Trajectory Generation (2/2)

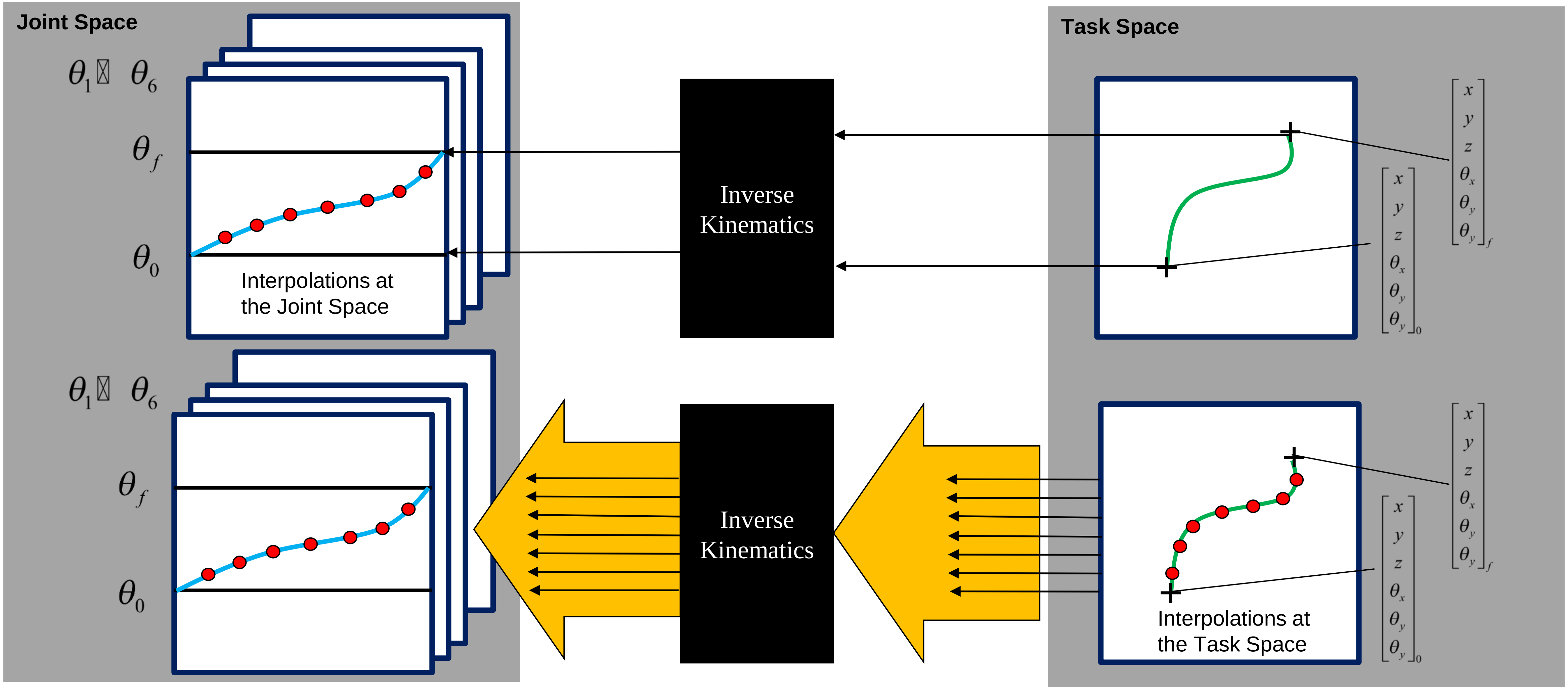
Video

https://youtu.be/_96lcvIBCSA?si=OTvwHuashrGGVZGH

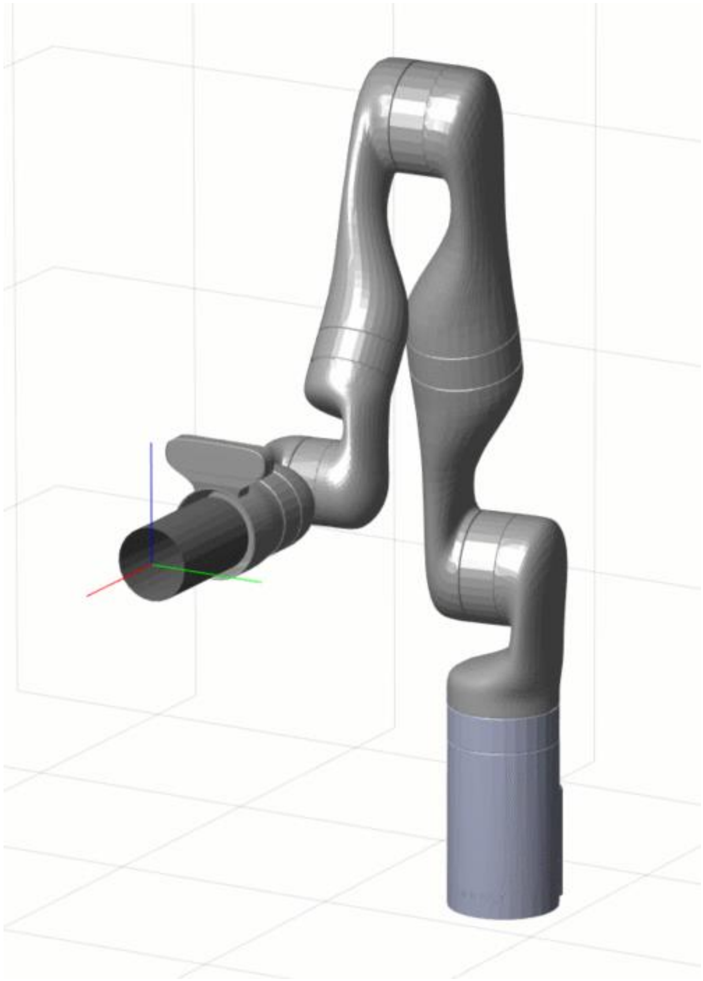
Task Space Schemes

General Discussion

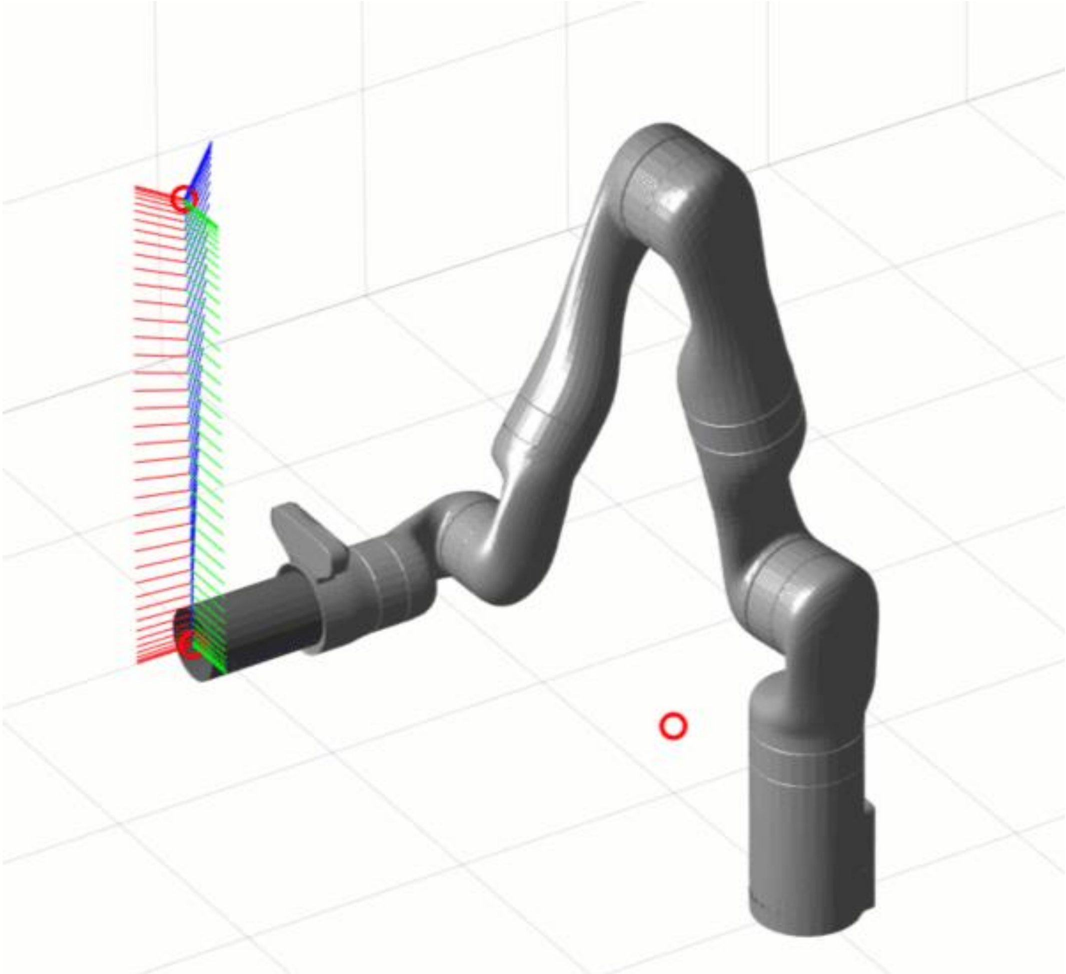
Task Space Versus Joint Space - Interpolations



Task Space Scheme – Problem Definition
Orientation Problem



Task Space Scheme – Problem Definition
Position / Orientation Problem – Trapezoid Velocity

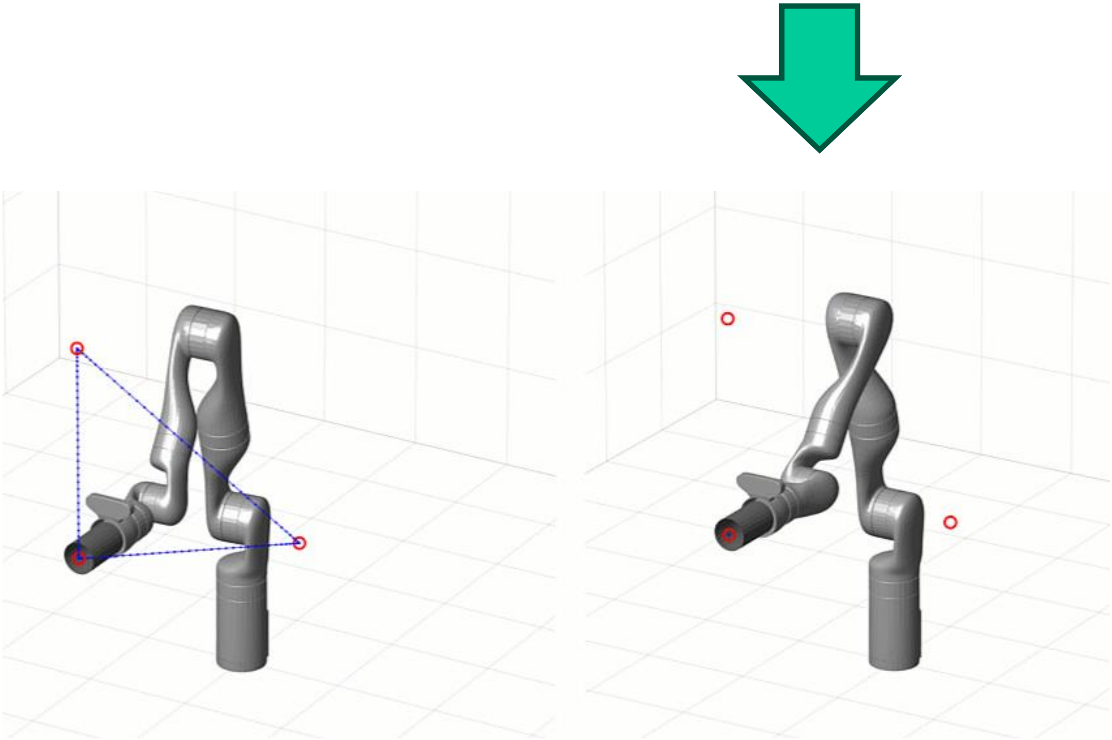


Join Space Versus Task Space – Comparison

Parameter	Joint Space	Task Space
Interpolation Space intermediate points along the trajectory	Joint Space	Task Space
Tool Trajectory Type / Length	Curved Line / Long	Straight Lines / Short
Invers Kinematics (IK) Usage	Low	High
Computation Expense (IK)	Low (IK for Start/Finish & Via Points)	High (IK for every single point / time steo on the trajectory)
Passing through Via Points	No (Correction by establishing Pseudo Points)	Yes
Via Points Defined in the Task Space	No	Yes
Path Dependency on a Specific Manipulator	Yes	No

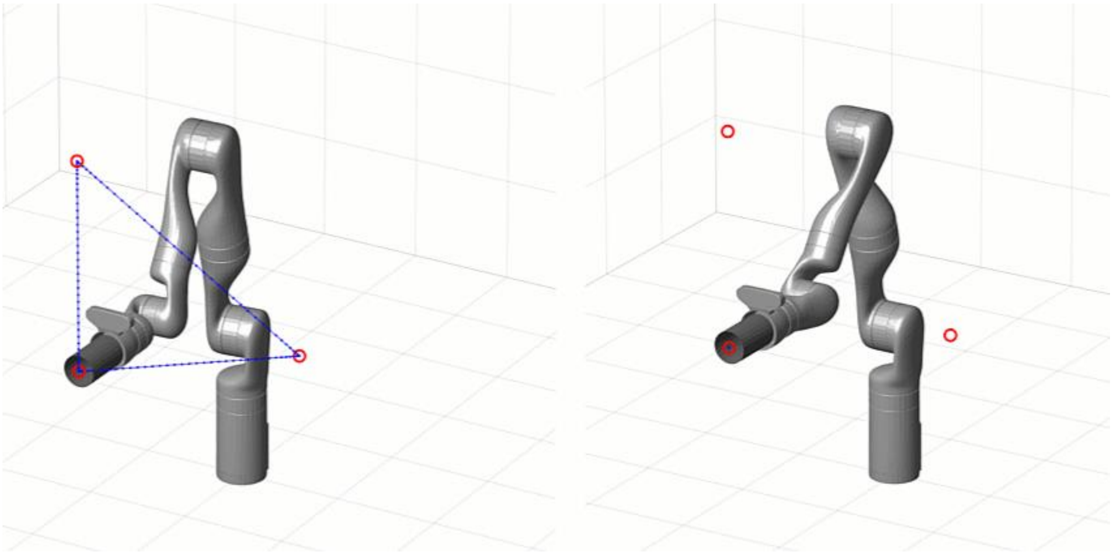
Cartesian Space Schemes – Introduction

- **Joint Space Schemes**
  - **Advantages**
    - Path go through all the via and goal points
    - Points can be specified by Cartesian frames.
  - **Disadvantages** -
    - End effector moves along a curved line (not a straight line - shortest distance).
    - Path depends on the particular joint kinematics of the manipulator i.e. if the type of the manipulator changes the path between the via points will change too.

