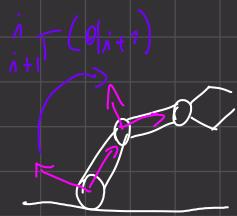


70/02/24

In this course we will mainly focus on manipulators.



We see that we could assign variables to joints,

$$\dot{q}(t) = \begin{bmatrix} \dot{q}_1(t) \\ \vdots \\ \dot{q}_n(t) \end{bmatrix}$$

in order to control the robot.

There are two important entities:

$$\left\{ \begin{array}{l} J_R(q_{int}) \\ \pi_{int}/(q_{int}) \end{array} \right.$$

This is what we called forward geometry.

In order to control the robot we need to use the inverse geometric model.

There is always a solution to the forward geometric model.
In the inverse problem we are in trouble, if the robot is redundant.
We can have infinite many solutions.

We also have to keep into consideration what happens when time passes.

Typically we have an end effector attached to the end-effector of the robot.

Variables change in time so it might be interested by the velocities generated by the end effector.

So we have two steps:

- What happens if for a given configuration at a given time, I want to assign a set of joint velocities:

$$\dot{q}(t) = \begin{bmatrix} \dot{q}_1(t) \\ \vdots \\ \dot{q}_n(t) \end{bmatrix}$$

If I can compute the velocities then I can compute the system velocity or all the art any frame of interest.
We can also compute also the linear velocity.

Forward kinematic problem

$$\begin{bmatrix} w_{e/\phi}(t) \\ v_{e/\phi}(t) \end{bmatrix} = J_{e/\phi}(q) \underbrace{\dot{q}(t)}_{\text{input}} \quad \boxed{\text{N.B. These are all function of time}}$$

↓ ↓ ↑

linear transformation

↑ ↑ ↑

Output Input

Computing this solves the forward Kinematic Problem

Inverse Kinematic Problem

Here control comes heavily into play.

Now we have to restate the problem.

In this case I want to achieve a particular angular velocity at the EE. this is the forward model of the robot. Not software

$$x^* = \begin{bmatrix} w^*(t) \\ v^*(t) \end{bmatrix} = J(q) \underbrace{\dot{q}(t)}_{\text{output}} \quad \begin{array}{l} \text{the output is not explicitly expressed.} \\ \text{this expression falls in this type of formule: } Ax = y \end{array}$$

↑ ↓ ↑

Input Output

Now we have to follow rules depending on the shape of J

- 1) Square → Have you got the solution (excluding sing.)
- 2) Flat (more columns than rows) → Infinite solutions
- 3) Tall (less columns than rows) → You might not find the solution
↳ This is the typical case of mobile robots

In the case of square problems we can compute:

$$> \text{If square matrix } = \dot{q}(t) = J^{-1}x(t)$$

$$> \text{if flat matrix } = \text{We have more solution } \begin{pmatrix} \text{the difference between rows} \\ \text{and cols number} \end{pmatrix}$$

The simplest way of solving this is using the least square solution.

$$\dot{q}(t) = \underbrace{J^T(JJ^T)^{-1}x^*}_{\text{Left pseudoinv}} \quad (\text{This is software})$$

There are restrictions because the motors or some joints may not be as powerful as required

This is a closed form formula.

The pseudoinverse has some properties: $J^\# J = I$

So far square and flat we can find only one solution.

In full configuration you have a least effort solution

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

In every case we still have problems with singularities.

In this case you get something close to the movement that you desire.

You go from J to $J^\#$ by using SVD

$$A = U \Sigma V^T$$
$$\downarrow$$
$$\begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n & \emptyset \end{bmatrix}$$
$$\sigma_1 > \dots > \sigma_n > 0$$

(Hi):

$$\check{\chi} = \frac{\sigma_1}{\sigma_n}$$

(CONDITION NUMBER OF THE MATRIX)

If χ is big then the system will be noisy and the result will be big. This effect gets bigger when you are close to singularities.

In practice you need to adjust the singular values in order to bound the result, and not have jerky motions.

The solution to this problem is to compute $J^\#$ like this:

$$J^\# = J^T (J J^T + \delta I)^{-1}$$

λ must be added when this is getting critical. λ generates a wrong solution but keeps everything bounded.

This is called regularization.

The pseudoinverse of A is technically computed as:

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

\Downarrow \mathbb{C}

(Orthonormal matrices) \Rightarrow

$$\begin{cases} U U^T = I_n \\ V^T V = I_m \end{cases}$$

$$\sum^{\#} = \begin{bmatrix} \frac{1}{\sigma_1} & & \\ & \ddots & \\ & & \frac{1}{\sigma_n} \end{bmatrix}$$

When the critical singular values get critical you can play with lambda.

Every inverse should be done with the pseudoinverse

Inverse geometry problem

I want to find two values:

$$eR^*(t)$$

$$eR(q)$$

$$n_{e/\phi}^*$$

$$n_{e/\phi}(q)$$

* means desired.

(displacement is a vector)

\curvearrowleft linear error

$$e_L(t) = n_{e/\phi} - n_{e/\phi}^*$$

\hookrightarrow actual position

\rightarrow desired position

misalignment between the two forces

$$\rho = v \theta$$

\hookrightarrow angular error.

My control objective is to $\left\{ \begin{array}{l} \rho \rightarrow \phi \\ e_L \rightarrow \phi \end{array} \right\}$ Here we want to reduce the distance

N.B. DISPLACEMENT \neq DISTANCE
(vector) (scalar)

$$\frac{1}{2} |e_L|^2 = \frac{1}{2} e_L \cdot e_L$$

\downarrow we want

\hookrightarrow the derivative of this $\sqrt{\text{negative}}$

$$\frac{1}{2} \frac{d}{dt} (e_L \cdot e_L) = \frac{1}{2} e_L \cdot \dot{e}_L + \frac{1}{2} \dot{e}_L \cdot e_L = e_L \cdot \dot{e}_L = e_L \cdot [v_{e/\phi} - v^*]$$

This is where I define my control structure.

Note that e_L is given

$$n_{e/\phi} \cdot e_L$$

$$n_{e/\phi}^* \cdot$$

\uparrow
this is the control signal

$\left(\begin{array}{l} \text{We want to choose } v_{e/\phi} \\ \text{so that the scalar product is negative} \end{array} \right)$

This process is called control synthesis

We can express:

$$e_L \cdot [v_{e_\phi} - v^*] = -\gamma(e_L \cdot e_L) \omega$$

$\underbrace{v_{e_\phi} - v^*}_{\sim \tau e_L}$

If I choose this for the term inside the matrix this will be guaranteed

We want to guarantee that:

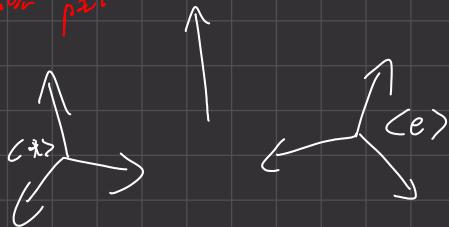
$$v_{e_\phi} - v^* = -\gamma e_L$$

↑ ↑ ↑
output "input" Control parameter input

$$\boxed{v_{e_\phi} = -\gamma e_L + v^*}$$

$\boxed{\dot{x}^*}$ ↓ (feed forward term providing velocity orientation
feedback control to keep track of the goal.
term.

Angular part



$$\frac{1}{2} |\rho \cdot \rho|^{\frac{1}{2}} = \frac{1}{2} \theta^2$$

This because ρ
is unit vector

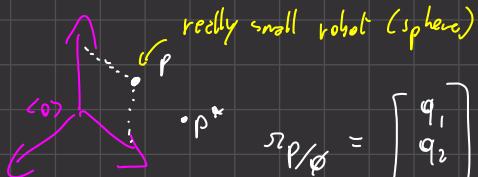
$$-\gamma \omega \theta$$

$$\frac{d}{dt} \frac{1}{2} (\rho \cdot \rho) = \theta r \cdot (\omega_\phi - \omega^*)$$

I want this to be opposite of
 ω because I want to turn
the object velocity around.

$$\boxed{\omega_\phi = -\gamma \omega \theta + \omega^*}$$

27/02/24



really small robot (sphere)

$$\rho/\phi = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix}$$

$$\dot{\rho}/\phi = \begin{bmatrix} \dot{q}_1 & \omega_j & \omega_k \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}$$

This is how we write
this in MCM

Want to move p toward p^*

The course will be divided in two loops:

- 1) Review of Fundamentals of mechanics
- 2) Computational Mechanics. (Newton Euler recursive equations)
- 3) Dynamic models of holonomic robots.

At the end the robot will be:

$$A(q)\ddot{q} + B(q, \dot{q})\dot{q} + C(q) = M + D$$

↓ ↓ ↓ ↑ ↑
inertie coriolis gravity torques disturbances
cont. pos.

REMARK

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

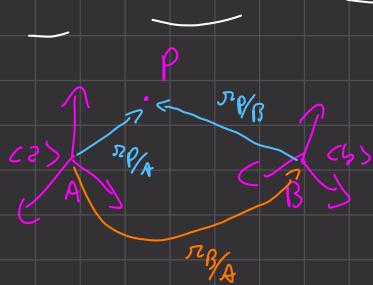
- 4) Control Algorithms. / Fundamentals of Robot dynamics contd.

- 5) Less standard control algorithms. (Dynamic Motion Primitives)

$$\frac{d}{dt} M(t) = \frac{d}{dt} \mathbf{r}(t) + \omega_{\mathbf{r}/\mathbf{d}} \times \mathbf{r}$$

↑
c_d
↑
c_r

version of the mother of all the formulas.



$$\frac{d}{dt} \left(\mathbf{r}_{P/A} \right) = \frac{d}{dt} \left(\mathbf{r}_{P/B} + \mathbf{r}_{B/A} \right) = \underbrace{\frac{d}{dt} \mathbf{r}_{B/A}}_{V_{B/A}} + \underbrace{\frac{d}{dt} \mathbf{r}_{P/B}}_{V_{P/B}} + \omega_{B/A} \times \mathbf{r}_{P/B}$$

$$V_{P/A} = V_{B/A} + V_{P/B} + \omega_{B/A} \times \mathbf{r}_{P/B} \quad (7)$$

$$\frac{dz}{dt} \left(v_{P/A} \right) = \frac{dz}{dt} \left[v_{B/A} + v_{P/B} + \omega_{B/A} \times r_{P/B} \right] =$$

*linear
velocity*

$$\Rightarrow \frac{dz}{dt} v_{B/A} + \frac{dz}{dt} v_{P/B} + \omega_{B/A} \times v_{P/B} + \left(\frac{dz}{dt} \omega_{B/A} \right) \times r_{P/B} + \omega_{B/A} \times \left(\frac{dz}{dt} r_{P/B} \right) =$$

$$= \dot{r}_{P/A} + \dot{r}_{P/B} + \omega_{B/A} \times v_{P/B} + \left(\frac{dz}{dt} \omega_{B/A} \right) \times r_{P/B} + \omega_{B/A} \times \left(v_{P/B} + \omega_{B/A} \times r_{P/B} \right) =$$

$$= \dot{r}_{P/A} = \dot{r}_{P/A} + \dot{r}_{P/B} + 2 \cdot (\omega_{B/A} \times v_{P/B}) + \left(\frac{dz}{dt} \omega_{B/A} \right) \times r_{P/B} + \omega_{B/A} \times (\omega_{B/A} \times r_{P/B})$$

centrifugal accel.

(2)

Proof that $\omega_{B/A} = \omega_{B/B}$

$$\frac{dz}{dt} \omega_{B/A} = \frac{dz}{dt} \omega_{B/A} + \cancel{\omega_{B/A} \times v_{B/A}}$$

So we can always write $\omega_{B/A}$

Point Mass

Ideal mechanical entity characterised by one quantity which is the mass. \Rightarrow

It is a point in euclidean sense

Isolated point

A point which is not interacting with any other point in the universe.

Force

Mechanical entity responsible the interaction between point's mass

Inertial reference frames

Reference frame which is not rotating wrt. the stars.

Newton Laws

1) An isolated point mass has constant velocity wrt. inertial reference frames.

2) Given a point mass $m \Rightarrow m \ddot{z} = F$

inertial quantities

$$3) \begin{array}{c} P \\ \nearrow Q \\ Q \\ \searrow Q \end{array} \left\{ \begin{array}{l} F_{Q \rightarrow P} = -F_{P \rightarrow Q} \\ F_{Q \rightarrow P} \times F_{P \rightarrow Q} = \emptyset \end{array} \right.$$

\rightarrow can be seen as a kind of control action.
 $m \ddot{r}_{P/\emptyset} = F(r_{P/\emptyset}, \dot{r}_{P/\emptyset}, t)$
 \hookrightarrow something in function of time.

1) Forward Dynamic Problem

Computing $r_{P/\emptyset}$ and $\dot{r}_{P/\emptyset}$ given $F(\cdot, \cdot, \cdot)$

$$\begin{matrix} r_{P/\emptyset}, \dot{r}_{P/\emptyset} \\ | \\ t=t_0 \end{matrix}$$

2) Inverse Dynamic Problem ("Control Problem")

I want the point to move with a given trajectory and I want to know the forces to move the point with that trajectory.

Given $r_{P/\emptyset}^*, \dot{r}_{P/\emptyset}^*, \ddot{r}_{P/\emptyset}^*$ compute F .

