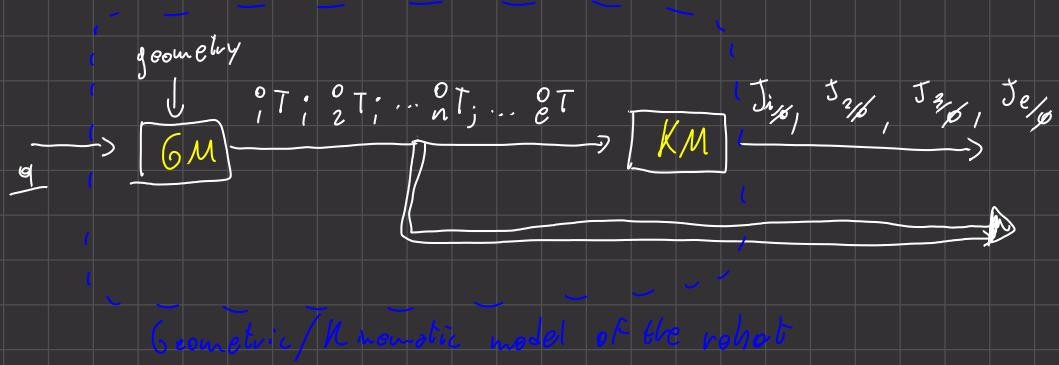


25/09/23

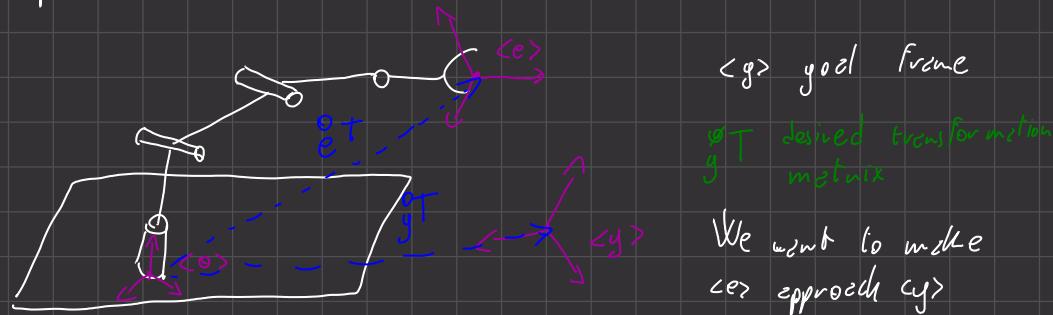
$$J_{1/6}(q) = J_{1/6}(\overset{0}{T}(q); \overset{1}{T}(q); \overset{2}{T}(q); \dots \overset{n}{T}(q))$$

The info that we need to describe 2 Jacobian is not only the angle of every joint but also the distances and all geometrical distances and all the transf.  
matrix contain this information. Inside  $J$  there are all these geometric distances. To contain all the parameters of the geometry. This connects to the geometric  
model.

We can now show a new block that can take the info from the 6M.



We have the means for building two blocks. One to know where the frames start and on the basis of that we can work on the kinematic model. The Jacobians provide the link from the cartesian to joint space?

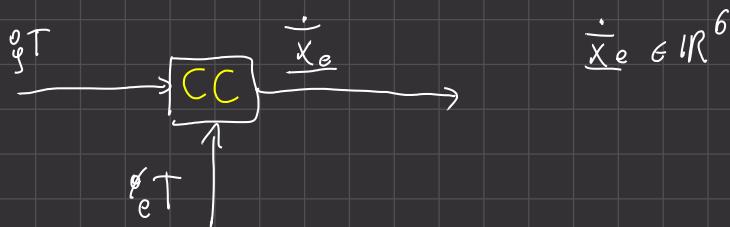


$c_0$  depends on how the robot moves and is constrained by that. Let's think for a moment that  $c_0$  was not constrained. First we want to understand how a frame not constrained should move to reach  $c_3$ .

If I want to go to position of  $c_3$  with  $c_0$  how should I move? For this we should have a vector in the direction from  $c_0$  to  $c_3$ . But keep in mind that this is not the only solution because any vector orthogonal to the distance of  $c_0$  would turn but get closer if there is a component towards  $c_3$ . In addition to this examples there are infinite vectors that work. We will also see solutions for rotation.

The block outputs the desired velocity for the frame

We have introduced the Cartesian Controller



This outputs the desired cartesian velocity ( $\dot{x}$ ), but we need the desired joint velocity vector ( $\dot{q}$ ).  
We have seen that Jacobian maps from joint to cartesian space.

