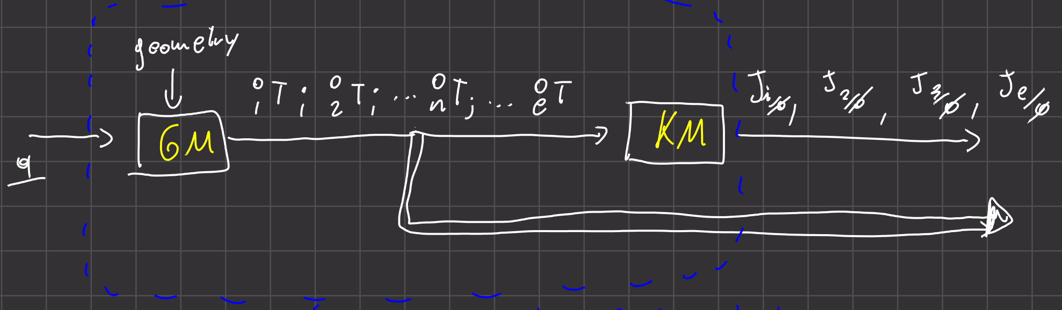
25/09/23

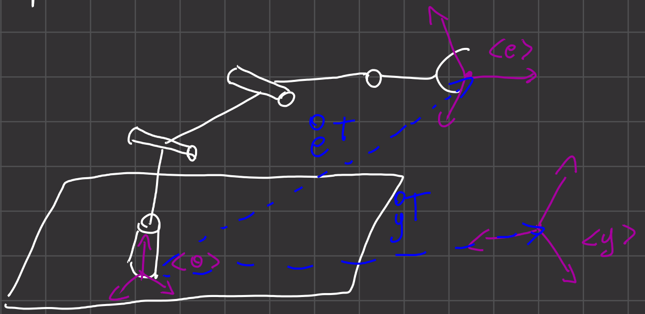
$$J_{i/0}(q) = J_{i/0}\left({}^0_1T(\underline{q});\ {}^0_2T(\underline{q});\ {}^0_3T(\underline{q}) \dots {}^0_mT(\underline{q})\right)$$

The info that we need to describe a Jacobian is not only the angle of every joint but also the distances and all geometrical distances and all the transf. matrices contain this information. Inside J there are all these geometric distances. J contains 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 GM.



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?



<g> goal frame

${}^0_gT$ desired transformation matrix

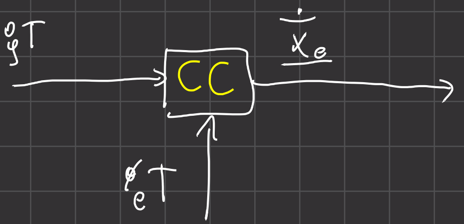
We want to make <e> approach <g>

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

If I want to go to position of <g> with <e> how should I move? For this we should have a vector in the direction from <e> to <g>. But keep in mind that this is not the only solution because any vector orthogonal to the distance of <e> would turn but get closer if there is a component towards <g>. 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 now introduce the Cartesian Controller



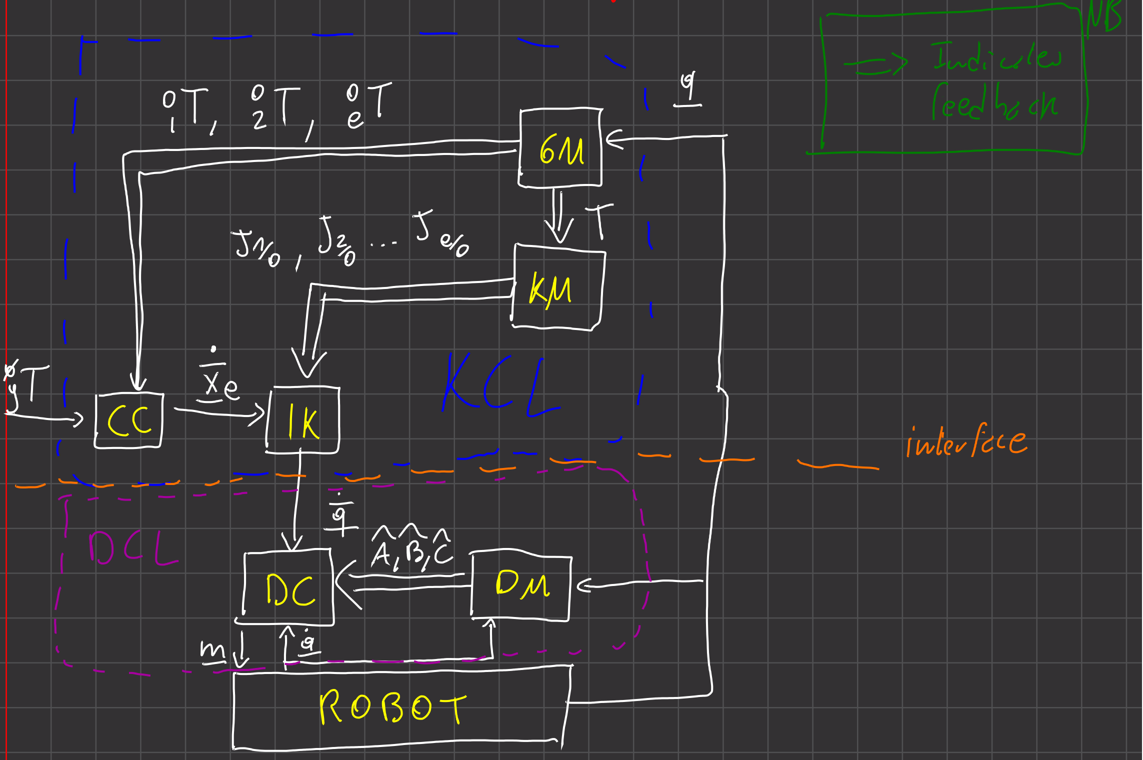
$\dot{x}_e \in \mathbb{R}^6$

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 a 4-degree of freedom robot we can go in a subset of the 6-dof space.
In the other hand if we have a more than 6-dof robot sometimes we cannot garantee 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 kinematic block. That provides the best approximation to this problem. Remember the workspace of the robot.

The overall control scheme is the following:



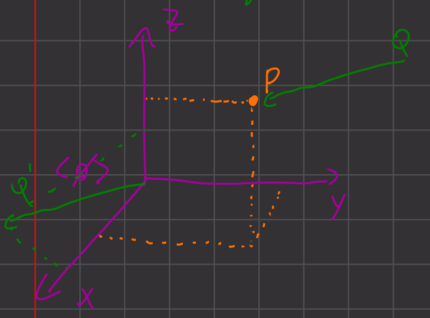
The robot wants torque (m). From the robot we measure q and $\dot{q}$. The first thing we do in the DCL is building a controller that builds the desired velocity. The objective of the DCL is to follow as close as possible the reference velocity. Once you have done this you need to drive the ee to the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

position of the Frames and with KM their movements with Jacobian.
Feeding this to the Jacobian we see how the system should move in space.
We get the desired velocity that gets mapped to the desired velocity with the inverse kinematic.

TODO

Ch. 1 Geometric Foundamentals

Coordinate Systems (Frame)



Oriented tripod of ortogonal 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 lenght of the vectors that allow me 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 wich I apply the force is very important, but when I want to know the distance the point on wich is applied is not very important.

Often we can also say:

Projected vector??

$$^{\phi}(P-Q) = {^{\phi}P} - {^{\phi}Q} \triangleq {^{\phi}\underline{v}_{p,q}}$$

the vector that joins q to p

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

Grassman Rule

$$^{\phi}P = {^{\phi}Q} + {^{\phi}\underline{v}_{p,q}}$$

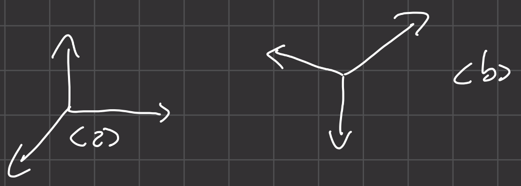
In the frame unit vectors are immediately representable:

$x \quad \emptyset\, i_\emptyset \quad [1\ 0\ 0]^T$

$y \quad \emptyset\, j_\emptyset \quad [0\ 1\ 0]^T$

$z \quad \emptyset\, k_\emptyset \quad [0\ 0\ 1]^T$

Let's now consider two frames



Set of information needed to describe how <b> is with respect to <a> is

$$\left\{ {}^a\underline{O}_b,\ \underbrace{{}^a i_b,\ {}^a j_b,\ {}^a k_b}_{\text{unit vectors of } b \text{ projected on } \langle a\rangle \text{ to code the rotation of frame } \langle b\rangle \text{ in respect to } \langle a\rangle.} \right\}$$

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 unit vectors of <b>.

$$\underline{v} = \underline{i}_b x_b + \underline{j}_b y_b + \underline{k}_b z_b$$

i, j, k form a base.

If now I perform projection of $\underline{v}$ on <a>

$$^a\underline{v} = {}^a\underline{i}_b x_b + {}^a\underline{j}_b y_b + {}^a\underline{k}_b z_b$$

Now representing as a vector-matrix multiplication we get:

$$^a\underline{v} = \underbrace{\left[{}^a\underline{i}_b \quad {}^a\underline{j}_b \quad {}^a\underline{k}_b \right]}_{{}^a_bR \,=\, \text{ROTATION MATRIX}} \begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix}$$

referred frame $\longrightarrow a$

referring frame $\longrightarrow b$

$${}^a_bR = \text{ROTATION MATRIX}$$

$$^a\underline{v} = {}^a_bR\ {}^b\underline{v}$$

This is another way of seeing the multiplication.

In Matlab this will be reported like a_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:

$$a\_distance = a\_R\_b \quad b\_distance.$$

R is a ortho-normal matrix
columns are orthogonal to each other; unitary lenght

$${}^a_bR \; {}^a_bR^T = I_{n \times n}$$

Given this the matrix has some properties:
Recalling that:

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

Beeing ortho-normal means:

$$\begin{cases} {}^a_bR \; {}^a_bR^T = I_{3\times3} \\ {}^a_bR^T \; {}^a_bR = I_{3\times3} \end{cases} \Rightarrow {}^a_bR^T = {}^a_bR^{-1} = {}^b_aR$$

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 $\rightarrow$ $\det({}^a_bR) = 1$

this to preserve the direction meaning the right handed frame.

For example

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

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

$$SO(n) = \{ R \in \mathbb{R}^{n\times n} : R^TR = I_{n\times n}, \; \underbrace{\det(R) = +1}_{\text{Special for this limitation}} \}$$

Special orthogonal group

This group has some properties : N.B

TODO on what? 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 $(R_1 R_2) R_3 = R_1 (R_2 R_3)$

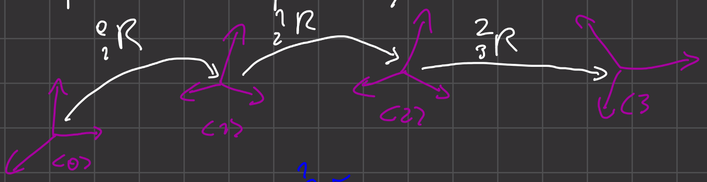
A member of $SO(3)$ is a Rotation Matrix

A consideration about ${}^a_bR^T = {}^b_aR$ is that this is also equivalent to:

$${}^a_bR = \begin{bmatrix} {}^a\underline{i}_b & {}^aj_b & {}^ak_b \end{bmatrix} = \begin{bmatrix} {}^bi_a^T \\ {}^bj_a^T \\ {}^bk_a^T \end{bmatrix}$$

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

Our manipulator is composed by a tree of frames:

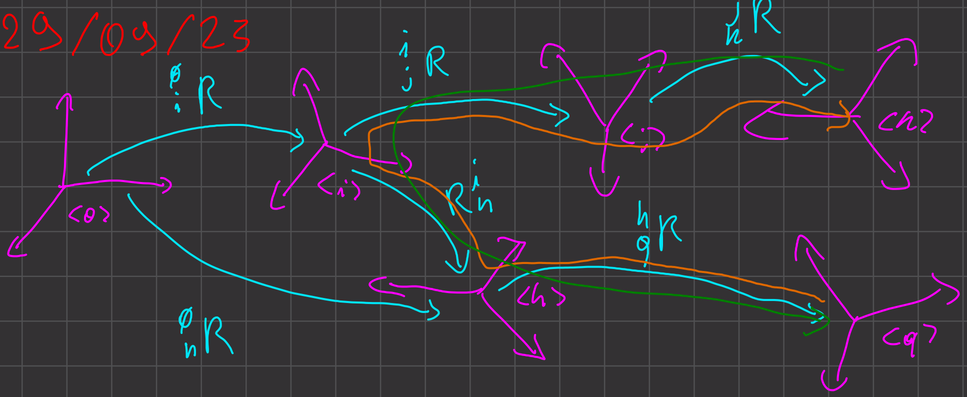


$${}^0v = \underbrace{{}^0_1R \overbrace{{}^1_2R \underbrace{{}^2_3R \; {}^3v}_{{}^2v}}^{{}^1v}}_{{}^0v}$$

The notation helps to check if everything is done correctly

N.B. This holds for every v so for definition ${}^0_1R \, {}^1_2R \, {}^2_3R = {}^0_3R$

29/09/23



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

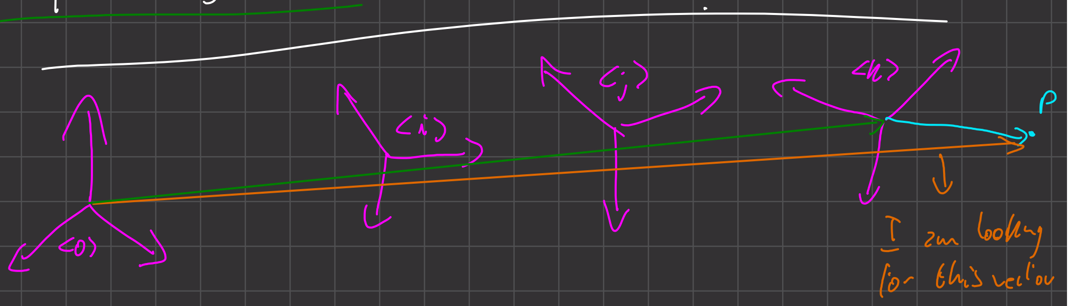
postmultiplication

How do you compute: ${}^{q}_{h}R = \underbrace{{}^{h}_{q}R^{T}\,{}^{i}_{h}R^{T}\,{}^{i}_{j}R\,{}^{j}_{h}R}_{} = \overrightarrow{{}^{q}_{h}R\,{}^{h}_{i}R\,{}^{i}_{j}R\,{}^{j}_{h}R}$

We can also do the same thing following this path:

premultiplication

$$\underline{{}^{q}_{h}R = \overleftarrow{{}^{h}_{q}R^{T}\,{}^{i}_{h}R^{T}\,{}^{i}_{j}R\,{}^{j}_{h}R}}$$



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>.

$${}^{K}p = {}^{K}(P - O_{h})$$

$${}^{0}p = {}^{0}O_{h} + {}^{0}(P - O_{h})$$

$${}^{0}p = {}^{0}O_{h} + {}^{0}_{h}R\,{}^{h}p$$

Omogeneous Coordinates

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

$${}^{0}\underline{\bar{p}} = {}^{0}_{h}T\,{}^{h}\underline{\bar{p}} \qquad {}^{0}_{h}T \triangleq \left[\begin{array}{c|c} {}^{0}_{h}R & {}^{0}O_{h} \\ \hline 0_{3\times 1} & 1 \end{array}\right] \in \mathbb{R}^{4\times 4}$$

So if I do

$$\underbrace{\left[\begin{array}{c|c} {}^{0}_{h}R & {}^{0}O_{h} \\ \hline 0_{3\times 1} & 1 \end{array}\right]}_{{}^{0}_{h}T}\underbrace{\begin{bmatrix} {}^{h}\underline{p} \\ 1 \end{bmatrix}}_{{}^{h}\bar{p}} \underset{=}{=} \underbrace{\begin{bmatrix} {}^{0}_{h}R\,{}^{h}p + {}^{0}O_{h} \\ 1 \end{bmatrix}}_{{}^{0}\bar{p}}$$

Inverse transformation matrix

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

We know that:

$${}^a_bT^{-1} = {}^b_aT$$

N.B.