Compute the forward dynamic model:

Input:

$$\begin{matrix} -F \\ -m \end{matrix}$$

Parameter:

$$-h \text{ (sample time)}$$

$$\ddot{r}_{P/\emptyset} = \frac{1}{m} \cdot F \quad \leftarrow \text{This is valid for a given time instant.}$$

initialization:

$$\begin{matrix} r_{P/\emptyset}(1) = * \\ \dot{r}_{P/\emptyset}(1) = * \end{matrix}$$

for $i=1:N$:

$$\dot{\dot{r}}_{P/\emptyset}(t+1) = \frac{1}{m} F(i+1) \quad \rightarrow$$

$$\dot{r}_{P/\emptyset}(t+1) = \dot{r}_{P/\emptyset}(i) + h \dot{r}_p(i)$$

$$r_{P/\emptyset}(t+1) = r_{P/\emptyset}(i) + h r_{P/\emptyset}(i)$$

$$\left\{ \begin{array}{l} \dot{x} = f(x) \\ \frac{x(t+h) - x(t)}{h} = f(x(t)) \end{array} \right. \quad \text{Explicit euler formula.}$$

end

$$\nabla F = -m \cdot g \cdot \delta \times \left[\text{force} \left(r_1, r_2; \gamma_1, \gamma_2; N+1 \right) \right] \quad ???$$

\Rightarrow Function $F = \text{force}(r_{p1}, r_{p2})$

Equilibrium (POINT)

$r_{p1}^E = \text{constant}$ (wrt some inertial reference frame)
At

$$\dot{r}_{p1}^E \equiv \emptyset \quad \text{The resultant forces are equal to } \emptyset.$$

$$\ddot{r}_{p1}^E \equiv \emptyset$$

Energy (scalar)

It's related to the capability of some system to generate work.

$$E = T + V$$

↳ Potential

\downarrow

$\frac{1}{2} \left(\dot{r}_{p1}^2 + \dot{r}_{p2}^2 \right) m$

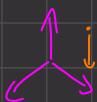
\downarrow

$F_i = \frac{\partial V}{\partial r_i}$

conservative
(they do not dissipate energy)

Examples of conservative forces

1) Gravity



$$F_g = -mg \hat{R}$$

$$V_g = mg (r_{p1} \cdot \hat{R})$$

2) Elastic Force



$$F_{Q \rightarrow P} = -k_E \underbrace{(Q - P)}_{r \approx p}$$

↳ Constant

$$V_e = \frac{1}{2} k_e \left(r_{q/p} \cdot r_{q/p} \right)$$

LINEAR MOMENTUM and angular ρ

Linear : $\underline{P} = m \cdot \underline{v}_p$

Angular : $\underline{L}_B = m \cdot r_{p/B} \times \underline{v}_p$

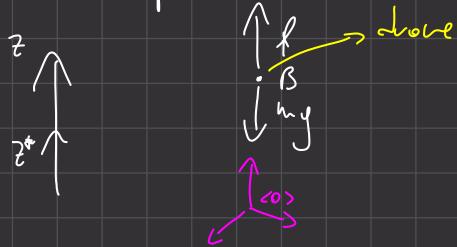
L_B is orthogonal to $r_{p/B}$ and v_p

Moment of a force (Torque)

$$M = r_{p/B} \times F$$

26/02/20

Another example



I build a model

$$\phi_m \dot{r}_{p/B/\phi} = \underline{F} \quad \text{result of external forces}$$

$$\phi_{r_{p/B}/\phi} = \begin{bmatrix} 0 \\ 0 \\ z \end{bmatrix} \quad \phi_F = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix}$$

So I collapse my model in:

$$m \ddot{\underline{r}} = \underline{F}_z$$

Forces in this example:

- gravity : $m g$
- the propellers: f

Can I control the model?

$$\dot{x} = \begin{bmatrix} 0 \\ 0 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ x_3 \end{bmatrix}$$

EXTERIOR FORCES: forces acting on the body

I want to drive my mass with \dot{x} in my desired position (z^*)

I look $m\ddot{z} = F_z$ and to change z I have to change \dot{x}
How to change \dot{x} ? Changing F_z

F_z is composed by 2 terms \cancel{x} any my $\left\{ \begin{array}{l} \text{in fixed} \\ \text{g fixed} \end{array} \right.$
So I should change \dot{x}

Law: $\dot{x} = K_x \underbrace{\omega}_{\substack{\downarrow \\ \text{constant}}} \quad \downarrow \text{Turning speed of the propellers}$

Propellers tend to push up the point, how to control ω ?

$$u(t) \rightarrow \boxed{\substack{\text{motor} + \\ \text{propeller}}} \rightarrow \omega$$

\downarrow
input / what I control,
it's reasonable that $\omega(t) = K\omega u(t)$

I rewrite the equation:

$$\ddot{z} = \frac{1}{m} \left[-my + \dot{x} \right] = \frac{1}{m} \left[-my + (K_x + K_u)u \right] = \\ = \frac{1}{m} \left[-my + K_u u \right]$$

So we are in this condition:

$$\overbrace{u(t)}^{\downarrow d} \rightarrow \boxed{S} \rightarrow z(t)$$

The gravity for example is a constant disturbance

My control \rightarrow my control law: specification to implement, your software, typically about

let's assume your system is in the desired position:

$$z(t) = z^* \quad \text{②}$$

Then we can define an error quantity:

$$e(t) = z(t) - z^* = \emptyset$$

This is what I want

So this means:

$$\dot{e}(t) = \dot{z}(t) - \dot{z}^* = \emptyset$$

$$\ddot{e}(t) = \ddot{z}(t) - \ddot{z}^* = \emptyset$$

③ this is valid under this assumption:

$$\ddot{z}(t) = \emptyset$$

So z^* is constant

We can claim:

$$\ddot{z} = \emptyset = \frac{1}{m} [-mg + K u]$$

$$\frac{1}{m} K u = g \Rightarrow \boxed{u_0(t) = \frac{1}{K} mg} \rightarrow \text{This is control}$$

To know $K \rightarrow$ dynamometer

m is another parameter, known with a scale.

g is known

u_0 is the control

What's wrong with this?

To maintain the height of my system my computer must generate the

Example:

$$m = \rho_1 v [kg]$$

$$K = 10^{-3} \frac{60}{2^4} \left[\frac{N}{v \cdot kg} \right]$$

$$g = \varphi, p \left[\frac{m}{kg} \right]$$

But generally the point is not in the \bar{z}^* .

u_0 is still control variable.

By computing in this way this is an open loop signal. It's not good to cancel out disturbances.

That's due to the fact that $z(\varphi) \neq \bar{z}^*$ so $u(t) = u_0(t) + \dots$

not sufficient; this only

cancels out gravity so
the point is fixed.

But I want to control the height!

$$\bar{z} - z^* \\ e(t) \left\{ \int_{z^*}^z \right\}$$



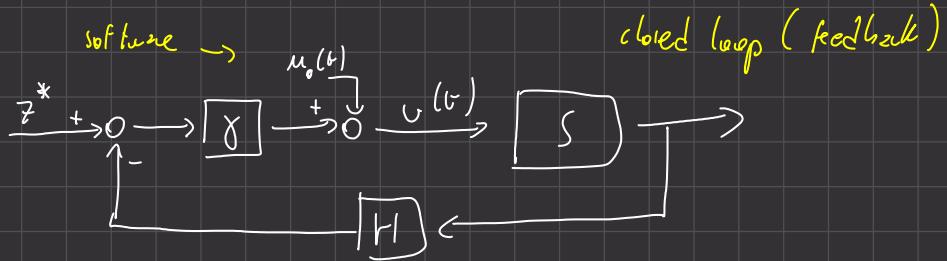
To slow down:

$$w_0(t) = K_w [u_0(t) - \varphi e(t)]$$

$$= K_w u(t)$$

If $e(t) > 0$ we are slowing down.

So what we do is this:



It's still not working because we are only analyzing the physics.

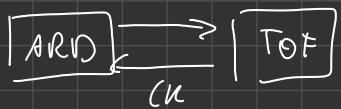
To measure what is the output of S you add the block $[M]$.

They are sensors. How to implement it?

BUS: I²C



SPI



?

For your software

For (; ;)

read_ic()

$$e = z - z^*$$

 $v = v_0 - \gamma e$ // This could be computed out of the loop here
write_ic();

<sync>

$$\dot{e}(t) = z - z^*$$

$$\dot{e}(t) = \ddot{z} - \ddot{z}^*$$

$$\dot{e}(t) - \ddot{z} - \ddot{z}^* = \frac{1}{m} [-m\gamma + m\mu] - \ddot{z}^*$$

I substitute what is μ

$$\dot{e} = \frac{1}{m} [\gamma e(t)] = -\frac{\gamma}{m} e(t)$$

this is the error dynamics

Here the control error is going with my action control signal

$$S_0: \ddot{e}(t) + \frac{1}{m} e(t) = 0 \quad 2^{\text{o}} \text{ order linear equation}$$

eq point.

Is this eq. point asymptotically stable?

I need to compute the poles associated with the system:

$$\lambda^2 + \frac{1}{m} = 0 \quad \text{strictly negative real parts of poles is required to be asym. stable eq. point.}$$

solutions:

$$\lambda_1, 2 = \pm j \sqrt{\frac{1}{m}}$$

This is not the case

It means if I apply this control law there is no way to reach indefinitely the goal.

We need to dissipate energy: add another term to check about where to the target. I have the relative velocity:

$$So: \dot{v}(t) = v_0(t) - \gamma e(t) - \alpha \dot{e}(t) \rightarrow \text{dissipating term}$$

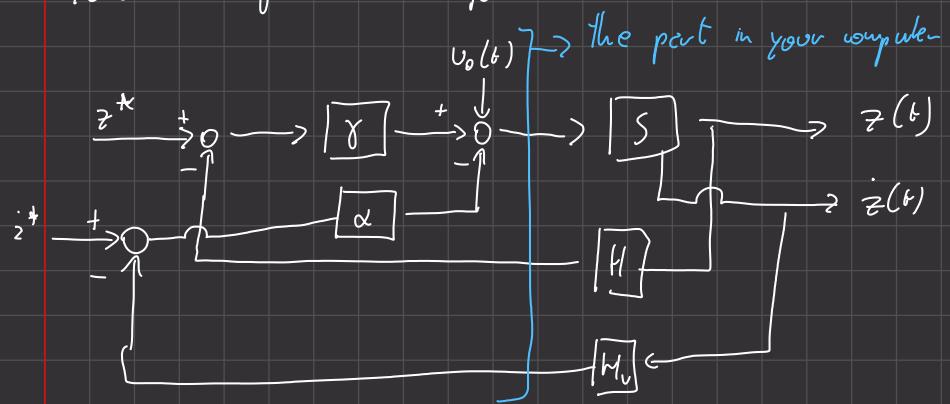
They are not real forces, the physics finishes with $v_0(t)$

Now we are considering control forces, they are virtual, but they have physical effects.

The sense of $- \alpha \dot{e}(t)$ is to decrease $\dot{e}(t)$ when you are away at the goal

But we have to estimate $\dot{e}(t)$: the velocity velocity:

You end up with this algorithm:



We add a control of velocity

So I find: $\ddot{e}(t) + \frac{\alpha}{m} \dot{e}(t) + \frac{\gamma}{m} e(t) = 0$

↳ closed loop error dynamics

$$\lambda^2 + \frac{\alpha}{m}\lambda + \frac{\gamma}{m} = 0$$

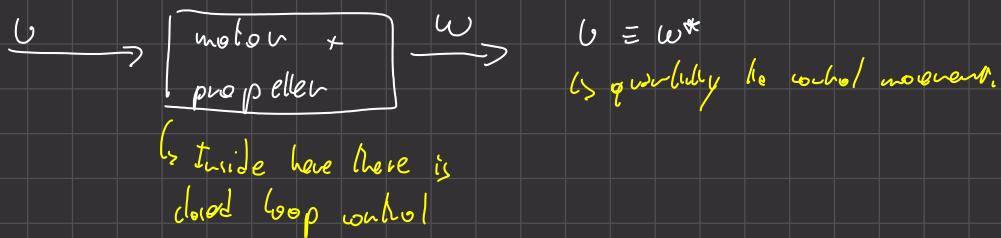
α and $\gamma > 0$ } cartesian theorem } $\begin{array}{c} x \\ \vdots \\ x \end{array} \rightarrow$
permutation of signs } theorem } strictly negative poles.

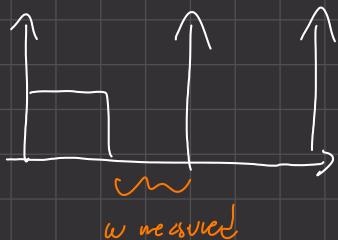
$\Rightarrow e(t)$ is an asym. stable equilibrium point. ✓

You must find a trade off between α and γ , because if one is greater than the other you can have oscillation or slowed down behaviour.

You should make experiments

$$\left\{ \begin{array}{l} \gamma \gg \alpha \text{ oscillations} \\ \alpha \gg \gamma \text{ slow motion} \end{array} \right.$$





sensor-less feed back control

$w_{measured}$ is a voltage and it is used in the block.

For us v is a digital number.

28/02/24

Study motion caused by external events. We need models to develop software.

model \rightarrow physical property of model

model \rightarrow control \rightarrow software \rightarrow simulation \rightarrow reality

A robot is a connection of bodies



In this case only rigid bodies

We consider a point B and its reference frame C equal for all points, because of rigid body.

Rigid Body constraint

$$|P - B| = \text{constant } \forall t$$

$$\text{Using our convention } |r_{P/B}| = \text{constant } \forall t$$

We have some considerations derived from mother of formulas.

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

(because $v_{B/B} = 0$ for rigid body constraint)

The velocity of the body is the sum of velocities of B wrt. ϕ and cross product between angular velocity of frame B wrt. ϕ and the displacement between P and B .

Also for the acceleration

$$\ddot{r}_{p/\phi} = \ddot{r}_{B/\phi} + \omega_{B/\phi} \times r_{p/B} + \omega_{B/\phi} \times (\omega_{B/\phi} \times r_{p/B})$$

these expressions hold for any point of the body and for any point which is fixed wrt. the body.

Center of Mass

The body is made of stuff connected to each other. If you cut 2 parts of the body \rightarrow it has a mass,

We can assume to have $\rho = \rho(p)$ TODO
 \downarrow (density?)

$$\downarrow m = \varphi dV$$

ρ is a positive function

If I sum all the small parts I have the mass:

$$m = \int_V \varphi(p) dV \quad \text{TODO}$$

$$(\bar{r}_c) = \frac{\int_V (p - \bar{r}_c) \rho dV}{\int_V \rho dV} = \frac{\int_V (p - \bar{r}_c) \rho dV}{m}$$

Formula for center of mass

The numerator is a sum of infinite vectors, the sum is weighted by the density.

In our notation:

$$\bar{r}_{c/\phi} = \frac{\int_V \bar{r}_{p/\phi} \rho dV}{m} \quad \text{point } \phi \text{ is an arbitrary point?} \quad \text{TODO}$$

Properties:

$$\bar{r}_{p/\phi} = \bar{r}_{p/B} + \bar{r}_{B/\phi}$$

$$\Rightarrow \bar{r}_{c/\phi} = \frac{\int_V (\underbrace{\bar{r}_{p/B} + \bar{r}_{B/\phi}}_{\bar{r}_{c/B}}) \rho dV}{m}$$

$$= \underbrace{\int_V r_{c/B} \rho dV}_{m} + \underbrace{\int_V r_{B/\phi} \rho dV}_{m} \xrightarrow{\text{not dependent on } \rho \text{ so out}} \text{TODO}$$

$$= \underbrace{\int_V r_{c/B} \varphi dV}_{m} + \underbrace{\cancel{\int_V r_{B/\phi} \varphi dV}}_{m}$$

$$r_{c/\phi} = r_{c/B} + r_{B/\phi}$$

We can express the center of mass wrt. any inertial reference frame \leftrightarrow by a translation ($r_{B/\phi}$)

② The body is the union of 2 distinct bodies.

Each one will have its own center of mass

$$B = B_1 \cup B_2 \quad \text{Example of a bottle}$$

$$B_1 \cap B_2 = \emptyset$$



$$\begin{aligned} \text{So } r_{c/\phi} &= \frac{\int_V r_{p/\phi} \rho dV}{m} \\ &= \frac{\int_{V_1} r_{p/\phi} \rho dV + \int_{V_2} r_{p/\phi} \rho dV}{m} \\ &= \frac{m_1 r_{c_1/\phi}}{m} + \frac{m_2 r_{c_2/\phi}}{m} \end{aligned}$$

Total center of mass is a linear combination of 2 center of masses

$$\begin{aligned} m \downarrow r_{c/\phi} &= m_1 r_{c_1/\phi} + m_2 r_{c_2/\phi} \\ (m_1+m_2) \end{aligned}$$

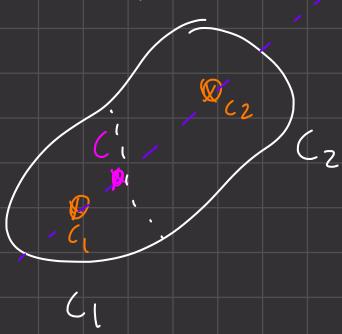
$$(m_1 r_{c/\phi} - m_1 r_{c_1/\phi}) + (m_2 r_{c/\phi} - m_2 r_{c_2/\phi}) = \phi$$

$$m_1 r_{c/c_1} + m_2 r_{c/c_2} = \phi$$

m_1 and m_2 are positive values and this means r_{c/c_1} and r_{c/c_2} must be parallel.