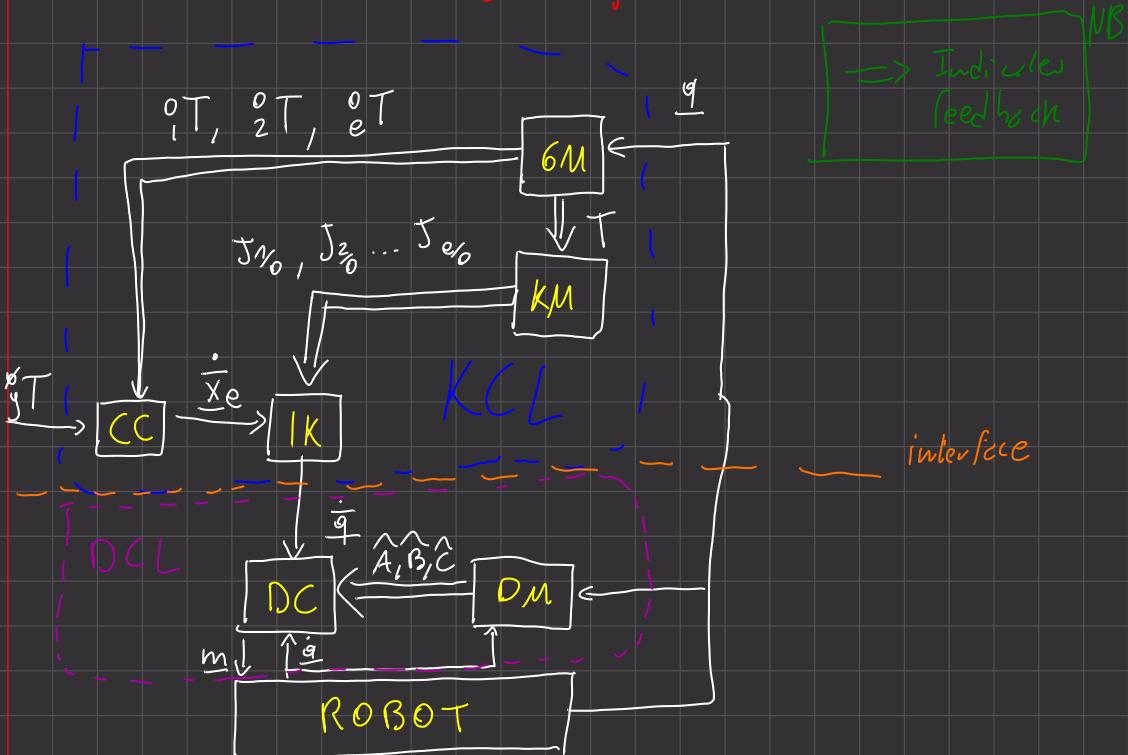
We will see that the inverse of Jacobian will offer in some way the inverse, even if the Jacobian won't be square. This block will output the velocity that best approximate the desired velocity.

If we have 2 q-degree of freedom what we can do in a subset of the 6-dof space.

In the other hand if we have 2 more than 6-dof what sometimes we cannot guarantee the velocity required. For example think when your arm is stretched and you want to extend even more. You can't.

We are still missing the inverse kinematics block. That provides the best approximation to this problem. Remember the workspace of the robot.

The overall control scheme is the following:



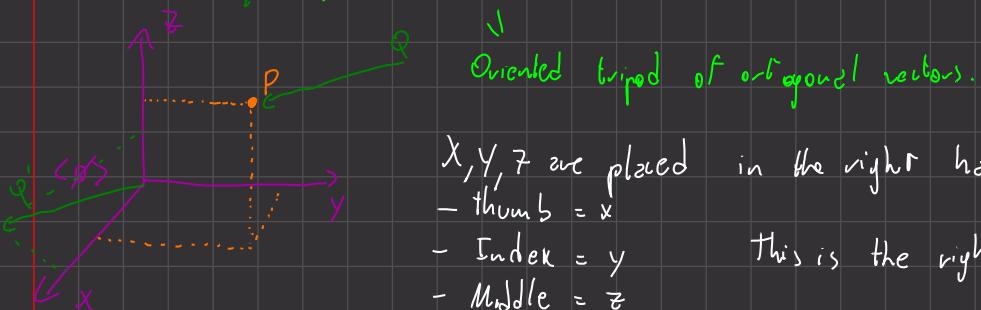
The robot exerts torque ( $m$ ). From the robot we measure  $q$  and  $\dot{q}$ . The first thing we do in the DCL is building a controller that builds the desired velocity - the objective of the DCL is to follow as close as possible the reference velocity. Once you have done this you need to drive the ee to the goal. Now I need to generate  $\dot{q}$ . From yearlong we work

position of the frames and with KM their movement with friction. Feeding this to the zidhun we see how the system should move in space. We get the derived velocity that gets mapped to the derived velocity and the inverse kinematic.

TODO

## CH. 7 Geometric Fundamentals

### Coordinate Systems (Frame)



With the concept of frame we can describe vectors and points. For example

$$\phi P = \begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix}$$

The components are the length of the vectors that allow us to reach  $P$  in the reference of this particular frame.

We also have vectors.

If we imagine  $Q$  in  $\phi$  of the frame it's easy to identify the components. This is called a projected vector.

$$\phi(P-Q) = \phi P - \phi Q$$

TODO ?

The point on which I apply the force is very important, but when I want to know the distance the point on which is applied is not very important.

