

RIGID BODY - ANGULAR KINEMATICS

A rigid body in 3D space is defined by 6 degrees of freedom:

- 3 spatial coordinates for the center of mass, \vec{r}_{CM}
- 3 angles, ϕ, θ, ψ

$$\vec{r}_{CM} \text{ follows: } \frac{d\vec{r}_{CM}}{dt} = \vec{v}_{CM} \leftarrow \text{easy}$$

The goal of this document is to derive the time-evolution equations for ϕ, θ and ψ :

$$\frac{d}{dt} \begin{pmatrix} \phi \\ \theta \\ \psi \end{pmatrix} = ?$$

INTUITIVE (BUT WRONG) ANSWER:

$$\frac{d}{dt} \begin{pmatrix} \phi \\ \theta \\ \psi \end{pmatrix} \stackrel{\text{No!}}{=} \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \quad \text{where } \omega_i \text{ are the angular velocities}$$

This is ONLY valid @ $\phi = \theta = \psi = 0$!

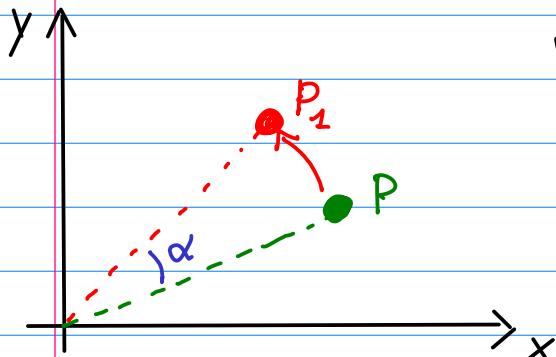
In the general case, $\frac{d}{dt}(\phi, \theta, \psi) = f(\vec{\omega}, \phi, \theta, \psi)$

Let's start by considering rotation matrices

RECAPS ON ROTATIONS & ROTATED REFERENCE SYSTEMS

ROTATING A VECTOR

Consider a 2D plane, (x, y) , and a point, P .



We can rotate the point, P , about the origin, by an angle α , using the rotation matrix

$$\underline{r}_{P1} = \underline{R} \underline{r}_P$$

POSITION OF THE NEW POINT, P_1 , IN THE SAME REFERENCE SYS, (x, y)

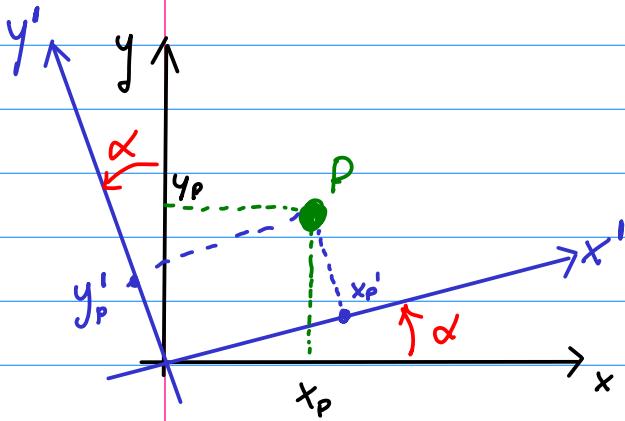
$$\underline{R} = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix}$$

Again, \underline{R} takes a vector and returns a new vector, rotated about the origin.

CHANGING REFERENCE SYSTEM

What if you wanted to transform a vector from a reference system, $(\hat{i}, \hat{j}, \hat{k})$, to a rotated frame, $(\hat{i}', \hat{j}', \hat{k}')$?

Some thing! Almost. Look



In the new reference system, the point, P, will have coordinates (x'_p, y'_p)

The rotation of the reference system by an angle α is equivalent to a rotation of the point, P, by an angle $-\alpha$.

Therefore:

$$\underline{r}'_P = \begin{pmatrix} x'_P \\ y'_P \end{pmatrix} = T(\alpha) \begin{pmatrix} x_P \\ y_P \end{pmatrix} = R(-\alpha) \begin{pmatrix} x_P \\ y_P \end{pmatrix}$$

Position of P in the original frame

position of P in the new "primed" rotated frame

Transformation matrix

Since $R(\alpha) = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} \Rightarrow T(\alpha) = \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix}$

ROTATIONS IN 3D

Let us now write the rotation matrices & transformation mat. for rotations in 3D.

$\phi \rightarrow$ rotation about \hat{i}

$\theta \rightarrow$ \hat{n} \hat{n} \hat{j}
 $\psi \rightarrow$ \hat{n} \hat{n} \hat{k}

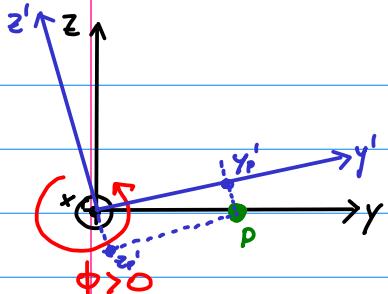
The thing is: the structure of the matrices is always

$$\begin{bmatrix} \cos(\cdot) & \pm \sin(\cdot) \\ \mp \sin(\cdot) & \cos(\cdot) \end{bmatrix}$$

And the question is: where is the minus?

