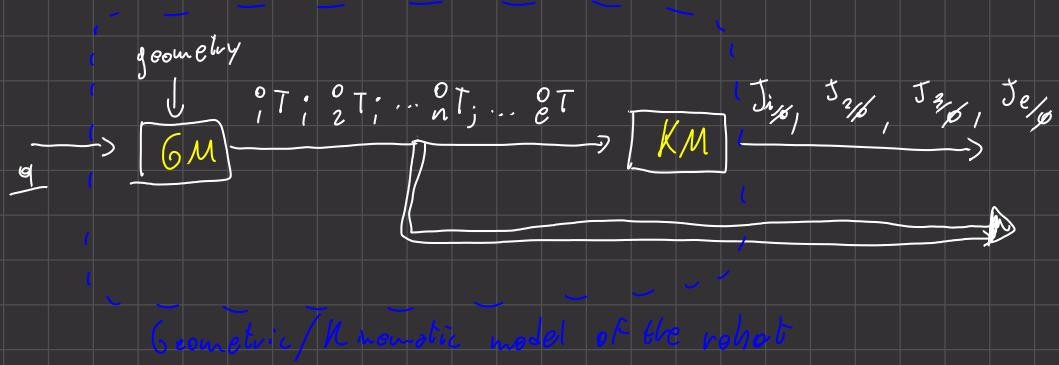


25/09/23

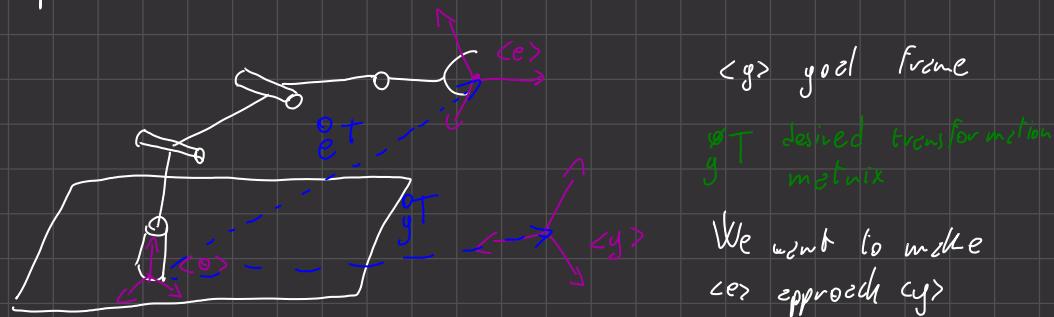
$$J_{1/6}(q) = J_{1/6}(\overset{0}{T}(q); \overset{1}{T}(q); \overset{2}{T}(q); \dots \overset{n}{T}(q))$$

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?

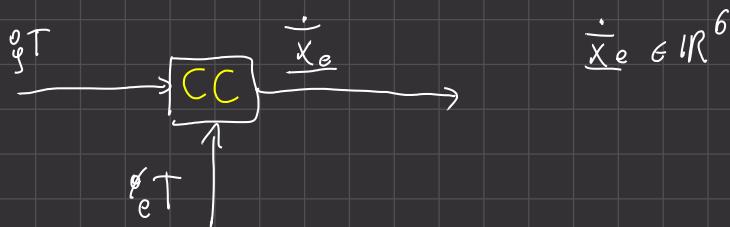


c_3 depends on how the robot moves and is constrained by that. Let's think for a moment that c_3 was not constrained. First we want to understand how a frame not constrained should move to reach c_3 .

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.

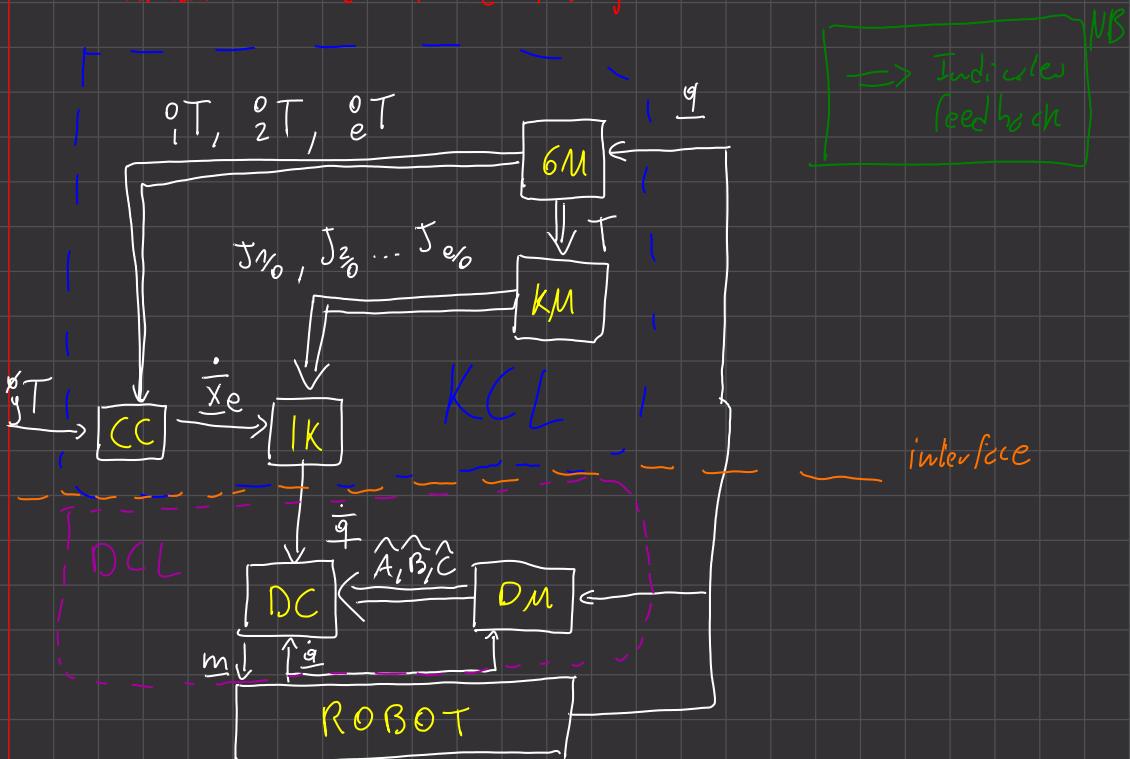
We will see that the inverse of Jacobian will offer in some way the inverse, even if the Jacobian won't be square. This block will output the velocity that best approximate the desired velocity.

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:



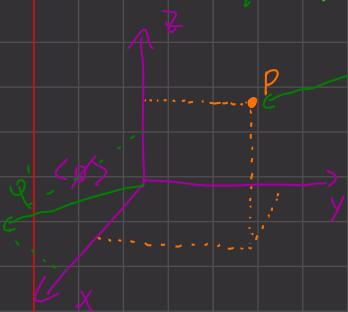
The robot exerts torque (m). From the robot we measure q and \dot{q} . The first thing we do in the DCL is building a controller that builds the desired velocity - the objective of the DCL is to follow as close as possible the reference velocity. Once you have done this you need to drive the ee to the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



Oriented tripod of orthogonal vectors.

x, y, z are placed in the right hand rule.
 - thumb = x
 - Index = y
 - Middle = z

This is the right handed 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 ϕ 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

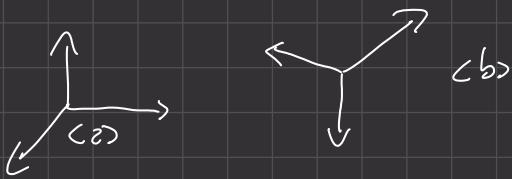
We can now recall the gressman rule to join point and vectors

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

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 vector-matrix multiplication we get:

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

↗ c_2
 ↗ b
 ↗ R = ROTATION MATRIX
 ↗ v
 ↗ c_b
 ↗ v

retained frame
referring frame

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

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 $(R_1 R_2) R_3 = R_1 (R_2 R_3)$

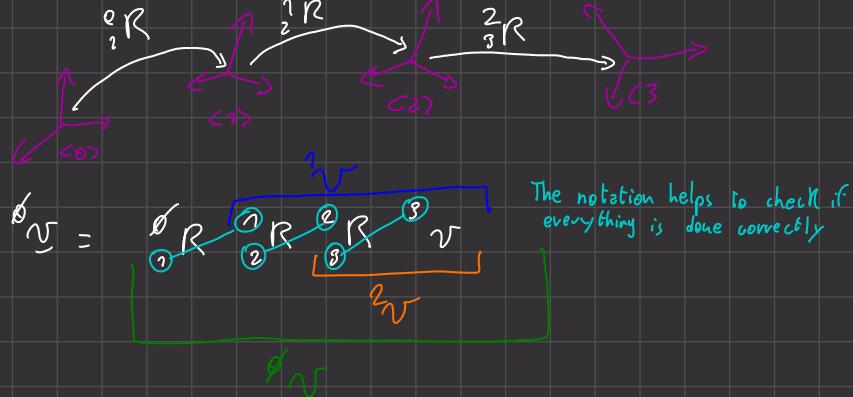
A member of $SO(3)$ is a rotation Matrix

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

$$\text{How do you compute: } {}^q_h R = {}^h_h R^+ {}^i_h R^T {}^j_h R \xrightarrow{\text{postmultiplication}} {}^q_h R {}^i_h R {}^j_h R$$

We can also do the same thing following this path:

premultiplication

$${}^q_h R = \xrightarrow{\text{premultiplication}} {}^q_h R^+ {}^i_h R^T {}^j_h R$$



To get from P to 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 $P + P$. This can be done iteratively going back towards $<0>$.

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

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

$${}^0 P = {}^0 O_n + {}^0 R {}^n P$$

Homogeneous Coordinates

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

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

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

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

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

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.

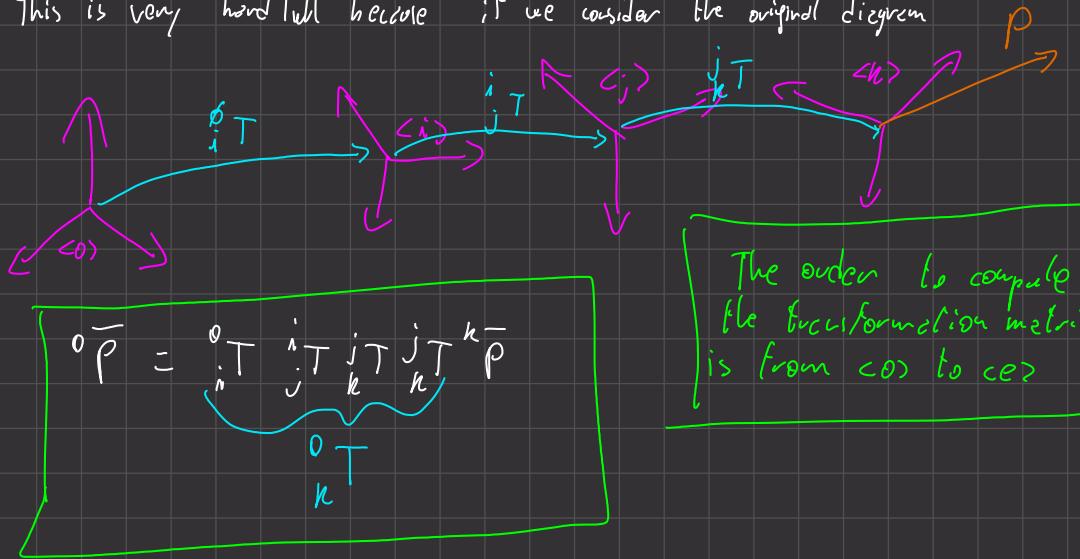
BUT

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

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

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

→ Pronounced Euler

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

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

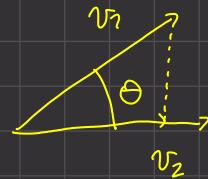
If they are linear independent they can form the base of your n-based dimension space.

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 3×3 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^{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:

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

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

$$\underbrace{({}^2\mathbf{R} {}^bA {}^b\mathbf{R}^T)}_{\text{Identity}} \underbrace{({}^2\mathbf{R} {}^bA {}^b\mathbf{R}^T)}_{\text{Identity}} \dots \underbrace{({}^2\mathbf{R} {}^bA {}^b\mathbf{R}^T)}_{\text{Identity}} \underbrace{({}^2\mathbf{R} {}^bA {}^b\mathbf{R}^T)}_{\text{Identity}}$$

k times

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

To prove this:

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

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

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

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

\underline{h} gives you the axis around which to rotate and θ gives you the quantity.

Rotation Vector

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

Angle axes vector

$$\underline{v} = \begin{bmatrix} h \\ \theta \end{bmatrix} \in \mathbb{R}^{3+1}$$

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

Let's now consider how the exponential is computed:

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

We split even and odd powers of this infinite summation:

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

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

Rodrigues Formula

$$= \mathbb{I}_{3 \times 3} + \underbrace{\sin \underline{\theta} [\underline{h}x]}_{\text{the first sum converges to } \sin} + \underbrace{(\underline{1}-\cos \underline{\theta}) [\underline{h}x]^2}_{\text{the second one converges to } (\underline{1}-\cos \underline{\theta})}$$

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

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

$$\text{Let's look at the } \text{tr}(R) = \text{tr} \left(\mathbb{I} + \sin \underline{\theta} [\underline{h}x] + (\underline{1}-\cos \underline{\theta}) [\underline{h}x]^2 \right) =$$

$$= 3 + 0 + (\underline{1}-\cos \underline{\theta})(\underline{1}-3) = 3 - 2(\underline{1}-\cos \underline{\theta}) = \underline{2} + 2\cos \underline{\theta}$$

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

Like this we can find more than one parameter

For example if:

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

$\underline{\theta} = \underline{\pi}$ then

Let's try it:

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

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

From this we can conclude that the modulus of h =

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

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

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

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

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

This extracts the
axis vector from the
skew symmetric Matrix

The goal at the end will be to make the rotation matrix converge to 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

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

These are defined by $\varphi \in \mathbb{R}^{dn}$

$$\underline{q} \triangleq \begin{bmatrix} \mu \\ \varepsilon \end{bmatrix} \quad \text{where} \quad \begin{cases} \mu \in \mathbb{R} \\ \varepsilon \in \mathbb{R} \end{cases}$$

$\|\mathbf{q}\|^2 = 1 \leftarrow$ this is why it's called unit quaternion

The relation between this and the rotation vector are the following:

First let's recall:

$$\text{J} \sin(\varphi + \beta) = \sin \varphi \cos \beta + \sin \beta \cos \varphi$$

$$\text{II} \cos(\varphi + \beta) = \cos \varphi \cos \beta - \sin \varphi \sin \beta$$

Let's now consider:

$$\varphi = \beta = \frac{\theta}{2}$$

Now we get what appears in the Rodriguez formula:

$$\sin \theta = 2 \sin \frac{\theta}{2} \cos \frac{\theta}{2}$$

$$\text{II } \cos \theta = \cos^2 \frac{\theta}{2} - \sin^2 \frac{\theta}{2}$$

Name if we do

$$\cos \theta = \cos^2 \frac{\theta}{2} - \sin^2 \frac{\theta}{2} + \cos^2 \left(\frac{\theta}{2} \right) - \cos^2 \left(\frac{\theta}{2} \right)$$

1

$$2 + \cos\theta = 2\cos^2\left(\frac{\theta}{2}\right)$$

Now if we add and subtract $\sin^2\left(\frac{\theta}{2}\right)$ here we get

$$1 - \cos \theta = 2 \sin^2 \left(\frac{\theta}{2} \right)$$

$$R = I + \sin\theta [\underline{h}x] + (1 - \cos\theta) [\underline{h}x]^2 = \\ = I + 2 \sin\left(\frac{\theta}{2}\right) \cos\left(\frac{\theta}{2}\right) [\underline{h}x] + 2 \sin^2\left(\frac{\theta}{2}\right) [\underline{h}x]^2$$

Now if we define $\begin{cases} \mu \triangleq \cos\frac{\theta}{2} \\ \varepsilon \triangleq \sin\frac{\theta}{2} h \end{cases}$ we can write: ???

$\sin\theta$ goes inside $[\underline{h}x]$

$R = I + 2\mu[\underline{\varepsilon}x] + 2[\underline{\varepsilon}x]^2$

→ this is a not singular representation of the rotation matrix.

If $\theta=0$ then $\varepsilon=0$. There is no undefined of the axis vector \underline{h} because it's always well defined.

Inverse mapping: from $R \rightarrow$ quaternions

To compute the inverse mapping we do:

$$\text{tr}(R) = 3 + 0 + 2 \text{tr}([\underline{\varepsilon}x]^2)$$

$$\text{tr}(R) = 3 + 2 \text{tr}(\underline{\varepsilon} \underline{\varepsilon}^T - \underline{\varepsilon}^T \underline{\varepsilon} I)$$

$$\text{tr}(R) = 3 + 2 (\|\underline{\varepsilon}\|^2 - 3\|\underline{\varepsilon}\|^2)$$

$$\text{tr}(R) = 3 - 4\|\underline{\varepsilon}\|^2$$

$$\text{tr}(R) = 4\mu^2 - 1$$

$$\mu = \frac{1}{2} \sqrt{\text{tr}(R) + 1}$$

let's now work on $\underline{\varepsilon}$

$$R - R^T = \cancel{I} + 2\mu[\underline{\varepsilon}x] + 2[\underline{\varepsilon}x]^2 + \\ - \cancel{I} + 2\mu[\underline{\varepsilon}x] - 2[\underline{\varepsilon}x]^2$$

$$R - R^T = q\mu[\underline{\varepsilon}x]$$

$$\frac{R - R^T}{2} = 2\mu[\underline{\varepsilon}x]$$

$$\underline{\varepsilon} = \frac{1}{2\mu} \text{vex}\left(\frac{R - R^T}{2}\right) \quad \text{if } \mu \neq 0$$

ϵ eigenvector of the matrix $\frac{R \cdot R^T}{2}$ and then pick the one with eigenvalue = 1 if $\mu=0$
 \hookrightarrow eigenvectors

Flipping the signs will give you the other eigenvalues?