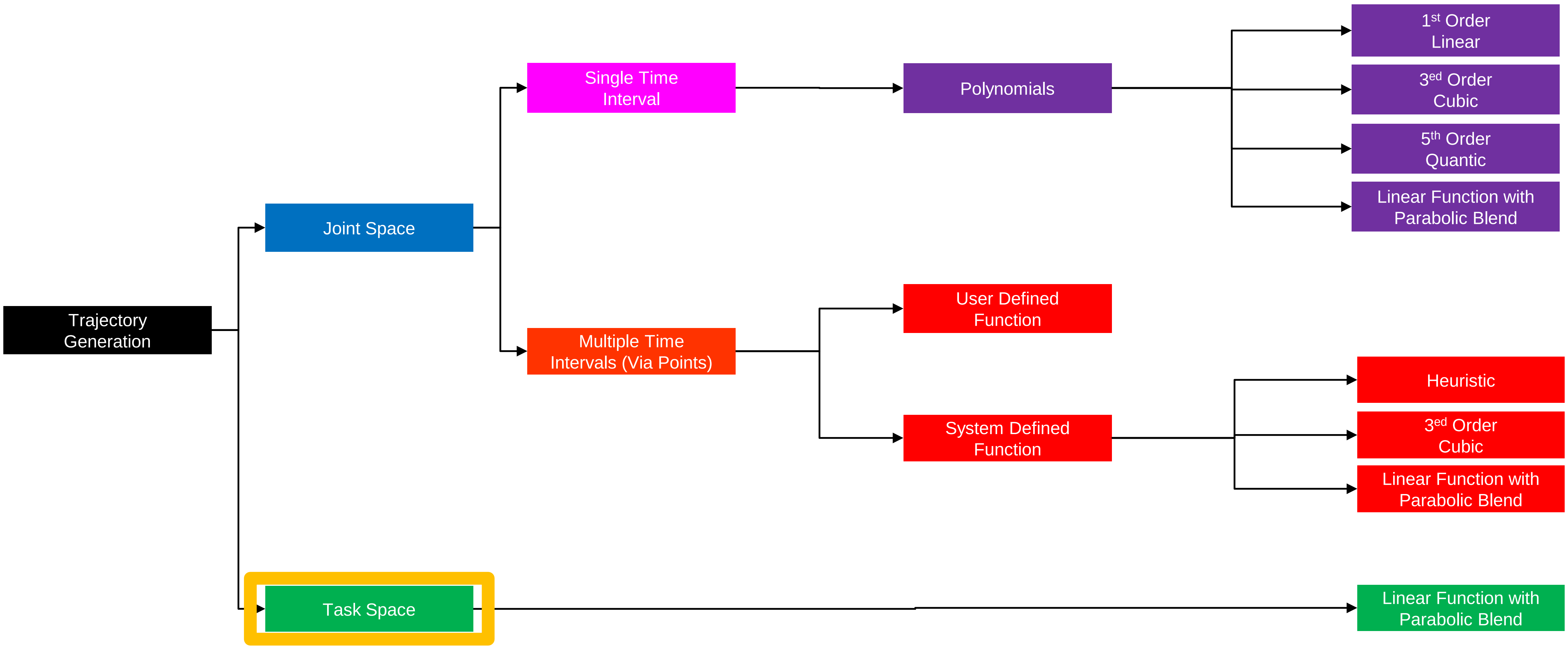


Cartesian Space Schemes – Introduction

- **Cartesian Space Scheme**
  - **Advantage**
    - Most common path is straight line (shortest). Other shapes can also be used.
  - **Disadvantage**
    - Computationally expansive to execute – At run time the inverse kinematics needs to be solved at path update rate (60-2000 Hz)

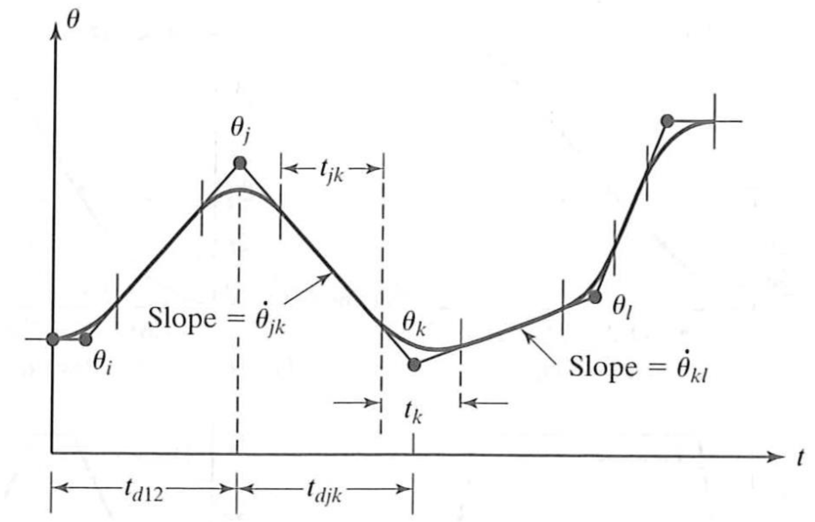


Trajectory Generation – Roadmap Diagram



Cartesian Space Scheme – Cartesian Straight Line

- **General Approach** - Define the path (in the Cartesian space) as
  - Straight lines (linear functions)
  - Parabolic lines (blends)

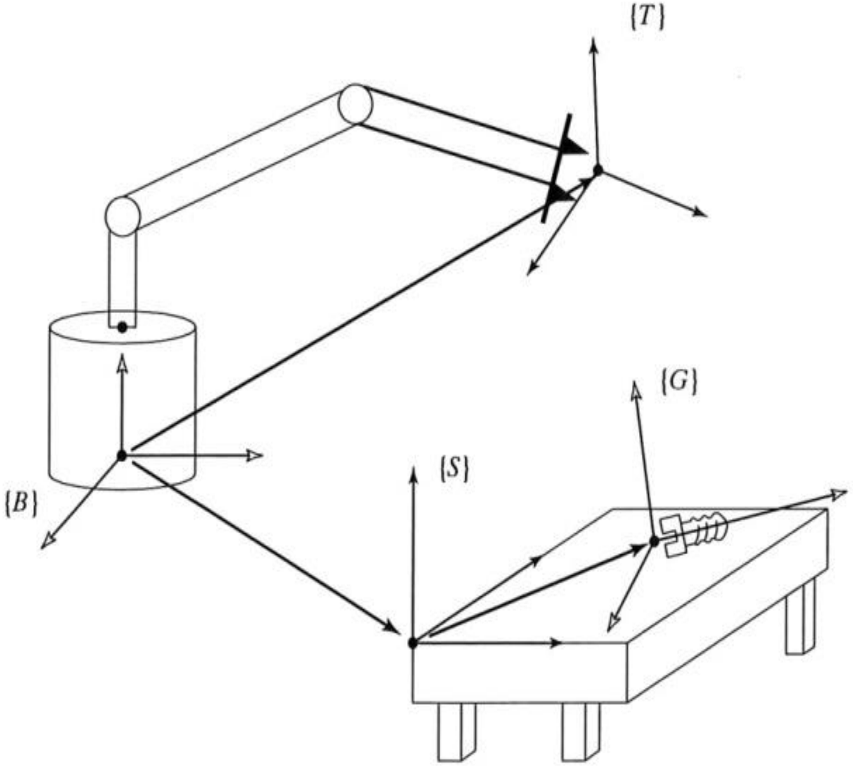


Task Space Scheme – Problem Definition
Position / Orientation Problem

- **General Approach (continue)**
  - Every point along the path is defined by position and orientation of the end effector

$$
{}^{S}_{A}T = \begin{bmatrix} & {}^{S}_{A}R & & {}^{S}P_{AORG} \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

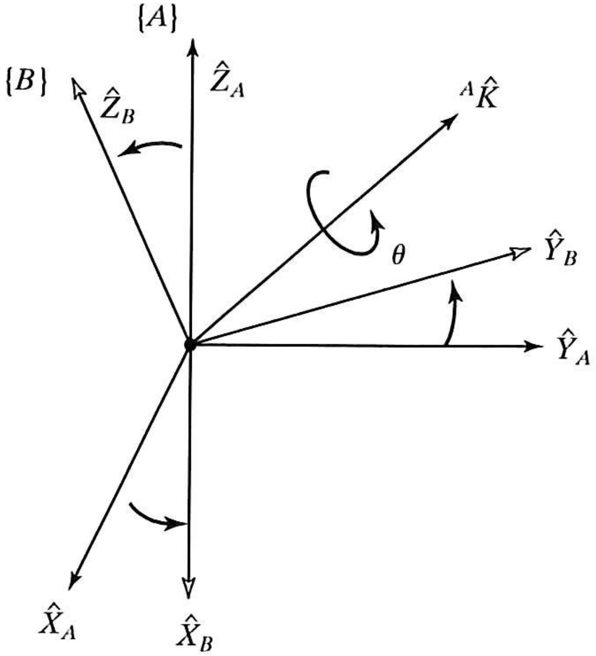
  - **End Effector Position** – Vector – Easy interpolation
  - **End Effector Ordination** – Matrix – Impossible to interpolate (interpolating the individual elements of the matrix violate the requirements that all column of the matrix must be orthogonal)



Task Space Scheme – Problem Definition
Orientation Problem - Equivalent Angle – Axis Representation

- **Euler's Rotation Theorem**
  - Any combination of rotations of a rigid body, is equivalent to a single rotation by θ about some axis that runs through the fixed point $\widehat{{}^{A}K}$.
  - Equivalent Angle – Axis Representation

$$ {}^{A}_{B}R(\hat{K},\theta) \quad or \quad R_{K}(\theta) $$



Task Space Scheme – Problem Definition
Position / Orientation Problem - Equivalent Angle – Axis Representation

- Combining the angle-axis representation of orientation with the 3x1 Cartesian position representation we have a 6x1 representation of Cartesian position and orientation.

- Consider a via point (Point A) specified relative to a station frame (S) as ${}^{S}_{A}T$

$$ {}^{S}_{A}T = \begin{bmatrix} & {}^{S}_{A}R & & {}^{S}P_{AORG} \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

  – Frame {A} specifies a via point
    - Position of the end effector given by ${}^{S}P_{AORG}$
    - Orientation of the end effector given by ${}^{S}_{A}R$

Task Space Scheme – Problem Definition
Orientation Problem - Equivalent Angle – Axis Representation

- **Conversion 1** - Conversion for single angle axis representation to rotation matrix representation

$$\left[P \xrightarrow{\theta,\hat{K}} P'\right] \longrightarrow P' = R_K(\theta)P$$

$$R_k(\theta) = \begin{bmatrix} k_x k_x v\theta + c\theta & k_y k_x v\theta - k_s s\theta & k_x k_Z v\theta + k_y s\theta \\ k_x k_y v\theta + k_z s\theta & k_y k_y v\theta + c\theta & k_y k_z v\theta + k_x s\theta \\ k_x k_z v\theta - k_y s\theta & k_y k_z vA + k_x s\theta & k_z k_z v\theta + c\theta \end{bmatrix}$$

$$\begin{cases} c\theta = \cos(\theta) \\ s\theta = \sin(\theta) \\ v\theta = 1 - \cos(\theta) \end{cases}$$

Task Space Scheme – Problem Definition
Orientation Problem - Equivalent Angle – Axis Representation

- **Conversion 2** – Conversion from a rotation matrix representation single axis represtation

$$P' = R_K(\theta)P \longrightarrow P \xrightarrow{\theta,\hat{K}} P'$$

$$R_K(\theta) = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

$$\theta = \tan^{-1}\left(\frac{\sqrt{(r_{32}-r_{23})^2+(r_{13}-r_{31})^2+(r_{21}-r_{12})^2)}}{r_{11}+r_{22}+r_{33}-1}\right)$$

$$\hat{K} = \frac{1}{2\sin\theta}\begin{bmatrix} r_{32}-r_{23} \\ r_{13}-r_{31} \\ r_{21}-r_{12} \end{bmatrix}$$

Task Space Scheme – Problem Definition
Position / Orientation Problem - Equivalent Angle – Axis Representation

- Convert the rotation matrix into an angle axis representation

$$ {}^{S}_{A}R = ROT({}^{S}\hat{K}_{A}, \theta_{SA}) = {}^{S}K_{A} $$

- Use the symbol χ to represent 6x1 position and orientation

$$ {}^{S}\chi_{A} = \begin{bmatrix} {}^{S}P_{AORG} \\ {}^{S}K_{A} \end{bmatrix} $$

- Where ${}^{S}K_{A}$ is formed by scaling the unite vector ${}^{S}\hat{K}_{A}$ by the amount of rotation θ_{SA}

Task Space Scheme – Cartesian Straight Line

- **Process** - For a given trajectory we describe a spline function that smoothly vary these six quantities from path point to path point as a function of time.

$$ {}^{S}\chi_{A} = \begin{bmatrix} {}^{S}P_{AORG} \\ {}^{S}K_{A} \end{bmatrix} $$

- **Spline type** - Once the vector is defined every single interpolation that is applicable at the Joint Space is also applicable in the task space

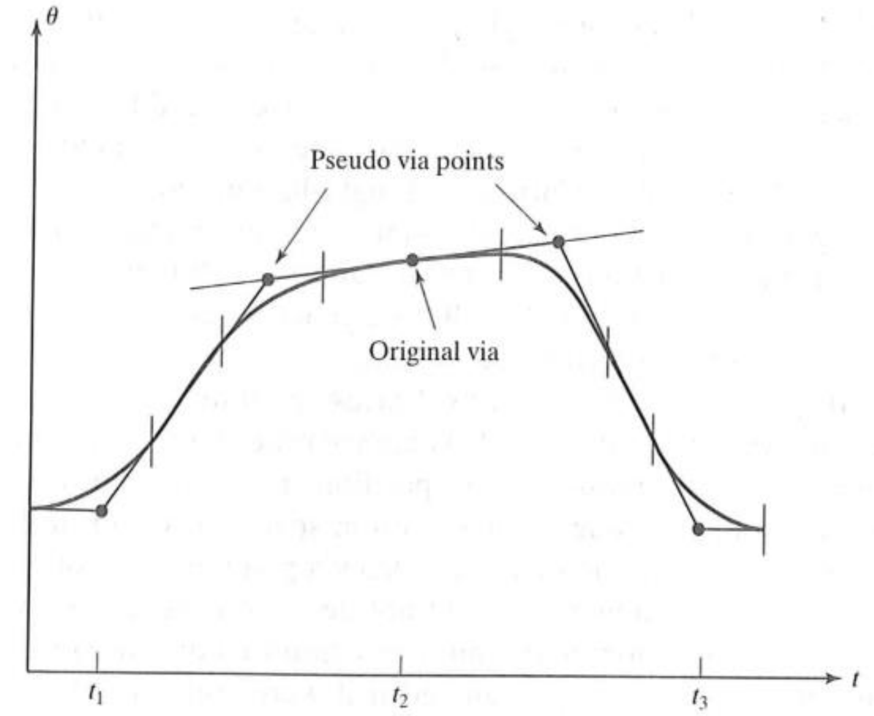
- **Common Spline** - Linear Spline with parabolic bland

Task Space Scheme – Cartesian Straight Line

- The splines are composed of linear and parabolic blend section

- **Constrain**
  - The transition between the linear segment and the parabolic segment for all the DOF must take place at the same time. Therefore using Pseudo via points in the task space is mandatory

$$ {}^{S}\chi_A = \begin{bmatrix} {}^{S}P_{AORG} \\ {}^{S}K_A \end{bmatrix} \qquad {}^{S}\chi_B = \begin{bmatrix} {}^{S}P_{BORG} \\ {}^{S}K_B \end{bmatrix} $$



Task Space Scheme – Cartesian Straight Line

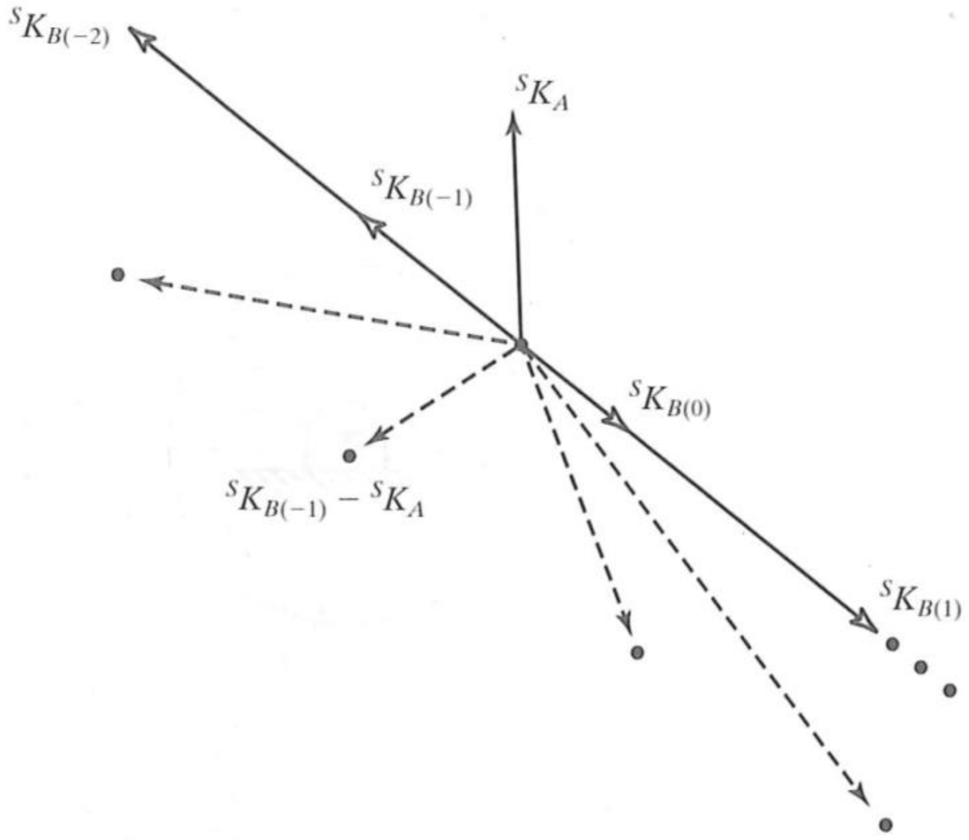
- **Complication –** The angle-axis representation is not unique

$$({}^S\hat{K}_B, \theta_{SB}) = ({}^S\hat{K}_B, \theta_{SB} \pm n360)$$

- In going from via point {A} to a via point {B}, the total amount of rotation should be minimized