To find it, we just consider a point located on x, and see what happens ...

• Rotation about \hat{x}



consider a point located along y ,
with $y_p \neq 0, z_p = 0$.

Then, $\begin{cases} y'_p = y_p \cos \phi \\ z'_p = -y_p \sin \phi \end{cases}$

or: $\begin{pmatrix} y'_p \\ z'_p \end{pmatrix} = \begin{bmatrix} \cos \phi & ? \\ -\sin \phi & ? \end{bmatrix} \begin{pmatrix} y_p \\ z_p \end{pmatrix} = T_\phi \begin{pmatrix} y_p \\ z_p \end{pmatrix}$

We see that the minus is here, so, the transformation matrix is:

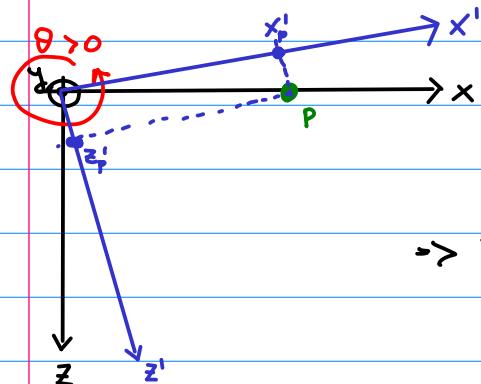
$$T_\phi^{(2D)} = \begin{bmatrix} \cos \phi & +\sin \phi \\ -\sin \phi & \cos \phi \end{bmatrix}$$

In 3D, we simply have to maintain x unchanged:

$$T_\phi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix}$$

and thus $R_\phi = T_\phi(-\phi) = T_\phi(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix}$

• Rotation about \hat{y} (angle theta)



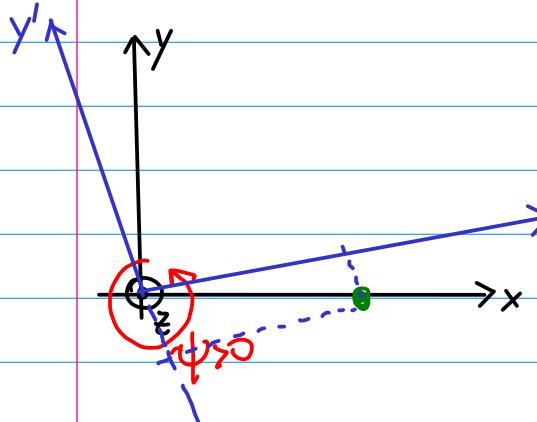
$$x'_p = x_p \cos \theta ; z'_p = x_p \sin \theta$$

$$\Rightarrow \begin{pmatrix} x'_p \\ z'_p \end{pmatrix} = \begin{bmatrix} \cos \theta & ? \\ \sin \theta & ? \end{bmatrix} \begin{pmatrix} x_p \\ z_p \end{pmatrix} = T_\theta \begin{pmatrix} x_p \\ z_p \end{pmatrix}$$

$$\Rightarrow T_\theta^{(2D)} = \begin{bmatrix} \cos \theta & 0 \\ 0 & +\sin \theta \\ +\sin \theta & \cos \theta \end{bmatrix}$$

$$\Rightarrow T_\theta = \begin{bmatrix} \cos \theta & 0 & 0 \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix}$$

• Rotation about \hat{z} (angle psi)



$$x'_p = x_p \cos \psi$$

$$y'_p = -x_p \sin \psi$$

$$T_\psi^{(2D)} = \begin{bmatrix} \cos \psi & \sin \psi \\ -\sin \psi & \cos \psi \end{bmatrix}$$

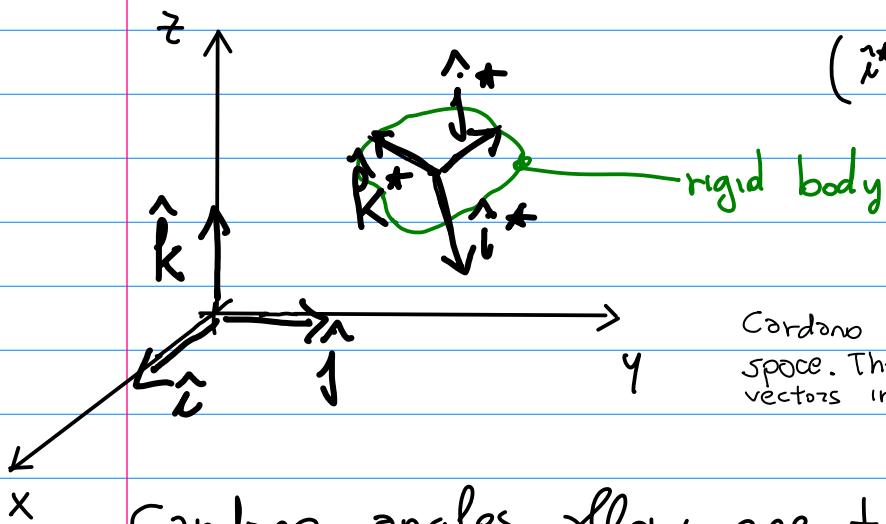
$$\Rightarrow T_\psi = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

USING THESE TRANSFORMATION MATRICES, ONE CAN EXPRESS VECTORS IN DIFFERENT REF. FRAMES, ROTATED WRT EACH OTHER

CARDANO ANGLES

When dealing with rigid bodies, one often employs Cardano angles.

