FORWARD KINEMATICS

$q \rightarrow \chi_e(q)$

$$T_{AB} = \begin{pmatrix} C_{AB} & {}_A r_{AB} \\ 0_{1\times 3} & 1 \end{pmatrix}$$

linear and angular velocities

$$\dot{r} = E_p(\chi_P)\dot{\chi}_P \longrightarrow \qquad {}_A\omega_{AB} = E_R(\chi_R)\dot{\chi}_R \longrightarrow \qquad \begin{pmatrix} \dot{r} \\ \omega \end{pmatrix} = \begin{pmatrix} E_P(\chi_P) \\ E_R(\chi_R) \end{pmatrix} \begin{pmatrix} \dot{\chi}_P \\ \dot{\chi}_R \end{pmatrix}$$

$$\omega = E(\chi)\dot{\chi}$$

$$\chi_e = \begin{pmatrix} \chi_{eP} \\ \chi_{eR} \end{pmatrix} = \chi_e(q) \qquad \dot{\chi}_e = \frac{\partial \chi_e}{\partial q}\dot{q}$$

end effector parametrization

ANALYTICAL JACOBIAN

$$\dot{\chi}_e = J_A(q)\,\dot{q}$$

$$w_e = \begin{pmatrix} v_e \\ \omega_e \end{pmatrix} = J_o(q)\,\dot{q}$$

GEOMETRIC GIACOBIAN

$$w_e = E_e(\chi)\dot{\chi} = \boxed{E_e(\chi)J_A(q)}\dot{q}$$

$$J_o = E(\chi)J_A$$

parametrization independent — parametrization dependent

ANALYTICAL

PARAMETRIZATION DEPENDENT

$$J_{eA}(q) = \frac{\partial \chi_e}{\partial q}$$

GEOMETRIC

$${}_I J_{eo}(q) = \begin{bmatrix} {}_I J_{eP}(q) \\ {}_I J_{eR}(q) \end{bmatrix} = \begin{bmatrix} {}_I m_1 \times {}_I r_{1E} & {}_I m_2 \times {}_I r_{2E} & \cdots & {}_I m_m \times {}_I r_{nE} \\ {}_I m_1 & {}_I m_2 & \cdots & {}_I m_m \end{bmatrix}$$

PARAMETRIZATION INDEPENDENT

${}_I m_2$ = rotation axis of ref. frame 2 expressed in the I frame

${}_I r_{2E}$ = vector from the origin of frame 2 to the end effector, expressed in I

INVERSE KINEMATICS (DIFFERENTIAL)

$$w_e = J_{eo}\,\dot{q}$$

$$\dot{q} = \begin{cases} J_{eo}^{-1}\,w_e & m = m \\ J_{eo}^{+}\,w_e & m \neq m \end{cases}$$

inverse

finite numb. sol.

MOORE - PENROSE PSEUDO-INVERSE

REDUNDANCY $w_e = J_{eo}(\dot{q} + N\dot{q}_o)$

infinite numb. of solutions (min $\dot{q}$)

$$J_{eo}^{+} = \begin{cases} J^T(JJ^T)^{-1} & m < m \quad \text{RIGHT P.I.} \\ (JJ^T)^{-1}J^T & m > m \quad \text{LEFT P.I.} \end{cases}$$

no exact sol. minimize MSE

SINGULARITY $\longrightarrow (JJ^T + \lambda^2 I)^{-1} J^T$

JOINT - SPACE — GENERALIZED COORDINATES : set of scalar parameters that uniquely define the robot's configuration

TASK - SPACE — END-EFFECTOR CONF. PARAM : set of parameters that completely specify position and orient. of the end eff.

OPERATIONAL SPACE COORD. : subset of end-effector param. that are independent (considering the structure of the robot)

MULTI-TASK CONTROL

NO PRIORITY

$$w_1 = J_1 \dot{q} \quad w_2 = J_2 \dot{q} \quad \vdots \quad w_n = J_n \dot{q} \longrightarrow \dot{q} = \begin{bmatrix} J_1 \\ J_2 \\ \vdots \\ J_n \end{bmatrix}^+ \begin{bmatrix} w_1^* \\ w_2^* \\ \vdots \\ w_n^* \end{bmatrix}$$

PRIORITY

$$w_1 = J_1 \dot{q}$$

$$\dot{q} = J_1^+ (w_1 + \underbrace{N_1 \dot{q}_0}_{\text{free parameter for extra tasks}})$$

$$\longrightarrow w_2 = J_2 \dot{q} = J_2 [J_1^+ (w_1 + N_1 \dot{q}_0)]$$

$$\dot{q}_0 = (J_2 N_1)^+ (w_2 - J_2 J_1^+ w_1)$$

$$\dot{q} = \dot{q}_{prev} + \dot{q}_0 = J_1^+ (w_1 + N_1 q_0) + (J_2 N_1)^+ (w_2 - J_2 J_1^+ w_1)$$

ITERATIVE INVERSE KINEMATICS

$$(\Delta q = J_{eA}^+ \Delta x)$$

algoritm that iteratively refines the pose

$q = q_0$

while $\| x_e^* - x_e(q) \| > tol$

$$J_{eA} \leftarrow J_{eA}(q) = \frac{\partial x}{\partial q}$$

$$\Delta x \leftarrow x^* \boxminus x_e(q)$$

pose error

$$\Delta q \leftarrow J_{eA}^+ \Delta x$$

Inverse kinematics can be realisez analitically (for a set of known mechanisms) or numerically (iterative inverse kinematics)

POSITION

$\Delta r = r^* - r(t)$

ROTATION

NOT $\Delta \varphi = \varphi^* - \varphi(t)$ (they are not normal vectors)