Often we can also say:

Projected vector ???

$$\phi(P-Q) = \phi P - \phi Q \triangleq \phi \underbrace{v}_{\substack{\uparrow \\ \text{the vector that joins } Q \text{ to } P}}_{P,Q}$$

the vector that joins  $Q$  to  $P$

We can now recall the gressman rule to join point and vectors

$$\left| \begin{array}{c} \text{Gressman Rule} \\ \hline \phi P = \phi Q + \underbrace{v}_{P,Q} \end{array} \right|$$

In the frame unit vectors are immediately representable:

$$\begin{array}{ll} x & \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T \\ y & \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T \\ z & \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \end{array}$$

Let's now consider two frames



Set of information needed to describe how  $c_b$  is with respect to  $c_2$  is

$$\left\{ \begin{bmatrix} i_b \\ j_b \\ k_b \end{bmatrix}, \begin{bmatrix} i_{c_2} \\ j_{c_2} \\ k_{c_2} \end{bmatrix} \right\}$$

unit vectors of  $b$  projected on  $c_2$  to code the rotation of frame  $c_b$  in respect to  $c_2$ .

This information is encoded into the rotation or orientation matrix.

## Orientation matrices (Rotation Matrices)

Let's consider a vector  $v$  and represent it as a function of  $c_b$ .

$$v = i_b x_b + j_b y_b + k_b z_b \quad i, j, k \text{ form a base.}$$

If now I perform projection of  $v$  on  $c_2$

$$v = i_{c_2} x_{c_2} + j_{c_2} y_{c_2} + k_{c_2} z_{c_2}$$

Now representing as a vector-matrix multiplication we get:

$$v = \begin{bmatrix} i_{c_2} & j_{c_2} & k_{c_2} \end{bmatrix} \begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix}$$

rotated frame  
 referring frame  $\longrightarrow_b^b R = \text{ROTATION MATRIX}$

$$v = b R v \quad \text{This is another way of seeing the multiplication.}$$

In Matlab this will be represented like  $v - R - b$ .

It's important that in the vector we include the name of the frame on which the vector should be projected on. For example  $b$ -distance.

So a multiplication could be:

$$z\text{-distance} = {}^aR_b \quad b\text{-distance}.$$

$R$  is a ortho-normal matrix  
columns are unit length  
orthogonal to each other

$${}^aR {}^bR^T = I_{n \times n}$$

Given this the matrix has some properties:

Reading list:

$$\det(AB) = \det(A) \det(B)$$

$$\det(A) = \det(A^T)$$

Being ortho-normal means:

$$\begin{cases} {}^aR {}^bR^T = I_{3 \times 3} \\ {}^aR^T {}^bR = I_{3 \times 3} \end{cases} \Rightarrow {}^bR^T = {}^aR^{-1} = {}^bR$$

↑ this because of  $\det(R) = 1$

This is important because:

$$\forall M: MM^T = I \Rightarrow \det(M) = \pm 1$$

For Rot. Mat the determinant equals 1  $\rightarrow \det({}^aR) = 1$

↓  
this to preserve the direction meaning  
the right handed frame.

For example

$$\begin{bmatrix} x & y & z \text{ on } b \\ -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \rightarrow \det = -1$$

We can see that  $y$  and  $z$  are the same while  $x$  is the opposite, but this is impossible if we consider 2 right handed frames. So this cannot happen.  $\det({}^aR) = -1$  does not respect right-hand rule.

From these properties we can also conclude that the transpose is the inverse.

It also makes sense if you think about it geometrically.

$$SO(n) = \left\{ R \in \mathbb{R}^{n \times n} : R^T R = I_{n \times n}, \underbrace{\det(R) = +1} \right\}$$

↑  
Special orthogonal group

Special for this limitation

This group has some properties:

N.B

topo on why? MULT DIV

- **CLOSURE:** given any two elements  $R_1, R_2$  of  $SO(n)$  then also  $R_1 \cdot R_2 \in SO(n)$
- **IDENTITY:** there exists an element  $I$  of  $SO(n)$  such that  $IR \in SO(n)$  for any  $R$
- **INVERSE:** for any  $R \in SO(n)$  there exists  $R^{-1}$  such that  $RR^{-1}=I$
- **ASSOCIATIVITY:** given three elements  $R_1, R_2, R_3$  in  $SO(n)$  it follows that  $(R_1 R_2) R_3 = R_1 (R_2 R_3)$

