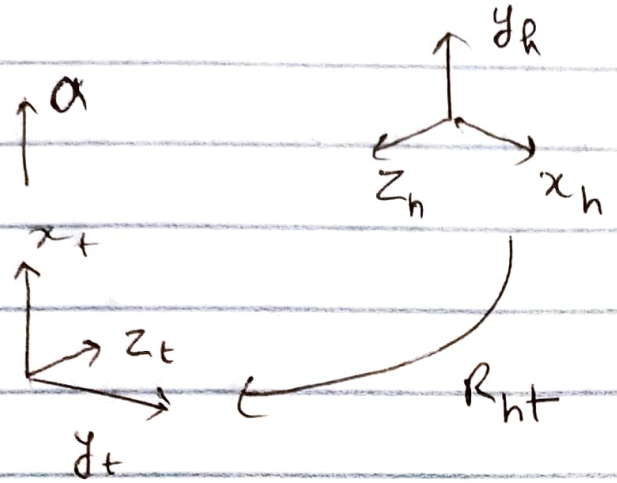
Nishant Bharali

PID: 906641308

HW 2

Robotics & Automation

(1)



We know that;

let us say; $\left(\dot{R} = \dot{R}_{ht}\right)$

$$\dot{R} = \frac{\partial R}{\partial t} = \begin{bmatrix} \omega \times x & \omega \times y & \omega \times z \end{bmatrix}$$

$$\dot{R} = [\omega] R \quad ([\omega] \rightarrow \text{skew-symmetric})$$

$$\therefore \dot{R}_{ht} = [\omega_h] R_{ht} \quad (\text{multiplying both sides by } R_{ht}^{-1})$$

$$\Rightarrow \dot{R}_{ht} R_{ht}^{-1} = [\omega_h] R_{ht} R_{ht}^{-1} \quad (\because R_{ht}^{-1} = R_{ht}^{T})$$

$$\Rightarrow \boxed{[\omega_h] = \dot{R} R_{ht}^{T}} \quad \text{or} \quad [\alpha] = \dot{R} R_{ht}^{T}$$

$[3\times3]$ $[3\times3]$

Similarly,

$$[\omega_t] = R_{ht}^{T} \dot{R}, \quad \text{or} \quad \boxed{[\alpha] = R_{ht}^{T} \dot{R}_{ht}}$$

Using change of reference frame subscript - cancellation rule:

$$\because \quad \dot{R}_{ht} = [\omega_h] R_{ht} = [\dot{\hat{x}}_t \quad \dot{\hat{y}}_t \quad \dot{\hat{z}}_t]$$

Thus:

$$\omega_t = R_{th}\,\omega_h = R_{ht}^{-1}\,\omega_h$$

$$\Rightarrow \quad \omega_t = R_{ht}^{T}\,\omega_h$$

Multiply by R_{ht} on left-side of both the sides of equation:

$$\Rightarrow \quad R_{ht}\,\omega_t = R_{ht}\,R_{ht}^{T}\,\omega_h$$

$$\therefore \quad \boxed{\omega_h = R_{ht}\,\omega_t}$$

1.2 ZXX Euler angles do not exist because they represent an invalid sequence of rotations that cannot be uniquely defined in 3D space.

The problem with ZXX sequence is that the two ~~consequ~~ consecutive rotations about the X-axis are redundant. When we rotate about the X-axis twice, the second rotation essentially cancels out the first one.

Thus, final orientation of the object can be acheived by a single rotation about the X-axis, and information about the X-axis rotation is lost.

[1.3]

$$R = \begin{bmatrix} 0.3642 & -0.7729 & 0.5090 \\ 0.7017 & -0.1296 & -0.7006 \\ 0.6124 & 0.6124 & 0.5 \end{bmatrix}$$

we know :-

$$\Rightarrow \theta = \cos^{-1}\left[\frac{1}{2}(\text{trace}(R) - 1)\right]$$

$$\therefore \text{trace}(R) = 0.7346$$

$$\therefore \theta = \cos^{-1}\left[\frac{1}{2}(-0.2654)\right]$$

$$\Rightarrow \theta = \cos^{-1}\left[-0.1327\right]$$

$$\theta = 1.703 \text{ radians} \quad \text{or} \quad 97.62^\circ$$

$$\therefore [\omega] = \frac{1}{2\sin\theta}(R - R^T) \qquad \sin(97.62) \cong 1$$

$$\therefore [\omega] = \frac{1}{2 \times 0.99}\left(\begin{bmatrix} 0.0018 & -1.4746 & 0.1034 \\ 1.4746 & 0 & -1.313 \\ 0.1034 & 1.313 & 0 \end{bmatrix}\right)$$