- Choose ${}^S\hat{K}_B$ such that

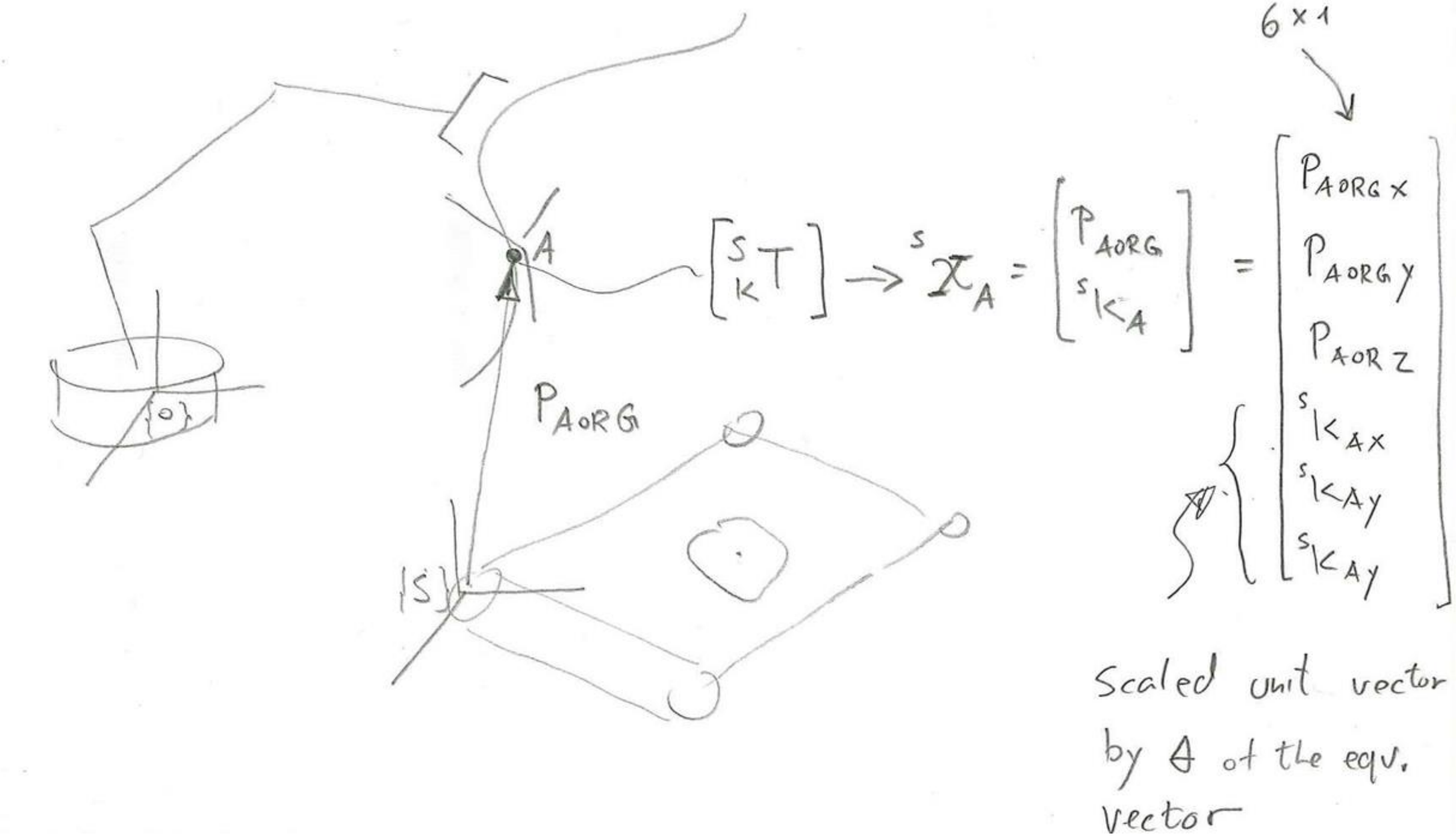
$$\min \left| {}^S\hat{K}_B - {}^S\hat{K}_A \right|$$



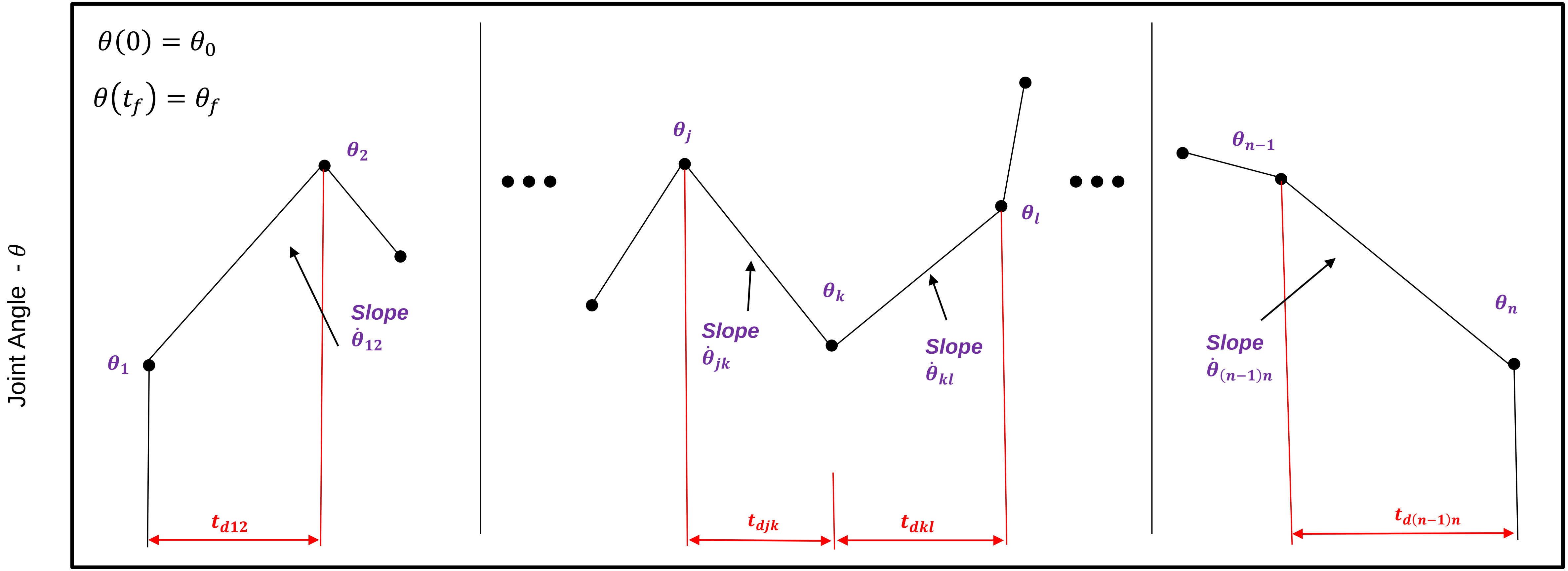
Path Generation – Summary

Task Space

Task Generation at Run Time – Task Space



Joint Space Schemes – Multiple Time Intervals – Via Points – Linear Polynomials



(+) Known - Time Duration between Start / Via / End Points
(+) Known – Joint Angles

Time - t

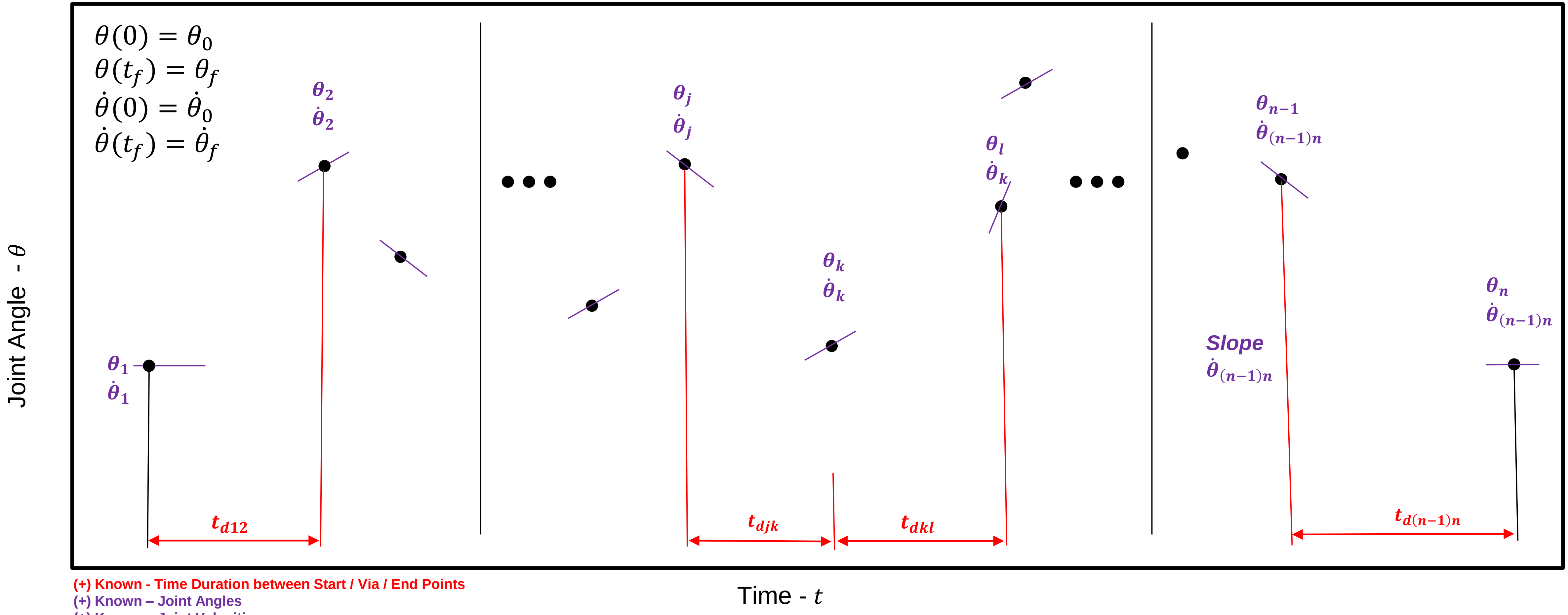
Joint Space Schemes – Multiple Time Intervals – Via Points – Linear Polynomials

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

$$\theta(0) = \theta_0$$

$$\theta(t_f) = \theta_f$$

Joint Space Schemes – Multiple Time Intervals – Via Points – Cubic Polynomials – Non Zero Velocity



(+) Known - Time Duration between Start / Via / End Points
(+) Known – Joint Angles
(+) Known – Joint Velocities

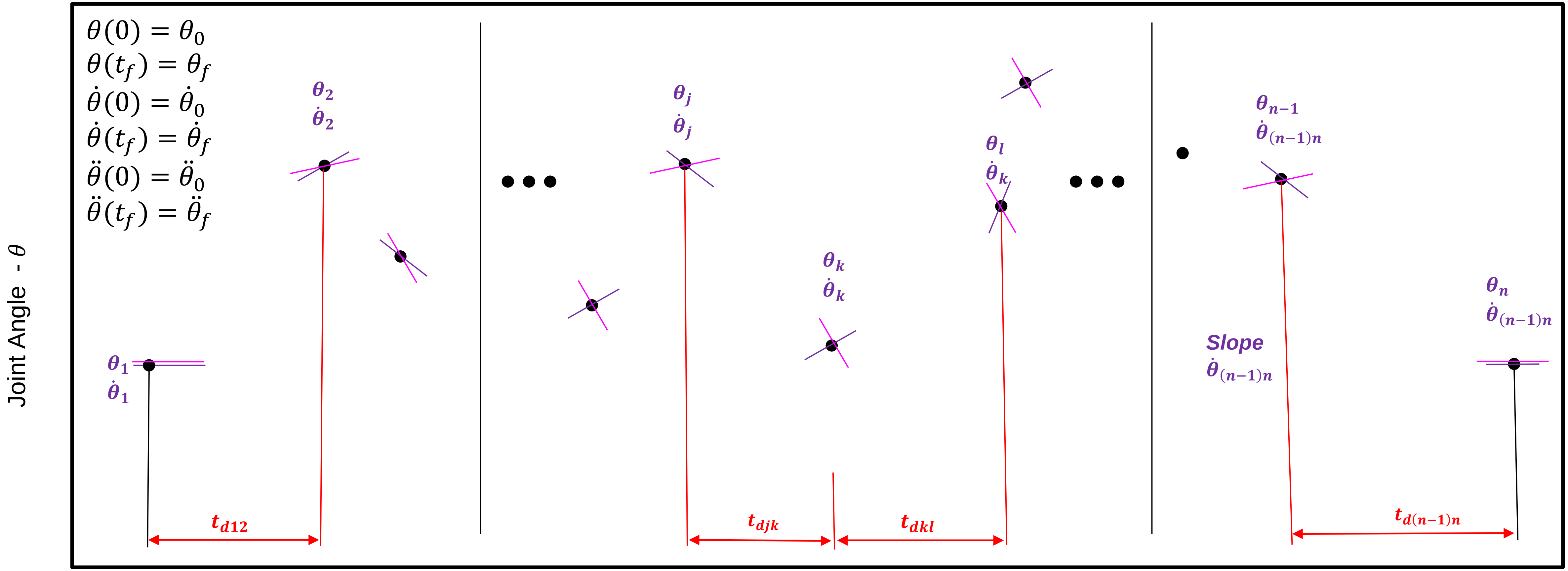
Joint Space Schemes – Multiple Time Intervals – Via Points – Cubic Polynomials – Non Zero Velocity

$$\theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$
$$\dot{\theta}(t) = a_1 + 2a_2 t + 3a_3 t^2$$
$$\ddot{\theta}(t) = 2a_2 + 6a_3 t$$

$$\theta(0) = \theta_0$$
$$\theta(t_f) = \theta_f$$
$$\dot{\theta}(0) = \dot{\theta}_0$$
$$\dot{\theta}(t_f) = \dot{\theta}_f$$

$$a_0 = \theta_0$$
$$a_1 = \dot{\theta}_0$$
$$a_2 = \frac{3}{t_f^2}(\theta_f - \theta_0) - \frac{2}{t_f}\dot{\theta}_0 - \frac{1}{t_f}\dot{\theta}_f$$
$$a_3 = -\frac{2}{t_f^3}(\theta_f - \theta_0) + \frac{2}{t_f^2}(\dot{\theta}_f + \dot{\theta}_0)$$

Joint Space Schemes – Multiple Time Intervals – Via Points – Quantic Polynomials - Non Zero Acceleration



(+) Known - Time Duration between Start / Via / End Points
(+) Known – Joint Angles
(+) Known – Joint Velocities