$(\hat{i}, \hat{j}, \hat{k}) \rightarrow$ Reference axes



$(\hat{i}^*, \hat{j}^*, \hat{k}^*) \rightarrow$ Body-fitted axes

Cardano angles allow one to rotate a vector in space. They are used to rotate the orientation unit vectors in space, to follow a rigid body

Cardano angles allow one to rotate a vector in space. They are used to rotate the orientation unit vectors, $\hat{i}, \hat{j}, \hat{k}$, into body-fitted vectors, $\hat{i}^*, \hat{j}^*, \hat{k}^*$. They work by performing three consecutive rotations:

$$\begin{array}{c}
 (\hat{i}, \hat{j}, \hat{k}) \xrightarrow[R_\psi]{\psi, \hat{k}} (\hat{i}_1, \hat{j}_1, \hat{k}_1) \xrightarrow[R_\theta]{\theta, \hat{j}_1} (\hat{i}_2, \hat{j}_2, \hat{k}_2) \xrightarrow[R_\phi]{\phi, \hat{i}_2} (\hat{i}^*, \hat{j}^*, \hat{k}^*)^* \\
 \text{INITIAL AXES} \qquad \qquad \qquad \text{TEMPORARY AXES} \qquad \qquad \qquad \text{BODY AXES}
 \end{array}$$

Therefore, we can write: $\underline{v}^* = \underline{R}_\phi \underline{R}_\theta \underline{R}_\psi \underline{v}$

rotated vector

↗ 1st rotation
 ↗ 2nd rotation
 ↗ 3rd rotation

whatever vector

THIS FORMULATION SUFFERS FROM THE GIMBAL LOCK → this is why people often uses quaternions
 (However, there are various tricks for avoiding the gimbal lock...)

The rotation matrix is $\underline{\underline{R}} = \underline{\underline{R}}_\phi \underline{\underline{R}}_\theta \underline{\underline{R}}_\psi$

\downarrow 1st rot
 \downarrow 2nd rot
 \downarrow 3rd rot

Note that $\underline{\underline{R}}$ rotates a vector, in 3D. If we want instead to change reference system (which is a different task), we need to use a transformation matrix,

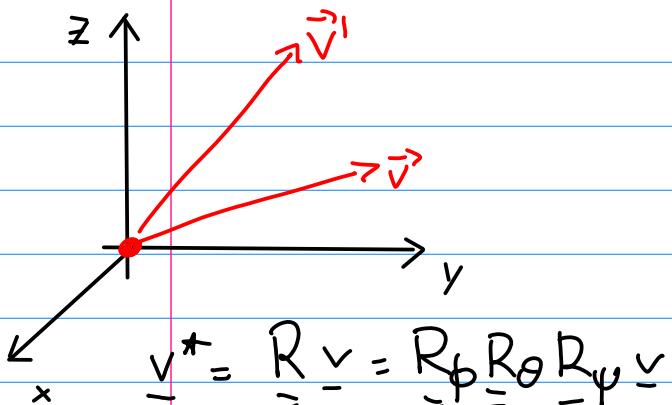
$$\underline{\underline{T}} = \underline{\underline{R}}^T = (\underline{\underline{R}}_\phi \underline{\underline{R}}_\theta \underline{\underline{R}}_\psi)^T = \underline{\underline{R}}_\psi^T \underline{\underline{R}}_\theta^T \underline{\underline{R}}_\phi^T = \underline{\underline{T}}_\psi \underline{\underline{T}}_\theta \underline{\underline{T}}_\phi$$

$$\boxed{\underline{\underline{T}} = \underline{\underline{T}}_\psi \underline{\underline{T}}_\theta \underline{\underline{T}}_\phi}$$

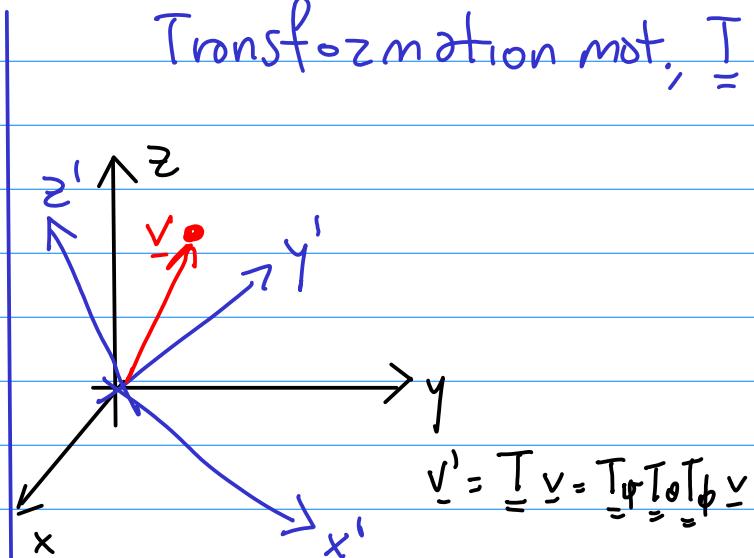
← this has the opposite order wrt $\underline{\underline{R}}$