YES

$$C_{IB} = \text{rotMat}(\varphi^*)$$

$$C_{IA} = \text{rotMat}(\varphi(t))$$

$$C_{AB} = C_{IA}^T C_{IB}$$

$$_A\Delta\varphi = \text{rotVec}(C_{AB})$$

$$_I\Delta\varphi = C_{IA} \, \text{rotVec}(C_{AB})$$

or directly $_I\Delta\varphi = \text{rotVec}(C_{IB} C_{IA}^T)$

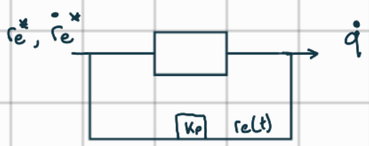
TRAJECTORY CONTROL

Feedback

$$\dot{q} = \begin{pmatrix} J_{eoP}^+ \\ J_{eoR}^+ \end{pmatrix} \begin{pmatrix} \dot{r}_e^* + K_{pp} \Delta r_e \\ w_e^* + K_{pR} \Delta \varphi \end{pmatrix}$$

How to get the joint velocity given a desired end effector trajectory?

we have reference trajectory r_e^* and velocity $\dot{r}_e^*(t)$. The input is the reference speed while the feedback is on the error in position



DYNAMICS

$$M(q)\ddot{q} + b(q,\dot{q}) + g(q) = \tau + J_c(q)^T F_c$$

NEWTON-EULER METHOD

decompose the body into independent components, adding the joint constraints. Apply the d'Alambert principle (virtual work) to each component.

$$0 = \delta W = \sum_{i=0}^{m_b} \begin{pmatrix} \delta r_i \\ \delta \Phi_i \end{pmatrix} \begin{pmatrix} \dot{p}_i - F_i \\ \dot{N}_i - T_i \end{pmatrix} \rightarrow \begin{matrix} F_{ext} = \dot{p} \\ T_{ext} = \dot{N} \end{matrix} \quad \begin{matrix} p = mv \\ N = \Theta\omega \end{matrix}$$

LAGRANGE II

$L = T - U$ lagrangian function

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q} = \tau \qquad \tau = \text{total generalized forces}$$

PROJECTED NEWTON-EULER

similar to newton-euler but in the end it plugs into the equations the relationships:

$$v = J_{eoP}\,\dot{q}$$
$$\omega = J_{eoR}\,\dot{q}$$

so that the equation becomes a function of the generalized coordinates

$$M(q)\ddot{q} + b(q,\dot{q}) + g(q) = 0$$

EXTERNAL FORCES

all forces and torques have to be mapped to generalized forces

$$\boxed{\begin{matrix}\tau_F \\ \tau_T\end{matrix}} \quad \begin{matrix} = J_{oP}^T F_{ext} & \text{— external forces} \\ = J_{oR}^T T_{ext} & \text{— external torque} \end{matrix}$$

generalized forces

$$\mathbf{M} = \sum_{i=1}^{n_b} \left({}_{\mathcal{A}}\mathbf{J}_{S_i}^T \cdot m \cdot {}_{\mathcal{A}}\mathbf{J}_{S_i} + {}_{\mathcal{B}}\mathbf{J}_{R_i}^T \cdot {}_{\mathcal{B}}\mathbf{\Theta}_{S_i} \cdot {}_{\mathcal{B}}\mathbf{J}_{R_i} \right)$$

$$\mathbf{b} = \sum_{i=1}^{n_b} \left({}_{\mathcal{A}}\mathbf{J}_{S_i}^T \cdot m \cdot {}_{\mathcal{A}}\dot{\mathbf{J}}_{S_i} \cdot \dot{\mathbf{q}} + {}_{\mathcal{B}}\mathbf{J}_{R_i}^T \cdot \left({}_{\mathcal{B}}\mathbf{\Theta}_{S_i} \cdot {}_{\mathcal{B}}\dot{\mathbf{J}}_{R_i} \cdot \dot{\mathbf{q}} + {}_{\mathcal{B}}\mathbf{\Omega}_{S_i} \times {}_{\mathcal{B}}\mathbf{\Theta}_{S_i} \cdot {}_{\mathcal{B}}\mathbf{\Omega}_{S_i} \right) \right)$$

$$\mathbf{g} = \sum_{i=1}^{n_b} \left(-{}_{\mathcal{A}}\mathbf{J}_{S_i}^T \, {}_{\mathcal{A}}\mathbf{F}_{g,i} \right)$$

DYNAMIC CONTROL - JOINT SPACE DYNAMICS

$$M(q)\ddot{q} + b(q,\dot{q}) + g(q) = \tau + J_c^T F_c$$

generalized contact forces

In trajectory tracking we use feedback to follow a desired position and velocity trajectory.

This is no more enough if we want to control the force applied by the end effector.

We need to add a feedback on joint torque.

$$M(q)\ddot{q} + b(q,\dot{q}) + g(q) = \tau$$

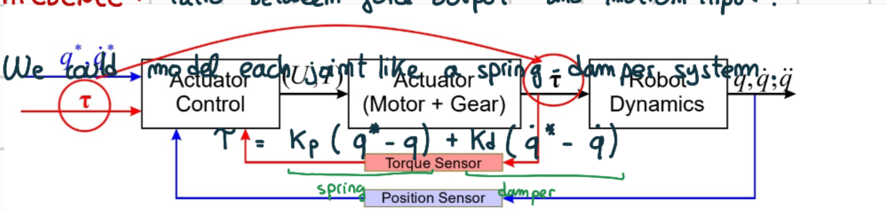
what torque should we use to follow a desired trajectory

JOINT IMPEDENCE CONTROL

CONFIGURATION DEPENDENT

IMPEDENCE : ratio between force output and motion input.