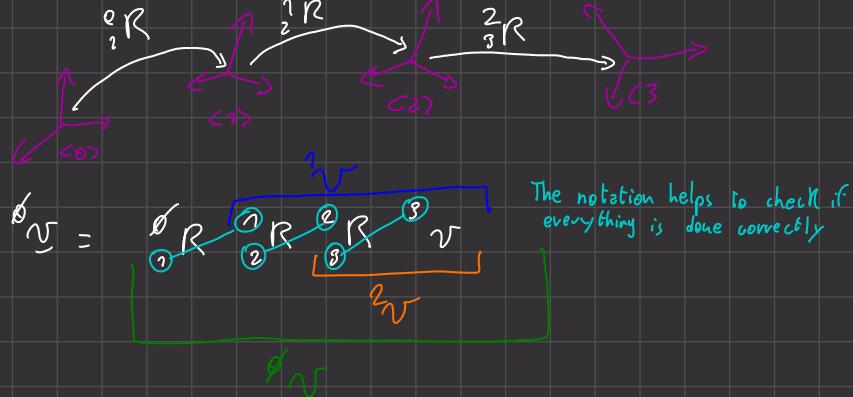
A member of  $SO(3)$  is a rotation Matrix

A consideration about  ${}^2_b R^T = {}^b_2 R$  is that this is also equivalent to:

$${}^b_2 R = \begin{bmatrix} {}^2_{1b} & {}^2_{2b} & {}^2_{3b} \end{bmatrix} = \begin{bmatrix} b_{11}^+ \\ b_{12}^+ \\ b_{13}^+ \\ b_{21}^+ \\ b_{22}^+ \\ b_{23}^+ \\ b_{31}^+ \\ b_{32}^+ \\ b_{33}^+ \end{bmatrix}$$

On the transpose you will see what you see on columns now on the rows.

Our manipulator is composed by a tree of frames:



Here we have some reference frames and suppose that we know some rotation matrices.

postmultiplication

$$\text{How do you compute: } {}^q R = {}^h R^+ {}^i R^T {}^j R {}^h R = \underbrace{{}^q R}_{} \underbrace{{}^i R}_{} \underbrace{{}^j R}_{} \underbrace{{}^h R}_{} \xrightarrow{\text{postmultiplication}}$$

We can also do the same thing following this path:

premultiplication

$${}^q R = \underbrace{{}^h R^+}_{} \underbrace{{}^i R^T}_{} \underbrace{{}^j R}_{} \underbrace{{}^h R}_{} .$$



To get from  $P$  to  $\bar{P}$  the position of all the frames is an essential parameter. I could for example use another vector  $P$  and now the description of  $P$  becomes  $P + \bar{P}$ . This can be done iteratively going back towards  $<0>$ .

$${}^K P = (P - O_n)$$

$${}^0 P = {}^0 O_h + {}^0 (P - O_n)$$

$${}^0 P = {}^0 O_h + {}^h R {}^h P$$

## Homogeneous Coordinates

$${}^K \bar{P} = \begin{bmatrix} {}^h P \\ 1 \end{bmatrix} \in \mathbb{R}^4$$

$${}^h \bar{P} = {}^0 T {}^h \bar{P}$$

$${}^h T \triangleq \begin{bmatrix} {}^0 R & | & {}^0 O_h \\ \hline 0_{3 \times 1} & | & 1 \end{bmatrix} \in \mathbb{R}^{4 \times 4}$$

$$\text{So if: } d_0 \begin{bmatrix} {}^0 R & | & {}^0 O_h \\ \hline 0_{3 \times 1} & | & 1 \end{bmatrix} \begin{bmatrix} {}^h P \\ 1 \end{bmatrix} = \begin{bmatrix} {}^0 R {}^h P + {}^0 O_h \\ 1 \end{bmatrix}$$

$${}^0 T = {}^0 \bar{P}$$

## Inverse transformation matrix

The increase in dimension is a gimmick to obtain the previous formula with a matrix by vector multiplication.

We know that:

$$\begin{matrix} {}^z b \\ {}^z b \end{matrix}^T = \begin{matrix} {}^b b \\ {}^b b \end{matrix}^T$$

N.B.