- So the center of mass lies on the line joining both centers of masses and it's closer to the bigger mass

On the picture:



So we summarize:

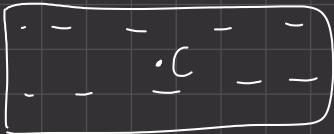
$$R_{C/\phi} = \frac{m_1 R_{C_1/\phi} + m_2 R_{C_2/\phi}}{m}$$

$$\text{and } R_{C/C_1} \parallel R_{C/C_2}$$

We can generalize this formula to n bodies:

$$R_{C/\phi} = \frac{1}{m} \sum_{i=1}^n m_i R_{C_i/\phi}$$

REMARK



⊗ C is not a physical point in the body.

⊗ C is the point where we consider all the mass to be concentrated.

For point mass:

$$p = m v_{p/\phi}$$

For linear movement:

$$L_B = m R_{p/B} \times v_{p/\phi}$$

I want to define these quantities for the whole body:

$$\begin{cases} dp = v_{p/\phi} dm \\ dL_B = (R_{p/B} \times v_{p/\phi}) dm \end{cases}$$

$$L_B = \int_V dL_B = \int (R_{p/B} \times v_{p/\phi}) \rho dv$$

$$= \int_V r_{p/B} \times [v_{B/\emptyset} + (\omega_{B/\emptyset} \times r_{p/B})] \rho dV$$

$$= \int_V (r_{p/B} \times v_{B/\emptyset}) \rho dV + \int_V r_{p/B} \times (\omega_{B/\emptyset} \times r_{p/B}) \rho dV$$

$$= m (r_{C/B} \times v_{B/\emptyset}) + I_B (\omega_{B/\emptyset})$$

\hookrightarrow It's a vector depending on the position of the center of mass w.r.t. B , which is an arbitrary point, and the velocity of point w.r.t. C .

B is arbitrary so I can choose it as C so $r_{C/C} = 0$

Inertia operator

$$I_B (\omega_{B/\emptyset}) \triangleq \int_V r_{p/B} \times (\omega_{B/\emptyset} \times r_{p/B}) \rho dV$$

It's a vector / precisely an operator
 \hookrightarrow transforms input into output

In this case : sum of infinite vectors derived from that operation.

The input is $\omega_{B/\emptyset}$, so I can see it as:

$$\omega_{B/\emptyset} \rightarrow \boxed{\quad} \rightarrow I_B(\cdot)$$

So I can write :

$$I_B(v) = \int_V r_{p/B} \times (m \times r_{p/B}) \rho dV$$

\hookrightarrow geometric vector

properties

D) The definition has arbitrary choice of B

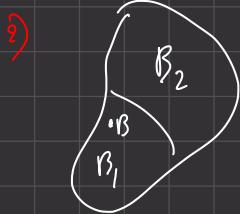
$$I_B(m) = I_C(m) + m r_{C/B} \times (m \times r_{C/B})$$

\hookrightarrow Center of mass

You condense the formula of inertia operator and the second term is considered as if the whole mass is concentrated in a single point, which is the center of mass.

\hookrightarrow The proof is in the notes

$$r_{C/B} = r_{p/C} + r_{C/B}$$



$$\beta = \beta_1 + \beta_2$$

$$\beta_1 \cap \beta_2 = \emptyset$$

We choose point P and we conclude:

$$I_B(\mu) = I_B^{(1)}(\mu) + I_B^{(2)}(\mu)$$

We notice the inertia operation is additive

3) $\mu \cdot I_B(\mu) \geq 0 \quad \forall \mu \neq 0 \quad \text{TODO}$

$\begin{cases} ? \\ ? \end{cases}$

This holds for physical objects

\hookrightarrow line is not

\hookrightarrow paper sheet depends

There might be some vectors for which the scalar prod is very small, so similar to zero.

Demonstration

$$\begin{aligned} \mu \cdot I_B(\mu) &= \mu \cdot \int_V \mu \cdot r_{P/B} \times (\mu \times r_{P/B}) \rho dV \\ &= \int_V \mu \left[r_{P/B} \times (\mu \times r_{P/B}) \right] \rho dV = \end{aligned}$$

$\boxed{\text{Note}}$

$$2 \cdot (b \times c) = c (a \times b) = b (c \times a)$$

$$= \int_V (\mu \times r_{P/B}) \cdot (\mu \times r_{P/B}) \rho dV$$

We have scalar prod of the same 2 vectors

$$= \int_V |(\mu \times r_{P/B})|^2 \rho dV \geq 0 \quad \Rightarrow \text{scalar prod so non-negative quantity}$$

$I^B > 0$ when μ is aligned with $(P-B)$ but notice:

- μ is arbitrary
- we are considering not pathological objects. For those reasons it holds

How do I compute these things?

Algebraic form of \mathbb{I}_B

$\mathbb{I}_B(\mu)$ is a geometric vector

I know it's an integral, integral = sum of bits.

$$\stackrel{\phi}{\mathbb{I}} \mathbb{I}_B(\mu) = \int_V \left[\mathbf{r}_{P/B} \times (\mu \times \mathbf{r}_{P/B}) \rho dV \right]$$

Project the operation on a frame for sake of simpl. cos2.

$$= \int_V \left[\stackrel{\phi}{\mathbb{I}} \mathbf{r}_{P/B} \times \right] \left\{ - \left[\mathbf{r}_{P/B} \times \right] \stackrel{\phi}{\mu} \right\} \rho dV =$$

NOTE

$$\stackrel{\phi}{\mathbb{I}} \mathbf{r} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad \stackrel{\phi}{\mathbb{I}} [\mathbf{r} \times] = \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix}$$

\mathbf{r} does not depend on ρ so out of the integral

, the - goes at the beginning and

$$-[\mathbf{r}_{P/B} \times] = [\mathbf{r}_{P/B} \times]^T$$

$$= \left[\int_V \stackrel{\phi}{\mathbb{I}} \left[\mathbf{r}_{P/B} \times \right] + \stackrel{\phi}{\mathbb{I}} \left[\mathbf{r}_{P/B} \times \right] \rho dV \right] \stackrel{\phi}{\mu}$$

This is a 3×3 frame expressed in frame ϕ

$$\stackrel{\phi}{\mathbb{I}} \mathbb{I}_B \in \mathbb{R}^{3 \times 3}$$

So:

$$\stackrel{\phi}{\mathbb{I}} \mathbb{I}_B = \int_V \left[\mathbf{r}_{P/B} \times \right] + \left[\mathbf{r}_{P/B} \times \right] \rho dV \in \mathbb{R}^{3 \times 3}$$

$$\text{and } \stackrel{\phi}{\mathbb{I}} \mathbb{I}_B = \stackrel{\phi}{\mathbb{I}} \mathbb{I}_B^T$$

This matrix changes depending on the position of the body.

If it's time varying in general

But I can choose a frame cob fixed with body:

$$\stackrel{b}{\mathbb{I}} \mathbb{I}_B = \int_V \left[\stackrel{b}{\mathbf{r}}_{P/B} \times \right]^T \left[\mathbf{r}_{P/B} \times \right] \rho dV$$

If I consider ab fixed with body the matrix does not change

because R_B/R is constant.

So ${}^b\mathbb{I}_B$ is constant.

How to link ${}^0\mathbb{I}_B$ and ${}^b\mathbb{I}_B$? Rotation Matrix

$${}^0\mathbb{I}_B = {}^bR \cdot {}^b\mathbb{I}_B \cdot {}^bR^{-1}$$

You will never compute the integral but the important thing is

Property 3 implies:

$$\boxed{M \cdot \mathbb{I}_B (u) \geq 0 \Leftrightarrow {}^bM \cdot {}^b\mathbb{I}_B \cdot {}^bu \geq 0}$$

${}^b\mathbb{I}_B$ is strictly positive defined matrix. It means:

$${}^b\mathbb{I}_B = {}^b\mathbb{I}_B^T \rightarrow \text{symmetric}$$

${}^b\mathbb{I}_B$ has 3 real positive eigenvalues

${}^b\mathbb{I}_B$ can be diagonalized: $\exists V \in \mathbb{R}^{3 \times 3}$ s.t. $V^{-1} \cdot {}^b\mathbb{I}_B \cdot V = \text{diag}(I_1, I_2, I_3)$

$$V \cdot V^T = I \quad (\text{e.g. } V \in \mathbb{R})$$

Eigenvalues

The eigenvalues: they describe principle directions and how the mass is distributed along the axis.

The smaller \approx the more the mass is not distributed.

Principle axis of inertia.

TOPOTII

Note: $\forall Q = Q^T \geq 0$

We have the property \rightarrow symmetry
 \rightarrow strictly real positive eigenvalues ν_1, \dots, ν_n
 ν_1, \dots, ν_n are on the normal vectors.

$$V = [\nu_1, \dots, \nu_n] \in \mathbb{R}^{n \times n}$$

this V diagonalizes Q , that means:

$$V^T Q V = \text{diag} \begin{bmatrix} d_1 & & \\ & \ddots & \\ & & d_n \end{bmatrix}$$

Definition of eigenvalue

The number which solves the equation

$$A v = \lambda v$$

\downarrow eigenvector \downarrow eigenvalue

So ${}^b I_B \in \mathbb{R}^{3 \times 3}$ $\lambda_1 = I_1 \geq 0$ $\lambda_2 = I_2 \geq 0$ $\lambda_3 = I_3 \geq 0$

${}^b I_B = {}^b I_B^T \Rightarrow \lambda_1 = I_1 \geq 0 \quad \left. \begin{array}{l} \text{They are called principle inertia} \\ \text{moments and are a property of the} \\ \text{body.} \end{array} \right\}$

$$V = [v_1, v_2, v_3] \in \mathbb{R}^{3 \times 3}$$

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

There exists a set of axis (z) for which J have an orthogonal matrix.

graphically:



These are the principle inertia direction.

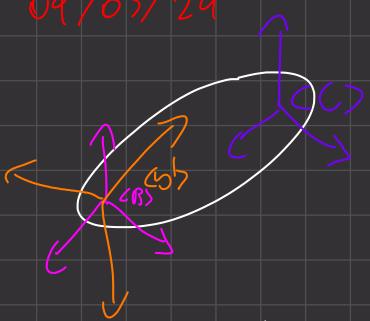
Note:

$$I_B(\mu) = I_c(\mu) + m r_{c/B} \times (\mu \lambda r_{c/B})$$

$${}^b I_B = {}^b I_c + m [{}^b r_{c/B} \times]^T [{}^b r_{c/B} \times]$$

size dim della precedente.

04/03/29



If the inertial reference frame is fixed w.r.t the frame of the body then the inertia operator is constant

$I_B(\cdot) \rightarrow^b I_B \rightarrow 3 \times 3 \text{ matrix and strictly positive, and symmetric}$

$$\begin{bmatrix} I_{11} & I_{12} & I_{13} \\ I_{21} & I_{22} & I_{23} \\ I_{31} & I_{32} & I_{33} \end{bmatrix}$$

We may think that
these come out the
c.c. and they define
the inertia.

$$\geq 0$$

Positive definite.

this has 3 eigenvalues and all strictly positive.

There exists a transformation orthogonal matrix that transforms
the inertia into diagonal. The terms on the diagonal are the
eigenvalues.

Every strictly positive matrix can be diagonalized ???

There exist as many eigenvectors as many eigenvalues.

$${}^b I_B v_i = \lambda_i v_i \quad i=1/s$$

$$\left[{}^b I_B v_1; {}^b I_B v_2; {}^b I_B v_3 \right] = \left[v_1 \lambda_1; v_2 \lambda_2; v_3 \lambda_3 \right]$$
$${}^b I_B \underbrace{\left[v_1; v_2; v_3 \right]}_{V} = \underbrace{\left[v_1; v_2; v_3 \right]}_{V} \underbrace{\begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}}_{\Delta}$$

$$v_i^T \cdot v_j = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases} \triangleq \delta_{ij}$$

$${}^b I_B V = V \Delta$$

↑
orthonormal basis (matrix) $\left(\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \left[v_1 \ v_2 \ v_3 \right] = I \right)$

I multiply to V^T

$$V^T {}^b I_B V = \Delta$$

There is a linear transformation to transform I_B into diagonal form.

V is an orthonormal matrix and can be seen in our case as a change of coordinates (change of space).

We change $\langle B \rangle$ to $\langle b' \rangle$ but it's still fixed in frame B .

$$V^T {}^b I_B V = {}^b I_B (\Delta)$$

To compute V I use SVD.

$$V = {}^b R \rightarrow \text{You want this to be right hand so } \det(V) = +1$$

If you compute V and let $i \neq j$ just swap two eigenvectors

We can now give physical meaning to Eigenvalues

$$\Delta = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} = \begin{bmatrix} {}^bI_{11} & {}^bI_{12} & 0 \\ {}^bI_{21} & {}^bI_{22} & 0 \\ 0 & 0 & {}^bI_{33} \end{bmatrix}$$

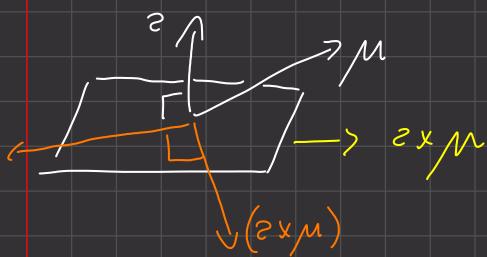
↳ 3 moments of inertia.

$${}^bI_B = {}^bI_C + m \begin{bmatrix} {}^b\mathbf{r}_{C/B} \times \mathbf{x} \end{bmatrix}^T \begin{bmatrix} {}^b\mathbf{r}_{C/B} \times \mathbf{x} \end{bmatrix}$$

↳ van N. 2 in journal because cross product square

↳ strictly positive definite

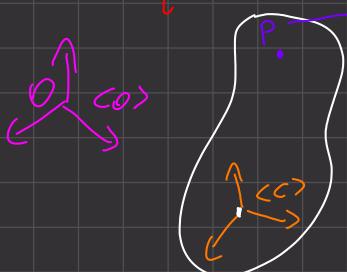
↳ semipositive definite



$$[z \times] ^T [z \times] I,$$

↳ \mathbf{z} belongs to the plane and we rotate by $(+)$ 90° on that plane.

Newton equations



Infinitesimally small mass

P will move following Newton's law (\approx):

$$\frac{d}{dt} \int_P \mathbf{v}_P \cdot d\mathbf{m} = \int_P \mathbf{f}_{P,ext} + \int_P \mathbf{f}_{P,int}$$

↳ result of all the forces which may be generated by all the other points of the body acting on P .

↳ force acting on P due to external forces.

density ρ , dV , volume

To explain this think of two balls connected by a spring and then put the balls in a piece of plastic



The internal force would be the spring and the tension

Let's apply this law to all parts of the body. So let's do an interval:

$$\int_V \frac{d}{dt} \dot{r}_{C/\phi} \rho dV = \int_V F_{EXT}^{\text{ext}} dV + \int_V F_{INT}^{\text{int}} dV$$

→ the integral is always equal to 0. It is not true for rigid bodies.
All this must be compensated otherwise the object would explode.