We could model each joint like a spring - damper system.



$$\tau = \underbrace{K_p(q^* - q)}_{spring} + \underbrace{K_d(\dot{q}^* - \dot{q})}_{damper}$$

This controller has a not removable static offset

$$\cancel{M(q)\ddot{q}} + \cancel{b(q,\dot{q})} + \underline{g(q)} = \underline{K_p(q^* - q)} + K_d(\cancel{\dot{q}^*} - \dot{q})$$

We can add a term to the controller to compensate gravity

$$\tau = K_p(q^* - q) + K_d(\dot{q}^* - \dot{q}) + \hat{g}(q)$$

the estimated gravity term

the gains are constant, but the true torque to apply depends on the configuration (different configurations can have equal $(q^* - q)$) $\rightarrow$ BAD PERFORMANCE

INVERSE DYNAMICS CONTROL

CONFIGURATION INDEPENDENT

(model-based compensation)

$$\tau = \hat{M}(q)\ddot{q}^* + \hat{b}(q,\dot{q}) + \hat{g}(q)$$

IDEAL CASE : no modelling errors

DYNAMICS = $\underline{M(q)}\ddot{q} + b(q,\dot{q}) + g(q) = \underline{\hat{M}(q)}\ddot{q}^* + \hat{b}(q,\dot{q}) + \hat{g}(q)$

$$\mathbb{I}\ddot{q} = \ddot{q}^*$$

In the ideal case, if there are no modelling errors, the output acceleration is exactly the input desired acceleration.

INPUT $\ddot{q}^*$ $\rightarrow$ OUTPUT $\ddot{q} = \ddot{q}^*$

To compute $\ddot{q}^*$ we can use a PD controller

$$\ddot{q}^* = \underline{K_p}(q^* - q) + \underline{K_d}(\dot{q}^* - \dot{q})$$

natural frequency damping ratio

$$\omega = \sqrt{K_p} \qquad D = \frac{K_d}{2\sqrt{K_p}}$$

Each joint behaves like a decoupled mass - spring - damper with unitary mass.

Often instead of a desired joint space acceleration, we have a desired task-space acceleration
But we know that:

$$w_e = \begin{pmatrix} v \\ \omega \end{pmatrix} = J_{eo}\,\dot q$$
$$\dot w_e = \dot J_{eo}\,\dot q + J_{eo}\,\ddot q$$

$$\ddot q^* = J_{eo}^+ \left(\dot w_e - \dot J_{eo}\,\dot q\right)$$ SINGLE TASK

$$\ddot q^* = \begin{bmatrix} J_1 \\ \vdots \\ J_m \end{bmatrix}^+ \left[\begin{pmatrix} \dot w_1 \\ \vdots \\ \dot w_n \end{pmatrix} - \begin{pmatrix} \dot J_1 \\ \vdots \\ \dot J_n \end{pmatrix} \dot q \right]$$ MULTY-TASK (NO PRIORITY)

$\ddot q =$ .... (formula) MULTY-TASK (PRIORITY)

$$\boxed{\text{desired task-space acceleration } \dot w_e^* \xrightarrow{\ddot q^* = J_{eo}^+(\dot w_e - \dot J_{eo}\dot q)} \text{desired joint-space acceleration } \ddot q^* \xrightarrow{\tau = \hat M(q)\ddot q^* + \hat b(q,\dot q) + \hat g(q)} \text{input torque}}$$

DYNAMIC CONTROL - TASK SPACE DYNAMICS

JOINT-SPACE | TASK-SPACE

$$M(q)\ddot q + b(q,\dot q) + g(q) = \tau \longrightarrow \begin{aligned} \Lambda &= (J_e M^{-1} J_e^T)^{-1} \\ \mu &= \Lambda J_e M^{-1} b - \Lambda \dot J_e \dot q \\ p &= \Lambda J_e M^{-1} g \end{aligned} \longrightarrow \underbrace{\Lambda}_{\text{end-effector inertia}} \dot w_e + \underbrace{\mu}_{\text{coriolis and centrifugal term}} + \underbrace{p}_{\text{gravitational part}} = F_e$$

$\tau = J_e^T F_e$

We can apply the same process that we did in the joint space, but in the task space.
Let's assume we have a reference trajectory x_e^*, w_e^*.
We can define the following desired acceleration $\dot w_e^*$ to follow the trajectory (PD control).

$$\dot w_e^* = K_p E \underbrace{(x_e^* - x_e)}_{\text{config. error (param. dependent)}} + K_d \underbrace{(w_e^* - w_e)}_{\text{velocity error}} + \underbrace{\dot w_e^*(t)}_{\text{feedforward term}}$$

$\Delta\phi = E_R(x_R)\cdot\Delta x_R$

given the desired end-effector acceleration $\dot w_e^*$, if we have a good estimate of the model, we can find

$$F_e^* = \hat\Lambda\,\dot w_e^* + \hat\mu + \hat p$$ (estimated model) INVERSE DYNAMIC CONTROL

From the desired external force F_e^*, we can also find the joint torque $\tau = J_e^T F_e$

Also in this case we can use the NULL-SPACE to meet other tasks.

$$\tau = J^T F = J^T(\Lambda\dot w + \mu + p) + N(J^T)\,\tau_0$$

$$N(J^T) = I - J^T (J M^{-1} J^T)^{-1} J M^{-1}$$

Motion and Force Control

Operational space control

Let's start with the EoM in the task-space, keeping into account that we might want to produce an acceleration in some directions or a force in others.