Recalling euler angles, there is the problem that if we have the sequence $x-y-x$ if $y=0$ then we have two times the same x

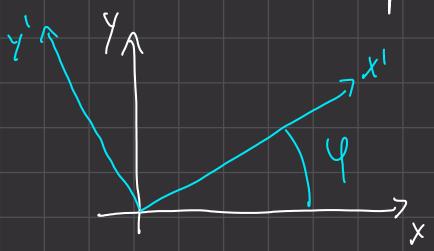
Let's also recall how an elementary rotation is written. So for example a rotation around the z axis of φ is:

$$R_z(\varphi) = \begin{bmatrix} \cos \varphi & -\sin \varphi & 0 \\ \sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_y(\theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix}$$

$$R_x[\phi] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix}$$

To find them for example:



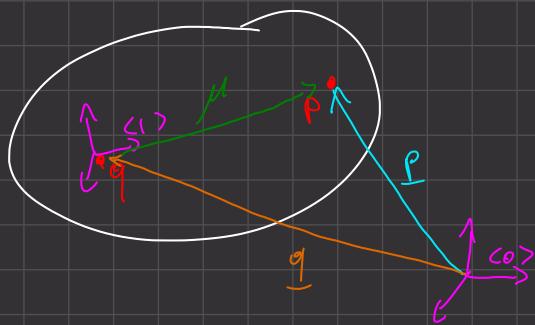
Right hand rule. the rotation is out of the page.

So there are the x' coordinates on the x axis

These are the y' on the y axis z' on the z .

Angular velocity vector

Let's consider a generic object with specific points where we have some frames



$$p = q + \mu$$

If we now start projecting:

$${}^0\dot{p} = {}^0\dot{q} + {}^0\dot{\mu} = {}^0\dot{q} + {}^1R{}^0\dot{\mu}$$

Now we want to consider:

$${}^0\dot{p} \triangleq \underbrace{\frac{d}{dt}({}^0R{}^1p)}_{\text{II}} \neq {}^1R \underbrace{\frac{d}{dt}{}^1p}_{\text{I}}$$

this is \downarrow the correct one because we are switching from frame ϕ ?
In the second one we are calculating from the frame ψ ?

This ambiguity will be solved with ${}^0Vp/0$ (this means seen from ϕ)

Using this notation the previous becomes:

$${}^0\dot{p} \triangleq \underbrace{\frac{d}{dt}({}^0R{}^1p)}_{\text{II}} \neq \underbrace{{}^1R \frac{d}{dt}{}^1p}_{\text{I}}$$

$$\quad \quad \quad {}^0Vp/0 \quad \quad \quad {}^1Vp/1$$

Let's continue:

$$\begin{aligned} {}^0\dot{p} &= {}^0\dot{q} + \frac{d}{dt}({}^0R{}^1\mu) \\ &= {}^0\dot{q} + {}^0R{}^1\dot{\mu} + {}^0R{}^1\ddot{\mu} \\ &= {}^0\dot{q} + \underbrace{{}^0R \underbrace{{}^0R^T {}^0R}_{{}^0I_{3x3}}}^{{}^0R} \dot{\mu} + {}^0R{}^1\ddot{\mu} \end{aligned}$$

0q is the position of frame
 \leftrightarrow to c_0

μ position from p to c_1

This tells us c_0 is changing in this term times respect to c_0 into account that c_1 is changing orientation to c_0

here μ changes in } this is ϕ if in the respect to c_1 than } rigid space of projected on c_0 } frame c_1

Now R is an element of $SO(3)$ so:

$$RR^T = I_{3 \times 3}$$

So if now I do:

$$\frac{d}{dt} ({}^0 R {}^0 R^T) = 0 = {}^0 \dot{R} {}^0 R^T + {}^0 R {}^0 \dot{R}^T$$
$$\underbrace{{}^0 \dot{R} {}^0 R^T}_{\text{skew-symmetric matrix}} = - {}^0 R {}^0 \dot{R}^T = - \underbrace{({}^0 \dot{R} {}^0 R^T)^T}_{\text{skew-symmetric matrix}}$$

We can always associate an axial vector to a skew symmetrical matrix.

$$({}^0 \dot{R} {}^0 R^T) = {}^0 \underline{\omega}_{1/0} \times \underline{\mu} \quad ???$$

Definition of angular velocity vector

$\forall R \in SO(3)$ there exists a unique $\omega \in \mathbb{R}^{3 \times 1}$ such

that

$${}^0 \dot{R} {}^0 R^T = [\underline{\omega}_{1/0}]$$

Stepdown equation

So if we have:

$${}^0 \dot{R} {}^0 R^T = [\underline{\omega}_{1/0} \times]$$

\Downarrow

inverse = transpose

$${}^0 \dot{R} = [\underline{\omega}_{1/0} \times] {}^0 R$$

$${}^0 \dot{R} = {}^0 R [\underline{\omega}_{1/0} \times]$$

$$[\underline{\omega}_{1/0}] = {}^0 R [\underline{\omega}_{1/0}] {}^0 R^T$$

Transformation of the operator

Example:

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

$\alpha \neq 0$

$$R_z(\alpha) = \begin{bmatrix} -\sin(\alpha) & -\cos(\alpha) & 0 \\ \cos(\alpha) & -\sin(\alpha) & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

If we directly calculate:

$$\dot{R}_2(\alpha) R_2^T(\alpha) = \dot{\alpha} \cdot \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} \omega x \\ 0 \\ 0 \end{bmatrix}$$

$$\text{So } \omega = \begin{bmatrix} 0 \\ 0 \\ \dot{\alpha} \end{bmatrix}$$

When you have the angular velocity (ω) you cannot directly integrate it but use something like the stepdown matrix.

Except when you are constantly rotating around one axis, meaning that I don't change axes in respect to which I rotate.

To better say in general you have to apply the stepdown for every infinitesimal rotation and compute ω .

Angular velocity properties and composition rules

$${}^0\dot{R} {}^0R^T = \begin{bmatrix} {}^0\omega_{0/1} \times \end{bmatrix} \xrightarrow{\text{N.B.}}$$

$${}^0\dot{R} \left[{}^1\dot{R} {}^0R^T \right] {}^0R^T \xrightarrow{\text{N.B.}} \begin{bmatrix} {}^0\omega_{0/1} \end{bmatrix}$$

$$= {}^0\dot{R} {}^1\dot{R} \left({}^0R^T {}^0R^T \right) = {}^0\dot{R} {}^1\dot{R} \xrightarrow{\text{N.B.}} A^T B = (B^T A)^T$$

$$= \left({}^0\dot{R} {}^1\dot{R} \right)^T = \underbrace{\left({}^0\dot{R} {}^0R^T \right)^T}_{\text{skew symmetric}} = - \underbrace{\left({}^0\dot{R} {}^0R^T \right)}_{\text{stepdown}} =$$

$$= - \left[{}^0\omega_{0/1} \times \right]$$

N.B.

$$\phi \omega_{0/1} = - \omega_{0/0}$$

We can conclude

The angular velocity is a relative concept

Let's see the composition:

$${}^0R = {}^1R {}^2R$$

$$\dot{{}^0R} {}^0R^T = \left[\frac{d}{dt} \left({}^1R {}^2R \right) \right] ({}^0R {}^1R)^T$$

$$= \left({}^0\dot{R} {}_2^1 R + {}^0R {}_2^1 \dot{R} \right) {}_2^1 R^T {}_1^0 R^T$$

$$= \underbrace{{}^0\dot{R} {}^0 R^T}_{\text{stepdown}} + {}^0R \underbrace{\left[{}_2^1 \dot{R} {}_2^1 R^T \right]}_{\text{stepdown}} {}_1^0 R^T$$

this would be:

$$\phi \omega_{z0} + \phi \begin{bmatrix} {}^1\omega_{x1} \\ {}^1\omega_{y1} \end{bmatrix}$$

$\underbrace{\text{Rotated to be on } \phi}$

From this follows:

$$\underline{\omega}_z = \underline{\omega}_{z0} + \underline{\omega}_{x1}$$

Galilean velocity composition

Angular velocity and the rotation vector

$$\underline{\omega} = \dot{\theta} \underline{h} + \sin \theta \underline{i} + (1 - \cos \theta) \underline{h} \times \underline{h}$$

\downarrow Base of a 3-dimensional space. $\Rightarrow \begin{cases} \underline{h} \text{ and } \underline{h} \text{ are orthogonal} \\ \text{because } \underline{h} \text{ is unitary in length and can only rotate, not change the length.} \end{cases}$

If $\underline{h} = 0$, means that the axis around which one frame is rotating respect to the other is not changing, and we can integrate.

Otherwise if $\underline{h} \neq 0$ integrating ω it does not give θ , and we are rotating around moving axes.

Mapping between ω and the 3-D axis vector

$$\underline{v} = \begin{bmatrix} \underline{h} \\ \theta \end{bmatrix}$$

This works only if

$$\cos \theta \neq 1$$

$$\underline{v} = \begin{bmatrix} -\frac{\sin \theta}{2(1 - \cos \theta)} [\underline{h} \times]^2 - \frac{1}{2} [\underline{h} \times] \\ \underline{h}^T \end{bmatrix} \underline{\omega}$$

You cannot integrate \underline{R} from the stepdown because otherwise you won't be in $SO(3)$. One solution could be to normalize the matrix. If you want to know the were orientation it could be convenient to find the mapping of your pitch-roll angles.

If we know how your pitch-roll relate in relation to ω then we are good.

When you know your pitch-roll you can now do a forward Euler integration. What you will get is another $so(3)$ element.

Angular velocity vector and time derivative of minimal parametrization vectors

$${}^2\omega_{b/2} = {}^bL_{b/2} \dot{\bar{\varphi}} \quad \bar{\varphi} = \begin{bmatrix} \psi \\ \theta \\ \phi \end{bmatrix}$$

N.B.

The order is
yaw-pitch-roll ($z-y-x$)

$$\begin{aligned} {}^2\omega_{b/2} &= \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \dot{\psi} + R_z(\psi) \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \dot{\theta} + R_z(\psi) R_y(\theta) \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \dot{\phi} \\ &= \begin{bmatrix} 0 & -s\psi & c\psi c\theta \\ 0 & c\psi & s\psi c\theta \\ 1 & 0 & -s\theta \end{bmatrix} \begin{bmatrix} \dot{\psi} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} \end{aligned}$$

$$\det(L_{b/2}) = -\cos(\theta) \leftarrow \text{This mapping becomes undefined whenever } \theta = \frac{\pi}{2} \text{ or better I cannot find the inverse mapping but from where I don't know the derivatives of } \psi, \theta, \phi$$

26/20

Unit quaternions and the angular velocity vector

$$\begin{cases} \dot{\mu} = -\frac{1}{2} \underline{\varepsilon}^T \underline{\omega} \\ \dot{\underline{\varepsilon}} = \frac{1}{2} (\mu \mathbb{I}_{3 \times 3} - [\underline{\varepsilon} \times]) \underline{\omega} \end{cases}$$

Inverse mapping

$$\underline{\omega} = 2(\mu \underline{\varepsilon} - \underline{\varepsilon} \mu) + 2 \underline{\varepsilon} \times \dot{\underline{\varepsilon}}$$

Keep in mind that doing a forward Euler integration you could spoil the relation:

$$\mu^2 + \|\underline{\varepsilon}\|^2 = 1$$

If you then use this not unit quaternion you would end up with a non rotation matrix so you need to normalize the unit quaternion before calculating the rotation matrix.

Euler angles do not have these problems, but they suffer from singularities.

For example to describe the rotation of an object to another with a unit quaternion is not too easy, while using yaw-pitch-roll it's easier.

23/20 - KINEMATICS

Revise rotation

Vectors can be described as pure geometrics so an abstract thing.

The geometric idea is useful to work models.

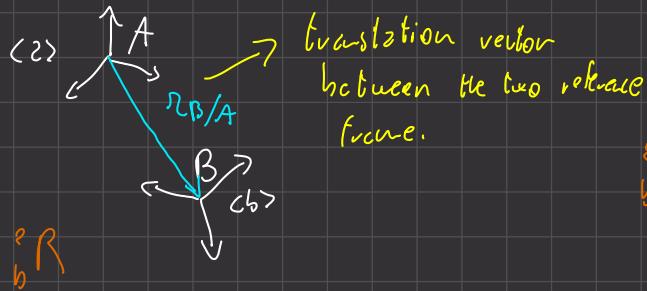
Then we have the algebraic representation

We are going to learn how to handle moving vectors in space.

We can observe these moving vectors from a frame called observer.

Each one of the body comprising a robot is home of one reference frame.

A, B, C are points
 $\{c_2, c_b\}$ are ref. frame



$b^l R$ this represent the axis of frame c_b as seen from frame c_2

The axis of frame c_2 will be described in two ways:

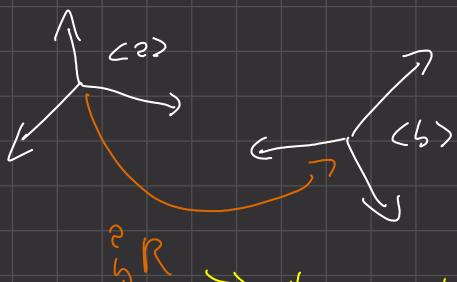
$$\left\{ \begin{array}{l} (\underline{i}_2, \underline{j}_2, \underline{k}_2) \\ (\underline{e}_1^2, \underline{e}_2^2, \underline{e}_3^2) \end{array} \right. \begin{array}{l} \text{in frame } l \\ \text{first vector} \end{array}$$

For example in frame B this have we:

$$\left\{ \begin{array}{l} (\underline{i}_2, \underline{j}_2, \underline{k}_2) \\ (\underline{e}_1^2, \underline{e}_2^2, \underline{e}_3^2) \end{array} \right.$$

So \mathbb{R} can be written as:

$${}^2 \mathbb{R} = \left[\left(e_i^1 \cdot e_j^2 \right) \right] = \begin{bmatrix} e_1^1 & e_1^2 \\ e_2^1 & e_2^2 \\ e_3^1 & e_3^2 \end{bmatrix} \quad {}^2 e_1^1 = {}^2 j_1 \quad {}^2 e_2^1 = {}^2 j_2 \quad {}^2 e_3^1 = {}^2 K_2$$



\rightarrow this is also a way to indicate how to rotate vectors.

We can also define an homogeneous transformation:

$${}^2 {}^b T = \begin{pmatrix} {}^2 {}^b R & {}^2 b/2 \\ 0 & 1 \end{pmatrix}$$

sometimes this is called H

Let's suppose that we get a vector $\underline{b_m}$.

We might be interested to transform this vector in frame 2

To do that first we have to write it in homogeneous coordinates:

$$\underline{\underline{b_m}} = \begin{bmatrix} \underline{b_m} \\ 1 \end{bmatrix}$$

$${}^2 {}^b T \underline{\underline{b_m}} = {}^2 \underline{\underline{b_m}} = \begin{bmatrix} {}^2 {}^b R \underline{b_m} + {}^2 R b/2 \\ 1 \end{bmatrix} \rightarrow \underline{\underline{b_m}}$$

Assumptions:

- All measurements are consistent
- Non consider relativistic effects

Let's consider a vector that is changing its length in time.

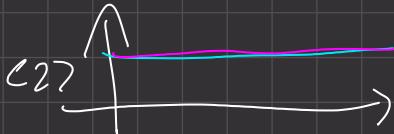
$\mu(t)$

Closed example

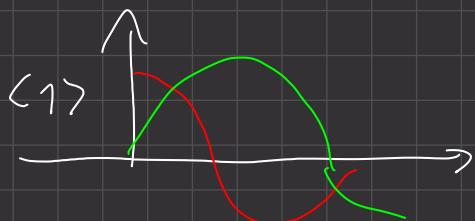
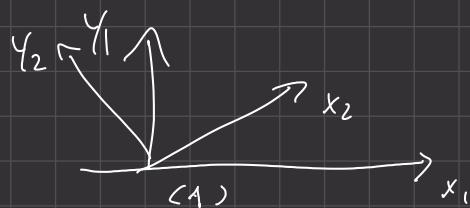


If we consider the frame of the axis the orientation

of the vectors is constant so if we map them



Now if I take another observer:



Now if I take the vectors taken from \mathcal{V} frame

$$\text{I get } \frac{d_2}{dt} x_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

If I take the derivatives from frame 1:

$$\frac{d}{dt} \underline{x}_2 = \begin{bmatrix} -\sin(\alpha) \\ \cos(\alpha) \end{bmatrix}$$

The derivatives are not absolute scalar functions

$$\mu(t) \rightarrow \frac{d}{dt} \underline{\mu}(t) \text{ in frame 2} = \frac{d}{dt} \sum_{i=1}^3 \underline{\mu}_i(t) e_i^2 \quad \left\{ x, i_1 + y_1 j_1 + z_1 k_1 \right\}$$

$$= \sum_{i=1}^3 \left(\frac{d^2}{dt^2} \underline{\mu}_i(t) \right) e_i^2 + \sum_{i=1}^3 \underline{\mu}_i(t) \cdot \frac{d^2}{dt^2} e_i^2$$

these are the same
common for all the observer. These are p
because represent how the frame
changes with respect
with himself.

equivalent

$$= \frac{d}{dt} \sum_{i=1}^3 {}^b \underline{\mu}_i(t) e_i^b = \sum_{i=1}^3 \left(\frac{d}{dt} {}^b \underline{\mu}_i(t) \right) e_i^b + \sum_{i=1}^3 {}^b \underline{\mu}_i(t) \frac{d}{dt} e_i^b$$

We can write that:

$${}^b \left(\frac{d}{dt} \underline{\mu} \right) = \begin{pmatrix} \frac{d}{dt} {}^b \mu_1 \\ \frac{d}{dt} {}^b \mu_2 \\ \frac{d}{dt} {}^b \mu_3 \end{pmatrix}$$

We know we want to compute the time derivative in frame 2 of a quantity expressed in frame b.

$$\frac{d^2}{dt^2} \underline{\mu}(t) = \frac{d^2}{dt^2} \sum_{i=1}^3 {}^b \underline{\mu}_i(t) e_i^b =$$

vector in b composed by derivatives of scalar functions that
are invariant to the reference frame.

$$= \sum_{i=1}^3 \left(\frac{d^2}{dt^2} {}^b \underline{\mu}_i(t) \right) e_i^b + \sum_{i=1}^3 {}^b \underline{\mu}_i(t) \left(\frac{d^2}{dt^2} e_i^b \right)$$

So now we can write:

$$\left(\begin{array}{c} \frac{d}{dt} \underline{\mu} \\ \hline \end{array} \right) = \left(\begin{array}{c} \frac{d}{dt} \underline{\mu}_1 \\ \frac{d}{dt} \underline{\mu}_2 \\ \frac{d}{dt} \underline{\mu}_3 \end{array} \right) = \frac{d^2}{dt^2} \left(\begin{array}{c} \underline{\mu}_1(t) \\ \underline{\mu}_2(t) \\ \underline{\mu}_3(t) \end{array} \right)$$

Now we know that:

$$\frac{d^2}{dt^2} \underline{\mu}(t) = \frac{d}{dt} \underline{\mu}(t) + \sum \underline{\mu}_i(t) \left(\frac{d}{dt} \underline{e}_i \right)$$

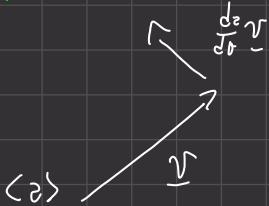
NOTE

Consider a vector \underline{v} s.t. it's $|\underline{v}| = \underline{v} \cdot \underline{v} = 1$

Let's now compute the time derivative:

$$\frac{d}{dt} (\underline{v} \cdot \underline{v}) = \phi \Rightarrow \underline{v} \cdot \frac{d}{dt} \underline{v} = 0$$

This tells us that the time derivative of a vector won't be orthogonal to the vector. The vector can only rotate, not change its length.