$\Rightarrow \frac{0.xxx}{2} \approx 0$ & $\sin(97.62^\circ) \approx 1$ (0.99)

$$\Rightarrow [\omega] = \frac{1}{2\times 1}\begin{bmatrix} 0 & -1.475 & 0 \\ 1.475 & 0 & -1.313 \\ 0 & 1.313 & 0 \end{bmatrix}$$

$$\Rightarrow [\omega] = \begin{bmatrix} 0 & -0.737 & 0 \\ 0.737 & 0 & -0.6565 \\ 0 & 0.6565 & 0 \end{bmatrix}$$

Converting skew-symmetric to ω :-

$$\therefore \omega = \begin{bmatrix} 0.6565 \\ 0 \\ 0.737 \end{bmatrix}$$

$\therefore \omega = [0.6565 \quad 0 \quad 0.737]$

$\theta = 97.62^\circ$ or 1.703 radians

Axis - angle representation is (ω, θ)

[2]

[2.1] Position of tennis ball after 2.5 seconds :

$$p(2.5) = p(0) + \int_0^t V \cdot dt$$

Here, V is the linear velocity vector,

$$V = \beta \cdot x_s \quad (\because +ve\ x_s\text{-axis})$$

$\therefore$ Final position :-

$$p(2.5) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \int_0^{2.5} \beta \cdot x_s \, dt$$

$$= \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \beta \cdot x_s \cdot (2.5 - 0)$$

$$= \begin{bmatrix} 2.5\beta \\ 0 \\ 0 \end{bmatrix}$$

$\therefore$ $p(2.5) = [2.5\beta \quad 0 \quad 0]^T$

2.2] To calculate orientation of tennis ball, we need to apply rotation about z-axis over time ;

$$\therefore R(t) = R(0) \cdot R_z(\theta, t)$$

Here, $\theta = \alpha$ rads/sec

$$\therefore R(2.5) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(2.5\alpha) & -\sin(2.5\alpha) & 0 \\ \sin(2.5\alpha) & \cos(2.5\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\Rightarrow \boxed{R(2.5) = \begin{bmatrix} \cos(2.5\alpha) & -\sin(2.5\alpha) & 0 \\ \sin(2.5\alpha) & \cos(2.5\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix}}$$

[3.11] Since R_1 & R_2 are non-identity rotational matrices, P_1 & P_2 are non-zero translation vectors:

$$\Rightarrow T_1 \cdot T_2 = \begin{bmatrix} R_1 & P_1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_2 & P_2 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_1 \cdot R_2 & R_1 P_2 + P_1 \\ 0 & 1 \end{bmatrix}$$

$$T_2 \cdot T_1 = \begin{bmatrix} R_2 & P_2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_1 & P_1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_2 \cdot R_1 & R_2 \cdot P_1 + P_2 \\ 0 & \end{bmatrix}$$

As we can see, unless R_1 & R_2 are special rotation matrices & P_1 and P_2 are carefully chosen translation vectors $T_1 \cdot T_2 \neq T_2 \cdot T_1$.

In general, transformation matrices do not commute, which means the order of transformation matters

The perfect condition would be when:

$$R_1 \cdot R_2 = R_2 \cdot R_1 \quad \text{(possible)}$$

$$\rightarrow \quad R_1 P_2 + P_1 = R_2 P_1 + P_2$$

$\leq$ Rare scenario in a 2D plane is possible for T_1 & T_2 to be commutative.

But no such special case exists in a 3D space as 2 transformation matrices to be commutative to each other without having $R_{1/2}$ as an identity matrix or $p_1 \times p_2$ as 0 vectors.

&, as $R_1 R_2 = R_2 R_1$ is possible,

we require $p_1 = p_2 = 0$, there exists no solution where non-identity $R_1 \times R_2$ and non-zero $p_1 \times p_2$ resulting in $T_1 \times T_2$ commuting.