Time - t

(+) Known – Joint Acceleration

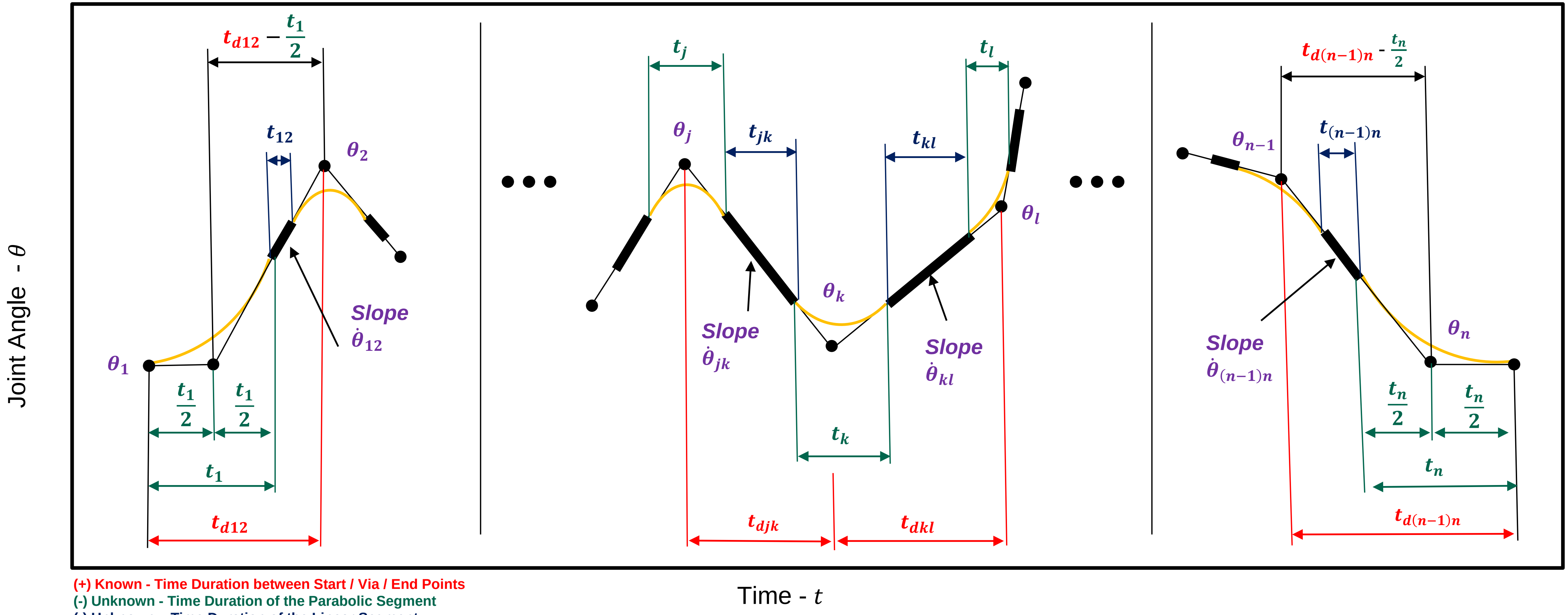
Joint Space Schemes – Multiple Time Intervals – Via Points – Quantic Polynomials - Non Zero Acceleration

$$\theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5$$
$$\dot{\theta}(t) = a_1 + 2a_2 t + 3a_3 t^2 + 4a_4 t^3 + 5a_5 t^4$$
$$\ddot{\theta}(t) = 2a_2 + 6a_3 t + 12a_4 t^2 + 20a_5 t^3$$

$$\theta(0) = \theta_0$$
$$\theta(t_f) = \theta_f$$
$$\dot{\theta}(0) = \dot{\theta}_0$$
$$\dot{\theta}(t_f) = \dot{\theta}_f$$
$$\ddot{\theta}(0) = \ddot{\theta}_0$$
$$\ddot{\theta}(t_f) = \ddot{\theta}_f$$

$$a_0 = \theta_0$$
$$a_1 = \dot{\theta}_0$$
$$a_2 = \frac{\ddot{\theta}_0}{2}$$
$$a_3 = \frac{20\theta_f - 20\theta_0 - (8\dot{\theta}_f + 12\dot{\theta}_0)t_f - (3\ddot{\theta}_0 - \ddot{\theta}_f)t_f^2}{2t_f^3}$$
$$a_4 = \frac{30\theta_0 - 30\theta_f + (14\dot{\theta}_f + 16\dot{\theta}_0)t_f + (3\ddot{\theta}_0 - 2\ddot{\theta}_f)t_f^2}{2t_f^4}$$
$$a_5 = \frac{12\theta_f - 12\theta_0 - (6\dot{\theta}_f + 6\dot{\theta}_0)t_f - (\ddot{\theta}_0 - \ddot{\theta}_f)t_f^2}{2t_f^5}$$

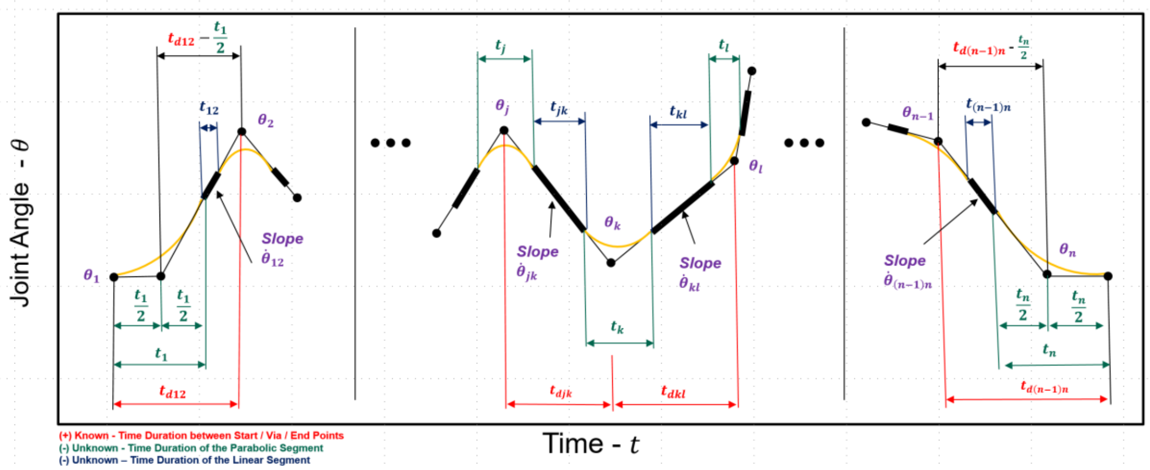
Joint Space Schemes – Multiple Time Intervals – Via Points – Linear Function With Parabolic Blend



(+) Known - Time Duration between Start / Via / End Points
(-) Unknown - Time Duration of the Parabolic Segment
(-) Unknown – Time Duration of the Linear Segment

Joint Space Schemes – Multiple Time Intervals – Via Points – Linear Function With Parabolic Blend

- **Tasks No. 1 – Time Intervals / Velocity / Acceleration**
  - Calculate the **time intervals** of the parabolic blending (marked in green)
  - Calculate the **time intervals** of the linear functions (marked in blue)
  - Calculate the **direction of the acceleration** during the linear bland
  - Calculate the velocity during the linear spline



- **Task No. 2 – Functions**
  - Define the Linear Functions (marked in black)
  - Defined the parabolic blend functions (marked in gray)

Assume 4 intervals

- First Interval: 1→2
- Intermediate Interval: 2 → 3
- Last Interval: 2 → 3

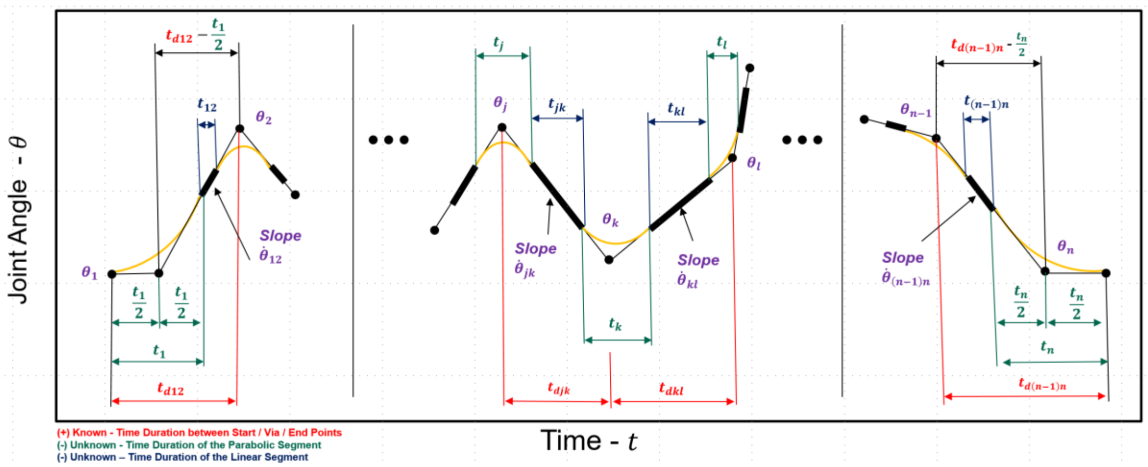
Joint Space Schemes – Multiple Time Intervals – Via Points – Linear Function With Parabolic Blend

- **Tasks No. 1**
  - **First Interval -** 1→2

$$t_1 = t_{d12} - \sqrt{t_{d12}^2 - \frac{2(\theta_2 - \theta_1)}{\ddot{\theta}_1}}$$

$$\ddot{\theta}_1 = SGN(\theta_2 - \theta_1)\left|\ddot{\theta}_1\right|$$

$$\dot{\theta}_{12} = \frac{\theta_2 - \theta_1}{t_{d12} - \frac{1}{2}t_1}$$



Joint Space Schemes – Multiple Time Intervals – Via Points – Linear Function With Parabolic Blend

- **Tasks No. 1**
  - **Intermediate Interval** (Repeat for any intermediate interval) - 2→3

$$\dot{\theta}_{jk} = \frac{\theta_k - \theta_j}{t_{djk}}$$

$$\begin{cases} j = 2 \\ k = 3 \end{cases} \rightarrow \dot{\theta}_{23} = \frac{\theta_3 - \theta_1}{t_{d23}}$$

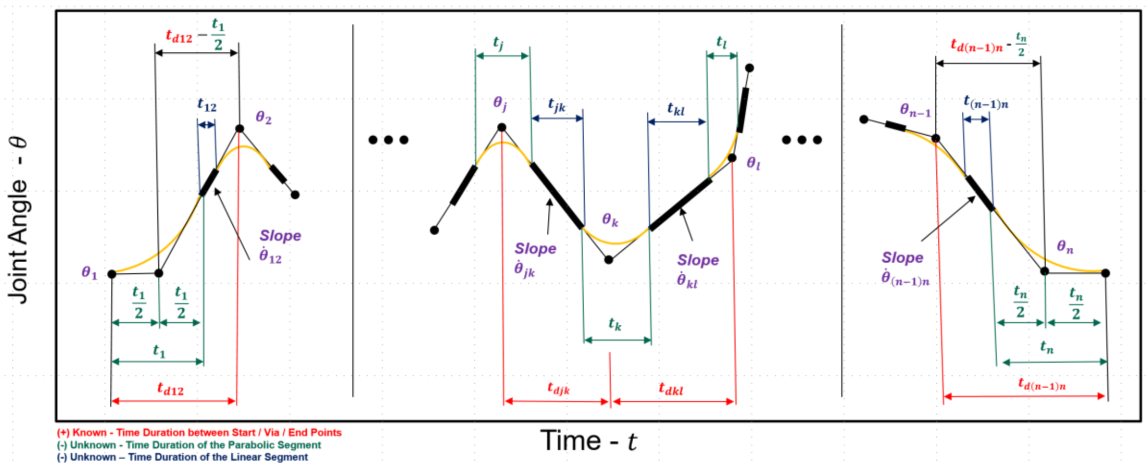
$$\ddot{\theta}_k = SIG(\dot{\theta}_{kl} - \dot{\theta}_{jk})|\ddot{\theta}_k|$$

$$\begin{cases} j = 1 \\ k = 2 \\ l = 3 \end{cases} \rightarrow \ddot{\theta}_2 = SIG(\dot{\theta}_{23} - \dot{\theta}_{12})|\ddot{\theta}_2|$$

$$t_k = \frac{\dot{\theta}_{kl} - \dot{\theta}_{jk}}{\ddot{\theta}_k}$$

$$\begin{cases} j = 1 \\ k = 2 \\ l = 3 \end{cases} \rightarrow t_2 = \frac{\dot{\theta}_{23} - \dot{\theta}_{12}}{\ddot{\theta}_2}$$

$$t_{d12} - t_1 - \frac{1}{2} t_2$$



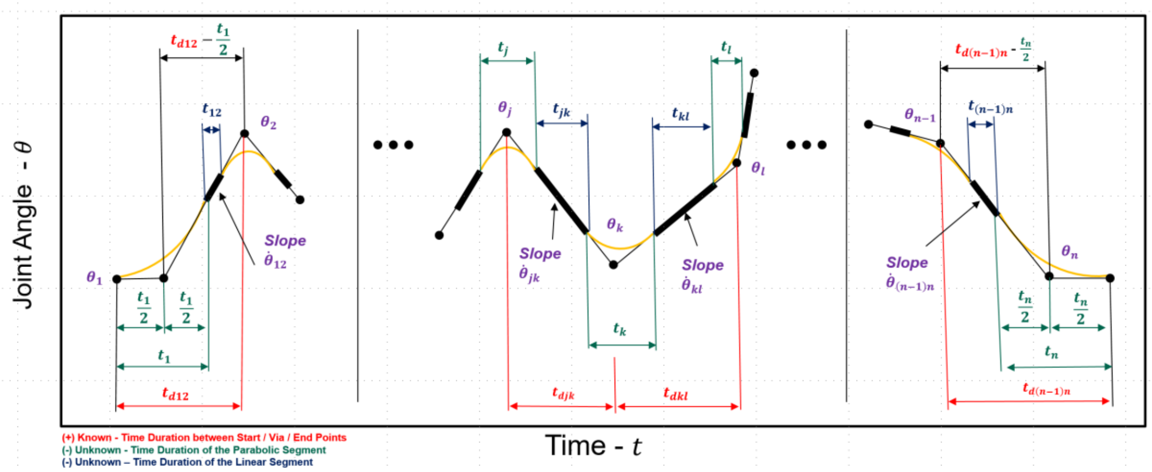
Joint Space Schemes – Multiple Time Intervals – Via Points – Linear Function With Parabolic Blend

- **Tasks No. 1**
  - **Final Interval -** 3→4

$$\ddot{\theta}_n = SGN(\theta_{n-1} - \theta_n)|\ddot{\theta}_n|$$
$$\mathrm{n} = 4 \rightarrow \ddot{\theta}_4 = SGN(\theta_3 - \theta_4)|\ddot{\theta}_4|$$

$$t_n = t_{d(n-1)n} - \sqrt{t_{d(n-1)n}^2 - \frac{2(\theta_{n-1} - \theta_n)}{\ddot{\theta}_n}}$$
$$\mathrm{n} = 4 \rightarrow t_4 = t_{d34} - \sqrt{t_{d34}^2 - \frac{2(\theta_3 - \theta_4)}{\ddot{\theta}_4}}$$

$$\dot{\theta}_{(n-1)n} = \frac{\theta_n - \theta_{n-1}}{t_{d(n-1)n} - \frac{1}{2}t_n}$$
$$n = 4 \rightarrow \dot{\theta}_{34} = \frac{\theta_4 - \theta_3}{t_{d34} - \frac{1}{2}t_4}$$



Joint Space Schemes – Multiple Time Intervals – Via Points – Linear Function With Parabolic Blend

