△ Inverse Kinematics

• Recall: forward kinematics

$T_{IE}(q) = \begin{bmatrix} C_{IE}(q) & {}_I r_{IE}(q) \\ 0 & 1 \end{bmatrix} \; SE(3)$

parametrize

$\chi_e = \begin{pmatrix} \chi_{P}(q) \\ \chi_{R}(q) \end{pmatrix} = f(q)$

$\chi_e = \begin{pmatrix} \chi_{eP} \\ \chi_{eR} \end{pmatrix}$

• Q: $q = f^{-1}(\chi_E)$? give you χ_E, give me q

△ sol^n

• Analytic :
  - for 3 intersecting neighbouring axes
• Geometric :
  - use length, theorem ... "geometric insight"
• Algebraic
  - use TFs to get joints
• numerical!

△ numerical method :
inverse differential kinematics

• recall

$w_e = J_{e} \dot{q}$

$\dot{q} = J_{e0}^{+} w_e^*$ — pseudo inverse; multiple sol; singular

• singularities
  - occur @ q_s ⇒ $J_e(q_s)$ is column deficient
  - $\dot{q} = J_{e0}^{-1} w_e^*$ — inf ← column deficient ← small
  - categories
    - boundary : when end-arm is @ its limitation; easy to prevent
    - internal : hard to prevent
  - damped version of Moore-Penrose pseudo inverse

$\dot{q} = J^+ w \iff \operatorname{argmin}_{\dot q} \| J\dot{q} - w \|_2$

$\iff \operatorname{argmin}_{\dot q} \| J\dot{q} - w \|_2^2 + \lambda \|w\|_2^2$

$\dot{q} = J^T (J J^T + \lambda^2 I)^{-1} w$

• Redundancy

- $q \in R^n$ $\quad w = J\dot{q}$
  $w \in R^m$
  $J \in R^{m \times n}$ $\quad m > n$ → redundancy

- $J_{e0} \dot{q} = w_e^*$
  ⇒ $J_{e0} (J_{e0}^+ w_e^* + N \dot{q}_0) = w_e^*$
  ⇒ $\dot{q} = J_{e0}^+ w_e^* + N \dot{q}_0$

  $N = \mathcal{N}(J_{e0})$ null space
  $J_{e0} N = 0$

- get N ?
  - $N = I - J_{e0}^+ J_{e0}$
  - SVD
  - QR

  end-up w/ different basis that span Null space

△ multi-task control
- track desired positions & orientations using kinematics
- break down tasks : $\{J_i, w_i^*\}$

- $\dot{q} = \begin{bmatrix} J_1 \\ \vdots \\ J_{n_t} \end{bmatrix}^+ \begin{bmatrix} w_1^* \\ \vdots \\ w_{n_t}^* \end{bmatrix}$
  $\qquad \bar{J} \qquad \bar{w}$

  doesn't necessarily map the end-effector state space. it's just an objective space. $\bar J$ is the mapping vector

  ⇒ $\min \| \bar{J} \dot{q} - \bar{w} \|_2$
  ⇒ $\min \| \dot{q} \|_2^2$
  s.t. $\bar{J}\dot{q} = \bar{w}$

- weighting $\quad W = \begin{bmatrix} w_1 & & \\ & \ddots & \\ & & w_{n_t} \end{bmatrix}$

  $\bar{J}^{+W} = (\bar{J}^T W \bar{J})^{-1} \bar{J}^T W$

- Prioritization

  recall: $\dot{q} = J_1^+ w_1^* + N_1 \dot{q}_0$

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

  ∴ $\dot{q} = \sum_{i=1}^{n_t} N_i \dot{q}_i$

  w/ $\dot{q}_i = (J_i \bar{N}_i)^+ (w_i^* - J_i \sum_{k=1}^{i-1} N_k \dot{q}_k)$

△ Back to inverse kinematics

- $\dot{\chi}_e = J_{eA}(q)\dot{q}$ — analytical
  $w = J_0 \dot{q}$ — geometric

- now $\Delta \chi_e = J_{eA}(q) \Delta q$

- tracking a point χ_e^*, $q_0 = q^0$
  pseudo-code

  $q \leftarrow q^0$
  while $\| \chi_e^* - \chi_e(q) \| \geq tol$ do
  - $J_{eA} \leftarrow \frac{\partial \chi_e}{\partial q}(q)$
  - $J_{eA}^+ \leftarrow (J_{eA})^+$
  - $\Delta\chi_e \leftarrow \chi_e^* - \chi_e(q)$
  - $q \leftarrow q + J_{eA}^+ \Delta\chi_e$

  (usually joint-space bigger than task-space → redundant system)

