Subject: Elements of Robots and Kinematics (BE04041011-GTU)

Topic: Introduction to Differential Motions and Velocities

Branch: Robotics & Automation Engineering (2nd Year/4th Semester Students)

Prepared by: Dr. Nilesh Pancholi *(www.nileshpancholi.com)*

Differential Motions

Differential motions describe infinitesimal changes in the position and orientation of a robot end-effector, coordinate frame, or any rigid body. Rather than analysing large, discrete movements (e.g., jumping from point A to B), we focus on local, continuous motion at a specific instant. This approach leverages calculus to model robot behaviour smoothly and predictably.

- Core Idea: Imagine zooming into a robot's trajectory at an instant—the motion looks like a tiny straight-line translation plus a small rotation. These "differential" elements are denoted as $\delta\mathbf{p}$ (position change) and $\delta\boldsymbol{\theta}$ (orientation change), where δ means "infinitesimally small."
- Why It Matters: Real-world robots can't make infinite jumps; they follow smooth paths. *Differential motions enable precise control, error minimization, and integration with sensors.*

Examples:

- A robot hand in precision assembly (e.g., inserting a steel coil pin in manufacturing) shifts by $\delta x = 0.01$ mm forward.
- A camera on an inspection robot rotates by $\delta\theta = 0.1°$ to track a weld seam without blurring.

Position in 3D space: $\mathbf{p}(t) = [x(t), y(t), z(t)]^T$.

Differential position: $d\mathbf{p} = \begin{bmatrix} dx \\ dy \\ dz \end{bmatrix} = \mathbf{v}\, dt$, where dt is infinitesimal time.

Linear and Angular Velocities

These are the time derivatives of differential motions, quantifying how fast position and orientation change.

- Linear Velocity ($\mathbf{v}$): Rate of change of position, $\mathbf{v} = \frac{d\mathbf{p}}{dt} = \lim_{\Delta t \to 0} \frac{\Delta \mathbf{p}}{\Delta t}$.
  Units: m/s. Direction aligns with motion; magnitude is speed.
  Example: A welding robot torch moves at $\mathbf{v} = [0.5, 0, 0]^T$ m/s along x-axis for a straight bead.
- Angular Velocity ($\boldsymbol{\omega}$): Rate of change of orientation, expressed in the body frame or world frame.
  For small rotations, $d\boldsymbol{\theta} = \boldsymbol{\omega}\, dt$.
  Units: rad/s. Uses axis-angle representation: $\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]^T$.
  Example: Robot gripper twists at $\boldsymbol{\omega} = [0, 0, 2]^T$ rad/s around z-axis during part alignment.

Relating both;
The velocity screw combines both: $\mathbf{V} = \begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix}$. This unifies motion in robotics kinematics.

Practical Example: Robot Welding Torch

In steel coil production (e.g., Steckel mill operations), a welding torch follows a curved seam. Large motions would cause defects; instead, approximate as infinite tiny steps:

- At time t, $\mathbf{v}$ keeps it tangent to the path.
- $\boldsymbol{\omega}$ adjusts orientation to maintain perpendicular contact.
  Sudden jumps? Approximate as summed differentials: $\Delta\mathbf{p} \approx \int \mathbf{v}\, dt$.

Illustration: Hand-Drawn Diagram of Robot Arm

```text
World Frame
     ^
     | z
     |     Robot Base
     |       /|
     |      / |  End-Effector (Torch)
     |     /  |
     |    /___| <-- Joint 2
     |   /
     |  /  <-- Joint 1
     | /
     o--------> x (trajectory)
    / y

Small arrows along path: → δp (linear), ↻ δθ (angular)
Zoomed inset: v (straight arrow), ω (curved arrow around axis)
```

Small arrows along path: → δp (linear), ↻ δθ (angular)

Zoomed inset: v (straight arrow), ω (curved arrow around axis)

Example 1: If a robot moves with $\mathbf{v} = [1, 0, 0]^T$ m/s and $\boldsymbol{\omega} = [0, 0.5, 0]^T$ rad/s for $\Delta t = 0.1$ s, approximate $\Delta\mathbf{p}$ and $\Delta\boldsymbol{\theta}$.