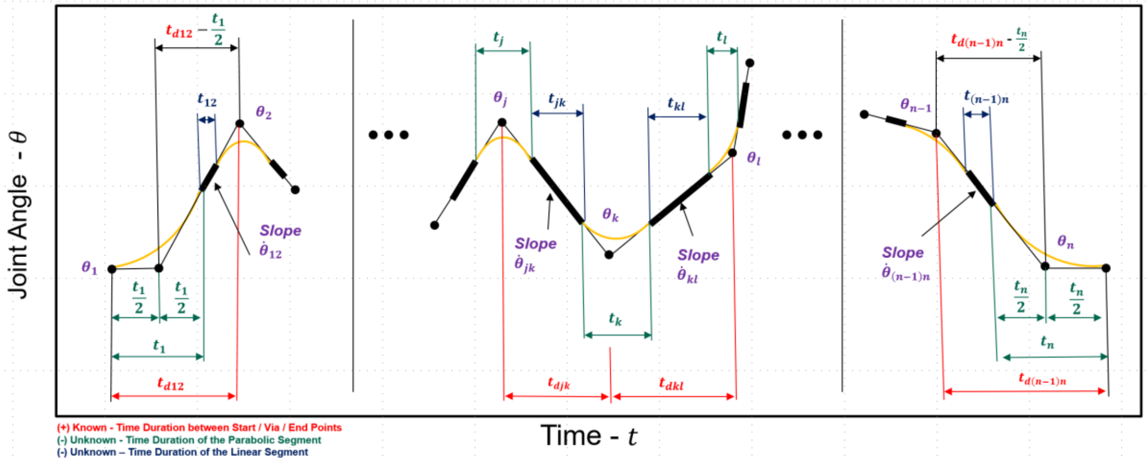
- **Tasks No. 1**
  - **Final Interval -** 3→4 (Continue)
  - (Interval Prior to the final interval)

$$\ddot{\theta}_k = SGN(\dot{\theta}_{kl} - \dot{\theta}_{jk})|\ddot{\theta}_k|$$

$$\begin{cases} j = 2 \\ k = 3 \\ l = 4 \end{cases} \rightarrow \ddot{\theta}_3 = SGN(\dot{\theta}_{34} - \dot{\theta}_{23})|\ddot{\theta}_3|$$

$$t_k = \frac{\dot{\theta}_{kl} - \dot{\theta}_{jk}}{\ddot{\theta}_k}$$

$$\begin{cases} j = 2 \\ k = 3 \\ l = 4 \end{cases} \rightarrow t_3 = \frac{\dot{\theta}_{34} - \dot{\theta}_{23}}{\ddot{\theta}_3}$$



Joint Space Schemes – Multiple Time Intervals – Via Points – Linear Function With Parabolic Blend

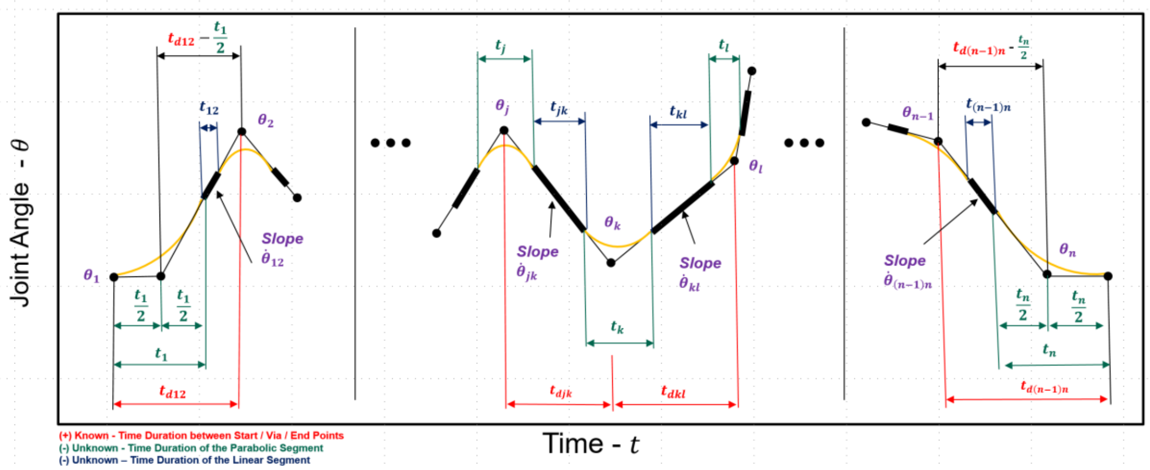
- **Tasks No. 1**
  - **Final Interval -** 3→4 (Continue)
  - (Interval Prior to final interval & Final Inetrval)

$$t_{jk} = t_{djk} - \frac{1}{2}t_j - \frac{1}{2}t_k$$

$$\begin{cases} j = 2 \\ k = 3 \end{cases} \rightarrow t_{23} = t_{d23} - \frac{1}{2}t_2 - \frac{1}{2}t_3$$

$$t_{(n-1)n} = t_{d(n-1)n} - t_n - \frac{1}{2}t_{n-1}$$

$$n = 4 \rightarrow t_{34} = t_{d34} - t_4 - \frac{1}{2}t_3$$



Joint Space Schemes – Multiple Time Intervals – Via Points – Linear Function With Parabolic Blend – Example

- **Tasks No. 2 – Linear & parabolic functions**
  - First Segment

$\theta = \theta_1 + \dot{\theta}_{12} t$

$\theta = \theta_0 + \frac{1}{2}\frac{\dot{\theta}_{12}}{t_1} t^2$

$t_{inb} = t$

  - Mid Segment

$\theta = \theta_j + \dot{\theta}_{jk} t$

$\theta = \theta_j + \dot{\theta}_{jk}(t - t_{inb}) + \frac{1}{2}\ddot{\theta}_k^2 t_{inb}$

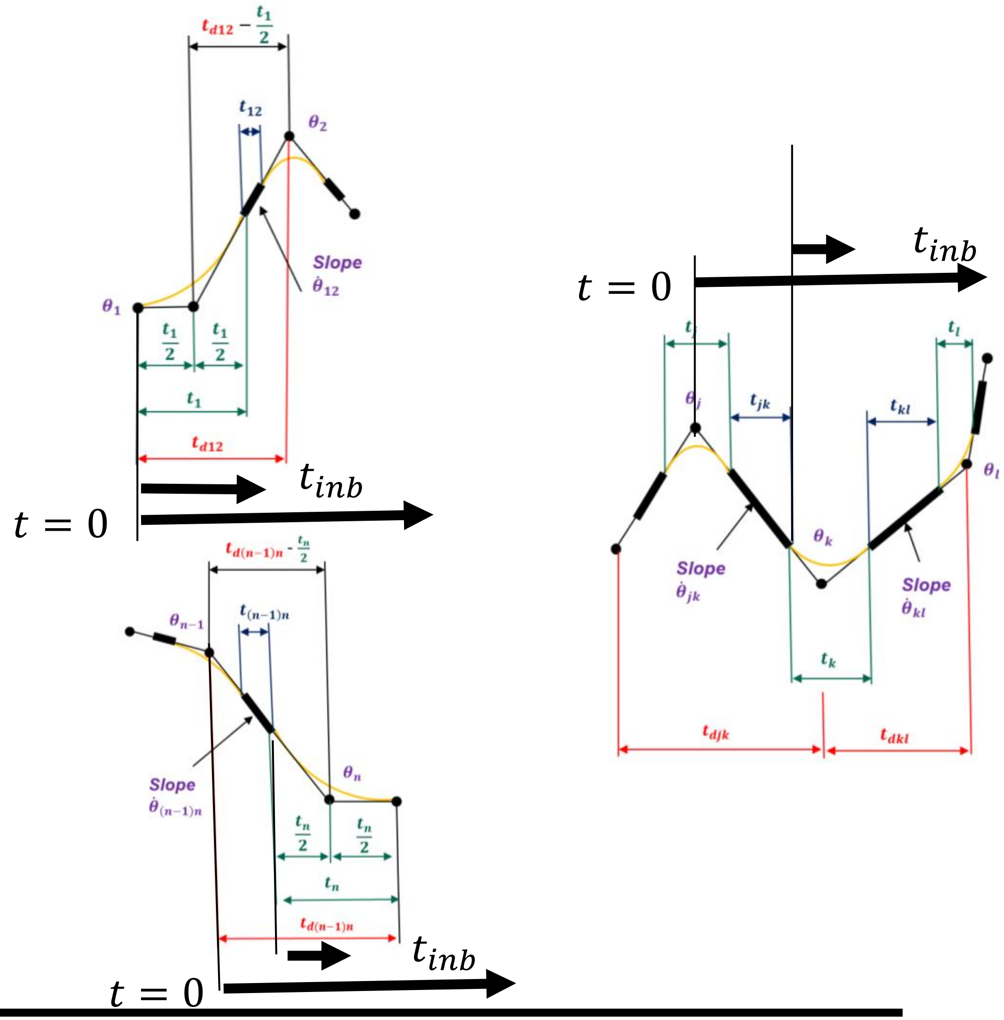
$t_{inb} = t - \left(\frac{1}{2} t_j + t_{jk}\right)$

  - Last Segment

$\theta = \theta_{n-1} + \dot{\theta}_{(n-1)n} t$

$\theta = \theta_{inb} + \theta_{(n-1)n} t_{inb} - \frac{1}{2}\frac{\dot{\theta}_{(n-1)n}}{t_{inb}} t_{inb}^2$

$t_{inb} = t - \left(\frac{1}{2} t_{n-1} + t_{(n-1)n}\right)$

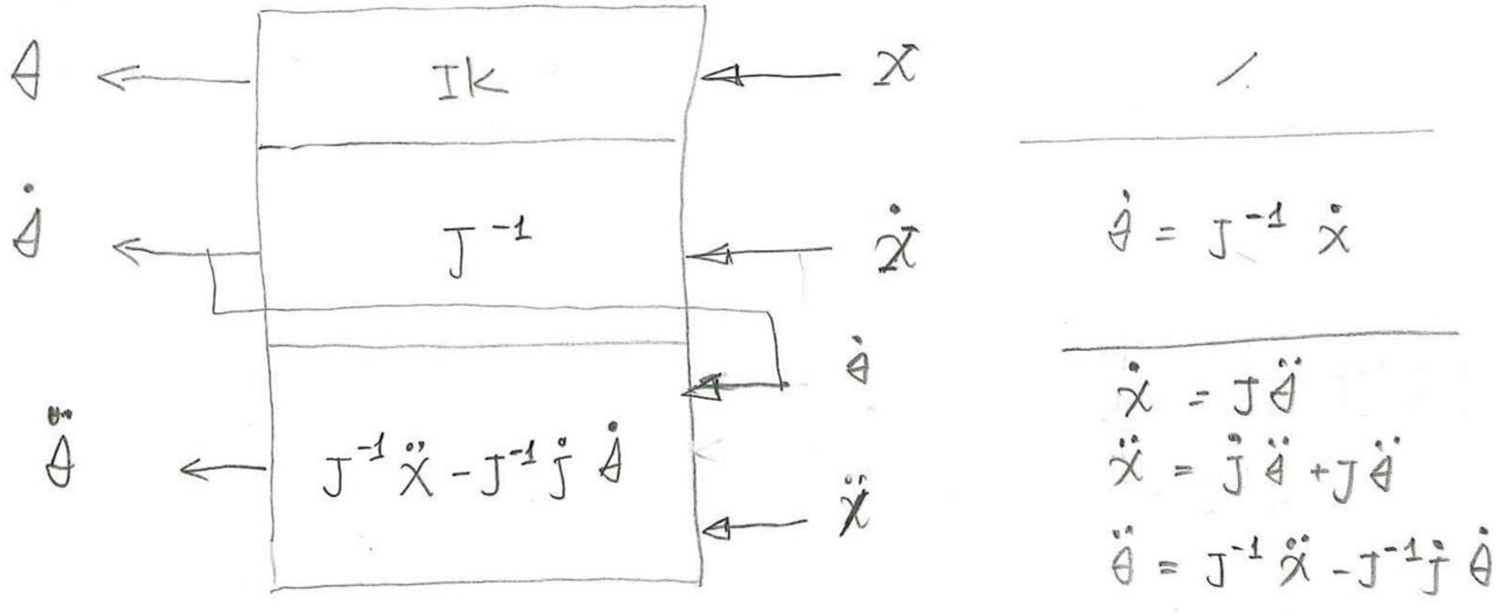


Path Generation & Run Time – Summary

Task Space

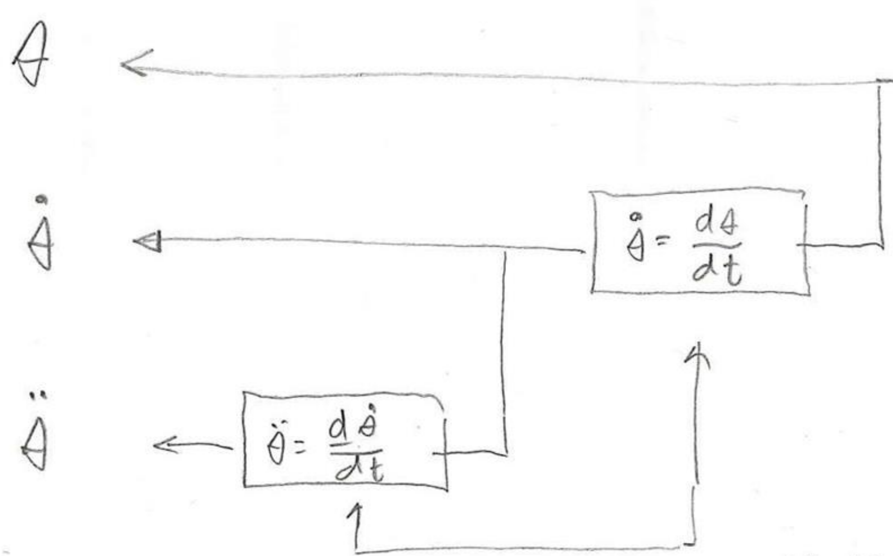
Mapping $\chi, \dot{\chi}, \ddot{\chi}$ from the task space to the joint space

Option 1 – time derivative are done at the task space



Task Generation at Run Time – Task/Joint Space Mapping

Option 2 – time derivative of $\dot{\theta}, \ddot{\theta}$ are done at the joint space



$$\dot{\theta} = \frac{\theta(t) - \theta(t - \delta t)}{\delta t}$$

$$\ddot{\theta} = \frac{\dot{\theta}(t) - \dot{\theta}(t - \delta t)}{\delta t}$$

Note:

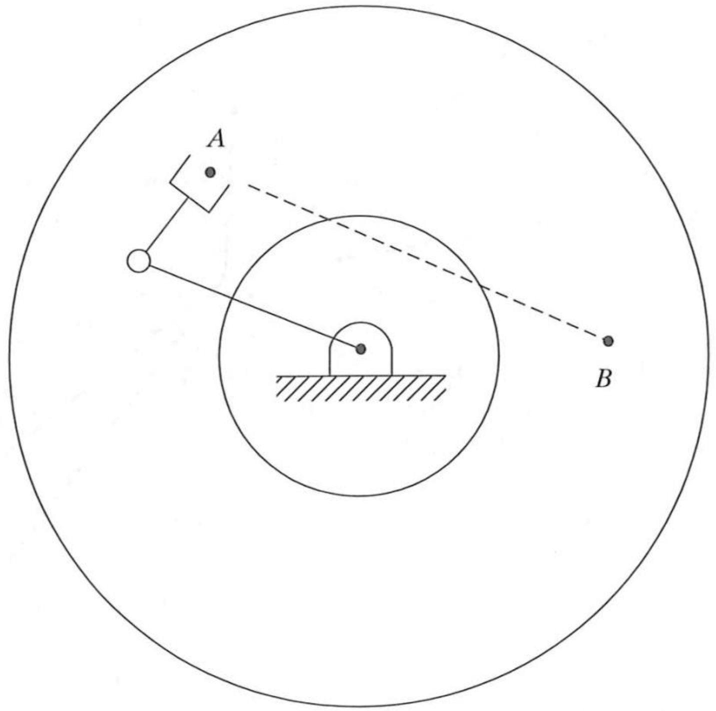
- This differentiation can be done off-line resulting in better quality of $\dot{\theta}, \ddot{\theta}$
- Many control systems do not require a $\ddot{\theta}$ input

Task Space Schemes

Geometric Problems with Paths in Task Space

Geometric Problems – Cartesian Paths

- **Problem Type 1 – Unreachable Intermediate Points**
- The initial and the final point are in the reachable workspace however some point along the path may be out of the workspace.
- **Solution**
  - Joint space path – unreachable
  - Cartesian straight Path – reachable



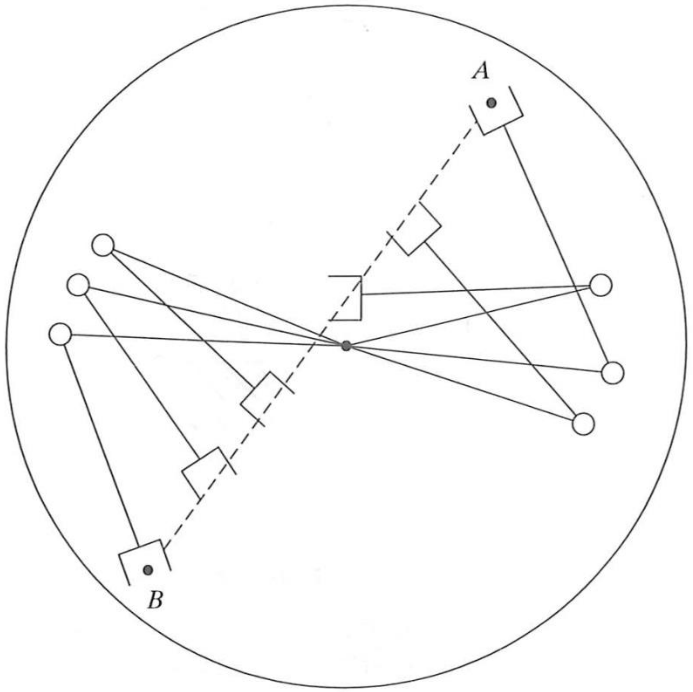
Geometric Problems – Cartesian Paths

- **Problem Type 2 – High Joint Rate Near Singularity.**

- In singularity the velocity of one or more joint approach infinity.