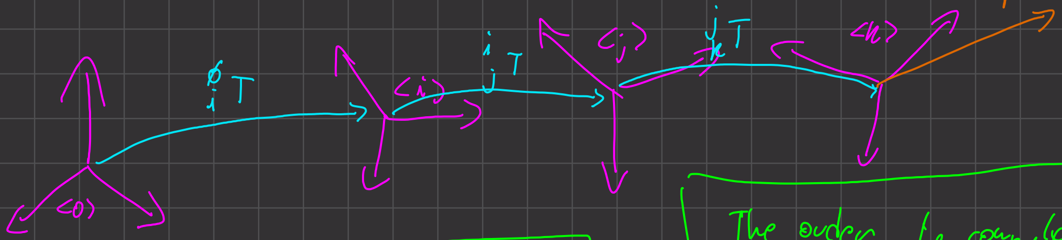
BUT

$${}^a_bT^{-1} \neq {}^a_bT^T$$

$${}^a_bT^{-1} = \begin{bmatrix} {}^a_bR^T & -{}^a_bR^T\,{}^aO_b \\ 0_{3\times 1} & 1 \end{bmatrix}$$

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

This is very handfull because if we consider the original diagram



$${}^0\bar{p} = \underbrace{{}^0_iT\;{}^i_jT\;{}^j_kT\;{}^j_kT}_{{}^0_kT}\;{}^k\bar{p}$$

The order to compute the transformation matrix is from <0> to <e>

It's also possible to describe a vector using homogeneous coordinates.
It's done by justaposing $\emptyset$ to the vector.

$$p = \begin{bmatrix} {}^np \\ \emptyset \end{bmatrix}_{4\times 1}$$

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

${}^{a}_{b}R$ - Rotation matrices are ortonormal. vectors are ortogonal to each other; modulus = 1 for every vector.

So I have:

$$\begin{cases} {}^{a}i_b^T \, {}^{a}i_b = 1 \\ {}^{a}j_b^T \, {}^{a}j_b = 1 \\ {}^{a}k_b^T \, {}^{a}k_b = 1 \end{cases} \qquad \begin{cases} {}^{a}i_b \cdot {}^{a}j_b^T = 0 \\ {}^{a}j_b^T \, {}^{a}k_b = 0 \\ {}^{a}k_b^T \, {}^{a}i_b^T = 0 \end{cases}$$

dot product

${}^{a}_{a}R$ 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 Oiler

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 ; . . .
- Tait-Bryan angles
  x-y-z ; z-y-x ; . . .

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

$$\boxed{z - y' - x''}$$

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

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

YAW PITCH ROLL

The apex stresses that we are talking about intrinsic rotation

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

02/10/23

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

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

Not needed to know by heart

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

$$\theta = \underbrace{\text{atan}2}_{\text{software function to compute atan on 4 quadrants instead of two.}}\left(-r_{3,1}, \sqrt{r_{11}^2 + r_{21}^2}\right) \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$$

If $\cos\theta \neq 0$: $\psi = \text{atan}2(r_{21}, r_{11})$
$\phi = \text{atan}2(r_{32}, r_{33})$

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

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

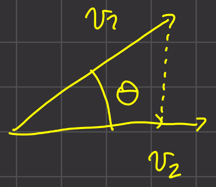
If they are linear independent they can form the base of your n-based dimensio frame.

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

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

- Scalar product

$$(\underline{v}_1 \cdot \underline{v}_2) = (\underline{v}_2 \cdot \underline{v}_1) = |v_2||v_1| \cos\theta$$



$$\boxed{(v_1 \cdot v_2) = \underbrace{{}^a\underline{v}_1^T \, {}^a\underline{v}_2}_{\text{projected on } \langle a \rangle}}$$

I can consider $(\underline{v}_1 \cdot) = (\underline{v}_1^T)_{1\times 3}$ linear operation

- Vector product

$$(\underline{v}_1 \wedge \underline{v}_2) = (\underline{v}_1 \times \underline{v}_2) = \begin{cases} \overbrace{|v_1 \times v_2|}^{\text{modulus}} = |v_1||v_2| \sin(\theta) \\ \underline{v}_1 \times \underline{v}_2 \perp (\underline{v}_1, \underline{v}_2) \end{cases}$$

↳ minimum angle

The direction is ortogonal to both v_1 and v_2 according to the right hand rule

- Properties:
  - Anti-commutative: $(\underline{v}_1 \times \underline{v}_2) = -(\underline{v}_2 \times \underline{v}_1)$
  - $^a(\underline{v}_1 \times \underline{v}_2) = \underbrace{{}^a[\underline{v}_1 \times]}_{\substack{\text{Operator} \\ 3\times 3}} \, {}^a\underline{v}_2 = \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix} {}^a\underline{v}_2$ → Components of $^a\underline{v}_1$

    ↳ This matrix is skew-simmetric

    For this reason
  - $^a[\underline{v}_1 \times]^T = -\,{}^a[\underline{v}_1 \times]$ → axial vector

    Sometimes this can also be represented like: $[\underline{v}_1 \times] = S(v_1)$
  - Inverse mapping: vex operator → $vex(S(\underline{v}_1)) = \underline{v}_1$

$$\boxed{\begin{array}{l} \forall M: M \in \mathbb{R}^{3\times 3}, M = -M^T \\ M\underline{y} = vex(M) \times \underline{y} \end{array}}$$

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

Power of skew-symmetric 3x3 real matrices associated with a unit vector

$h \in \mathbb{R}^{3\times1}$

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

$[\underline{h}\times]^2 = (\underline{h}\,\underline{h}^T - I)$

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

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

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

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

$\vdots$

After the 4th power they repeat from the first one.

- In general we state that:

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

Rule for odd numbers (skew symmetric)

$$[h\times]^{2(i+1)} = (-1)^i (h h^T - I)$$

Rule for even numbers (symmetric)

$(I - hh^T)$ projects a vector on the plane orthogonal to h (toto check)

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

$(I - \underline{h}\,\underline{h}^T)\underline{h} = \emptyset$ because $(h - h\underbrace{h^T h}_{1}) = \emptyset$

- $trace([h\times]^{2i+1}) = 0$

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

tr $\to$ sum of all the elements on the main diagonal in a matrix

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

$A = A^T$ symmetrical

$A = -A^T$ skew symmetrical

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

no symm or skew-symm

Linear Operators between Vectors

Consider: · frame $<b>$
- any projected vector ${}^b\underline{v}$
- ${}^b\underline{\omega}$ subject to ${}^b\underline{\omega} = {}^bL\,{}^b\underline{v}$ — linear operator $L \in \mathbb{R}^{3\times3}$

$${}^a\underline{\omega} = {}^a_bR\,{}^b\underline{\omega} = {}^a_bR\,{}^bL\,{}^b\underline{v} = \boxed{{}^a_bR\,{}^bL\,{}^a_bR^T}\,{}^a\underline{v} = {}^aL\,{}^a\underline{v}$$

$$\boxed{{}^aL = {}^a_bR\,{}^bL\,{}^a_bR^T}$$

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

For example:

Vector product operator

$${}^b(\underline{\omega} \times \underline{v}) = {}^b[\underline{\omega}\times]\,\underline{v}$$

$${}^a(\underline{\omega} \times \underline{v}) = {}^a_bR\,{}^b[\underline{\omega}\times]\,{}^a_bR^T\,{}^a\underline{v} = {}^a[\underline{\omega}\times]\,{}^a\underline{v}$$

06/10/23

Matrix exponential operator

$${}^bL = e^{{}^bA} \qquad {}^bA \in \mathbb{R}^{3\times3}$$

$${}^bL = \sum_{k=0}^{\infty} \frac{({}^bA)^k}{k!}$$

As any other linear operator we can write:

$${}^b\underline{\omega} = {}^bL\,{}^b\underline{v} = e^{{}^bA}\,{}^b\underline{v}$$

Now we want to know ${}^aL$:

$${}^aL = {}^a_bR\,{}^bL\,{}^a_bR^T = {}^a_bR\,e^{{}^bA}\,{}^a_bR^T$$

Now if we substitute the definition of the exponential:

$$= {}^a_bR\left(\sum_{k=0}^{\infty}\frac{({}^bA)^k}{k!}\right){}^a_bR^T = \sum_{k=0}^{\infty}\frac{{}^a_bR\,({}^bA)^k\,{}^a_bR^T}{k!}$$

$$= \sum_{k=0}^{\infty}\frac{({}^a_bR\,{}^bA\,{}^a_bR^T)^k}{k!}$$

this step is possible because:

$$\underbrace{({}^a_bR\,{}^bA\,\cancel{{}^a_bR^T})(\cancel{{}^a_bR}\,{}^bA\,\cancel{{}^a_bR^T})\cdots(\cancel{{}^a_bR}\,{}^bA\,\cancel{{}^a_bR^T})(\cancel{{}^a_bR}\,{}^bA\,{}^a_bR^T)}_{k \text{ times}}$$

Identity

Identity

identity

To conlude

$$^{a}L = e^{^a_bR\, ^bA\, ^a_bR^T} = e^{^aA}$$

Properties of the matrix exponential

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

Exponential representation of rotation matrixes.

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

To prove this:

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

$$= e^{-\theta[\underline{h}\times]} e^{\theta[\underline{h}\times]} = I_{3\times3}$$

This determinant is either +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 traspose is $I_{3\times3}$ so it's a Rotation Matrix.

Angle axes parameters.

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

$h \in \mathbb{R}^{3\times1}, \ |h| = 1$ axis

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

n gives you the axis around wich to rotate and θ gives you the quantity.

Rotation Vector

$\rho = \theta h \in \mathbb{R}^{3\times1}$ is called the rotation vector associated to the rotation matrix $e^{\theta[\underline{h}\times]}$

Angle axes vector

$$\underline{v} = \begin{bmatrix} \underline{h} \\ \theta \end{bmatrix} \in \mathbb{R}^{4\times 1}$$

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

Let's now consider how the exponential is computed:

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

We split even and odd powers of this infinite summation:

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

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

Rodrigues Formula

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

N.B.

→ h is always a unit vector

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

Equivalent Angle-axis inverse problem

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

On the diagonal you get the square of the components.

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

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

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

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

Like this we can find more than one parameter

For example if:

$\theta = \emptyset$ then h is arbitrary

$\theta = \pi$ then

Let's try it:

$$e^{\pi[h\times]} = I_{3\times3} + 2(\underline{h}\underline{h}^T - I) = 2hh^T - I_{3\times3} =$$

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

From this we can conclude that the modulus of h =

$$|h_i| = +\sqrt{\frac{r_{ii}+1}{2}} \qquad \forall_i = 1,2,3$$

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

$$h_{\bar{i}} = \pm\sqrt{\frac{r_{\bar{i}\bar{i}}+1}{2}}$$

The generic $h_j = \operatorname{sgn}(h_{\bar{i}})\operatorname{sgn}(r_{\bar{i}j})\sqrt{\frac{r_{jj}+1}{2}} \qquad \forall_j \in (1,2,3)\setminus\bar{i}$

If you pick $\bar{i}=1$
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

$$R - R^T = \sin\theta[h\times] - \sin\theta[h\times]^T$$

$$R - R^T = \cancel{I_{3\times3}} + \sin\theta[h\times] + \cancel{(1-\cos\theta)[h\times]^2}$$
$$-\cancel{I_{3\times3}} - \sin\theta[h\times] - \cancel{(1-\cos\theta)[h\times]^2}$$

symmetrical
=

$$= 2\sin\theta[\underline{h}\times]$$

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

This extracts the
axial vector from the
skew symmetrical Matrix

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

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

Euler theorem

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

08/10/23

Unit quaternion

These are defined by $\underline{q} \in \mathbb{R}^{4\times1}$

$$\underline{q} \triangleq \begin{bmatrix} \mu \\ \varepsilon \end{bmatrix} \quad \text{where} \begin{cases} \mu \in \mathbb{R} \\ \varepsilon \in \mathbb{R}^{3\times1} \end{cases}$$

$\mu^2 + \|\varepsilon\|^2 = 1$ ← this is why it's called unit quaternion

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

Trigonometric Properties

First let's recall:

I $\sin(\varphi+\beta) = \sin\varphi\cos\beta + \sin\beta\cos\varphi$

II $\cos(\varphi+\beta) = \cos\varphi\cos\beta - \operatorname{sen}\varphi\operatorname{sen}\beta$

} Prostaferesi Formulas

Let's now consider:

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

Now we get what appears in the Rodriguez formula:

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

II $\cos\theta = \cos^2\frac{\theta}{2} - \operatorname{sen}^2\frac{\theta}{2}$

Now if we do

$$\cos\theta = \cos^2\frac{\theta}{2} \underbrace{- \operatorname{sen}^2\frac{\theta}{2} + \cos^2\left(\frac{\theta}{2}\right)}_{=1} - \cos^2\left(\frac{\theta}{2}\right)$$

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

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

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

Rodriguez V2

$$R = I + \sin\theta\,[\underline{h}\times] + (1-\cos\theta)\,[\underline{h}\times]^2 =$$

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

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

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

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

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

If $\theta = 0$ then $\varepsilon = 0$. There is no indefinition of the axis vector $\underline{h}$ because I am using sin. It's always well defined.

Inverse mapping: from R → quaternions

To compute the inverse mapping we do:

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

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

$$tr(R) = 3 + 2(\|\varepsilon\|^2 - 3\|\varepsilon\|^2)$$

$$tr(R) = 3 - 4\|\varepsilon\|^2$$

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

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

Let's now work on ε

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

$$R - R^T = 4\mu[\varepsilon\times]$$

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

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

$\exists$ eigenvector of the matrix $\frac{R-R^T}{2}$ and then pick the one with eigenvalue $=1$ if $\mu=0$
↳ autovalore

Flipping the signs will give you the other
or values?

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

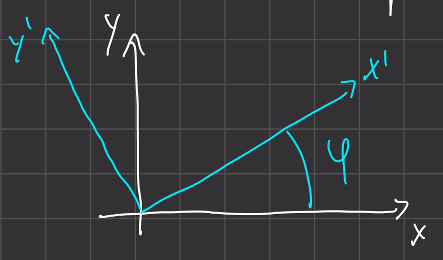
Let's also recall how an elementary rotation is written. So fo example a rotation around the z axis of φ is:

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

$$R_y(\theta) = \begin{bmatrix} \cos\varphi & 0 & \text{sen}\,\varphi \\ 0 & 1 & 0 \\ -\text{sen}\,\varphi & 0 & \cos\varphi \end{bmatrix}$$

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

To find them for example:



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

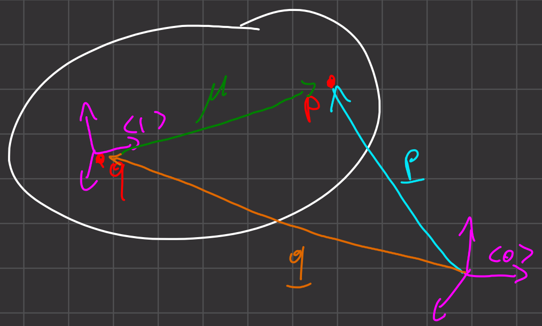
So these are the x' coordinates on the x axis

These the y' on the y

thes z' on the z.