3.2| Program implementation in Python in google colab is shown bellow in the snippets :

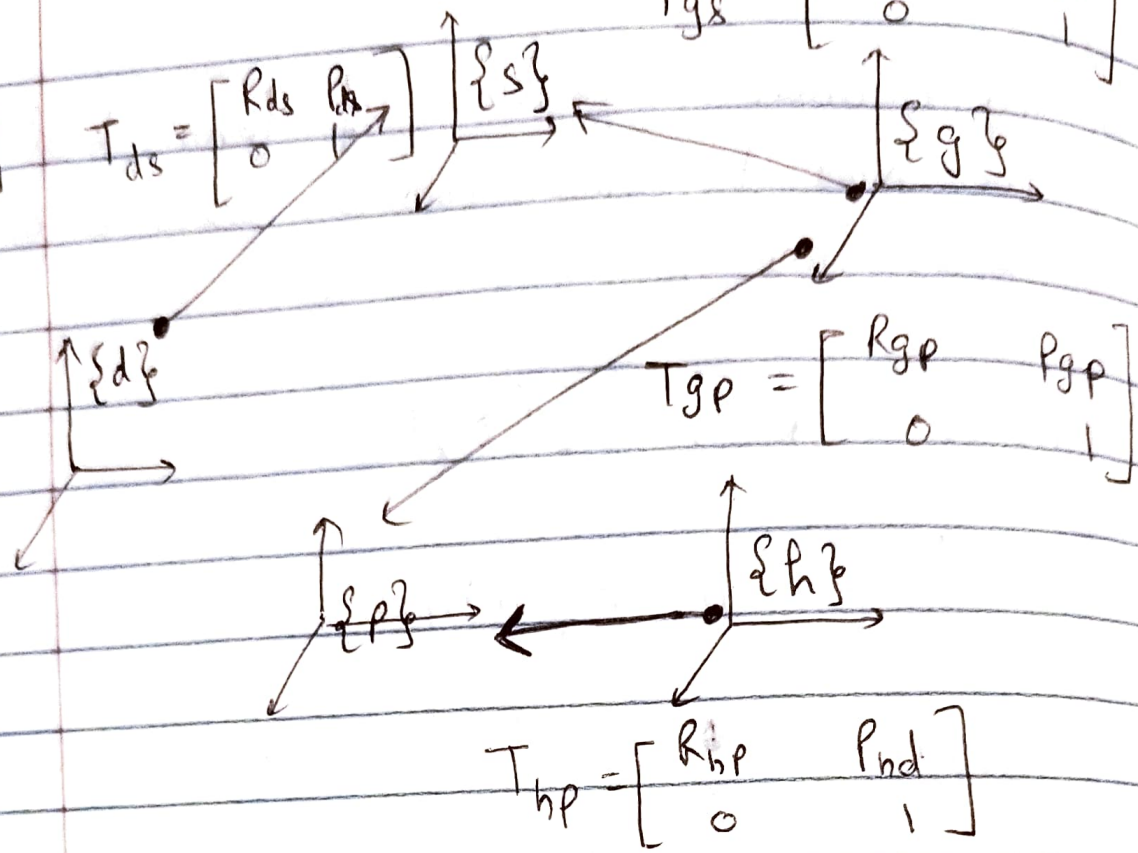
(colab files are attached as well)

```python
import numpy as np

def isTransform(X):

  R = X[:3,:3]
  p = X[:3,3]

  is_rot = (np.allclose(np.linalg.norm(R[0]), 1) and
            np.allclose(np.linalg.norm(R[1]), 1) and
            np.allclose(np.linalg.norm(R[2]), 1) and
            np.abs(np.linalg.det(R) - 1) < 1e-12)

  is_homog = (X.shape == (4,4)) and (X[3,3] == 1)

  return is_rot and is_homog
# Test cases
X1 = np.array([[1/np.sqrt(2), -1/np.sqrt(2), 0, -5],
               [1/np.sqrt(2), 1/np.sqrt(2), 0, -1.7],
               [0, 0, 1, -11],
               [0, 0, 0, 1]])

X2 = np.array([[0.5, -1/np.sqrt(2), 0.5, 0],
               [0.5, 1/np.sqrt(2), 0.5, 0],
               [-1/np.sqrt(2), 0, 1/np.sqrt(2), 0],
               [0, 0, 0, 1]])

X3 = np.array([[-1/np.sqrt(2), -0.5, 0.5, -1.1],
               [1/np.sqrt(2), -0.5, -0.5, 4.5],
               [0, 1/np.sqrt(2), 1/np.sqrt(2), 1.6],
               [0, 0, 0, 1]])

print(isTransform(X1)) # True
print(isTransform(X2)) # True
print(isTransform(X3)) # False
```

```
print(isTransform(X1)) # True
print(isTransform(X2)) # True
print(isTransform(X3)) # False
```

```
True
True
False
```