$\dot{r}_{C/\phi} = \omega_{C/\phi} \times r_{P/C}$

resultant of all external forces.

$$\frac{d}{dt} \dot{r}_{C/\phi} = \ddot{r}_{C/\phi} + \omega_{C/\phi} \times \dot{r}_{P/C} = \ddot{r}_{C/\phi} + \omega_{C/\phi} \times \dot{r}_{P/C} + \omega_{C/\phi} \times \left[\frac{d}{dt} \omega_{C/\phi} \times r_{P/C} \right]$$

$$\int_V \ddot{r}_{C/\phi} \rho dV + \int_V \omega_{C/\phi} \times \dot{r}_{P/C} \rho dV + \int_V \omega_{C/\phi} \times (\omega_{C/\phi} \times \dot{r}_{P/C}) \rho dV$$

$$m \ddot{r}_{C/\phi} + \omega_{C/\phi} \times \int_V \dot{r}_{P/C} \rho dV + \omega_{C/\phi} \times \left(\omega_{C/\phi} \times \int_V \dot{r}_{P/C} \rho dV \right)$$

$m \ddot{r}_{C/\phi}$ || $\ddot{r}_{P/C}$ || $\omega_{C/\phi} \times \dot{r}_{P/C}$

$$m \ddot{r}_{C/\phi} = F_{EXT}^{\text{ext}}$$

Formulation of the three Newton laws for rotational forces.
Forces on the body act as if they act on the center of mass.

$$m \ddot{r}_{C/\phi} = \underline{\underline{F}_{EXT}}$$

$$m \frac{d}{dt} \dot{r}_{C/\phi} = \underline{\underline{F}_{EXT}}$$

$\frac{d \dot{r}_{C/\phi}}{dt}$

$$\Rightarrow X = \begin{bmatrix} \dot{r}_{C/\phi} \\ \ddot{r}_{C/\phi} \end{bmatrix}$$

This extracts \mathbf{v}_{C_p}

$$\frac{d}{dt} \mathbf{x}(t) = \dot{\mathbf{x}}(t) = \begin{bmatrix} \phi_3 & \mathbf{I}_3 \\ \mathbf{0}_3 & \phi_2 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} \phi_3 \\ \mathbf{I}_3 \end{bmatrix} u(t)$$

Note: $\dot{\mathbf{x}}(t) = A\mathbf{x} + Bu$

Remark #1

$$\text{If } \begin{cases} F^{\text{ext}} = \phi & \forall t \\ \phi \mathbf{v}_{C/\phi}(t_0) = \phi \end{cases}$$

What is the solution of the previous formulas.

$$\Rightarrow \begin{cases} \mathbf{v}_{C/\phi}(t) = \phi \\ r_{C/\phi}(t) = x_\phi \end{cases} \quad \forall t$$

Equilibrium condition.

Example:

Bottle of water



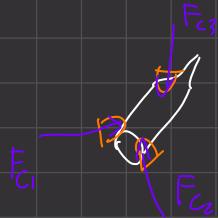
Each point is feeling gravity: $(g \text{ dm})$

$$y \text{ dm} = f_p^{\text{ext}}$$

$$\int_V f_p^{\text{ext}} dV = \int_V g dm = \int_V y \rho dV = y \int_V \rho dV = gm$$

Gravity acts as a pure force on the center of mass.

I know that on the bottle is acting one external force due to gravity and then I decide to grasp it



Let's assume that I can generate these forces

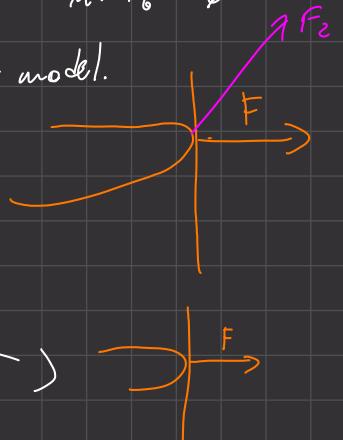
$$F_G = f_{C1} + f_{C2} + f_{C3}$$

↓
grasping force

To keep the bottle still I must guarantee that $F_m + F_6 = \emptyset$

for there kind of forces you need a contact model.

Point contact $\begin{cases} \text{Friction} \\ \text{No friction} \end{cases}$

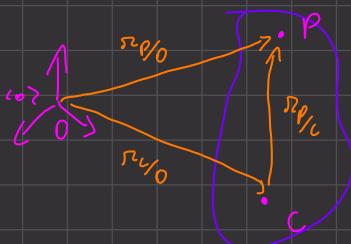


Soft finger contact

There is also some torque that the finger applies to the object not allowing rotation

Euler Equations 06/03/24

they describe the motion of the robot.



$$\frac{d\omega}{dt} V_{P/O} dm = F_p^{EXT} + F_p^{INT} \quad \text{and} \quad m_c = r_{P/C} \times F$$

So the moment is:

$$\begin{aligned} m_c &= r_{P/C} \times \frac{d\omega}{dt} V_{P/O} dm \\ &= V_{P/C} \times \not{F}_p^{EXT} + r_{P/C} \times \not{F}_p^{INT} \\ &= m_{P/C}^{EXT} + m_{P/C}^{INT} \end{aligned}$$

To have the moment for the whole body while keeping generality, we integrate:

$$\int_V \left(r_{P/C} \times \frac{d\omega}{dt} V_{P/O} \right) \rho dV = \int_V m_{P/C}^{EXT} dV + \int_V m_{P/C}^{INT} dV$$

Note:

$$P \cdot \cancel{f}_{Q \rightarrow P} \quad \cancel{f}_{P \rightarrow Q}$$

$$(P-C) \times \cancel{f}_{Q \rightarrow P} + (Q-C) \times \cancel{f}_{P \rightarrow Q} = [(P-C) - (Q-C)] \times \cancel{f}_{Q \rightarrow P} = (P-Q) \times \cancel{f}_{Q \rightarrow P} = \emptyset$$

Newton 3 law Newton 3 law

Valid for every $\cancel{f}_{Q \rightarrow P}$, so for every point of a rigid body

$$\frac{d\phi}{dt} \left(r_{p/c} \times v_{p/\phi} \right) = \frac{d\phi}{dt} r_{p/c} \times v_{p/0} + r_{p/c} \times \frac{d\phi}{dt} v_{p/0}$$

$$\left(r_{p/c} \times \frac{d\phi}{dt} v_{p/\phi} \right) = \frac{d\phi}{dt} \left(r_{p/c} \times v_{p/0} \right) - \frac{d\phi}{dt} [r_{p/0} - r_{c/0}] \times v_{p/\phi}$$

$\left(v_{p/\phi} - v_{c/0} \right) \times v_{p/0}$

$$= \frac{d\phi}{dt} \left(r_{p/c} \times v_{p/\phi} \right) + v_{c/\phi} \times v_{p/0}$$

$v_{p/c} + \omega_{b/\phi} \times r_{p/c}$
 $\phi \rightarrow \text{N.E.}$
 moving

$$= \frac{d\phi}{dt} \left[r_{p/c} \times \left(v_{c/\phi} + \omega_{b/\phi} \times r_{p/c} \right) \right] + v_{c/\phi} \times \left(\omega_{b/\phi} \times r_{p/c} \right)$$

$$\int_V \left(r_{p/c} \times \frac{d\phi}{dt} v_{p/\phi} \right) \rho dV = \int_V \frac{d\phi}{dt} [\star] \rho dV + \int_V v_{c/0} \times (\omega_{b/\phi} \times r_{p/c}) \rho dV$$

$$= \frac{d\phi}{dt} \int_V r_{p/c} \times \left[v_{c/0} + \omega_{b/\phi} \times r_{p/c} \right] \rho dV + v_{c/\phi} \times \left[\omega_{b/\phi} \times \int_V r_{p/c} \rho dV \right]$$

in right from def
 of const of mass
 and so always = 0

$$= \frac{d\phi}{dt} \int_V r_{p/c} \times \underbrace{v_{c/\phi}}_{\text{Not depending on } p} \rho dV + \frac{d\phi}{dt} \int_V r_{p/c} \times \left[\omega_{b/\phi} \times r_{p/c} \right] \rho dV$$

$$= \frac{d\phi}{dt} \left\{ \left(\int_V r_{p/c} \rho dV \right) \times v_{c/0} \right\} + \frac{d\phi}{dt} I_c(\omega_{b/\phi})$$

$$\frac{d\phi}{dt} I_c(\omega_{b/\phi}) = M^{\text{ext}}$$

\hookrightarrow resultant of external moments.

$$\frac{d\phi}{dt} I_c(\omega_{b/\phi}) = \frac{dI_c}{dt} I_c(\omega_{b/\phi}) + \omega_{b/\phi} \times I_c(\omega_{b/0})$$

$$\frac{dI_c}{dt} I_c(\omega_{b/\phi}) = \frac{dI_c}{dt} \int_V r_{p/c} \times \left(\omega_{b/\phi} \times r_{p/c} \right) \rho dV$$

$$= \int_V \frac{dI_c}{dt} \left[\underbrace{r_{p/c} \times}_{\text{always } \phi \text{ in } b} \left(\omega_{b/\phi} \times r_{p/c} \right) \rho dV \right]$$

$$= \int_V \rho_{PC} \times \left[\left(\frac{db}{dt} \omega_{\gamma\phi} \right) \times \mathbf{r}_{PC} \right] \rho dV = \mathbf{I}_C (\dot{\omega}_{\gamma\phi})$$

but $\boxed{\frac{db}{dt} \omega_{\gamma\phi} = \frac{d\theta}{dt} \omega_{\gamma\phi}}$ so $\dot{\omega}_{\gamma\phi} \triangleq \dot{\omega}_{\gamma\phi}$

Falcon equation of motion

$$\mathbf{I}_C (\dot{\omega}_{\gamma\phi}) + \omega_{\gamma\phi} \times \mathbf{I}_C (\omega_0) = M^{\text{ext}}$$

We need to numerically compute these quantities so let's see in algebraic form

Assume to project this equation in frame c :

You get $\boxed{{}^b I_C {}^b \dot{\omega}_{\gamma\phi} + [{}^b \omega_{\gamma\phi} \times]} + {}^b I_C {}^b \omega_{\gamma\phi} = {}^b M^{\text{ext}}$

\downarrow
unbroken
 3×3 time varying matrix
(depends on body attitude)

First order diff equation.

We need to couple this equation with the stay down:

$$\boxed{{}^b \dot{R} = [{}^b \omega_{\gamma\phi} \times] {}^b R}$$

can be computed offline.

$$\left\{ \begin{array}{l} \boxed{{}^b I_C {}^b \dot{\omega}_{\gamma\phi} + [{}^b \omega_{\gamma\phi} \times]} {}^b I_C {}^b \omega_{\gamma\phi} = {}^b M^{\text{ext}}} \\ \text{comes out from ccd} \end{array} \right. \quad \begin{array}{l} \text{same as the one} \\ \text{before, but computationally} \\ \text{much easier.} \end{array}$$

VERY
NICE

NOTE

$$\boxed{\begin{aligned} {}^2 A {}^b m &= [A(m)] \\ {}^b A {}^b m &= [A(m)] \end{aligned}}$$

$${}^2 R {}^b A {}^b m = \underbrace{{}^2 R}_{{}^2 A} \underbrace{{}^b A}_{{}^b m} {}^b m$$

$${}^b\dot{w}_b = {}^bI_c^{-1} \left[[{}^b\omega_{b/x}] {}^bI_c {}^b\omega_{r/\phi} + {}^0M^{ext} \right]$$

You have to check

that the vectors
are orthogonal
 $\hat{v}_{R_b} = col_2({}^bR[{}^b\omega_{b/x}])$

(is not a function of linear or angular velocity)

$${}^b\dot{R} = {}^0R \left[{}^b\omega_{r/\phi} x \right] \Rightarrow \begin{matrix} \hat{v}_{R_b} = col_2() \\ \hat{v}_{R_b} = col_3() \end{matrix} \quad (\text{You could also use quaternions})$$

These are non linear differential equations

Conceptually there are \gg first order diff equation because you need 3 parameters for orientation.

Angular velocity and orientation can be done with roll angles.

Summarize Newton-Euler for the rigid body,

They are two sets of equations describing translation of the body looking at the center of mass.

$$m \frac{d}{dt} v_{c/g} = F^{ext}$$

$$I_c(\dot{\omega}_{r/\phi}) + \omega_{r/\phi} I_c(\omega_{r/\phi}) = M^{ext}$$

} Forward dynamics

REMARK



All the external forces acting on the bottle are my

$$F^{ext} = mg$$

Now let's see for ang. vel.

integrate for every point

$$m_{p/C}^{ext} \rightarrow M^{ext} = \int_V r_{p/C} \times g dm = \int_V (r_{p/C} \times g) \rho dV$$

$$m_{p/C}^{ext} = r_{p/C} \lambda f_p^{ext}$$

$$\hookrightarrow = \left[\int_V r_{p/C} \rho dV \right] \times g = \phi$$

The quantity is not generally moment.

The inertia matrix can be chosen in smart way:

REMARK



$$\text{Assume } \hookrightarrow \text{ such that } \hookrightarrow I_c = \begin{bmatrix} I_1 & I_2 & \phi \\ \phi & I_3 \end{bmatrix}$$

$I_{ii} \geq 0$ for rigid bodies. i.e. (1..3)

We have 3 conditions:

① General structure $I_1 > I_2 > I_3$
 rotations around the intermediate axis are unstable

Rotation around the largest moment of inertia is stable, and basically asymptotically stable.

② Spherical structure when $I_1 = I_2 = I_3$ Any frame is a principle inertia Note that geometrically could be ∇^2 spheroid if made of different density material

③ Gyroscopic structure $I_1 \neq I_2 \neq I_3$

$$I_C = \begin{bmatrix} I_1 & \phi \\ \phi & I_3 \end{bmatrix} \quad \omega_{\text{gyr}} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad \text{scalar functions. } M^{\text{ext}} = \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix}$$

$$I_C \dot{\omega}_{\text{gyr}} + \begin{bmatrix} \omega_y \\ \omega_z \\ \omega_x \end{bmatrix} = M^{\text{ext}}$$

$$\begin{bmatrix} I_1 & \phi \\ \phi & I_3 \end{bmatrix} \begin{bmatrix} \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{bmatrix} + \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & \phi & -\omega_x \\ -\omega_y & \omega_x & \phi \end{bmatrix} \begin{bmatrix} I_1 & \phi \\ \phi & I_3 \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = M^{\text{ext}}$$

$$I_1 \dot{\omega}_x + [-\omega_z I_2 \omega_y + \omega_y I_3 \omega_z] = M_x^{\text{ext}}$$

$$I_2 \dot{\omega}_y + [\omega_z I_1 \omega_x - \omega_x I_3 \omega_z] = M_y^{\text{ext}}$$

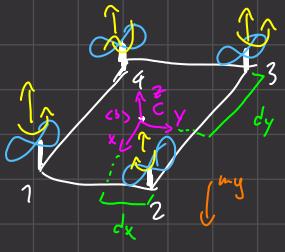
$$I_3 \dot{\omega}_z + [-\omega_y I_1 \omega_x + \omega_x I_2 \omega_y] = M_z^{\text{ext}}$$

$$\begin{cases} I_1 \dot{\omega}_x + (I_3 - I_2) \omega_z \omega_y = M_x^{\text{ext}} \\ I_2 \dot{\omega}_y + (I_3 - I_1) \omega_x \omega_z = M_y^{\text{ext}} \\ I_3 \dot{\omega}_z + (I_1 - I_2) \omega_x \omega_y = M_z^{\text{ext}} \end{cases}$$

The signs depend on the type.