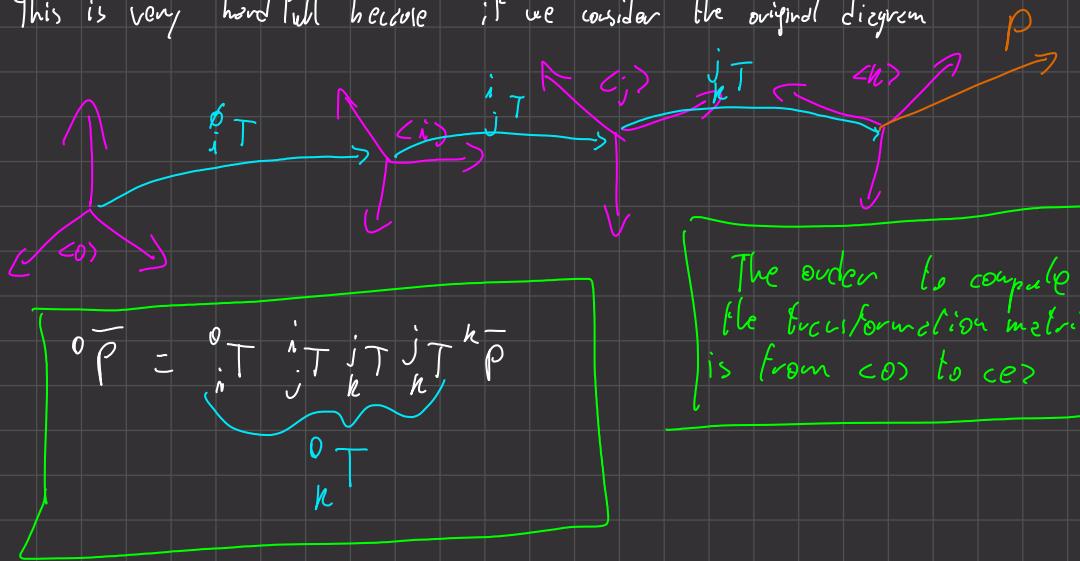
BUT

$$\begin{matrix} {}^z b \\ {}^b b \end{matrix}^T \neq \begin{matrix} {}^z b \\ {}^z b \end{matrix}^T$$

$$\boxed{\begin{matrix} {}^z b \\ {}^b b \end{matrix}^T = \begin{bmatrix} {}^z b^T & {}^z R^T & {}^z O_b \\ 0_{3 \times 1} & I \end{bmatrix}}$$

The origin of  ${}^z b$  on  ${}^b b$  would be the opposite vector and to put it in the right direction I need a minus in front.

This is very hard full because if we consider the original diagram



It's also possible to describe a vector using homogeneous coordinates. It's done by just adding 0 to the vector.

$$p = \begin{bmatrix} {}^n p \\ 0 \end{bmatrix}_{4 \times 1}$$

This because the origin in this case does not come into play.

The inertial frame it's not different from any other frame.

## Three parameter representation of rotation matrices

$\overset{3}{\underset{b}{R}}$

- Rotation matrices are orthonormal. vectors are orthogonal to each other  
modulus = 1 for every vector.

So I have:

$$\left\{ \begin{array}{l} \overset{2}{\lambda_b^T} \overset{2}{\lambda_b} = 1 \\ \overset{2}{j_b^T} \overset{2}{j_b} = 1 \\ \overset{2}{k_b^T} \overset{2}{k_b} = 1 \end{array} \right. \quad \left\{ \begin{array}{l} \downarrow \text{dot product} \\ \overset{2}{\lambda_b^T} \overset{2}{j_b} = 0 \\ \overset{2}{j_b^T} \overset{2}{k_b} = 0 \\ \overset{2}{k_b^T} \overset{2}{\lambda_b} = 0 \end{array} \right.$$

$\overset{2}{\lambda_b}$  satisfy all these equations

We have 6 constraints and 9 numbers so we only have to pick 3 that are unknown.

For this reason we can talk about 3 number representation of rotation matrices.

The easiest of these representation is called

Euler angles

they express that any rotation can be represented by three rotations between 3 different axes.

We can choose between:

- extrinsic rotation
- intrinsic rotation

- I describe the final orientation with rotation around a fixed coordinate system. This unfortunately is not so easy to implement because when we multiply R. matrices we are not doing intrinsic rotation. In this case we first rotate around the original x, then around the original y and finally around the original z.
- this is easier to implement. We first rotate around x, then the rotated y and then the rotated z. When we multiply with Rot. Matrices we are doing intrinsic rotations.

Intrinsic angles can be divided in:

- proper Euler angles

$z-x-z$ ;  $y-x-y$ ;  $x-z-x$ ; ...

these represent the sequence of axes around which you do the rotation

- Tait-Bryan angles

$x-y-z$ ;  $z-y-x$ ; ...

$$z - y^1 - x^1$$

This is the most used one and we will use this one

$${}^3 R = R_z(\psi) \ R_y(\theta) \ R_x(\phi)$$

YAW

PITCH

ROLL

The open stresses that we are talking about intrinsic rotation

It's possible to demonstrate that an extrinsic rotation x-y-z is the same as an intrinsic rotation z-y-x.

02/10/23

$${}^3 R = R_z(\psi) \ R_y(\theta) \ R_x(\phi) =$$

$$= \begin{bmatrix} C\psi C\theta & -s\psi C\phi + c\psi s\theta s\phi & s\psi s\phi + c\psi c\phi s\theta \\ S\psi C\theta & C\psi C\phi + s\phi s\theta s\psi & -c\psi s\phi + s\theta s\psi c\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix}$$

Not needed to know by heart

It's useful to know the structure to know the inverse.

$$\theta = \text{atan2}(-r_{31}, \sqrt{r_{11}^2 + r_{21}^2}) \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$$

software function to compute atan2 on a quadrant instead of two.

$$\text{If } \cos\theta \neq 0 : \quad \psi = \text{atan2}(r_{21}, r_{11}) \\ \phi = \text{atan2}(r_{32}, r_{31})$$

If  $\cos\theta = 0$  it's an interesting configuration because leads to a singularity.

In the sequence of rotation if we rotate in pitch of  $\pm \frac{\pi}{2}$  the x axis becomes the same as the z axis of the original frame. We have infinite combination of first and third axis that end up representing the same output. We get a singularity when the pitch is  $\pm \frac{\pi}{2}$ .