$$F_c + \Lambda \dot{w}_e + \mu + p = F_e$$

$$\tau = J_e^T F_e = J_e^T (\Lambda \dot{w}_e + F_c + \mu + p)$$

to select the direction allowed for motion and force we use the selection matrices

$$\tau = J_e^T (\Lambda S_M \dot{w}_e + S_F F_c + \mu + p)$$

$$S_M = \begin{bmatrix} C^T \Sigma_p C & 0 \\ 0 & C^T \Sigma_R C \end{bmatrix} \qquad \Sigma_p = \begin{bmatrix} \sigma_{px} & 0 & 0 \\ 0 & \sigma_{py} & 0 \\ 0 & 0 & \sigma_{pz} \end{bmatrix} \qquad \sigma_i = \begin{cases} 1 & \text{direction allowed} \\ 0 & \text{direction \underline{not} allowed} \end{cases}$$

$$S_F = \begin{bmatrix} C^T (I - \Sigma_p) C & 0 \\ 0 & C^T (I - \Sigma_R) C \end{bmatrix} \qquad \Sigma_R = \begin{bmatrix} \sigma_{Rx} & 0 & 0 \\ 0 & \sigma_{Ry} & 0 \\ 0 & 0 & \sigma_{Rz} \end{bmatrix}$$

Inverse dynamics as QP

The control problems we have seen so far are Quadratic Programming optimization problems.

1. INVERSE DYNAMICS $\quad \dot{w}_e^* \longrightarrow \ddot{q}^* = J^+ (\dot{w}_e^* - \dot{J}\dot{q}) \longrightarrow \tau^* = \hat{M}(q)\ddot{q}^* + \hat{b}(q,\dot{q}) + \hat{g}(q) = 0$
2. OPERATIONAL SPACE CONTROL $\quad \dot{w}_e^* \longrightarrow F_e^* = \Lambda \dot{w}_e^* + \mu + p \longrightarrow \tau^* = J_e^T F_e$

QP PROBLEM

$$\min \underbrace{\|\ddot{q}\|_2}_{m<n} \text{ or } \underbrace{\|\dot{w}_e - \dot{J}\dot{q} - J\ddot{q}\|}_{m>n}$$

s.t. $\quad H\ddot{q} + b + g = \tau$

$\quad \dot{w}_e = \dot{J}\dot{q} + J\ddot{q}$

it depends on the dimensions of J:
- m = number of constraints
- n = # generalized coord. = DOF

I can also decide to minimize for example the torque.

!!! DO S6 QUIZ (last one)

LEAST SQUARED OPTIMIZATION

$$\mathbf{Ax} - \mathbf{b} = \mathbf{0} \longrightarrow \mathbf{x} = \mathbf{A}^+\mathbf{b} \qquad \min_{\mathbf{x}} \|\mathbf{Ax} - \mathbf{b}\|_2 \qquad \min \|\mathbf{x}\|_2$$

$$\mathbf{A}_1\mathbf{x}_1 - \mathbf{b} = -\mathbf{A}_2\mathbf{x}_2 \longrightarrow \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix} = [\mathbf{A}_1 \quad \mathbf{A}_2]^+ \mathbf{b} \qquad \min_{\mathbf{x}_1,\mathbf{x}_2} \left\| [\mathbf{A}_1 \quad \mathbf{A}_2] \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix} - \mathbf{b} \right\|_2 \qquad \min \left\| \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix} \right\|_2$$

$$\begin{matrix} \mathbf{A}_1\mathbf{x} - \mathbf{b}_1 = \mathbf{0} \\ \mathbf{A}_2\mathbf{x} - \mathbf{b}_2 = \mathbf{0} \end{matrix} \xrightarrow{\text{Equal priority}} \mathbf{x} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{bmatrix}^+ \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{pmatrix} \qquad \min_{\mathbf{x}} \left\| \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{bmatrix} \mathbf{x} - \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{pmatrix} \right\|_2 \qquad \min \|\mathbf{x}\|_2$$

Hierarchy

$$\left.\begin{matrix} \mathbf{x} = \mathbf{A}_1^+\mathbf{b}_1 + \mathcal{N}(\mathbf{A}_1)\mathbf{x}_0 \\ \mathbf{A}_2\mathbf{x} - \mathbf{b}_2 = \mathbf{A}_2\left(\mathbf{A}_1^+\mathbf{b}_1 + \mathcal{N}(\mathbf{A}_1)\mathbf{x}_0\right) - \mathbf{b}_2 = \mathbf{0} \\ \mathbf{x}_0 = \left(\mathbf{A}_2\mathcal{N}(\mathbf{A}_1)\right)^+ \left(\mathbf{b}_2 - \mathbf{A}_2\mathbf{A}_1^+\mathbf{b}_1\right) \end{matrix}\right\} \qquad \begin{matrix} \min_{\mathbf{x}} \|\mathbf{A}_1\mathbf{x} - \mathbf{b}_1\|_2 \\ \left\{\begin{matrix} \min_{\mathbf{x}} \|\mathbf{A}_2\mathbf{x} - \mathbf{b}_2\|_2 \\ s.t. \|\mathbf{A}_1\mathbf{x} - \mathbf{b}_1\| = c_1 \end{matrix}\right. \end{matrix}$$

Robot Dynamics |

Algorithm 2 Hierarchical Least Square Optimization

n_T = Number of Tasks
$\mathbf{x} = \mathbf{0}$ ▷ initial optimal solution
$\mathbf{N}_1 = \mathbb{I}$ ▷ initial null-space projector
for $i = 1 \to n_T$ **do**
$\quad \mathbf{x}_i = (\mathbf{A}_i\mathbf{N}_i)^+ (\mathbf{b}_i - \mathbf{A}_i\mathbf{x})$
$\quad \mathbf{x} = \mathbf{x} + \mathbf{N}_i\mathbf{x}_i$
$\quad \mathbf{N}_{i+1} = \mathcal{N}\left([\mathbf{A}_1^T \quad \dots \quad \mathbf{A}_i^T]^T\right)$
end for

if you have multiple task with hierarchical priority:
1. $A_1 x - b_1 = 0$
2. $A_2 x - b_2 = 0$

$\vdots$

(m) $A_m x - b_m = 0$

there are numerically more stable algorithms.