△ iterative method

• problem 1

$q^{i+1} = q^i + J_{eA}^+ \Delta\chi$ — if $\Delta\chi$ ↑ & J_{eA} is ill-conditioned

∴ $q^{i+1} = q^i + k J_{eA}^+ \Delta\chi$ — slower converge

• problem 2
  Jacobian is rank deficient → bad condition

∴ $q^{i+1} = q^i + J_{eA}^T (J_{eA} J_{eA}^T + \lambda^2 I)^{-1} \Delta\chi$
or $q^{i+1} = q^i + \alpha J_{eA}^T \Delta\chi$

△ Orientation
- depends on parametrization
- $\in SO(3)$
- $\Delta\chi_{rot} \neq \Delta\varphi$

$C_{GS}(\Delta\varphi) = C_{GS}(\varphi^*) C_{CS}^T(\varphi^t)$

⇒ $q \leftarrow q + k_{pR} J_{e0R}^+ \Delta\varphi$ — use geometric one

△ Traj. ctrl.

• given
  $r_e^*(t)$
  $\dot{r}_e^*(t)$

• feedback
  - $\Delta r_e^t = r_e^*(t) - r_e(q^t)$
  - $\dot{q}_P^* = J_{e0P}^+(q^t) \cdot (\dot{r}_e^*(t) + k_{pp} \Delta r_e^t)$
  - $\dot{q}_R^* = J_{e0R}^+(q^t) \cdot (w_e^*(t) + k_{pR} \Delta\varphi_e^t)$

△ L.A. revisit

△ determinants
- $\det(A) = 0$ — squeeze original state space → lower dimension
- $\det(A) = c$ — $\mathrm{Area} = c$; if $\operatorname{sgn}(c) = (-)$ then ...

△ Rank
rank$(A) = n$
then n is the dimension of the spanned space of the output column space

△ Nullspace (Kernel)
$Ax = 0 \qquad A \in R^{m \times n}$
is a set of vectors, which form a space, get "squeeze" to the origin;
note that all A has trivial null space $\{0\}$;
null space $\in R^{n-m}$ if $n > m$; ...

△ ill/well-condition of matrix
$Ax = c$

ill conditioned ? / well conditioned ? } small change in $[A]$ → small/large change in $[c]$?

vice-versa → small change in $[c]$ → small/large change in $[A]$?

- small ΔA → small Δc "well-conditioned"
- small ΔA → large Δc "ill-conditioned"

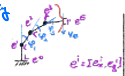
e.g. ...

- condition number
  cond$[A] \approx 1.0$ well
  cond$[A] > 1.0$ ill

  cond$[A] = \|A\|_\infty \|A^{-1}\|_\infty$

Rule of thumb : $\log[\mathrm{cond}[A]]$ ≈ decimals of accuracy lose

△ inverse kinematics case study



(1) given $q_0 = [\pi/6, \pi/3, \pi/3]^T$
${}_I r_{IE,z} = [1, 1]^T$
find $\dot{q}$ (generalized velocity)

$w = J\dot{q}$ (assume $l_i = 1$)

- $r_E = \begin{bmatrix} \ldots \\ \ldots \end{bmatrix}$

⇒ $J_e = \begin{bmatrix} \frac{\partial r_1}{\partial q_1} & \frac{\partial r_1}{\partial q_2} & \frac{\partial r_1}{\partial q_3} \\ \frac{\partial r_2}{\partial q_1} & \frac{\partial r_2}{\partial q_2} & \frac{\partial r_2}{\partial q_3} \end{bmatrix}$

⇒ w/ $q = [\pi/6, \pi/3, \pi/3]^T$

⇒ $\dot{q} = J_e^+ w$
$= J^T (J J^T)^{-1} [\,]$
$= \begin{bmatrix} 0.069 \\ -0.56 \\ -0.075 \end{bmatrix}$

check!

(2) multi-task
we want $q_1 = q_2 = 0$

- parallel (no priority)

⇒ $\dot{q} = \begin{bmatrix} -0.133 \\ -0.067 \\ -1.132 \end{bmatrix}$

check!

(3) w/ priority, task 1 most important!

⇒ $\dot{q} = J_1^+ w_1 + N_1 \dot{q}_0$
⇒ $J_2 \dot{q} = J_2 J_1^+ w_1 + J_2 N_1 \dot{q}_0 = w_2$
⇒ $\dot{q}_0 = (J_2 N_1)^+ (w_2 - J_2 J_1^+ w_1)$
⇒ $\dot{q} = J_1^+ w_1 + N_1 \ldots$
$= \begin{bmatrix} -0.164 \\ -0.485 \\ -1.070 \end{bmatrix}$

check!

