△ Inverse Kinematics

- Recall: forward kinematics

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

parameterize

$$\chi_e = \begin{pmatrix} r_e(q) \\ \phi_e(q) \end{pmatrix} = f(q)$$

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

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

△ sol^n

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

△ numerical method: inverse differential kinematics

- recall

$w_e = J_{e0}\,\dot q$

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

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

$\dot q = J^{+} w \Leftrightarrow \arg\min_{\dot q} \| J\dot q - w \|_2$

$\Leftrightarrow \arg\min_{\dot q} \| J\dot q - w\|_2^2 + \lambda^2 \|\dot q\|_2^2$

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

- Redundancy $\quad w = J\dot q$
  - $q \in R^n$
  - $w \in R^m$
  - $J \in R^{m \times n}$ $\quad m > n$ → redundancy
- $J_{e0}\,\dot q = w_e^*$
  - $\Rightarrow J_{e0}(J_{e0}^{+} w_e^* + N\dot q_0) = w_e^*$
  - $\Rightarrow \dot q = J_{e0}^{+} w_e^* + N \dot q_0$

$N = \mathcal{N}(J_{e0})$ 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 position & orientation
- ensure kinematics
- break down tasks; $\{J_i, w_i^*\}$

$$\dot q = \begin{bmatrix} J_1 \\ \vdots \\ J_{n_t} \end{bmatrix}^{+} \begin{bmatrix} w_1^* \\ \vdots \\ w_{n_t}^* \end{bmatrix}$$

$\bar J \qquad \bar w$

doesn't necessarily mean the end-effector. it's just an objective space. J is the mapping

$\Rightarrow \min \| \bar J \dot q - \bar w \|_2$

$\Rightarrow \min \| \dot q \|_2^2$ s.t. $\bar J \dot q = \bar w$

- weighting $\quad \begin{bmatrix} w_1 & \\ & \ddots \end{bmatrix}$

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

- Prioritization

recall: $\dot q = J_1^{+} w_1^* + N_1 \dot q_0$

$\Rightarrow w_2 = J_2 \dot q$

$= J_2 (J_1^{+} w_1^* + N_1 \dot q_0)$

$\Rightarrow \dot q_0 = (J_2 N_1)^{+}(w_2^* - J_2 J_1^{+} w_1^*)$

$\Rightarrow \dot q = J_1^{+} w_1^* + N_1 [J_2 N_1]^{+} [w_2^* - J_2 J_1^{+} w_1^*]$ $\quad \dot q_0 \hookrightarrow J_2 \dot q \cong w_2$

$\therefore \dot q = \sum_{i=1}^{n_T} N_i \dot q_i$

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

△ Back to inverse Kinematics

- $\dot\chi_E = J_{eA}(q)\,\dot q$ — analytical (always dependent on parameterization)
- $w = J_0\,\dot q$ — geometric
- now $\Delta \chi_e = J_{eA}(q)\,\Delta q$
- tracking a point χ_e^*, $q_0 = q^0$

pseudo-code

```
q ← q^0
while ||χ_e* − χ_e(q)|| ≥ tol do
    J_eA ← ∂χ_e/∂q (q)
    J_eA^+ ← (J_eA)^+
    Δχ_e ← χ_e* − χ_e(q)
    q ← q + J_eA^+ Δχ_e
```

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

△ iterative method

- problem 1

$q^{i+1} = q^i + J_{eA}^{+} \Delta X$

if ΔX ↑ & J_{eA} is ill-condition

$\therefore q^{i+1} = q^i + k J_{eA}^{+} \Delta X$ — slower convergence

- problem 2

Jacobian is rank deficient → bad condition

$\therefore q^{i+1} = q^i + J_{eA}^T (J_{eA} J_{eA}^T + \lambda^2 I)^{-1} \Delta X$

or $q^{i+1} = q^i + \alpha J_{eA}^T \Delta X$

△ Orientation

- depends on parameterization
- $\in SO(3)$
- $\Delta X_{rotate} = \Delta\varphi$

$C_{GS}(\Delta\varphi) = C_{GI}(\varphi^*)\, C_{SI}^T(\varphi^t)$

$\Rightarrow q \leftarrow q + k_{pR}\, J_{eOR}^{+}\, \Delta\varphi$ — use geometric one (rotation vector)

△ Traj. ctrl.