11/03/24

Drone dynamics



$$\text{Mass} = m > 0 \\ I_x = I_y \neq I_z > 0$$



$${}^b I_c = \begin{bmatrix} I_x & \emptyset \\ \emptyset & I_y \\ \emptyset & I_z \end{bmatrix}$$

\dot{q}_i = ang vel of the propellers.

$$f_i = \hat{k}_b \gamma_p \dot{q}_i$$

constant related to
 type of the propeller
 forces generated by motor

$${}^b \hat{k}_b = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$F^{ext} = mg + f_1 + f_2 + f_3 + f_4$$

$$\Phi F^{ext} = \underbrace{m \Phi g}_{\text{wing}} + {}^b R \left[{}^b K_b \gamma_p \dot{q}_1 + {}^b K_b \gamma_p \dot{q}_2 + {}^b K_b \gamma_p \dot{q}_3 + {}^b K_b \gamma_p \dot{q}_4 \right] =$$

$$\gamma_p \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} = \Phi \dot{q}$$

$${}^b K_b = \begin{bmatrix} \Phi k_b \\ j_b \\ i_b \end{bmatrix}$$

$$\Phi R \begin{bmatrix} 0 \\ b \\ 1 \end{bmatrix} = \begin{bmatrix} \Phi i_b \\ \Phi j_b \\ \Phi k_b \end{bmatrix} \begin{bmatrix} 0 \\ b \\ 1 \end{bmatrix} = \Phi K_b$$

$$N.R.$$

$$m_i = \tau N_c \times f_i$$

$${}^b R_{i,c} = \begin{bmatrix} \frac{dx}{d\theta} \\ \frac{dy}{d\theta} \\ 0 \end{bmatrix}$$

$${}^b f_i = \begin{bmatrix} 0 \\ 0 \\ x \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + \gamma_p \Phi K_b \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} \dot{q} = \Phi F^{ext}$$

$${}^b m_i = \begin{bmatrix} 0 & 0 & -dx \\ 0 & 0 & -dy \\ dx & dy & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ f_i \end{bmatrix} =$$

$$= \begin{bmatrix} -dx \\ -dy \\ \Phi \end{bmatrix} f_i$$

$$M^{ext} = (-\tau_1 - \tau_2 - \tau_3 - \tau_4) + \hat{i}_b (\underbrace{f_2 + f_3 - f_1 - f_4}_{\text{proportional}}) dx + \hat{j}_b (\underbrace{f_3 + f_4 - f_1 - f_2}_{\text{proportional}}) dy$$

$${}^b M^{ext} = -\gamma_m \begin{bmatrix} 0 \\ 0 \end{bmatrix} \underbrace{\begin{bmatrix} 1 & -1 & 1 & -1 \end{bmatrix}}_2 \dot{q}_i + \gamma_p \begin{bmatrix} 0 \\ 0 \end{bmatrix} \underbrace{\begin{bmatrix} -1 & 1 & 1 & -1 \end{bmatrix}}_3 \dot{q} + \gamma_p \begin{bmatrix} 0 \\ 0 \end{bmatrix} \underbrace{\begin{bmatrix} -1 & -1 & 1 & 1 \end{bmatrix}}_4 \dot{q}$$

remove ${}^b K_b$

The motor has to generate torque to move the propeller. The 3rd Newton law can be applied to torque i.e. the motor generates torque and the propeller generates torque on the motor.

We can assume that τ_i is proportional to the spinning of the propeller.

$$\boxed{\tau_i = \hat{k}_b \gamma_m \dot{q}_i}$$

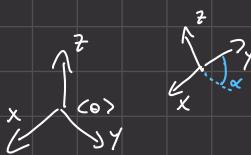
If you put these on top of each other the matrix is non singular so you can choose a speed for the motors that produce correct thrust and torque. This is invertible NB.

$$\begin{aligned} {}^b M_{ext} &= \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} = \left\{ \begin{array}{l} M_x = \gamma_p [-1 \ 1 \ -1] \dot{q} dx \\ M_y = \gamma_p [-1 \ -1 \ 1] \dot{q} dy \\ M_z = \gamma_m [1 \ -1 \ 1] \dot{q} \end{array} \right\} \text{ set of linear equation coupled with } \textcircled{1} \end{aligned}$$

Calculate moments w.r.t the body cb and forces w.r.t the origin.

$\boxed{{}^b R = {}^b R^+ [{}^b \omega_b \phi \ x]}$

Remember the step down.
By integrating this you get ${}^b R$



$${}^b R = {}^b R^+ {}^1 R {}^2 R {}^3 R = \textcircled{1}$$

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

$${}^2 R = R_y(\beta) = \begin{bmatrix} c\beta & 0 & s\beta \\ 0 & 1 & 0 \\ -s\beta & 0 & c\beta \end{bmatrix}$$

$${}^3 R = R_z(\gamma) = \begin{bmatrix} c\gamma & -s\gamma & 0 \\ s\gamma & c\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\textcircled{1} = \begin{bmatrix} 1 & 0 & s\beta \\ 0 & 1 & -s\alpha c\beta \\ 0 & s\alpha & c\alpha c\beta \end{bmatrix}$$

total twist

$$F_x = \gamma_p s\beta \overline{\mu_T}$$

$$F_y = \gamma_p (-s\alpha c\beta) \overline{\mu_T}$$

$$F_z = \gamma_p (c\alpha c\beta) \overline{\mu_T}$$

Dynamic equations

$$m \frac{d\vec{v}}{dt} \approx \vec{c}_\phi = \vec{F}_{ext}$$

$$\left\{ \begin{array}{l} m \ddot{x}(t) = F_x = \gamma_p (\sin \beta) \mu^\top \\ m \ddot{y}(t) = F_y = \gamma_p (\cos \beta) \mu^\top \\ m \ddot{z}(t) = F_z = -mg + \gamma_p (\cos \beta) \mu^\top \end{array} \right.$$

μ^\top is always different than μ so
the change force changes the orientation of the
drone or whatever.

$$^b I_c \dot{\omega}_b + [^b \omega_b] \times ^b I_c ^b \omega_b = ^b M_{ext}$$

$$I_x \dot{\omega}_x + (I_2 - I_y) \omega_y \omega_z = M_x$$

$$I_y \dot{\omega}_y + (I_x - I_z) \omega_x \omega_z = M_y$$

$$I_z \dot{\omega}_z = M_z$$

$$^b \omega_{\psi\phi} = \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix}$$

Compute $F_x, F_y, F_z, M_x, M_y, M_z$ for a steady drone and compute μ^\top

so from here $\mu^\top = \frac{1}{\gamma_p} my$

$$\left\{ \begin{array}{l} \alpha = \rho \\ \beta = \theta \\ m \ddot{z}(u) = g \end{array} \right.$$

Another exercise: Velocity in euler form

$$^b R = R_x(\alpha) R_y(\beta) R_z(\gamma)$$

Represent angular velocity of two frames with $(\dot{\alpha}, \dot{\beta}, \dot{\gamma})$

$$\omega_{1\phi} = \hat{i}_1 \cdot \dot{\alpha} = \hat{i}_\phi \cdot \dot{\alpha}$$

$$\omega_{21} = \hat{j}_2 \cdot \dot{\beta} = \hat{j}_1 \cdot \dot{\beta}$$

$$\omega_{32} = \hat{k}_3 \cdot \dot{\gamma} = \hat{k}_2 \cdot \dot{\gamma}$$

$$\omega_{\psi\phi} \cdot \omega_{30} = \omega_{1\phi} + \omega_{21} + \omega_{32} = i_0 \dot{\alpha} + j_1 \dot{\beta} + k_2 \dot{\gamma}$$

$$^b \omega_{\psi\phi} = \left[{}^b J^A (\alpha, \beta, \gamma) \right] \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix} \quad \boxed{\text{Do the calculation}}$$

J^A is Jacobian

now fix (angular part)

If you compute the derivative of J you get:

$${}^b \dot{\omega}_{\psi\phi} = {}^b J^A \begin{bmatrix} \ddot{\alpha} \\ \ddot{\beta} \\ \ddot{\gamma} \end{bmatrix} + {}^A J^A \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix} \quad \rightarrow \text{YOU compute this term}$$

If you use the persons found in the Euler formula you get 3 diff equations that allow you to not use the integration of the step function.

$$\int_C \underline{\omega}_b + [\underline{\omega}_{\psi}] \times \int_C \underline{y}_{\text{Cent}} = \underline{M}^{\text{ext}}$$

|S

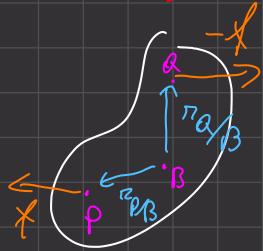
$$\int_C^A \underline{I}_C \underline{J}^A \begin{pmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{pmatrix} + \int_C^A \underline{B}(\alpha, \beta, \gamma, \dot{\alpha}, \dot{\beta}, \dot{\gamma}) = \int_C^A \underline{M}^{\text{ext}}$$

↓ ↓
inertial

73/09/29

Basic Remarks

P and Q are arbitrary



A pure moment can be expressed as a couple of forces

Parallel \Rightarrow equal magnitude, but opposite, they act on different lines

Remark 1

$\underline{f}_{\text{tot}} = f \cdot f^{\perp}$ \rightarrow the object is not subject to forces

Forces

$$M_{\text{tot}} = r_{P/B} \times f + r_{Q/B} \times (-f) = [r_{P/B} - r_{Q/B}] \times f = \underline{r}_{P/Q} \times f$$

M.B invariant w.r.t the choice of P

$$F_{\text{tot}} = \sum_{i=1}^N f_i = f_1, f_2, \dots, f_N$$

\hookrightarrow EXT forces (because int \Rightarrow because of rigid body)

(You can change the application point)

$$M_{\text{tot}} = \sum_{i=1}^N m_i + \sum_{i=1}^N r_{P/B} \times f_i$$

Remark 2



Tell force F acting on body :

center of mass express it wrt. C.

In this case $f_{\text{tot}} = F$

$$M_{\text{tot}} = r_{P/c} \times F$$

Project this equation in the reference frame you like:

I choose frame (c)

$$\dot{\varphi}_{M_{tot}} = \mathbb{1}_3 \cdot 0 + [\sigma_{\alpha c k}] f$$

$$\dot{\varphi}_f_{tot} = 0 + \mathbb{1}_3 f$$

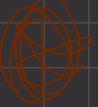
Adding on now

$$\dot{\varphi}_{M_{tot}} = \mathbb{1}_3 m + [\sigma_{\alpha c k}] f$$

$$\dot{\varphi}_{F_{tot}} = 0 + \mathbb{1}_3 f$$

In matrix form

$$\begin{bmatrix} \dot{\varphi}_{M_{tot}} \\ \dot{\varphi}_{F_{tot}} \end{bmatrix} = \begin{bmatrix} \mathbb{1}_3 & [\sigma_{\alpha c k}] \\ \mathbb{0}_3 & \mathbb{1}_3 \end{bmatrix} \begin{bmatrix} \dot{\varphi}_m \\ f \end{bmatrix}$$



Always invertible MRE forces and torques

Result

The jacobian is used to transform joint velocities (link end) to

$$\begin{bmatrix} \dot{\varphi}_{w_{eq}} \\ \dot{\varphi}_{v_{eq}} \end{bmatrix} = \dot{\varphi}_{J_{eq}} q, \quad \dot{\varphi}_{J_{eq}} \in \mathbb{R}^{6 \times n}$$

Basic version: $\dot{\varphi}_{J_{eq}}$

$$\begin{bmatrix} \dot{\varphi}_{w_{eq}} \\ \dot{\varphi}_{v_{eq}} \end{bmatrix} = \dot{\varphi}_{J_{eq}} q$$

The relationship between the previous formulas:

$$\begin{cases} \omega_{eq} = \omega_{eq} \rightarrow no \text{ rotation} \\ v_{eq} = v_{eq} + \omega_{eq} \times r_{eq} \end{cases}$$

So we have:

$$\begin{bmatrix} \omega_{\text{exp}} \\ \theta_{\text{verexp}} \end{bmatrix} = \begin{bmatrix} \beta_3 & \phi_3 \\ [\beta_{\text{exp}}]^\top & \beta_3 \end{bmatrix} \begin{bmatrix} \theta_{\text{verexp}} \\ \theta_{\text{verphi}} \end{bmatrix}$$

$$= \int_{\text{exp}} \begin{bmatrix} \dot{\phi}_{\text{verexp}} \\ \dot{\phi}_{\text{verphi}} \end{bmatrix}$$

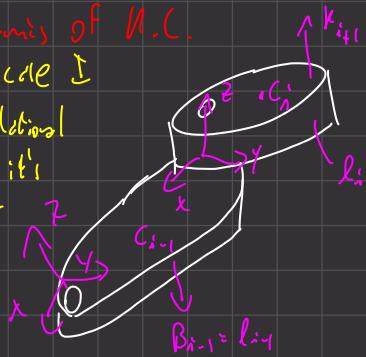
New trajectory $\geq t$



$$\begin{bmatrix} \phi_{M_{\text{rot}}} \\ \phi_{F_{\text{rot}}} \end{bmatrix} = \begin{bmatrix} \beta_3 & [\phi_{D_{\text{ver}}}, \dot{x}] \\ \phi_3 & \beta_3 \end{bmatrix} \begin{bmatrix} \theta_m \\ \theta_f \end{bmatrix}$$

Dynamics of N.C.

In this code I consider rotational joints but it's the same for trans. over



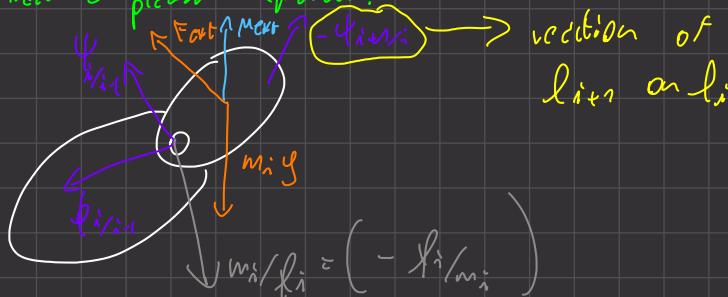
Part of open rheumatic

Problem: Motion of this bodies when it's subjected to forces

- 1) Specify the type of forces and torques that come from internal links,
- 2) N bodies \Rightarrow $6n$ differential equations.
- 3) Interconnectivity: constraints \rightarrow there are joints that limit the motion,

We want a method to consider less equations and come to our purpose.

Consider a planar sketch:

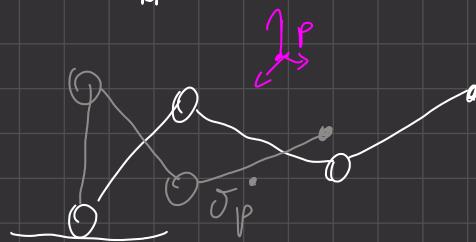


reaction of tongue from link on link

$$m_i/f_i = \left(-l_i/m_i \right)$$

The internal forces are now important because they keep everything together.

Compute $\begin{cases} F_{\text{int}} \\ M_{\text{int}} \end{cases}$



What happens assigning them arbitrary variation to each joint variables?

$\delta p \rightarrow ???$ TODO

$\delta \theta \rightarrow$ virtual displacement (rotation)

$\delta q_i \rightarrow$ virtual joint displacement

For open kinematic chains the virtual joint displacements are arbitrary and independent

If I consider closed loop NC:



You can't choose all the angles as you want.

How do I put in relation δp and $\delta \theta$?