Using first-order (small-motion) approximation:

Given
$\mathbf{v} = [1, 0, 0]^T$m/s
$\boldsymbol{\omega} = [0, 0.5, 0]^T$rad/s
$\Delta t = 0.1$s

Linear displacement

$$\Delta\mathbf{p} \approx \mathbf{v}\,\Delta t = [1,0,0]^T \times 0.1 = [0.1, 0, 0]^T \text{ m}$$

Angular displacement

$$\Delta\boldsymbol{\theta} \approx \boldsymbol{\omega}\,\Delta t = [0,0.5,0]^T \times 0.1 = [0, 0.05, 0]^T \text{ rad}$$

✅ **Final Answer**

$$\boxed{\Delta\mathbf{p} = [0.1, 0, 0]^T \text{ m}, \Delta\boldsymbol{\theta} = [0, 0.05, 0]^T \text{ rad}}$$

MATBAB Code [approximation of Δp and Δθ]

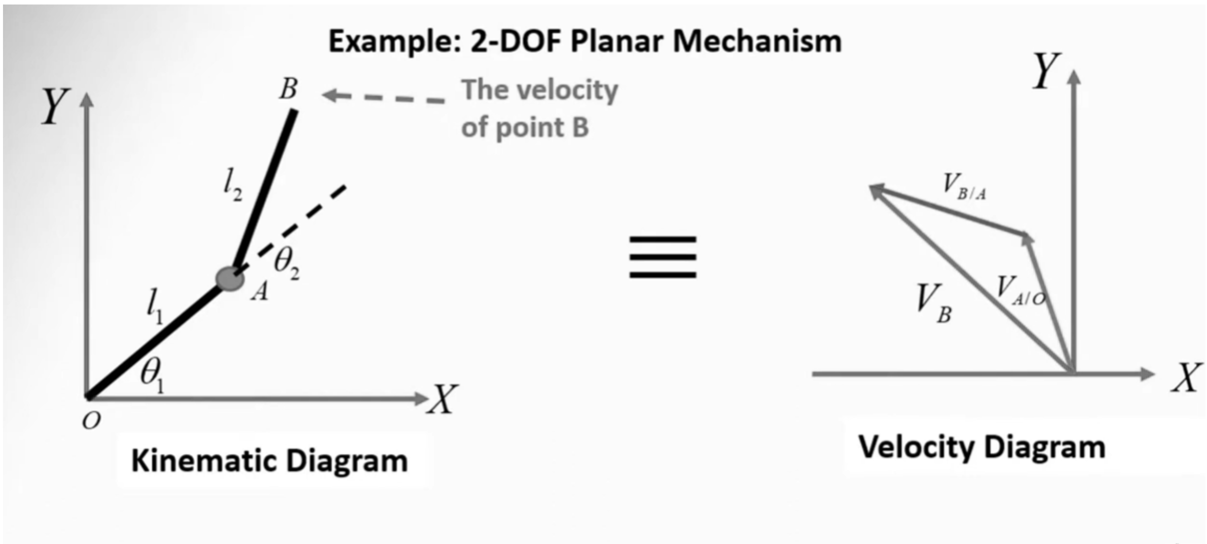
```
% Differential motion calculation
% Given linear and angular velocity
v = [1; 0; 0];       % Linear velocity (m/s)
omega = [0; 0.5; 0]; % Angular velocity (rad/s)
dt = 0.1;            % Time interval (s)
% Differential position
delta_p = v * dt;
% Differential orientation
delta_theta = omega * dt;
% Display results
disp('Differential Position (m):');
disp(delta_p);
disp('Differential Orientation (rad):');
disp(delta_theta);
```