Angular velocity vector

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



$$\underline{p} = \underline{q} + \underline{\mu}$$

If we now start projecting:

$$^0p = {^0q} + {^0\mu} = {^0q} + {^0_1R}\,{^1\mu}$$

Now we want to consider:

$$^0\dot{p} \triangleq \frac{d}{dt}\left({^0_1R}\,{^1p}\right) \neq {^0_1R}\,\frac{d}{dt}\,{^1p}$$

this is the correct one because we are watching from frame <0>
In the second one we are calculating from the frame <1>

This ambiguity will be solved with $^0v_{p/0}$ -> this means seen from 0

Using this notation the previous becomes:

$$^0\dot{p} \triangleq \underbrace{\frac{d}{dt}\left({^0_1R}\,{^1p}\right)}_{= \; ^0\underline{v}_{p/0}} \neq \underbrace{{^0_1R}\,\frac{d}{dt}\,{^1p}}_{= \; ^0v_{p/1}}$$

Let's continue:

$$^0\dot{p} = {^0\dot{q}} + \frac{d}{dt}\left({^0_1R}\,{^1\mu}\right)$$

$$= {^0\dot{q}} + {^0_1\dot{R}}\,{^1\mu} + {^0_1R}\,{^1\dot{\mu}}$$

$$= \underbrace{^0\dot{q}} + \underbrace{{^0_1\dot{R}}\,\underbrace{{^0_1R}^T\,{^0_1R}}_{I_{3\times3}}\,{^1\underline{\mu}}} + \underbrace{{^0_1R}\,{^1\dot{\underline{\mu}}}}$$

this tells how <1> is changing in respect to <0>

this term takes into account that <1> is changing orientation to <0>

how μ changes in respect to <1> then projected on <0> } this is 0 if in the rigid space of frame <1>

q is the position of frame <1> to <0>
μ position from p to <1>

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

$$R R^T = I_{3\times3}$$

So if now I do:

$$\frac{d}{dt}\left({}^0_1R\,{}^0_1R^T\right) = 0 = {}^0_1\dot{R}\,{}^0_1R^T + {}^0_1R\,{}^0_1\dot{R}^T$$

$$\underbrace{{}^0_1\dot{R}\,{}^0_1R^T}_{} = -{}^0_1R\,{}^0_1\dot{R}^T = \underbrace{-\left({}^0_1\dot{R}\,{}^0_1R^T\right)^T}_{}$$

skew-simmetric matrix

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

$$\left({}^0_1\dot{R}\,{}^0_1R^T\right) = {}^0\underline{\omega}_{1/0} \times \underline{\mu}$$

???

Definition of angular velocity vector

$\forall\ R \in SO(3)$ there exists a unique $\omega \in \mathbb{R}^{3\times1}$ such that

$$\dot{R}R^T = [\underline{\omega}\times]$$

Strapdown equation

So if we have:

$${}^0_1\dot{R}\,{}^0_1R^T = [{}^0\underline{\omega}_{1/0}\times]$$

$$\Downarrow$$

$${}^0_1\dot{R} = [{}^0\underline{\omega}_{1/0}\times]\,{}^0_1R$$

$${}^0_1\dot{R} = {}^0_1R\,[{}^1\underline{\omega}_{1/0}\times]$$

inverse = transpose

$$\underbrace{[{}^0\omega_{1/0}] = {}^0_1R[{}^1\omega_{1/0}]\,{}^0_1R^T}\cdot{}^0_1R$$

Transformation of the operator

Example:

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

$\dot{\alpha} \neq 0$

$$\dot{R}_z(\alpha) = \begin{bmatrix} -\mathrm{sen}(\alpha) & -\cos(\alpha) & 0 \\ \cos(\alpha) & -\mathrm{sen}(\alpha) & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

If we directly calculate:

$$\dot{R}_z(\alpha)\, R_z^T(\alpha) = \dot{\alpha} \cdot \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = [\underline{\omega} \times]$$

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

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 axis in respect to wich 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

$${}^1_0\dot{R}\, {}^1_0R^T = [{}^1\underline{\omega}_{0/1} \times] = \text{N.B.}$$

$${}^0_1R\, [{}^1_0\dot{R}\, {}^1_0R^T]\, {}^0_1R^T \triangleq \underline{[{}^0\omega_{0/1}]}$$

$$= {}^0_1R\, {}^1_0\dot{R} (\cancel{{}^1_0R^T\, {}^0_1R^T}) = {}^1_0R^T\, {}^1_0\dot{R} \longrightarrow \quad A^TB = (B^TA)^T$$

$$= ({}^1_0\dot{R}^T\, {}^1_0R)^T = \underline{({}^0_1\dot{R}\, {}^0_1R^T)^T} = -\underline{({}^0_1\dot{R}\, {}^0_1R^T)} =$$

skew simmetric

stepdown

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

N.B.

$$\boxed{{}^0\underline{\omega}_{0/1} = -{}^0\omega_{1/0}}$$

We can conclude

the angular velocity is a relative concept

Let's see the composition:

$${}^0_2R = {}^0_1R\, {}^1_2R$$

$${}^0_2\dot{R}\, {}^0_2R^T = \left[\frac{d}{dt}({}^0_1R\, {}^1_2R)\right]({}^0_1R\, {}^1_2R)^T$$

$$= \left({}^0_1\dot{R}\,{}^1_2R + {}^0_1R\,{}^1_2\dot{R} \right) {}^1_2R^T\,{}^0_1R^T$$

$$= \underbrace{{}^0_1\dot{R}\,{}^0_1R^T}_{\text{skewdown}} + \underbrace{{}^0_1R \left[\underbrace{{}^1_2\dot{R}\,{}^1_2R^T}_{\text{skewdown}} \right] {}^0_1R^T}_{\text{Rotated to be on } \emptyset}$$

this would be:

$${}^0\omega_{1/0} + {}^\emptyset\left[{}^1\omega_{2/1} \right]$$

From this follows:

$$\boxed{\underline{\omega}_{2/0} = \underline{\omega}_{1/0} + \underline{\omega}_{2/1}}$$

Galilean velocity composition

Angular velocity and the rotation vector

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

Base of a 3-dimensional space. => h and $\dot{h}$ are ortogonal because h is unitary in length and can only rotate, not change the length.

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

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

Mapping between ω and the 4-D axis vector

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

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

This works only if $\cos\theta \neq 1$

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

If we know how yaw-pitch, roll derivative behave in relation to ω then we are good.

When you know yaw-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

$$^{a}\omega_{b/a} = {}^{b}L_{b/a}\,\dot{\bar{\varphi}} \qquad \bar{\varphi} = \begin{bmatrix} \psi \\ \theta \\ \phi \end{bmatrix}$$

N.B.
The order is yaw-pitch-roll (z-y-x)

$$^{a}\omega_{b/a} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}\dot{\psi} + R_z(\psi)\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}\dot{\theta} + R_z(\psi)R_y(\theta)\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}\dot{\phi} =$$

$$= \begin{bmatrix} 0 & -s\psi & c\psi c\theta \\ 0 & c\psi & s\psi c\theta \\ 1 & 0 & -s\theta \end{bmatrix}\begin{bmatrix} \dot{\psi} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix}$$

$$\det(L_{b/a}) = -\cos(\theta) \leftarrow$$

This mappings becomes undefined whenever $\theta = \frac{\pi}{2}$ or better I cannot find the inverse meaning that from omega I don't know the derivatives of ψ, θ, ϕ

16/10

Unit quaternions and the angular velocity vector

$$\begin{cases} \dot{\mu} = -\frac{1}{2}\,\underline{\varepsilon}^T\omega \\ \dot{\underline{\varepsilon}} = \frac{1}{2}\left(\mu I_{3x3} - [\underline{\varepsilon}\times]\right)\underline{\omega} \end{cases}$$

Inverse mapping

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

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

$$\mu^2 + \|\varepsilon\|^2 = 1$$

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.

23/10 - KINEMATICS

Revise notation

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

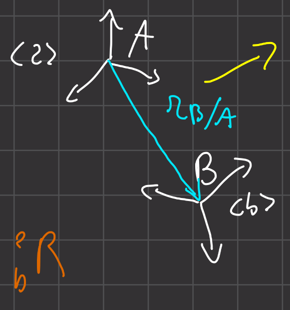
The geometric idea is useful to make models.

Then we have the algebraic representation

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

We are going observe these moving vectors from a frame called observer.

Each one of the body composing a robot is home of one reference frame



translation vector between the two reference frame.

A, B, C are points
<a>, <b> are ref. frame

^a_bR this represent the axis of frame <b> as seen from frame <a>

The axis of frame <a> will be described in two ways:

$$\begin{cases} (\underline{i}_1, \underline{j}_1, \underline{k}_1) \\ (\underline{e}_1^1, \underline{e}_2^1, \underline{e}_3^1) \end{cases}$$

in frame 1

first vector

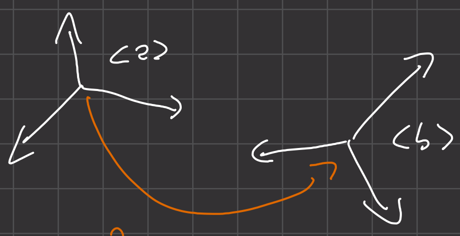
For example in frame B this becomes:

$$\begin{cases} (i_1, j_1, k_1) \\ (e_1^2, e_2^2, e_3^2) \end{cases}$$

So R can be written as:

$$ {}^a_bR = \left[\left(e_i^1 \cdot e_j^a\right)\right] = \begin{bmatrix} \boxed{\begin{matrix} e_1^1 \, e_1^a \\ e_2^1 \, e_1^a \\ e_3^1 \, e_1^a \end{matrix}} & \boxed{\begin{matrix} e_1^1 \, e_2^a \\ * \\ * \end{matrix}} & \boxed{\begin{matrix} e_1^1 \, e_3^a \\ * \\ * \end{matrix}} \end{bmatrix} $$

$$ {}^a e_1^a = {}^a i_a \qquad {}^a e_2^a = {}^a j_a \qquad {}^a e_3^a = {}^a k_a $$



${}^a_bR$ → this is also a way to indicate how to rotate vectors.

We can also define an homogeneous transformation:

$$ {}^a_bT = \left[\begin{array}{c|c} {}^a_bR & {}^a r_{b/a} \\ \hline 0 \quad 0 \quad 0 & 1 \end{array}\right] $$

sometimes this is called H

Let's suppose that we put a vector ${}^b m$.

We might be interested to transform this vector in frame <a>

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

So $ {}^b\bar{m} = \begin{bmatrix} {}^b m \\ 1 \end{bmatrix} $

$$ {}^a_bT \, {}^b\bar{m} \triangleq {}^a\bar{m} = \begin{bmatrix} {}^a_bR \, {}^b m + {}^a r_{b/a} \\ 1 \end{bmatrix} \longrightarrow {}^a m $$

Assumptions:

- All measurements are consistent
- Non consider relativistic effects

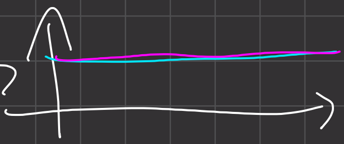
Let's consider a vector that is changing its lenght in time.

$u(t)$

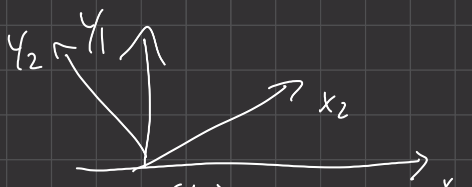
Crucial example

If we consider the frame of the axis the orientation of the vectors is constant so if we map them

<2>

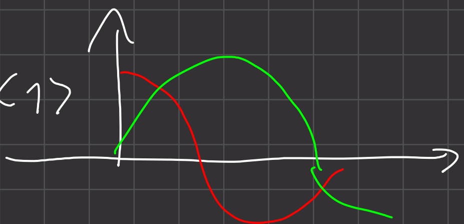


Now if I make another observer:



<1>

<1>



Now if I take the derivates taken from the first frame I get

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

If I take the derivative from frame 1:

$$\frac{d_1}{dt} x_2 = \begin{bmatrix} -\sin(\alpha) \\ \cos(\alpha) \end{bmatrix}$$

the derivatives are not absolute

in frame 2

scalar functions

$$\mu(t) \rightarrow \frac{d_2}{dt} \mu(t) = \frac{d}{dt} \sum_{i=1}^{3} \mu_i(t)\, e_i^2 \quad \{x_1 i_1 + y_1 j_1 + z_1 k_1$$

$$\frac{d_b}{dt} \mu(t)$$

$$= \underbrace{\sum_{i=1}^{3} \left(\frac{d_2}{dt}\, {}^{2}\mu_i(t)\right) e_i^2}_{\text{equivalent}} + \underbrace{\sum_{i=1}^{3} {}^{2}\mu_i(t) \cdot \frac{d_2}{dt} e_i^2}$$

these are the same

common for all the observer.

these are $\not{0}$ because represent how the frame changes with respect with himself.

$$= \frac{d_b}{dt} \sum_{i=1}^{3} {}^{b}\mu_i(t)\, e_i^b = \sum_{i=1}^{3} \left(\frac{d_b}{dt}\, {}^{b}\mu_i(t)\right) e_i^b + \sum_{i=1}^{3} {}^{b}\mu_i(t) \frac{d_b}{dt} e_i^b$$

We can write that:

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

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

$$\frac{d_2}{dt} \mu(t) = \frac{d_2}{dt} \sum_{i=1}^{3} {}^{b}\mu_i(t)\, e_i^b =$$

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

$$= \underbrace{\sum_{i=1}^{3} \left(\frac{d_2}{dt}\, {}^{b}\mu_i(t)\right) e_i^b}_{\frac{d_b}{dt} \mu(t)} + \sum_{i=1}^{3} {}^{b}\mu_i(t) \left(\frac{d_2}{dt} e_i^b\right)$$

So now we can write:

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

Now we know that:

$$ \frac{d_a}{dt}\mu(t) = \frac{d_b}{dt}\mu(t) + \sum {}^b\mu_i(t)\left(\frac{d_a}{dt}\,{}^b e_i\right) $$

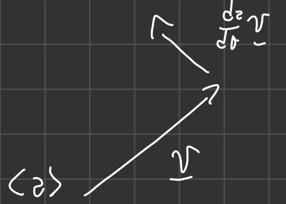
NOTE

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

Let's now compute the time derivative:

$$ \frac{d}{dt}(\underline{v}\cdot\underline{v}) = \emptyset \implies 2\underline{v}\cdot\frac{d}{dt}v = 0 $$

this tells us that the time derivative of a vector must be orthogonal to the vector. the vector can only rotate, not change its length.



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

So we assume that I can find a vector ω that:

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

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

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

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

Now we have a way to describe:

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

$\omega_{b/a} \rightarrow$ the angular velocity of frame b in respect to a.

$$\frac{d_a}{dt}\mu(t) = \frac{d_b}{dt}\mu(t) + \sum \omega_{b/a} \times (\mu_i \cdot e_i^b) =$$

$$= \frac{d}{dt}\mu(t) + \omega_{b/a} \times \underbrace{\sum_{i=1}^{3} (\mu_i \cdot e_i^b)}_{\underline{\mu(t)}}$$

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

the mother of all laws

$$\boxed{\frac{d_a}{dt}\mu(t) = \frac{d_b}{dt}\mu(t) + \omega_{b/a} \times \mu(t)}$$

$\rightarrow$ value of vector

how the observer rotate.