△ matrix norm

$[A] = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}$

$\|A\|_\infty = \max\{ |1| + |2|, |0| + |1| \}$ — row sum w/ abs.
$= \max\{3, 1\} = 3$

$\|A\|_1$ = sum of row 1 = 3
$\|A\|_2$ = sum of row 2 = 1

e.g. recall cond$[A] = \|A\|_\infty \|A^{-1}\|_\infty$

⇒ cond$(A) = 3603$

△ more on matrix norm
• recall vector norms
  L_1-norm $\|u\|_1 = |x_1| + |x_2| + \ldots$
  L_2-norm $\|u\|_2 = \sqrt{x_1^2 + x_2^2}$
• matrix norms "based on" vector norm
  matrix → linear transformation
  ∴ matrix norm : "HOW BIG MY MATRIX CAN GET MY OUTPUT VECTOR".

$\|A\|_p = \max_{\|u\|_p = 1} \|Au\|_p$

• $\|AB\|_p \leq \|A\|_p \|B\|_p$
• $\|A^k\|_p = \max_{\|u\|_p = 1} \|A^k u\|_p$

$Au = \lambda u$
⇒ $\|Au\|_p = |\lambda| \|u\|_p \leq \|A\|_p \|u\|_p$
⇒ $|\lambda| \leq \|A\|_p$
⇒ $\max |\lambda_i| \leq \|A\|_p$
⇒ $\max |\lambda_i|^k \leq \|A^k\|_p$

△ Dynamics

- kinematic v dynamic
  kinematic: don't care about the cause, let me know the relation!
  dynamic: what's the source — source of motion; kinematic: description of motion
- cause of motion

  $\tau \longrightarrow \ddot{q}$
  cause → description of motion!

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

- Newton-Euler (free cut, conservation of impulse & angular momentum)
- Projected Newton-Euler (generalized coordinates)
- Lagrange II (energy)

△ Principle of virtual work (D'Alembert's Principle)

• Dynamic equilibrium ⇒ 0 virtual work

$\delta W = \int_B \delta r^T (\ddot{r}\, dm - dF_{ext}) = 0$

• Newton's law for every particle
force $F = ma$

⇒ $F = \frac{d}{dt}(mv) = \dot{p}$ — p momentum (linear impulse)

$r \times F = r \times ma$
$\Gamma = \frac{d}{dt}(r \times mv) = \dot{N}$ — N angular momentum

△ Derivation of virtual displacements of single rigid bodys

- from rigid body kinematics

$r = r_S + p \qquad [\Omega]_\times p = -[p]_\times \Omega$
$\dot{r} = v_S + \Omega \times p = [I_{3\times3} \; -[p]_\times] \begin{bmatrix} v_S \\ \Omega \end{bmatrix}$
$\ddot{r} = a_S + \psi \times p + \Omega \times (\Omega \times p)$
$= [I_{3\times3} \; -[p]_\times] \begin{bmatrix} a_S \\ \psi \end{bmatrix} + [\Omega]_\times [\Omega]_\times p$
$\delta r = \delta r_S + \delta\Phi \times p$
$= [I_{3\times3} \; -[p]_\times] \begin{bmatrix} \delta r_S \\ \delta\Phi \end{bmatrix}$

- Apply principle of virtual work

$\delta W = \int_B \delta r^T (\ddot{r}\, dm - dF_{ext}) = 0$

$= \int_B \begin{pmatrix} \delta r \\ \delta\Phi \end{pmatrix}^T \begin{pmatrix} I_{3\times3} \\ [p]_\times \end{pmatrix} \cdot \left(\left\{ [I_{3\times3} \; -[p]_\times] \begin{bmatrix} a_S \\ \psi \end{bmatrix} + [\Omega]_\times [\Omega]_\times p \right\} dm - dF_{ext} \right)$

$= \begin{pmatrix} \delta r \\ \delta\Phi \end{pmatrix}^T \int_B \left(\begin{bmatrix} I\, dm & [p]_\times^T dm \\ [p]_\times dm & -[p]_\times^2 dm \end{bmatrix} \begin{pmatrix} a_S \\ \psi \end{pmatrix} + \begin{pmatrix} [\Omega]_\times^2 p\, dm \\ [p]_\times [\Omega]_\times^2 p\, dm \end{pmatrix} - \begin{pmatrix} dF_{ext} \\ [p]_\times dF_{ext} \end{pmatrix} \right)$

w/ $\int_B dm = m$
$\int_B p\, dm = 0 \qquad S = COG$ moment.
$\int_B -[p]_\times^2 dm = \int_B [p]_\times [p]_\times^T dm = \Theta_S$ — moment of inertia