ONCE AGAIN:

Rotation matrix, $\underline{\underline{R}}$



Transformation mat., $\underline{\underline{T}}$



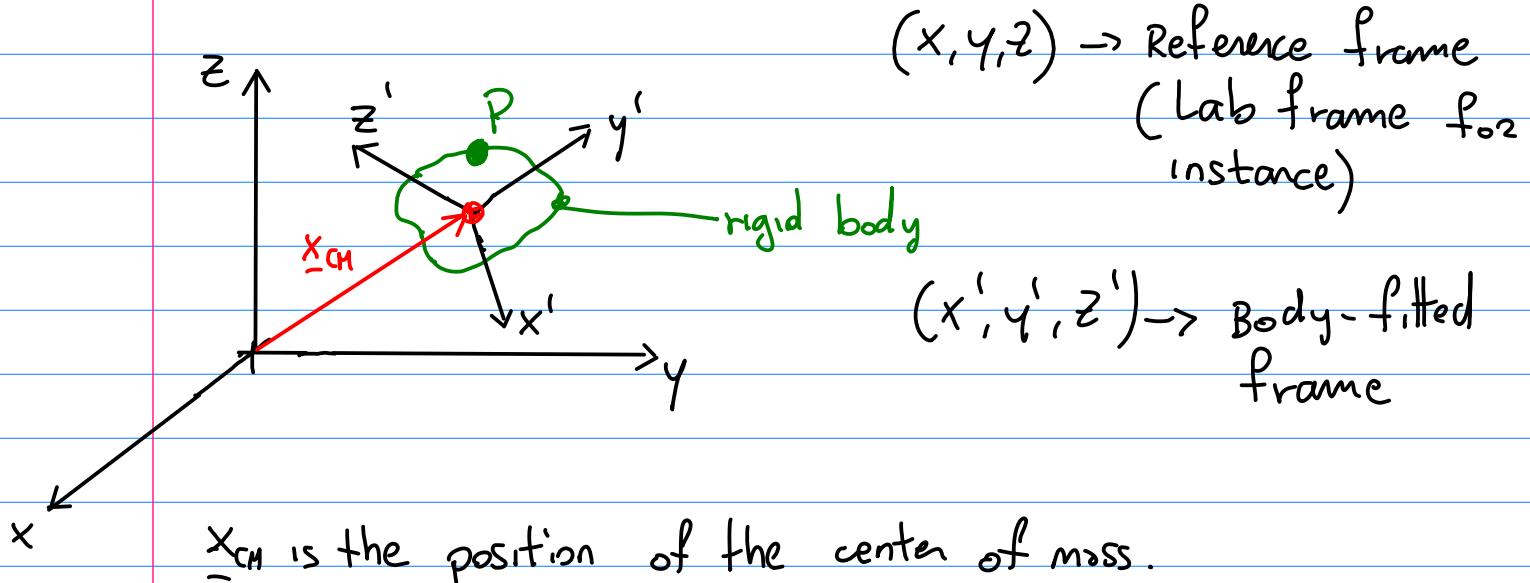
$\underline{\underline{v}}^*$ is a vector, rotated from $\underline{\underline{v}}$, and expressed in the (x, y, z) frame

$\underline{\underline{v}}'$ is the very same vector as $\underline{\underline{v}}$, but is expressed in a rotated frame, (x', y', z')

ANGULAR KINEMATICS

OK, we are ready to derive the equations for $\dot{\phi}, \dot{\theta}, \dot{\psi}$.

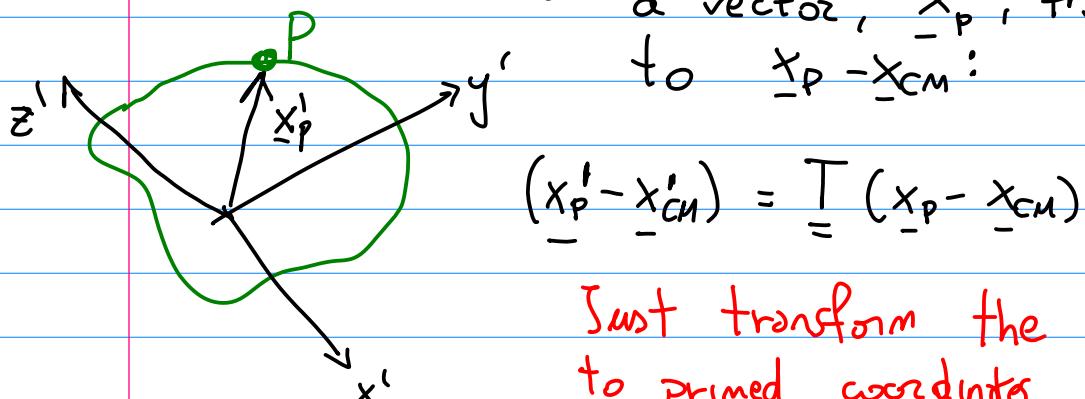
Keep in mind this:



We can write the position of a point, P , on the rigid body as:

$$\underline{x}_P = \underline{x}_{cm} + \underbrace{(\underline{x}_P - \underline{x}_{cm})}_{\text{relative position}} < \text{obviously.}$$

Now, in the body frame, P will be identified by a vector, \underline{x}'_P , that can be related to $\underline{x}_P - \underline{x}_{cm}$:



Just transform the vector $\underline{x}_P - \underline{x}_{cm}$ to primed coordinates.

Now, in this specific case, the "primed" body-fitted frame is fixed in the center of mass, so we have $\underline{x}_{CM}' = 0$

$$\Rightarrow \underline{x}_P' = \underline{T} (\underline{x}_P - \underline{x}_{CM})$$

\downarrow Relative position in the lab frame

Relative position in the body frame.

$$\Rightarrow \underline{x}_P = \underline{x}_{CM} + (\underline{x}_P - \underline{x}_{CM}) = \underline{x}_{CM} + \underline{T}^{-1} \underline{x}_P'$$

IMPORTANT

Property of transformation (& rotation) matrices: $\underline{T}^{-1} = \underline{T}^T$

If we compute the velocity of point P,

$$\underline{v}_P = \frac{d}{dt} \underline{x}_P = \underline{v}_{CM} + \frac{d}{dt} (\underline{T}^{-1} \underline{x}_P') = \underline{v}_{CM} + \frac{d\underline{T}^{-1}}{dt} \underline{x}_P' + \underline{T}^{-1} \frac{d\underline{x}_P'}{dt}$$

this is zero! For a rigid body,
 \underline{x}_P' is a fixed quantity

$$\Rightarrow \underline{v}_P = \underline{v}_{CM} + \frac{d\underline{T}^{-1}}{dt} \underline{x}_P' \quad \Rightarrow \quad \underline{v}_P = \underline{v}_{CM} + \boxed{\frac{d\underline{T}^{-1}}{dt} \underline{T}^{-1}} (\underline{x}_P - \underline{x}_{CM})$$

From rigid body theory, we know that

$$\underline{v}_P = \underline{v}_O + \boxed{\underline{\omega} \wedge} (\underline{r} - \underline{r}_O)$$

We found that

$$\frac{d\underline{T}^{-1}}{dt} \underline{T}^{-1} = \underline{\omega} \wedge = \underline{\Omega} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

" $\underline{\omega}_\lambda$ " = "cross" matrix and is emisymmetric: $\underline{\Omega} = -\underline{\Omega}^T$

Anyway:

$$\frac{d \underline{T}}{dt} \underline{T}^{-1} = \underline{\Omega}$$

Let's find σ property for this... consider again the velocity of P:

$$\begin{aligned} \underline{v}_P &= \frac{d}{dt} \underline{x}_P = \frac{d}{dt} \underline{x}_{CM} + \frac{d}{dt} (\underline{T}^{-1} \underline{x}'_P) = \underline{v}_{CM} + \frac{d \underline{T}^{-1}}{dt} \underline{x}'_P + \underline{T}^{-1} \frac{d}{dt} \underline{x}'_P = \\ &= \underline{v}_{CM} + \frac{d \underline{T}^{-1}}{dt} \underline{T} (\underline{x}_P - \underline{x}_{CM}) + \underline{T}^{-1} \frac{d}{dt} [\underline{T} (\underline{x}_P - \underline{x}_{CM})] = \\ &= \underline{v}_{CM} + \frac{d \underline{T}^{-1}}{dt} \underline{T} (\underline{x}_P - \underline{x}_{CM}) + \underline{T}^{-1} \left[\frac{d \underline{T}}{dt} (\underline{x}_P - \underline{x}_{CM}) \right] + \cancel{\underline{T}^{-1} \underline{T} (\underline{v}_P - \underline{v}_{CM})}^1 \end{aligned}$$

Simplify \underline{v}_P and \underline{v}_{CM} , and you get:

$$\left[\frac{d \underline{T}^{-1}}{dt} \underline{T} + \underline{T}^{-1} \frac{d \underline{T}}{dt} \right] (\underline{x}_P - \underline{x}_{CM}) = 0 \quad \forall P$$

\Rightarrow PROPERTY:

$$\boxed{\frac{d \underline{T}^{-1}}{dt} \underline{T} = - \underline{T}^{-1} \frac{d \underline{T}}{dt}}$$