4.1



4.2 Since, we know;

$$T_{ab}^{-1} = T_{ba} \quad \text{(inverse rule)}$$

We have to find T_{dh} using known transformation matrices :

⇒ ~~$T_{dh}^{-1} = T_{ds}$~~

$$T_{dh} = T_{ds}\, T_{gs}^{-1}\, T_{gp}\, T_{hp}^{-1}$$

Using subscript cancellation rule, we verify:

$\Rightarrow T_{dh} = T_{ds} T_{sg} T_{gp} T_{ph}$

$T_{dh} = T_{dh}$ ✓

Thus, looking at the diagram, the analogy and direction of persepective is correct for each T matrix.

[5] Using the diagram, we can get T_{sb} as:

$$T_{sb} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

[5.1] Rotation at {s} by $\pi/4$:

Fixed-frame rotation; We pre-multiply by a rotation around Z_s:

$$T_{sb}' = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{sb}' = \begin{bmatrix} -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & -5/\sqrt{2} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 5/\sqrt{2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

5.2 Here, rotate by $-\pi/4$ at {b} :

$\therefore$ Body-frame motion, thus post multiply to T_{sb}' from 5.1 by a rotation around Z_s :

$$\Rightarrow T_{sb}'' = \begin{bmatrix} -1/\sqrt{2} & -1/\sqrt{2} & 0 & -5/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} & 0 & 5/\sqrt{2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 & \\ -1/\sqrt{2} & 1/\sqrt{2} & 0 & \\ 0 & 0 & 1 & \\ 0 & 0 & 0 & \end{bmatrix}$$

$$\Rightarrow T_{sb}'' = \begin{bmatrix} 0 & -1 & 0 & -5/\sqrt{2} \\ 1 & 0 & 0 & 5/\sqrt{2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

5.3 for $\pi/2$ at {b}, again :

Body-frame motion, thus post-multiply to T_{sb}'' from 5.2 due to continuation by a rotation of Z_s ;

$$\Rightarrow T_{sb}''' = \begin{bmatrix} 0 & -1 & 0 & -5/\sqrt{2} \\ 1 & 0 & 0 & 5/\sqrt{2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\Rightarrow T_{sb}^{'''} = \begin{bmatrix} -1 & 0 & 0 & -5/\sqrt{2} \\ 0 & -1 & 0 & 5/\sqrt{2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$T_{sb} = T_{sb}^{'''}$ is the final transformation after all the motions above

[6]

(6-1) Let's define V_s and V_b in terms of the spatial twist V_s and the body twist V_b:

$$V_s = [\omega_s \quad v_s]$$
$$V_b = [\omega_b \quad v_b]$$

v_s is linear velocity of drone in person's ref frame
v_b is linear velocity of drone in its own frame.
ω_s in angular velocity of drone in person's frame
ω_b is ang. velocity of drone in its own frame

- $V_s = V_b$ is only possible if both linear and angular velocities are equal to their respective velocities in each other's frames.

Conditions:

→ Linear Velocity condition: $v_s = v_b$

→ Angular velocity condition: $\omega_s = \omega_b$

- for $\|V_s\| = \|V_b\|$, it implies that the magnitudes of both linear and angular velocities are equal for $V_s \neq V_b$.