⇒ $0 = \delta W = \begin{pmatrix} \delta r_S \\ \delta\Phi \end{pmatrix}^T \left(\begin{bmatrix} I_{3\times3} m & 0 \\ 0 & \Theta_S \end{bmatrix} \begin{bmatrix} a_S \\ \psi \end{bmatrix} + \begin{pmatrix} 0 \\ [\Omega]_\times \Theta_S \Omega \end{pmatrix} - \begin{pmatrix} F_{ext} \\ T_{ext} \end{pmatrix} \right)$

$\forall \begin{pmatrix} \delta r_S \\ \delta\Phi \end{pmatrix}$ → ★ free-body can move in all directions

⇒ means that

$\begin{cases} m a_S - F_{ext} = 0 \\ \Theta_S \psi + [\Omega]_\times \Theta_S \Omega - T_{ext} = 0 \end{cases}$

note that

$\begin{cases} p_S = m v_S & \text{momentum} \\ \dot{p}_S = m a_S \; (f) & \text{change in momentum} \end{cases}$

$\begin{cases} N_S = \Theta_S \Omega_S & \text{angular momentum} \\ \dot{N}_S = \Theta_S \psi + \Omega \times \Theta_S \Omega & \text{change in angular momentum} \end{cases}$

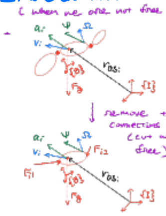
⇒ $\begin{cases} \dot{p}_S = F_{ext} & \text{force} & \text{newton} \\ \dot{N}_S = T_{ext} & \text{torque} & \text{euler} \end{cases}$

"change in linear/angular momentum"

△ some reminder

	linear	rotational
p	x	θ
v	v	ω
a	a	α
motion	$x = vt$	$\theta = \omega t$
	$v = v_0 + at$	$\omega = \omega_0 + \alpha t$
	$x = v_0 t + \frac{1}{2} a t^2$	$\theta = \omega_0 t + \frac{1}{2}\alpha t^2$
	$v^2 = v_0^2 + 2ax$	$\omega^2 = \omega_0^2 + 2\alpha\theta$
mass	m	I
	$F = ma$	$\tau = I\alpha$
	$P = mv$	$L = I\omega$
work	Fd	$\tau\theta$
	$\frac{1}{2} m v^2$	$\frac{1}{2} I \omega^2$
power	Fv	$\tau\omega$

△ More on Newton-Euler method
(when we are not free body)



↓ remove the connections (cut all bodies free) → produce constraining forces

• based on this, w/ n rigid bodies
  - 6n equation
  - 5n constrained force

why 5n: if 1 DOF joints you are imposing 5 DOF constraints in each joint

△ projected Newton-Euler
(in generalized motion direction)

- recall

$0 = \delta W = \sum_{i=1}^{n_b} \begin{pmatrix} \delta r_{S_i} \\ \delta\Phi_{S_i} \end{pmatrix}^T \left(\begin{bmatrix} \dot{p}_{S_i} \\ \dot{N}_{S_i} \end{bmatrix} - \begin{bmatrix} F_{ext,i} \\ T_{ext,i} \end{bmatrix} \right) \quad \forall \begin{pmatrix} \delta r_{S_i} \\ \delta\Phi_{S_i} \end{pmatrix}$

now, this is limited due to linkages!

- now express in generalized coordinates to represent the joints
- express linear/angular momentum in generalized coordinates

$\begin{cases} \begin{bmatrix} v_S \\ \Omega \end{bmatrix} = \begin{bmatrix} J_P \\ J_R \end{bmatrix} \dot{q} \\ \begin{bmatrix} a_S \\ \psi \end{bmatrix} = \begin{bmatrix} J_P \\ J_R \end{bmatrix} \ddot{q} + \begin{bmatrix} \dot{J}_P \\ \dot{J}_R \end{bmatrix} \dot{q} \end{cases}$

⇒ $\begin{bmatrix} \dot{p}_{S_i} \\ \dot{N}_{S_i} \end{bmatrix} = \begin{bmatrix} m a_{S_i} \\ \Theta_{S_i} \psi_{S_i} + \Omega_{S_i} \times \Theta_{S_i} \Omega_{S_i} \end{bmatrix}$
$= \begin{pmatrix} m J_{S_i} \\ \Theta_{S_i} J_{R_i} \end{pmatrix} \ddot{q} + \begin{pmatrix} m \dot{J}_{S_i} \dot{q} \\ \Theta_{S_i} \dot{J}_{R_i} \dot{q} + J_{R_i} \dot{q} \times \Theta_{S_i} J_{R_i} \dot{q} \end{pmatrix}$

- virtual displacement in generalized coordinates

$\begin{pmatrix} \delta r_S \\ \delta\Phi \end{pmatrix} = \begin{bmatrix} J_P \\ J_R \end{bmatrix} \delta q$