We can think of associating a cross product $\underline{\omega} \left(\frac{d}{dt} \underline{v} \right)$.

So we assume that I can find a vector $\underline{\omega}$ that:

$$\underline{v} \cdot \underbrace{\frac{d}{dt} \underline{v}}_{\underline{\omega} \times \underline{v}}$$

Assume $\omega_1 \neq \omega_2$ so $\omega_1 \times v = \omega_2 \times v$

$$(\omega_1 - \omega_2) \times v = 0 \quad \forall v \text{ so } \omega \text{ is unique}$$

ω represent the angular velocity of v to respect to b .

Now we have 2 way to describe:

$$\sum b_i \mu_i(t) \left(\underbrace{\frac{d}{dt} e_i^b}_{\omega_b \times e_i^b} \right)$$

$$\uparrow$$

$\omega_b \times e_i^b \rightarrow$ the angular velocity of frame b in respect to $?$

$$\begin{aligned} \frac{d}{dt} \mu(t) &= \frac{db}{dt} \mu(t) + \sum \omega_{b/2} \times (\mu_i \cdot e_i^b) = \\ &= \frac{d}{dt} \mu(t) + \omega_{b/2} \times \underbrace{\sum_{i=1}^3 (\mu_i \cdot e_i^b)}_{\mu(t)} \end{aligned}$$

We have just derived the law that associates a derivative from 2 frame to another.

the mother of all laws

$$\boxed{\frac{d}{dt} \mu(v) = \frac{db}{dt} \mu(v) + \omega_{b/2} \times \mu(v)}$$

value of vector
is the object value.

Sythetic representation this means scalar multiplying what's inside [] by e_1^b, e_2^b, e_3^b

$$\boxed{\left[\frac{d}{dt} \mu(t) \right] = {}^b_R \left[\frac{db}{dt} \mu(t) \right] + {}^b_R \left[\omega_{b/2} \times \mu(t) \right]}$$

this is what they

$$\frac{d}{dt} \begin{bmatrix} {}^2\mu_1(t) \\ {}^2\mu_2(t) \\ {}^2\mu_3(t) \end{bmatrix} = \frac{d}{dt} \begin{bmatrix} {}^b\mu_1(t) \\ {}^b\mu_2(t) \\ {}^b\mu_3(t) \end{bmatrix} \xrightarrow{3 \times 3} \begin{bmatrix} {}^2\omega_{b/2} x \\ {}^2\mu(t) \end{bmatrix}$$

skew symmetric

$$\frac{d}{dt} \mu = \sum \left(\frac{d}{dt} {}^e\mu_i \right) e_i^2 + \dots$$

this is always a standard derivative
because these quantities are scalar

$$\frac{d}{dt} {}^b\mu = \sum \left(\frac{d}{dt} {}^b\mu_i \right) e_i^b$$

$$\frac{d}{dt} {}^2\mu = \frac{d}{dt} \left(\sum {}^b\mu_i e_i^b \right) = \sum \left(\frac{d}{dt} {}^b\mu_i \right) {}^b e_i + \sum {}^b\mu_i \left(\frac{de}{dt} e_i^b \right)$$

Let's now consider an example.

$$\frac{d}{dt} {}^2 e_i^b = \frac{d}{dt} {}^b e_i^b + \left(\omega_{b/2} \times e_i^b \right) \quad \forall i = 1, 2, 3$$

\downarrow
this is ϕ

$${}^e \left(\frac{d}{dt} {}^b e_i^b \right) = {}^b \dot{R} = \begin{bmatrix} {}^2 \omega_{b/2} x \end{bmatrix} {}^b R$$

some trickery as before

Properties:

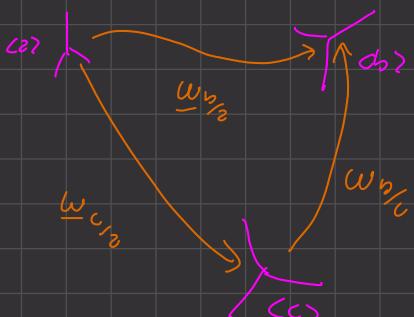
identity

$$) \quad {}^2 R {}^b R = \mathbb{I} \rightarrow \frac{d}{dt} \left({}^2 R {}^b R \right) = [\phi] \Rightarrow {}^2 R {}^b \dot{R} + {}^b R {}^2 \dot{R} = [\phi].$$

$${}^2 \dot{R} {}^b R = - {}^b \dot{R} {}^2 R = - {}^b R \left[{}^2 \omega_{b/2} \right] {}^2 R$$

$${}^2 \dot{R} = - {}^b R \left[{}^e \omega_{b/2} \right]$$

2) Composition of angular velocities



$$\begin{cases} \frac{d\omega}{dt} \mu = \frac{d\omega_b}{dt} \mu + \omega_{b/2} \times \mu \\ - \frac{d\omega_c}{dt} \mu = \frac{d\omega_b}{dt} \mu + \omega_{b/2} \times \mu \\ + \frac{d\omega}{dt} \mu = \frac{d\omega_c}{dt} \mu + \omega_{c/2} \times \mu \end{cases}$$

$$\cancel{\frac{d\omega}{dt} \mu} + \cancel{\frac{d\omega_c}{dt} \mu} - \cancel{\frac{d\omega}{dt} \mu} = \cancel{\frac{d\omega_b}{dt} \mu} + \frac{d\omega_b}{dt} \mu + \left[(\omega_{c/2} + \omega_{b/2}) - \omega_{b/2} \right] \times \mu$$

$$[(\omega_{c/2} + \omega_{b/2}) - \omega_{b/2}] \times \mu = \emptyset$$

The assumed μ arbitrary so this implies that the square brackets is a null vector:

||

$$\omega_{b/2} = \omega_{c/2} + \omega_{b/2}$$

Angular velocities are additive quantities.

Very easy equality:

$$\omega_{2/2} = \omega_{b/2} + \omega_{2/2} \Rightarrow \omega_{2/2} = -\omega_{b/2}$$

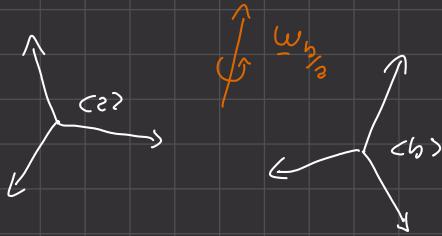
Knowing this we can immediately write the previous equation

$$\overset{b}{\omega} \overset{2}{R} \overset{2}{\dot{R}} = -\overset{b}{\omega} \overset{2}{R} \overset{2}{\dot{R}} = -\overset{b}{\omega} \overset{2}{R} [\overset{2}{\omega}_{b/2}] \overset{2}{\dot{R}} = \overset{b}{\omega} \overset{2}{R} [\overset{2}{\omega}_{2/2}]$$

Note

$$\overset{2}{A} = \overset{b}{\omega} \overset{2}{R} \overset{2}{A} \overset{b}{\omega}$$

30/20



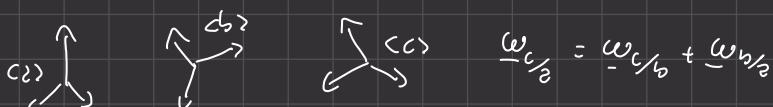
last lesson we proved that if we have a generic time varying vector then we can compute the derivative in both frames and we have relations between different frames

$$\frac{d^2}{dt^2} \mu = \frac{db}{dt} \mu(b) + \underbrace{\omega_{b/s} \times \mu(b)}_{\text{this takes into consideration the relative movement}}$$

$$\frac{d^2}{dt^2} \mu = {}^b R \frac{d^2}{dt^2} {}^b \mu + \left[{}^b \omega_{b/s} \times \right] \underbrace{{}^b \mu(t)}_{{}^b R \mu^b(t)}$$

the stepdown equation integrating?

$$\left. \begin{aligned} {}^b R &= \left[{}^2 \omega_{b/s} \times \right] {}^b R \\ {}^b \dot{R} &= \left[{}^b \omega_{b/s} \times \right] {}^b R \end{aligned} \right\} \text{we have exploited that } \omega_{c/b} = -\omega_{b/c}$$



ENVIRONMENT

RECAP



trajectory

$$\underline{\underline{r}}_{P/A} \triangleq (P-A)$$

$$\underline{\underline{r}}_{P/B} \triangleq (P-B)$$

$$\underline{\underline{r}}_{B/A} \triangleq (B-A)$$

frame b is both rotating and translating with respect to frame c2s.

$$\underline{\underline{r}}_{P/A} = \underline{\underline{r}}_{P/B} + \underline{\underline{r}}_{B/A}$$

position of p with respect to A

we also have:

$${}^2 R = \left[e_i^a \cdot e_j^b \right]$$

We want to compute velocity of P for observer in frame 2

$$\frac{d}{dt} {}_2 r_{P/A} = \frac{d}{dt} \left[{}_2 r_{P/B} + {}_2 r_{B/A} \right] = \frac{d}{dt} {}_2 r_{P/B} + \frac{d}{dt} {}_2 r_{B/A}$$

$\frac{d}{dt} {}_2 r_{P/B} + \omega_{b/2} \times {}_2 r_{P/B}$

$v_{P/b}$

This can be called
 $v_{B/2}$
velocity of point B computed
in frame 2.

NOTE: a is lowercase

Now we can say that:

$$\frac{d}{dt} {}_2 r_{P/A} = v_{P/b}$$

So now,

this are due to the effect of the motion of b

$$v_{P/b} = v_{P/b} + \omega_{b/2} \times {}_2 r_{P/B} + v_{B/b}$$

Galileo theorem for velocity distribution

$$v_{P/b} = v_{P/b} + \omega_{b/2} \times {}_2 r_{P/B} + v_{B/b}$$

$$\begin{aligned} \frac{d}{dt} v_{P/b} &= \frac{d}{dt} \left[v_{P/b} + \omega_{b/2} \times {}_2 r_{P/B} + v_{B/b} \right] = \\ &= \frac{d}{dt} v_{P/b} + \frac{d}{dt} \omega_{b/2} \times {}_2 r_{P/B} + \frac{d}{dt} v_{B/b} \\ &= \underbrace{\frac{d}{dt} v_{P/b}}_{\text{acceleration } \underline{\Omega} P/b} + \underbrace{\omega_{b/2} \times v_{P/b}}_{\text{angular acceleration of } b \text{ with respect to } 2} + \underbrace{\frac{d}{dt} v_{B/b}}_{\underline{\Omega} B/2} \end{aligned}$$

$$\underline{\Omega}_{P/B} = \underline{\Omega}_{P/B} + 2 \underbrace{\omega_{B/Z}}_{\text{Coriolis acc.}} \times \underline{v}_{P/B} + \left(\frac{d\omega}{dt} \right) \underline{\omega}_{B/Z} \times \underline{r}_{P/B} + \underline{\omega}_{B/Z} \times [\underline{\omega}_{B/Z} \times \underline{r}_{P/B}] + \underline{\Omega}_{B/Z}$$

Coriolis acc.
centrifugal acc. (quadratic function)

Make some notes on the previous formula:

Note:

$$\frac{d\omega}{dt} \underline{\omega}_{B/Z} = \frac{d\omega}{dt} \underline{\omega}_{B/Z} + \underline{\omega}_{B/Z} \times \underline{\omega}_{B/Z} = \phi$$

\uparrow
let's work assume
this is generic
vector $\omega(t)$

this allows us
to write this

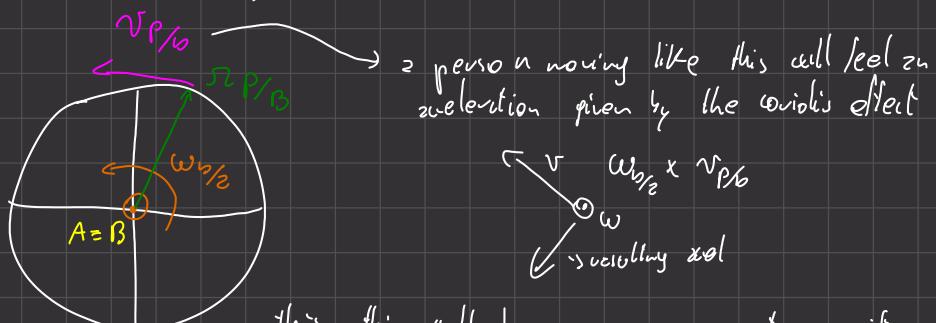
ω is in general the time derivative of NO vector and ~~spun~~ is the time derivative of a vector. Note the rate of change of the angular velocity is the same for both observers.

$$\underline{\Omega}_{P/Z} = \underline{\Omega}_{P/B} + 2 \underbrace{\underline{\omega}_{B/Z} \times \underline{v}_{P/B}}_{\text{Coriolis acc.}} + \underbrace{\underline{\omega}_{B/Z} \times (\underline{\omega}_{B/Z} \times \underline{r}_{P/B})}_{\text{Cent. acc.}} + \underline{\Omega}_{B/Z}$$

NOTE:

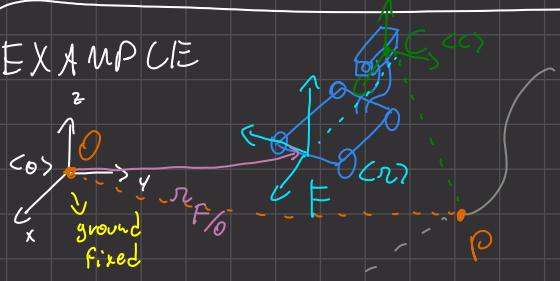
Sometime in science fiction movies you see elements in artificial gravity in centrifuge to generate centrifugal gravity.

This is done by using this quantity.



This thing will become even more strange if you start jumping.

EXAMPLE



Now this robot is moving along a certain trajectory. Now on top of this robot there is a camera.

The movement of the robot will be on the x - y plane.

Using the measurements of P we would like to give a quantitative measurement of how P is moving with respect to the ground station.

We assume that the camera will not move with respect to the platform.

We want to predict the motion of P in space given the camera movement.

