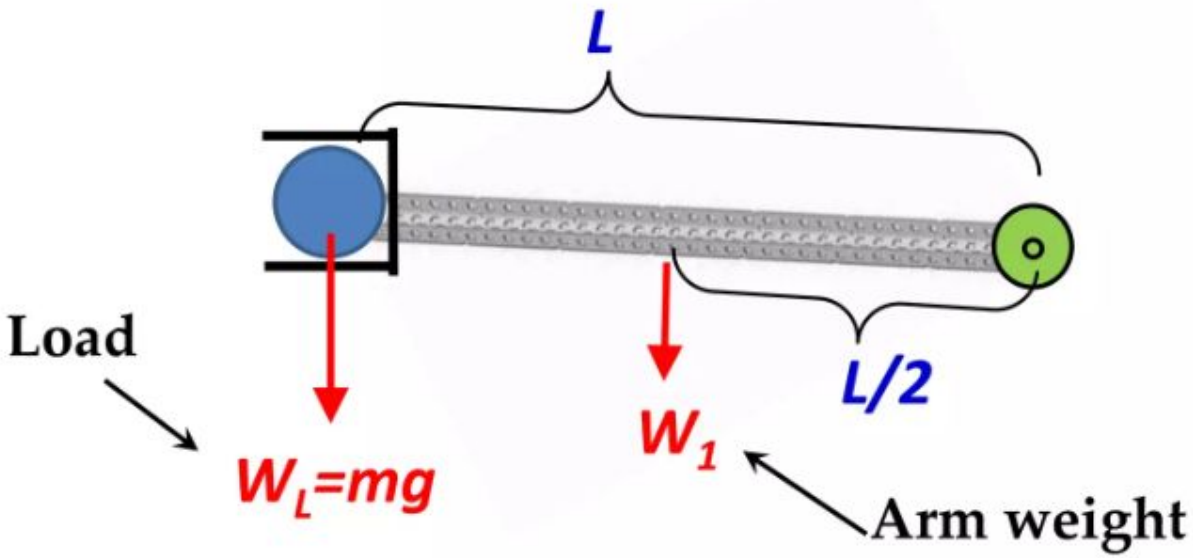


As arm is rotated clockwise, L, the perpendicular distance decreases from L3 to L1 (L1=0). Therefore the greatest torque is at L3 (F does not change) and torque is zero at L1.

Motors are subjected to the highest torque when the arm is stretched out horizontally

Robot Arm Torque

You must also add the torque imposed by the arm itself



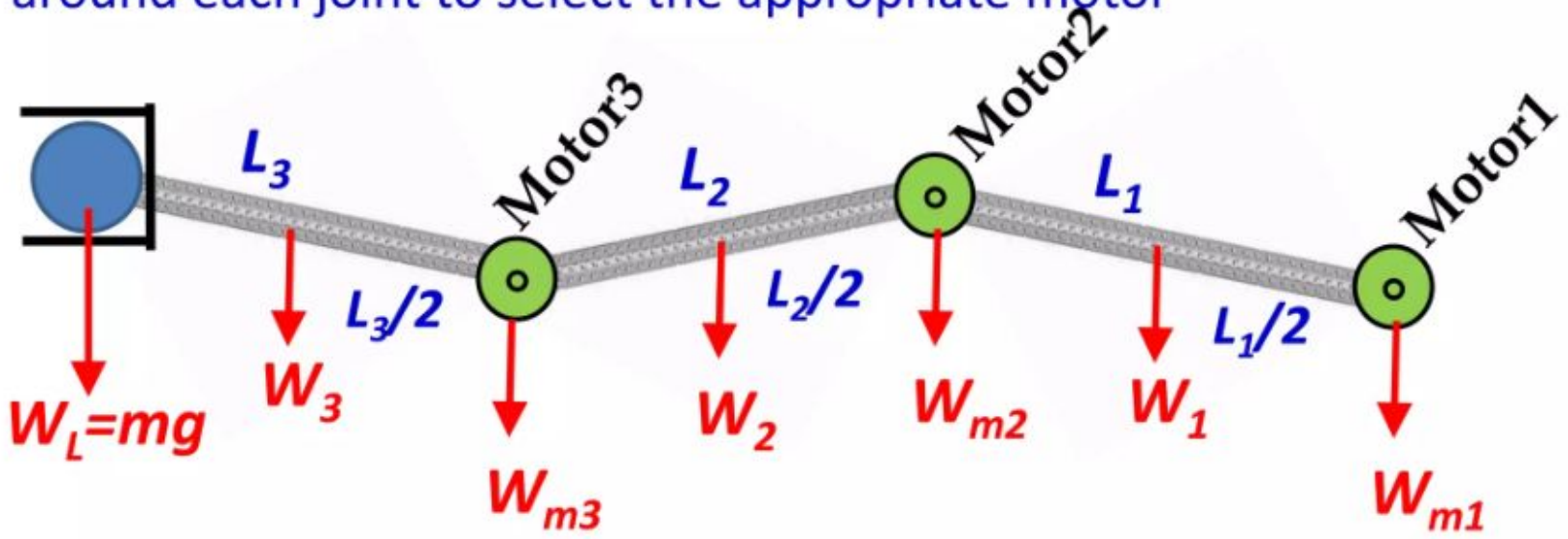
$$\tau = (mg \times L) + (W_1 \times L/2)$$
$$= L(mg + W_1/2)$$