If you think about it, this is kind of obvious, because

$$\frac{d \underline{T}^{-1}}{dt} \underline{T} = \underline{\Omega} \text{ emisymmetric...} \Rightarrow (\underline{\Omega})^T = \underline{\Omega}^T = -\underline{\Omega}$$

Let's go back & use this property:

$$\frac{d \underline{\underline{T}}}{dt} = -\underline{\underline{T}}^{-1} \frac{d \underline{\underline{T}}}{dt} = \underline{\underline{\Omega}}$$

$$\Rightarrow \frac{d \underline{\underline{T}}}{dt} = -\underline{\underline{T}} \underline{\underline{\Omega}}$$

From this property, we'll be able to find $\dot{\phi}, \dot{\theta}, \dot{\psi}$. We just have to expand $\underline{\underline{T}}$ using the chain rule:

$$\frac{d \underline{\underline{T}}}{dt} = \frac{d}{dt} \left[\underline{\underline{T}}_{\psi}(\psi) \underline{\underline{T}}_{\theta}(\theta) \underline{\underline{T}}_{\phi}(\phi) \right] =$$

$$= \left[\frac{d \underline{\underline{T}}_{\psi}}{d\psi} \underline{\underline{T}}_{\theta} \underline{\underline{T}}_{\phi} \right] \dot{\psi} + \left[\underline{\underline{T}}_{\psi} \frac{d \underline{\underline{T}}_{\theta}}{d\theta} \underline{\underline{T}}_{\phi} \right] \dot{\theta} + \left[\underline{\underline{T}}_{\psi} \underline{\underline{T}}_{\theta} \frac{d \underline{\underline{T}}_{\phi}}{d\phi} \right] \dot{\phi} = -\underline{\underline{T}} \underline{\underline{\Omega}}$$

These are 9 relations, that we can use to find the angular kinematics (aka $\dot{\phi}, \dot{\theta}, \dot{\psi}$)

First, we shall use Maxima (or Mathematica or Maple... or do that by hand...) to find all these matrices.

$$\underline{\underline{T}} = \underline{\underline{T}}_{\psi} \underline{\underline{T}}_{\theta} \underline{\underline{T}}_{\phi} = \begin{pmatrix} \cos(\psi) \cos(\theta) & \sin(\phi) \cos(\psi) \cos(\theta) + \cos(\phi) \sin(\psi) & \sin(\phi) \sin(\psi) - \cos(\phi) \cos(\psi) \sin(\theta) \\ -\sin(\psi) \cos(\theta) & \cos(\phi) \cos(\psi) \cos(\theta) - \sin(\phi) \sin(\psi) \sin(\theta) & \cos(\phi) \sin(\psi) \sin(\theta) + \sin(\phi) \cos(\psi) \sin(\theta) \\ \sin(\theta) & -\sin(\phi) \cos(\theta) & \cos(\phi) \cos(\theta) \end{pmatrix}$$

$$\frac{d \underline{\underline{T}}_{\psi}}{d\psi} = \begin{pmatrix} -\sin(\psi) & \cos(\psi) & 0 \\ -\cos(\psi) & -\sin(\psi) & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad \frac{d \underline{\underline{T}}_{\theta}}{d\theta} = \begin{pmatrix} -\sin(\theta) & 0 & -\cos(\theta) \\ 0 & 0 & 0 \\ \cos(\theta) & 0 & -\sin(\theta) \end{pmatrix} \quad \frac{d \underline{\underline{T}}_{\phi}}{d\phi} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -\sin(\phi) & \cos(\phi) \\ 0 & -\cos(\phi) & -\sin(\phi) \end{pmatrix}$$

Then, the full system is:

$$\frac{d\mathbf{I}}{dt} = \begin{pmatrix} -\sin(\psi) \cos(\theta) & \cos(\phi) \cos(\psi) - \sin(\phi) \sin(\psi) \sin(\theta) & \cos(\phi) \sin(\psi) \sin(\theta) + \sin(\phi) \cos(\psi) \\ -\cos(\psi) \cos(\theta) & -\sin(\phi) \cos(\psi) \sin(\theta) - \cos(\phi) \sin(\psi) & \cos(\phi) \cos(\psi) \sin(\theta) - \sin(\phi) \sin(\psi) \\ 0 & 0 & 0 \end{pmatrix} \dot{\psi}$$

$$+ \begin{pmatrix} -\cos(\psi) \sin(\theta) & \sin(\phi) \cos(\psi) \cos(\theta) & -\cos(\phi) \cos(\psi) \cos(\theta) \\ \sin(\psi) \sin(\theta) & -\sin(\phi) \sin(\psi) \cos(\theta) & \cos(\phi) \sin(\psi) \cos(\theta) \\ \cos(\theta) & \sin(\phi) \sin(\theta) & -\cos(\phi) \sin(\theta) \end{pmatrix} \dot{\theta}$$