∴ $0 = \delta W = \sum_{i=1}^{n} \begin{pmatrix} \delta r_{S_i} \\ \delta\Phi_{S_i} \end{pmatrix}^T \left(\begin{bmatrix} \dot{p}_{S_i} \\ \dot{N}_{S_i} \end{bmatrix} - \begin{bmatrix} F_{ext,i} \\ T_{ext,i} \end{bmatrix} \right)$

$= \delta q^T \sum_{i=1}^{n} \Big\{ \underbrace{\begin{pmatrix} J_{S_i} \\ J_{R_i} \end{pmatrix}^T \begin{pmatrix} m J_{S_i} \\ \Theta_{S_i} J_{R_i} \end{pmatrix}}_{M(q)} \ddot{q} + \underbrace{\begin{pmatrix} J_{S_i} \\ J_{R_i} \end{pmatrix}^T \begin{pmatrix} m \dot{J}_{S_i} \dot{q} \\ \Theta_{S_i} \dot{J}_{R_i} \dot{q} + J_{R_i} \dot{q} \times \Theta_{S_i} J_{R_i} \dot{q} \end{pmatrix}}_{b(q, \dot{q})} - \underbrace{\begin{pmatrix} J_{S_i} \\ J_{R_i} \end{pmatrix}^T \begin{pmatrix} F_{ext,i} \\ T_{ext,i} \end{pmatrix}}_{g(q)} \Big\} \quad \forall \delta q$

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

$M = \sum_{i=1}^{n_b} ({}_A J_{S_i}^T m\, {}_A J_{S_i} + {}_B J_{R_i}^T \, {}_B\Theta_{S_i} \, {}_B J_{R_i})$

$b = \sum_{i=1}^{n_b} ({}_A J_{S_i}^T m\, {}_A \dot{J}_{S_i} \dot{q} + {}_B J_{R_i}^T ({}_B\Theta_{S_i} \, {}_B \dot{J}_{R_i} \dot{q} + {}_B\Omega_{S_i} \times {}_B\Theta_{S_i} \, {}_B\Omega_{S_i}))$

$g = \sum_{i=1}^{n_b} (-{}_A J_{S_i}^T \, {}_A F_{g,i})$

△ Lagrange II

• Lagrangian $\mathcal{L} = T - \mathcal{U}$ (T: kinetic, $\mathcal{U}$: potential)

• Lagrangian Equation

$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}} \right) - \frac{\partial \mathcal{L}}{\partial q} = \tau$

• ∵ $\mathcal{U} = \mathcal{U}(q)$
$T = T(q, \dot{q})$

∴ $\underbrace{\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}} \right) - \frac{\partial T}{\partial q}}_{\text{inertial force}} + \underbrace{\frac{\partial \mathcal{U}}{\partial q}}_{\text{gravity vector}} = \tau$

△ Kinetic energy

- joint space $T = \frac{1}{2} \dot{q}^T M \dot{q}$
- bodies space

$T = \sum_{i=1}^{n_b} \left(\frac{1}{2} m_i \, {}_A\dot{r}_{S_i}^T \, {}_A\dot{r}_{S_i} + \frac{1}{2} {}_B\Omega_{S_i}^T \, {}_B\Theta_{S_i} \, {}_B\Omega_{S_i} \right)$

w/ $\dot{r}_{S_i} = J_{S_i} \dot{q}$
$\Omega_{S_i} = J_{R_i} \dot{q}$

∴ $T(q, \dot{q}) = \frac{1}{2} \dot{q}^T \underbrace{\left(\sum_{i=1}^{n_b} (J_{S_i}^T m J_{S_i} + J_{R_i}^T \Theta_{S_i} J_{R_i}) \right)}_{M(q)} \dot{q}$

△ potential energy

- gravity

$F_{g_i} = m_i g_I e_z \Rightarrow \mathcal{U}_g = -\sum_{i=1}^{n_b} r_{S_i}^T F_{g_i}$

- spring

$F_{S_j} = k_j (\|r - r_0\| - d_0) \frac{r - r_0}{\|r - r_0\|}$

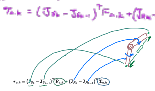
⇒ $\mathcal{U}_{S_j} = \frac{1}{2} k_j (d(q) - d_0)^2$

△ External Force

- given $n_{f,ext}$ $\quad F_j \quad T_k$
- $\tau_{F,ext} = \sum_{j=1}^{n_f} J_{P,j}^T F_j$ — joint space; task space.
- $\tau_{T,ext} = \sum_{k=1}^{n_m} J_{R,k}^T T_k$

actuator torques

$\tau_{a,k} = (J_{S_k} - J_{S_{k-1}})^T F_{a,k} + (J_{R_k} - J_{R_{k-1}})^T T_{a,k}$



△ Dynamic Control

$M(q)\ddot q + b(q,\dot q) + g(q) = \tau + J_c^T F_c$ — how do we get this

Actuator control $\xrightarrow{VA}$ Actuator $\xrightarrow{\tau}$ Robot Dynamics ; torque sensor ; $q, \dot q$; $(q^*, \dot q^*)$

- position-based control
  - don't care about dynamics
  - high gain PID: good ...
  - disturbances are compensated by PID
  - cannot control force directly
  - ⇒ interaction force can only be controlled w/ compliant surface

△ integrate force-feedback (Dynamic control)
- active regulation of system dynamics
- model-based load compensation
- interaction force control.