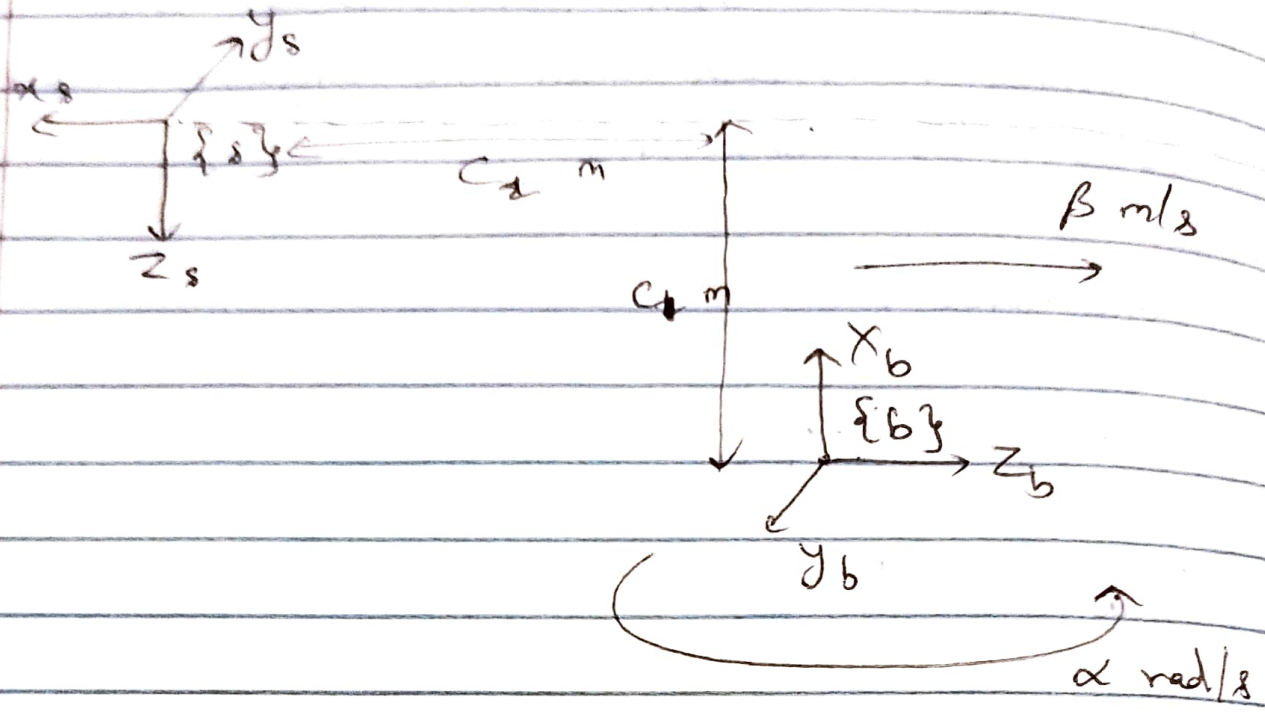
∴ Conditions :

→ $\|v_s\| = \|v_b\|$

→ $\|\omega_s\| = \|\omega_b\|$

Please note that conditions above don't depend on $T_{sb} = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix}$ explicitly, they are based on equality of magnitudes of linear & angular velocities in different reference frames.

6.2



Body twist :

$$V_b = \begin{bmatrix} \alpha \\ 0 \\ 0 \\ 0 \\ 0 \\ \beta \end{bmatrix}$$

Skater is rotating around $+x_b$ axis & translating along $+z_b$ axis.

Spatial twist :

Skater is rotating around the $-z_s$ axis and translating along $-x_s$ axis :

$$\therefore \quad \omega_s = [0 \;\; 0 \;\; -\alpha]^T$$

$$\dot{p}_{sb} = [-\beta \;\; 0 \;\; 0]^T = \dot{p}$$

We also need to address linear velocity at $\{s\}$ due to angular velocity of skater.

$$\therefore \quad p_{sb} = [-c_2 \;\; 0 \;\; c_1]^T = p$$

So, spatial twist is :