- The velocity of the mechanism are upper bounded, approaching singularity results in the manipulator's deviation form the desired path.

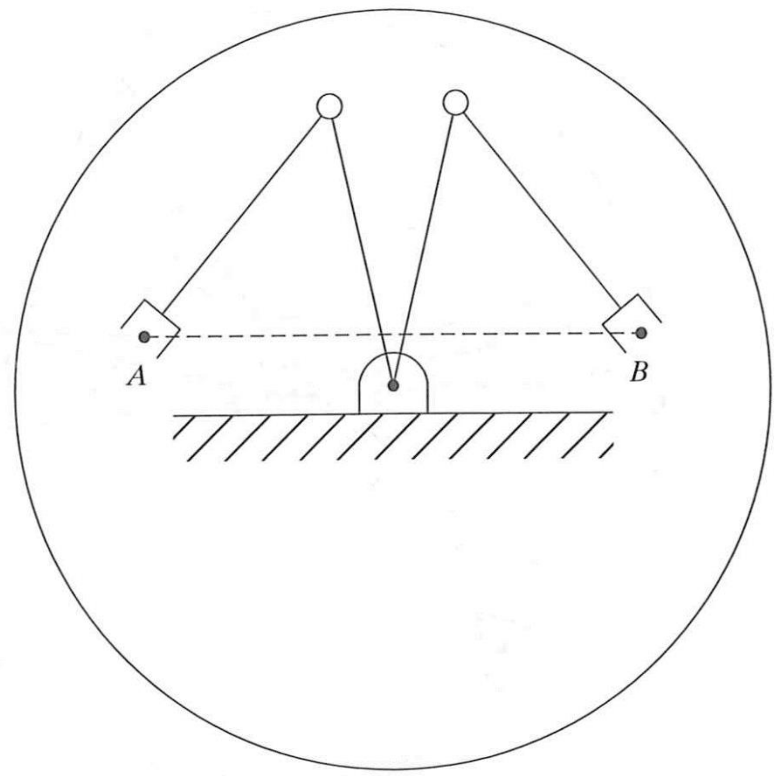
- **Solution**
  - Slow down the velocity such that all the joint velocities will remain in their bounded velocities



- **Problem Type 3 – Start and Goal reachable in different solutions**

- Joint limits may restrict the number of solutions that the manipulator may use given a goal point.

- Solution
  - Switch between joint space (default) and Cartesian space trajectories (used only if needed)

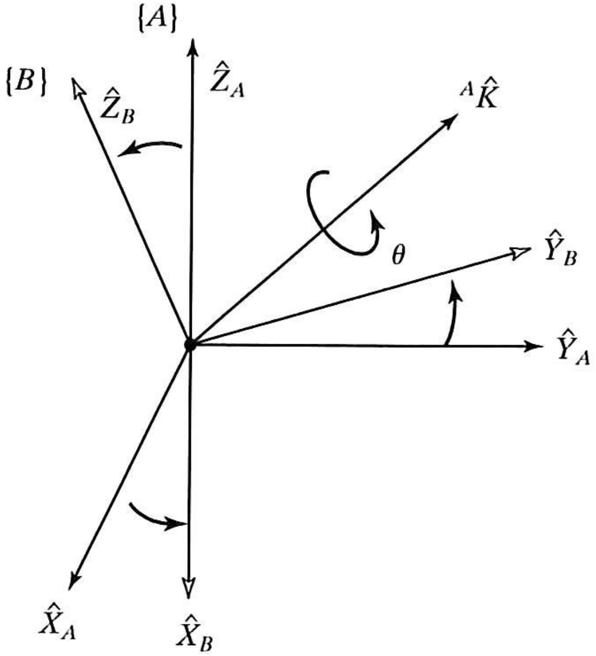


Euler’s Theorem - Equivalent Axis

Derivation

Task Space Scheme – Problem Definition
Orientation Problem - Equivalent Angle – Axis Representation

- **Euler's Rotation Theorem**
  - Any combination of rotations of a rigid body, is equivalent to a single rotation by θ about some axis ${}^{A}\hat{K}$ that runs through the fixed point.
  - Equivalent Angle – Axis Representation



$$ {}^{A}_{B}R(\hat{K},\theta) \quad or \quad R_K(\theta) $$

Task Space Scheme – Problem Definition
Orientation Problem - Equivalent Angle – Axis Representation

- Start with the frame coincident with a know frame {A}; then rotate frame {B} about a vector ${}^{A}\hat{K}$ by an angle θ according to the right hand rule.

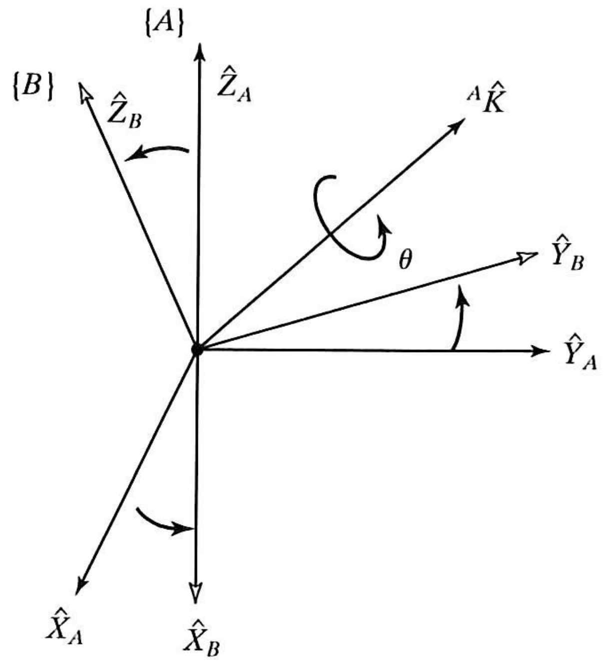
- Equivalent Angle – Axis Representation

$$ {}^{A}_{B}R(\hat{K},\theta) \quad or \quad R_K(\theta) $$

- Vector ${}^{A}\hat{K}$ is called the equivalent axis of a finite rotation.

- The specification of ${}^{A}\hat{K}$ requires two parameters since it length is always 1.

- The angle specify the third parameter



Task Space Scheme – Problem Definition
Orientation Problem - Equivalent Angle – Axis Representation

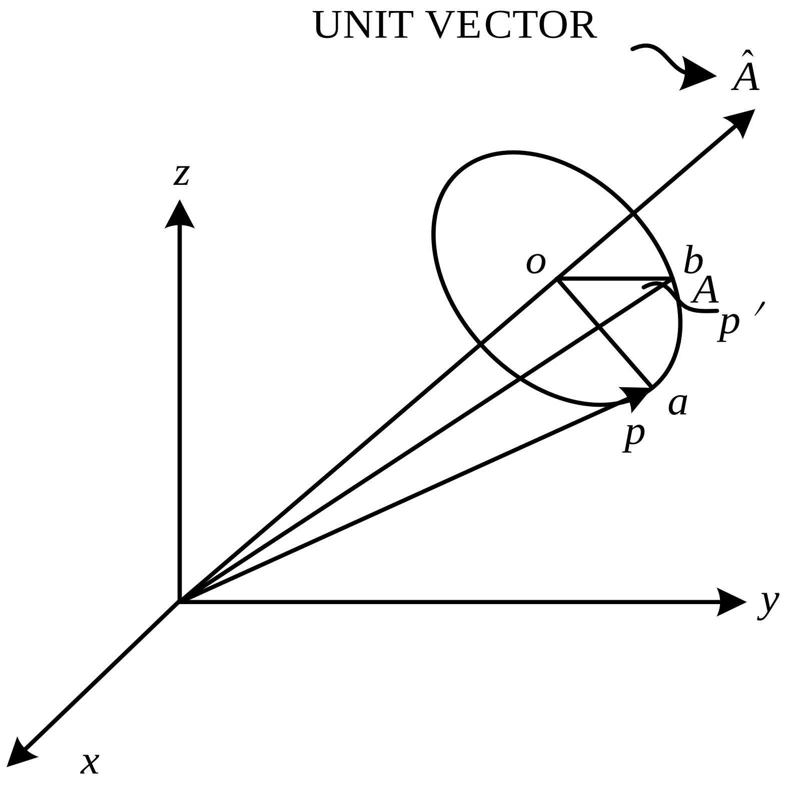
- Rotate a vector P through an angle θ about an arbitrary axis whose direction is represented by a unite vector $\hat{A}$

- Types of Transformations
  - Transformation for a vector/angle form to a matrix form

$$\left[P \xrightarrow{\theta,\hat{A}} P'\right] \longrightarrow P' = R_A(\theta)P$$

  - Transformation from a matrix to a vector angle form

$$P' = R_A(\theta)P \longrightarrow P \xrightarrow{\theta,\hat{A}} P'$$



Task Space Scheme – Problem Definition
Orientation Problem - Equivalent Angle – Axis Representation

Transformation for a vector/angle form to a matrix form

$$\left[P \xrightarrow{\theta,\hat{A}} P'\right] \longrightarrow P' = R_A(\theta)P$$

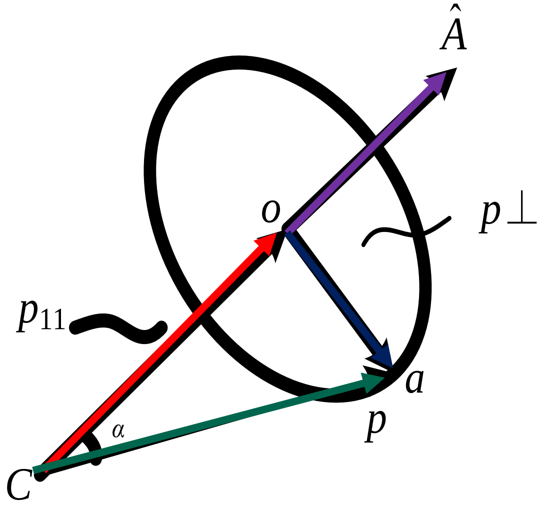
Task Space Scheme – Problem Definition
Orientation Problem - Equivalent Angle – Axis Representation

- Decompose the vector *P* into two components that are:
  - Parallel to *A*

$$co = P_{\parallel A} = (P \cdot A)A = P\cos\alpha$$

  - Perpendicular to *A*

$$oa = P_{\perp A} = P - (P \cdot A)A = P \cdot \sin\alpha$$

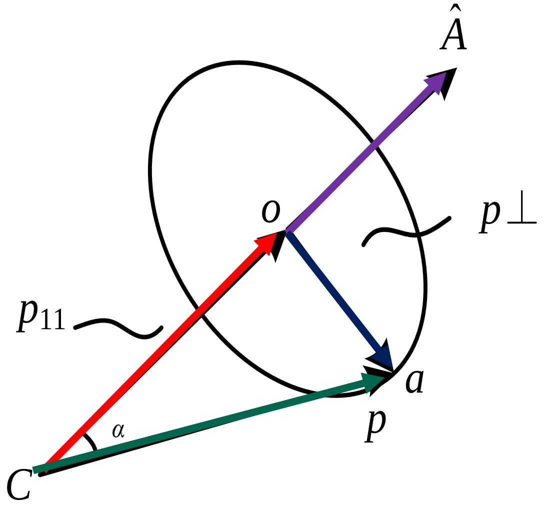


Task Space Scheme – Problem Definition
Orientation Problem - Equivalent Angle – Axis Representation

- Scalar multiplied by A
- Vector along A with a magnitude of the projection of P an A
- Note: A is a unite vector

$$\vec{P}_{\|A} = (\vec{P} \cdot \vec{A})\ \vec{A}$$

Dot product PA cosα (scallar)
Projection of vector P on vector A



$$\vec{P} = \vec{P}_{\perp A} + \vec{P}_{\|A}$$

$$\vec{P}_{\perp A} = \vec{P} - P_{\|A} = \vec{P} - (\vec{P} \cdot \vec{A})\vec{A} = \|P\| \sin\alpha$$

Task Space Scheme – Problem Definition
Orientation Problem - Equivalent Angle – Axis Representation

- The cross product $A \times P$ creates a vector that is perpendicular to the plane COA (including the two vectors A and P) therefor by definition

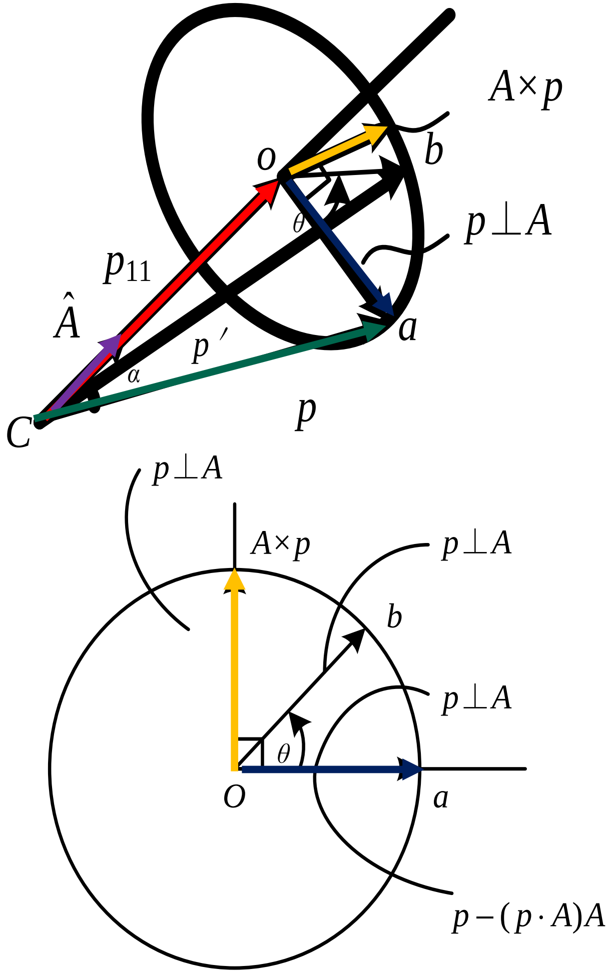
$$|A \times P| = P \sin \alpha$$

- As indicated before magnitude of $|oa| = |P_{\perp A}|$

$$|oa| = |P_{\perp A}| = P \sin \alpha$$

- As a result we are allowed to equate the two terms

$$|P_{\perp A}| = |A \times P|$$



Task Space Scheme – Problem Definition
Orientation Problem - Equivalent Angle – Axis Representation