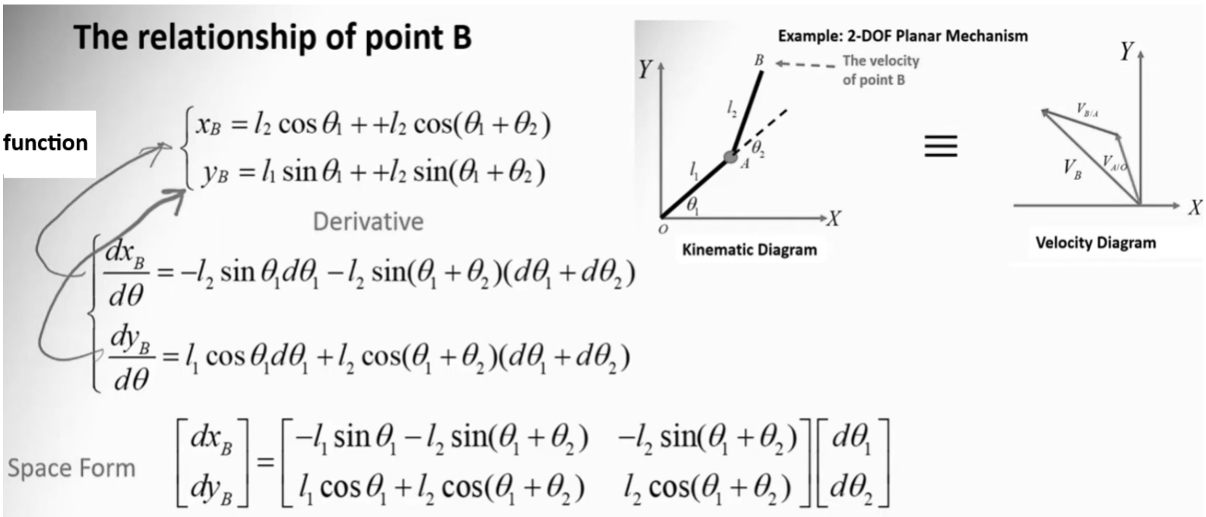
Expected Output

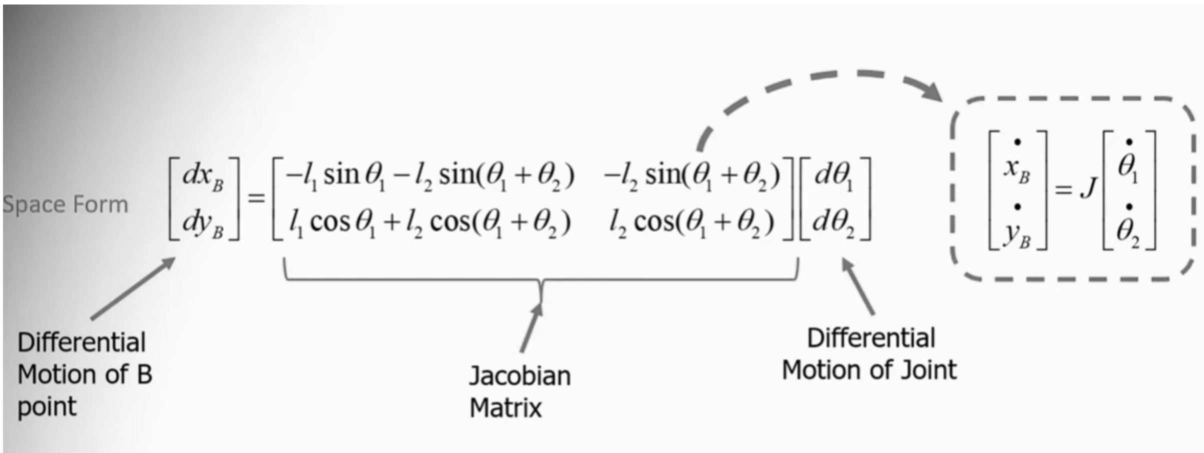
```
Differential Position (m):
  0.1000
       0
       0
Differential Orientation (rad):
       0
  0.0500
       0
```

Two Degree of Freedom Planar Mechanism

Figure shows the kinematic and velocity diagram and relative new position of end effector point – B considering trigonometry







Where; The **Jacobian (J)** relates joint velocities to end-effector velocities:

$v = J(q) \cdot \dot{q}$

Where:

- v = [linear velocity; angular velocity]
- $\dot{q}$ = joint velocity vector

Each column of the Jacobian represents the effect of **one joint** on the hand motion.

Illustration idea: Show each joint contributing a velocity arrow at end-effector.

Practical Example:

- Robot control
- Trajectory planning
- Singularity analysis

Types of Differential motions

Differential motions of robot frame are;

1. Differential translation
2. Differential rotation
3. Differential transformations (translation and rotation)

1 Translation

Translation moves a point or object:

- Without changing shape, size and orientation; only the position changes.

If a point P(x, y) is translated by:

- tx units in X-direction
- ty units in Y-direction

✅ New coordinates:

$$x' = x + t_x$$
$$y' = y + t_y$$

◆ Matrix Form (Basic CAD Concept)

$$[x' \quad y' \quad 1] = [x \quad y \quad 1]\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$$

Example: Translate point (2, 3) by (4, −1)

Solution:

$$x' = 2 + 4 = 6$$
$$y' = 3 - 1 = 2$$

➡ New point (6, 2)

2 Rotation

Rotation turns a point or object:

- About a fixed point (origin or any other point), by an angle θ, without changing shape or size; only the orientation changes.
- If a point **P(x, y)** is rotated by angle **θ** (counter-clockwise) about the origin:

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

Rotation Matrix (Very Important)

$$[x' \quad y' \quad 1] = [x \quad y \quad 1]\begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\,\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Example: Rotate point **(1, 0)** by **90° CCW**

Solution:

$$\cos 90° = 0, \quad \sin 90° = 1$$
$$x' = 1(0) - 0(1) = 0$$
$$y' = 1(1) + 0(0) = 1$$

➞ New point **(0, 1)**

» **Differential Translation**:

$$Trans(d_x, d_y, d_z)$$

» **Differential Rotations**

$$Rot(k, d\theta)$$

> Used approximation $sin\theta = \theta$ and $cos\theta = 1$

> Neglecting higher orders such as $(\theta_x)^2 \ll \theta_x$

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

» Example:

A Frame B has translated a differential amount of Trans(0.01,0.05,0.03) units Find its new location and orientation.

$$B = \begin{bmatrix} 0.5 & 0 & 0.6 & 5 \\ 0 & 0.3 & 0 & 4 \\ 0.707 & 0 & 1 & 10 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Solution:

» Example:

A Frame B has translated a differential amount of Trans(0.01,0.05,0.03) units Find its new location and orientation.

$$B = \begin{bmatrix} 0.5 & 0 & 0.6 & 5 \\ 0 & 0.3 & 0 & 4 \\ 0.707 & 0 & 1 & 10 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Solution:

$$B_{new} = Trans(B_{px}, B_{py}, B_{pz}) \times B$$

» Example:

A Frame B has translated a differential amount of Trans(0.01,0.05,0.03) units Find its new location and orientation.

$$B = \begin{bmatrix} 0.5 & 0 & 0.6 & 5 \\ 0 & 0.3 & 0 & 4 \\ 0.707 & 0 & 1 & 10 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Solution:

$$B_{new} = Trans(B_{px}, B_{py}, B_{pz}) \times B$$

$$\Rightarrow Trans(B_{px}, B_{py}, B_{pz}) = \begin{bmatrix} 1 & 0 & 0 & 0.01 \\ 0 & 1 & 0 & 0.05 \\ 0 & 0 & 1 & 0.03 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Solution (contd.):

$$B_{new} = \begin{bmatrix} 1 & 0 & 0 & 0.01 \\ 0 & 1 & 0 & 0.05 \\ 0 & 0 & 1 & 0.03 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.5 & 0 & 0.6 & 5 \\ 0 & 0.3 & 0 & 4 \\ 0.707 & 0 & 1 & 10 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Solution (contd.):

$$B_{new} = \begin{bmatrix} 1 & 0 & 0 & 0.01 \\ 0 & 1 & 0 & 0.05 \\ 0 & 0 & 1 & 0.03 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.5 & 0 & 0.6 & 5 \\ 0 & 0.3 & 0 & 4 \\ 0.707 & 0 & 1 & 10 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The new location of the frame

$$B_{new} = \begin{bmatrix} 0.5 & 0 & 0.6 & 5.01 \\ 0 & 0.3 & 0 & 4.05 \\ 0.707 & 0 & 1 & 9.03 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Coordinate

Example: A 2-DOF planar robot has two revolute joints with link lengths $l_1 = 1$ m, $l_2 = 1$ m.

At a certain instant:

- $\theta_1 = 30°$
- $\theta_2 = 45°$
- Joint velocities:

$$\dot{\theta}_1 = 0.5 \text{ rad/s}, \dot{\theta}_2 = 0.3 \text{ rad/s}$$

Find the linear velocity of the end-effector using Jacobian.

Solution:

Step 1: End-Effector Position Equation

For a 2R planar robot:

$$x = l_1 \cos\ \theta_1 + l_2 \cos\ (\theta_1 + \theta_2)$$
$$y = l_1 \sin\ \theta_1 + l_2 \sin\ (\theta_1 + \theta_2)$$

Step 2: Jacobian Matrix

$$\text{J} = \begin{bmatrix} \frac{\partial x}{\partial \theta_1} & \frac{\partial x}{\partial \theta_2} \\ \frac{\partial y}{\partial \theta_1} & \frac{\partial y}{\partial \theta_2} \end{bmatrix}$$

Partial derivatives:

$$\frac{\partial x}{\partial \theta_1} = -l_1 \sin\ \theta_1 - l_2 \sin\ (\theta_1 + \theta_2)$$
$$\frac{\partial x}{\partial \theta_2} = -l_2 \sin\ (\theta_1 + \theta_2)$$
$$\frac{\partial y}{\partial \theta_1} = l_1 \cos\ \theta_1 + l_2 \cos\ (\theta_1 + \theta_2)$$
$$\frac{\partial y}{\partial \theta_2} = l_2 \cos\ (\theta_1 + \theta_2)$$

Step 3: Substitute Values

$$\theta_1 = 30^\circ, \theta_1 + \theta_2 = 75^\circ$$
$$\sin\ 30^\circ = 0.5, \cos\ 30^\circ = 0.866$$
$$\sin\ 75^\circ = 0.966, \cos\ 75^\circ = 0.259$$

Jacobian becomes:

$$\mathrm{J} = \begin{bmatrix} -(1)(0.5) - (1)(0.966) & -(1)(0.966) \\ (1)(0.866) + (1)(0.259) & (1)(0.259) \end{bmatrix}$$
$$\mathrm{J} = \begin{bmatrix} -1.466 & -0.966 \\ 1.125 & 0.259 \end{bmatrix}$$