We want to compute: $\underline{r}_{P/\theta}$, $\underline{v}_{P/\theta}$, $\underline{\alpha}_{P/\theta}$

- assume these quantities are known at $t = t_m$ \rightarrow time of the measurement.
- assume these is "smooth" time function
 $\underbrace{\text{differentiable and to third derivative continuity}}$

GOAL: "predict" $\underline{r}_{P/\theta}$ for $t > t_m$

Given that these functions are smooth we can use second order Taylor expansion:

$$\underline{r}_{P/\theta}(t) \approx \underline{r}_{P/\theta}(t_m) + \underline{v}_{P/\theta}(t_m)(t - t_m) + \frac{1}{2} \underline{\alpha}_{P/\theta}(t_m)(t - t_m)^2$$

In practice we are going to implement:

$$\underline{r}_{P/\theta} \approx \underline{r}_{C/F} + \underline{v}_{C/F}(t - t_m) + \frac{1}{2} \underline{\alpha}_{C/F}(t - t_m)^2$$

Note - that C and F are positioned arbitrary on the robot

- we assume that the camera is calibrated with respect to frame C/F (C_1, C_2, C_3) are calibrated w.r.t. (F, C_{22})

- assume $\underline{r}_{C/F}$ Known, $\underline{v}_{C/F}$ Known

$$\textcircled{1} \quad \underline{r}_{P/\theta} = \underline{r}_{P/C} + \underline{r}_{C/F} + \underline{r}_{F/\theta}$$

$$\underline{r}_C = \underline{r}_n + \underline{r}_c$$

\uparrow \nwarrow Known from calibration
assumed Known

\textcircled{2} Computing:

$$\underline{v}_{P/\theta} = \frac{d}{dt} \underline{r}_{P/\theta} = \frac{d}{dt} \left[\underline{r}_{P/C} + \underline{r}_{C/F} + \underline{r}_{F/\theta} \right] =$$

$$= \left[\frac{d}{dt} \underline{r}_{P/C} + \omega_{C/F} \times \underline{r}_{C/F} \right] + \left[\frac{d}{dt} \underline{r}_{C/F} + \omega_{n/F} \times \underline{r}_{n/F} \right] + \underline{v}_{F/\theta}$$

velocity of P is estimated by the camera

$$\omega_{C/F} + \omega_{n/F} \rightarrow \text{this is } 0 \text{ if the camera is not rotating.}$$

$$\underline{\underline{v}}_{P/O} = \underline{\underline{v}}_{P/C} + \underline{\omega}_{R/O} \times (\underline{r}_{P/C} + \underline{r}_{C/P}) + \cancel{\frac{d}{dt} \underline{r}_{C/F}} \text{ for us because the center is fixed}$$

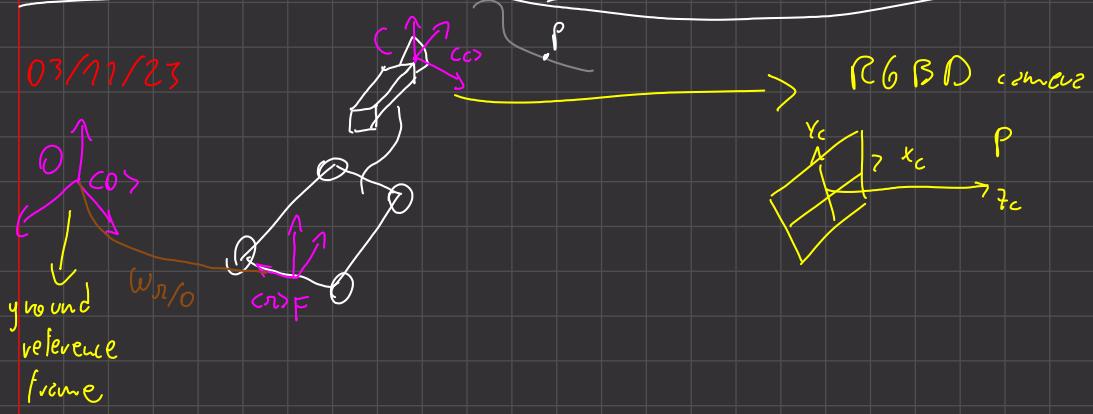
$${}^0 \underline{\underline{v}}_{P/O} = {}^0 R {}^c \underline{\underline{v}}_{P/C} + [{}^0 \omega_{R/O} \times] [{}^0 R {}^c \underline{r}_{P/C} + {}^0 R {}^R \underline{r}_{C/F}] + {}^0 \underline{\underline{v}}_{F/O}$$

try doing the anal. by yourself.

Final notes:

$$\frac{d}{dt} \underline{r}_{F/O} = - \frac{d}{dt} \underline{r}_{O/F} = \frac{d}{dt} \underline{r}_{O/B} + \underline{\omega}_{R/O} \times \underline{r}_{O/F}$$

03/11/23

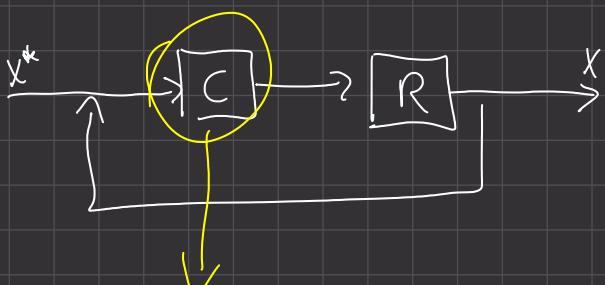


From $r_{O/C}$ we have a consistent rotation

$$\begin{cases} r_{O/C} \\ r_{R/C} \\ {}^c R \end{cases}$$

these are constants
→ calibration parameters

MP1E



for(;;){

<read sensors>

<compute errors>

<compute control action>

<send control to motors>

<suspends until the next sampling period starts>

idea of
digital control

Use current to compute $\begin{cases} \text{R}_P/0(t_m) \\ \text{V}_P/0(t_m) \\ \text{A}_P/0(t_m) \end{cases}$

$$p_0(t) = p_0(t_m) + v_{p_0}(t_m)(t-t_m) + \frac{1}{2} \ddot{v}_{p_0}(t_m)(t-t_m)^2$$

$t \geq t_m$ Taylor expansion?

$$R_p = R_{p/c} + R_{c/F} + R_{F/p}$$

$$\frac{d}{dt} \underline{\omega}_C = \underline{\omega}_{P/C} + \underline{\omega}_{C/F} \times \underline{r}_{P/C} + \frac{d}{dt} \underline{r}_{C/F} + \underline{\omega}_{n/o} \times \underline{r}_{C/F} + \underline{\omega}_{F/o}$$

↓
 $\underline{\omega}_{P/C}$
 ↓
 could be estimated

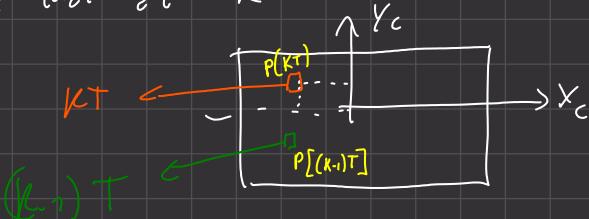
↓
 $\underline{\omega}_{C/F}$
 ↓
 " "
 " "
 given

↓
 $\underline{r}_{C/F}$
 ↓
 " "
 " "
 given

↓
 $\underline{\omega}_{n/o}$
 ↓
 " "
 " "
 given

↓
 $\underline{\omega}_{F/o}$
 ↓
 " "
 " "
 given

Assume that $\alpha t \ll kT$



$$c_{\text{approx}} \simeq \frac{1}{T} \left\{ {}^c P(K_T) - {}^c P[(K-1)T] \right\}$$

approximation of the velocity

↳ we see that V_{FC} is given

$$\underline{V}_{P/O} = \underline{V}_{P/C} + \underline{\omega}_{N/O} \times \left[\underline{I}_{C/F} + \underline{I}_{P/C} \right] + \underline{V}_{F/O}$$

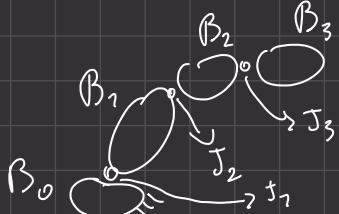
 this is often and measured with medical sensors.
usually \approx $2\text{ kg}/\text{m}^2$

$$\frac{d}{dt} \mathbf{v}_{P_0} = \frac{d}{dt} \mathbf{v}_{P_C} + \omega_{r_{P_C}} \times \mathbf{v}_{P_C} + \omega_{r_{P_F}} \times r_{P_F} + \omega_{r_{P_F}} \times \left[\frac{dr}{dt} - \mathbf{v}_{P_F} + \omega_{r_{P_F}} \times r_{P_F} \right] \xrightarrow{\text{N.B.}} + \ddot{r}_{P_F}$$

$\underbrace{\omega_{r_{P_C}}}_{\text{c } \dot{r}_{P_C}}$ $\underbrace{\omega_{r_{P_F}}}_{\text{c } \dot{r}_{P_F}}$
 $\underbrace{\mathbf{v}_{P_C}}_{\text{c } \mathbf{v}_{P_C}}$ $\underbrace{\mathbf{v}_{P_F}}_{\text{c } \mathbf{v}_{P_F}}$
 $\underbrace{\omega_{r_{P_F}} \times r_{P_F}}_{\text{c } \omega_{r_{P_F}} \times r_{P_F}}$

Kinematic Chains (Holonomic Robots)

A Kinematic chain is a connection of rigid bodies.



$B_i \rightarrow$ body
 $J_i \rightarrow$ joint

Connected by mechanism that allow movement according to some laws.

In a holonomic mechanism the relative positions are specified by numbers.

We will assume that there will be the ground body.

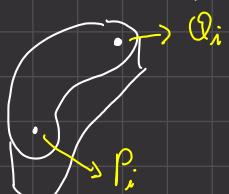
We are working in the forward direction $\begin{cases} \text{forward geometry} \\ \text{forward kinematics} \end{cases}$

Assumptions:

B_i is rigid body $\left(\text{a rigid body is an entity where the relative distance of two points taken on the object do not change} \right)$

Each body in the Kinematic chain is a link

In the rigid body we have some relevant points:



the ancestor is connected to P_i , the successor to Q_i .

Types of joints

- rotational joints (about axis) so 1-D rotational joints.

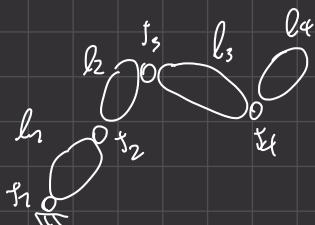
We can associate a variable q_i to the joint to tell the rotation
 \downarrow joint angle

1 joint describe the number of degrees of freedom of the successor with respect to the ancestor.

- prismatic joint (1D). q_i , joint variable that now describes the joint displacement.

06/11/23

Kinematic chains



Each joint induce the movement of each body to which they are connected.

We have:

- revolitional joint (1D)

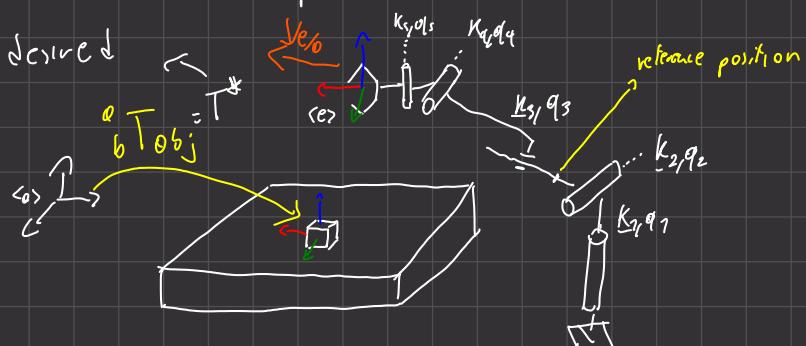
- prismatic joint (1D)

For each joint we have: $(\theta_j, \underline{k}_j)$

θ_j angle or linear displacement
 \underline{k}_j joint variable

We want to investigate:
- forward geometry (specifies how those bodies are in space. It has only one solution)
- forward kinematic problem

When there are the inverse problems:



The solution of the forward kinematic problem allows us to say how the what will be positioned.

e T is what is calculated by the forward kinematic problem.

$${}^0_e T = \left[\begin{array}{c|c} {}^0 R_{e/0} & {}^0 v_{e/0} \\ \hline 0 & 1 \end{array} \right] = {}^0_e T(q_1, q_2, q_3, q_4, q_5) \rightarrow \text{J.o.f. of the kinematic chain.}$$

It can be seen as a matrix function
of the joint positions.

We define $\underline{q} \triangleq \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_N \end{bmatrix}$

joint (variable) vector

$${}^0_e T = {}^0_e T(\underline{q})$$

There is not a direct formula
but computational steps.

There is then the inverse problem. That is: How should I position the joints to reach my destination.

Now we want:

$$T^* = {}^0_e T(\underline{q})$$

We want to compute \underline{q} such that T^* is valid. (Inverse geometric problem)

This problem usually is not well defined because it could exist 1, infinite or no solution.

Note

Yaw-pitch-roll are the simplest inverse kinematic problem.

$$(\theta, \varphi, \psi) \Rightarrow R = R_z(\theta) R_x(\varphi) R_z(\psi)$$



The problem now is: what happens to the velocity of the end effector given $\dot{\underline{q}}$

$$\begin{bmatrix} {}^0 \omega_{e/0} \\ {}^0 v_{e/0} \end{bmatrix} = f(\underline{q}, \dot{\underline{q}}) = J_e \dot{\underline{q}}$$

$f_e = J_e(\underline{q})$
jacobian matrix

By solving the inverse velocity problem we will also solve the inverse geometry problem.

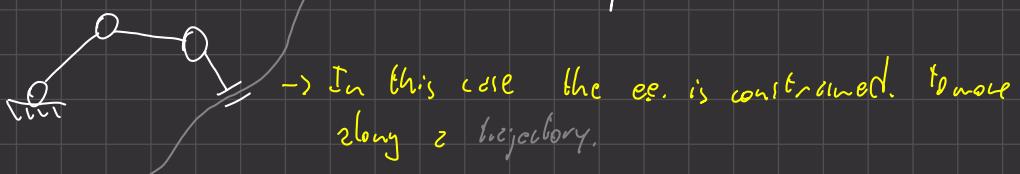
When you talk about q you will think geometry, $q/q \rightarrow$ velocity,
 $\dot{q}, \ddot{q} \rightarrow$ dynamics.

7 types of Kinematic chains.

In this course we will talk about a serial, open, kinematic chain.
 one after the other
 you can freely move every
 of the body specifying one
 of the variables.



Serial closed loop K.C.



In this case the robot has m d.o.f. < n number of joints because it has to follow a particular trajectory.

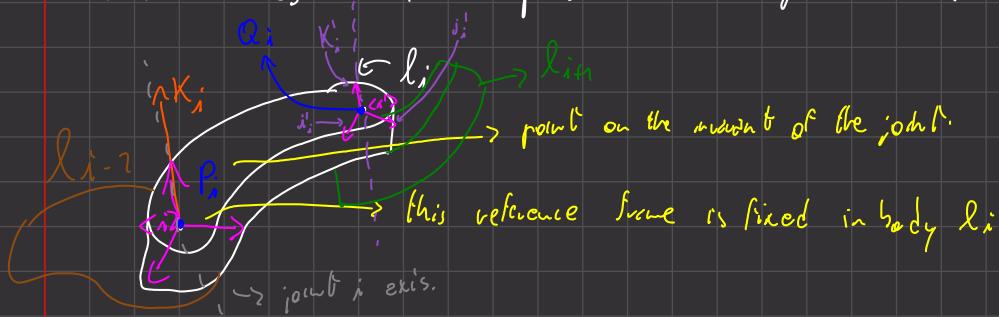
Branched or tree structured Kinematic chain



You can think of this like two kinetic chains that share one chain link.

One example of this kind of robot is a robot hand.

Each link has a father except from the ground base.



Each link l_i must have a relative motion with respect to his ancestor.

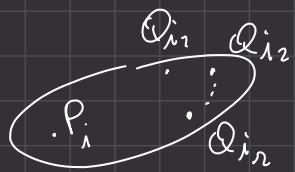
$\frac{R_{Q_i}}{P_i}$ is constant. This are three numbers. Position of the z-axis of the next body with respect to frame i .

R is constant

We need the z-axis of $C_i \rightarrow C_{i+1}$ to be on the direction of the instant of the joints.

$$T_i = \begin{bmatrix} R \\ P_i \\ 0 \end{bmatrix} \rightarrow \text{this is expressed in frame } C_i$$

Note



For each one of this points you will have a transformation matrix like the one above.

We need to choose the orientation of i^1 and j^1

We have that $P_{i+1} \equiv Q_i$ if the joint $i+1$ is rotational

We have that $P_{i+1} \equiv Q_i$ if the joint $i+1$ is prismatic and $q_{in} = 0$

K_{i+1} represents the z-axis of the frame $i+1$.

K_{i+1} must be aligned with the axis of J^1 so: $K_{i+1} \equiv K_i^1$

This means $C_{i+1} \equiv C_i^1$ if the joint $i+1$ is prismatic. If the joint is rotational the condition is the same if $q_{in} = 0$

Rotation in this way are always studied around the z-axis

Note that i and j on C_i^1 should be placed in the same position of C_{i+1} when $q_{in} = 0$

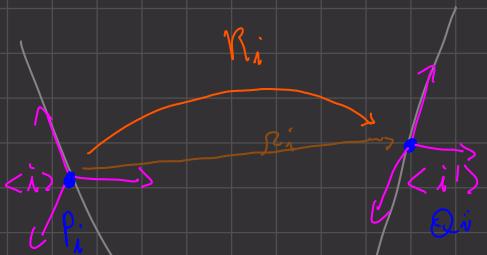
Let's define 2 symbolic vectors

$$\Sigma = \underbrace{\left[\begin{array}{c} * \\ \vdots \\ * \end{array} \right]}_{N \text{ elements}}$$

$$\begin{cases} * & R - \text{rotational} \\ * & P - \text{prismatic} \end{cases}$$

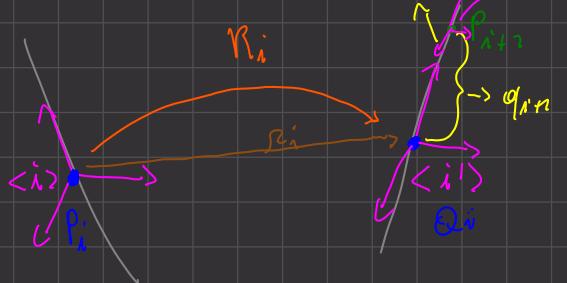
$$\Sigma_i = \begin{cases} R \\ P \end{cases} \quad \Gamma = [R, P, R_1, P_1, R_1] \quad \text{---}$$

Repeating the previous example only with the useful stuff we get:



probabilistic cycle

$$\text{if } \Gamma_{in} = P$$



$$\boxed{\begin{aligned} {}^i \pi_{i+1} &= {}^i p_{i+1} / p_i = \pi_i + K_{i+1} q_{i+1} \\ {}^i \pi_{i+2} &= {}^i \pi_i + {}^i K_{i+1} q_{i+1} \\ &\quad {}^i K_{i+1} \rightarrow \text{constant} \end{aligned}}$$

NOTE

$${}^{in} K_{in} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad \left. \begin{array}{l} \\ \\ \end{array} \right\}$$

this is where the choice of common τ is effective

$${}^{i'} K_{i'} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\text{Now } {}^i K_i = R_i {}^i K_i \text{ is a constant}$$

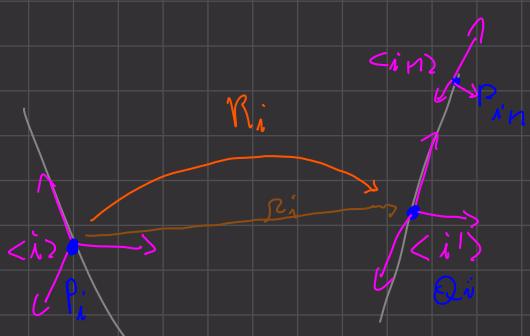
This is just the third column of the matrix.

$${}^{in} R = R_i$$

relative

$$\Gamma_{i+1} = R$$

Note R_{ii} and Q_{ii} are constant, but have reduced separated to not make it mess.



$${}_{i+1} \tau = \underline{\tau}_i$$

$${}_{i+1} R = R_i R_2(q_{i+1})$$

$$R_2(\alpha) = \begin{bmatrix} c\alpha & -s\alpha & 0 \\ s\alpha & c\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

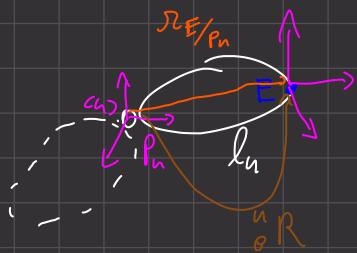
Forward geometry

$$\underline{\tau}_{i+1} = \begin{cases} R_{i+1} & \text{if } \Gamma_i = R \\ \underline{\tau}_{i+1} + K_i q_i & \text{if } \Gamma_i = P \end{cases}$$

$${}_{i+1} R = \begin{cases} R_{i+1} R_2(q_i) & \text{if } \Gamma_i = R \\ R_{i+1} & \text{if } \Gamma_i = P \end{cases}$$

Digital link: farthest body in the chain

Usually to the last body l_n is an "auxiliary" body called e.g. is attached. E is fixed to the last body



r_E/p_n is the last joint we are interested in

Complete robot forward geometry

$P_0, Q_0 \dots P_{n-1}, Q_{n-1}; P_N, (E)$ ↑ one or more.