△ Joint Impedance Control

- $M(q)\ddot q + b(q,\dot q) + g(q) = \tau$
- get desired τ

△ torque as function of P/V error

$\tau^* = k_p(q^* - q) + k_d(\dot q^* - \dot q)$ — can think of it as spring force & damping

⇒ $M(q)\ddot q + b(q,\dot q) + g(q) = k_p(q^* - q) + k_d(\dot q^* - \dot q)$

static offset due to gravity (when static $M\ddot q \cdot b = 0$, $\dot q^* - \dot q = 0$)

△ impedance control & gravity compensation

$\tau^* = k_p(q^* - q) + k_d(\dot q^* - \dot q) + \hat g(q)$ — estimated

configuration dependent, eg CoG shifts

△ independent of configuration: inverse dynamics control

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

get $\ddot q^*$, & throw into this EoM, and get the final τ. based on inverse kinematics → inverse dynamics
- assume our dynamic modelling ... result in $\ddot q = \ddot q^* = k_p(q^* - q) + k_d(\dot q^* - \dot q)$ & $\omega = \sqrt{k_p}$, $D = \frac{k_d}{2\sqrt{k_p}}$

- describe from task space

$\dot w_e = \begin{pmatrix}\dot r\\ \dot\omega\end{pmatrix}_e = J_e \ddot q + \dot J_e \dot q$

∴ $\ddot q = J_e^+(\dot w_e - \dot J_e \dot q)$

& similarly, multi-task

- $\ddot q = \begin{bmatrix}J_1\\ \vdots\\ J_{n_T}\end{bmatrix}^+ \left(\begin{bmatrix}\dot w_1\\ \vdots\\ \dot w_{n_T}\end{bmatrix} - \begin{bmatrix}\dot J_1\\ \vdots\\ \dot J_{n_T}\end{bmatrix}\dot q\right)$ — parallel

- $\ddot q = \sum_{i=1}^{n_T} N_i \ddot q_i$ w/ $\ddot q_i = (J_i N_i)^+\left(\dot w_i^* - \dot J_i \dot q - J_i \sum_{k=1}^{i-1} N_k \ddot q_k\right)$

- get $\ddot q$, & insert back to E.o.M.

△ task-space dynamics

- recall joint space: $M(q)\ddot q + b(q,\dot q) + g(q) = \tau$
- for end-effector: $\Lambda \dot w_e + \mu + p = F_e$

$\begin{cases}\tau = J_e^T F_e \\ \dot w_e = \begin{pmatrix}\dot r\\ \dot\omega\end{pmatrix}_e = J_e \ddot q + \dot J_e \dot q\end{cases}$; $\ddot q = M^{-1}(\tau - b - g)$

derivation

⇒ $\dot w_e = J_e M^{-1}(\tau - b - g) + \dot J_e \dot q$

⇒ $\dot w_e - \dot J_e \dot q + J_e M^{-1} b + J_e M^{-1} g = J_e M^{-1}\tau$

⇒ $\dot w_e - \dot J_e \dot q + J_e M^{-1} b + J_e M^{-1} g = J_e M^{-1} J_e^T F_e$ — μ, p, Λ^{-1}

⇒ $\Lambda \dot w_e + \mu + p = F_e$

geometric interpretation: inertial ellipsoid (depend on our configuration)

∴ get $\dot w_e^* = k_p E(\chi_e^* - \chi_e) + k_d(w_e^* - w_e) + \dot w_e(\tau)$ — feed-forward

determine desired acceleration → task-level force → Jacobi transformation → joint-space torques

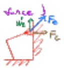
$\tau^* = \hat J^T(\hat\Lambda \dot w_e^* + \hat\mu + \hat p)$

△ interaction control

operation space control

(generalized framework to control motion & force)