FLOATING BASE SYSTEM - KINEMATICS

GENERALIZED COORDINATES

$$q = \begin{pmatrix} q_B \\ q_J \end{pmatrix} \quad q_B = \begin{pmatrix} q_{B,P} \\ q_{B,R} \end{pmatrix} \in \mathbb{R}^3 \times SO(3)$$

base

joints

GENERALIZED VELOCITY

the generalized coordinates q are parametrization dependent.
Let's define the generalized velocities and accelerations as

$$u = \begin{pmatrix} {}_I v_B \\ {}_B \omega_{IB} \\ \dot\varphi_1 \\ \vdots \\ \dot\varphi_m \end{pmatrix} \begin{matrix} \}\, \mathbb{R}^6 \\ \}\, \mathbb{R}^{n_j} \end{matrix} \in \mathbb{R}^{6+n_j} = \mathbb{R}^{n_u} \qquad \dot u = \begin{pmatrix} {}_I a_B \\ {}_B \psi_{IB} \\ \ddot\varphi_1 \\ \vdots \\ \ddot\varphi_m \end{pmatrix}$$

base

joints

$u, \dot u$ are parametrization independent

$$u = \begin{pmatrix} u_{B,P} \\ u_{B,R} \\ u_J \end{pmatrix} = \begin{bmatrix} \mathbb{I}_3 & 0 & 0 \\ 0 & E_{\chi R} & 0 \\ 0 & 0 & \mathbb{I}_{m_j} \end{bmatrix} \begin{pmatrix} \dot q_{B,P} \\ \dot q_{B,R} \\ \dot q_j \end{pmatrix}$$

$$u = E_{fb}\, \dot q$$

POSITION OF A POINT

$${}_I r_{IQ}(q) = {}_I r_{IB}(q) + C_{IB}(q)\, {}_B r_{BQ}(q)$$

VELOCITY OF A POINT

we get this equation from the time derivative of the position of a point

$${}_I v_Q = \underbrace{\left[\, \mathbb{I}_3 \quad -C_{IB} \cdot [{}_B r_{BQ}]_\times \quad C_{IB} \cdot {}_B J_{P_{q_i}}(q_i) \,\right]}_{{}_I J_Q(q)} \underbrace{\begin{bmatrix} u_{B,P} \\ u_{B,R} \\ u_J \end{bmatrix}}_{u}$$

a different jacobian for each joint

$${}_I v_Q = {}_I J_Q\, u$$

CONTACT CONSTRAINT

a contact point has zero velocity and acceleration

$${}_I \dot r_{IQ} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \rightarrow {}_I J_{c_i}\, u = 0 \qquad \wedge \qquad {}_I \ddot r_{IQ} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \rightarrow {}_I J_{c_i}\, \dot u + {}_I \dot J_{c_i}\, u = 0$$

For each contact point both constraints are valid $\forall c_i$ $\begin{cases} J_{c_i} u = 0 \\ J_{c_i} \dot u + \dot J_{c_i} u = 0 \end{cases}$

For multiple contact points we can stack the constraints

$$J_c = \begin{bmatrix} J_{c_1} \\ \vdots \\ J_{c_{n_c}} \end{bmatrix} \in \mathbb{R}^{3n_c \times n_n} \rightarrow \begin{cases} J_c u = 0 \\ J_c \dot u + \dot J u = 0 \end{cases}$$

To recap: the following equation derives from the robot's structure

$$ {}_I v_Q = J_Q \, u $$

velocity of point Q — generalized velocities $u = \begin{pmatrix} u_b \\ u_j \end{pmatrix} = \begin{pmatrix} u_{b,p} \\ u_{p,r} \\ u_j \end{pmatrix}$

For contact points ($Q = C_i$) velocity and acceleration is 0. Each contact point C_i has its own jacobian J_{C_i}.
We can stack all the contact points Jacobians and write all the contact constraints in a compact matrix way.

$$ J_c = \begin{bmatrix} J_{c_1} \\ \vdots \\ J_{c_{n_c}} \end{bmatrix} \longrightarrow \begin{cases} J_c u = 0 \\ J_c \dot{u} + \dot{J}_c u = 0 \end{cases} $$

Jacobian of all the contact points

$$ J_c u = 0 $$

$$ \begin{bmatrix} J_{c,b} & J_{c,j} \end{bmatrix} \begin{bmatrix} u_b \\ u_j \end{bmatrix} = 0 $$

relationship between the base motion and the contact point velocity — contact point velocity

rank (J_c) = # independent contact constraint
rank $(J_{c,b})$ = # base constraints

rank(J_c) − rank$(J_{c,b})$ = # internal constraints

m JOINTS (ACTUATED)
BASE (NOT ACTUATED)

m+6 DoF

UNCONTROLLABLE DoFs: $6 - rk(J_{c,b})$

TO UNDERSTAND BETTER

$$ J_{c,b} \cdot u_b = 0 $$

u_b are the generalized base velocities ($\dim(u_b) = 6$).
This is an homogeneous system of equations.
Which u_b allow us to get a 0 velocity at the contact point?
The trivial solution $u_b = 0$ is always valid: if the base doesn't move the contact point won't move either. But the question is: can I move the base without moving the contact points?
To do so, we need non-trivial solutions to exist → rank$(J_{c,b}) = 6$
If rank$(J_{c,b}) = 6$, there exist at least one $u_b \neq 0$ (base motion) that fulfills the contact constraint $J_{c,b} u_b = 0$.