Point of ground reference frame where my robot is attached



What is the position of c_n with respect to c_0

$$r_{w0} = r_{z0} + r_{y1} + r_{yz2} + \dots + r_{wn-1}$$

$${}^0 R_n = {}^0 R {}^1 R {}^2 R \dots {}^{n-1} R$$

example → in general is function of q_i

$${}^0 T = \begin{bmatrix} {}^0 R & ; & {}^0 r_0 \\ \vdots & \vdots & \vdots \\ 0 & ; & 1 \end{bmatrix} \rightarrow \text{this depends on the type of joint.}$$

either one or the other changes with q_i .

Generalization

$${}^{i-1} T_i = \begin{bmatrix} {}^{i-1} R(q_i) & ; & {}^{i-1} r_{i-1} \\ \vdots & \vdots & \vdots \\ 0 & ; & 1 \end{bmatrix} \Rightarrow {}^0 T_n = \prod_{i=1}^n {}^{i-1} T(q_i)$$

10/12/23 R_i, K_i

$${}^{i-1} r_{i-1} = \begin{cases} -{}^{i-1} r_{i-1} & \Gamma_i = R \\ -{}^{i-1} r_{i-1} + K_i q_i & \Gamma_i = P \end{cases}$$

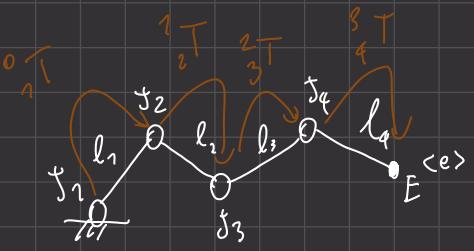
\hookrightarrow constant in $(i-1)$

$${}^{i-1} R = \begin{cases} R_{i-1} R_z(q_i) & \Gamma_i = R \\ R_{i-1} & \Gamma_i = P \end{cases}$$

$R \in AP$

R_\emptyset

$\emptyset R$



$${}^0 T = \begin{bmatrix} {}^0 R & {}^0 t \\ 0_{3 \times 3} & 1 \end{bmatrix} = {}^0 T(q_0)$$

↙ this is the only varying value.
the rest is constant and given by the geometry.

$$R_2(q_1) = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

↙ this keeps in the resulting of the product the last column.

Note that:

$${}^1 K_1 = {}^1 K_{1,1} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \text{ by construction.}$$

When we need to transform this in frame $i-1$

$${}^0 T = {}^0 T {}^1 T \dots {}^{i-1} T {}^i T = {}^0 T(q_i)$$

We can think to characterize the last link with:

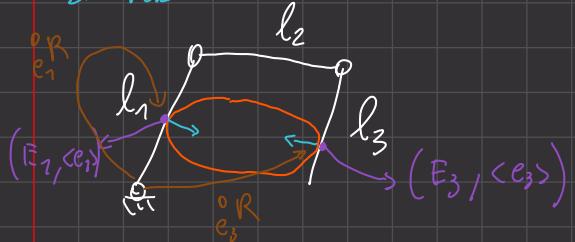
$$\left\{ \begin{array}{l} {}^n \Sigma e/n \\ {}^n R \end{array} \right. \Rightarrow {}^n e T \left[\begin{array}{c|c} {}^n R & {}^n \Sigma e/n \\ \hline {}^0 T & 1 \end{array} \right] = {}^0 e T = {}^0 T \cdot {}^n e T$$

the type of end effector could change and therefore this quantity can change while keeping the structure of the robot the same.

remove
(

) The ee. could in principle be attached to any link. This changes nothing. The only difference is that now the sequence of products should stop at the link on which the ee. is mounted.

EXAMPLE



I assume that some quantities are known by sensors:

$$\left({}^0 \tau_{E_1/1}, {}^0 R_{e_1} \right), \quad \left({}^3 \tau_{E_3/3}, {}^3 R_{e_3} \right)$$

I want to compute $\phi_{\bar{E}_3/E_1} = \phi_{\bar{E}_3/E_3} - \phi_{\bar{E}_3/E_1}$ → Homogeneous so we do

$$= {}^0 T {}^3 \tau_{E_3/3} - {}^0 T {}^1 \tau_{E_1/1} \rightarrow \text{this is for the transformation}$$

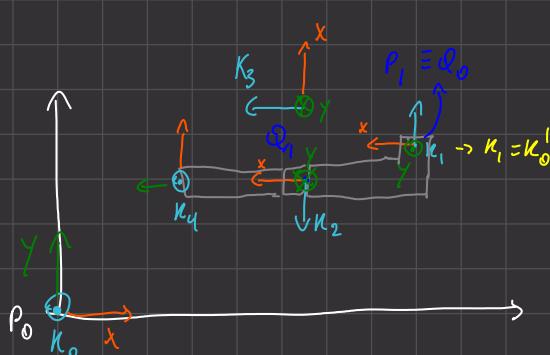
this is the position
of E_3 with respect
to the origin

Now to compute the rotations:

$$\begin{aligned} {}^0 R_{e_3} &= {}^0 R_{e_1} {}^1 R_{e_3} \Rightarrow {}^0 R_{e_3} = {}^0 R_{e_1} \tau^{\phi} R_{e_3} \\ &\quad \downarrow \\ {}^0 R_{e_1} \tau^{\phi} R_{e_1} &= ({}^0 R_{e_1} \cdot {}^1 R_{e_1})^T \end{aligned}$$

EXAMPLE

(P_1, Ω_1) can be set arbitrarily on the body.



$$\begin{aligned} {}^0 \tau_{\phi/0} &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \\ {}^0 R_{\phi/0} &= R_0 = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \end{aligned}$$

ROTATIONAL joint

$$\underline{\underline{\tau}}_0 = \underline{\underline{\phi}}_0 \underline{\underline{\phi}}_0^T$$

This two vectors describe the rotation of x and y wrt $\underline{\underline{\phi}}$

$${}_1 R = {}_0 R R_z(q_1) = \begin{pmatrix} * & * & 0 \\ * & * & 1 \\ * & * & 0 \end{pmatrix}$$

Let's now define the model for link 2

$$(P_1, Q_1)$$

$${}^1 \underline{\underline{\tau}}_{2/1} = \begin{pmatrix} 2 \\ 0 \\ -1 \end{pmatrix}$$

$${}^1 R = R_1 \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix}$$

$$\left\{ \begin{array}{l} {}^1 R_{2/1} = {}^1 \underline{\underline{\tau}}_{2/1} \\ {}^1 R = R_1 R_z(q_2) \end{array} \right.$$

Now for link 2 we have the voluntary force vector:

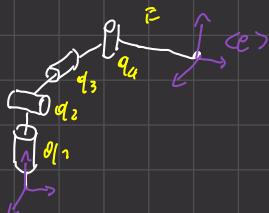
$$(P_2, Q_2) \quad {}^2 \underline{\underline{\tau}}_{2/2} = \begin{pmatrix} 0 \\ 0 \\ \phi \end{pmatrix} \quad {}^2 R = R_2 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix}$$

Now we work on the ee

$$(P_3, E) \quad {}^3 \underline{\underline{\tau}}_{E/3} = \begin{pmatrix} 0 \\ 0 \\ 2 \end{pmatrix}$$

$$\text{Done } {}^3 R_e$$

20/11/23



$${}^0 T = {}^1 T(q)$$

$${}^0 T = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_N \end{bmatrix}$$

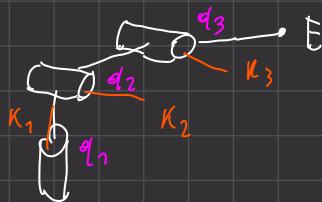
$${}^0eT = {}^0_1T(q_1) {}^1_2T(q_2) {}^2_3T(q_3) \dots {}^{n-1}_nT(q_n) {}^n_eT$$

I general this formula is not known explicitly.

If we fix q we know the position of e . This is the forward kinematic problem.

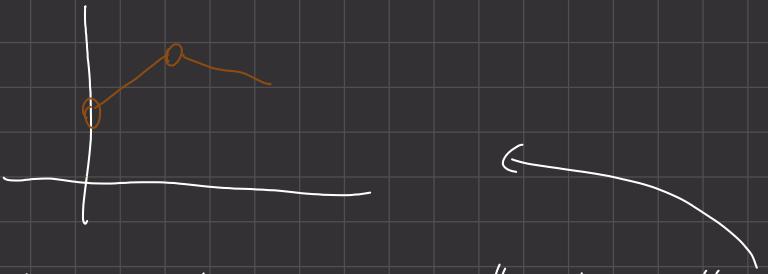
The inverse is when we know the position of the EE. and we get the positions of all the joints.

The inverse problem has solution only in some cases. Let's make an example:



K_2 and K_3 are always parallel.

For any value of q_1 you can identify a plane like this:



If you slice the space with planes this mechanism can be solved trivially.

Now we want to find the solution of the end effect

Let's now show a spherical wrist



A redundant Robot is a robot where the number of joints is larger than the number of dof expected of the EE.

In general to solve a robot and we have that has ≥ 6 dofs then the robot is redundant and there is not an unique solution to the inverse kinematics.

$$q = \begin{bmatrix} q_1 \\ \vdots \\ q_N \end{bmatrix} \in \mathbb{R}^N \quad N \geq 6$$

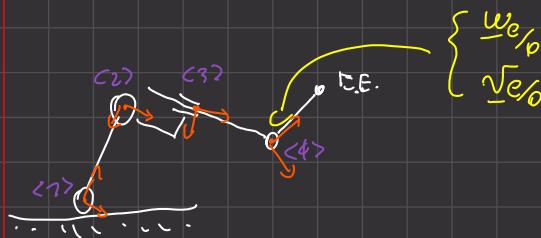
Robot Kinematics

If I change the position of the joint variables how does the robot moves.

More specifically if I change all the joint variables velocities what is the velocity that the ee will have.

The inputs from the kinematics will be position and velocity and the output the velocity (linear or angular) position of the joints.

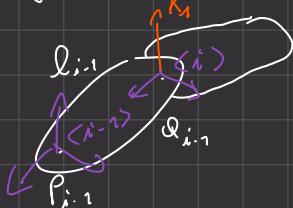
We have a robot mechanism:



If I apply any possible configuration of joint velocity what is the velocity of the ee. We want to know $\begin{cases} \dot{w}_{ee/0} \\ \dot{v}_{ee/0} \end{cases}$

To understand this we are going to reduce everything to one single joint.

In the modeling of joints we defined:



We want to compute the velocities of points fixed in body i and the reference frame in body $i-1$. The first step will be: what is the velocity of point P_i w.r.t P_{i-1} .

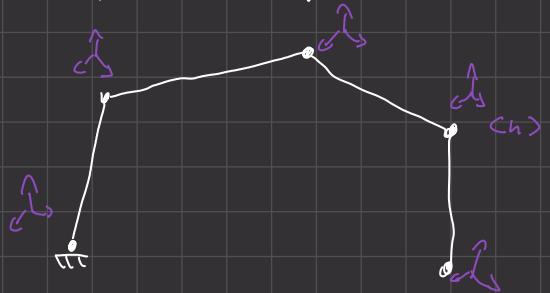
$$\dot{v}_{P_i/P_{i-1}} \triangleq \frac{d\dot{v}_{i-1}}{dt} (P_i - P_{i-1}) = \begin{cases} \emptyset & \Gamma_i = R \\ K_i \dot{q}_i & \Gamma_i = P \end{cases}$$

scalar
the derivative of scalar is the rate for every possible orientation

$$\dot{w}_{P_i/P_{i-1}} \triangleq \begin{cases} K_i \dot{q}_i & \Gamma_i = R \\ \emptyset & \Gamma_i = P \end{cases}$$



The outputs of our dynamics are



$$\begin{cases} \dot{\varphi} w_{n/\phi} = \dot{\varphi} J_h^A(q) \\ \dot{\varphi} v_{n/\phi} = \end{cases}$$

Quantities
that I
want to
compute

$$\left\{ \begin{array}{l} \underline{w}_{n/\phi} = \underline{w}_{n-1} + \underline{w}_{n-2} + \dots + \underline{w}_1 + \underline{w}_0 \\ \underline{v}_{n/\phi} : \end{array} \right.$$

N.B

$w_{n/\phi}$ can be expressed as the sum of all the velocities of the r.f.
that we encounter by going towards the ee.

$$w_{n/\phi} = \underbrace{\underline{w}_0}_{\substack{\uparrow J_0^A \cdot \\ q_1}} + \underbrace{\underline{w}_1}_{\substack{\uparrow J_1^A \cdot \\ q_2}} + \dots + \underbrace{\underline{w}_{n-1}}_{\substack{\uparrow J_{n-1}^A \cdot \\ q_n}} = \sum_{i=0}^{N-1} J_i^A \cdot \dot{q}_i$$

(A stands for angular)

This quantity could be K_2 or ω

It's very important that they are linear in (for ex. in this case) q_2 .

Linear combination of vectors where the coefficients are the vectors.

This is a linear matrix and so can be expressed as a product by a vector.

$$J_i^A = \begin{cases} K_i & \Gamma_i = R \\ \emptyset & \Gamma_i = P \end{cases}$$

We can now double to calculate:

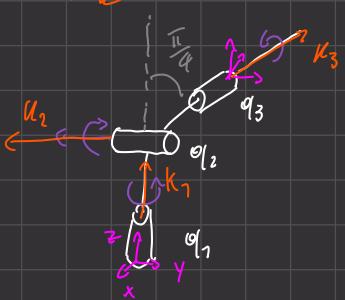
$$\dot{\varphi} w_{n/\phi} = \sum_{i=0}^{N-1} \dot{\varphi} J_i^A \cdot \dot{q}_i = \underbrace{\left[\dot{\varphi} J_0^A, \dot{\varphi} J_1^A, \dots, \dot{\varphi} J_{n-1}^A \right]}_{\substack{\text{3D vector} \\ \text{this is a mixed vector-product.}}} \underbrace{\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_n \end{bmatrix}}_{\dot{q}}$$

ANGULAR JACOBIAN MATRIX

$$\dot{\varphi} J_h^A \in \mathbb{R}^{3 \times N}$$

To move the robot in the space you need at least 3 independent values
so 3 rotational joints (Piper theorem satisfied)

EXAMPLE



$$q_1 = \phi$$

$$q_2 = \frac{\pi}{4}$$

$$q_3 = \phi$$

This is a roll-pitch-roll sequence.

$$\mathcal{J}_1^A = \begin{cases} K_1 & \Gamma_1 = R \\ \emptyset & \Gamma_1 = P \end{cases}$$

$$\underline{\omega}_{v_0} = \underline{\omega}_{\frac{1}{0}} + \underline{\omega}_{\frac{2}{1}} + \underline{\omega}_{\frac{3}{2}} = \mathcal{J}_1^A \dot{q}_1 + \mathcal{J}_2^A \dot{q}_2 + \mathcal{J}_3^A \dot{q}_3 = \\ = K_1 \dot{q}_1 + K_2 \dot{q}_2 + K_3 \dot{q}_3$$

$$\Phi \underline{\omega}_{v_0} = \begin{bmatrix} \emptyset & K_1 & \emptyset \\ \emptyset & \emptyset & K_2 \\ \emptyset & K_3 & \emptyset \end{bmatrix}$$

\hookrightarrow This is strongly affected by q_2

Let's assume that K_2 with $q_2 = 0$ is aligned with i_2 and so

$$\emptyset K_2 \equiv \emptyset i_2 \in \emptyset i_2$$

In practice we have:

$$\Phi \underline{\omega}_{v_0} = \begin{bmatrix} \emptyset & K_1 & \emptyset \\ \emptyset & \emptyset & K_2 \\ \emptyset & K_3 & \emptyset \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & * \\ 1 & 0 & * \end{bmatrix} \underbrace{\dot{q}_1}_{\text{these depends on } q_2}$$

this is possible if
 $q_2 \neq \{0, \pi\}$

$$\emptyset \mathcal{J}_3^A$$

If $q_2 = \{\emptyset, \pi\}$ K_3 becomes dependent with K_1 .