Suppose now that I want to control roll to a certain angle, when i get  $\frac{\pi}{2}$  in pitch yaw and roll become the same. It's an ill defined situation.

The use of Euler angles is fine but we must have regard for pitch being close to  $\frac{\pi}{2}$

# Vector operations

## - Linear combination

$$\underline{v} = c_1 \underline{v}_1 + c_2 \underline{v}_2 + \dots + c_n \underline{v}_n$$

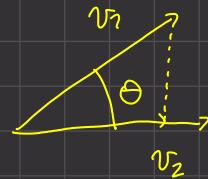
If they are linear independent they can form the base of your n-based dimension space.

If I take into consideration the projection of a frame I also need to project on the same frame all the components.

$${}^2 \underline{v} = c_1 {}^2 \underline{v}_1 + c_2 {}^2 \underline{v}_2 + \dots + c_n {}^2 \underline{v}_n$$

## - Scalar product

$$(\underline{v}_1 \cdot \underline{v}_2) = (\underline{v}_2 \cdot \underline{v}_1) = |\underline{v}_1| |\underline{v}_2| \cos \theta$$



$$(\underline{v}_1 \cdot \underline{v}_2) = {}^2 \underline{v}_1^\top {}^2 \underline{v}_2$$

projected on  
c2)

$$\left[ \begin{array}{l} \text{I can consider } (\underline{v}, \cdot) = (\underline{v}^\top)_{1 \times 3} \\ \text{linear operation} \end{array} \right]$$

## - Vector product

$$(\underline{v}_1 \wedge \underline{v}_2) = (\underline{v}_1 \times \underline{v}_2) = \begin{cases} \text{modulus} & |\underline{v}_1 \times \underline{v}_2| = |\underline{v}_1| |\underline{v}_2| \sin(\theta) \xrightarrow{\text{minimum angle}} \\ \underline{v}_1 \times \underline{v}_2 \perp (\underline{v}_1, \underline{v}_2) & \end{cases}$$

The direction is orthogonal to both  $\underline{v}_1$  and  $\underline{v}_2$  according to the right hand rule

### - Properties:

$$\bullet \text{ Anti-commutative: } (\underline{v}_1 \times \underline{v}_2) = -(\underline{v}_2 \times \underline{v}_1)$$

$$\bullet {}^2(\underline{v}_1 \times \underline{v}_2) = {}^2[\underline{v}_1 \times] {}^2 \underline{v}_2 = \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix} {}^2 \underline{v}_2$$

Components of  ${}^2 \underline{v}_1$

For this version

↳ This matrix is skew-symmetric

$$\bullet {}^2 [\underline{v}_1 \times]^T = - {}^2 [\underline{v}_1 \times]$$

axial vector

sometimes this can also be represented like:  $[\underline{v}_1 \times] = S(\underline{v}_1)$

$$\bullet \text{ Inverse mapping: } \text{vex operation} \rightarrow \text{vex}(S(\underline{v}_1)) = \underline{v}_1$$

$$\boxed{\forall M: M \in \mathbb{R}^{3 \times 3} / M = -M^T}$$
$$M \underline{y} = \text{vex}(M) \times \underline{y}$$

Any skew-symmetric has an axial vector. Vex is needed to get x, y and z

Power of skew-symmetric  $3 \times 3$  real matrices associated with a unit vector

$$h \in \mathbb{R}^{3 \times 1}$$

$$\underline{[h \ x]} = \underline{h \ x}$$

$$\underline{[h \ x]}^2 = (\underline{h} \ \underline{h}^\top - I)$$

After the  $q^{\text{th}}$  power they repeat from the first one.

$$\underline{[h \ x]}^3 = - \underline{[h \ x]}$$

$$\underline{[h \ x]}^4 = - \underline{[h \ x]}^2$$

$$\underline{[h \ x]}^5 = \underline{[h \ x]}$$

$$\underline{[h \ x]}^6 = \underline{[h \ x]}^2$$

⋮

- In general we state that:

$$[\underline{h \ x}]^{2i+1} = (-1)^i [\underline{h \ x}]$$

Rule for odd numbers  
(skew symmetric)

$$[\underline{h \ x}]^{2i+2} = (-1)^i (\underline{h} \ \underline{h}^\top - I)$$

Rule for even numbers  
(symmetric)

$(I - \underline{h} \underline{h}^\top)$  projects a vector on the plane orthogonal to  $\underline{h}$  (TODOLICHT)

- $(I - \underline{h} \underline{h}^\top)$  is the projection operator defined by  $\underline{h}$

$$(I - \underline{h} \underline{h}^\top) \underline{h} = \emptyset \text{ because } (\underline{h} - \underline{h} \underbrace{\underline{h}^\top \underline{h}}_{\sim}) = \emptyset$$