Each contact point provides 3 independent constraints.

m contact points → 3m ind. constr. → rank$(J_c) = 3m$

To fully control the base configuration, however, we need

controllable base configur. ⟷ rank$(J_{c,b}) = 6$

EXAMPLE for a legged robot
- 2 contact points: $rk(J_c) = 6$, $rk(J_{c,b}) = 5$ ✗
- 3 contact points: $rk(J_c) = 9$, $rk(J_{c,b}) = 6$ ✓

EXTRA TASK

We want our robot to fullfill other tasks while respecting the contact constraints → hierarchical multi-task

① contact constraints
- $J_c u = 0 \equiv J_c u - J_c N_c u_0 = 0$ ($N_c = \mathcal{N}(J_c)$ null-space)

$$ J_c (u - N_c u_0) = 0 $$

$$ u = N_c u_0 \quad ★ $$

- $J_c \dot{u} + \dot{J}_c u = 0 \equiv J_c \dot{u} + \dot{J}_c u - J_c N_c \dot{u}_0 = 0$

$$ J_c (\dot{u} - N_c \dot{u}_0) = -\dot{J}_c u $$

$$ \dot{u} = J_c^+ (-\dot{J}_c u) + N_c \dot{u}_0 \quad ★ $$

② task

$$ w_t^* = J_t u \quad \text{or} \quad \dot{w}_t^* = \dot{J}_t u + J_t \dot{u} $$

★ ★ ★

FLOATING BASE SYSTEM - DYNAMICS

EoM :

$$M(q)\dot{u} + b(q,u) + g(q) + J_c^T F_c = S^T \tau$$

F_c exerted by the robot on the environment

$$M(q)\dot{u} + b(q,u) + g(q) = S^T \tau + J_c^T F_c$$

F_c exerted on the robot

$S = [\, 0_{m \times 6} \quad I_{n_j \times n_j} \,]$

selection matrix that removes the generalized forces relative to the base since it is not actuated

τ = generalized torques acting in direction of generalized coordinates

SOFT CONTACT modelled as a spring-damper force

$$F_c = K_p (r_c - r_{c0}) + K_d \dot{r}_c$$

phisical parameters (K_p, K_d) leads to numerical instability.
Trade-off between physical accuracy and numerical stability.

HARD CONTACT

contact $\ddot{r} = 0$ $F \geq 0$
non contact $\ddot{r} > 0$ $F = 0$

$\} \quad \ddot{r} \cdot F = 0$

Given the hard contact of the system, we can compute a dinamically consistent support null-space matrix

$$\mathbf{N}_c = \mathbb{I} - \mathbf{M}^{-1}\mathbf{J}_c^T \left(\mathbf{J}_c\mathbf{M}^{-1}\mathbf{J}_c^T\right)^{-1}\mathbf{J}_c.$$

and write down the constraint consistent EoM

$$\mathbf{N}_c^T \left(\mathbf{M}\dot{\mathbf{u}} + \mathbf{b} + \mathbf{g}\right) = \mathbf{N}_c^T \mathbf{S}^T \boldsymbol{\tau}.$$

Recapitulation: Support Consistent Dynamics

- Equation of motion $\mathbf{M}(\mathbf{q})\dot{\mathbf{u}} + \mathbf{b}(\mathbf{q},\mathbf{u}) + \mathbf{g}(\mathbf{q}) + \mathbf{J}_c^T\mathbf{F}_c = \mathbf{S}^T\boldsymbol{\tau}$
  - Cannot directly be used for control due to the occurrence of contact forces
- Contact constraint $\ddot{\mathbf{r}}_c = \mathbf{J}_c\dot{\mathbf{u}} + \dot{\mathbf{J}}_c\mathbf{u} = 0$
- Contact force $\mathbf{F}_c = \left(\mathbf{J}_c\mathbf{M}^{-1}\mathbf{J}_c^T\right)^{-1}\left(\mathbf{J}_c\mathbf{M}^{-1}\left(\mathbf{S}^T\boldsymbol{\tau} - \mathbf{b} - \mathbf{g}\right) + \dot{\mathbf{J}}_c\mathbf{u}\right)$
  - Back-substitute in (1), replace $\dot{\mathbf{J}}_c\dot{\mathbf{q}} = -\mathbf{J}_c\ddot{\mathbf{q}}$ and use support null-space projection $\mathbf{N}_c = \mathbb{I} - \mathbf{M}^{-1}\mathbf{J}_c^T\left(\mathbf{J}_c\mathbf{M}^{-1}\mathbf{J}_c^T\right)^{-1}\mathbf{J}_c$
- Support consistent dynamics $\mathbf{N}_c^T\left(\mathbf{M}\ddot{\mathbf{q}} + \mathbf{b} + \mathbf{g}\right) = \mathbf{N}_c^T\mathbf{S}^T\boldsymbol{\tau}$
- Inverse-dynamics $\boldsymbol{\tau}^* = \left(\mathbf{N}_c^T\mathbf{S}^T\right)^+\mathbf{N}_c^T\left(\mathbf{M}\ddot{\mathbf{q}} + \mathbf{b} + \mathbf{g}\right)$
- Multiple solutions $\boldsymbol{\tau}^* = \left(\mathbf{N}_c^T\mathbf{S}^T\right)^+\mathbf{N}_c^T\left(\mathbf{M}\ddot{\mathbf{q}}^* + \mathbf{b} + \mathbf{g}\right) + \mathcal{N}\left(\mathbf{N}_c^T\mathbf{S}^T\right)\boldsymbol{\tau}_0^*$