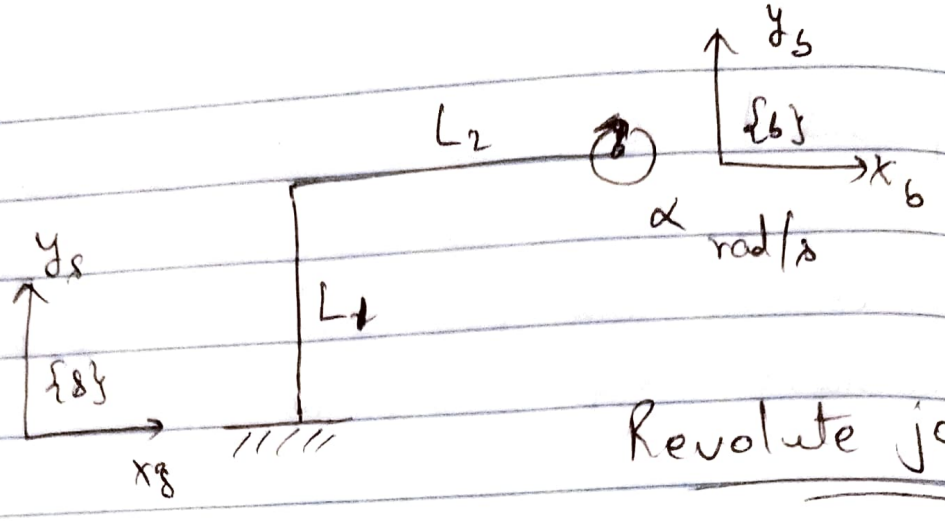
$$V_s = \begin{bmatrix} \omega_s \\ -\omega_s \times p + \dot{p} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -\alpha \\ -\beta \\ -\alpha c_2 \\ 0 \end{bmatrix}$$

Steps:

$$\Rightarrow \; -\omega_s \times p + \dot{p} = \begin{bmatrix} 0 \\ 0 \\ \alpha \end{bmatrix} \times \begin{bmatrix} -c_2 \\ 0 \\ c_1 \end{bmatrix} + \dot{p} = \begin{bmatrix} 0 \\ -\alpha c_2 \\ 0 \end{bmatrix} + \dot{p} = \begin{bmatrix} -\beta \\ -\alpha c_2 \\ 0 \end{bmatrix}$$

7

7.11



Revolute joint

- for Spatial twist V_s :

we know,

$$V_s = \begin{bmatrix} \omega_s \\ v_s \end{bmatrix} = \begin{bmatrix} \omega_s \\ -\omega_s \times p + \dot{p} \end{bmatrix}$$

$$\therefore \quad \omega_s = [0 \; 0 \; \alpha]$$

joint is rotating around $+Z_s$ axis ;

Here, $\dot{p}$ does not exist or is $[0 \; 0 \; 0]^T$ as there is no linear velocity of the joint.

$$\therefore \quad V_s = \begin{bmatrix} \omega_s \\ -\omega_s \times p \end{bmatrix}$$

$$\Rightarrow \quad p = p_{sb} = [L_2 \; L_1 \; 0]^T$$

$$\Rightarrow \quad -\omega_s \times p = \begin{bmatrix} 0 \\ 0 \\ -\alpha \end{bmatrix} \times \begin{bmatrix} L_2 \\ L_1 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} \alpha L_1 \\ -\alpha L_2 \\ 0 \end{bmatrix}$$

$$\therefore \quad V_s = \begin{bmatrix} 0 & 0 & \alpha & \alpha L_1 & -\alpha L_2 & 0 \end{bmatrix}^T$$

- Screw S for a revolute joints, since there is no linear velocity; Using diagram, we get

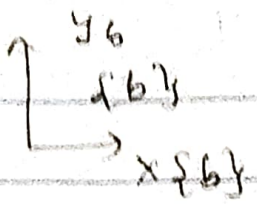
$$S = \begin{bmatrix} \omega_s \\ -\omega_s \times p \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ L_1 \\ -L_2 \\ 0 \end{bmatrix}$$

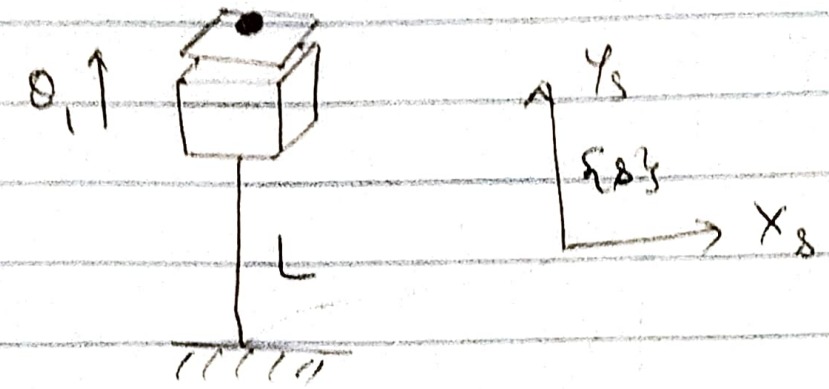
$$\omega_s = \begin{bmatrix} 0 \\ 0 \\ \alpha \end{bmatrix}$$

