Chapter 4 – Differential Relationship

4.1 Linear and Rotational Velocity

4.2 Motion of the Links of a Robot

4.3 $dx = Jdq$

4.4 Δ and ${}^{T}\Delta$

4.5 Manipulator Jacobian

4.6 Inverse Jacobian

4.7 Singularity

4.1 Linear and Rotational Velocity

- Analyze the relationship of

$$dx = Jdq$$
$$J^{-1}dx = dq$$

- Linear velocity:
  - Let Q be a vector, to specify its linear velocity, we need to select the reference frame. If a frame {B} is chosen, then:

$$\frac{d\,{}^{B}Q}{dt} = \lim_{\Delta t \to 0} \frac{{}^{B}Q(t+\Delta t) - {}^{B}Q(t)}{\Delta t} = {}^{B}V_{Q}$$

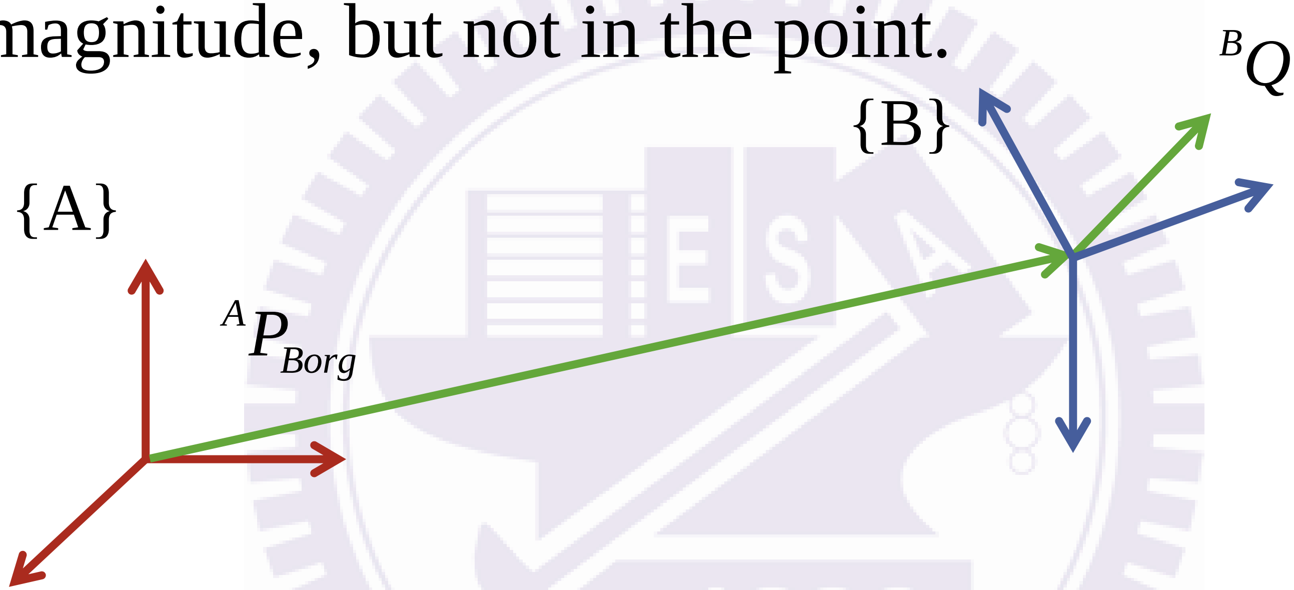
  - When expressed in the frame {A}, then:

$${}^{A}({}^{B}V_{Q}) = \frac{{}^{A}d\,{}^{B}Q}{dt} = {}^{A}_{B}R\,{}^{B}V_{Q}$$

  - Where ${}^{A}_{B}R$ is the transformation from {A} to {B}

4.1 Linear and Rotational Velocity

- Note: that ${}^{B}V_{Q}$ is significant only in direction and magnitude, but not in the point.