Task Space Control as Quadratic Program

A general problem

$$\boxed{\min_{\mathbf{x}} \quad \|\mathbf{A}_i\mathbf{x} - \mathbf{b}_i\|_2 \qquad \mathbf{x} = \begin{pmatrix}\dot{\mathbf{u}} \\ \mathbf{F}_c \\ \boldsymbol{\tau}\end{pmatrix}}$$

- We search for a solution that fulfills the equation of motion

$\mathbf{M}(\mathbf{q})\dot{\mathbf{u}} + \mathbf{b}(\mathbf{q},\mathbf{u}) + \mathbf{g}(\mathbf{q}) + \mathbf{J}_c^T\mathbf{F}_c = \mathbf{S}^T\boldsymbol{\tau} \quad \Rightarrow \mathbf{A} = \begin{bmatrix}\hat{\mathbf{M}} & \hat{\mathbf{J}}_c^T & -\mathbf{S}^T\end{bmatrix} \quad \mathbf{b} = -\hat{\mathbf{b}} - \hat{\mathbf{g}}$

- Motion tasks: $\mathbf{J}\dot{\mathbf{u}} + \dot{\mathbf{J}}\mathbf{u} = \dot{\mathbf{w}}^*$ $\Rightarrow \mathbf{A} = \begin{bmatrix}\hat{\mathbf{J}}_i & 0 & 0\end{bmatrix}$ $\mathbf{b} = \dot{\mathbf{w}}^* - \hat{\dot{\mathbf{J}}}_i\mathbf{u}$
- Force tasks: $\mathbf{F}_i = \mathbf{F}_i^*$ $\Rightarrow \mathbf{A} = \begin{bmatrix}0 & \mathbb{I} & 0\end{bmatrix}$ $\mathbf{b} = \mathbf{F}_i^*$
- Torque min: $\min\|\boldsymbol{\tau}\|_2$ $\Rightarrow \mathbf{A} = \begin{bmatrix}0 & 0 & \mathbb{I}\end{bmatrix}$ $\mathbf{b} = 0$

LEGGED ROBOTS

ACTUATION

HIGH GEAR + TORQUE SENSOR + ELASTICITY

The motor spins at higher speeds than the end-actuator. This allows the motor to work at efficient velocities.

A geared system however is also not robust to impacts and multiple gears can affect efficiency.

The motor can be very compact but still having a big reflected inertia.

adding a spring:
+ robust to impact
+ power / speed
+ temporary energy storage → efficiency
− low control bandwidth

LOW GEAR + CURRENT CONTROL

Low gears → robust to impacts
→ to achieve big power you need big motors

※ slides (lecture 8)

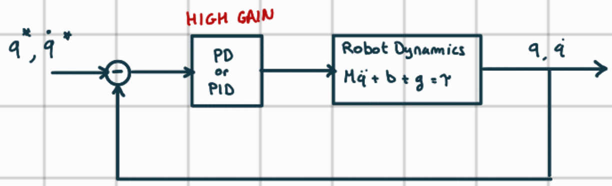
HYDRAULIC

CONTROL

STATIC STABILITY : CoG projection falls into the support polygon
+ safe
− slow, inefficient

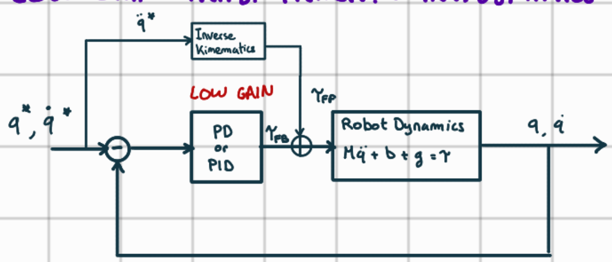
DYNAMIC WALKING : needs to move to not fall.
+ fast, efficient
− needs more control

HIGH GAIN TRAJECTORY TRACKING



With the high gain the robot stays close to the reference trajectory. This control requires perfect knowledge of the environment, since the dynamic is not compliant to disturbances.

LOW GAIN TRAJ. TRACK. + INV. DYNAMICS + FORCE CONTROL



FEEDBACK : low gain PID

FEEDFORWARD : support consistent inverse kinematics

↓

COMPLIANT LOCOMOTION

SUPPORT CONSISTENT INVERSE DYNAMICS CONTROL

STACKED OBJECTIVES

SWING LEG $\ddot{r}_{OF} = J_F \ddot{q} + \dot{J}_F \dot{q}$

BASE MOTION $\dot{\omega}_B = J_B \ddot{q} + \dot{J}_B \dot{q}$

CONTACT CONSTRAINT $\ddot{r}_c = J_c \ddot{q} + \dot{J}_c \dot{q} = 0$

$$\rightarrow \quad \ddot{q}_{des} = \begin{bmatrix} J_F \\ J_B \\ J_c \end{bmatrix}^{+} \left(\begin{pmatrix} \ddot{r}_{OF}^{*} \\ \dot{\omega}_B^{*} \\ 0 \end{pmatrix} - \begin{bmatrix} \dot{J}_F \\ \dot{J}_B \\ \dot{J}_c \end{bmatrix} \dot{q} \right) \quad \rightarrow \quad \tau = \left(\hat{N}_c^T S^T \right)^{+} \hat{N}_c^T \left(\hat{M} \ddot{q}_{des} + \hat{b} + \hat{g} \right)$$