I consider generic link i:

$$\phi_{w_{1/\phi}} = \phi_{q_{1/\phi}} \cdot \dot{q}$$

$$\phi_{v_{1/\phi}} = \phi_{f_{p_{1/\phi}}} \cdot \dot{q}$$

Very tiny interval, calling it δt : $\dot{q} = \frac{dq}{dt} \approx \frac{\delta q}{\delta t}$

And I get:

$$\phi_{w_{1/\phi}} \delta t = \delta \theta = \phi_{f_{1/\phi}} \delta q$$

(\Rightarrow my angle integrated over an infinitesimal amount of time)

$$\delta r_{p_{1/\phi}} \delta t = \delta g_{p_{1/\phi}} = \phi_{f_{p_{1/\phi}}} \delta q$$

Virtual Work

Remark:

Work of a force is $w = f \cdot \Delta p$

We consider 2 types of forces with association with

$$\begin{cases} \text{forces} & \delta A = f \cdot \delta p \\ \text{torques} & \delta A = m \cdot \delta \theta \end{cases}$$



Torques:

$$\delta A = m_i \delta \theta_i = \varphi_{m_i}^T \underbrace{\delta f_{B_i/\varphi}}_{\text{Point } B_i} \quad \delta q = \delta q^T (\delta f_{B_i/\varphi})^T \varphi_{m_i}$$

Point B_i : Recalcul the scalar product in a different way

Forces:

$$\delta A_i = f_i \cdot \delta p_i = \varphi_{f_i}^T \delta f_{B_i/\varphi} \quad \delta q = \delta q^T (\delta f_{B_i/\varphi})^T \varphi_{f_i}$$

Remember set of 2 dim vectors with φ and 1 unit vect. or const.

$$\delta f_{B_i/\varphi} = \left[\varphi_{f_1}^A : \varphi_{f_2}^A ; \dots ; \varphi_{f_i}^A : \varphi_{f_{i+1}}^A ; \dots ; \varphi_{f_n}^A \right]$$

$$f_i^A = \begin{cases} h_i & \Gamma_i = R \\ \varphi & \Gamma_i = P \end{cases}$$

$$\text{So: } \delta f_{B_i/\varphi}^A \cdot m_i = \begin{bmatrix} f_1^A \cdot m_1 \\ f_2^A \cdot m_2 \\ \vdots \\ f_i^A \cdot m_i \\ \vdots \\ \varphi \end{bmatrix}$$

$$f_i^A \cdot m_i = \begin{cases} k l \cdot m_i & \Gamma_i = R \\ \varphi & \Gamma_i = P \end{cases}$$

Some consideration for linear point.

$$\delta f_{B_i/\varphi}^L = \left[\varphi_{f_{11}}^L : \varphi_{f_{12}}^L ; \dots ; \varphi_{f_{i1}}^L : \varphi_{f_{i2}}^L ; \dots ; \varphi_{f_{n1}}^L : \varphi_{f_{n2}}^L \right]$$

$$\mathcal{J}_{\beta_i/\ell}^L = \begin{cases} (K_e \times r_{\beta_i/\ell}) & \Gamma_i = R \\ K\ell & \Gamma_i = P \end{cases}$$

$\text{So: } \mathcal{J}_{\beta_i/\ell}^L \cdot f_i = \begin{pmatrix} \mathcal{J}_{\beta_i/\ell}^L & f_i \\ \vdots & \vdots \\ \vdots & \vdots \end{pmatrix} \leftarrow \ell$

$$\mathcal{J}_{\alpha_i/\ell}^L \cdot f_i = \begin{cases} (K_e \times r_{\alpha_i/\ell}) f_i & \Gamma_i = R \\ K\ell \times f_i & \Gamma_i = P \end{cases}$$

$$(K_e \times r_{\alpha_i/\ell}) f_i = \underbrace{(r_{\alpha_i/\ell} \times f_i) \cdot K_e}_{\text{cyclic permutation}}$$

torque generated by forces and projected on ℓ

Let's consider this set of forces and torques:

M_i :

F_{C_i}

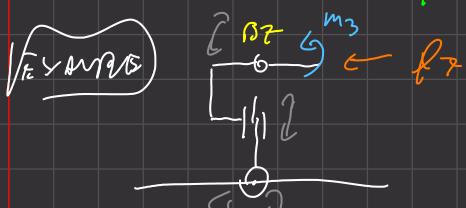
$$\sum M^{(\text{tot})} = \sum_{i=1}^m \sum M_{C_i}^{(u)} + \sum M_{C_i}^{(f)} = \sum_{i=1}^m \sum q^T \begin{bmatrix} \mathcal{J}_{C_i/\ell}^A \\ \mathcal{J}_{C_i/\ell}^L \end{bmatrix} M_i + \sum q^T \begin{bmatrix} \mathcal{J}_{C_i/\ell}^A \\ \mathcal{J}_{C_i/\ell}^L \end{bmatrix} \bar{F}_{C_i}$$

$\sum q$ does not depend on i :

$$\sum q^T \sum_{i=1}^m \begin{bmatrix} \mathcal{J}_{C_i/\ell}^A & ; & \mathcal{J}_{C_i/\ell}^L \end{bmatrix} \begin{bmatrix} M_i \\ \bar{F}_{C_i} \end{bmatrix} \quad (\text{cancel term})$$

$\sum M^{(\text{tot})}$ is the same for M_1, \dots and you get the
 $\sum M_i$ with all following formulae:

$$\sum M^{(u)} = \sum q^T \sum_{i=1}^m \mathcal{J}_{C_i/\ell}^A \begin{bmatrix} M_i \\ \bar{F}_{C_i} \end{bmatrix}$$



$$\mathcal{J}_{C_i/\ell}^A = \begin{bmatrix} qM_i & q\bar{F}_{C_i} \end{bmatrix}$$

$$\phi \int_{B_3/\phi}^{A^T} m_3 = \begin{bmatrix} K_1 \cdot m_3 \\ \emptyset \\ K_3 \cdot m_3 \end{bmatrix}$$

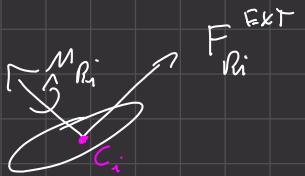
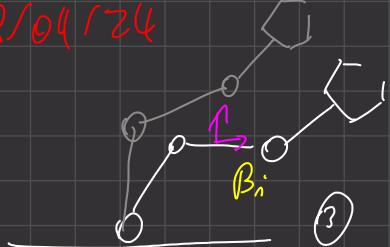
$$\text{if : } m_3 = \begin{bmatrix} \emptyset \\ \emptyset \\ \emptyset \end{bmatrix} \quad K_1 \cdot m_3 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \emptyset$$

$$K_3 \cdot m_3 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \emptyset$$

$$\phi \int_{B_3/\phi}^{L^T} f_3 = \begin{cases} (k_1 \times n_{n_1/1}) \cdot f_3 \\ k_2 \times f_3 \\ (k_3 \times n_{n_3/3}) \cdot f_3 \end{cases} \rightarrow \begin{array}{l} \text{Dependency of direction} \\ \text{of } f_3 \text{ (by the sign)} \end{array}$$

// so the result is \emptyset

18/04/24



$\sum q_i$

$\sum \beta_i$

$\sum \Theta_i$

$\sum \beta_i = \phi \int_{B_i/\phi}^{L_i} \sum q$

$\sum \Theta_i = \phi \int_{A/\phi}^{A^T} \sum q$

Sum for each body

$$\sum L_i = \left\{ F_{C_i} \cdot \sum C_i = \sum q^T \left(\phi \int_{C_i/\phi}^L \cdot F_{C_i}^{\text{ext}} \right) \right\} \text{ N.B.}$$

This is very important

$$(M_{C_i} \sum \Theta_i = \sum q^T \left(\phi \int_{C_i/\phi}^{A^T} \cdot M_{C_i} \right))$$

It's the projection of the force acting on i and projected on the joint space. This is a geometric projection.

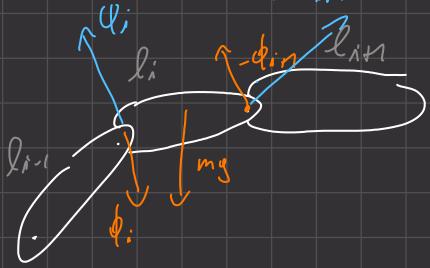
$$\sum L = \sum_{i=1}^n \sum q^T \phi \int_{C_i/\phi}^L \left(\frac{M_{C_i}}{F_{C_i}} \right) = \sum q^T \left[\sum_{i=1}^n \phi \int_{C_i/\phi}^T \left(\frac{M_{C_i}}{F_{C_i}} \right) \right] \rightarrow \begin{array}{l} \text{transformation forces} \\ \text{acting on body into} \\ \text{joint coordinates} \end{array}$$

Wrench
n-dimensional vectors

$$\phi \int_{C_i/\phi}^{A^T} = \sum_{C_i/\phi} \phi \int_{C_i/\phi}$$

$$6 \left[\underbrace{\begin{bmatrix} \mathbf{J} \end{bmatrix}}_n \right]$$

What happens if we connect many bodies:



The external forces are those who are not depending on the interconnection of the chains. mg will always be felt on the robot arm.

\vec{l}_{int} is producing forces acting on i

$$\underbrace{\phi_i + (\phi_{int}) = 0}_{\text{they cancel}} \quad \underbrace{\psi_i + (\psi_{int}) = 0}_{\text{they cancel}}$$

there are also actuation forces and torques

There exist actuation torques m_i which are producing the relative motion between body i and $i-1$.

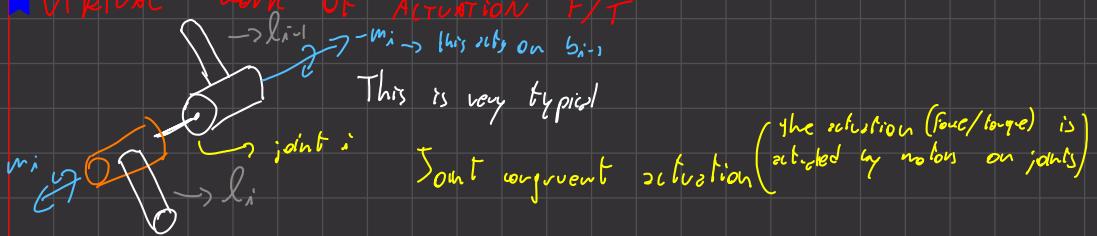
Our software will have to send this numbers to the motors. The magnitude of the vector is what we send.

There will also be a m_{int} counteracting on the joints. (Extension of 3rd Newton law)

As an external force there is also the gravity.

Note it's important to compute the virtual task for every interconnection of every body on the others.

VIRTUAL WORK OF ACTUATION F/T



We can think of the connection of two links as a rot motor. One of the critical things in what is the varying of the cable.

There is also actuation expressed by tendons and pulleys.



It could happen that the motor is very bad in the chain.

If we have to handle this situation the action of the forces is as if they are external forces.

Let's split between rotational and translational joints:

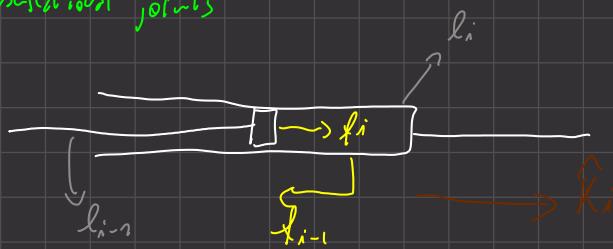
Rotational joints

$$\delta L_i^{(act)} = m_i \cdot \delta \theta_i - m_i \cdot \delta \theta_{i-1} = \underbrace{m_i [\delta \theta_i - \delta \theta_{i-1}]}_{K_i \cdot \delta q_i} = (\underbrace{m_i K_i}_{\text{scalar}}) \delta q_i$$

this can be reduced to a scalar

geometrically this is the projection of the rotation on the rot axis so this is a scalar.

Translational joints



$$\delta L_i^{(act)} = f_i \cdot \delta P_i - f_i \cdot \delta Q_{i-1} = f_i \cdot [\underbrace{\delta P_i - \delta Q_{i-1}}_{K_i \cdot \delta q_i}] = (f_i \cdot \hat{R}_i) \delta q_i$$

$K_i \cdot \delta q_i$ scalar f_i

projection of the force
on the axis of rotation.

\hat{r}_i is the generalized activation force or torque!

$$\hat{r}_i = \begin{cases} m_i \cdot \hat{R}_i & m_i \quad \Gamma_i = R \\ f_i \cdot \hat{R}_i & f_i \quad \Gamma_i = p \end{cases}$$

N.B.

$$\delta L = \sum_{i=1}^n \delta L_i^{(act)} = \underbrace{\sum_{i=1}^n \hat{r}_i \delta q_i}_{\delta q^\top \hat{T}}$$

this is a scalar product
between two vectors

$$\hat{T} = \begin{bmatrix} \hat{r}_1 \\ \vdots \\ \hat{r}_n \end{bmatrix} \rightarrow \text{general f/t activation vector}$$

If all the components of this vector can be generated by independent motors the robot is fully actuated.

If there are 6 DOF rotations then it is under actuated

The drone is under actuated $\left\{ \begin{array}{l} 4 \text{ motors} \\ 6 \text{ parameters} \end{array} \right.$

Virtual work of motion F/T

Ideal joints

One for which the virtual work of the reaction force or torque is equal to 0.

$$\delta \lambda_i = \phi_i \cdot \delta P_i - \psi_i \cdot \delta Q_{i-1} = \phi_i \cdot \begin{cases} \phi & \Gamma = R \\ \tilde{R}_i \delta q_i & \Gamma = P \end{cases}$$

↓
The effect of reaction forces and torques at joint i.

$$\delta \lambda_i = \psi_i \cdot \delta \theta_i - \psi_i \cdot \delta \theta_{i-1} = \psi_i \cdot \begin{cases} \tilde{R}_i \delta g_i & \Gamma = R \\ \phi & \Gamma = P \end{cases}$$

$$\delta \lambda_i^{\text{tot}} = \begin{cases} (\psi_i \cdot \tilde{R}_i) \delta q_i & \Gamma_i = R \\ (\phi_i \cdot K_i) \delta q_i & \Gamma_i = P \end{cases}$$

$= \phi + \delta q_i$
If ideal this is true

Physically means that the reaction torque keeping the bodies together do not have any components on the rot axis and the same goes for translation motion. We are supposing the joints to be frictionless.

What happens if we have to consider friction? We will see later.

What's going to happen is the total reaction of forces and torques which appear in the LC:

$$\boxed{\delta \lambda_i = \sum_{i=1}^n \delta_i \lambda_i^{\text{tot}} = \phi}$$

Our joints are modelled as frictionless joints.

We have 3 classes of forces (extens, extens, reaction)

ROBOT Equilibrium equations.

We are doing the same reasoning of the law for a general chain.

$$\delta L^{(\text{ext})} = \delta q^T \sum_{i=1}^n \phi_j C_{ij} \phi_j \left(\frac{\delta M_{ci}}{\delta F_{ci}} \right) = \delta q^T \sum_{i=1}^n \phi_j \phi_j^T \left(\frac{\delta M_i}{\delta F_i} \right) \quad \text{← TODO place this}$$

$$\delta L^{(\text{ext})} = \delta q^T \uparrow$$

$$\delta L^{(R)} = \phi$$

The robot is in equilibrium with a mental reference frame when it has 0 angular vel. so:

$$\begin{cases} F_{c,i}^{(ext)} \text{ on each body on the chain} \\ F_{c,i} = \emptyset \\ M_{c,i} = \emptyset \end{cases}$$

Here I have (6xn equations)

The unknowns are
 { the velocities ($\dot{\theta}/t$)
 the reaction (F/t) }

We know that the only relevant values are the ones along the reaction/ F axis.