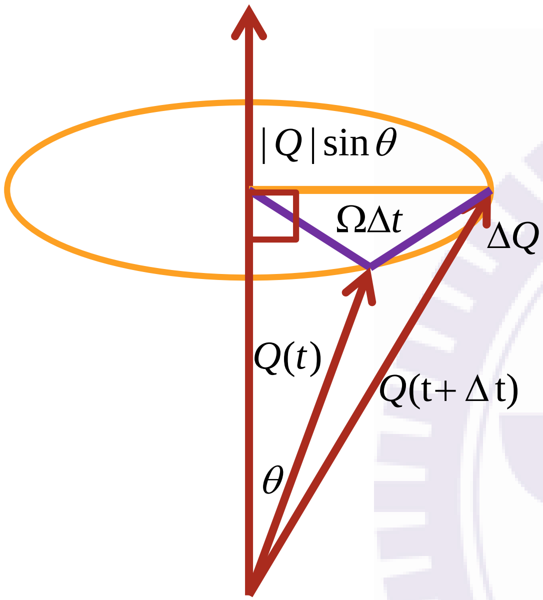
- Consider the movement of the origin of frame {B}

$${}^{A}V_{Q} = {}^{A}V_{Borg} + {}^{A}_{B}R\,{}^{B}V_{Q}$$

4.1 Linear and Rotational Velocity

- Rotational velocity:
  - Describe the rotational motion of a frame. Let ${}^{A}\Omega_{B}$ denote the rotation of frame {B} relative to {A}. ${}^{A}\Omega_{B}$ can be expressed in terms of different coordinates frames.
  - The question: How does a vector change with time as viewed from {A} when it is fixed in {B} and the systems are rotating?
  - Now the orientation of frame {B} with respect to frame {A} is changing in time, let the rotational velocity of {B} related to {A} is described by a vector called ${}^{A}\Omega_{B}$ when ${}^{B}Q$ is fixed in {B} and viewed from frame {B} (${}^{B}V_{Q} = 0$). However Q will have a velocity viewed from {A} due to ${}^{A}\Omega_{B}$.

4.1 Linear and Rotational Velocity



$$|\Delta Q| = \left(|{}^{A}Q|\sin\theta\right)\left(|{}^{A}\Omega_B \Delta t|\right)$$

$${}^{A}V_Q = {}^{A}\Omega_B \times {}^{A}Q$$

$$|v| = |r\omega|$$

Q may be changing with respect to frame {B}

$${}^{A}V_Q = {}^{A}({}^{B}V_Q) + {}^{A}\Omega_B \times {}^{A}Q$$

$$= {}^{A}R_B\,{}^{B}V_Q + {}^{A}\Omega_B \times ({}^{A}R_B\,{}^{B}Q)$$

Both linear and rotational

$${}^{A}V_Q = {}^{A}V_{Borg} + {}^{A}R_B\,{}^{B}V_Q + {}^{A}\Omega_B \times ({}^{A}R_B\,{}^{B}Q) - (*)$$

4.2 Motion of the Links of a Robot

- Angular velocity of link $i+1$ with respect to coordinate of link i is about the Z_i axis ($\theta_i + 1$)

$$W_s = \begin{cases} \vec{Z}_i \dot{q}_{i+1} & \text{If link i+1 is rotational} \\ 0 & \text{If link i+1 is translational} \end{cases}$$

$$W_{i+1} = W_i + W_s = \begin{cases} W_i + \vec{Z}_i \dot{q}_{i+1} & \text{If link i+1 is rotational} \\ W_i & \text{If link i+1 is translational} \end{cases}$$

- Linear velocity
  - Define: $P^*_{i+1} = P_{i+1} - P_i$

$$V_{i+1} = \begin{cases} W_s \times P^*_{i+1} + W_i \times P^*_{i+1} + V_i, & i+1, \quad R \\ \vec{Z}_i \cdot \dot{q}_{i+1} + W_i \times P^*_{i+1} + V_i, & i+1, \quad P \end{cases}$$

$$= \begin{cases} W_{i+1} \times P^*_{i+1} + V_i, i+1, R \\ \vec{Z}_i \cdot \dot{q}_{i+1} + W_{i+1} \times P^*_{i+1} + V_i, i+1, P \end{cases}$$