support consistent τ, given $\ddot{q}_{des}$

MULTI-TASK SEQUENTIAL QP

Solve locomotion as a sequence of optimization problems of decreasing priority

- Step 1: move base $\min_{\ddot{\mathbf{q}}} \left\| \dot{\mathbf{w}}_{B,des}(t) - \mathbf{J}_B \ddot{\mathbf{q}} - \dot{\mathbf{J}}_B \dot{\mathbf{q}} \right\|$

  s.t.
  - $\mathbf{M}\ddot{\mathbf{q}} + \mathbf{b} + \mathbf{g} + \mathbf{J}_c^T \mathbf{F}_c = \mathbf{S}^T \boldsymbol{\tau}$ <= equation of motion holds
  - $\mathbf{J}_C \ddot{\mathbf{q}} + \dot{\mathbf{J}}_C \dot{\mathbf{q}} = \mathbf{0}$ <= contact constraint holds
  - $\mathbf{F}_{c,n_i} > F_{n,\min}$ <= minimal normal contact force
  - $\mu \mathbf{F}_{c,n_i} > \left\| \mathbf{F}_{c,t_i} \right\|_2$ <= contact force in friction cone

optimizes the motion given motion and contact constraints that cannot be violated

- Step 2: move swing leg $\min_{\ddot{\mathbf{q}}} \left\| \ddot{\mathbf{r}}_{OF,des}(t) - \mathbf{J}_F \ddot{\mathbf{q}} - \dot{\mathbf{J}}_F \dot{\mathbf{q}} \right\|$

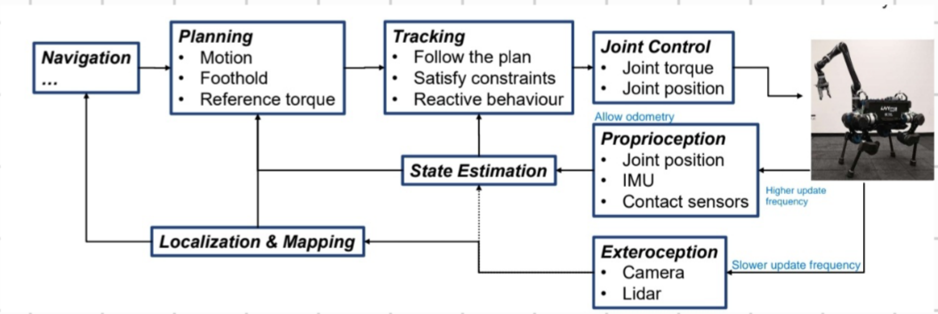
  s.t.
  - $c_1 = \ddot{\mathbf{x}}_{B,des}(t) - \mathbf{J}_B \ddot{\mathbf{q}} - \dot{\mathbf{J}}_B \dot{\mathbf{q}}$ <= higher priority task is not influenced
  - $\mathbf{M}\ddot{\mathbf{q}} + \mathbf{b} + \mathbf{g} + \mathbf{J}_c^T \mathbf{F}_c = \mathbf{S}^T \boldsymbol{\tau}$ <= equation of motion holds
  - $\mathbf{F}_{c,n_i} > F_{n,\min}$ <= minimal normal contact force
  - $\mu \mathbf{F}_{c,n_i} > \left\| \mathbf{F}_{c,t_i} \right\|_2$ <= contact force in friction cone

the prior task's cost now becomes a constraint that cannot be violated

- Last step: minimize e.g torque $\min \|\boldsymbol{\tau}\|$ or tangential contact forces $\min \left\| \mathbf{F}_{s,t_i} \right\|$

  s.t. all other task are still fulfilled

ANYMAL - CASE STUDY

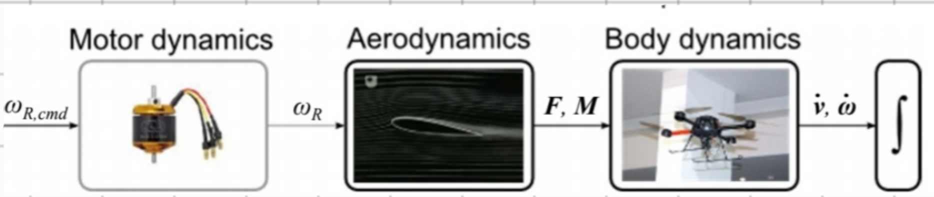


#no summary

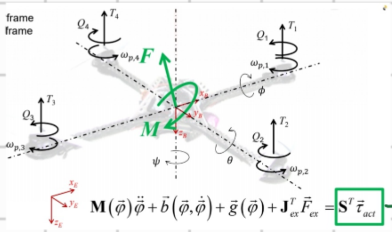
ROTOCRAFTS

MODELLING

ASSUMPTION
- origin ≡ CoG
- structures are rigid
- fuselage drag neglected



we need to know the dimensions as well as the mass and inertia of the quadrotor



the forces and torques from the 4 propellers can be combined in a unique force and torque applied in the origin

$$\mathbf{M}(\vec{\varphi})\ddot{\vec{\varphi}}+\vec{b}(\vec{\varphi},\dot{\vec{\varphi}})+\vec{g}(\vec{\varphi})+\mathbf{J}_{ex}^T\vec{F}_{ex}=\mathbf{S}^T\vec{\tau}_{act}$$

$\ddot{\vec{\varphi}}$	Generalized coordinates
$\mathbf{M}(\vec{\varphi})$	Mass matrix
$\vec{b}(\vec{\varphi},\dot{\vec{\varphi}})$	Centrifugal and Coriolis forces
$\vec{g}(\vec{\varphi})$	Gravity forces
$\vec{\tau}_{act}$	Actuation torque
$\vec{F}_{ex}$	External forces (end-effector, ground contact, propeller thrust, ...)
$\mathbf{J}_{ex}$	Jacobian of external forces
$\mathbf{S}$	Selection matrix of actuated joints