Algebraic representation

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

$${}^a\left[\underbrace{\frac{d}{dt}\mu(t)}\right] = {}^a_bR\ {}^b\left[\underbrace{\frac{d_b}{dt}\mu(t)}\right] + {}^a\left[\underbrace{\omega_{b/a} \times \mu(t)}\right]$$

this is what they are

$$\frac{d}{dt}\begin{bmatrix} {}^{a}\mu_1(t) \\ {}^{a}\mu_2(t) \\ {}^{a}\mu_3(t) \end{bmatrix} \qquad \frac{d}{dt}\begin{bmatrix} {}^{b}\mu_1(t) \\ {}^{b}\mu_2(t) \\ {}^{b}\mu_3(t) \end{bmatrix} \qquad \underbrace{[\omega_{b/a} \times]}_{3\times 3 \text{ skew symmetric}} {}^{a}\mu(t)$$

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

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

this is always a standard derivative because these quantities are scalar

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

Let's now consider an example:

$$\frac{d_a}{dt} e_i^b = \underbrace{\frac{d_b}{dt} e_i^b}_{\text{this is } \emptyset} + \left(\omega_{b/a} \times e_i^b\right) \qquad \forall i = 1,2,3$$

$${}^{a}\left(\frac{d_a}{dt} e_i^b\right) = {}^{a}_{b}\dot{R} = \left[{}^{a}\omega_{b/a} \times\right] {}^{a}_{b}R$$

Properties:

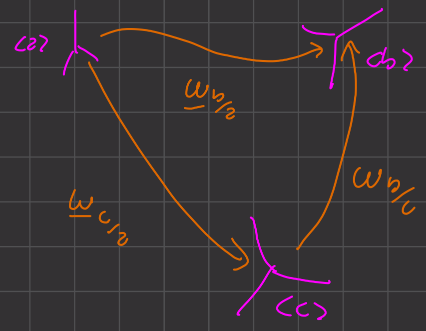
1) ${}^{b}_{a}R\, {}^{a}_{b}R = \mathbb{1}$ (identity) $\rightarrow \frac{d}{dt}\left({}^{b}_{a}R\, {}^{a}_{b}R\right) = [\emptyset] \Rightarrow {}^{b}_{a}\dot{R}\, {}^{a}_{b}R + {}^{b}_{a}R\, {}^{a}_{b}\dot{R} = [\emptyset]$.

same trickery as before

$${}^{b}_{a}\dot{R}\, \cancel{{}^{a}_{b}R} = -{}^{b}_{a}R\, {}^{a}_{b}\dot{R} = -{}^{b}_{a}R\left[{}^{a}\omega_{b/a}\right] \cancel{{}^{a}_{b}R}$$

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

2) Composition of angular velocities



$$\frac{d_a}{dt}\mu = \frac{d_b}{dt}\mu + \omega_{b/a} \times \mu$$

$$\frac{d_c}{dt}\mu = \frac{d_b}{dt}\mu + \omega_{b/c} \times \mu$$

$$\frac{d_a}{dt}\mu = \frac{d_c}{dt}\mu + \omega_{c/a} \times \mu$$

$$\cancel{\frac{d_c}{dt}\mu} + \cancel{\frac{d_a}{dt}}\mu - \cancel{\frac{d_a}{dt}} = \cancel{\frac{d_c}{dt}}\mu + \frac{d_b}{dt}\mu - \frac{d_b}{dt}\mu + \left[\left(\omega_{c/a} + \omega_{b/a}\right) - \omega_{b/a}\right] \times \mu$$

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

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

$$\Downarrow$$

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

Angular velocities are additive quantities.

Very easy corollary:

$$\underset{\emptyset}{\omega_{a/a}} = \omega_{b/a} + \omega_{a/b} \Rightarrow \omega_{a/b} = -\omega_{b/a}$$

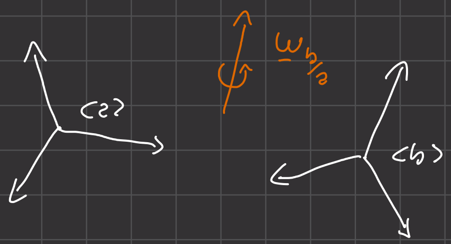
Knowing this we can immediately write the previous equation

$$ {}^b_a\dot{R}\, \cancel{{}^a_b R} = -{}^b_a R\, {}^a_b\dot{R} = -{}^b_a R \left[{}^a\omega_{b/a}\right] \cancel{{}^a_b R} = {}^b_a R \left[{}^a\omega_{a/b}\right]$$

NOTE

$${}^a A = {}^a_b R\, {}^a A\, {}^b_a R$$

30/10



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

$$\frac{d_a}{dt}\mu = \frac{d_b}{dt}\mu(t) + \underbrace{\omega_{b/a} \times \mu(t)}_{\text{this takes into consideration the relative movement}}$$

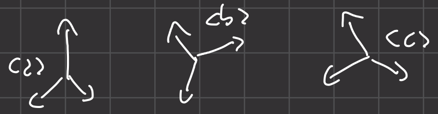
$$\frac{d}{dt}{}^a\mu = {}^a_bR \frac{d}{dt}{}^b\mu + \left[{}^a\underline{\omega}_{b/a} \times\right] \underbrace{{}^a\mu(t)}_{{}^a_bR\,\mu^b(t)}$$

The strapdown equation

integrating?

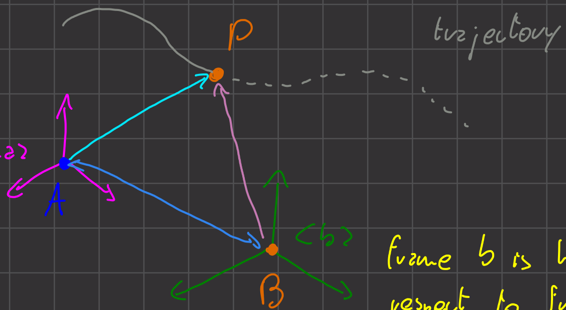
$${}^a_b\dot{R} = \left[{}^a\underline{\omega}_{b/a} \times\right] {}^a_bR$$
$${}^b_a\dot{R} = \left[{}^b\underline{\omega}_{a/b} \times\right] {}^b_aR$$

we have exploited that $\omega_{a/b} = -\omega_{b/a}$



$$\underline{\omega}_{c/a} = \underline{\omega}_{c/b} + \underline{\omega}_{b/a}$$

END RECAP



$$r_{P/A} \triangleq (P - A)$$
$$r_{P/B} \triangleq (P - B)$$
$$r_{B/A} \triangleq (B - A)$$

frame b is both rotating and translating with respect to frame <a>.

$$\underbrace{\underline{r}_{P/A}}_{\text{Position of p with respect to A}} = \underline{r}_{P/B} + \underline{r}_{B/A}$$

We also have:

$${}^{a}_{b}R = \left[e^{a}_{i} \cdot e^{b}_{j} \right]$$

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

$$\frac{d_a}{dt} \underline{r}_{P/A} = \frac{d_a}{dt}\left[\underline{r}_{P/B} + \underline{r}_{B/A}\right] = \frac{d_a}{dt} \underline{r}_{P/B} + \underbrace{\frac{d_a}{dt} \underline{r}_{B/A}}$$

this can be called

$\underline{v}_{B/a}$

↓

velocity of point B computed in frame a.

NOTE a is baseframe

$$\frac{d_a}{dt} \underline{r}_{P/B} = \underbrace{\frac{d_b}{dt} \underline{r}_{P/B}}_{\underline{v}_{P/b}} + \underline{\omega}_{b/a} \times \underline{r}_{P/B}$$

Now we can say that:

$$\frac{d_a}{dt} \underline{r}_{P/A} = \underline{v}_{P/a}$$

So now:

this are due to the effect of the motion of b

$$\underline{v}_{P/a} = \underline{v}_{P/b} + \overbrace{\underline{\omega}_{b/a} \times \underline{r}_{P/B} + \underline{v}_{B/a}}$$

Galileo theorem for velocity distribution

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

$$\underbrace{\frac{d_a}{dt} \underline{v}_{P/a}}_{\underline{a}_{P/a}\ \text{acceleration}} = \frac{d_a}{dt}\left[v_{P/b} + \omega_{b/a} \times r_{P/B} + v_{B/a} \right] =$$

$$= \underbrace{\frac{d_a}{dt} \underline{v}_{P/b}} + \underbrace{\frac{d_a}{dt} \underline{\omega}_{b/a} \times \underline{r}_{P/B}} + \underbrace{\frac{d_a}{dt} \underline{v}_{B/a}}_{\underline{a}_{B/a}}$$

$$= \underbrace{\frac{d_b}{dt} v_{P/b}}_{\underline{a}_{P/b}} + \omega_{b/a} \times v_{P/b}$$

$$\underbrace{\left(\frac{d_a}{dt} \underline{\omega}_{b/a}\right)}_{\text{angular acceleration of } b \text{ with respect to } a} \times \underline{r}_{P/B} + \underline{\omega}_{b/a} \times \overbrace{\left[\underline{v}_{P/b} + \underline{\omega}_{b/a} \times r_{P/B} \right]}^{\frac{d_a}{dt} r_{P/b}}$$

$$a_{P/a} = a_{P/b} + \underbrace{2\,\omega_{b/a} \times v_{P/b}}_{\text{coriolis accel.}} + \left(\frac{d_a}{dt}\,\omega_{b/a}\right) \times r_{P/b} + \underbrace{\omega_{b/a} \times \left[\omega_{b/a} \times r_{P/B}\right]}_{\text{centripetal accel.}} + a_{B/a}$$

(quadratic function)

Now some notes on the previous formula:

Note:

$$\frac{d_a}{dt}\,\omega_{b/a} = \frac{d_b}{dt}\,\omega_{b/a} + \cancel{\omega_{b/a} \times \omega_{b/a}}^{=0} \Rightarrow \frac{d_a}{dt}\,\omega_{b/a} = \dot{\omega}_{b/a}$$

let's now assume this as a generic vector $u(t)$

this allows us to write this

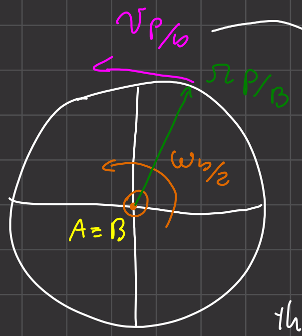
ω is in general the time derivative of NO vector and angular accel is the time derivative of a vector. Now the rate of change of the angular velocity is the same for both observers.

$$a_{P/a} = a_{P/b} + \underbrace{2\,\omega_{b/a} \times v_{P/b}}_{\text{coriolis acc.}} + \dot{\omega}_{b/a} \times r_{P/B} + \underbrace{\omega_{b/a} \times \left(\omega_{b/a} \times r_{P/B}\right)}_{\text{cent. acc.}} + a_{B/a}$$

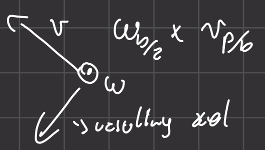
NOTE:

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

This is done by using this quantity.

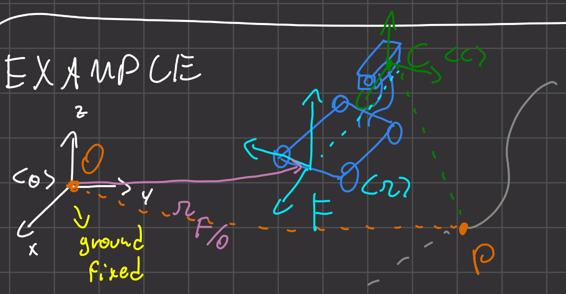


a person moving like this will feel an acceleration given by the coriolis effect



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: $\underbrace{r_{P/O}\ ,\ v_{P/O}\ ,\ a_{P/O}}$

- assume these quantities are known at $t = t_m$ — time of the measurement.
- assume these as "smooth" time function — differentiable and to third derivative continuity

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

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

$$r_{P/O}(t) \simeq \underline{r}_{P/O}(t_m) + \underline{v}_{P/O}(t_m)(t - t_m) + \frac{1}{2} a_{P/O}(t_m)(t - t_m)^2$$

In practice we are going to implement:

$$ {}^{O}\underline{r}_{P/O} \simeq {}^{O}\underline{r}_{P/O} + {}^{O}\underline{v}_{P/O}(t - t_m) + \frac{1}{2}\, {}^{O}a_{P/O}(t - t_m)^2$$

Note - that <r> and F are positioned arbitrary on the robot
- we assume that the camera is calibrated with respect to frame <r>
  (C, <c>) are calibrated w.r.t. (F, <r>)

- assume ${}^{r}r_{C/F}$ known, ${}^{r}_{c}R$ known

① $\underline{r}_{P/O} = \underline{r}_{P/C} + \underline{r}_{C/F} + r_{F/O}$

$${}^{O}_{c}R = \underbrace{{}^{O}_{r}R}_{\text{assumed known}}\ \underbrace{{}^{r}_{c}R}_{\text{known from calibration}}$$

② Computing:

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

$$= \left[\underbrace{\frac{d_c}{dt}\underline{r}_{P/C}}_{\text{velocity of p as estimated by the camera}} + \omega_{c/O} \times \underline{r}_{P/C}\right] + \left[\frac{d_r}{dt}\underline{r}_{C/F} + \omega_{r/O} \times \underline{r}_{C/F}\right] + v_{F/O}$$

$\underbrace{\omega_{c/r} + \omega_{r/O}}$ → this is $\emptyset$ if the camera is not rotating.

$$\underline{v}_{P/O} = \underline{v}_{P/C} + \underline{\omega}_{r/O} \times \left(\underline{r}_{P/C} + \underline{r}_{C/F}\right) + \underbrace{\cancel{\frac{d_r}{dt}\, \underline{r}_{C/F}}}_{} + \underline{v}_{F/O}$$

$=\emptyset$ for us because the camera is fixed

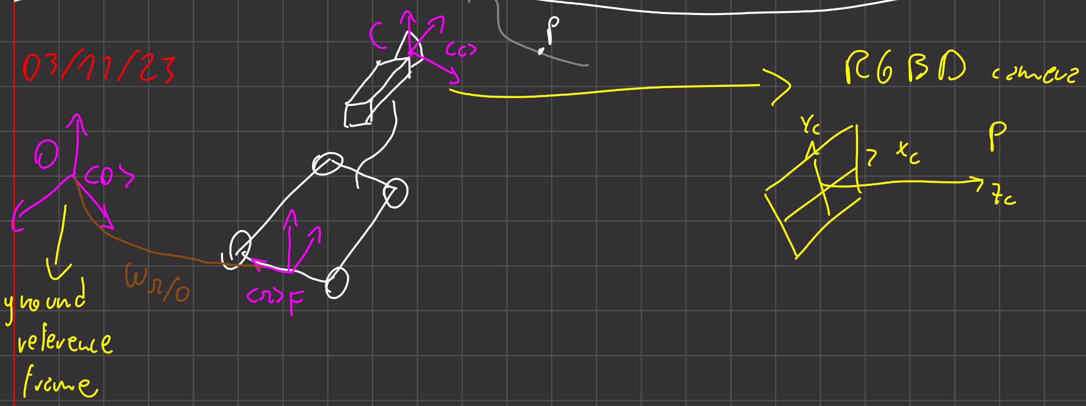
$$ {}^{O}v_{P/O} = {}^{O}_{C}R\, {}^{C}v_{P/C} + \left[{}^{O}\omega_{r/O} \times\right]\left[{}^{O}_{C}R\, {}^{C}r_{P/C} + {}^{O}_{r}R\, {}^{r}\underline{r}_{C/F}\right] + {}^{O}v_{F/O}$$

try doing the calc. by yourself.

Final notes:

$$\frac{d_O}{dt}\, \underline{r}_{F/O} = -\frac{d_O}{dt}\, \underline{r}_{O/F} = \frac{d_r}{dt}\, \underline{r}_{O/r} + \overset{-\omega_{O/r}}{\underline{\omega}_{r/O}} \times \underline{r}_{O/F}$$

03/11/23

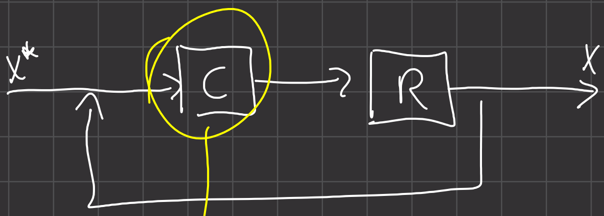


From r to c we have a constant rotation

$$\begin{cases} {}^{r}\underline{r}_{C/F} \\ {}^{r}_{C}R \end{cases}$$ these are constants

→ calibration parameters

NOTE




```
for(;;){
  <read sensors>
  <compute errors>
  <compute control action>
  <send control to motors>
  <suspend> until the next sampling period starts
}
```

idea of digital control

We want to compute $\begin{cases} \underline{r}_{P/O}(t_m) \\ \underline{v}_{P/O}(t_m) \\ \underline{a}_{P/O}(t_m) \end{cases}$

$$\underline{r}_{P/O}(t) = \underline{r}_{P/O}(t_m) + \underline{v}_{P/O}(t_m)(t-t_m) + \frac{1}{2}\underline{a}_{P/O}(t_m)(t-t_m)^2$$

$t \geq t_m$

taylor expansion?