4.3 $dx = Jdq$

$$dX = \begin{bmatrix} dx \\ dy \\ dz \\ \delta x \\ \delta y \\ \delta z \end{bmatrix} \qquad dq = \begin{bmatrix} d\theta_1 \\ d\theta_2 \\ \dots \\ \dots \\ \dots \\ d\theta_N \end{bmatrix}$$

$$X = f(q)$$

$$\frac{dX}{dt} = \frac{df(q)}{dt} = \frac{df(q)}{dq} \times \frac{dq}{dt}$$

$$\dot{X} = J * \dot{q} \qquad POS = BASE * T_N * TOOL$$

4.3 $dx = Jdq$

- Differential Translation and Rotation
  - ① With respect to base:

$$T + dT = Trans(\mathrm{dx}, \mathrm{dy}, \mathrm{dz}) * \mathrm{Rot}(\mathrm{k}, \mathrm{d}\,\theta) * \mathrm{T}$$

$$dT = [Trans(\mathrm{dx}, \mathrm{dy}, \mathrm{dz}) * \mathrm{Rot}(\mathrm{k}, \mathrm{d}\,\theta) - \mathrm{I}] * \mathrm{T} = \Delta * \mathrm{T}$$

For $Rot(k, d\theta)$

When $d\theta \to 0$ $\begin{cases} \cos d\theta \approx 1 \\ \sin d\theta \approx d\theta \\ Vers(d\theta) \approx 0 \end{cases}$

4.3 $dx = Jdq$

$$Rot(k,d\theta)=\begin{pmatrix}1 & -k_z d\theta & k_y d\theta & 0\\ k_z d\theta & 1 & -k_x d\theta & 0\\ -k_y d\theta & k_x d\theta & 1 & 0\\ 0 & 0 & 0 & 1\end{pmatrix}$$

$$\Delta=\begin{pmatrix}1 & 0 & 0 & dx\\ 0 & 1 & 0 & dy\\ 0 & 0 & 1 & dz\\ 0 & 0 & 0 & 1\end{pmatrix}\begin{pmatrix}1 & -k_z d\theta & k_y d\theta & 0\\ k_z d\theta & 1 & -k_x d\theta & 0\\ -k_y d\theta & k_x d\theta & 1 & 0\\ 0 & 0 & 0 & 1\end{pmatrix}-\begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{pmatrix}$$

$$\Delta=\begin{pmatrix}0 & -k_z d\theta & k_y d\theta & dx\\ k_z d\theta & 0 & -k_x d\theta & dy\\ -k_y d\theta & k_x d\theta & 0 & dz\\ 0 & 0 & 0 & 0\end{pmatrix}$$

4.3 $dx = Jdq$

- With respect to T

$$T + dT = T * Trans(dx, dy, dz) * Rot(k, d\theta)$$

$$dT = T * [Trans(dx, dy, dz) * Rot(k, d\theta) - I] = T * {}^{T}\Delta$$

$Rot(k, d\theta)$ can also be formulated as differential rotations δ_x, δ_y, δ_z around x, y, z axes

$$Rot(x,\delta x) \cong \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & -\delta x & 0 \\ 0 & \delta x & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad Rot(y,\delta y) \cong \begin{pmatrix} 1 & 0 & \delta y & 0 \\ 0 & 1 & 0 & 0 \\ -\delta y & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad Rot(z,\delta z) \cong \begin{bmatrix} 1 & -\delta z & 0 & 0 \\ \delta z & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Rot(x,\delta x) * Rot(y,\delta y) * Rot(z,\delta z) = \begin{pmatrix} 1 & -\delta z & \delta y & 0 \\ \delta z & 1 & -\delta x & 0 \\ -\delta y & \delta x & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