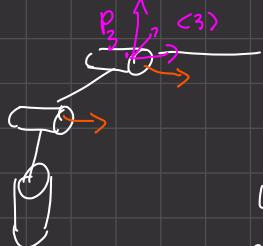
K_1 becomes parallel to K_3

this is a singularity. the rank of the matrix depends from 3 to 2.
orthogonal to the axis

This means that there are rotation ∇_0 of the joints that cannot be generated in this position.

Sometimes we need the jacobian to be full rank.

Example



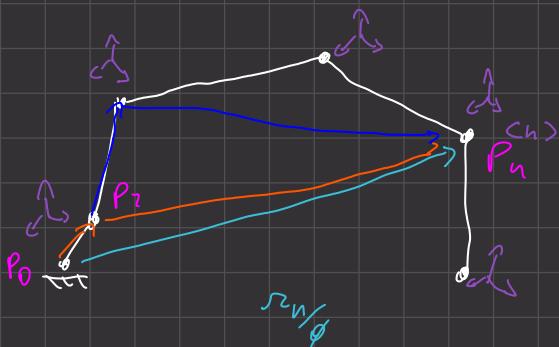
$$\phi_{\sum_{i=0}^4} = \begin{bmatrix} \phi_{k_1}, \phi_{k_2}, \phi_{k_3} \end{bmatrix}$$

Note: v is composed M.B

What are the linear values of these vectors u_2 and u_3 ?
are always parallel so $\text{rank}(\phi_{\sum_{i=0}^4}) = 2$

$$A \in \mathbb{R}^{3 \times n}$$

$$v_{n/\phi} \triangleq \frac{d\phi}{dt} (P_n - P_\phi) = \frac{d\phi}{dt} r_{n/\phi}$$



$$v_{n/\phi} = \frac{d\phi}{dt} r_{n/\phi} = \frac{d\phi}{dt} \left[r_{n/1} + r_{1/0} \right] = \quad \text{given by the joint model}$$

$$= \underbrace{\frac{d\phi}{dt} r_{n/0}}_{\text{this we know}} + \underbrace{\frac{d\phi}{dt} r_{n/1}}_{v_{1/0}} = \underbrace{\frac{d\phi}{dt} r_{n/0}}_{\text{given by the forward}} + \underbrace{\frac{d\phi}{dt} r_{n/1}}_{\text{geometry}} + \underbrace{\omega_{1/0} \times r_{n/1}}_{\text{functions of } \dot{\phi}_1} =$$

$$= (r_{n/0} + \omega_{1/0} \times r_{n/1}) + \frac{d\phi}{dt} r_{n/1} =$$

And now the same step is repeated

$$= (r_{n/0} + \omega_{1/0} \times r_{n/1}) + \frac{d\phi}{dt} [r_{n/1} + r_{n/2}] =$$

$$= (r_{n/0} + \omega_{1/0} \times r_{n/1}) + r_{n/1} + \frac{d\phi}{dt} r_{n/2} + \omega_{2/1} \times r_{n/2} =$$

$$= (\underline{v}_{z_0} + \underline{\omega}_{z_0} \times \underline{r}_{n_{z_0}}) + (\underline{v}_{z_1} + \underline{\omega}_{z_1} \times \underline{r}_{n_{z_1}}) + \frac{d\varphi}{dt} \underline{r}_{n_z} =$$

skipping to the last step

this is ϕ

$$= (\underline{v}_{z_0} + \underline{\omega}_{z_0} \times \underline{r}_{n_0}) + (\underline{v}_{z_1} + \underline{\omega}_{z_1} \times \underline{r}_{n_1}) + \dots + (\underline{v}_{z_{n-1}} + \underline{\omega}_{z_{n-1}} \times \underline{r}_{n_{n-1}}) =$$

$$= \sum_{i=1}^n (\underline{v}_{z_{i-1}} + \underline{\omega}_{z_{i-1}} \times \underline{r}_{n_{i-1}}) \stackrel{\Delta}{=} \underbrace{\quad}_{\text{this is a linear function of } \dot{q}_i}$$

$$\stackrel{\Delta}{=} \sum_{i=1}^n {}^0 J_{n_i}^L \dot{q}_i$$

$${}^0 J_{n_i}^L = \begin{cases} K_i & \Gamma_i = P \\ K_i \times r_{n_i} & \Gamma_i = R \end{cases}$$

$${}^0 V_{n_0} = \left[{}^0 J_{n_1}^L; {}^0 J_{n_2}^L; \dots; {}^0 J_{n_n}^L \right] \dot{q}$$

skew matrix $\rightarrow {}^0 J_{n_0}^L \in \mathbb{R}^{3 \times n}$

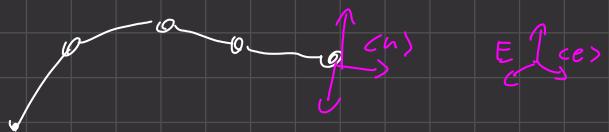
$$\begin{bmatrix} {}^0 \omega_{n_0} \\ {}^0 v_{n_0} \end{bmatrix} = \begin{bmatrix} {}^0 J_{n_0}^A \\ {}^0 J_{n_0}^L \end{bmatrix} \dot{q} \stackrel{\Delta}{=} \begin{bmatrix} {}^0 J_{n_0}^L \\ \underbrace{{}^0 J_{n_0}^A} \end{bmatrix} \dot{q}$$

$\mathbb{R}^6 \quad \mathbb{R}^{6 \times n}$

If the robot is redundant you have more than 6 d.o.f.

$$({}^0 J_{n_0}^L = {}^0 J_{n_0}(\dot{q}) \rightarrow \text{Basic Robot Kinematics})$$

If the rank of J is less than 6 then we cannot compute in an unique way the positions of the joints.



the reference frame (e) is fixed with (n)

$$\omega_{e/\phi} = \omega_{n_0}$$

$$V_{e/\phi} = \frac{d\phi}{dt} \left[r_{n_0} + r_{e/n} \right] = \frac{d\phi}{dt} r_{n_0} + \frac{d\phi}{dt} r_{e/n} =$$

$$= V_{n/\phi} + \frac{d\phi}{dt} r_{e/n} + \omega_{n/\phi} \times r_{e/n}$$

$$\omega_{e/\phi} = \begin{bmatrix} 1_3; \phi_3 \end{bmatrix} \begin{bmatrix} \omega_{n_0} \\ \phi_{V_{n_0}} \end{bmatrix} \rightarrow -r_{e/n} \times \omega_{n_0}$$

Now ϕ (- $r_{e/n} \times \omega_{n_0}$) becomes hazardous

$$[r_{e/X} \times]^T$$

$$\omega_{e/\phi} = \begin{bmatrix} r_{e/n}^T & 1_3 \end{bmatrix} \begin{bmatrix} \omega_{n_0} \\ \phi_{V_{n_0}} \end{bmatrix}$$

$$\begin{bmatrix} \omega_{e/\phi} \\ \phi_{V_{e/\phi}} \end{bmatrix} = \int_{E/N} \omega_{N/\phi} \cdot$$

Rigid point jacobian of point E in body N.

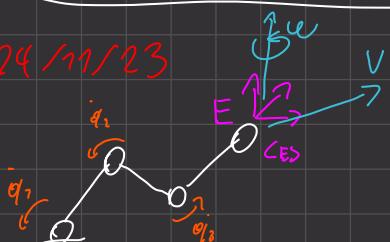
Structure of S :

$$\begin{bmatrix} 1_3 & | & \phi_{3 \times 3} \\ \cdots & | & \cdots \\ \begin{bmatrix} r_{e/N} \times \end{bmatrix}^T & | & 1_3 \end{bmatrix}$$

this is always invertible.

$$S \in \mathbb{R}^{6 \times 6}$$

24/11/23



We have found a model that allows to transform a set of joint velocities into $\dot{\omega}_{ij}$, and they represent velocity of some point of interest.

$$\dot{q} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_n \end{bmatrix}$$

So for any given configuration of the robot, and specifying the joint velocities whatever they are.

We will transform \dot{q} into a \mathbb{R}^6 vector that contains all this info. This transformation is linear. Linear is important because it means matrix.

We can say that we are able to compute:

$$\begin{pmatrix} \omega \\ \tau \end{pmatrix} = \underbrace{\mathcal{J}}_{\text{jacobian}} \dot{q}$$

\mathcal{J} is a function of the robot configuration:

$$\mathcal{J} = \mathcal{J}(q)$$

We will see later how to generate \dot{q} in order to get some desired (ω, τ) .

\mathcal{J}_{y_0} is the basic Jacobian

$$\begin{bmatrix} \mathcal{J}_{y_0}^A & 3 \times 3 \\ \mathcal{J}_{y_0}^L & 3 \times 3 \\ \emptyset & 6 \times 3 \end{bmatrix}$$

We can give to the basic Jacobian a general structure:

$$\begin{bmatrix} * & * & * & * & * & * \\ - & - & - & - & - & - \\ * & * & * & * & * & * \end{bmatrix} \rightarrow \begin{array}{l} \text{these will be } \alpha \text{ or } \phi \text{ or} \\ \text{the axis of rotation in case} \\ \text{of rotational joints} \end{array}$$

$\left\{ \begin{array}{l} \mathcal{J}_i \\ \mathcal{J}_i \times \mathcal{J}_{y_0} \\ \Gamma_i = P \\ \Gamma_i = R \end{array} \right\}$

This is by construction equal to \mathcal{J}

We also see that there would be some joints configurations where the rank of the matrix would decrease.

Singularities emerge in many different ways and they are why the rank of the matrix will decrease.

We then defined $\mathcal{S}_{E/n} \rightarrow$ the Rigid Body Jacobian

Knowing the two quantities V and ω we can transform those two quantities in the frame of the EE.

$$\mathcal{S}_{E/n} = \begin{bmatrix} 1 & ; & \phi \\ - & - & - \\ [r_{E/n}]^T & ; & 1 \end{bmatrix}$$

$$\omega_{E/0} = \dot{\alpha}_{E/n} + \omega_{n/0}$$

$$V_{E/0} = V_{n/0} + \omega_{n/0} \times r_{E/n} = V_{n/0} (-r_{E/n} \times \omega_{n/0}) =$$

$$= \begin{bmatrix} r_{E/n} \times \end{bmatrix}^T \omega_{n/0}$$

$$\begin{bmatrix} \omega_{E/0} \\ V_{E/0} \end{bmatrix} = \mathcal{S}_{E/n} \begin{bmatrix} \omega_{n/0} \\ V_{n/0} \end{bmatrix}$$

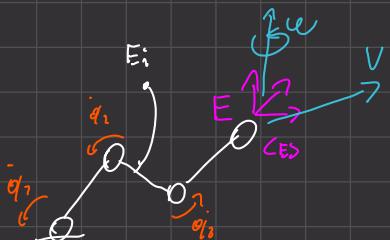
$\Downarrow 6 \times 6$ always invertible

Inversion of \mathcal{S}

$$\mathcal{S}_{E/n} = \begin{bmatrix} 1 & ; & \phi \\ - & - & - \\ A & ; & 1 \end{bmatrix}$$

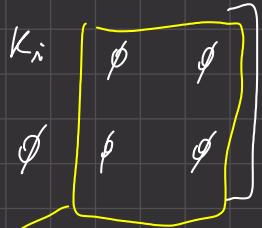
$$\mathcal{S}_{E/n}^{-1} = \begin{bmatrix} 1 & ; & \phi \\ - & - & - \\ A^T & ; & 1 \end{bmatrix}$$

I supposed that the EE is attached to the last link but this is not required.



Instead of having a function of P_n you have a function of P_i , so:

$$\phi_{\mathcal{J}_{i_1 i_2}} = \begin{cases} K_1 & K_2 \\ (K_1 \times \mathcal{R}_{i_1}) & (K_2 \times \mathcal{R}_{i_2}) \end{cases}$$



the rest of the \mathcal{J} will be ϕ .

Now we will just need to add the

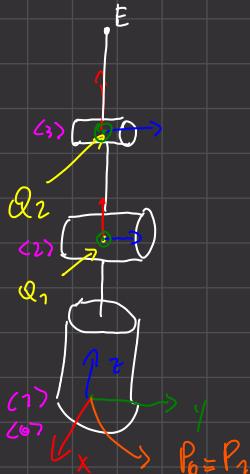
$S_{E_{i_1 i_2}}$ at the end.

Let's make an example

Example

$$\text{Link } \emptyset: \phi_{\mathcal{R}_{Q_0/\emptyset}} = \phi$$

$$\text{Link } 1: {}^1\mathcal{R}_{Q_1/\emptyset} = l_1 {}^1K_1 = {}^1\begin{bmatrix} \emptyset \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \phi \\ \emptyset \\ l_1 \end{bmatrix}$$



$$\text{Link } 2: {}^2\mathcal{R}_{Q_2/\emptyset} = l_2 {}^2i_2 = \begin{bmatrix} l_2 \\ \emptyset \\ \emptyset \end{bmatrix}$$

$$\text{Link } 3: {}^3\mathcal{R}_{E/\emptyset} = l_3 {}^3i_3 = \begin{bmatrix} l_3 \\ \emptyset \\ \emptyset \end{bmatrix}$$

This is the geometry configuration of our robot at home configuration.

$$\text{Joint } 1: \phi_{\mathcal{R}_{\mathcal{N}_\emptyset/\emptyset}} = \phi_{\mathcal{R}_{Q_0/\emptyset}} \quad {}^0R = R_\emptyset R_3(q_3)$$

$$R_\emptyset = \mathbb{I}_3$$

$$\text{Joint } 2: {}^1\mathcal{R}_{Q_1/\emptyset} = {}^1\mathcal{R}_{Q_1/\emptyset}$$

$$R_1 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

$${}^1R = R_1 R_2(q_2)$$

Joint 3 :

$${}^2 R = R_2 R_3(q_3)$$

$${}^2 \mathcal{R}_{3/2} = {}^2 \mathcal{R}_{Q_2/2}$$

$$R_2 = \mathbb{1}_3$$

Now we need to compute the geometry:

$${}^0 T = {}^0 T {}^1 T {}^2 T {}^3 T \quad \xrightarrow{\text{All home configuration}}$$

$$\phi J_{3/0} = \begin{bmatrix} \phi R_1 & \phi K_2 & \phi K_3 \\ (\mathcal{R}_1 \times R_{3/1}) & (\mathcal{R}_2 \times R_{3/2}) & \phi \end{bmatrix} \quad \leftarrow \text{All of these are geometric quantities.}$$

$\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$ this is the spin of regular colour but you can generate

$${}^0 T = \begin{bmatrix} {}^0 R & \phi \mathcal{R}_{1/0} \\ 0_{3 \times 3} & 1 \end{bmatrix} \quad \xrightarrow{\mathcal{R}_1 \text{ is the third column of this matrix}}$$

$$\mathcal{N}, \mathcal{P}$$

So you calculate ${}^0 R$ and then pick the third column.

$${}^0 T = \begin{bmatrix} {}^0 R & \phi \mathcal{R}_{2/0} \\ 0_{3 \times 3} & 1 \end{bmatrix} \quad \xrightarrow{\mathcal{R}_2 \text{ is the third column of this matrix}}$$

$${}^0 T = \begin{bmatrix} {}^0 R & \phi \mathcal{R}_{3/0} \\ 0_{3 \times 3} & 1 \end{bmatrix} \quad \xrightarrow{\mathcal{R}_3 \text{ is the third column of this matrix}}$$

$$\phi \mathcal{R}_{3/2} = \phi \mathcal{R}_{2/0} - \phi \mathcal{R}_{1/0}$$

$$\mathcal{R}_{3/2} = \mathcal{R}_{2/0} - \mathcal{R}_{1/0}$$

Let's see what happens at home configuration:

$$\begin{matrix} \text{if } \dot{\gamma}_3/\theta \\ q = 0 \end{matrix} \Rightarrow \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

The singularity is given because you have only one possible translation.

cross product of perpendicular vectors

To calculate the velocity of the ee, we have to calculate the rigid body Jacobian S :

$$\begin{matrix} \text{if } S_{e_3} \\ S_{e_3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ \left[\begin{matrix} \dot{\gamma}_{E/3} \times \end{matrix} \right]^T \end{bmatrix}$$

$${}^3\dot{\gamma}_{E/3} = l_3 {}^3\dot{\lambda}_3 \rightarrow {}^3\dot{\gamma}_{E/3} = \begin{bmatrix} l_3 \\ 0 \\ 0 \end{bmatrix}$$

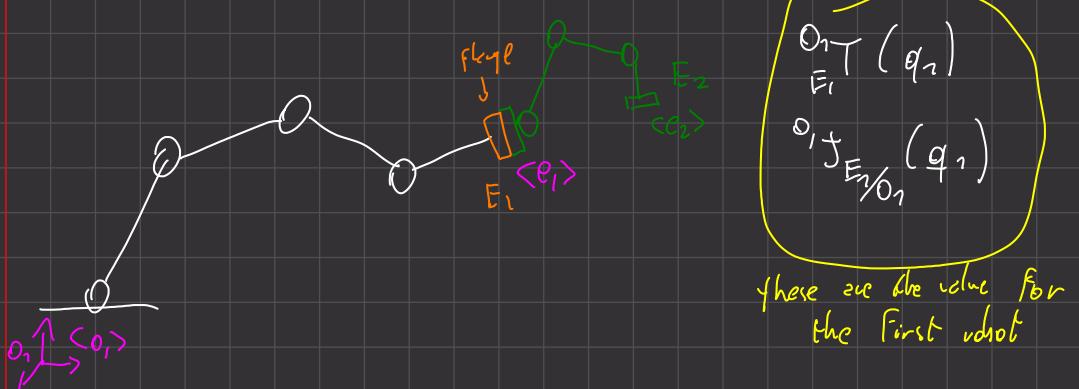
$$\begin{matrix} \dot{\gamma}_{E/3} \\ \dot{\gamma}_{E/3} = \dot{\gamma}_3 T {}^3\dot{\gamma}_{E/3} \rightarrow \text{It's not homogeneous} \end{matrix}$$