$$+ \begin{pmatrix} 0 & \cos(\phi) \cos(\psi) \sin(\theta) - \sin(\phi) \sin(\psi) & \sin(\phi) \cos(\psi) \sin(\theta) + \cos(\phi) \sin(\psi) \\ 0 & -\cos(\phi) \sin(\psi) \sin(\theta) - \sin(\phi) \cos(\psi) & \cos(\phi) \cos(\psi) - \sin(\phi) \sin(\psi) \sin(\theta) \\ 0 & -\cos(\phi) \cos(\theta) & -\sin(\phi) \cos(\theta) \end{pmatrix} \dot{\phi} = -\underline{\underline{T}} \underline{\underline{\Omega}} =$$

$$\underline{\underline{\omega}} = \begin{pmatrix} \omega_x (\sin(\phi) \cos(\psi) \sin(\theta) - \cos(\phi) \sin(\psi)) - \omega_y (\cos(\phi) \cos(\psi) \sin(\theta) - \sin(\phi) \sin(\psi)) & \omega_x (\cos(\phi) \cos(\psi) \sin(\theta) - \sin(\phi) \sin(\psi)) + \omega_z \cos(\psi) \cos(\theta) & -\omega_x (-\sin(\phi) \cos(\psi) \sin(\theta) - \cos(\phi) \sin(\psi)) - \omega_y \cos(\psi) \cos(\theta) \\ \omega_z (\sin(\phi) \sin(\psi) \sin(\theta) - \cos(\phi) \cos(\psi)) - \omega_y (-\cos(\phi) \sin(\psi) \sin(\theta) - \sin(\phi) \cos(\psi)) & \omega_x (-\cos(\phi) \sin(\psi) \sin(\theta) - \sin(\phi) \cos(\psi)) - \omega_z \sin(\psi) \cos(\theta) & \omega_y \sin(\psi) \cos(\theta) - \omega_x (\sin(\phi) \sin(\psi) \sin(\theta) - \cos(\phi) \cos(\psi)) \\ \omega_z \sin(\phi) \cos(\theta) + \omega_y \cos(\phi) \cos(\theta) & \omega_z \sin(\theta) - \omega_x \cos(\phi) \cos(\theta) & -\omega_y \sin(\theta) - \omega_x \sin(\phi) \cos(\theta) \end{pmatrix}$$

Now, the best idea is probably to project this on the $\hat{i}, \hat{j}, \hat{k}$ axes. I will just consider some individual relations instead.

For instance, entry (3,1) is very simple, and reads:

$$(13) \Rightarrow \cancel{\dot{\phi}} + \cos \theta \dot{\theta} + \cancel{\omega \dot{\phi}} = \omega_z \sin \phi \cos \theta + \omega_y \cos \phi \cos \theta$$

$$\Rightarrow \dot{\theta} = \omega_y \cos \phi + \omega_z \sin \phi$$

[Note: for $\phi = 0$, $\dot{\theta} = \omega_y \checkmark$]

$$\text{Entry (3,2)}: \cancel{\dot{\phi}} + \sin \theta \sin \phi \dot{\theta} - \cos \phi \cos \theta \dot{\phi} = \omega_z s\theta - \omega_x c\phi c\theta$$

$$\Rightarrow \cos \phi \cos \theta \dot{\phi} = \sin \phi \sin \theta [\omega_y c\phi + \omega_z s\phi] - \omega_z s\theta + \omega_x c\phi c\theta$$

$$\cos \phi \cos \theta \dot{\phi} = \omega_x c\phi c\theta + \omega_y (\sin \phi \sin \theta c\phi) + \omega_z (\sin \phi \sin \theta s\phi - s\theta)$$

$$\Rightarrow \dot{\phi} = \omega_x + (\omega_y \sin \phi - \omega_z \cos \phi) \tan \theta$$

[Note: for $\theta = 0$, $\dot{\phi} = \omega_x \checkmark$]

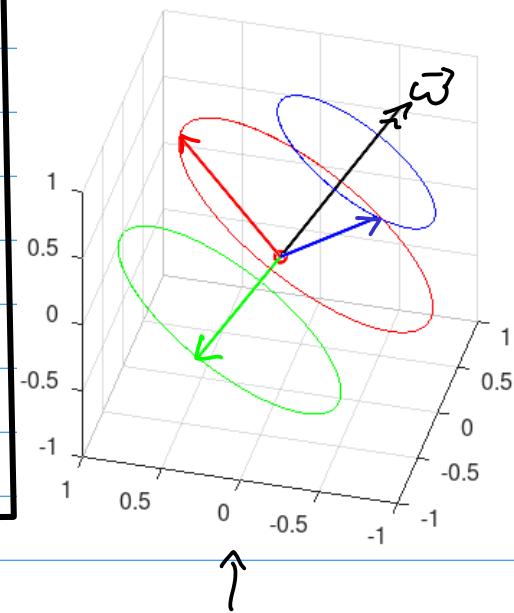
$$\begin{aligned}
 \text{Entry (1,1): } & -s\psi c\theta \dot{\psi} - c\psi s\theta \dot{\theta} + \cancel{o} \dot{\phi} = \\
 & = -\omega_z [s\phi c\psi s\theta + c\phi s\psi] - \omega_y [c\phi c\psi s\theta - s\phi s\psi] \\
 & = -s\psi c\theta \dot{\psi} - c\psi s\theta [\omega_y \cos\phi + \omega_z \sin\phi] \\
 \Rightarrow & -s\psi c\theta \dot{\psi} = c\psi s\theta \cancel{c\phi \omega_y} + c\psi s\theta \cancel{s\phi \omega_z} - \omega_z [s\phi c\psi s\theta + c\phi s\psi] \\
 & \quad | \\
 & = -c\phi s\psi \omega_z + s\phi s\psi \omega_y
 \end{aligned}$$