- true  $([\underline{h \ x}]^{2i+2}) = 0$  for  $\rightarrow$  sum of all the elements on the main diagonal in a matrix

$$t_r([\underline{h \ x}]^{2i+2}) = (-1)^{(i+1)} \cdot 2$$

- $A \in \mathbb{R}^{n \times n}$

$$A = A^\top \text{ symmetrical}$$

$$A = -A^\top \text{ skew symmetrical}$$

$$\forall A \in \mathbb{R}^{n \times n} \quad A = \underbrace{\frac{A+A^\top}{2}}_{\text{symmetric}} + \underbrace{\frac{A-A^\top}{2}}_{\text{skew-symmetric}}$$

no symm  
skew-symm

# Linear Operators between Vectors

Consider: frame  $\mathbf{b}$

- any projected vector  ${}^b\mathbf{v}$
- ${}^b\mathbf{w}$  subject to  ${}^b\mathbf{w} = {}^b\mathbf{L} {}^b\mathbf{v}$

$${}^2\mathbf{w} = {}^2\mathbf{R} {}^b\mathbf{w} = {}^2\mathbf{R} {}^b\mathbf{L} {}^b\mathbf{v} = \boxed{{}^2\mathbf{R} {}^b\mathbf{L} {}^b\mathbf{R}^T} {}^2\mathbf{v} = {}^2\mathbf{L} {}^2\mathbf{v}$$

$$\boxed{{}^2\mathbf{L} = {}^2\mathbf{R} {}^b\mathbf{L} {}^b\mathbf{R}^T}$$

This is the operation to change the operator from one frame to another.

For example:

vector product operator

$${}^b(\mathbf{w} \times \mathbf{v}) = {}^b[\mathbf{w} \times] \mathbf{v}$$

$${}^2(\mathbf{w} \times \mathbf{v}) = {}^2\mathbf{R} {}^b[\mathbf{w} \times] {}^b\mathbf{R}^T {}^2\mathbf{v} = {}^2[\mathbf{w} \times] {}^2\mathbf{v}$$

06/10/23

Matrix exponential operator

$${}^b\mathbf{L} = e^A \quad A \in \mathbb{C}^{3 \times 3}$$

$${}^b\mathbf{L} = \sum_{k=0}^{\infty} \frac{({}^bA)^k}{k!}$$

As any other linear operator we can write:

$${}^b\mathbf{w} = {}^b\mathbf{L} {}^b\mathbf{v} = e^A {}^b\mathbf{v}$$

Now we want to know  ${}^2\mathbf{L}$ :

$${}^2\mathbf{L} = {}^2\mathbf{R} {}^b\mathbf{L} {}^b\mathbf{R}^T = {}^2\mathbf{R} e^A {}^b\mathbf{R}^T$$

Now if we substitute the definition of the exponential:

$$= {}^2\mathbf{R} \left( \sum_{k=0}^{\infty} \frac{({}^bA)^k}{k!} \right) {}^b\mathbf{R}^T = \sum_{k=0}^{\infty} \frac{{}^2\mathbf{R} ({}^bA)^k {}^b\mathbf{R}^T}{k!}$$

$$= \sum_{k=0}^{\infty} \frac{({}^2\mathbf{R} {}^bA {}^b\mathbf{R}^T)^k}{k!} \quad \text{this step is possible because :}$$

$$\overbrace{({}^2\mathbf{R} {}^bA {}^b\mathbf{R}^T) ({}^2\mathbf{R} {}^bA {}^b\mathbf{R}^T) \dots ({}^2\mathbf{R} {}^bA {}^b\mathbf{R}^T) ({}^2\mathbf{R} {}^bA {}^b\mathbf{R}^T)}^{k \text{ times}} \quad \text{Identity}$$

To conclude

$$L = e^{\theta R} A R^T = e^{\theta A}$$

## Properties of the matrix exponential

- $M e^{\theta M} = e^{\theta M} M \quad \theta \in \mathbb{R}$
- $(e^{\theta M})^T = e^{\theta M^T}$
- $(e^{\theta M})^{-1} = e^{-\theta M}$

## Exponential representation of rotation matrices.

Every  $e^{\theta[\underline{h}\underline{x}]} \in SO(3)$ , so this is a rotation matrix

To prove this:

$$(e^{\theta[\underline{h}\underline{x}]})^T e^{\theta[\underline{h}\underline{x}]} = e^{\theta[\underline{h}\underline{x}]^T} e^{\theta[\underline{h}\underline{x}]} = \\ = e^{-\theta[\underline{h}\underline{x}]} e^{\theta[\underline{h}\underline{x}]} = I_{3 \times 3}$$

This determinant is either +1 or -1. If  $\theta=0$  the determinant is +1. There is a continuity rule also for the determinant. It cannot jump from +1 to -1. For this continuity rule we can state that the determinant is always +1.

So we have  $\det = 1$  and the product for the transpose is  $I_{3 \times 3}$  so it's a Rotation Matrix.

## Angle axis parameters.

$\theta \in \mathbb{R}$  angle

$n \in \mathbb{R}^{3 \times 1}$ ,  $|n|=1$  axis

together they are the angle-axis representation of the rotation matrix  $e^{\theta[\underline{h}\underline{x}]}$

$n$  gives you the axis around which to rotate and  $\theta$  gives you the quantity.