$\therefore$ Screw axis $\mathcal{S} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$

- We can see that S is a normalized twist V_s because ω_s is a unit vector, i.e. $[0\ 0\ 1]^T \times -\omega_s \times p = [L_1\ -L_2\ 0]^T = v_s$ for screw S.

7.2





$$S = \begin{bmatrix} 0 \\ v_s \end{bmatrix}$$

Here,

$$S = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad ; \quad T(0) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & L \\ 0 & 0 & 1 & 0 \\ & 0 & & 1 \end{bmatrix}$$

$$\because T(\theta) = e^{[S]\theta} T(0)$$

$$\Rightarrow T(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & L+\theta_1 \\ 0 & 0 & 1 & 0 \\ & 0 & & 1 \end{bmatrix} = T_{sb}$$

{b} translates along y_s axis

A code representation with an example is here as follows:

screw2matrix.m

/MATLAB Drive/screw2matrix.m

```
function T = screw2matrix(S, theta)
    T = expm(bracket(S) * theta);

function bracket_S = bracket(S)
    bracket_S = [0 -S(3) S(2) S(4);
        S(3) 0 -S(1) S(5);
        -S(2) S(1) 0 S(6); 0 0 0 0];
    end
end
```

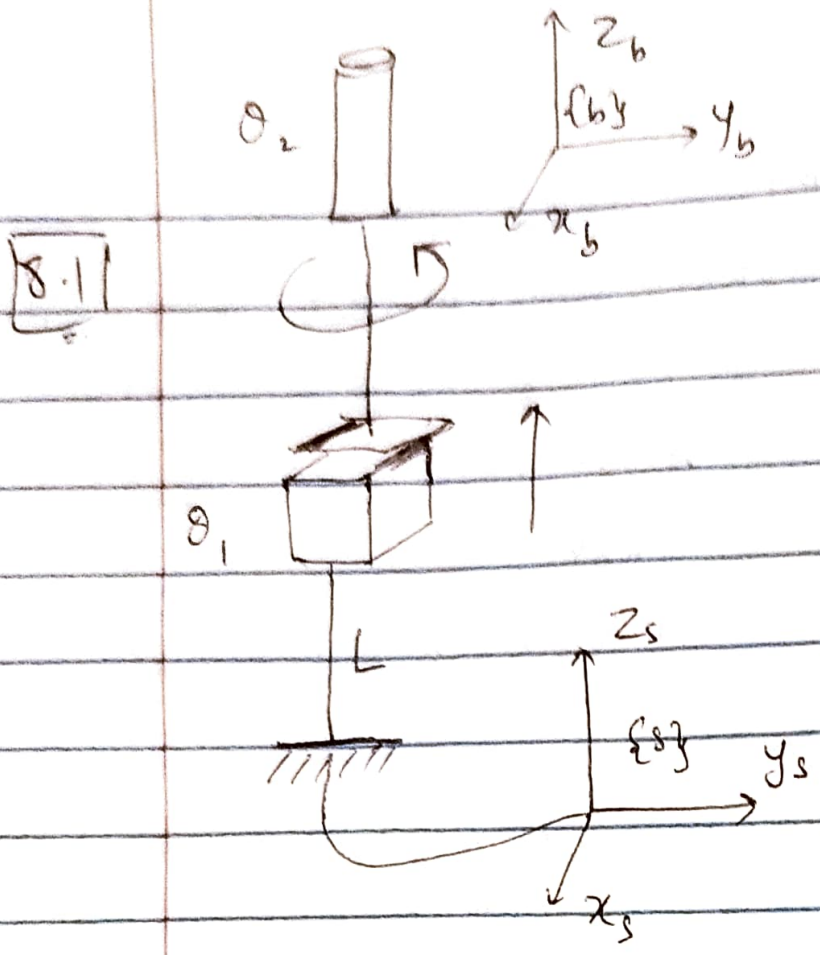
Command Window

```
Error using open
Failed to open MATLAB Editor.

>> S = [0; 0; 0; 1; 0; 1];
theta = 0;

% Compute transformation matrix
T = screw2matrix(S, theta)

T =

     1     0     0     0
     0     1     0     0
     0     0     1     0
     0     0     0     1

>> S = [0; 0; 0; 1; 0; 1];
theta = pi/2;

% Compute transformation matrix
T = screw2matrix(S, theta)

T =

    1.0000         0         0    1.5708
         0    1.0000         0         0
         0         0    1.0000    1.5708
         0         0         0    1.0000

>>
```

```python
#Problem 7.2
import numpy as np

def bracket(S):
    bracket_S = np.array([[0, -S[2], S[1], S[3]],
                          [S[2], 0, -S[0], S[4]],
                          [-S[1], S[0], 0, S[5]],
                          [0, 0, 0, 0]])
    return bracket_S

def screw2matrix(S, theta):
    bracket_S = bracket(S)
    T = np.eye(4) + bracket_S * theta
    return np.exp(T)

# Problem 7.2: Example of a screw with a normalized Vs and theta = 0 and pi/2 for two cases
S = [0, 0, 0, 0, 1, 0]
theta1 = 0 # 0 degrees
theta2 = np.pi/2  # 90 degrees

T_0 = screw2matrix(S, theta1)
T_90 = screw2matrix(S, theta2)

print("T_0 =\n", T_0, "\n")
print("T_90 =\n", T_90, "\n")
```

```
print("T_0 =\n", T_0, "\n")
print("T_90 =\n", T_90, "\n")
```

```
T_0 =
 [[2.71828183 1.         1.         1.        ]
 [1.         2.71828183 1.         1.        ]
 [1.         1.         2.71828183 1.        ]
 [1.         1.         1.         2.71828183]]

T_90 =
 [[2.71828183 1.         1.         1.        ]
 [1.         2.71828183 1.         4.81047738]
 [1.         1.         2.71828183 1.        ]
 [1.         1.         1.         2.71828183]]
```