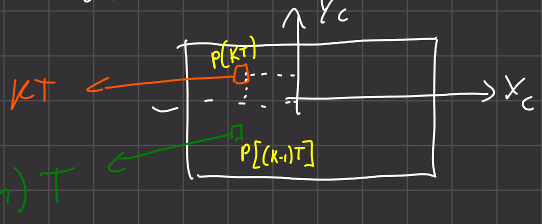
$$\underline{r}_{P/O} = \underline{r}_{P/C} + \underline{r}_{C/F} + \underline{r}_{F/O}$$

$$\frac{d_O}{dt}\underline{r}_{P/O} \triangleq \underline{v}_{P/O} = \underbrace{\frac{d_C}{dt}\underline{r}_{P/C}}_{\underline{v}_{P/C}} + \underbrace{\underline{\omega}_{C/O}}_{\cancel{\omega_{C/R}} + \omega_{R/O}} \times \underline{r}_{P/C} + \underbrace{\frac{d_R}{dt}\underline{r}_{C/F}}_{=\emptyset} + \underline{\omega}_{R/O} \times \underbrace{\underline{r}_{C/F}}_{\text{given}} + \underline{v}_{F/O}$$

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

$\omega_{C/R} = \emptyset$, $\omega_{R/O}$ ↓ given

Assume that at KT



$${}^{C}\underline{v}_{P/C} \simeq \frac{1}{T}\left\{ {}^{C}P(KT) - {}^{C}P[(K-1)T] \right\}$$ approximation of the velocity

↳ we see that $v_{P/C}$ is given

$$\underline{v}_{P/O} = \underline{v}_{P/C} + \underline{\omega}_{R/O} \times \left[\underline{r}_{C/F} + \underline{r}_{P/C}\right] + \underline{v}_{F/O}$$

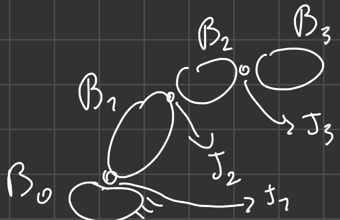
↳ this is given and measured with inertial sensors.

↳ usually ${}^{R}\omega_{R/O}$

$$\underline{\frac{d_O}{dt} v_{P/O}} \triangleq a_{P/O} = \underbrace{\frac{d_C}{dt} v_{P/C}}_{{}^C a_{P/C}} + \underbrace{\cancel{\omega_{C/O}}}_{\underbrace{\omega_{r/O}}_{{}^r\omega_{r/O}}} \times \underbrace{v_{P/C}}_{{}^C v_{P/C}} + \overset{\text{N.B.}}{\omega_{r/O}} \times \underbrace{r_{P/F}}_{\underbrace{{}^C r_{P/F}}_{{}^C r_{P/C},\ {}^r r_{C/F}}} + \omega_{r/O} \times \left[\frac{d_r}{dt} r_{P/F} + \omega_{r/O} \times r_{P/F}\right] + a_{F/O}$$

Kinematic Chains (Holonomic Robots)

A Kinematic chain is a connection of rigid bodies.



$B \to$ body

$J \to$ joint

Connected by mechanism that allow movement according to some laws.

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

We will assume that there will be the ground body.

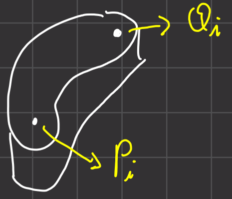
We are modelling in the forward direction $\begin{cases} \text{forward geometry} \\ \text{forward kinematics} \end{cases}$

Assumptions:

B_i is rigid body (a rigid body is an entity where the relative distance of two points taken on the object do not change)

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

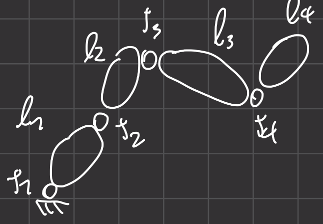
- rotational joints (about 1 axis) so 1-D rotational joints.
  We can associate a variable q_i to the joint to tell the rotation
  $q_i \to$ joint angle

A 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 reduce the movement of each body to which they are connected.

We have:
- rotational joint (1D)
- prismatic joint (1D)

For each joint we have:

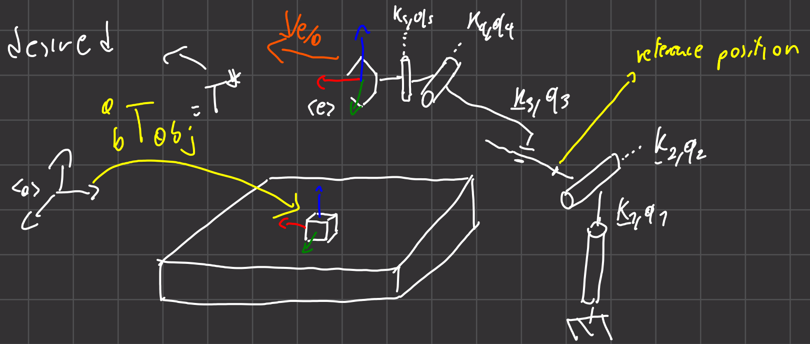
$$(q_j, k_j)$$

q_j: joint variable

k_j: angle or linear displacement increment.

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

Then there are the inverse problems:



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

$^{o}_{e}T$ is what is calculated by the forward kinematic problem.

$$ {}^{0}_{e}T = \begin{bmatrix} {}^{0}_{e}R & {}^{0}r_{e/0} \\ 0 & 1 \end{bmatrix} = {}^{0}_{e}T\underbrace{(\overbrace{q_1, q_2, q_3, q_4, q_5}^{\text{d.o.f of the kinematic chain}})} $$

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

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

joint (variable) vector

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

$\Downarrow$

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:

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

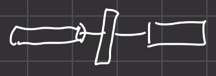
We want to compute q such that is valid. (Inverse geometric problem)

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.

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



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

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

$$ J_e = J_e(\underline{q}) $$

jacobian at point e

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

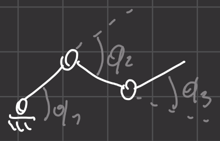
When you talk about q you talk about geometry, $q, \dot{q} \rightarrow$ velocity, $q, \dot{q}, \ddot{q} \rightarrow$ dynamics.

Types of Kinematic chains

In this course we will talk about a serial, open kinematic chain.

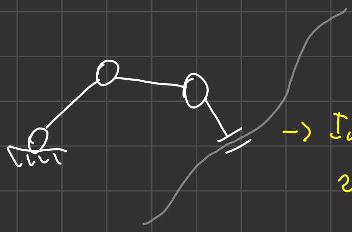
one after the other

you can freely move anyone of the body specifying one of the variables.



n d.o.f.

Serial closed loop K.C.



→ In this case the e.e. is constrained to move along a trajectory.

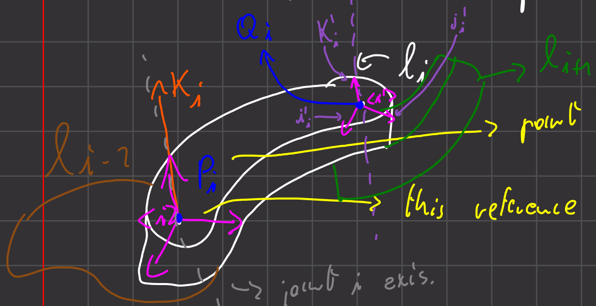
In this case the robot has m d.o.f $< n$ number of joints because it has to follow a particular trajectory.

Branched or tree structured kinematic chain

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

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

Each link has a father apart from the ground base.



→ point on the movement of the joint.

→ this reference frame is fixed in body l_i

→ joint i axis.

l_0 ground

l_i has exactly one father denoted as l_{i-1}

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

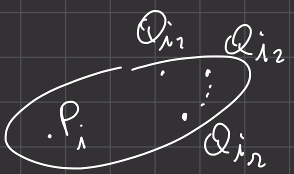
${}^{i}\Omega_{Q_i/P_i}$ is constant. this are three numbers. Position of the attachment of the next body with respect to frame <i>

${}^{i}_{i'}R$ is constant

We need the z axis of <i> <i'> to be on the direction of the movement of the joints.

$$T_i = \left[\begin{array}{c|c} {}^{i}_{i'}R & {}^{i}\Omega_{Q_i/P_i} \\ \hline 0^T & 1 \end{array}\right] \rightarrow$$ this is expressed in frame <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_i' and j_i'

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 $q_{i+1} = 0$

K_{i+1} represents the z-axis of the frame $i+1$.
K_{i+1} must be aligned with the axis of J' so: $K_{i+1} \equiv K_i'$

this means $\langle i+1 \rangle \equiv \langle i' \rangle$ if the joint $i+1$ is prismatic.
If the joint is rotational the condition is the same if $q_{i+1} = \emptyset$

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

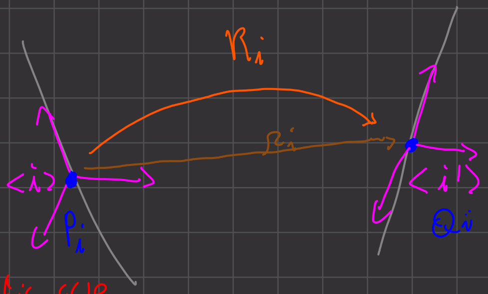
Note that i and j on <i'> should be placed in the same position of <i+1> when $q_{i+1} = 0$

Let's define a symbolic vector Γ

$$\Gamma = [\underbrace{\quad ⊛ \quad}_{N \text{ elements}}] \qquad ⊛ \begin{cases} R - \text{rotational} \\ P - \text{prismatic} \end{cases}$$

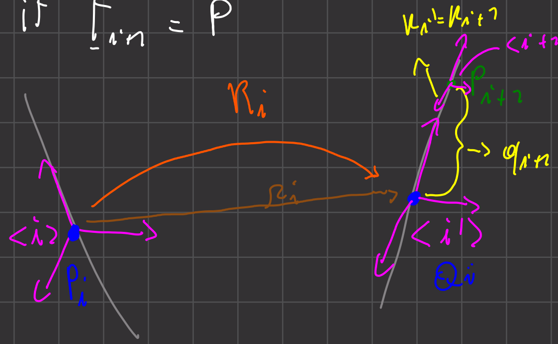
$$\Gamma_i = \begin{cases} R \\ P \end{cases} \qquad \Gamma = [R, P, R, P, R, R]$$

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



prismatic case

if $\Gamma_{i+1} = P$



$$\boxed{\begin{aligned} r_{i+1/i} &\equiv r_{P_{i+1}/P_i} = r_i + K_{i+1} q_{i+1} \\ {}^{i} r_{i+1/i} &= {}^{i}\underline{r}_i + \underbrace{{}^{i}K_{i+1}}_{{}^{i}\underline{K}'_i \ \sim \text{ constant}} q_{i+1} \end{aligned}}$$

note

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

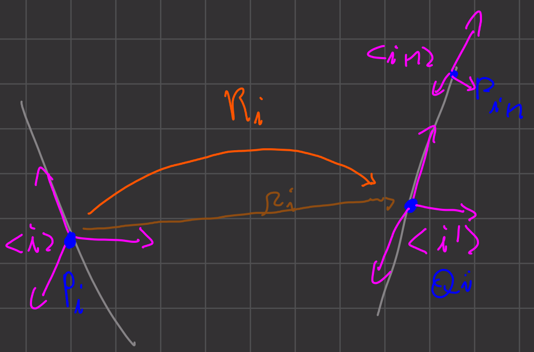
this is due the choice of common z is effective

Now ${}^{i}K'_i = \underbrace{R_i \, {}^{i'}K'_i}_{\text{this is just the third column of the matrix}}$ is a constant

$${}^{i}_{i+1}R = R_i$$

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

Note P_{i+1} and Q_i are coincident, but have are drawn separated to not make a mess.



$$ {}^{i}_{i+1}r = r_i $$

$$ {}^{i}_{i+1}R = R_i \, R_z(q_{i+1}) $$

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

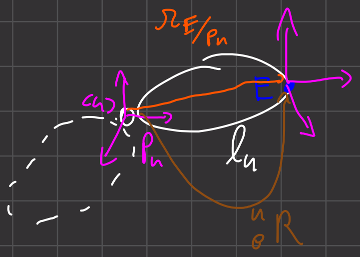
Forward geometry

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

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

Distal link: farthest body in the chain

Usually to the last body ℓ_n an "auxiliary" body called e.e. 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

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

Point of ground reference frame

where my robot is attached

What is the position of <n> with respect to <0>

$$r_{n/0} = r_{1/0} + r_{2/1} + r_{3/2} + \dots + r_{n/n-1}$$

$${}^{0}_{n}R = {}^{0}_{1}R\,{}^{1}_{2}R\,{}^{2}_{3}R \cdots {}^{n-1}_{n}R$$

example

→ in general is function of q_1

$${}^{0}_{1}T = \begin{bmatrix} {}^{0}_{1}R & {}^{0}r_{1/0} \\ 0 & 1 \end{bmatrix}$$ → this depends on the type of joint.

either one or the other changes with q.

generalization

$${}^{i-1}_{i}T = \begin{bmatrix} {}^{i-1}_{i}R(q_i) & {}^{i-1}r_{i/i-1} \\ 0_{1\times 3} & 1 \end{bmatrix} \Rightarrow \boxed{{}^{0}_{n}T = \prod_{i=1}^{n} {}^{i-1}_{i}T(q_i)}$$

10/11/23

$R_i\ K_{i-1}$

RECAP

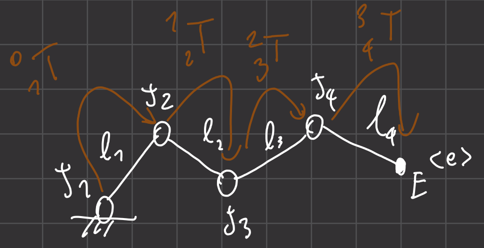
$$r_{i/i-1} = \begin{cases} \underline{r}_{i-1/i-1} & \Gamma_i = R \\ \underline{r}_{i-1/i-1} + k_i q_i & \Gamma_i = P \end{cases}$$

constant in <i-1>

${}^{i-1}_{i}R(q=0)$

$$R_\emptyset \qquad {}^{\emptyset}_{1}R$$

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



$$ {}^{i-1}_{i}T = \left[\begin{array}{c|c} {}^{i-1}_{i}R & {}^{i-1}\underline{\Omega}_{i/i-1} \\ \hline \emptyset_{1\times 3} & 1 \end{array}\right] = {}^{i-1}_{i}T(q_i) $$

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

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

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

Note that:
$$ {}^{i}\underline{K}_i = {}^{i}K_{i-1} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad \text{by construction.} $$

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

$$ {}^{0}_{n}T = {}^{0}_{1}T\,{}^{1}_{2}T \cdots {}^{n-2}_{n-1}T\,{}^{n-1}_{n}T = {}^{0}_{n}T(q) $$

We can think to characterize the last link with:

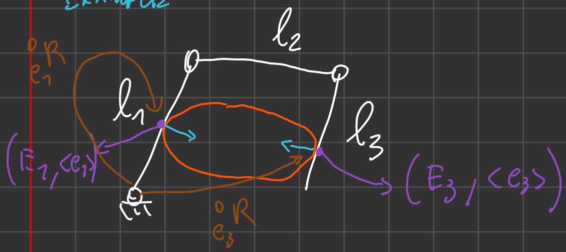
$$ \begin{cases} {}^{n}\underline{\Omega}_{E/n} \\ {}^{n}_{e}R \end{cases} \Rightarrow {}^{n}_{e}T \left[\begin{array}{c|c} {}^{n}_{e}R & {}^{n}\underline{\Omega}_{E/n} \\ \hline 0^T & 1 \end{array}\right] = {}^{0}_{e}T = {}^{0}_{n}T \cdot {}^{n}_{e}T $$

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

Remark

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:

$$\left({}^{1}\underline{r}_{E_1/1},\ {}^{1}_{e_1}R\right),\quad \left({}^{3}\underline{r}_{E_3/3},\ {}^{3}_{e_3}R\right)$$

I want to compute ${}^{\phi}\underline{r}_{E_3/E_1} = {}^{\phi}\underline{r}_{E_3/\phi} - {}^{\phi}\underline{r}_{E_1/\phi} =$) Homogeneous so 2x4

$$= \underbrace{{}^{0}_{3}T \cdot {}^{3}\underline{r}_{E_3/3}}_{\text{this is the position of } E_3 \text{ with respect to the origin}} - {}^{0}_{1}T\,{}^{1}\underline{r}_{E_1/1}$$

-> this is for the transformation

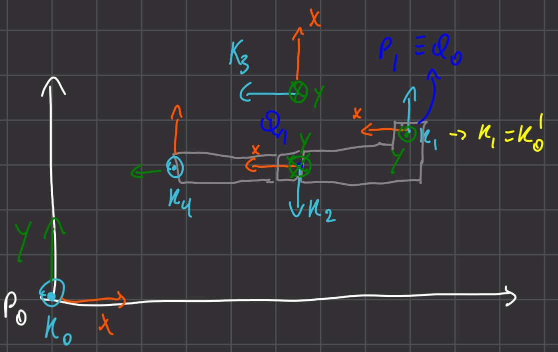
Now to compute the Rotations:

$${}^{0}_{e_3}R = {}^{0}_{e_1}R\,{}^{e_1}_{e_3}R \Rightarrow {}^{e_1}_{e_3}R = {}^{\phi}_{e_1}R^{T}\,{}^{\phi}_{e_3}R$$

$${}^{\phi}_{e_1}R^{T} = \left({}^{\phi}_{1}R \cdot {}^{1}_{e_1}R\right)^{T} \qquad {}^{\phi}_{3}R\,{}^{3}_{e_3}R$$

EXAMPLE

$\underbrace{(P_i, Q_i)}_{{}^{i}\underline{r}_{i'/i}}$ can be set arbitrarly on the body.



$${}^{\phi}\underline{r}_{\phi'/\phi} = \begin{pmatrix} 20 \\ 10 \\ 6 \end{pmatrix} \qquad {}^{\phi}_{\phi'}R = R_0 = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

ROTATIONAL JOINT

$$^{\phi}\underline{\Omega}_{1/0} = {}^{\phi}\underline{\Omega}_{\phi'/\phi}$$

this two vectors describe the rotation of x and y wrt <ϕ>

$$^{\phi}_{1}R = {}^{\phi}_{1'}R \; R_z(q_1) = \begin{pmatrix} * & * & 0 \\ * & * & 1 \\ * & * & 0 \end{pmatrix}$$

Let's now define the model for link 1

(P_1, Q_1)

$$^{1}\underline{\Omega}_{1'/1} = \begin{pmatrix} 2 \\ 0 \\ -1 \end{pmatrix} \qquad {}^{1}_{1'}R = R_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix}$$

$$\begin{cases} {}^{1}\Omega_{2/1} = {}^{1}\Omega_{1'/1} \\ {}^{1}_{2}R = R_1 \, R_z(q_2) \end{cases}$$

Now for link 2 we have the rotating forearm:

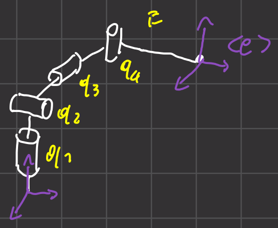
$$(P_2, Q_2) \qquad {}^{2}\underline{\Omega}_{2'/2} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \qquad {}^{2}_{2'}R = R_2 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix}$$

Now we work on the ee

(P_3, E) Derive $^{3}_{e}R$

$$^{3}\Omega_{E/3} = \begin{pmatrix} 0 \\ 0 \\ 2 \end{pmatrix}$$

20/11/23



$$^{0}_{E}T = {}^{0}_{E}T(\underline{q}) \qquad \underline{q} = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_N \end{bmatrix}$$

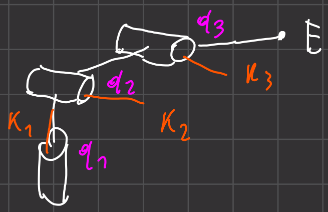
$$ {}^0_eT = {}^0_1T(q_1)\,{}^1_2T(q_2)\,{}^2_3T(q_3)\cdots{}^{n-1}_{\;\;n}T(q_n)\,{}^n_eT $$

I general this formula is not known explicitly.

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

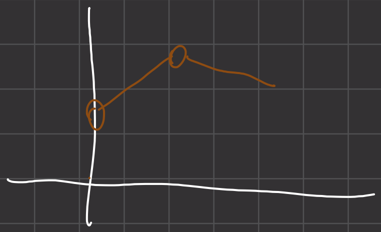
The inverse is when we know the position of the EE. and we get the positions of all the joints.

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



K_2 and K_3 are always parallel.

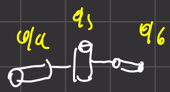
For any value of q_1 you can identify a plane like this:



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

Now we need 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 joint is larger than the number of dof expected of the EE.

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

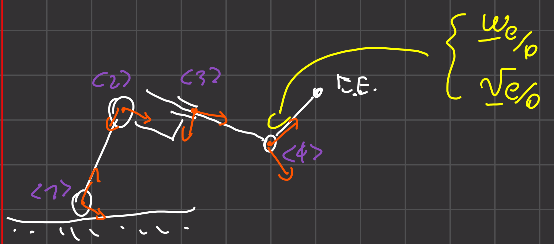
$$ q = \begin{bmatrix} q_1 \\ \vdots \\ q_N \end{bmatrix} \in \mathbb{R}^N \qquad N \geq 6 $$

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 e.e. will have.

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

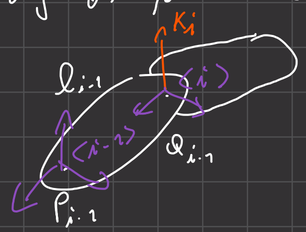
We have a robot mechanism:



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

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

In the modelling of joints we defined:



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

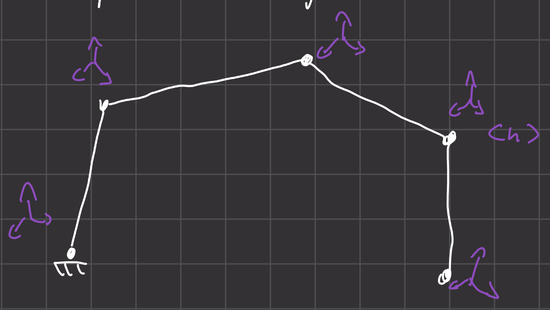
$$\underline{v}_{i/i-1} \triangleq \frac{d_{i-1}}{dt}(P_i - P_{i-1}) = \begin{cases} \underline{\emptyset} & \Gamma_i = R \\ \underline{K}_i \dot{q}_i & \Gamma_i = P \end{cases}$$

$\dot{q}_i$ scalar (the derivative of scalar is the same for every possible observer)

$$\underline{\omega}_{i/i-1} \triangleq = \begin{cases} \underline{K}_i \dot{q}_i & \Gamma_i = R \\ \underline{\emptyset} & \Gamma_i = P \end{cases}$$

$\dot{q}_i$: rate of change of angle q_i

The outputs of our algorithms are



$$\begin{cases} {}^{\phi}\underline{\omega}_{n/\phi} = {}^{\phi}J_n^A(\underline{q}) \\ {}^{\phi}v_{n/\phi} = \end{cases}$$

Quantities that I want to compute

$$\begin{cases} \underline{\omega}_{n/0} = \underline{\omega}_{n/n-1} + \underline{\omega}_{n-1/n-2} + \dots + \underline{\omega}_{2/1} + \underline{\omega}_{1/0} \\ v_{n/0} : \end{cases}$$

N.B

$\omega_{n/0}$ can be expressed as the sum of all the velocities of the r.f. that we encounter by going towards the ee.

$$\underline{\omega}_{n/0} = \underbrace{\underline{\omega}_{1/0}}_{J_1^A \dot{q}_1} + \underbrace{\omega_{2/1}}_{J_2^A \dot{q}_2} + \dots + \underbrace{\omega_{n/n-1}}_{J_n^A \dot{q}_n} = \sum_{i=1}^{N} J_i^A \dot{q}_i$$

linear combination of vectors where the coefficients are the vectors.

This is a [illegible] matrix and so can be expressed as a matrix by a vector.

(A stands for angular)

this quantity could be k_2 or $\underline{0}$

It's very important that they are linear in (for ex. in this case) $\dot{q}_2$

$$J_i^A = \begin{cases} k_i & \Gamma_i = R \\ \underline{0} & \Gamma_i = P \end{cases}$$

We can now decide to calculate:

$${}^{\phi}\omega_{n/\phi} = \sum_{i=1}^{N} \underbrace{{}^{\phi}J_i^A}_{\text{3D vector}} \dot{q}_i = \underbrace{\left[{}^{\phi}J_1^A, {}^{\phi}J_2^A, \dots, {}^{\phi}J_n^A \right]}_{{}^{\phi}J_n^A} \underbrace{\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_n \end{bmatrix}}_{\dot{\underline{q}}}$$

this is a matrix vector product.

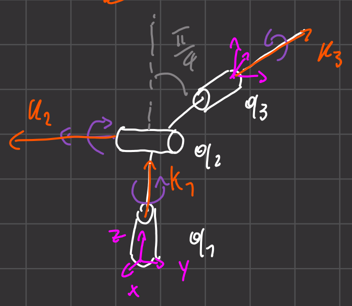
ANGULAR JACOBIAN MATRIX

$${}^{\phi}J_n^A \in \mathbb{R}^{3\times N}$$

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

EXAMPLE

this is a roll-pitch-roll sequence.



$q_1 = 0$

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

$q_3 = 0$

$$J_i^A = \begin{cases} K_i & \Gamma_i = R \\ 0 & \Gamma_i = P \end{cases}$$

$$\underline{\omega}_{3/0} = \omega_{1/0} + \omega_{2/1} + \omega_{3/2} = J_1^A \dot{q}_1 + J_2^A \dot{q}_2 + J_3^A \dot{q}_3 =$$

$$= K_1 \dot{q}_1 + K_2 \dot{q}_2 + K_3 \dot{q}_3$$

$$^0\omega_{3/0} = \left[{}^0\underline{K}_1 , {}^0K_2 ; {}^0K_3 \right]$$

↳ this is strongly affected by q_2

Let's assume that K_2 with $q_2 = 0$ is aligned with i_1 and i_0

$$^0K_2 \equiv {}^0i_1 \equiv {}^0i_0$$

In practice we have:

$$^0\omega_{3/0} = \left[{}^0\underline{K}_1 , {}^0K_2 ; {}^0K_3 \right] = \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & * \\ 1 & 0 & * \end{bmatrix}}_{^0J_3^A} \underline{\dot{q}}$$

↳ these depends on q_2

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

If $q_2 = \{0, \pi\}$ K_3 becomes dependent with K_1.

K_1 becomes parallel to K_3

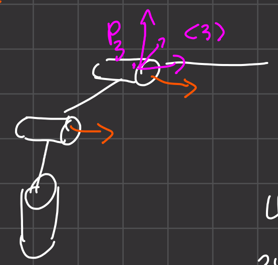
N.B

this is a singularity. the rank of the matrix degenerates from 3 to 2.

this means that there are rotation ∇_0 (orthogonal to the axis) of the joints that cannot be generated in this position.

Sometimes we need the jacobian to be full rank.

Exemple



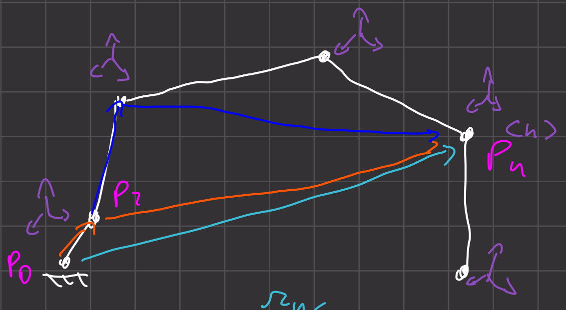
$$^{\phi}J^{A}_{3/0} = \left[\,^{\phi}\underline{k}_1 ,\ ^{\phi}\underline{k}_2 ,\ ^{\phi}\underline{k}_3\right]$$

value is computed w.r.t. B

Whatever is the current values of these vectors k_2 and k_3 are always parallel so $\text{rank}\left(^{\phi}J^{*}_{3/0}\right) = 2$

$$^{A}J_{n/0} \in \mathbb{R}^{3\times n}$$

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



$$v_{n/\phi} = \frac{d_\phi}{dt} r_{n/0} = \frac{d_0}{dt}\left[r_{n/1} + r_{1/0}\right] =$$

$$= \underbrace{\frac{d_0}{dt} r_{1/0}}_{\substack{\text{this we know} \\ v_{1/0}}} + \frac{d_0}{dt} r_{n/1} = \underbrace{\frac{d_0}{dt} r_{1/0}}_{v_{1/0}} + \frac{d_1}{dt} r_{n/1} + \underline{\omega}_{1/0} \times \underline{r}_{n/1} =$$

given by the joint model

given by the forward geometry

Functions of $\dot{q}_1$

$$= \left(\underline{v}_{1/0} + \underline{\omega}_{1/0} \times \underline{r}_{n/1}\right) + \frac{d_1}{dt} \underline{r}_{n/1} =$$

And now the same step is repeated

$$= \left(\underline{v}_{1/0} + \underline{\omega}_{1/0} \times \underline{r}_{n/1}\right) + \frac{d_1}{dt}\left[r_{2/1} + r_{n/2}\right] =$$

$$= \left(\underline{v}_{1/0} + \underline{\omega}_{1/0} \times \underline{r}_{n/1}\right) + \underline{v}_{2/1} + \frac{d_2}{dt} \underline{r}_{n/2} + \underline{\omega}_{2/1} \times \underline{r}_{n/2} =$$

$$= \left(\underline{v}_{1/0} + \underline{\omega}_{1/0} \times \underline{r}_{n/1}\right) + \left(v_{2/1} + \omega_{2/1} \times r_{n/2}\right) + \frac{d_2}{dt} r_{n/2} =$$

skipping to the last step

$$= \left(\underline{v}_{1/0} + \underline{\omega}_{1/0} \times \underline{r}_{n/1}\right) + \left(\underline{v}_{2/1} + \underline{\omega}_{2/1} \times \underline{r}_{n/2}\right) + \dots + \left(\underline{v}_{n/n-1} + \underline{\omega}_{n/n-1} \times \overset{\substack{\text{this is } \phi \\ \uparrow}}{r_{n/n}}\right) =$$

$$= \underbrace{\sum_{i=1}^{n} v_{i/i-1} + \omega_{i/i-1} \times r_{n/i}}_{\text{this is a linear function of } \dot{q}_i} \triangleq$$

$$\triangleq \sum_{i=1}^{n} \underline{J}^L_{n/i}\, \dot{q}_i$$

$$\boxed{\underline{J}^L_{n/i} = \begin{cases} \underline{K}_i & \Gamma_i = P \\ \underline{K}_i \times \underline{r}_{n/i} & \Gamma_i = R \end{cases}}$$