RMF (motor specs) **Robot arm torque**

$$\tau \times rps = L(mg + W1/2)(rps)(1/efficiency)$$

Robot Arm Torque

If your arm has multiple points, you must determine the torque around each joint to select the appropriate motor

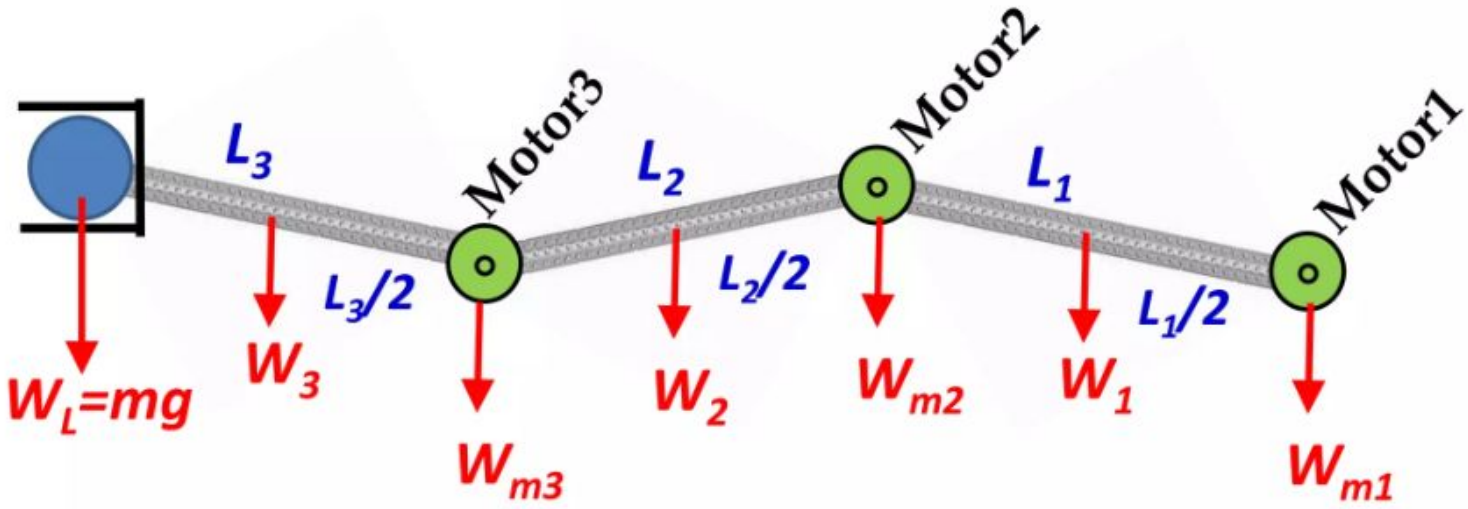


$$\tau_3 = (mg \times L_3) + (W_3 \times L_3 / 2)$$

$$\tau_2 = \left[mg \times (L_3 + L_2)\right] + \left[W_3 \times (L_2 + {}^{L_3}\!/_2)\right] + (W_{m3} \times L_2) + (W_2 \times {}^{L_2}\!/_2)$$

$$\tau_1 = \left[mg \times (L_3 + L_2 + L_1)\right] + \left[W_3 \times (L_1 + L_2 + {}^{L_3}\!/_2)\right] + \left[W_{m3} \times (L_1 + L_2)\right] + \left[W_2 \times (L_1 + {}^{L_2}\!/_2)\right] + (W_{m2} \times L_1) + (W_1 \times {}^{L_1}\!/_2)$$

Robot Arm Torque



Link to Robot Arm Calculator