## Rotation Vector

$g = \theta n \in \mathbb{R}^{3 \times 1}$  is called the rotation vector associated to the rotation matrix  $e^{\theta[\underline{h}\underline{x}]}$

Angle axes vector

$$\underline{v} = \begin{bmatrix} h \\ \theta \end{bmatrix} \in \mathbb{H}^{[1 \times 2]}$$

This is also called 4D-vector associated to the matrix  $e^{\theta [\underline{h} \times]}$

Let's now consider how the exponential is computed:

$$e^{\theta [\underline{h} \times]} = \sum_{k=0}^{\infty} \frac{(\theta \underline{h})^k}{k!} =$$

We split even and odd powers of this infinite summation:

$$= \mathbb{I}_{3 \times 3} + \left( \theta - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} \dots \right) [\underline{h} \times] + \left( \frac{\theta^2}{2!} - \frac{\theta^4}{4!} + \dots \right) [\underline{h} \times]^2$$

We can do this for the properties of skew-symmetrical matrices

Rodrigues Formula

$$= \mathbb{I}_{3 \times 3} + \underbrace{\sin \theta [\underline{h} \times]}_{\text{the first sum converges to } \sin} + \underbrace{(\gamma - \cos \theta) [\underline{h} \times]^2}_{\text{the second one converges to } (\gamma - \cos \theta)}$$

N. B.

$\underline{h}$  is already a unit vector

From the Rodrigues formula we can find the mere mapping from a rotation matrix to the angle-axes mapping.

On the diagonal you get the square of the components.

Equivalent Angle-axes inverse problem

$$R \xrightarrow{?} \theta, \underline{h}$$

$$(\underline{h} \underline{h}^T - \mathbb{I})$$

$$\text{Let's look at the } \text{tr}(R) = \text{tr} \left( \mathbb{I} + \sin \theta [\underline{h} \times] + (\gamma - \cos \theta) [\underline{h} \times]^2 \right) =$$

$$= 3 + 0 + (\gamma - \cos \theta)(\gamma - 3) = 3 - 2(\gamma - \cos \theta) = \gamma + 2\cos \theta$$

$$\text{So: } \theta = \cos^{-1} \left( \frac{\text{tr}(R) - \gamma}{2} \right)$$

Like this we can find more than one parameter

For example if:

$\theta = 0$  then  $\underline{h}$  is arbitrary

$\theta = \pi$  then

Let's try it:

$$e^{\hat{\theta}[\underline{h}x]} = \mathbb{I}_{3 \times 3} + 2(\underline{h}\underline{h}^T - \mathbb{I}) = 2\underline{h}\underline{h}^T - \mathbb{I}_{3 \times 3} =$$

$$= \begin{bmatrix} 2h_1^2 - 1 & 2h_1h_2 & 2h_1h_3 \\ 2h_1h_2 & 2h_2^2 - 1 & 2h_2h_3 \\ 2h_1h_3 & 2h_2h_3 & 2h_3^2 - 1 \end{bmatrix} = R$$

From this we can conclude that the modulus of  $h$  =

$$|h_i| = \pm \sqrt{\frac{r_{ii} + 1}{2}} \quad \forall i = 1, 2, 3$$

To find the sign of this vector we can select a random component  $i \in \{1, 2, 3\}$ .

$$h_i = \pm \sqrt{\frac{r_{ii} + 1}{2}}$$

$$\text{The generic } h_j = \text{sign}(h_i) \text{sign}(r_{ij}) \sqrt{\frac{r_{jj} + 1}{2}} \quad \forall j \in \{1, 2, 3\} \setminus i$$

If you pick  $i = 1$   
then you try for 2 and 3

What about all the other cases?

Let's consider  $R - R^T$

First we expand  $R$  and  $R^T$  with the Rodriguez formula:  
we are left with

$$R - R^T = \sin \theta [\underline{h}x] - \sin \theta [\underline{h}x]^T$$

$$\begin{aligned} R - R^T &= \cancel{\mathbb{I}_{3 \times 3}} + \sin \theta [\underline{h}x] + (1 - \cos \theta)[\underline{h}x]^2 \\ &\quad - \cancel{-\mathbb{I}_{3 \times 3}} - \sin \theta [\underline{h}x] - (1 - \cos \theta)[\underline{h}x]^2 = \\ &= 2 \sin \theta [\underline{h}x] \end{aligned}$$

$$\frac{R - R^T}{2} = \sin \theta [\underline{h}x] \quad \text{so} \quad \sin \theta \underline{h} = \text{vex}\left(\frac{R - R^T}{2}\right)$$

This extracts the  
axis vector from the  
skew symmetric Matrix

The goal at the end will be to make the rotation matrix converge to  $\mathbb{I}$  and the norm of the rotation vector to 0.

Both yaw-pitch-roll and rotation matrices show that the mapping from  $R$  to these 2/3 parameters is not unique

## Euler Theorem

It states that any 3d frame can be aligned with any other arbitrary fixed frame by one rotation  $\theta$  around one axis.