Step 4: Velocity Equation

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \mathrm{J} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$
$$= \begin{bmatrix} -1.466 & -0.966 \\ 1.125 & 0.259 \end{bmatrix} \begin{bmatrix} 0.5 \\ 0.3 \end{bmatrix}$$

Step 5: Final Calculation

$$\dot{x} = (-1.466)(0.5) + (-0.966)(0.3)$$
$$\dot{x} = -0.733 - 0.290 = \boxed{-1.023 \text{ m/s}}$$
$$\dot{y} = (1.125)(0.5) + (0.259)(0.3)$$
$$\dot{y} = 0.562 + 0.078 = \boxed{0.640 \text{ m/s}}$$

✅ Final Answer

$$\boxed{\mathrm{v}_{ee} = \begin{bmatrix} -1.023 \\ 0.640 \end{bmatrix} \text{ m/s}}$$

MATLAB Code [Jacobian & End-Effector Velocity (2-DOF Robot)]

```
clc;

clear;

% Given link lengths (meters)

l1 = 1;

l2 = 1;

% Joint angles (degrees)

theta1 = 30;

theta2 = 45;

% Convert degrees to radians
```

```
t1 = deg2rad(theta1);
t2 = deg2rad(theta2);
% Joint velocities (rad/s)
theta1_dot = 0.5;
theta2_dot = 0.3;
% Jacobian matrix for 2R planar robot
J = [ ...
    -l1*sin(t1) - l2*sin(t1+t2),   -l2*sin(t1+t2);
     l1*cos(t1) + l2*cos(t1+t2),    l2*cos(t1+t2)
    ];
% Joint velocity vector
q_dot = [theta1_dot; theta2_dot];
% End-effector linear velocity
v = J * q_dot;
% Display results
disp('Jacobian Matrix J =');
disp(J);
disp('End-Effector Linear Velocity [x_dot; y_dot] (m/s) =');
disp(v);
```

Expected Output (Approx.)

```
Jacobian Matrix J =
  -1.4660   -0.9660
   1.1250    0.2588
End-Effector Linear Velocity =
  -1.0230
   0.6400
```

⚠️ Physical Meaning of Singularity (2R Robot)

Singularity occurs when:

- $\theta_2 = 0°$or $180°$
- Both links are collinear
- End-effector cannot move in some directions

Differential Motion of a Frame about General Axis - K

"A rigid body frame can change its position in two ways: **translation** and **rotation**.

A *differential transformation* means a **very small change** in both."

So this process is about:

- tiny translation → dx, dy, dz
- tiny rotation → $d\theta$

 T→ original frame (position + orientation)

 dT→ very small change in the frame

 $T + dT$→ new frame after a tiny motion

<u>*Combined Effect: Translation + Rotation*</u>

This line is key:

$$[T + dT] = [\text{Trans}(dx, dy, dz)\ \text{Rot}(k, d\theta)]\ [T]$$

"To get the new frame, we first apply a small translation, then a small rotation, to the original frame.

From the equation:

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

I= identity matrix (no motion)

So:

- **Transformation – Identity** gives **only the change**

💡 Analogy "Just like displacement = final − initial position."

Differential Operator Δ

Now introduce this cleanly:

$$[dT] = [\Delta][T]$$

Where:

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

"Instead of writing long expressions again and again, we define a **differential operator Δ**."

So:

- Δcontains **only small motions**
- $T$is the original frame
- Output = **change in frame**

Matrix Form:

$$\Delta = \text{Trans}(dx, dy, dz) \times \text{Rot}(k, d\theta) - I$$

$$= \begin{bmatrix} 1 & 0 & 0 & dx \\ 0 & 1 & 0 & dy \\ 0 & 0 & 1 & dz \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & -\delta z & \delta y & 0 \\ \delta z & 1 & -\delta x & 0 \\ -\delta y & \delta x & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & -\delta z & \delta y & dx \\ \delta z & 0 & -\delta x & dy \\ -\delta y & \delta x & 0 & dz \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

→ Differential operator Δ is not a transformation matrix, nor a frame

Physical Meaning is listed as below:

- $dx, dy, dz \rightarrow$ linear displacement
- $d\theta \rightarrow$ angular displacement
- $\Delta \rightarrow$ motion generator
- $dT \rightarrow$ velocity-level motion of the frame

This is **not position**, this is **instantaneous motion**.

"When we divide this differential motion by time, it becomes **velocity**. That is why Jacobian maps joint rates to end-effector velocity."

Differential transformation represents an infinitesimal change in a frame due to small translation and small rotation, expressed using a differential operator.