- Express the rotation of $P_{\perp A}$ through an angle θ as

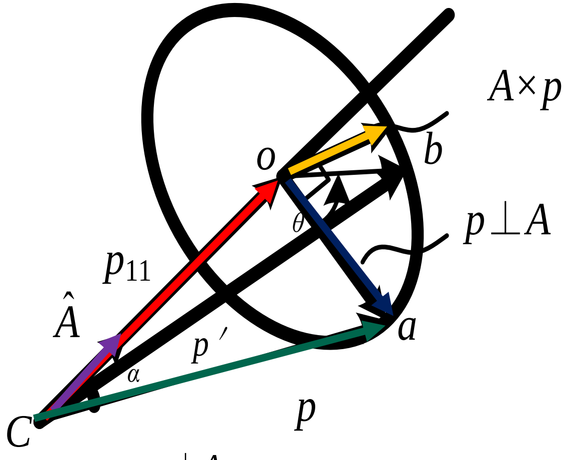
$$P_{\perp A} = \boxed{P_{\perp A}} \cos\theta + \boxed{P_{\perp A}} \sin\theta$$

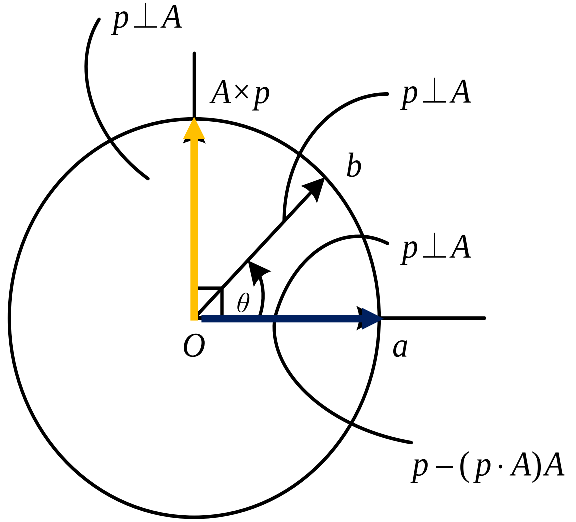
- Base on the two expressions

$$|P_{\perp A}| = |A \times P|$$
$$\vec{P}_{\perp A} = \vec{P} - P_{\parallel A} = \vec{P} - (\vec{P} \cdot \vec{A})\vec{A} = \|P\| \sin\alpha$$

- We can rewrite the expression

$$P_{\perp A} = ob = \boxed{[P - (P \cdot A)A]} \cos\theta + \boxed{(A \times P)} \sin\theta$$



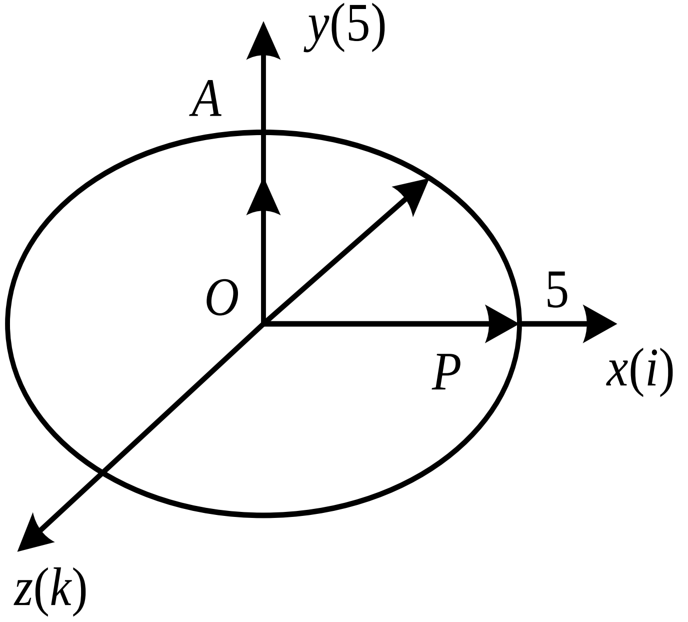


Task Space Scheme – Problem Definition
Orientation Problem - Equivalent Angle – Axis Representation

- Example: Multiplying a vector (*P*) by a unit vector (*A*) creates a vector that is perpendicular to the plane of vector (*A*) and (*P*) and has the some magnitude as (*P*)

$$A \times P = \begin{vmatrix} i & j & k \\ 0 & 1 & 0 \\ 5 & 0 & 0 \end{vmatrix} = 0i + 0j - 5k$$

$$|A||P| \sin\theta = 1 \cdot 5 \cdot \sin(90°) = 5$$



Task Space Scheme – Problem Definition
Orientation Problem - Equivalent Angle – Axis Representation

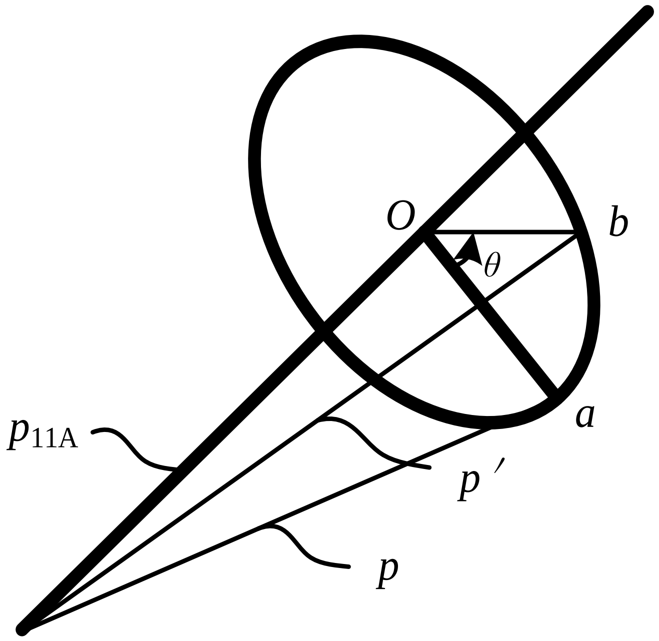
- The new vector P' which results from rotating vector P by A is expressed as

$$P' = ob + P_{\|A}$$

$$P' = \underbrace{[P - (P \cdot A)A]\cos\theta + (A \times P)\sin\theta}_{ob} + \underbrace{(P \cdot A)A}_{P_{\|A}}$$

- Rearranging this expression resulted in

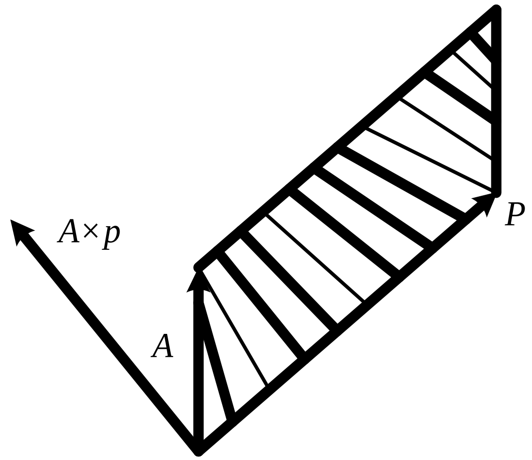
$$P' = P\cos\theta + (A \times P)\sin\theta + (P \cdot A)A[1 - \cos\theta]$$



Task Space Scheme – Problem Definition
Orientation Problem - Equivalent Angle – Axis Representation

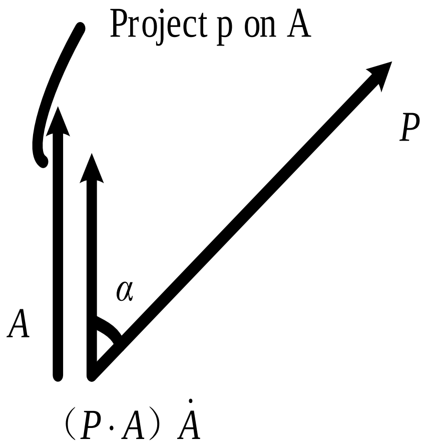
- Cross Product of vector A with P (Matrix form)

$$A \times P = \begin{bmatrix} 0 & -A_z & A_y \\ A_z & 0 & -A_x \\ -A_y & A_x & 0 \end{bmatrix} \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix}$$



- Projection of vector P on vector A (Matrix Form)

$$\mathit{Project}(P) = (P \cdot A)A = \frac{1}{\|A\|^2} \begin{bmatrix} A_x^2 & A_x A_y & A_x A_z \\ A_x A_y & A_y^2 & A_y A_z \\ A_x A_z & A_y A_z & A_z^3 \end{bmatrix} \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix}$$



- Note: If *A* is a unit vector $\|A\| = 1$

Task Space Scheme – Problem Definition
Orientation Problem - Equivalent Angle – Axis Representation

- Plugging the matrix definitions into the expression of

$$P' = P\cos\theta + (A \times P)\sin\theta + (P \cdot A)A(1-\cos\theta)$$

$$P' = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} P\cos\theta + \begin{bmatrix} 0 & -A_z & A_y \\ A_z & 0 & -A_x \\ -A_y & A_x & 0 \end{bmatrix} P\sin\theta + \begin{bmatrix} A_x^2 & A_xA_y & A_xA_z \\ A_xA_y & A_y^2 & A_yA_z \\ A_xA_z & A_yA_z & A_z^2 \end{bmatrix} P(1-\cos\theta)$$

- Setting $\begin{cases} c = \cos\theta \\ s = \sin\theta \end{cases}$

- Combing the terms gives us the formulation for matrix $R_A(\theta)$ that rotates a vector *P* by an angle θ about the axis *A*

Task Space Scheme – Problem Definition
Orientation Problem - Equivalent Angle – Axis Representation

$$P' = \underbrace{\left\{ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cos\theta + \begin{bmatrix} 0 & -A_z & A_y \\ A_z & 0 & -A_x \\ -A_y & A_x & 0 \end{bmatrix} \sin\theta + \begin{bmatrix} A_x^2 & A_xA_y & A_xA_z \\ A_xA_y & A_y^2 & A_yA_z \\ A_xA_z & A_yA_z & A_z^2 \end{bmatrix} (1-\cos\theta) \right\}}_{R_A(\theta)} P$$

$$P' = \underbrace{\begin{bmatrix} c+(1-c)A_x^2 & (1-c)A_xA_y - sA_z & (1-c)A_xA_z + SA_y \\ (1-c)A_xA_y + sA_z & c+(1-c)A_y^2 & (1-c)A_yA_z - SA_x \\ (1-c)A_xA_z - sA_y & (1-c)A_yA_z + SA_x & c+(1-c)A_z^2 \end{bmatrix}}_{R_A(\theta)} P$$

Task Space Scheme – Problem Definition
Orientation Problem - Equivalent Angle – Axis Representation

- Note: From the general rotation transformation around and arbitrary axis A we can obtain each one of the elementary rotation translation

$$R(x,\theta) = R\left(\overbrace{\begin{matrix} A_x & A_y & A_z \\ [1, & 0, & 0] \end{matrix}}^{\hat{A}}, \theta\right) = \begin{bmatrix} c + (1-c)1 & 0 & 0 \\ 0 & c & -s \\ 0 & s & c \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c & -s \\ 0 & s & c \end{bmatrix}$$

$$R(y,\theta) = R\left(\overbrace{(0,1,0)}^{\hat{A}}, \theta\right)$$

$$R(z,\theta) = R\left(\overbrace{(0,0,1)}^{\hat{A}}, \theta\right)$$

Task Space Scheme – Problem Definition
Orientation Problem - Equivalent Angle – Axis Representation

Transformation for a a matrix form to a vector/angle form

$$P' = R_A(\theta)P \longrightarrow P \xrightarrow{\theta,\hat{A}} P'$$

Task Space Scheme – Problem Definition
Orientation Problem - Equivalent Angle – Axis Representation

- Given any arbitrary rotation transformation $(R\ )$ we can use the eq. expressing $(R_A(\theta))$ to obtain an axis k about which an equivalent rotation θ by equating $(R\ )$ to $R(k,A)$

$$n_x = k_x^2 v\theta + c$$
$$o_y = k_y^2 v\theta + c$$
$$a_z = k_z^2 v\theta + c$$

$$\underbrace{\begin{bmatrix} n_x & o_x & a_x \\ n_y & o_y & a_y \\ n_z & o_z & a_z \end{bmatrix}}_{given} = \underbrace{\begin{bmatrix} k_x^2 v\theta + c & k_x k_y v\theta - k_z s & k_x k_z v\theta + k_y s \\ k_x k_y v\theta + k_z s & k_y^2 v\theta + c & k_y k_z v\theta - k_x s \\ k_x k_z v\theta - k_y s & k_y k_z v\theta + k_x s & k_z^2 v\theta + c \end{bmatrix}}_{\text{find } k,\theta}$$

Eq. 1

Task Space Scheme – Problem Definition
Orientation Problem - Equivalent Angle – Axis Representation

- Summing the diagonal terms of Eq. 1 we obtain

$$(1,1) + (2,2) + (3,3) \rightarrow$$

$$n_x + o_y + a_z = (k_x^2 v\theta + c) + \left(k_y^2 v\theta + c\right) + (k_z^2 v\theta + c)$$

$$n_x + o_y + a_z = \underbrace{\left(k_x^2 + k_y^2 + k_z^2\right)}_{1} \underbrace{v\theta}_{1-c} + 3c = 1 + 2c$$

$\hat{k}$ is a Unit Vector

- Solving for C i.e. the cosine of the angle of the rotation is resulted in

$$c = \cos\theta = \frac{1}{2}\left(n_x + o_y + a_z - 1\right)$$

Task Space Scheme – Problem Definition
Orientation Problem - Equivalent Angle – Axis Representation

- Differencing pairs of the off-diagonal terms in Eq. 1 we obtain

$$(3,2)-(2,3) \rightarrow o_z - a_y = 2k_x s \Rightarrow k_x = \frac{o_z - a_y}{2s}$$

$$(1,3)-(3,1) \rightarrow a_x - n_z = 2k_y S \Rightarrow k_y = \frac{a_x - n_z}{2s} \qquad \text{Eq. 2}$$

$$(2,1)-(1,2) \rightarrow n_y - o_x = 2k_z s \Rightarrow k_z = \frac{n_y - o_x}{2s}$$

- Squaring and adding the previous equations we obtain an expression for $\sin\theta$ that we will further refer to as Eq. 2

$$(o_z - a_y)^2 + (a_x - n_z)^2 + (n_y - o_x)^2 = 4\underbrace{(k_x^2 + k_y^2 + k_z^2)}_{1} s^2$$

$$s = \sin\theta = \pm\frac{1}{2}\sqrt{(o_z - a_y)^2 + (a_x - n_z)^2 + (n_y - o_x)^2} \qquad \text{Eq. 3}$$

Task Space Scheme – Problem Definition
Orientation Problem - Equivalent Angle – Axis Representation

- We may define the rotation to be positive about the vector $\hat{k}$ such that $0 < \theta < 180$.
- In this case the $+$ sign is appropriate in Eq. 3 and thus the angle of the rotation θ is uniquely define as

$$c = \cos\theta = \frac{1}{2}\left(n_x + o_y + a_z - 1\right)$$

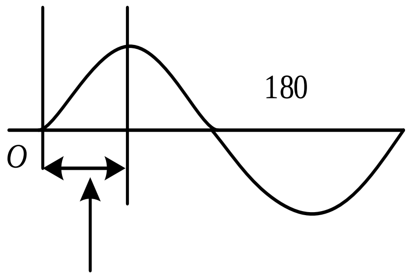
$$s = \sin\theta = \pm\frac{1}{2}\sqrt{\left(o_z - a_y\right)^2 + \left(a_x - n_z\right)^2 + \left(n_y - o_x\right)^2}$$

$$\tan\theta\Big|_{0<\theta<180} = \frac{\sqrt{\left(o_z - a_y\right)^2 + \left(a_x - n_z\right)^2 + \left(n_y - o_x\right)^2}}{n_x + o_y + a_z - 1}$$

Task Space Scheme – Problem Definition
Orientation Problem - Equivalent Angle – Axis Representation