S_1 Screw for the prismatic joint (position θ_1).

Prismatic joint is translating along z_s axis

Screw is :

$$S_1 = \begin{bmatrix} 0 \\ v_s \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

v_s is the unit vector (linear velocity)

(Revolute joint)

for S_2 :

$\omega_s \rightarrow$ unit vector

$$S_2 = \begin{bmatrix} \omega_s \\ -\omega_s \times p \end{bmatrix} \quad ; \quad \omega_s = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \text{ (for screw)}$$

$$p = p_{sb} = \begin{bmatrix} 0 \\ L+\theta_1 \\ 0 \end{bmatrix} \quad \& \quad -\omega_s \times p = \begin{bmatrix} L+\theta_1 \\ 0 \\ 0 \end{bmatrix}$$

$$\Rightarrow \quad S_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ L+\theta_1 \\ 0 \\ 0 \end{bmatrix}$$

8.2 Converting screws to body-motion :

$$T(\theta) = e^{[S]\theta} T(0)$$

$$T(0) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & L \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

for S_1 ;

$$T_{sb_1}(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & L+\theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

for S_2 screw;

$$T(0) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & L+\theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & & 0 & 1 \end{bmatrix}$$

Using $T(\theta) = e^{[S]\theta} T(0)$

$\therefore T_{sb_2}(\theta) =$

$$\begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & 0 \\ \sin\theta_2 & \cos\theta_2 & 0 & L+\theta_1 \\ 0 & 0 & 1 & 0 \\ & 0 & & 1 \end{bmatrix}$$

Since, the rotation is around $+z_s$ axis of $\{b\}$

8.3 $T_{sb_1}(\pi/2) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & l+s \\ 0 & 0 & 1 & 0 \\ 0 & & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 6 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

$$T_{sb_2}(\pi/2) = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 6 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{s6}(\theta) = T_{s6_1}(\pi/2)\, T_{s6_2}(\pi/2)$$

$$= \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 12 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

I am confused whether this multiplication was necessary or do I leave it at the $T_{s6}(\pi/2)$ calculations for separate joints.