4.3 $dx = Jdq$

$$
{}^{T}\Delta = \begin{pmatrix} 0 & -\delta z & \delta y & dx \\ \delta z & 0 & -\delta x & dy \\ -\delta y & \delta x & 0 & dz \\ 0 & 0 & 0 & 0 \end{pmatrix}
$$

$$
\begin{cases} \delta x = k_x d\theta \\ \delta y = k_y d\theta \\ \delta z = k_z d\theta \end{cases}
$$

4.4 Δ and ${}^{T}\Delta$

$$dT = \Delta * T = T * {}^{T}\Delta$$

$${}^{T}\Delta = T^{-1} * \Delta * T$$

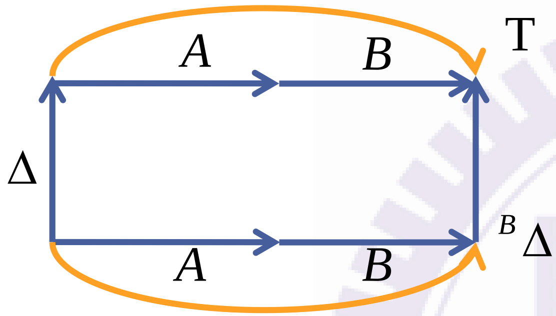
$$= \begin{pmatrix} n\cdot(\delta\times n) & n\cdot(\delta\times o) & n\cdot(\delta\times a) & n\cdot[(\delta\times P)+d] \\ o\cdot(\delta\times n) & o\cdot(\delta\times o) & o\cdot(\delta\times a) & o\cdot[(\delta\times P)+d] \\ a\cdot(\delta\times n) & a\cdot(\delta\times o) & a\cdot(\delta\times a) & a\cdot[(\delta\times P)+d] \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\text{From} \quad a\cdot(b\times c) = -b\cdot(a\times c) = b\cdot(c\times a)$$

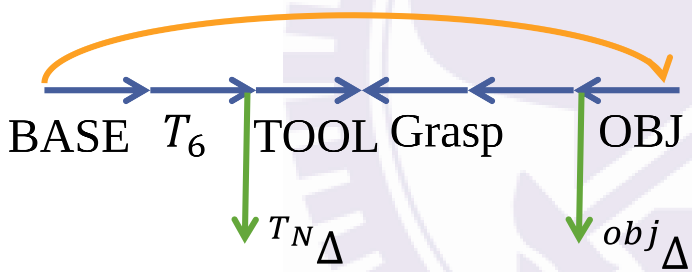
$$\Rightarrow \begin{pmatrix} 0 & -\delta\cdot a & \delta\cdot o & (\delta\times P)\cdot n + d\cdot n \\ \delta\cdot a & 0 & -\delta\cdot n & (\delta\times P)\cdot o + d\cdot o \\ -\delta\cdot o & \delta\cdot n & 0 & (\delta\times P)\cdot a + d\cdot a \\ 0 & 0 & 0 & 1 \end{pmatrix}, T_d = \begin{pmatrix} (\delta\times P+d)\cdot n \\ (\delta\times P+d)\cdot o \\ (\delta\times P+d)\cdot a \end{pmatrix}, T_\delta = \begin{pmatrix} \delta\cdot n \\ \delta\cdot o \\ \delta\cdot a \end{pmatrix}$$