» Example:

Find the total differential transformation of the robot rotational for about 3-axes: $\delta_x = 0.1, \delta_y = 0.05, \delta_z = 0.02$ radians

Solution:

$$Rot(k, d\theta) = \begin{bmatrix} 1 & -\delta z & \delta y & 0 \\ \delta z & 1 & -\delta x & 0 \\ -\delta y & \delta x & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

» Example:

Find the total differential transformation of the robot rotational for about 3-axes: $\delta_x = 0.1, \delta_y = 0.05, \delta_z = 0.02$ radians

Solution:

$$Rot(k, d\theta) = \begin{bmatrix} 1 & -0.02 & 0.05 & 0 \\ 0.02 & 1 & -0.1 & 0 \\ -0.05 & 0.1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Example:

Find the differential operator Δfor:

- $dx = 0.5$, $dy = 0.3$, $dz = 0.1$units
- $\delta x = 0.02$, $\delta y = 0.04$, $\delta z = 0.06$ radians

Given

Differential translations:

$$dx = 0.5, dy = 0.3, dz = 0.1$$

Differential rotations:

$$\delta x = 0.02, \delta y = 0.04, \delta z = 0.06$$

Write the differential rotation matrix

For small rotations, the differential rotation matrix is:

$$\text{Rot}(k, d\theta) \approx \begin{bmatrix} 1 & -\delta z & \delta y & 0 \\ \delta z & 1 & -\delta x & 0 \\ -\delta y & \delta x & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Substitute values:

$$\text{Rot} = \begin{bmatrix} 1 & -0.06 & 0.04 & 0 \\ 0.06 & 1 & -0.02 & 0 \\ -0.04 & 0.02 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Write the differential translation matrix

$$\text{Trans}(dx, dy, dz) = \begin{bmatrix} 1 & 0 & 0 & dx \\ 0 & 1 & 0 & dy \\ 0 & 0 & 1 & dz \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Substitute values:

$$\text{Trans} = \begin{bmatrix} 1 & 0 & 0 & 0.5 \\ 0 & 1 & 0 & 0.3 \\ 0 & 0 & 1 & 0.1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Multiply Translation × Rotation

$$\text{Trans} \times \text{Rot} = \begin{bmatrix} 1 & -0.06 & 0.04 & 0.5 \\ 0.06 & 1 & -0.02 & 0.3 \\ -0.04 & 0.02 & 1 & 0.1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Subtract Identity Matrix

Identity matrix:

$$I = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Now:

$$\Delta = (\text{Trans} \times \text{Rot}) - I$$

Final Differential Operator Δ

$$\boxed{\Delta = \begin{bmatrix} 0 & -0.06 & 0.04 & 0.5 \\ 0.06 & 0 & -0.02 & 0.3 \\ -0.04 & 0.02 & 0 & 0.1 \\ 0 & 0 & 0 & 0 \end{bmatrix}}$$

MATLAB Code [Differential Operator Δ]

```
clc;
clear;
%% Given differential translations
dx = 0.5;
dy = 0.3;
dz = 0.1;
%% Given differential rotations
delx = 0.02;
dely = 0.04;
delz = 0.06;
%% Differential rotation matrix (small-angle approximation)
Rot = [ 1     -delz   dely    0;
        delz   1     -delx    0;
       -dely  delx    1       0;
```

```
        0      0      0       1 ];
%% Differential translation matrix
Trans = [ 1  0  0  dx;
          0  1  0  dy;
          0  0  1  dz;
          0  0  0  1 ];
%% Identity matrix
I = eye(4);
%% Differential operator Delta
Delta = Trans * Rot - I;
%% Display results
disp('Differential Rotation Matrix =');
disp(Rot);
disp('Differential Translation Matrix =');
disp(Trans);
disp('Differential Operator Delta =');
disp(Delta);
```

Expected Output (Key Result)

```
     0   -0.0600    0.0400    0.5000
 0.0600         0   -0.0200    0.3000
-0.0400    0.0200         0    0.1000
     0         0         0         0
```

Example:

(i) Find the effect of a differential operator of 0.1rad about the y-axis, followed by a differential translation [0.1, 0, 0.2] for the given frame B.

(ii) Find the new location of frame B.