- $F_c + \Lambda \dot w_e + \mu + p = F_e$



△ selection force

$\hat\Lambda S_M \dot w_e + S_F F_c + \hat\mu + \hat p = F_e$

⇒ $\tau^* = \hat J^T F_e$

$\begin{cases}\Sigma_p = \begin{bmatrix}\sigma_{px} & 0 & 0\\ 0 & \sigma_{py} & 0\\ 0 & 0 & \sigma_{pz}\end{bmatrix} \\ \Sigma_F = I_3 - \Sigma_p\end{cases}$ $\sigma \in \{0,1\}$; 1 → can move; 0 → can apply force

$\begin{cases}\Sigma_R = \begin{bmatrix}\sigma_{Rx} & 0 & 0\\ 0 & \sigma_{Ry} & 0\\ 0 & 0 & \sigma_{Rz}\end{bmatrix} \\ \Sigma_{RF} = I_3 - \Sigma_R\end{cases}$

w/ C. transformation between contact local frame ↕ world frame

∴ $S_M = \begin{bmatrix}C^T\Sigma_p C & 0\\ 0 & C^T\Sigma_R C\end{bmatrix}$

$S_F = \begin{bmatrix}C^T(I_3 - \Sigma_p)C & 0\\ 0 & C^T(I_3 - \Sigma_R)C\end{bmatrix}$

∴ inject back to

$\hat\Lambda S_M \dot w_e + S_F F_c + \hat\mu + \hat p = F_e$

△ inverse dynamic as QP

1. classic inverse dynamics

recall

$w = J\dot q$

$\dot w = J\ddot q + \dot J \dot q \Rightarrow \ddot q^* = J^+(\dot w - \dot J \dot q)$

∴ $\tau^* = M(q)\ddot q^* + b(q,\dot q) + g$
$= M(q)[J^+(\dot w - \dot J \dot q)] + b(q,\dot q) + g$

2. operation space control

$\tau^* = J_t^T F_e$
$= J_t^T(\Lambda \dot w_t^* + \mu + p)$ — Not only sel^n

∴ $\tau^* = J_t^T(\Lambda \dot w_t^* + \mu + p) + N(J_t^T)\tau_0$ — null space; can change my ...

$N(J_t^T) = (I - J^T[JM^{-1}J^T]^{-1}JM^{-1})$ — Mass-weighted pseudo inverse; (w.o. changing desired $\dot w$) proof

$\dot w = J\ddot q + \dot J \dot q$
$= JM^{-1}(\tau^* - b - g) + \dot J \dot q$
$= JM^{-1}(J_t^T(\Lambda \dot w_t^* + \mu + p) + (I - J^T(JM^{-1}J^T)^{-1}JM^{-1})\tau_0 - b - g) + \dot J \dot q$ — $= 0$

- we can also see:

$\begin{cases}\tau^* = M(q)\ddot q^* + b(q,\dot q) + g \\ \tau^* = J_t^T F_e = J_t^T(\Lambda \dot w_t^* + \mu + p) \\ \quad = J_t^T((J_t M^{-1} J_t^T)^{-1}\dot w + \mu + p)\end{cases}$

⇒ $\ddot q = M^{-1}(\tau^* - b - g)$
$= M^{-1}(J_t^T(J_t M^{-1} J_t^T)^{-1}\dot w + \mu + p) - b - g)$ — some sort of mass-matrix weighted pseudo-inverse

3. QP

- $\begin{rcases}\tau = M\ddot q + b + g \\ J\ddot q + \dot J \dot q = \dot w \\ \min \|\ddot q\| \\ \min \|\tau\|\end{rcases}$ ↔ classic inverse dynamic

- $[M \ \ {-I}]\begin{bmatrix}\ddot q\\ \tau\end{bmatrix} + b + g = 0$
$[J_e \ \ 0]\begin{bmatrix}\ddot q\\ \tau\end{bmatrix} + \dot J \dot q = \dot w_e^*$ — x

⇓

$x = \begin{bmatrix}\ddot q\\ \tau\end{bmatrix}$

multi-task:

$\min_x \left\| \begin{bmatrix}M & -1\\ J_e & 0\end{bmatrix} x - \begin{bmatrix}-b-g\\ \dot w_e^* - \dot J \dot q\end{bmatrix}\right\|_2$

priority:

$\min_x \|[J_e \ \ 0]x - (\dot w_e^* - \dot J \dot q)\|_2$

s.t. $[M \ \ {-I}]x - (-b-g) = 0$

△ Least-square Optimization

- $Ax = b \Rightarrow x = A^+ b$
⇕
$\min_x \|Ax - b\|_2$
or
$\min_x \|x\|_2$
s.t. $Ax = b$