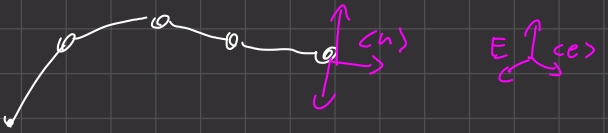
$${}^\phi v_{n/0} = \underbrace{\left[\, {}^\phi J^L_{n/1} \;\vdots\; {}^\phi J^L_{n/2} \;\vdots\; \dots \;\vdots\; {}^\phi J^L_{n/m} \right]}_{3 \times n \text{ matrix}} \dot{q} \quad \to \quad {}^\phi J^L_{n/0} \in \mathbb{R}^{3\times n}$$

$$\boxed{\underbrace{\begin{bmatrix} {}^\phi \underline{\omega}_{n/0} \\ {}^\phi v_{n/0} \end{bmatrix}}_{\mathbb{R}^6} = \begin{bmatrix} {}^\phi J^A_{n/0} \\ {}^\phi J^L_{n/0} \end{bmatrix} \dot{\underline{q}} \triangleq \underbrace{{}^\phi J_{n/0}}_{\mathbb{R}^{6\times n}} \dot{q}} \quad \text{N.B.}$$

If the robot is redundant you have more than 6 dofs.

$${}^\phi J_{n/0} = {}^\phi J_{n/0}(q) \to$$ BASIC ROBOT JACOBIAN

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>

$$\omega_{e/0} = \omega_{n/0}$$

$$\underline{v}_{e/0} = \frac{d_0}{dt}\left[\underline{r}_{n/0} + \underline{r}_{e/n}\right] = \frac{d_0}{dt} r_{n/0} + \frac{d_0}{dt} \underline{r}_{e/n} =$$

$$= v_{n/0} + \underbrace{\cancel{\frac{d_n}{dt} r_{E/n}}}_{0} + \underbrace{\omega_{n/0} \times r_{e/n}}_{\to\, -r_{e/n} \times \omega_{n/0}}$$

$${}^0\omega_{e/0} = \left[\mathbb{1}_3 \,;\, 0_3\right] \begin{bmatrix} {}^0\omega_{n/0} \\ {}^0v_{n/0} \end{bmatrix}$$

Now ${}^0\left(-r_{e/n} \times \omega_{n/0}\right)$ becomes because skew $\left[r_{E/x}\times\right]^T$

$${}^0v_{e/0} = \left[\left[r_{E/N}\right]^T \,\vdots\, \mathbb{1}_3\right] \begin{bmatrix} {}^0\omega_{n/0} \\ {}^0v_{n/0} \end{bmatrix}$$

$$\begin{bmatrix} {}^0\omega_{e/0} \\ {}^0v_{e/0} \end{bmatrix} = {}^0S_{E/N}\; {}^0J_{N/0}\; \dot{\underline{q}}$$

↳ Rigid point jacobian of point E in body N.

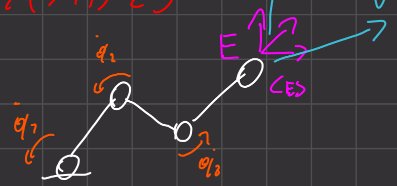
Structure of S:

$$\left[\begin{array}{c|c} \mathbb{1}_3 & 0_{3\times 3} \\ \hline \left[{}^0r_{E/N}\times\right]^T & \mathbb{1}_3 \end{array}\right]$$

this is always invertible.

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

24/11/23



We have found a model that allows to transform a set of joint velocities into $\begin{pmatrix} \underline{\omega} \\ \underline{v} \end{pmatrix}$, 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 ifos. this transformation is linear. Linear is important because it means matrix.

We can say that we are able to compute :

$$\begin{pmatrix} \omega \\ v \end{pmatrix} = \underset{\uparrow}{J}\, \dot{q}$$

jacobian

J is a function of the robot configuration:

$$J = J(q)$$

We will see later how to generate $\dot{q}$ in order to get some desired $\begin{pmatrix} \omega \\ v \end{pmatrix}$.

$${}^{\phi}J_{n/0} \text{ is the basic Jacobian} = \underbrace{\begin{bmatrix} \underbrace{{}^{\phi}J^{A}_{n/0}}_{3\times3} \\ \underbrace{{}^{\phi}J^{L}_{n/\phi}}_{3\times3} \end{bmatrix}}_{6\times3}$$

We can give to the basic Jacobian a general structure:

$$\begin{bmatrix} * & * & * & * & * & * \\ \hdashline * & * & * & * & * & * \end{bmatrix}$$

these will be on ϕ or the axis of rotation in case of rotational joints

This is by construction equal to ϕ

$$\begin{cases} {}^{\phi}K_i & \Gamma_i = P \\ K_i \times r_{n/i} & \Gamma_i = R \end{cases}$$

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 $S_{E/n}$ → the Rigid Body Jacobian

Knowing the two quantities v and ω we can transform these two quantities in the frame of the EE.

$$^{\phi}S_{E/n} = \left[\begin{array}{c|c} \mathbb{1} & \phi \\ \hline [^{\phi}r_{E/n} \times]^{T} & \mathbb{1} \end{array}\right]$$

$$^{\phi}\omega_{e/0} = \cancel{\omega_{e/n}} + {}^{\phi}\omega_{n/0}$$

$$v_{E/\phi} = v_{n/\phi} + \omega_{n/\phi} \times r_{E/N} = v_{n/0} \; {}^{\phi}(-r_{E/n} \times \omega_{n/\phi}) =$$

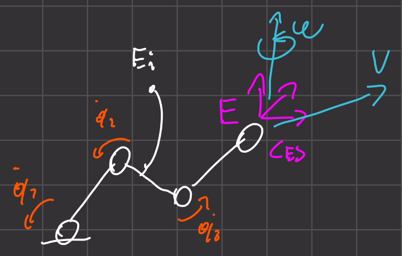
$$= {}^{\phi}[r_{E/N} \times]^{T\phi} \omega_{n/0}$$

$$\begin{bmatrix} ^{\phi}\omega_{E/0} \\ ^{\phi}v_{E/\phi} \end{bmatrix} = \underbrace{^{\phi}S_{E/n}}_{6\times6 \text{ always invertible}} \; \underbrace{^{\phi}J_{n/0}\,\dot{q}}_{\begin{bmatrix} ^{\phi}\omega_{n/0} \\ ^{\phi}v_{n/0} \end{bmatrix}}$$

Inversion of S

$$^{\phi}S_{E/n}' = \left[\begin{array}{c|c} \mathbb{1} & \phi \\ \hline A & \mathbb{1} \end{array}\right] \qquad ^{\phi}S_{E/n}^{-1} = \left[\begin{array}{c|c} \mathbb{1} & \phi \\ \hline A^{T} & \mathbb{1} \end{array}\right]$$

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:

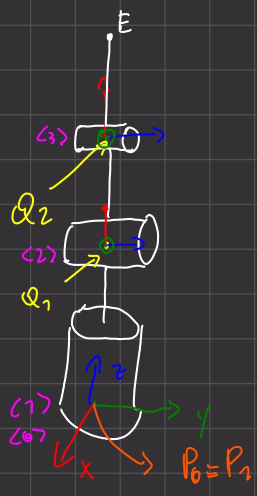
$$ {}^{\phi}J_{i/\phi} = \begin{bmatrix} K_1 & K_2 & K_i & \phi & \phi \\ (K_1 \times r_{i/1}) & (K_2 \times r_{i/2}) & \phi & \phi & \phi \end{bmatrix} $$

the rest of the J will be ϕ.

Now we will just need to add the $S_{Ei/i}$ at the end.

Let's make an example

Example



Link ϕ: ${}^{\phi}r_{Q_0/\phi} = \phi$

Link 1: ${}^{1}r_{Q_1/1} = l_1 \, {}^{1}\underline{K}_1 = a \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \phi \\ \phi \\ l_1 \end{bmatrix}$

Link 2: ${}^{2}r_{Q_2/2} = l_2 \, {}^{2}i_2 = \begin{bmatrix} l_2 \\ \phi \\ \phi \end{bmatrix}$

Link 3: ${}^{3}r_{E/3} = l_3 \, {}^{3}i_3 = \begin{bmatrix} l_3 \\ \phi \\ \phi \end{bmatrix}$

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

Joint 1: ${}^{\phi}r_{1/\phi} = {}^{\phi}r_{Q_\phi/\phi}$ $\qquad {}^{0}_{1}R = R_\phi \, R_z(q_1)$

$R_\phi = \mathbb{1}_3$

Joint 2: ${}^{1}r_{2/1} = {}^{1}r_{Q_1/1}$ $\qquad {}^{1}_{2}R = R_1 \, R_z(q_2)$

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

Joint 3:

$${}^2 r_{3/2} = {}^2 r_{Q_2/2} \qquad {}^2_3 R = R_2 R_z(q_3)$$

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

Now we need to compute the geometry:

$${}^0_3 T = {}^0_1 T \; {}^1_2 T \; {}^2_3 T$$

At home configuration

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

$${}^0 J_{3/0} = \begin{bmatrix} {}^0 k_1 & {}^0 k_2 & {}^0 k_3 \\ {}^0(k_1 \times r_{3/1}) & {}^0(k_2 \times r_{3/2}) & 0 \end{bmatrix}$$

← All of these are geometric quantities

$$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$ this is the span of angular velocity that you can generate

$${}^0_1 T = \left[\begin{array}{c|c} {}^0_1 R & {}^0 r_{1/0} \\ \hline 0_{1\times3} & 1 \end{array}\right]$$

→ k_1 is the third column of this matrix

N.B.

So you calculate ${}^0_1 R$ and then pick the third column

$${}^0_2 T = \left[\begin{array}{c|c} {}^0_2 R & {}^0 r_{2/0} \\ \hline 0_{1\times3} & 1 \end{array}\right]$$

→ k_2 is the third column of this matrix

$${}^0_3 T = \left[\begin{array}{c|c} {}^0_3 R & {}^0 r_{3/0} \\ \hline 0_{1\times3} & 1 \end{array}\right]$$

→ k_3 is the third column of this matrix

$${}^0 r_{3/1} = {}^0 r_{3/0} - {}^0 r_{1/0}$$

$${}^0 r_{3/2} = {}^0 r_{3/0} - {}^0 r_{2/0}$$

Let's see what happens at home configuration:

$$
{}^{\phi}J_{3/\phi}\Big|_{q=\phi} = \left[\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ \hline 0 & l_2 & 0 \\ 0 & \phi & 0 \\ 0 & \phi & 0 \end{array}\right]
$$

cross product of perpendicular vectors

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

To calculate the velocity of the ee. we have to calculate the rigid body jacobian S:

$$
{}^{\phi}S_{e/3} = \left[\begin{array}{c|c} \mathbb{1} & \phi \\ \hline \left[{}^{\phi}r_{E/3}\times\right]^{T} & \mathbb{1} \end{array}\right]
$$

$$
{}^{3}r_{E/3} = l_3\,{}^{3}i_3 \rightarrow {}^{3}\bar{r}_{E/3} = \begin{bmatrix} l_3 \\ 0 \\ 0 \\ \phi \end{bmatrix}
$$

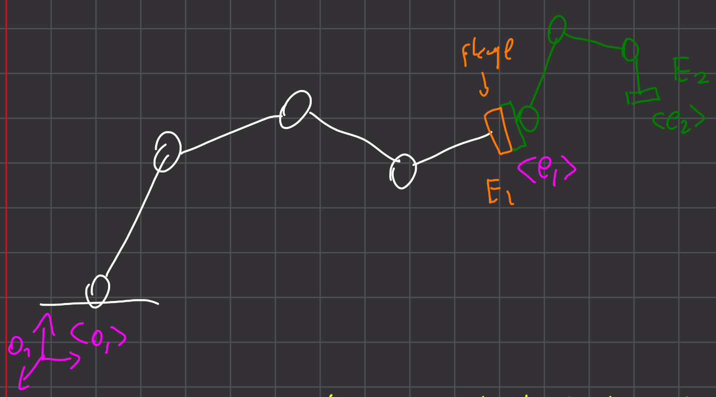
$\rightarrow$ To not have the $+{}^{\phi}r_{3/\phi}$ part

$$
{}^{\phi}\bar{r}_{E/\phi} = {}^{\phi}_{3}T\,{}^{3}\tilde{r}_{E/3} \cancel{+ {}^{\phi}r_{3/\phi}}
$$

$\rightarrow$ It's not homogeneous

01/12/23

More on Jacobians



$$
{}^{O_1}_{E_1}T(q_1) \qquad {}^{O_1}J_{E_1/O_1}(q_1)
$$

these are the value for the first robot

We may think to have also values for the second robot

$$
\begin{cases} {}^{O_2}_{E_2}T(q_2) \\ {}^{O_2}J_{E_2}\,{}_{E_2/O_2}(q_2) \end{cases}
$$

I assume without loss in generality that

$$
\begin{cases} E_1 \equiv O_2 \\ \langle e_1 \rangle \equiv \langle O_2 \rangle \\ {}^{e_1}_{O_2}T = \mathbb{1}_4 \end{cases}
$$

We can write that
$$\begin{cases} \underline{\omega}_{e_2/0} = \underline{\omega}_{e_1/0} + \underline{\omega}_{e_2/e_1} \\ \underline{v}_{e_2/0} = \underline{v}_{e_1/0} + \underline{\omega}_{e_1/0} \times \underline{r}_{E_2/E_1} + \underline{v}_{e_2/e_1} \end{cases}$$

N.B.

Why not 2nd rot Transl

$${}^{0}\omega_{e_2/\phi} = {}^{0}J^{A}_{e_1/\phi}\,\dot{q}_1 + {}^{0}_{0_2}R\;{}^{0_2}J^{A}_{e_2/0_2}\,\dot{q}_2 = \begin{bmatrix} {}^{0}J^{A}_{e_1/0} & \vdots & {}^{0}_{e_1}R\;{}^{0_2}J^{A}_{e_2/0_2} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix}$$

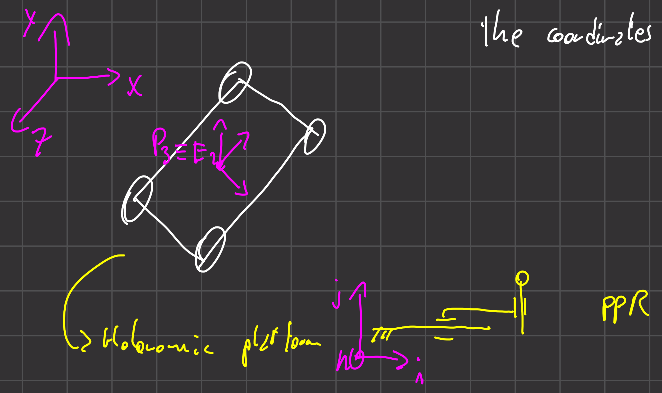
$$v_{E_2/0} = {}^{0}J^{L}_{e_1/0}\,\dot{q}_1 + [r_{E_2/E_1} \times]^{T}\,{}^{0}J^{A}_{E_1/0} + {}^{0}_{e_1}R\;{}^{0_2}J^{L}_{e_2/0_2}\,\dot{q}_2 =$$

$$= \underbrace{\left([{}^{0}r_{e_2/e_2} \times]^{T} \;\vdots\; \mathbb{1}_3 \right)}_{\text{Refer to the first Robot}} {}^{\phi}J_{e_1/0_1}\,\dot{q}_1 + \underbrace{{}^{0}_{e_1}R\;{}^{0_2}J^{L}_{E_2/0_2}\,\dot{q}_2}_{\text{Refer to the second robot}}$$

$$\begin{bmatrix} {}^{0}\underline{\omega}_{e_2/0} \\ {}^{0}\underline{v}_{e_2/0} \end{bmatrix} = \begin{bmatrix} {}^{\phi}S_{E_2/E_1}\;{}^{0}J_{E_1/0} & \vdots & {}^{0}J_{E_2/0_2} \end{bmatrix} \dot{q} \qquad \dot{q} \rightarrow \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix}$$