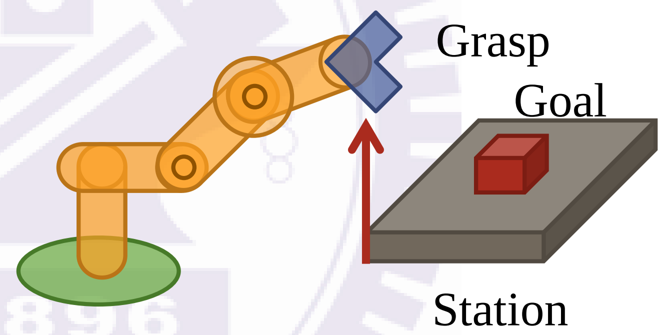
4.4 Δ and ${}^{T}\Delta$

- Assume T=A*B



$$ {}^{B}\Delta = T^{-1} * \Delta * T $$
$$ = (AB)^{-1} * \Delta * (AB) $$





Know ${}^{Obj}\Delta$ find ${}^{T6}\Delta$

$$ T = Grasp * TOOL^{-1} $$

$$ T = OBJ^{-1} * BASE * T_6 $$

$$ {}^{T_N}\Delta = T^{-1} \cdot {}^{obj}\Delta \cdot T $$

4.5 Manipulator Jacobian

$$T_N = A_1 * A_2 * \ldots * A_N$$

$$q_i = \begin{cases} \theta_i, \textit{ revolute} \\ d_i, \textit{ prismatic} \end{cases}$$

$$A_i = \begin{pmatrix} C\theta_i & -S\theta_i C\alpha_i & S\theta_i S\alpha_i & a_i C\theta_i \\ S\theta_i & C\theta_i C\alpha_i & -C\theta_i S\alpha_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

For revolute joint

$$\frac{dA_i}{d\theta_i} = \begin{pmatrix} -S\theta_i & -C\theta_i C\alpha_i & C\theta_i S\alpha_i & -a_i S\theta_i \\ C\theta_i & -S\theta_i C\alpha_i & S\theta_i S\alpha_i & a_i C\theta_i \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \boxed{A_i}$$

4.5 Manipulator Jacobian

For prismatic joint

$$\frac{dA_i}{dd_i} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \qquad dA_i = {}^{i-1}\Delta_i A_i dq_i = A_i\, {}^{i}\Delta_{i+1} dq_i$$

$$\begin{pmatrix} {}^{i-1}d_i \\ {}^{i-1}\delta_i \end{pmatrix} \qquad \begin{pmatrix} {}^{i}d_{i+1} \\ {}^{i}\delta_{i+1} \end{pmatrix}$$

For revolute joint, only (rotational error) around Z_{i-1} axis

$${}^{i-1}d_i = 0,\ {}^{i-1}\delta_i = \begin{pmatrix} 0 \\ 0 \\ \boxed{1} \end{pmatrix}$$

For prismatic joint, only (translational error) along Z_{i-1} axis

$${}^{i-1}\delta_i = 0,\ {}^{i-1}d_i = \begin{pmatrix} 0 \\ 0 \\ \boxed{1} \end{pmatrix}$$

4.5 Manipulator Jacobian

Due to errors in $dA_i, i = 1, 2, \dots, N$

$$T_N + dT_N = A_1 * A_2 * \dots * (A_i + dA_i) * \dots * A_N$$

$$\Downarrow$$

$$[A_i + {}^{i-1}\Delta_i \cdot A_i \cdot dq_i]$$

$$= T_N + (A_1 * \dots * A_{i-1} * {}^{i-1}\Delta_i * A_i * A_{i+1} * \dots * A_N) dq_i$$

$$dT_N = \boxed{A_1 * \dots * A_N} * \boxed{[A_i * \dots * A_N]^{-1} * {}^{i-1}\Delta_i * [A_i * \dots * A_N]} * dq_i$$

$$= \boxed{T_N} * \boxed{{}^{T_N}\Delta_i} * dq_i$$

$${}^{T_N}\Delta_i = [A_i * \dots * A_N]^{-1} * {}^{i-1}\Delta_i * [A_i * \dots * A_N]$$

$$= U_{i-1}^{-1} * {}^{i-1}\Delta_i * U_{i-1}$$

Where $U_{i-1} = A_i * \dots * A_N$

4.5 Manipulator Jacobian

For revolute joint

$$^{i-1}d_i = 0,\ ^{i-1}\delta_i = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

$$^{T_N}d_i = \begin{pmatrix} n \cdot [(^{i-1}\delta_i \times P) + {}^{i-1}d_i] \\ o \cdot [(^{i-1}\delta_i \times P) + {}^{i-1}d_i] \\ a \cdot [(^{i-1}\delta_i \times P) + {}^{i-1}d_i] \end{pmatrix} = \begin{pmatrix} P_x n_y - n_x P_y \\ P_x o_y - o_x P_y \\ P_x a_y - a_x P_y \end{pmatrix}$$