REMARKS

$$F_{c,i} \rightarrow F_i$$

$$M_{c,i} \rightarrow M_i = M_{c,i} + r_{c,i} \times F_i$$

$$\textcircled{1}_{\text{ws}} = \begin{cases} F_i = \emptyset & (6xn) \text{ equation} \\ M_i = \emptyset \end{cases} \quad \text{We want } 6 \text{xn equations} \rightarrow \text{to n equations}$$

$$\textcircled{2} \quad \sum P_i \cdot F_i + \sum \Theta_i \cdot M_i = \emptyset \quad \forall i = 1..n$$

We can now compute the total virtual work:

$$\textcircled{3} \quad \delta L = \sum_{i=1}^n \sum P_i \cdot F_i + \sum \Theta_i \cdot M_i = \emptyset \quad \text{single scalar equation which depends on virtual displacement which depends on absolute displacement.}$$

$$F_i = F_i^{(A)} + F_i^{(R)}$$

$$M_i = M_i^{(A)} + M_i^{(R)}$$

$$\textcircled{4} \quad \sum_{i=1}^n \left(\sum P_i F_i^{(A)} + \sum \Theta_i M_i^{(A)} \right) + \underbrace{\left(\sum_{i=1}^n \sum P_i F_i^{(R)} + \sum \Theta_i M_i^{(R)} \right)}_{\text{Virtual work due to all reaction forces and torques.}} = \emptyset \quad \text{Equilibrium}$$

$$\delta L^{(\text{ext})} = \emptyset \quad \text{if ideal}$$

We get to a point where reaction forces and torque disappear.

REMARK

$$F_i^{(A)} = F_i^{(\text{ext})} + f_i$$

\hookrightarrow reaction force

$$M_i^{(A)} = M_i^{(\text{ext})} + M_i$$

\hookrightarrow reaction torque

If you substitute this on the previous one you get some other equations.

20/04/24

Robot equilibrium condition

N.E eq at eq

$$\textcircled{1} \quad \begin{cases} M_i = \emptyset \\ F_i = \emptyset \end{cases} \quad \forall i \quad \begin{cases} M_i = \emptyset \quad \text{wrt point } P_i \text{ (simpler)} \\ F_i = \emptyset \end{cases}$$

Let's think about the ideas behind these formulae.

When you see what is a fixed position is because there are blocks. We are sending the torques to keep the robot in eq. condition.

$$\textcircled{2} \quad \delta L_i \xrightarrow{\text{virtual work}} = \delta \theta_i \cdot M_i + \delta p_i \cdot F_i = \emptyset \quad \text{virt work}$$

$$\textcircled{3} \quad \delta L^{\text{tot}} = \sum_{i=1}^n \delta L_i \quad \begin{pmatrix} \emptyset \delta \theta \\ \emptyset \delta p_i \end{pmatrix} = \emptyset J_i^T \delta q \quad \text{tot work},$$

$$\textcircled{4} \quad \delta L^{\text{tot}} = \delta q^T \cdot \left[\sum_{i=1}^n \emptyset J_i^T \begin{pmatrix} M_i \\ F_i \end{pmatrix} \right] = \emptyset$$

$$\textcircled{5} \quad \begin{cases} M_i = M_i^{(A)} + M_i^{(B)} \\ F_i = F_i^{(A)} + F_i^{(B)} \end{cases} \quad M_i^{(A)} = M_i^{\text{ext}} + M_i^{(A)} \xrightarrow{\text{actuation}} \\ F_i^{(A)} = F_i^{\text{ext}} + F_i^{(A)}$$

$$\textcircled{6} \quad \delta L^{\text{tot}} = \delta q^T \left[\sum_{i=1}^n \emptyset J_i^T \begin{pmatrix} M_i^{(A)} \\ F_i^{(A)} \end{pmatrix} \right] + \delta q^T \left[\sum_{i=1}^n \emptyset J_i^T \begin{pmatrix} M_i^{(B)} \\ F_i^{(B)} \end{pmatrix} \right]$$

$\underbrace{\quad}_{\substack{\text{Total virtual work of actuated} \\ \text{forces and torques}}} \quad \underbrace{\quad}_{\delta q^T = \emptyset}$

The assumption is always that joints are frictionless.

$$\textcircled{6} \quad \delta L^{\text{tot}} = \delta q^T \left[\sum_{i=1}^n \emptyset J_i^T \begin{pmatrix} M_i^{(\text{ext})} \\ F_i^{(\text{ext})} \end{pmatrix} \right] + \delta q^T \left[\sum_{i=0}^n \emptyset J_i^T \begin{pmatrix} M_i^{(A)} \\ F_i^{(A)} \end{pmatrix} \right]$$

$\underbrace{\quad}_{\delta L^{\text{tot}}_{\text{ext}}} \quad \underbrace{\quad}_{\delta L^{\text{tot}}_{\text{act}}}$

$\delta q^T \xrightarrow{\text{joint congruent activation}}$

If the robot is under-actuated some components of γ might be \emptyset .

$$\textcircled{7} \quad \delta L^{\text{tot}} = \delta q^T \left[\sum_{i=1}^n \emptyset J_i^T \begin{pmatrix} M_i^{\text{ext}} \\ F_i^{\text{ext}} \end{pmatrix} + \gamma \right] = \emptyset \quad \xrightarrow{\text{set of unactuated forces and torques at the joints. This is what the software generates. This is the free control vector of the robot.}}$$

LEMMA

if $\mathbf{z}^T \cdot \mathbf{x} = \emptyset \quad \forall x \Rightarrow z = \emptyset$

Proof

$x = [0 \ 0 \ z \ 0 \dots \emptyset] \leftarrow$ let's assume x in this way
 \downarrow
 i.e.

then $\mathbf{z}^T \cdot \mathbf{x} = z_i \underbrace{= \emptyset}_{\text{for the hypothesis}}$

$\forall i$

\uparrow

\sum is a scalar product of this kind.

If for any possible i the scalar product is of the term in bracket must be \emptyset .

All the components of the bracket must be \emptyset .

$$\mathbf{T} + \underbrace{\sum_{i=1}^n \emptyset \mathbf{f}_i^T}_{\text{Effect of forces and bodies}} \begin{pmatrix} M_i^{ext} \\ F_i^{ext} \end{pmatrix} = \emptyset \quad (\text{n equations})$$

Effect of forces and bodies
applied to every joint

This is basically a model because let's assume the robot $\left\{ \begin{array}{l} \text{still} \\ \omega_{i/\emptyset} = \emptyset \\ v_{i/\emptyset} = \emptyset \end{array} \right\}$ at time t/\emptyset

The question is what is the torque that I have to send to keep the robot still?

$$\text{We can specify } \mathbf{T}^{eq} = - \sum_{i=1}^n \emptyset \mathbf{f}_i^T \begin{pmatrix} M_i^{ext} \\ F_i^{ext} \end{pmatrix}$$

This is our software specification starting from \emptyset val

Now the question is how to get M_i^{ext} and F_i^{ext}

Remember

$$\mathbf{T} + \sum_{i=1}^n \emptyset \mathbf{f}_i^T \begin{pmatrix} M_i^{ext} \\ F_i^{ext} \end{pmatrix}$$

Let's check if this can be applied to a

$$\emptyset \mathbf{f}_{ci/\emptyset} = \emptyset \mathbf{f}_{ci/\emptyset} \emptyset \mathbf{f}_{i/\emptyset}$$

$$\emptyset \mathbf{f}_{ci/\emptyset} = \begin{bmatrix} \pi \\ \begin{bmatrix} \pi_{ci/\emptyset} \end{bmatrix}^T \end{bmatrix} \pi$$

better singular always
multiple motion

$$\phi \mathfrak{f}_{i,\phi} = \phi \sum_{c_i \neq i}^{-1} \phi \mathfrak{f}_{c_i, \phi}$$

$$\left(\rightarrow \begin{bmatrix} 1 & \phi \\ -[\mathfrak{f}_{c_i, i}]^T & 1 \end{bmatrix} = \begin{bmatrix} 1 & \phi \\ [\mathfrak{f}_{c_i, i}]^T & 1 \end{bmatrix} = \sum_i \frac{1}{c_i} \right)$$

REMARK

$$M = \begin{bmatrix} A & \phi \\ B & C \end{bmatrix} \quad \text{this is invertible only if } A \text{ and } C \text{ are invertible.}$$

$$\Rightarrow \begin{bmatrix} A & \phi \\ B & C \end{bmatrix} \begin{bmatrix} X & \phi \\ Y & Z \end{bmatrix} = 1$$

$$AX = 1 \Rightarrow X = A^{-1}$$

$$BX + CY = \phi$$

$$Y = -C^{-1} BX$$

$$CZ = 1 \Rightarrow Z = C^{-1}$$

$$Y = -C^{-1} BA^{-1}$$

Inductives for us.

Getting back to the computations:

$$\gamma + \sum_{i=1}^n \phi \mathfrak{f}_i + \begin{pmatrix} \phi M_i^{(\text{ext})} \\ \phi F_i^{(\text{ext})} \end{pmatrix} = \gamma + \sum_{i \in \phi} \left(\sum_{j \in c_i} \phi \mathfrak{f}_{j,i} \right)^T \begin{pmatrix} \phi M_i^{(\text{ext})} \\ \phi F_i^{(\text{ext})} \end{pmatrix} =$$

$$= \gamma + \sum_{i \in \phi} \phi \mathfrak{f}_i^T \sum_{j \in c_i} \begin{pmatrix} \phi M_j^{(\text{ext})} \\ \phi F_j^{(\text{ext})} \end{pmatrix}$$

this is the computation from point p. 1 to c.

$$\boxed{\begin{aligned} \phi_i &= \phi_{c_i} \\ m_{c_i} &= m_i + \sum_{j \in c_i} \phi_{j,i} \end{aligned}}$$

Always equivalent
and equal to ϕ

$$= \gamma + \sum_{i=1}^n \phi \mathfrak{f}_i^T \sum_{j \in c_i} \begin{pmatrix} \phi M_{j,i}^{(\text{ext})} \\ \phi F_{j,i}^{(\text{ext})} \end{pmatrix}$$

You can express forces and torques w.r.t. the CM on any other point

this is important because we need to take into account gravity.
 Note also that the force in the center of mass does not produce any torque.

Reformulation

$$F_{C_i}^{(ext)} \rightarrow F_{C_i}^{(ext)} + m_i \cdot g$$

$$\tau^{EQ} = - \sum_{i=1}^n \mathcal{J}_{C_i/\phi}^+ \left(M_{C_i}^{(ext)} \right) - \sum_{i=1}^n \mathcal{J}_{C_i/\phi}^+ \left(\begin{pmatrix} \phi \\ \delta_g \end{pmatrix} m_i \right)$$

$\underbrace{\quad\quad\quad}_{\text{vector sum of all external forces except gravity}}$

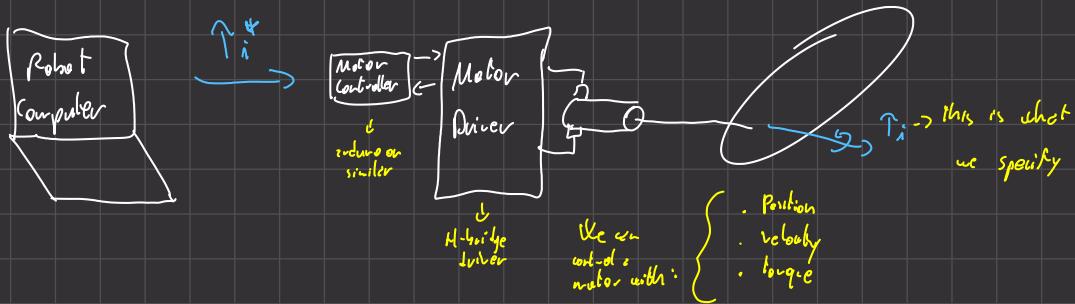
$$= - \sum_{i=1}^n \mathcal{J}_{C_i/\phi}^+ \left(M_{C_i}^{(ext)} \right) - \left(\sum_{i=1}^n m_i \mathcal{J}_{C_i/\phi}^+ \right) \left(\begin{pmatrix} \phi \\ \delta_g \end{pmatrix} \right)$$

gravity compensation term $\rightarrow \left(\begin{pmatrix} \phi \\ \delta_g \end{pmatrix} \right)$

function of the position of the robot.

REMARK

What is a robot?



- $\frac{T_{Ethernet}}{T_{100}/T_{16}/T_{200}}$

UDP

TCP/IP

- Ethercat

T 200



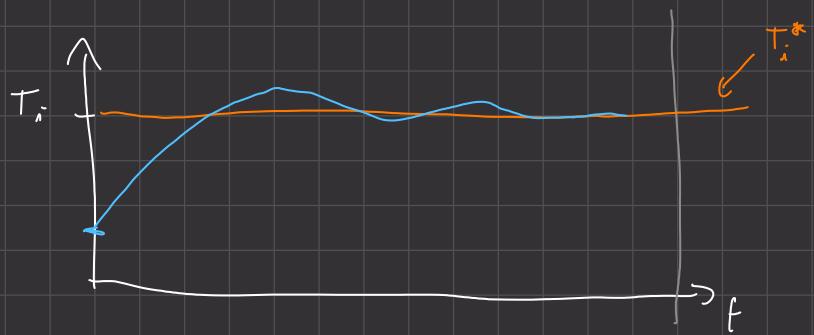
the jitter time is in nanoseconds

- CAN/CAN-FD

~100µs

~80µs

T_i can assume that $T_i \approx T_i^*$ most of the time



We want the current response to be much slower than the temporal deadline.

EXAMPLES of relevant cases

① No external forces and torques.

$$\dot{\gamma}^{EQ} = C(q) \rightarrow \text{gravity compensation term.}$$

② Robot holding something

minus because $C(q)$ has the minus



$$\dot{\gamma}^{EQ} = C(q) - \dot{\phi}_J^T \begin{pmatrix} \phi \\ y \end{pmatrix} m_b$$

$$\dot{\phi}_J^T = \left(\int_{E/B} \dot{\phi}_J^T \right)$$

Depending on what you grip you will change S and m

③

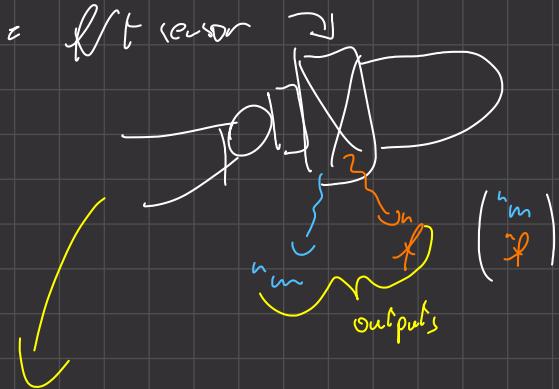


$$\dot{\gamma}^{EQ} = C(q) - \dot{\phi}_J^T \begin{pmatrix} \phi \\ y \end{pmatrix} m_b - \dot{\phi}_J^T \int_{E/B}^T \dot{\phi}_f^T \begin{pmatrix} \phi \\ F_i^{\text{ext}} \end{pmatrix}$$

We usually don't know position and magnitude of the force.

One way for estimating the force is using robot skin.

To measure forces and torques at the B.E. usually it's wanted



They use strain gauges or deformation sensors.

Here I don't move more when the force is applied with the skin [can move where].