$$\dot{\psi} = [\omega_z \cos\phi - \omega_y \sin\phi] / \cos\theta$$

Note: for $\theta=0, \phi=0$
 $\Rightarrow \dot{\psi} = \omega_z \checkmark$

Summary of the equations

$$\left\{
 \begin{array}{l}
 \dot{\phi} = \omega_x + (\omega_y \sin\phi - \omega_z \cos\phi) \tan\theta \\
 \dot{\theta} = \omega_y \cos\phi + \omega_z \sin\phi \\
 \dot{\psi} = [\omega_z \cos\phi - \omega_y \sin\phi] / \cos\theta
 \end{array}
 \right.$$



NUMERICAL VERIFICATION

NUMERICAL IMPLEMENTATION

Octave/ Matlab

```

close all
clear
clc

% Initialize axes
I0 = [1; 0; 0];
J0 = [0; 1; 0];
K0 = [0; 0; 1];

% Angular velocity
om_x = 1.0;
om_y = -2.0;
om_z = 3.0;

om = [om_x; om_y; om_z];

% ----- Time loop -----
% Initialize axes
I = I0;
J = J0;
K = K0;

t_vect = linspace(0,3,10000);
dt = t_vect(2) - t_vect(1);

phi = 0.0;
theta = 0.0;
psi = 0.0;

figure
for t_ID = 1:numel(t_vect)

    phi_dot = om_x + (om_y*sin(phi) - om_z*cos(phi))*tan(theta);
    theta_dot = om_y*cos(phi) + om_z*sin(phi);
    psi_dot = (om_z*cos(phi) - om_y*sin(phi))/cos(theta);

    % Integrate one step (forward Euler)
    phi = phi + dt*phi_dot;
    theta = theta + dt*theta_dot;
    psi = psi + dt*psi_dot;

```

(1)

```

% Build transformation matrix from new angles
T = zeros(3,3);

Tphi = [1, 0, 0;
         0, cos(phi), sin(phi);
         0, -sin(phi), cos(phi)];

Tth = [cos(theta), 0, -sin(theta);
        0, 1, 0;
        sin(theta), 0, cos(theta)];

Tpsi = [cos(psi), sin(psi), 0;
        -sin(psi), cos(psi), 0;
        0, 0, 1];

T = Tpsi*(Tth*Tphi);

% The rotation matrix is the transpose of the transformation matrix
R = T';

% Rotate axes from the global ones (beware gimbal lock!)
I = R*I0;
J = R*J0;
K = R*K0;

% Save for plotting
I_save(:,t_ID) = I;
J_save(:,t_ID) = J;
K_save(:,t_ID) = K;

```

(2)

```

if mod(t_ID, 100) == 0 % Plot every 100 time steps [om_x; om_y; om_z];
    % ----- Time loop -----
    hold off
    plot3([0, I(1)], [0, I(2)], [0, I(3)], 'r', 'LineWidth', 2); %es
    hold on
    plot3([0, J(1)], [0, J(2)], [0, J(3)], 'g', 'LineWidth', 2);
    plot3([0, K(1)], [0, K(2)], [0, K(3)], 'b', 'LineWidth', 2);
    plot3([0, om_x], [0, om_y], [0, om_z], 'k', 'LineWidth', 2);
    t_vect = linspace(0,3,10000);
    dt = t_vect(2) - t_vect(1);

    % Superimpose a trace
    plot3(I_save(1, 1:t_ID), I_save(2, 1:t_ID), I_save(3, 1:t_ID), 'r', 'LineWidth', 1)
    plot3(J_save(1, 1:t_ID), J_save(2, 1:t_ID), J_save(3, 1:t_ID), 'g', 'LineWidth', 1)
    plot3(K_save(1, 1:t_ID), K_save(2, 1:t_ID), K_save(3, 1:t_ID), 'b', 'LineWidth', 1)
    psi = 0.0;

    figure
    for t_ID = 1:numel(t_vect)

        phi_dot = om_x + (om_y*sin(phi) - om_z*cos(phi));
        theta_dot = om_y*cos(phi) + om_z*sin(phi);
        psi_dot = (om_z*cos(phi) - om_y*sin(phi))/cos(theta);

        % Integrate one step (forward Euler)
        phi = phi + dt*phi_dot;
        theta = theta + dt*theta_dot;
        psi = psi + dt*psi_dot;
    end
end

```

(3)