~~$\dot{\gamma}_3 T$~~

+ $\dot{\gamma}_{E/3}$ part

07/12/23

More on Jacobians



We may think to have values for the second robot

$$\begin{cases} O_2 T (q_2) \\ O_2 J_{E_2/E_1} (q_2) \end{cases}$$

I assume without loss in generality that

$$\begin{cases} E_1 \equiv O_2 \\ <e_1> \equiv <O_2> \\ e_1 T = T q \end{cases}$$

We can write that $\begin{cases} \omega_{e_2/0} = \omega_{e_1/0} + \omega_{e_2/e_1} \\ \nu_{e_2/0} = \nu_{e_1/0} + \omega_{e_1/0} \times r_{E_2/E_1} + \nu_{e_2/e_1} \end{cases}$

NB

Why not end at Transl

$${}^0\omega_{e_2/\emptyset} = {}^0J_{e_1/\emptyset}^A \dot{q}_1 + {}^0R {}^0J_{e_2/0_2}^A \dot{q}_2 = \begin{bmatrix} {}^0J_{e_1/0}^A & {}^0R {}^0J_{e_2/0_2}^A \\ \vdots & \vdots \\ e_1 & e_2 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_n \end{bmatrix}$$

$$\begin{aligned} \nu_{E_2/0} &= {}^0J_{e_1/0}^L \dot{q}_1 + \left[{}^0r_{E_2/E_1} \chi \right]^T {}^0J_{e_1/0}^A + {}^0R {}^0J_{e_2/0_2}^L \dot{q}_2 = \\ &= \left(\left[{}^0r_{E_2/e_2} \chi \right]^T \vdots \Pi_3 \right) {}^0J_{e_1/0_1}^{\phi} \dot{q}_1 + {}^0R {}^0J_{e_2/0_2}^L \dot{q}_2 \end{aligned}$$

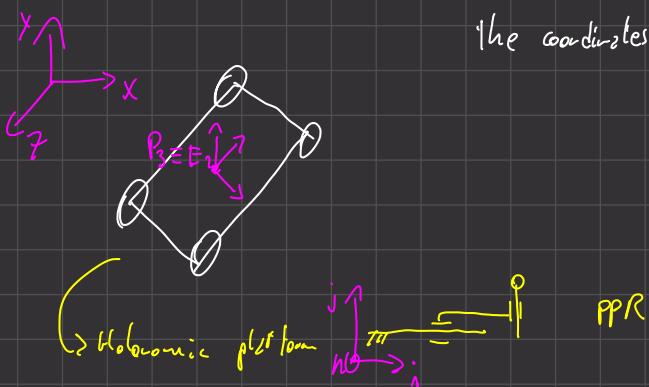
Refer
to the lesson
notes

Refer to the first photo

$$\begin{bmatrix} {}^0\omega_{e_2/0} \\ {}^0\nu_{e_2/0} \end{bmatrix} = \begin{bmatrix} {}^0J_{E_2/E_1}^{\phi} & {}^0J_{E_2/0}^L \\ \vdots & \vdots \\ {}^0J_{E_2/0_2}^L & \vdots \end{bmatrix} \dot{q}$$

■ SERIAL COMPOSITION OF ROBOT

EXAMPLE



The coordinates are expressed by:

$$\begin{cases} q_1 = x \\ q_2 = y \\ q_3 = \theta \end{cases}$$

$$\begin{cases} \dot{q}_1 = \dot{x} \\ \dot{q}_2 = \dot{y} \\ \dot{q}_3 = \dot{\theta} \end{cases}$$

We can think of building a Jacobian for this device

$$J = [P \ P \ R]$$

In this case the rotation is trivial. We assume that $E_1 = P_3$

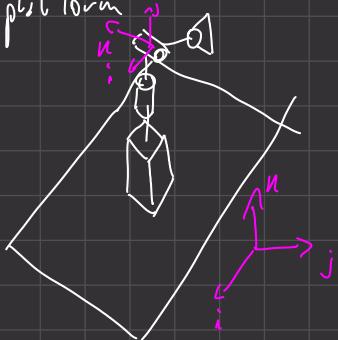
$${}^0 \mathcal{J}_{E_1/P} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$${}^0 \mathcal{J}_{F_1/\phi} = \begin{bmatrix} \phi & \phi & -K_\phi \\ -\phi & K_1 & \phi \\ \phi & -K_2 & \phi \end{bmatrix} \rightarrow E_1 = P_3$$

because of
d'Alambert's principle

$$R_0 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix}$$

Let's now assume that we have 2 axes on the platform



$${}^0 \mathcal{J}_{2/O_2}^{(2)} = \begin{bmatrix} 0, K_1 & 0, K_2 \\ \phi & \phi \end{bmatrix}$$

means the 2nd robot

We set
 $P_1 = P_0$
 the rotation is
 in the Z axis

$$R_1 \quad R_2(O_2)$$

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

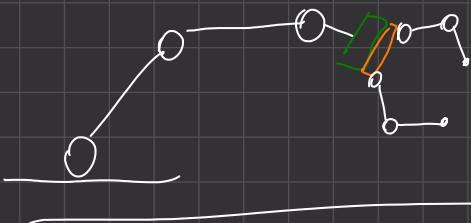
$$\begin{bmatrix} {}^0 \mathcal{J}_{e_1/P_0} & 0 & R_2 \mathcal{J}_{e_1/E_2/O_2} \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}$$

${}^0 \mathcal{J}_{E_2/O_2} \quad {}^0 \mathcal{J}_{2/O_2}$ to take into account the corner calculation

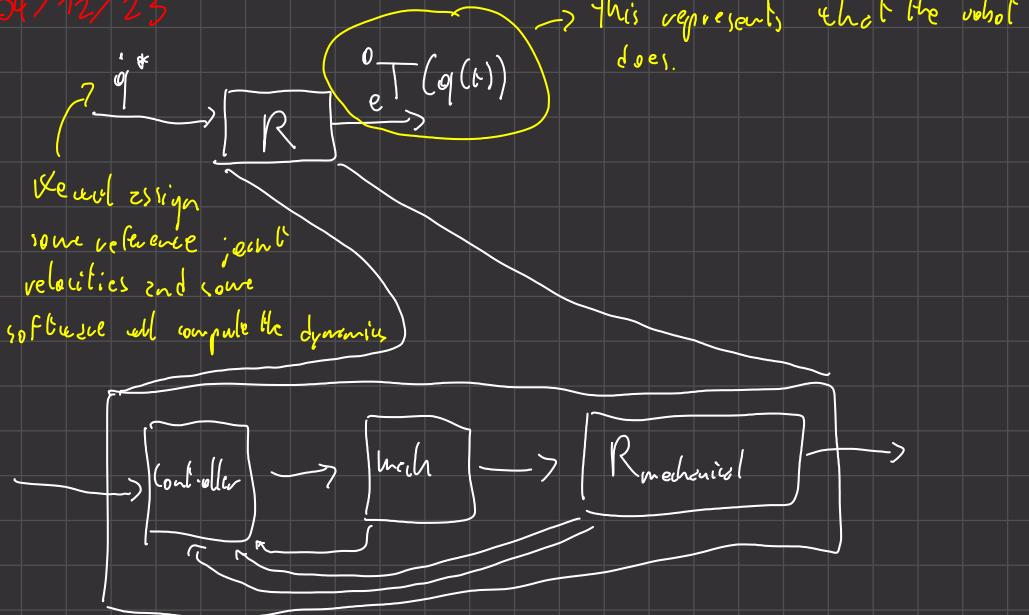
Transformation Matrices

$${}^0 T = \begin{bmatrix} {}^0 T_1 & T_2 & T_3 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_0 & 0 \\ 0 & 1 \end{bmatrix} \quad \begin{bmatrix} R_1 & 0 \\ 0 & 1 \end{bmatrix} \quad \begin{bmatrix} R_2(O_2) & 0 \\ 0 & 1 \end{bmatrix}$$

EX 2



04/12/23



Our objective is to generate \dot{q}^*

We know how to compute $\begin{cases} {}^0eT(q) \rightarrow \text{position and orientation} \\ {}^0\dot{e}_T(q) \rightarrow \text{velocity and angular} \end{cases}$
 \hookrightarrow this will be written as \dot{q} for simplicity

The problem that we need to approach is the following as:

What to send to the ee to have 2 given known and angular velocity.

Inverse Kinematic Problem

Computing \dot{q} s.t. :

$$\begin{bmatrix} {}^0\omega_{e_0} \\ {}^0v_{e_0} \end{bmatrix} \approx \begin{bmatrix} \underline{\omega}^* \\ \underline{v}^* \end{bmatrix} = \underline{x}^*$$

\uparrow \uparrow
 we angular
velocity of the ee. desired v
and ω

So we want to solve $\underline{x}^* = J\dot{q}$

There are three possible situations

1) $n=6$ (some number of dot. \Rightarrow the number of spaces in which vector x exists)

If J is square and the solution will be $\hat{q} = J^{-1}x^*$ if $\det(J) \neq 0$
 $|J| \neq 0$

If we have $|J|=0$ then (\exists ! solution)
we are in a singularity.

2) $n > 6 \Rightarrow \exists$ infinite many solutions (∞^{n-6}) = $\begin{bmatrix} A \\ b \end{bmatrix}$) \hookrightarrow Linear combinations of

3) $n < 6 \Rightarrow \nexists$ solution for any choice of x^*)
the column of A

You can find solutions here where the column of A form a subspace that includes the vector b .

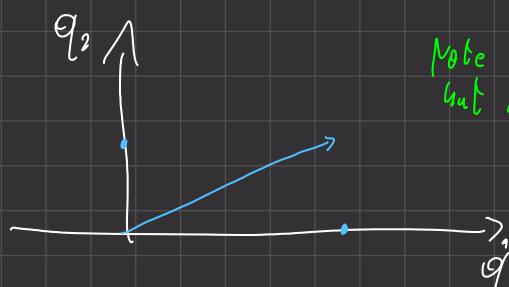
When this condition exists, then $b \in \text{span}(A)$. And this is why (3) might not have a solution.

The idea of getting a solution in these ways, the idea is to do it as an optimization problem.

The idea is to generate the smallest \hat{q} such that $\hat{q} = J\hat{q}'$ is satisfied.

We can say that

$$\hat{q} = \arg \min_{\hat{q}'} \frac{1}{2} \|\hat{q}\|^2 = \arg \min_{\hat{q}'} \frac{1}{2} \hat{q}^T \hat{q}$$



Note that \hat{q} is not a geometric vector but only an interpretation

$$\|\hat{q}\| = \sqrt{q_1^2 + q_2^2} = \hat{q}^T \hat{q}$$

We are going to search for the smallest joint combination that are contained by these conditions.

Note

$$f(\hat{q}) = \hat{q}^T A \hat{q}$$

Square symmetric matrix
 $A = A^T = R^{n \times n}$

symmetric \Rightarrow all eigenvalues strictly

positive \Rightarrow always diagonalizable

$A > 0$
This means that the matrix is strictly positive defined

$f(q) > 0$ is valid for any $q \in \mathbb{R}$

this means positive

A function like this one has already a minimum because it's convex

The important note to take from this note is that $q^T q$ is a particular case of $q^T A q$ where $A = I$

We will use the Lagrange multipliers method.

1) $\mathcal{L}(q, \lambda) \rightarrow$ we define the Lagrangian formula. This is a cost function

$$\frac{1}{2} q^T q + \lambda^T [Jq - x^*] \rightarrow \text{this is a scalar function}$$

$\frac{\partial}{\partial q} q^T q = 2q$ \downarrow
 $q + \lambda J = \dot{q} + \dot{\lambda} J$ \downarrow
 linear function in lambda

2) $\frac{\partial \mathcal{L}}{\partial q} = 0 \iff \dot{q} + J^T \lambda = 0 \rightarrow$ everything here is composed.

$$\frac{\partial \mathcal{L}}{\partial \lambda} = 0 \iff J\dot{q} - x^* = 0$$

3) $\dot{q} = J^T (J J^T)^{-1} x^*$

This is the normal formula. In practice we will use something else

We can write (3) as:

$$\dot{q} = J^{\#} x^*$$

right pseudoinverse of J.

$$J^{\#} = J^T (J [J^T J^{\#}]^{-1})$$

In the particular case of J where $n=6$ then $J^{\#} = J^{-1}$

$$\begin{cases} \dot{q} = J^T \lambda \\ J\dot{q} = x^* \end{cases} \quad -J J^T \lambda = x^*$$

If J is not singular then this is always full rank and square ($J^T J$) & 6×6 and symmetric
 $(J^T J)^T = (J J^T)$

Note that this is also semidefinite positive in the same case. Otherwise def pos

$$\lambda = (J J^T)^{-1} x^*$$

\hookrightarrow I can invert this for what I need positively

$$\dot{q} = J^T (J J^T)^{-1} x^*$$

This is our solution

Right Pseudoinverse of the J Matrix

Note. Matrix inversion should never be done in practice, it's only an analytical tool.

We have to ask ourselves two things:

What happens in a singular configuration? $f^\#$ is zero.

If this algorithm is implemented in a computer then the numbers are always wrong.

What happens if the whole is done to a singular configuration?

To address this we need the **Singular Value Decomposition**.

Assume a generic $A^{n \times m}$.

It already exists two orthonormal matrices such that:

$$A = U^T \Sigma V$$

$$U \in \mathbb{R}^{n \times n} \quad V \in \mathbb{R}^{m \times m}$$

$$\begin{bmatrix} \lambda_1 & & & \\ & \ddots & & \\ & & \lambda_n & \\ & \emptyset & \emptyset & \emptyset \end{bmatrix}$$

$$U^T = U^{-1}$$

$$V^T = V^{-1}$$

$$\begin{bmatrix} \lambda_1 & & & \\ & \ddots & & \\ & & \lambda_n & \\ & \emptyset & \emptyset & \emptyset \end{bmatrix}$$

$$\lambda_i = \sqrt{\sigma_i (A A^T)}$$

eigenvalue

this means eigenvalue
of $A A^T$. It's not multiplication

semipositive
and it's in the same
case.

We also have that $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n \geq 0$

$\underbrace{\lambda_n}_{=0}$ if in singular configuration.

Note that the singular values tend to go to 0.

We started from $\dot{x} = f \dot{x}^*$

and ended up with: $\dot{x} = f^\# \dot{x}^*$

Let's now assume that $f = U^T \Sigma V$

$$f^T = V^T \Sigma^T U$$

$$f^\# = V^T \Sigma^+ U \left[U^T \Sigma V \right]^{-1} \left(A B \right)^{-1} = B^{-1} A^{-1} \text{ but } U^{-1} = U^T$$

$$\left[U^T \left[\begin{array}{cccc} \lambda_1^2 & & & \\ & \ddots & & \\ & & \lambda_n^2 & \\ & \emptyset & \emptyset & \emptyset \end{array} \right] U \right]^{-1} = U^T \left[\begin{array}{cccc} \frac{1}{\lambda_1^2} & & & \\ & \ddots & & \\ & & \frac{1}{\lambda_n^2} & \\ & \emptyset & \emptyset & \emptyset \end{array} \right] U$$

$$J^{\#} = V^T \Sigma^{\#} U = \underbrace{V^T}_{\text{1}} \left[\begin{array}{ccc} \frac{1}{\lambda_1^2} & & \\ & \ddots & \\ & & \frac{1}{\lambda_n^2} \end{array} \right] U =$$

$$= V^T \left[\begin{array}{ccc} \frac{1}{\lambda_1} & & \emptyset \\ \emptyset & \ddots & \frac{1}{\lambda_n} \\ \emptyset & & \emptyset \end{array} \right] U$$

The steps to compute: $\hat{q} = \underset{q}{\operatorname{arg\min}} \frac{1}{2} q^T q$ are:

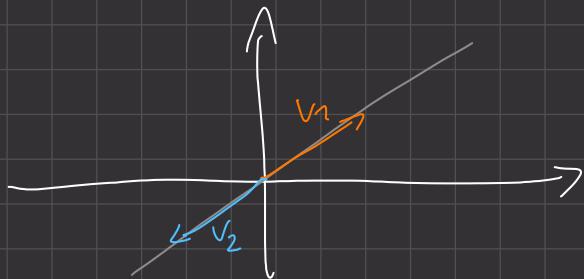
1) Compute U, V, Σ

2) Compute $\Sigma^{\#} = \left[\frac{1}{\lambda_1}, \dots, \frac{1}{\lambda_n} \right]$

3) $\hat{q} = J^{\#} \hat{x}^* = V^T \Sigma^{\#} U \hat{x}$ No inversions.

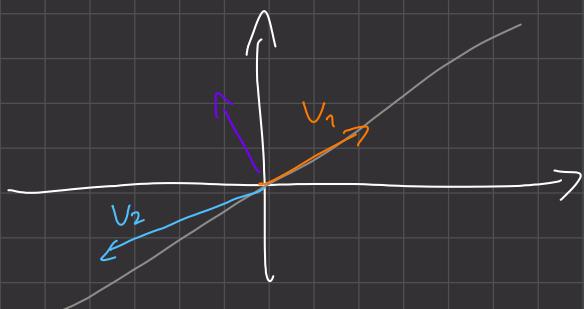
How to address sensitivity

If you consider that $\begin{bmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{bmatrix} = \begin{bmatrix} u_1 & u_2 \end{bmatrix}$ is singular which means that the column are parallel so:



Now if we want to find a solution for $Ay = b$ you can do it only if b is on the same direction

What happens if they are not aligned but quite close to it:



The problem is that to gain a component on this direction then you will need to scale the other two vectors with a huge coefficient.

So we have a general problem

$Ay = b$ and we want to estimate how much the solution varies when b is varied by a tiny amount:

$$\frac{\|\delta y\|}{\|y\|} \leq \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)} \frac{\|b\|}{\|\delta b\|} \quad N.B.$$

relative error due to the normal solution

Condition number of A

$$A(y + \delta y) = (b + \delta b)$$

High condition number is not strictly connected to singularity.