Where n, o, a, p are of U_{i-1}

$$^{T_N}\delta_i = \begin{pmatrix} n \cdot {}^{i-1}\delta_i \\ o \cdot {}^{i-1}\delta_i \\ a \cdot {}^{i-1}\delta_i \end{pmatrix} = \begin{pmatrix} n_z \\ o_z \\ a_z \end{pmatrix}$$

4.5 Manipulator Jacobian

For prismatic joint

$$
{}^{i-1}d_i = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \; {}^{i-1}\delta_i = 0 \qquad {}^{T_N}d_i = \begin{pmatrix} n_Z \\ o_Z \\ a_Z \end{pmatrix}, \; {}^{T_N}\delta_i = 0
$$

In total $\quad {}^{T_N}\Delta = \sum_{i=1}^{N} (U_{i-1}^{-1} * {}^{i-1}\Delta_i * U_{i-1})$

$$
\begin{bmatrix} {}^{T_N}d \\ {}^{T_N}\delta \end{bmatrix} = \begin{pmatrix} \vdots & \vdots & \vdots & \vdots \\ {}^{T_N}\Delta_1 & {}^{T_N}\Delta_2 & \cdots & {}^{T_N}\Delta_N \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix} \begin{pmatrix} dq_1 \\ dq_2 \\ \vdots \end{pmatrix}
$$

4.6 Inverse Jacobian

- Using $dq = J^{-1}dx$ to invert the J matrix directly is very inefficient. We need to consider the case when J^{-1} doesn't exist ($detJ = 0$, not full rank)

  ① R. Paul's method

  a) From ${}^{T6}d, {}^{T6}q$ compute $\mathrm{d}T_6 = T_6 * {}^{T6}\Delta$

$$= \begin{pmatrix} dn_6 & do_6 & da_6 & dp_6 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

4.6 Inverse Jacobian

b) Kinematic solution.

- Differentiate the solution given a value of T_6, the expressions for each differential changes are functions of dT_6 and those $d\theta's$ that are already obtained.

- Example: PUMA 560
  - For θ_1

$$-S_1P_x + C_1P_y = d_3$$

$$(-C_1P_x - S_1P_y)d\theta_1 - (S_1dP_x - C_1dP_y) = 0$$

$$d\theta_1 = -\frac{S_1dP_x - C_1dP_y}{C_1P_x + S_1P_y}$$

  - Similarly $d\theta_2 \ldots d\theta_6$ can be obtained

4.6 Inverse Jacobian

② Method frequently used in the industries in the past

a) Compute dT_6

b) Compute $T_6^N = T_6 + dT_6 \Rightarrow T^T\Delta = \frac{\partial T\Delta}{\partial q}\partial q$

c) Compute new joint solution from T_6^N

d) $dq = q^N - q$

③ Neural Network Approach

④ Fuzzy Approach

⑤ "An efficient Solution of Differential Inverse Kinematics Problem for Wrist-Partitioned Robots" IEEE Trans. Rob. & Auto. Vol. 6(1) PP. 117-123, 1990

$$J^{wrist} = \begin{pmatrix} J_1 & 0 \\ J_2 & J_3 \end{pmatrix} \qquad J^{-1} = [J_1]^{-1}[J_3]^{-1}$$

4.7 Singularity

- $|J| = 0 \Rightarrow dq = J^{-1}dx$ can not be obtained
- 3 cases for PUNA560:
  1. Links 2 and 3 are fully extended.
  2. Joint 5 at its zero position.
  3. The end of link 3 aligns with the axis of robot trunk.