- $A_1x_1 + A_2x_2 = b$
⇒ $[A_1 \ A_2]\begin{bmatrix}x_1\\ x_2\end{bmatrix} = b$ ⇒ $\begin{bmatrix}x_1\\ x_2\end{bmatrix} = [A_1 \ A_2]^+ b$
⇕
$\min_{x_1,x_2} \left\|[A_1 A_2]\begin{bmatrix}x_1\\ x_2\end{bmatrix} - b\right\|_2$
or
$\min_{x_1,x_2} \left\|\begin{bmatrix}x_1\\ x_2\end{bmatrix}\right\|_2$ s.t. $[A_1 A_2]\begin{bmatrix}x_1\\ x_2\end{bmatrix} = b$

- $A_1x - b_1 = 0$
$A_2x - b_2 = 0$
- Equal priority: $x = \begin{bmatrix}A_1\\ A_2\end{bmatrix}^+\begin{bmatrix}b_1\\ b_2\end{bmatrix}$
- Hierarchy
$x = A_1^+ b_1 + N(A_1)x_0$; $\min_x \|A_2x - b_2\|_2$ s.t. $\|A_1x - b_1\| = c$
$A_1(A_1^+ b_1 + N(A_1)x_0)$
$= A_1A_1^+ b_1 + 0 = b_1$
⇒ $A_2x - b_2 = 0$
⇒ $A_2(A_1^+ b_1 + N(A_1)x_0) - b_2 = 0$
⇒ $x_0 = [A_2 N(A_1)]^+[b_2 - A_2A_1^+ b_1]$ — try to fulfill $A_2x = b_2$

△ case study

△ solving a set of QPs

- w/ different priorities

methods:
- exploit Nullspace of tasks w/ higher priority
- everystep = QP w/ constraints
- iteratively nullspace projection

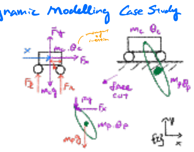
- pseudo-code

n_T = number of tasks
$x = 0$
$N_1 = I$
for $i = 1 \rightarrow n_T$ do
  $x_i = (A_iN_i)^+(b_i - A_ix)$
  $x = x + N_ix_i$
  $N_{i+1} = N([A_1^T \cdots A_i^T]^T)$
endfor

e.g. @ $i = 1$
$x = 0$
$N_1 = I$
$x_1 = (A_1 I)^+(b_1 - A0)$
$= A_1^+ b_1$
$x := 0 + IA_1^+ b_1$
$N_2 = N(A_1)$
@ $i = 2$
$x = A_1^+ b_1$
$N_2 = N(A_1)$
$x_2 = [A_2 N(A_1)]^+[b_2 - A_2A_1^+ b_1]$
$x := A_1^+ b_1 + N(A_1)[A_2 N(A_1)]^+[b_2 - A_2A_1^+ b_1]$

→ use solver (most of the time)

△ Dynamic Modelling Case Study



△ get linear / angular momentum of cart

$m_c \ddot x_c = F_x$
$m_c \ddot y_c = F_y + F_l + F_r - m_c g$
$\theta_c \ddot\varphi_c = F_r b - F_l b$

△ get linear / angular momentum of pendulum

$m_p \ddot x_p = -F_x$
$m_p \ddot y_p = -F_y - m_p g$
$\theta_p \ddot\varphi_p = F_x l\cos(\varphi_p) - F_y l\sin(\varphi_p)$

△ kinematics

$x_c = x$
$y_c = 0$
$\varphi_c = 0$

$[x, \varphi]^T$ ⇒ generalized coordinates

$x_p = x + l\sin(\varphi)$
$y_p = -l\cos(\varphi)$
$\varphi_p = \varphi$

$\ddot x_p = \ddot x + \ddot\varphi l\cos(\varphi) - \dot\varphi^2 l\sin(\varphi)$
$\ddot y_p = \ddot\varphi l\sin(\varphi) + \dot\varphi^2 l\cos(\varphi)$
$\ddot\varphi_p = \ddot\varphi$

- (7),(10-12) in (1) and (4-6)

$m_c\ddot x = F_x$ (13)

$m_p\left(\ddot x + \ddot\varphi l\cos(\varphi) - \dot\varphi^2 l\sin(\varphi)\right) = -F_x$ (14)

$m_p\left(\ddot\varphi l\sin(\varphi) + \dot\varphi^2 l\cos(\varphi)\right) = -F_y - m_pg$ (15)

$\theta_p\ddot\varphi = F_x l\cos(\varphi) + F_y l\sin(\varphi)$ (16)

- From (13) and (14) remove F_x

$(m_p + m_c)\ddot x + m_p l\cos(\varphi)\ddot\varphi - \dot\varphi^2 l m_p\sin(\varphi) = 0$ ✱

- Insert (13) and (15) in (16) to remove F_x and F_y

$(\theta_p + m_p l^2)\ddot\varphi + m_p l\cos(\varphi)\ddot x + g l m_p\sin(\varphi) = 0$ ✱