The best possible solution is $\chi(A) := 1$

In general $\chi(A) \geq 1$

If $\chi(A)$ is in the hundreds it's ok. In the thousands it becomes a problem.

In practice if the robot is leaning to a singular configuration then the smallest singular value goes to 0 .

The solution is the regularization and we add a tiny constant term to the smaller singular value so that we keep bounded the condition number. The price is that we introduce an error.

$$J^{\#} = J^+ (J J^+ + \gamma I_6)^{-1}$$

\downarrow
small coefficient

To check if you are close to a singularity you check the smallest singular value and adjust it if too low

$$\begin{aligned} \dot{q} &= \operatorname{argmin}_{\dot{q}} \dot{q}^T \dot{q} \\ \text{s.t. } J \dot{q} &= \underline{x}^* \end{aligned}$$

Least squares solution

There could be another way of formulating the problem where we are going to minimize the square of all the joint velocities with some weights.

These weights would depend on the power of the motors.

$$\dot{q} = \underset{\dot{q}}{\text{argmin}} \frac{1}{2} \dot{q}^T \underbrace{W}_{\substack{\text{weights of} \\ \text{the joints}}} \dot{q} \quad \text{s.t.} \quad J \dot{q} = \dot{x}^*$$

$$W = W^T > 0$$

Examples

$$\textcircled{2} \quad W = \begin{bmatrix} w_1 & \cdot & \phi \\ \phi & \cdot & w_n \end{bmatrix}$$

\textcircled{3} $|q_i| \leq q_i^{max} \rightarrow$ the layer q_i^{max} , the layer I accept values.

\textcircled{4} $W \leftarrow A(q_i)$, Robot inertia matrix $\dot{q}^T A(q_i) \dot{q} \equiv \text{Kinetic energy}$
 $A = A^T > 0$

$$\mathcal{L}(\dot{q}, \lambda) = \frac{1}{2} \dot{q}^T W \dot{q} + \lambda^T [J \dot{q} - \dot{x}^*]$$

$$\frac{\partial \mathcal{L}}{\partial \dot{q}} = 0 \quad W \dot{q} + J^T \lambda = 0$$

$$\dot{q} = -W^{-1} J^T \lambda$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = 0 \quad J \dot{q} = \dot{x}^* = 0$$

$$(J W J^T) \lambda = \dot{x}^* \Rightarrow$$

$$\lambda = -(J W J^T)^{-1} \dot{x}^*$$

$$\dot{q} = W^{-1} J^T [J W^{-1} J^T]^{-1} \dot{x}^*$$

\hookrightarrow Weighted pseudoinverse matrix

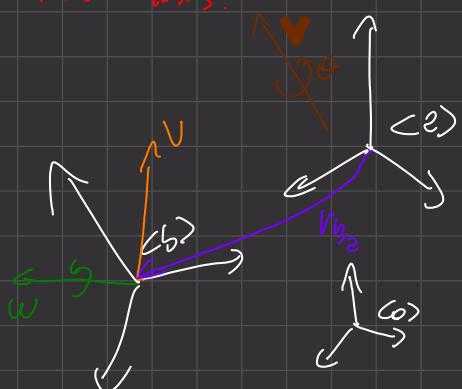
$$W = W^T > 0 \quad W \stackrel{?}{=} \dots$$

The square roots of the system values in this case are the same as eigenvalues?

N.B

$\int \psi^4$ Decoupled least squares.

Robot tasks.



The point is here to drive e_2 in order to send e_2 in the same position of (e_2) .

So we have 0T_e and ${}_bT_e$

I want to move b such that after z while reaches e_2 .

$${}^0T_e = {}^0_bT$$

If 0R_e is attached to e vector we can compute the IRP and controllable joints velocities.

Now to calculate v and ω . I can define this vector v_{b2}

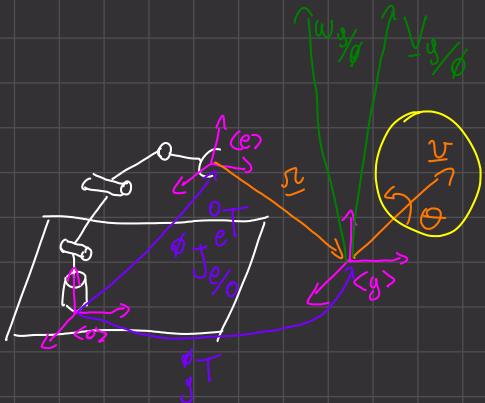
I must send v_{b2} to ϕ to put the frames coincident.

There exists also ${}_bR$ that needs to go to 1

Instead of R we send one vector v and an angle θ that allow for the same rotation R .

11/12/23

Closed loop kinematic control



We also need to know the velocity of the goal frame

This is like saying gR .
It's not the velocity of the goal frame.

If V_g say ω are constant we will reach steady state in the error between the positions of c_g and c_e .

$$\lim_{t \rightarrow \infty} e^T = g^T \Leftrightarrow \lim_{t \rightarrow \infty} g^T = I$$

$$\Rightarrow \begin{cases} \lim_{t \rightarrow \infty} e^R = I \Leftrightarrow \lim_{t \rightarrow \infty} R = 0 \\ \lim_{t \rightarrow \infty} \Sigma = \emptyset \end{cases}$$

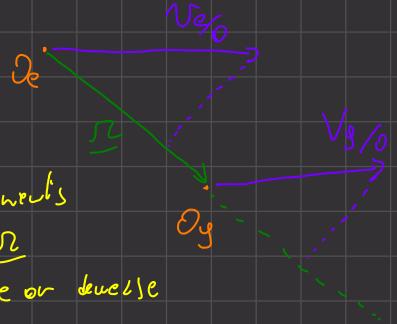
DISTANCE ZEROING

$$\lim_{t \rightarrow \infty} \underline{r} = 0 \quad \underline{r} = (O_g - O_e)$$

$$\frac{d\phi}{dt} \underline{r} = \frac{d\phi}{dt} (O_g - O_e) = \underline{v}_{g/\theta} - \underline{v}_{e/\theta}$$

The derivative of r w.r.t. θ

this is the actual derivative of r

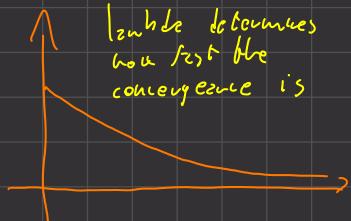


Only the components that are on \underline{r} make r increase or decrease in length.

Desired derivative of r ($\dot{\underline{r}}$)

$$\dot{\underline{r}} \triangleq -\lambda \underline{r}$$

\rightsquigarrow First order differential equation



We substitute the desired derivative with the actual derivative

$$\dot{\underline{r}} = -\lambda \underline{r} = \underline{v}_{g/\theta} - \underline{v}_{e/\theta}^* \rightarrow \text{desired}$$

$$\underline{v}_{e/\theta}^* = \underline{v}_{g/\theta} + \lambda \underline{r}$$

\hookrightarrow feed forward element

With v_g we have the possibility to be on top of the goal for 2nd follow it. Otherwise we will have to follow it at a distance because if $r \neq 0$ then v_{eg} would be 0

There is another way to calculate $\dot{v}_{e\phi}^*$. Previously it was implying that r is not rotating. With this the r cannot rotate which is not a very general behaviour.

To extend this, if r is a unit vector multiplied by a scalar, we can consider the norm to go to ϕ .

Decreasing the norm of the vector.

$$\underline{r} = \underline{\mu} \underline{r} \quad \text{derivative of } \underline{r}$$

$$\dot{\underline{r}} = \underline{\mu} \cdot (\underline{v}_{g\phi} - \underline{v}_{e\phi}) \longrightarrow ???$$

If now we want \underline{r} to be tangent to ϕ then I can do a similar step:

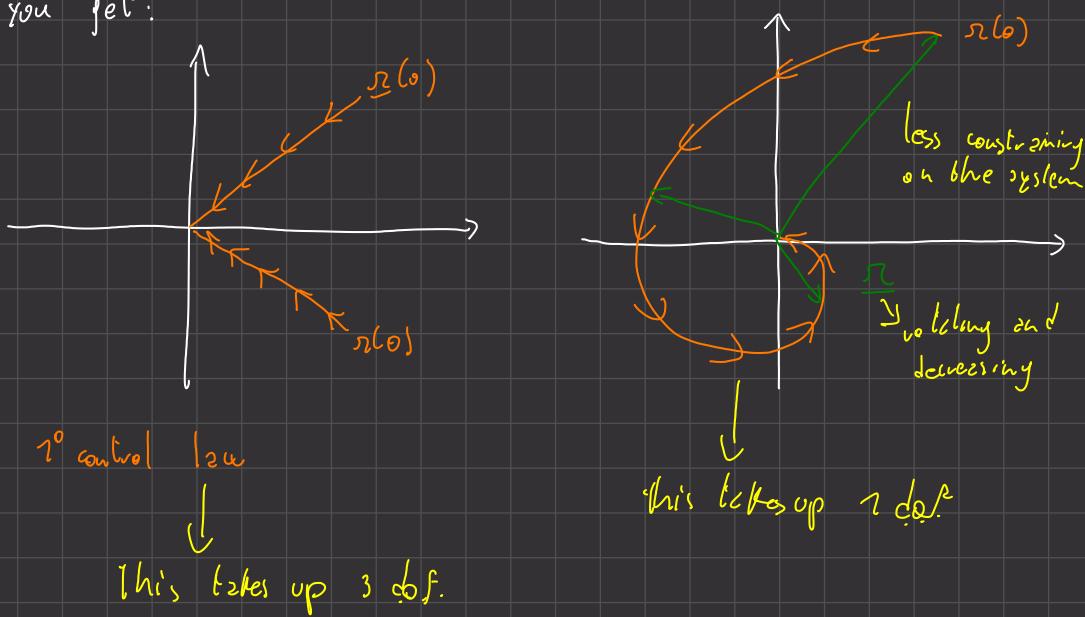
$$\dot{\underline{r}} = -\lambda \underline{r}$$

$$-\lambda \underline{r} = \underline{\mu} \cdot (\underline{v}_{g\phi} - \underline{v}_{e\phi}^*)$$

$$-\lambda \underline{r} = \underline{\mu} \cdot \underline{v}_{g\phi} - \underline{\mu} \underline{v}_{e\phi}^* \quad \rightarrow \text{if we want the derivative of the scalar to be zero then the Jacobian should be } 0. \quad \text{N.B.}$$

$$\underline{\mu} \cdot \underline{v}_{e\phi}^* = \underline{\mu} \cdot \underline{v}_{g\phi} + \lambda \underline{r}$$

The behaviour in cartesian space of those two different equations you get:



Misalignment zeroing (angular part)

$$D_\theta \rho = D_\theta \rho + \omega_{e\phi} \times \rho \rightarrow \text{law of change of differentiation frame.}$$

$$\left(\frac{d}{dt} \right) D_\theta \rho = D_\theta (\underline{v} \theta) = \underline{v} \dot{\theta} + \theta D_\theta (\underline{v}) =$$

$$D_\theta \rho = D_\theta (\underline{v} \theta) = \underline{v} \dot{\theta} + \theta D_\theta (\underline{v}) =$$

$$= \underbrace{v(v) \omega_{g/e}}_{\text{parallel to } \underline{v}} + \underbrace{N(\theta) \omega_{g/e}}_{\text{orthogonal to } \underline{v}}$$

Operator that gives us how omega makes ρ rotate
operator equivalent to v^T

$$N(\theta) = \frac{\theta}{2} \left[\frac{1}{\tan(\frac{\theta}{2})} - [\underline{v} \times] \right] (1 - \underline{v}(\underline{v}))$$

this is the actual derivative of ρ

$$\dot{\rho}_e = -\lambda \rho \rightarrow \text{desired behaviour}$$

$= -\lambda \theta \underline{v}$ If θ is ϕ then the orthogonal part goes to 0 in the ρ derivative, so we must go along v otherwise

$$\omega_{g/e} = -\lambda \theta \underline{v} = -\lambda \rho$$

$$\omega_{g/\phi} - \omega_{e/\phi}^* = -\lambda \rho$$

$$\boxed{\omega_{e/\phi}^* = \omega_{g/\phi} + \lambda \rho}$$

Reference behavior for $\underline{v}_{e/\phi}$

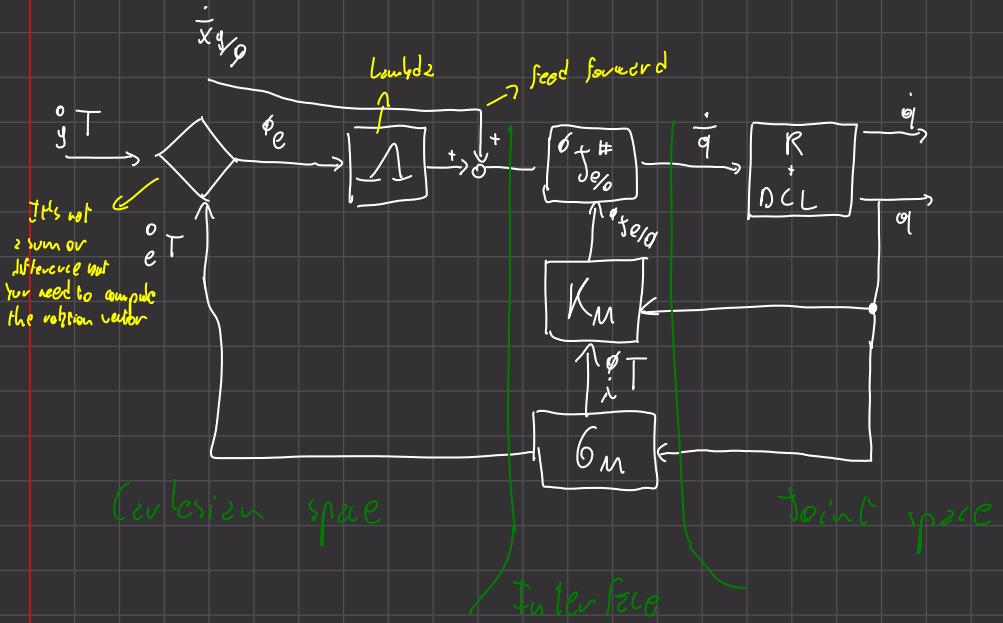
$$\begin{bmatrix} \dot{\phi}_{x_{e/\phi}} \\ \dot{\phi}_{v_{e/\phi}} \end{bmatrix} = \Delta \begin{bmatrix} \phi_e \\ \rho \end{bmatrix} + \begin{bmatrix} \dot{\phi}_{x_{g/\phi}} \\ \dot{\phi}_{v_{g/\phi}} \end{bmatrix}$$

↓ feed forward

$$\Delta = \begin{bmatrix} \lambda_1 I & \phi_{3x3} \\ \phi_{3x3} & \lambda_2 I \end{bmatrix}$$

$$\boxed{\dot{\phi}_{x_{e/\phi}} = \Delta \underline{\phi}_e + \dot{\phi}_{x_{g/\phi}}} \rightarrow 6^{\text{d}} \text{ equation}$$

Scheme



Alternative ways to control orientation

$$\dot{\varphi} = \begin{bmatrix} \psi \\ \theta \\ \phi \end{bmatrix} \quad \text{YAW - PITCH - ROLL}$$

$$\tilde{\dot{\varphi}} = \dot{\varphi}_g - \dot{\varphi}_e \quad \rightarrow \text{error}$$

$$\ddot{\tilde{\varphi}} = \ddot{\tilde{\varphi}}_g - \ddot{\tilde{\varphi}}_e = \quad \rightarrow \text{actual derivative of yaw-pitch-roll angles}$$

$$= T_{g/\theta}^{-1}(\varphi_y) \omega_{y/\theta} - T_{e/\theta}^{-1}(\varphi_e) \omega_{e/\theta}$$

$${}^2 T_{g/\theta}(\ell) \dot{\varphi} = \begin{bmatrix} 0 & -s\psi & c\psi c\theta \\ 0 & c\psi & s\psi c\theta \\ 1 & 0 & -s\theta \end{bmatrix} \begin{bmatrix} \dot{\psi} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \omega_b/\theta \quad N.B.$$

$$\ddot{\tilde{\varphi}}^* = -\lambda \tilde{\dot{\varphi}}$$

$$\boxed{\omega_{e/\theta}^* = T_{e/\theta}^{-1}(\varphi_e) \left[T_{g/\theta}^{-1}(\varphi_y) \omega_{y/\theta} + \lambda \tilde{\dot{\varphi}} \right]}$$

this is a coupling of the euler angles to the desired pose

The resultant vec is different because in general we want the rotmat and the rot vec. It would be of type $-dp$.

Alternative 2

$$C \triangleq gR e^T \rightarrow \text{we want this to go to 1} \\ \rightarrow you cannot do this product unless from 100\% at the limit$$

$$C = gR e^T$$

$$\underbrace{C^T = [w_c x]}_{\text{step down}} = g^T R^T e^T = [w_{y/e} x]$$

$$\rightarrow w_c = w_{y/e} = w_{y_0} - \omega_{e_0}$$

$$w_{c_0} = w_{y_0} - \omega_c \quad \rightarrow \text{single axis corresponding to the rotation matrix}$$

if we choose $w_c = -K p_c$

$$\dot{p}_c = -K \dot{p}_c$$

$$w_c^* = w_{y_0} + K \dot{p}_c$$

Same as the first one, just a different way of deriving it.

$$\text{If we choose } w_c = -K v_{ex} \left(\frac{C - C^T}{2} \right) = -K v_c \sin(\theta_c)$$

$$\dot{p}_c = -K \underbrace{v_c \sin \theta_c}_{\rightarrow \text{the vex gives you this part but it's not exponential.}} \rightarrow \text{This is not an exponential conveyance}$$

The advantage of this solution is that you avoid the problem when you not have sign of θ so v is undefined. If I have this problem why he I can use the previous formula.