- The component of *k* may be obtained from Eq. 2

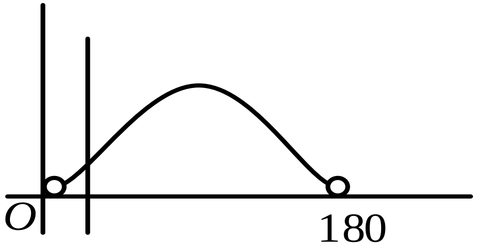
$$\left\{\begin{aligned} k_x &= \frac{o_z - a_y}{2s} \\ k_y &= \frac{a_x - n_z}{2s} \\ k_z &= \frac{n_y - o_x}{2s} \end{aligned}\right|_{\text{For } 0 < \theta < 90} \qquad \text{Eq. 4}$$



Task Space Scheme – Problem Definition
Orientation Problem - Equivalent Angle – Axis Representation

- **Pathological Case 1 – Normalizing K for $\boldsymbol{\theta}$→0 or $\boldsymbol{\theta}$→180**
  - When the angle of rotation (*A*) approaches θ→0 or θ→180 the axis of rotation is physically not well defined due to the small magnitude of both the numerator and the dominator in Eq. 4
  - The vector $\hat{k}$ should be renormalized to ensure that $|k| = 1$

- If θ→0; θ→180



- Then

$$\|k\| = \sqrt{k_x^2 + k_y^2 + k_z^2}$$

$$\hat{k} = \frac{k_x}{\|k\|}; \frac{k_y}{\|k\|}; \frac{k_z}{\|k\|}$$

Task Space Scheme – Problem Definition
Orientation Problem - Equivalent Angle – Axis Representation

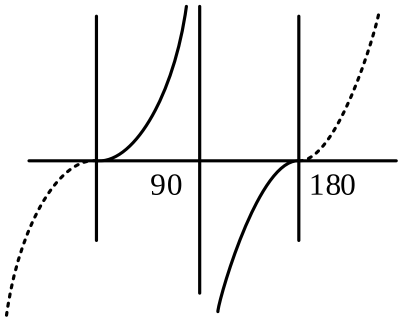
- **Pathological Case 2 – Singularity at $\boldsymbol{\theta}$=0 or $\boldsymbol{\theta}$=180**
  - At θ=0 or θ=*180* Eq. 4 are taking the form of $\frac{0}{0}$ yielding no information at all about a physical defection vector *k*
  - If θ=0 θ=180
  - Then $\hat{k} = \left[\frac{0}{0}; \frac{0}{0}; \frac{0}{0}\right]$

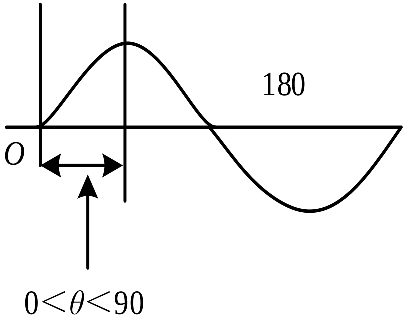
    $\uparrow$

    Undefined
  - Resulting in Singularity

Task Space Scheme – Problem Definition
Orientation Problem - Equivalent Angle – Axis Representation

- If the angle of rotation is greater than 0, $90 < \theta < 180$, than we must follow a different approach in determining *k* *otherwise we will get the same values since the sine has the same value in both regines*.





- Equating the diagonal elements of Eq 1

$$n_x = k_x^2 v\theta + c = k_x^2(1-c) + c$$
$$o_y = k_y^2 v\theta + c = k_y^2(1-c) + c$$
$$a_z = k_z^2 v\theta + c = k_z^2(1-c) + c$$

- Solving for k_x, k_y, k_z

Task Space Scheme – Problem Definition
Orientation Problem - Equivalent Angle – Axis Representation

- resulting in

$$\left\{\begin{aligned} k_x &= \pm\sqrt{\frac{n_x - \cos\theta}{1-\cos\theta}} \\ k_y &= \pm\sqrt{\frac{o_y - \cos\theta}{1-\cos A}} \\ k_z &= \pm\sqrt{\frac{a_z - \cos\theta}{1-\cos\theta}} \end{aligned}\right|_{\text{For } 90<\theta<180} \qquad \text{Eq. 5}$$

What should be the sign

Task Space Scheme – Problem Definition
Orientation Problem - Equivalent Angle – Axis Representation

- The largest component of *k* defined by equation Eq. 5 corresponds to the most positive components of n_x, o_y, a_z.

- For this largest element, the sign of the radical can be obtained from Eq. 2

- As the sine of the angle of rotation θ must be positive, then the sign of the component of *k* defined by Eq. 2 must be the same as the left hand side of these equations.

- Thus we may combined Eq. 5 with the information contained in Eq. 2 as follows

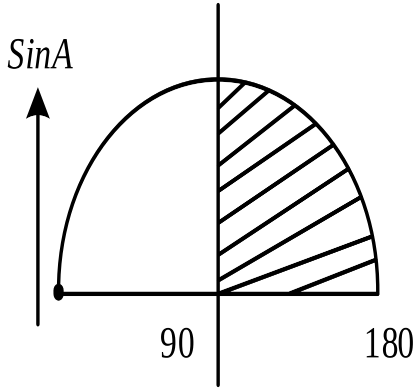
Task Space Scheme – Problem Definition
Orientation Problem - Equivalent Angle – Axis Representation

- Since $\theta \rightarrow 90 \rightarrow 180$ The sine function is always positive $\sin\theta > 0$

$$k_x = \frac{o_z - a_y}{2s} \quad \operatorname{sgn} k_x = \operatorname{sgn}(o_z - a_y)$$

$$k_y = \frac{a_x - n_z}{2s} \quad \operatorname{sgn} k_y = \operatorname{sgn}(a_x - n_z)$$

$$k_z = \frac{n_y - o_x}{2s} \quad \operatorname{sgn} k_z = \operatorname{sgn}(n_y - o_x)$$



- Rewriting Eq. 5

Fine the largest component of K

$$\begin{cases} k_x = \operatorname{sgn}(o_z - a_y)\sqrt{\dfrac{n_x - \cos\theta}{1 - \cos\theta}} \\ k_y = \operatorname{sgn}(a_x - n_z)\sqrt{\dfrac{o_y - \cos\theta}{1 - \cos\theta}} \\ k_z = \operatorname{sgn}(n_y - o_x)\sqrt{\dfrac{a_z - \cos\theta}{1 - \cos\theta}} \end{cases} \quad \begin{matrix} \operatorname{sgn}(e) = +1 \\ e > 0 \\ \operatorname{sgn}(e) = -1 \\ e < 0 \end{matrix}$$

Eq. 6

Task Space Scheme – Problem Definition
Orientation Problem - Equivalent Angle – Axis Representation

- Only the longest element of k is determined from Eq .6 corresponding to the most positive element of n_x, o_y, and a_z.

- The remaining element are more accurately determined by the following equations formed by summing pairs of off-diagonal element of Eq. 1

$$\begin{cases} n_y + o_x = 2k_x k_y v\theta & \text{Eq. 71} \\ o_z + a_y = 2k_y k_z v\theta & \text{Eq. 72} \\ n_z + a_x = 2k_z k_x v\theta & \text{Eq. 73} \end{cases}$$

Task Space Scheme – Problem Definition
Orientation Problem - Equivalent Angle – Axis Representation

- If $\boxed{k_x}$ is the largest $\begin{cases} k_y = \dfrac{n_y + o_x}{2k_x v\theta} & \text{from Eq 71} \\ k_z = \dfrac{a_x + n_z}{2k_x v\theta} & \text{from Eq 73} \end{cases}$

- If $\boxed{k_y}$ is the largest $\begin{cases} k_x = \dfrac{n_y + o_x}{2k_y v\theta} & \text{from Eq 71} \\ k_z = \dfrac{o_z + a_y}{2k_y v\theta} & \text{from Eq 72} \end{cases}$

- If $\boxed{k_z}$ is the largest $\begin{cases} k_x = \dfrac{a_x + n_z}{2k_z v\theta} & \text{from Eq 73} \\ k_y = \dfrac{o_z + a_y}{2k_z v\theta} & \text{from Eq 72} \end{cases}$

Task Space Scheme – Problem Definition
Orientation Problem - Equivalent Angle – Axis Representation

- Example: Determine the equivalent axis $(\hat{k})$ and angle (θ) of the following rotation matrix

$$\mathrm{Rot}(y,90)\mathrm{Rot}(z,90) = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Step 1 - Determine $\cos(\theta)$ and the $\sin(\theta)$

$$c = \cos\theta = \frac{1}{2}\left(n_x + o_y + a_z - 1\right) = \frac{1}{2}(0+0+0-1) = -\frac{1}{2}$$

$$s = \sin\theta = \pm\frac{1}{2}\sqrt{\left(o_z - a_y\right)^2 + (a_x - n_z)^2 + \left(n_y - o_x\right)^2} = \pm\frac{1}{2}\sqrt{(1-0)^2 + (1-0)^2 + (1-0)^2} = \frac{\sqrt{3}}{2}$$

$$\theta = tan^{-1}\left(\frac{\sqrt{3}/2}{-1/2}\right) = 120°$$

- **Step 2** - As *θ*>90, we determine the largest component of *k* corresponding to the largest element or the diagonal. As all the diagonal elements are equal in this example we may pick anyone of them

- For the purpose of this example we will pick k_x given in eq

$$k_x = \operatorname{sgn}(\theta_z - a_y)\sqrt{\frac{n_x - \cos\theta}{1 - \cos\theta}}$$

$$k_x = +\sqrt{\frac{0 + 0.5}{1 + 0.5}} = \frac{1}{\sqrt{3}}$$

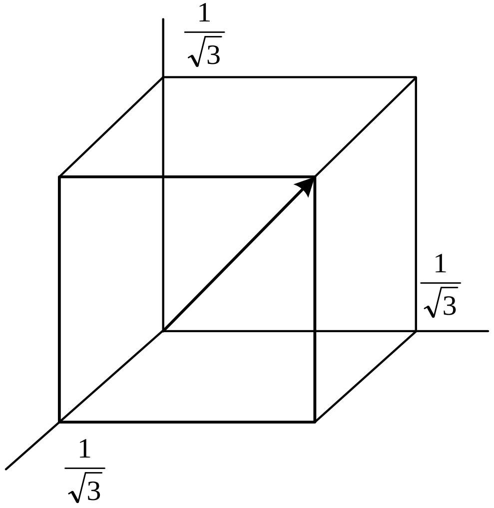
Task Space Scheme – Problem Definition
Orientation Problem - Equivalent Angle – Axis Representation

- **Step 3** - Since we have selected k_x to be than largest we may determine k_y and k_z respectively

$$k_y = \frac{n_y + o_x}{2k_x \underset{\substack{\uparrow \\ (1-c)}}{v\theta}} = \frac{1+0}{2\frac{1}{\sqrt{3}}(1+0.5)} = \frac{1}{\frac{3}{\sqrt{3}}} = \frac{1}{\sqrt{3}}$$

$$k_z = \frac{a_x + n_z}{2k_y v\theta} = \frac{1+0}{2\frac{1}{\sqrt{3}}(1+0.5)} = \frac{1}{\frac{3}{\sqrt{3}}} = \frac{1}{\sqrt{3}}$$

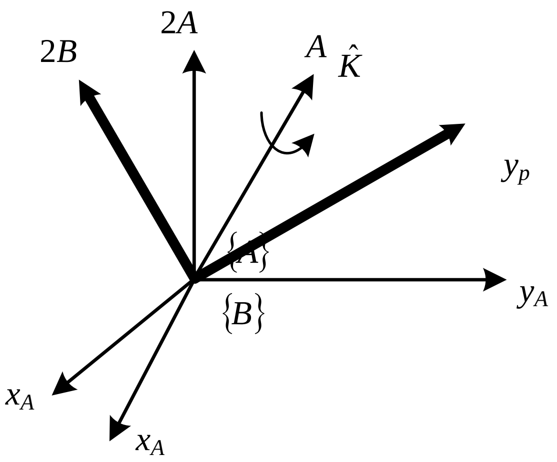
Task Space Scheme – Problem Definition
Orientation Problem - Equivalent Angle – Axis Representation



- In summary then

$$\mathrm{Rot}(y,90)\mathrm{Rot}(z,90) = R\left(\overbrace{\left[\frac{1}{\sqrt{3}}i + \frac{1}{\sqrt{3}}j + \frac{1}{\sqrt{3}}k\right]}^{k}, \overset{\theta}{\overset{\downarrow}{120}}\right)$$

Task Space Scheme – Problem Definition
Orientation Problem - Equivalent Angle – Axis Representation



- ① Start with the frame {B} coincident with known from {A}
- ② Rotate {B} about the vector $\hat{k}$ by an angle (θ) according to the right rule (note $\hat{k}$ is a unit vector
- $\sqrt{k_x^2 + k_y^2 + k_z^2} = 1$)

Equivalent Angle – Axis Representation – Summary

- **Conversion 1** - Conversion for single angle axis representation to rotation matrix representation

$$\left[P \xrightarrow{\theta,\hat{K}} P'\right] \longrightarrow P' = R_K(\theta)P$$

$$R_K(\theta) = \begin{bmatrix} k_x k_x v\theta + c\theta & k_x k_y v\theta - k_z s\theta & k_x k_z v\theta + k_y s\theta \\ k_x k_y v\theta + k_z s\theta & k_y k_y v\theta + c\theta & k_y k_z v\theta - k_x s\theta \\ k_x k_z v\theta - k_y s\theta & k_y k_z v\theta - k_x s\theta & k_z k_z v\theta + c\theta \end{bmatrix}$$

$$c\theta = \cos\theta$$
$$s\theta = \sin\theta$$
$$v\theta = 1 - \cos\theta$$

Equivalent Angle – Axis Representation – Summary

- **Conversion 2** – Compute ${}^{A}\hat{K}$ and θ given a rotation a matrix

$$P' = R_K(\theta)P \longrightarrow P \xrightarrow{\theta,\hat{K}} P'$$

$$\underbrace{\begin{bmatrix} n_x & o_x & a_x \\ n_y & o_y & a_y \\ n_z & o_z & a_z \end{bmatrix}}_{given}$$

↓

$$\tan\theta\Big|_{0<\theta<180} = \frac{\sqrt{(o_z - a_y)^2 + (a_x - n_z)^2 + (n_y - o_x)^2}}{n_x + o_y + a_z - 1}$$

$90 < \theta < 180$ ← → $0 < \theta < 90$

Equivalent Angle – Axis Representation – Summary

$90 < \theta < 180$

$$\begin{cases} k_x = \operatorname{sgn}(o_z - a_y)\sqrt{\dfrac{n_x - \cos\theta}{1-\cos\theta}} \\ k_y = \operatorname{sgn}(a_x - n_z)\sqrt{\dfrac{o_y - \cos\theta}{1-\cos\theta}} \\ k_z = \operatorname{sgn}(n_y - o_x)\sqrt{\dfrac{a_z - \cos\theta}{1-\cos\theta}} \end{cases} \quad \begin{matrix} \operatorname{sgn}(e) = +1 \\ e > 0 \\ \operatorname{sgn}(e) = -1 \\ e < 0 \end{matrix}$$

$0 < \theta < 90$

$$\begin{cases} k_x = \dfrac{o_z - a_y}{2s} \\ k_y = \dfrac{a_x - n_z}{2s} \\ k_z = \dfrac{n_y - o_x}{2s} \end{cases} \quad \text{For } 0 < \theta < 90$$

$$If\ k_x\ is\ the\ largest \begin{cases} k_y = \dfrac{n_y + o_x}{2k_x v\theta} \\ k_z = \dfrac{a_x + n_z}{2k_x v\theta} \end{cases}$$

$$If\ k_y\ is\ the\ largest \begin{cases} k_x = \dfrac{n_y + o_x}{2k_y v\theta} \\ k_z = \dfrac{o_z + a_y}{2k_y v\theta} \end{cases}$$

$$If\ k_z\ is\ the\ largest \begin{cases} k_x = \dfrac{a_x + n_z}{2k_z v\theta} \\ k_y = \dfrac{o_z + a_y}{2k_z v\theta} \end{cases}$$