Given frame B:

$$\boldsymbol{B} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{10} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{5} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{3} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix}$$

Differential motion parameters:

$$dx = 0.1, dy = 0, dz = 0.2$$
$$\delta x = 0, \delta y = 0.1, \delta z = 0$$

<u>Solution (i)</u>

Step 1: Write the differential operator Δ

General differential operator:

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

Step 2: Substitute given values

Since rotation is **about y-axis only**:

- $\delta y = 0.1$
- $\delta x = \delta z = 0$

$$\Delta = \begin{bmatrix} 0 & 0 & 0.1 & 0.1 \\ 0 & 0 & 0 & 0 \\ -0.1 & 0 & 0 & 0.2 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Step 3: Differential change in frame

$$[dB] = [\Delta][B]$$

$$[dB] = \begin{bmatrix} 0 & 0 & 0.1 & 0.1 \\ 0 & 0 & 0 & 0 \\ -0.1 & 0 & 0 & 0.2 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 10 \\ 1 & 0 & 0 & 5 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Step 4: Matrix Multiplication

First row:

$$[0\ 0\ 0.1\ 0.1] \Rightarrow [0\quad 0.1\quad 0\quad 0.4]$$

Second row:

$$[0\ 0\ 0\ 0] \Rightarrow [0\quad 0\quad 0\quad 0]$$

Third row:

$$[-0.1\ 0\ 0\ 0.2] \Rightarrow [0\quad 0\quad -0.1\quad -0.8]$$

Fourth row:

$$[0\ 0\ 0\ 0] \Rightarrow [0\quad 0\quad 0\quad 0]$$

Step 5: Final Answer

$$\boxed{[dB] = \begin{bmatrix} 0 & 0.1 & 0 & 0.4 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -0.1 & -0.8 \\ 0 & 0 & 0 & 0 \end{bmatrix}}$$

<u>Solution (ii)</u>

What we already have (from previous calculation)

Original frame *B*

$$B = \begin{bmatrix} 0 & 0 & 1 & 10 \\ 1 & 0 & 0 & 5 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Differential operator Δ

Rotation: 0.1rad about **y-axis**
Translation: [0.1，0，0.2]

$$\Delta = \begin{bmatrix} 0 & 0 & 0.1 & 0.1 \\ 0 & 0 & 0 & 0 \\ -0.1 & 0 & 0 & 0.2 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Differential change in frame *B*

$$[dB] = [\Delta][B]$$

$$[dB] = \begin{bmatrix} 0 & 0.1 & 0 & 0.4 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -0.1 & -0.8 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

the updated frame is:

$$\boxed{B_{\text{new}} = B + dB}$$

Step 2: Add $B + dB$

$$B_{\text{new}} = \begin{bmatrix} 0 & 0 & 1 & 10 \\ 1 & 0 & 0 & 5 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0.1 & 0 & 0.4 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -0.1 & -0.8 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Step 3: Perform Matrix Addition

$$\boxed{B_{\text{new}} = \begin{bmatrix} 0 & 0.1 & 1 & 10.4 \\ 1 & 0 & 0 & 5 \\ 0 & 1 & -0.1 & 2.2 \\ 0 & 0 & 0 & 1 \end{bmatrix}}$$

Emphasizing;

- **ΔB → velocity level**
- **B + ΔB → position level**
- This is the bridge between **Jacobian and forward kinematics**

MATLAB Code [Effect of Rotation and Translation of Differential Operator]

Differential operator acting on frame B

```
clc;

clear;

%% Given frame B

B = [ 0  0  1  10;

    1  0  0   5;

    0  1  0   3;
```

```
    0  0  0   1 ];
%% Given differential translation
dx = 0.1;
dy = 0;
dz = 0.2;
%% Given differential rotation (about y-axis)
delx = 0;
dely = 0.1;   % rotation about y-axis
delz = 0;
%% Differential operator Delta
Delta = [  0     -delz   dely   dx;
          delz   0     -delx   dy;
         -dely   delx    0     dz;
           0      0      0      0 ];
%% Differential change in frame B
dB = Delta * B;
%% Display results
disp('Given Frame B =');
disp(B);
disp('Differential Operator Delta =');
disp(Delta);
disp('Differential Change in Frame dB =');
disp(dB);
```

Inverse Kinematics of Differential Motion

Introduction to Inverse Jacobian:

The Inverse Jacobian is used to determine joint velocities required to achieve a desired end-effector velocity. While the Jacobian maps joint space to task space, the inverse Jacobian performs the reverse mapping.

This concept is fundamental in robot control, trajectory tracking, and motion planning.

"Given a desired small change in end-effector position/orientation, what small joint changes are required?"

Mathematically: forward velocity relation:

$$\boxed{\Delta\mathbf{x} = \mathbf{J}(\mathbf{q})\,\Delta\mathbf{q}} \text{ or } \dot{\mathbf{x}} = J(\mathbf{q})\,\dot{\mathbf{q}}$$

So, Inverse velocity relation:

$$\boxed{\Delta\mathbf{q} = \mathbf{J}^{-1}\,\Delta\mathbf{x}} \text{ or } \dot{\mathbf{q}} = J^{-1}(\mathbf{q})\,\dot{\mathbf{x}}$$

(if **J** is square and non-singular)

This is why it's also called **velocity kinematics**.