03/06/24 Robot dynamics model

Result

$$\Rightarrow \begin{cases} M_{ci} = \emptyset \\ F_{ci} = \emptyset \end{cases} \quad \forall i \in \text{Equilibrium.}$$

$$\Rightarrow \tau + \sum_{i=1}^n \underbrace{\omega_i \times J_{ci,0}^T}_{\substack{\text{but all} \\ \text{stays} \\ \text{joint}}} \underbrace{\begin{pmatrix} M_{ci}^{\text{ext}} \\ F_{ci}^{\text{ext}} \end{pmatrix}}_{\substack{\text{takes into account} \\ \text{gravity}}} - ((\dot{q}) = \emptyset \quad \text{Behaviour of the robot at equilibrium.}$$

$$\tau = -\tilde{\tau}^{\text{ext}} + ((\dot{q}))$$

Lesson

Modeling of the centripetal effect

$$\textcircled{1} \quad \begin{cases} I_{ci}(\omega_{ci,0}) + \omega_{ci,0} \times J_{ci}(\omega_i) = M_{ci} \\ m_i \quad r_{ci,0} = F_{ci} \end{cases} \quad \forall i \in 1..n$$

These are equality: if what's on the right it's force then what's on the left is force.

$$\textcircled{2} \quad \begin{cases} I_{ci}(\omega_{ci,0}) + \omega_{ci,0} \times J_{ci}(\omega_i) - M_{ci} = \emptyset \\ m_i \quad r_{ci,0} = F_{ci}^0 \end{cases} \quad - F_{ci} = \emptyset$$

"physical forces and torques"

Dynamical equilibrium condition

Dynamic forces and torques

NOTE:

$$*(\begin{pmatrix} \omega_{c_i/\phi} \\ v_{c_i/\phi} \end{pmatrix}) = \phi J_{c_i/\phi}(q) \quad q$$

$$\left(\begin{array}{c} \dot{\omega}_{c_i/\phi} \\ \dot{v}_{c_i/\phi} \end{array} \right) = \phi \dot{J}_{c_i/\phi} \quad q + \phi \ddot{J}_{c_i/\phi} \quad q$$

"operation in q"

elements break in q · q

each entry is
the derivative of the
original

$$\boxed{\alpha} \xrightarrow{\frac{d}{dt}} \boxed{\frac{d}{dt} \alpha}$$

\downarrow
 $\frac{d}{dt} \alpha(q) = \frac{\partial \alpha}{\partial q} q \quad ?$

α element of J_{c_i}

NOTE

$$\phi \left[(\omega_{i/\phi} \times I_{c_i} (\omega_{i/\phi})) \right]$$

linear in ω

omega is a linear function of q from *

Quadratic function in q

Note

$$\begin{pmatrix} \delta \Theta_i \\ \delta C_i \end{pmatrix} = \phi J_{c_i/\phi} \delta q$$

(3) Compute the virtual work:

$$\delta L_{C_i} = \delta \Theta_i \cdot [M_{c_i}^D - M_{c_i}] + \delta C_i [F_{c_i}^D - F_{c_i}] = \phi \quad \text{Virtual work is } \phi$$

$$\delta L_{C_i} = \begin{pmatrix} \delta \Theta_i \\ \delta C_i \end{pmatrix}^\top \left\{ \begin{pmatrix} M_{c_i}^D \\ F_{c_i}^D \end{pmatrix} - \begin{pmatrix} M_{c_i} \\ F_{c_i} \end{pmatrix} \right\} = \phi$$

Work is always scalar.

Repeating step 2

$$\begin{pmatrix} \phi M_{c_i} \\ \phi F_{c_i} \end{pmatrix} = \begin{pmatrix} I_{c_i} & \omega_{i/\phi} \\ m_i & v_{c_i/\phi} \end{pmatrix} + \begin{pmatrix} \omega_{i/\phi} \times I_{c_i} (\omega_{i/\phi}) \\ \phi \end{pmatrix} =$$

$$\bar{A}_i(q) = \begin{pmatrix} \phi I_{c_i} & \phi \\ \phi & m_i \mathbb{1}_3 \end{pmatrix}$$

$$= \begin{pmatrix} \phi I_{c_i} & \phi \\ \phi & m_i \mathbb{1}_3 \end{pmatrix} \begin{pmatrix} \dot{\omega}_{i/\phi} \\ \dot{v}_{c_i/\phi} \end{pmatrix} + \begin{pmatrix} \omega_{i/\phi} \times I_{c_i} (\omega_{i/\phi}) \\ \phi \end{pmatrix}$$

6x6 invertible
function of q → $\bar{A}_i(q)$

$$\bar{A}_i(\dot{q}) \left[{}^q J_{c_i/\theta} \ddot{q} + {}^q \dot{J}_{c_i/\theta} \dot{q} \right]$$

linear function
 of joint vel
 some variable
 denoted
 to velocity

$$\begin{pmatrix} M_{c_i}^D \\ F_{c_i}^D \end{pmatrix} = \bar{A}_i {}^q J_{c_i/\theta} \ddot{q} + \left[\bar{A}_i {}^q \dot{J}_{c_i/\theta} \dot{q} + \frac{\omega_i \times I_{c_i} (\omega_i)}{\phi} \right]$$

linear const. joint vel
 $\bar{B}_i(q, \dot{q}) \dot{q}$

NOTE

$$\begin{pmatrix} \delta \Theta_i \\ \delta C_i \end{pmatrix}^T = \begin{pmatrix} {}^q J_{c_i/\theta} \delta q \end{pmatrix}^T = \delta q^T {}^q J_{c_i/\theta}^T$$

$$\delta L_{c_i} = \delta q^T {}^q J_{c_i/\theta}^T \left[\bar{A}_i {}^q \dot{J}_{c_i/\theta} \dot{q} + \bar{B}_i \dot{q} - \begin{pmatrix} M_{c_i} \\ F_{c_i} \end{pmatrix} \right] = \phi$$

$$\left({}^q J_{c_i/\theta}^T \bar{A}_i {}^q \dot{J}_{c_i/\theta} \right) \triangleq A_i(q)$$

positive definite symmetric matrix
 Except for singular config of the robot this is always invertible

$$\left({}^q J_{c_i/\theta}^T \bar{B}_i \right) \triangleq B_i(q, \dot{q})$$

$$\begin{pmatrix} M_{c_i} \\ F_{c_i} \end{pmatrix} = \begin{pmatrix} M_{c_i}^A \\ F_{c_i}^A \end{pmatrix} + \begin{pmatrix} M_{c_i}^R \\ F_{c_i}^R \end{pmatrix}$$

This disappears by considering the joint frictionless

$$\delta L_{c_i} = \delta q^T \left[A_i \ddot{q} + B_i \dot{q} - {}^q J_{c_i/\theta}^T \begin{pmatrix} M_{c_i}^A \\ F_{c_i}^A \end{pmatrix} - {}^q J_{c_i/\theta}^T \begin{pmatrix} M_{c_i}^R \\ F_{c_i}^R \end{pmatrix} \right]$$

$$= \delta q^T \left[A_i \ddot{q} + B_i \dot{q} - {}^q J_{c_i/\theta}^T \begin{pmatrix} M_{c_i}^A \\ F_{c_i}^A \end{pmatrix} \right] - \delta q^T {}^q J_{c_i/\theta}^T \begin{pmatrix} M_{c_i}^R \\ F_{c_i}^R \end{pmatrix} = \phi$$

$$\textcircled{4} \quad \delta L^{TOT} = \sum_{i=1}^n \delta L_{c_i} = \sum_{i=1}^n \delta q^T \left[A_i \ddot{q} + B_i \dot{q} - {}^q J_{c_i/\theta}^T \begin{pmatrix} M_{c_i}^A \\ F_{c_i}^A \end{pmatrix} \right] - \delta q^T \left[{}^q J_{c_i/\theta}^T \begin{pmatrix} M_{c_i}^R \\ F_{c_i}^R \end{pmatrix} \right] = \phi$$

$$= \delta q^T \sum_{i=1}^n \left[A_i \ddot{q} + B_i \dot{q} - {}^q J_{c_i/\theta}^T \begin{pmatrix} M_{c_i}^A \\ F_{c_i}^A \end{pmatrix} \right] - \sum_{i=1}^n \delta \Lambda_i = \phi$$

virtual work principle
 principle \rightarrow total virtual work and we proved that
 this is a

$$\mathcal{J}^{\text{TOT}} = \mathcal{J}_{\dot{q}}^T \left[\left(\sum_{i=1}^n A_i \right) \dot{q} + \left(\sum_{i=1}^n B_i(q, \dot{q}) \right) \dot{q} - \sum_{i=1}^n \mathcal{J}_{C_i}^T \left(M_{C_i}^{ext} \right) \right] = \phi$$

↓
non

$$\dot{T} + \dot{T}^{\text{EXT}} - C(q)$$

$$\sum_{i=1}^n \mathcal{J}_{C_i}^T \left(M_{C_i}^{ext} \right)$$

$$\sum_{i=1}^n \mathcal{J}_{C_i}^T \left(\phi m_i \dot{q} \right)$$

Assume solution equivalent to joint
 Assume we can isolate gravity out
 of the term.

$$\begin{cases} M_{C_i}^A = M_{C_i}^{\text{EXT}} \\ F_{C_i}^A = F_{C_i}^{\text{EXT}} + m_i g \end{cases}$$

$$A(q) \triangleq \sum_{i=1}^n A_i(q)$$

non symmetric positive definite

$$B(q, \dot{q}) \triangleq \sum_{i=1}^n B_i(q, \dot{q})$$

non symmetric matrix

(5)

$$\mathcal{J}_L^{\text{TOT}} = \mathcal{J}_{\dot{q}}^T \left[A(q) \dot{q} + B(q, \dot{q}) \dot{q} + C(q) - \dot{T} - \sum_{i=1}^n \mathcal{J}_{C_i}^T \left(M_{C_i}^{ext} \right) \right] = \phi$$

↓
until joint
decelerant vector

Since $\mathcal{J}_{\dot{q}}^T$ is arbitrary we have that the bracket must be equal to 0.

$$(6) A(q) \dot{q} + B(q, \dot{q}) \dot{q} + C(q) = \dot{T} + \dot{T}^{\text{EXT}}$$

↑
 non linear ↗
 Not in normal form

NOTE

$$\ddot{q} = A^{-1}(q) \left[\dot{T} + \dot{T}^{\text{EXT}} - B(q, \dot{q}) \dot{q} - C(q) \right]$$

Somebody writes thus $\ddot{q} = f(q, \dot{q}, \dot{T}, \dot{T}^{\text{EXT}})$

Direct Dynamic Problem. and Inverse Dynamic Problem

Direct is the simulation, meaning that you set the velocity and initial configuration and fix the inputs (τ) and compute the overall joint movement.

Dynamics can be seen as transforming

$$\tau(t), \tau^{\text{ext}}(t), q(t_0), \dot{q}(t_0) \rightarrow \begin{cases} q(t) \\ \dot{q}(t) \\ \ddot{q}(t) \end{cases} \quad \forall t$$

Mother instance is going to solve the forward dynamic model

the inverse does given the motion in cartesian space, want to know the joint torques causing that motion.

Let's assume that we know $q(t)$, $\dot{q}(t)$, $\ddot{q}(t)$ and τ^{ext} we can calculate τ

We can compute

$$\ddot{q}_i : A^{-1} \left[\tau_i + \tau^{\text{ext}}_i + B(q, \dot{q})\dot{q}_i - C(q) \right]$$

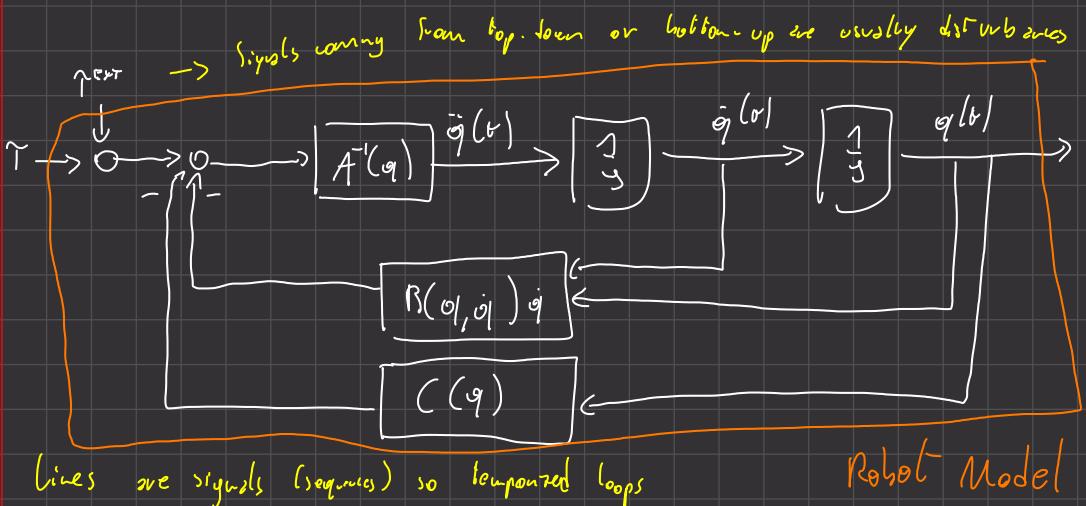
Now we can integrate

Getting q and \dot{q}

$$\dot{q}(t) = \int_0^t \ddot{q}(l) dl + q(0)$$

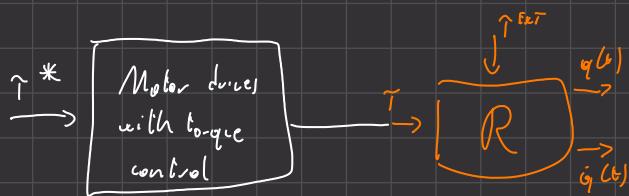
$$q(t) = \int_0^t \dot{q}(l) dl + q(0)$$

Block diagram.



Play and Friction are not taken into account but usually described at control level.

The Robot Model can be described as



Fundamental properties

- $A(q) \in \mathbb{R}^{n \times n}$
 - $A = A^T$
 - $A > 0$
- $$\Rightarrow \begin{cases} A^{-1} \text{ exists} \\ \exists P : P^T A P = \text{diag}(\lambda_1, \dots, \lambda_n) \quad \lambda_i > 0 \\ P \cdot P^T = I_n \end{cases} \rightarrow \text{This is a kind of norm rotation matrix}$$
- $\lambda_{\min} \|\underline{x}\|^2 \leq \underline{x}^T A \underline{x} \leq \lambda_{\max} \|\underline{x}\|^2 \rightarrow$ way of giving the sum of the matrix quadratic form
 - $\frac{1}{2} \dot{A}(q) - B(q, \dot{q}) = N \quad \text{and} \quad N = -N^T \quad (\text{so } N \text{ symmetric})$
 - $\|B(q, \dot{q})\| \leq b_0 \|\dot{q}\| \quad \text{or} \quad \|B(q, \dot{q})\dot{q}\| \leq b_0 \|\dot{q}\|^2$
this is not growing faster than a quadratic
 - $\|(Cq)\| = \sqrt{C^T C} \leq c_0$
↳ constant (May be not true for long joints)

Note

$F \in \mathbb{R}^{m \times n}$ (induced Norm)?

$$\|F\| = ?$$

Let's start from vector: $\underline{x} \in \mathbb{R}^m$

$$\|\underline{x}\|_2 = \sqrt{\underline{x}^T \underline{x}} \quad \|\underline{x}\|_p = \sqrt[p]{\sum_{i=1}^m |x_i|^p}$$

$$\|\underline{x}\|_1 = \sum_{i=1}^m |x_i|$$

$$\|\underline{x}\|_\infty = \max_i |x_i|$$

$$\|\underline{x}\|_\infty \leq \|\underline{x}\|_p \leq \|\underline{x}\|_2 \leq \|\underline{x}\|_1$$

$$\|F\| = \max \|\underline{F}x\|$$

$\|\underline{x}\|=1$, unit vectors