SERIAL COMPOSITION OF ROBOTS

EXAMPLE



The coordinates are expressed by:
$$\begin{cases} q_1 = x \\ q_2 = y \\ q_3 = \theta \end{cases} \qquad \begin{cases} \dot{q}_1 = \dot{x} \\ \dot{q}_2 = \dot{y} \\ \dot{q}_3 = \dot{\theta} \end{cases}$$

We can think of building a Jacobian for this device

$$\Gamma = [P\,P\,R]$$

In this case the Jacobian is trivial. We assume that $[E_1 \equiv P_3]$

$${}^{\emptyset}J_{E_1/\emptyset} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ \hline 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad {}^{0}J_{E_1/0} = \begin{bmatrix} \emptyset & \emptyset & K_\emptyset \\ \hline {}^{\emptyset}K_1 & {}^{\emptyset}K_2 & \emptyset \end{bmatrix}$$

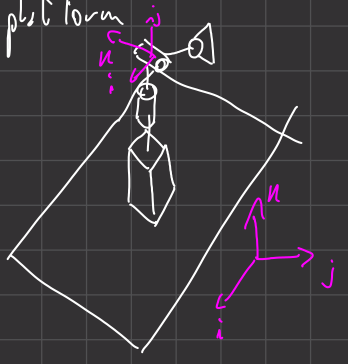
because of $E_1 \equiv P_3$

$R_0 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$

direction of motion always on z

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

Let's now assume that now we have a camera on the platform



means the 2nd robot

$${}^{0_2}J^{(2)}_{2/0_2} = \begin{bmatrix} {}^{0_2}K_1 & {}^{0_2}K_2 \\ \hline \emptyset & \emptyset \end{bmatrix}$$

We set $P_1 \equiv P_0$

the rotation is on the z axis

$R_1 \; R_z(q_2)$

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

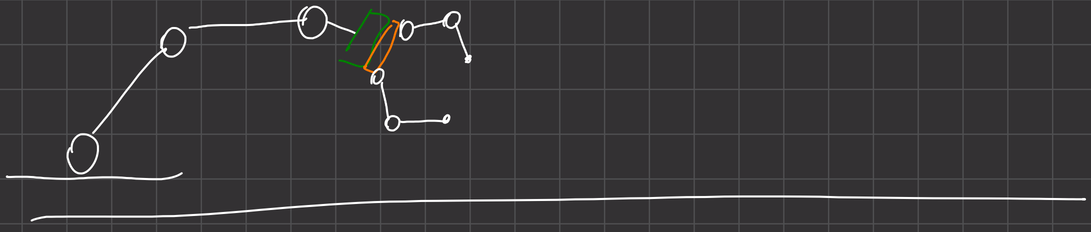
$$\left[{}^{0}J^{A}_{e_1/0} \;\vdots\; {}^{0}_{e_1}R \, {}^{0_2}J^{A}_{E_2/0_2} \right] \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix}$$

${}^{0_2}S_{E_2/2} \; {}^{0_2}J^{(2)}_{2/0_2}$ to take into account the camera calibration

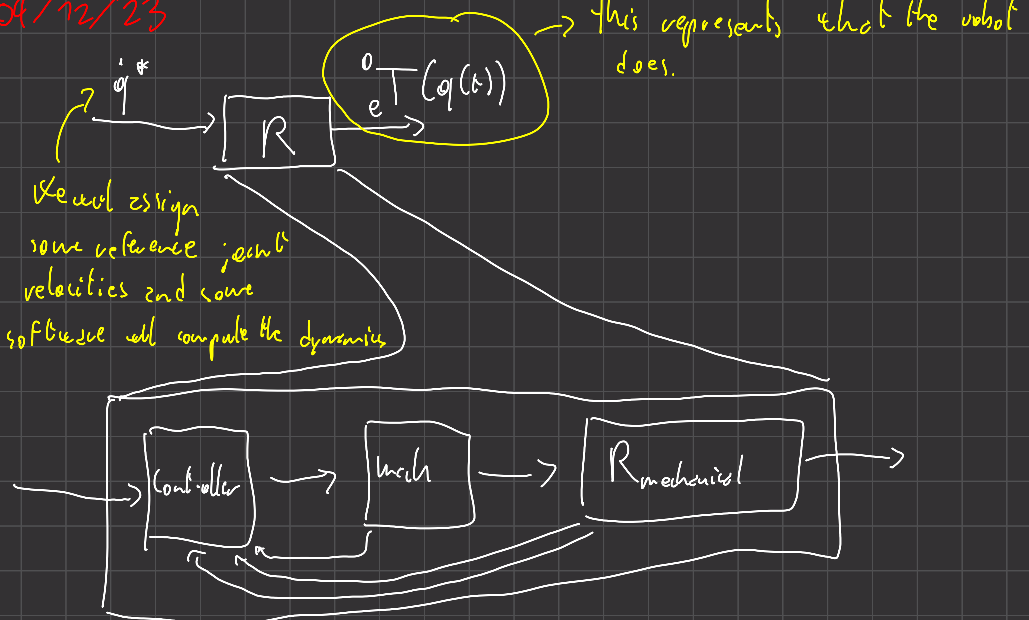
Transformation Matrices

$${}^{0}T_{F_1} = {}^{0}T_1 \, {}^{1}T_2 \, {}^{2}T_3 = \begin{bmatrix} R_0 & \begin{matrix} 0 \\ 0 \\ \ell \end{matrix} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_1 & \begin{matrix} 0 \\ 0 \\ y \end{matrix} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_z(q_1) & \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} \\ 0 & 1 \end{bmatrix}$$

EX 2



07/12/23



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

We know how to compute
$$\begin{cases} {}^0_e T(\underline{q}) \rightarrow \text{position and orientation} \\ {}^0 J_{e/0}(\underline{q}) \rightarrow \text{velocity and angular.} \end{cases}$$

↳ this will be written as $J \in \mathbb{R}^{6 \times n}$ for simplicity

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

What to send to the ee to have a given linear and angular velocity.

Inverse Kinematic Problem

Computing $\dot{q}$ s.t. :

$$\begin{bmatrix} {}^0 \omega_{e/0} \\ {}^0 v_{e/0} \end{bmatrix} \simeq \begin{bmatrix} \underline{\omega}^* \\ \underline{v}^* \end{bmatrix} = \dot{\underline{x}}^*$$

the angular velocity of the ee. — desired v and ω

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

There are three possible situations

1) $n=6$ (same number of d.o.f as the number of space in which velocity expressed)

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

$|J| \neq \emptyset$

($\exists!$ solution)

If we have $|J|=0$ then we are in a singularity.

2) $n>6 \Rightarrow \exists$ infinite many solutions $(\infty^{n-6}) = [Ay=b]$ ↳ Linear combinations of the column of A

3) $n<6 \Rightarrow \nexists$ solution for any choice of $\dot{x}^*$

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

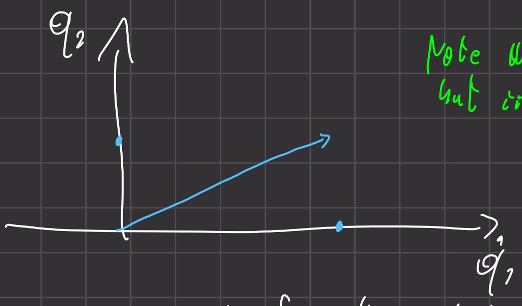
When this condition exist, then $b \in \text{span}(A)$. And this is why (3) might not have a solution

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

The idea is to generate the smallest $\dot{q}$ such that $\dot{x}^* = J\dot{q}$ is satisfied.

We can say that

$$\dot{q} = \underset{\dot{q}}{\arg\min} \frac{1}{2} \|\dot{q}\|^2 = \underset{\dot{q}}{\arg\min} \frac{1}{2} \dot{q}^T \dot{q}$$



Note that $\dot{q}$ is not a geometric vector but it's only an interpretation

$$\|\dot{q}\| = \dot{q}_1^2 + \dot{q}_2^2 = \dot{q}^T \dot{q}$$

We are going to search for the smallest joint velocities that are constrained by these conditions.

Note

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

square symmetric matrix

$A = A^T = \mathbb{R}^{n \times n}$

$A > 0$

This means that the matrix is strictly positive defined

symmetric ⇒ all eigenvalues strictly positive ⇒ always diagonalizable

$f(\underline{q}) > 0$ is valid for any $\underline{q} \neq \underline{\emptyset}$

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 $\dot{q}^T q$ is a particular case of $\dot{q}^T A \dot{q}$ where $A = \mathbb{1}$

We will use the Lagrange multipliers method.

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

$$= \frac{1}{2} \dot{q}^T \dot{q} + \underbrace{\lambda^T}_{\text{linear function in lambda}} [J \dot{q} - \dot{x}^*]$$ → this is a scalar function

2) $\frac{\partial \mathcal{L}}{\partial \dot{q}} = \emptyset \iff \underline{\dot{q} + J^T \lambda} = \emptyset$ → everything here is transposed.

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

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

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

We can write ③ as:

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

right pseudoinverse of J.

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

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

$$\left.\begin{array}{l} \dot{q} = -J^T \lambda \\ J \dot{q} = \dot{x}^* \end{array}\right\} -J J^T \lambda = \dot{x}^*$$

If J is not singular then this is always full rank and square $(J J^T) \in \mathbb{R}^{6 \times 6}$ and symmetric

$$(J J^T)^T = (J J^T)$$

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

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

↳ I can invert this for what I said previously

$$\underline{\dot{q} = J^T (J J^T)^{-1} \dot{x}^*}$$ 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? $J^{\#}$ is void.
If this algorithm is implemented in a computer then the numbers are always wrong.
What happens if the robot is close to a singular configuration?

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

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

It always exists two orthonormal matrices such that:

$$A = U^T \Sigma V \qquad U \in \mathbb{R}^{n \times n} \quad V \in \mathbb{R}^{m \times m}$$

$$U^T = U^{-1} \qquad V^T = V^{-1}$$

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

these are the singular values. Not eigenvalues

$$\lambda_i = \sqrt{\underbrace{\sigma_i}_{\text{eigenvalue}} \underbrace{(A A^T)}_{\text{semipositive matrix in the zero case.}}}$$

this means eigenvalue of AA^T. It's not a multiplication

We also have that $\lambda_1 \geq \lambda_2 \geq \ldots \geq \underbrace{\lambda_n}_{=0} \geq 0$ if in singular configuration.

Note that the singular values tend to go to $\emptyset$.

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

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

Let's now assume that $J = \underbrace{U^T \Sigma V}_{\text{SVD}}$

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

$$J^{\#} = V^T \Sigma^T U \left[U^T \Sigma \underbrace{\cancel{V} \cancel{V^T}}_{I} \Sigma^T U \right]^{-1}$$

$(AB)^{-1} = B^{-1} A^{-1}$ but $U^{-1} = U^T$

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

$$J^{\#} = V^T \Sigma^T \underbrace{\cancel{U U^T}}_{1} \begin{bmatrix} \frac{1}{\lambda_1^2} & & \\ & \ddots & \\ & & \frac{1}{\lambda_n^2} \end{bmatrix} U =$$

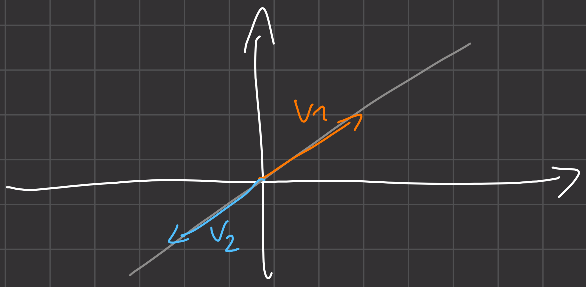
$$= V^T \begin{bmatrix} \frac{1}{\lambda_1} & & \emptyset \\ & \ddots & \\ \emptyset & & \frac{1}{\lambda_n} \\ & \emptyset & \end{bmatrix} U$$

The steps to compute: $\dot{q} = \underset{\dot{q}}{\operatorname{argmin}} \frac{1}{2} \dot{q}^T \dot{q}$ are:

1) Compute U, V, Σ
2) Compute $\Sigma^{\#} = \begin{bmatrix} 1/\lambda_1 & & \\ & \ddots & \\ & & 1/\lambda_n \end{bmatrix}$
3) $\dot{q} = J^{\#} \dot{x}^* = V^T \Sigma^{\#} U \underline{\dot{x}^*}$ No inversions.

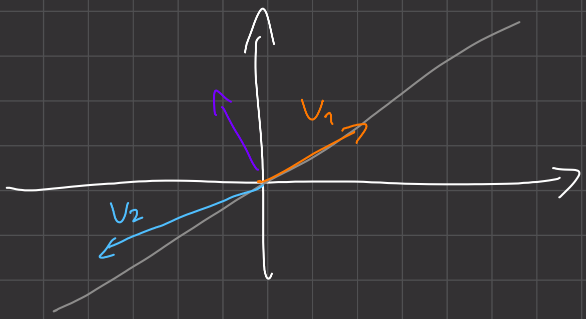
How to address sensitivity

If you consider that $\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} \underline{v_1} & \underline{v_2} \end{bmatrix}$ is singular which means that the column are parallel so:



Now if we want to find a solution to $Ay = b$ 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 moved by a tiny amount:

$$\frac{\|\delta y\|}{\|y\|} \leq \underbrace{\overbrace{\frac{\lambda_{max}(A)}{\lambda_{min}(A)}}^{\chi(A)} \frac{\|\delta b\|}{\|b\|}}$$

N.B

relative error w.r.t. to the nominal 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 tending to a singular configuration then the smallest singular value goes to $\emptyset$.

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.

$$J^{\#} = J^{T}(JJ^{T} + \underbrace{\gamma}_{\text{small coefficient}} \mathbb{1}_6)^{-1}$$

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

---

$$\underbrace{\dot{q} = \underset{\dot{q}}{\operatorname{argmin}}\ \dot{q}^T \dot{q}}_{\text{Least squares solution}} \quad \text{s.t. } J\dot{q} = \dot{x}^*$$

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

These weights could depend on the power of the motors.

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

$W = W^T > \phi$

Examples

a) $W = \begin{bmatrix} w_1 & & \phi \\ & \ddots & \\ \phi & & w_n \end{bmatrix}$

b) $|\dot{q}_i| \leq \dot{q}_i^{max}$ → the larger $\dot{q}_i^{max}$, the larger I accept values

c) $W \triangleq A(q)$ → Robot inertia matrix $\quad \dot{q}^T A(q) \dot{q} \triangleq$ Kinetic energy

$A = A^T > \phi$

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

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

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

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

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

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

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

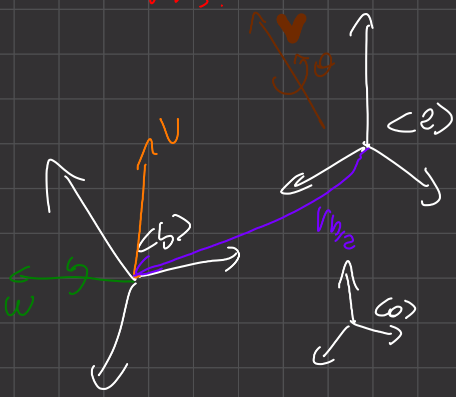
→ Weighted pseudoinverse matrix

$W = W^T > 0 \qquad W^{\frac{1}{2}}$ The square roots of the singular values in this case are the same as eigenvalues?

MB

$(J W_2^A)$ Weighted least squares.

Robot tasks.



The point is how to drive <b> in order to send <b> in the same position of <e>.

So we have ${}^0_eT$ and ${}^0_bT$

I want to move b such that after a while reaches e.

$${}^0_eT = {}^0_bT$$

If <B> is attached to a robot we can compute the IKP and calculate joint's velocities.

Now to calculate V and ω. I can define this vector $r_{b/e}$

I must send $r_{b/e}$ to ∅ to put the frames coincident.

There exists also ${}^e_bR$ that needs to go to 1

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