Conditions for Existence of Inverse Jacobian:

The inverse Jacobian exists only when:

1. Jacobian matrix is square (DOF = task variables)
2. Jacobian matrix is non-singular

Mathematical condition:

$$\det(J) \neq 0$$

If $\det(J) = 0$, the robot reaches a singular configuration.

Physical Meaning of Inverse Jacobian:

The inverse Jacobian answers:

How much should each joint move to generate a given end-effector motion?

Physical interpretation:

- Near singularity, very large joint velocities are required
- Small Cartesian velocity may cause excessive joint motion
- Indicates sensitivity of robot configuration

Robot Velocity Control:

Given a desired end-effector velocity:

$\dot{x}_d = [v_x \; v_y \; v_z \; \omega_x \; \omega_y \; \omega_z]^T$

Joint velocities are calculated using:

$$\dot{q} = J^{-1}\dot{x}_d$$

Used in real-time control systems.

Example: For 2-DOF Planar Robot Arm find Inverse Jacobian for;

- $L_1 = 1\,m$
- $L_2 = 1\,m$
- $\theta_1 = 30^\circ$
- $\theta_2 = 45^\circ$

Solution:

End-effector equations:

$$x = L_1 \cos\ \theta_1 + L_2 \cos\ (\theta_1 + \theta_2)$$
$$y = L_1 \sin\ \theta_1 + L_2 \sin\ (\theta_1 + \theta_2)$$

Step 1: Jacobian Matrix

$$\mathrm{J} = \begin{bmatrix} -L_1 \sin\ \theta_1 - L_2 \sin\ (\theta_1 + \theta_2) & -L_2 \sin\ (\theta_1 + \theta_2) \\ L_1 \cos\ \theta_1 + L_2 \cos\ (\theta_1 + \theta_2) & L_2 \cos\ (\theta_1 + \theta_2) \end{bmatrix}$$

Substitute values:

$$\mathrm{J} = \begin{bmatrix} -1.673 & -0.966 \\ 0.966 & 0.259 \end{bmatrix}$$

Step 2: Desired End-Effector Change

$$\Delta \mathrm{x} = \begin{bmatrix} 0.01 \\ 0 \end{bmatrix} \text{(move 1 cm in x direction)}$$

Step 3: Inverse Jacobian

$$\Delta \mathrm{q} = \mathrm{J}^{-1} \Delta \mathrm{x}$$

After inversion:

$$\Delta\theta_1 = -0.0026\, rad$$
$$\Delta\theta_2 = +0.0097\, rad$$

MATLAB Code [Inverse Jecobian]

```
L1 = 1; L2 = 1;
t1 = deg2rad(30);
t2 = deg2rad(45);

J = [-L1*sin(t1)-L2*sin(t1+t2), -L2*sin(t1+t2);
     L1*cos(t1)+L2*cos(t1+t2),  L2*cos(t1+t2)];

dx = [0.01; 0];
dq = inv(J)*dx
```

Model Question Bank:

1. What is meant by small-angle assumption? Elaborate the linear velocity and angular velocity with suitable examples.
2. Explain differential motion of a rigid body with neat sketch and mathematical representation
3. Define and explain differential motion in robotics. Why differential motions are preferred over finite motions in robot kinematics.
4. Explain the relationship between linear velocity and angular velocity using the velocity screw (twist) concept.
5. A robot end-effector moves with;
6. $v = [1,0,0]^T$ m/s, $\omega = [0,0.5,0]^T$ rad/s for $\Delta t = 0.1$ s.
   Calculate: (i) Differential position (ii) Differential orientation
7. A robot tool rotates at $\omega = [0, 0, 2]^T$ rad/s for 0.05 s.
   Find the differential rotation vector.
8. Explain differential transformation of a robot frame combining translation and rotation with neat diagram.
9. A 2-DOF planar robot has two revolute joints with link lengths $l_1 = 1$ m, $l_2 = 1$ m.

   At a certain instant:

   - $\theta_1 = 30°$
   - $\theta_2 = 45°$
   - Joint velocities: $\dot{\theta}_1 = 0.5$ rad/s, $\dot{\theta}_2 = 0.3$ rad/s

   Find the linear velocity of the end-effector using Jacobian.

10. Find the differential operator Δ for:

    $dx = 0.5$, $dy = 0.3$, $dz = 0.1$units

    $\delta x = 0.02$, $\delta y = 0.04$, $\delta z = 0.06$ radians

11. Find the effect of a differential operator of 0.1rad about the y-axis, followed by a differential translation $[0.1,\ 0,\ 0.2]$for the given frame B. Also; find the new location of frame B.
12. Write the general form of the Jacobian for an n-DOF manipulator with physical significance of each column of it.