- given $r_e^*(t)$, $\dot r_e^*(t)$
- feedback
  - $\Delta r_e^t = r_e^*(t) - r_e(q^t)$
  - $\dot q_P^* = J_{eOP}^{+}(q^t)\cdot(\dot r_e^*(t) + k_{pp}\Delta r_e^t)$
  - $\dot q_R^* = J_{eOR}^{+}(q^t)\cdot(w_e(t)^* + k_{pR}\Delta\phi^t)$

△ L.A. revisit

△ determinants

- $\det(A) = 0$ — squeeze original space space → to lower dimension
- $\det(A) = c$ — if: Area = c
- if: $\mathrm{sign}(c) = (-)$ then: "flipped"

$Ax = b$

- if $\det(A) = 0$:
  - A transform x to "lower dimension" cannot unpack. i.e., 逆 transform back
  - Sometimes can! already "lie" on the new "lower dim." → sol^n!
- if $\det(A) \neq 0$:
  - then can reverse lah!

△ Rank

$Ax = b$ — DoF dimension

rank$(A) = n$

then n is the dimension of the spanned space of the output column space = all possible outputs Ax

△ Nullspace (kernel)

$Ax = 0 \quad A \in R^{m \times n}$

↳ a set of vectors, which form a space, whose pt(s) get "squeeze" to the origin;

note that all A has trivial null space $\{0\}$;

null space $\in R^{m \times n}$ if $n > m$; reduced rank

△ ill/well-condition of matrix

$Ax = c$

ill conditioned? well conditioned? } small change in $[A]$

small/large change change in $[c]$?

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

△ small ΔA → small Δc "well-conditioned"

△ small ΔA → large Δc "ill-conditioned"

e.g. $\begin{bmatrix} 400 & -201 \\ -800 & 401 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 200 \\ -200 \end{bmatrix}$ $\quad x = \begin{bmatrix} -100 \\ -200 \end{bmatrix}$

$\begin{bmatrix} 401 & -201 \\ -800 & 401 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 200 \\ -200 \end{bmatrix}$ $\quad x = \begin{bmatrix} 40000 \\ 79800 \end{bmatrix}$

△ condition number

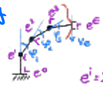
cond$[A] \approx 1.0$ well

cond$[A] > 1.0$ ill

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

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

△ inverse kinematics case study



$e^i = [e_x^i, e_y^i]$

(1) given $q_t = [\pi/6, \pi/3, \pi/3]^T$

${}_0\dot r_{E,t} = [1, 1]^T$

find $\dot q$ (generalized velocity)

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

△ $r_E = \begin{bmatrix} S\varphi_1 + S(\varphi_1+\varphi_2) + S(\varphi_1+\varphi_2+\varphi_3) \\ 1 + C\varphi_1 + C(\varphi_1+\varphi_2) + C(\varphi_1+\varphi_2+\varphi_3) \end{bmatrix}$

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

$= [\ldots]$

$\Rightarrow$ w/ $q = [\pi/6 \;\; \pi/3 \;\; \pi/3]^T$

$J_E = \frac{1}{2}\begin{bmatrix} 0 & -\sqrt3 & -\sqrt3 \\ -4 & -3 & -1 \end{bmatrix}$

$\Rightarrow \dot q = J_E^{+} w$

$= J^T (J J^T)^{-1} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

$= \begin{bmatrix} 0.069 \\ -0.56 \\ -0.595 \end{bmatrix}$

check! $\frac{1}{2}\begin{bmatrix} 0 & -\sqrt3 & -\sqrt3 \\ -4 & -3 & -1 \end{bmatrix}\begin{bmatrix} 0.069 \\ -0.56 \\ -0.595 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

(2) multi-task

we want $\dot q_1 = \dot q_2 = 0$

$J_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$, $w_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

- parallel (no priorities)

$$\dot q = \begin{bmatrix} \frac12\begin{bmatrix} 0 & -\sqrt3 & -\sqrt3 \\ -4 & -3 & -1 \end{bmatrix} \\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \end{bmatrix}^{+} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

$\uparrow R^{3\times1}$ $\quad [R^{4\times3}]^{+}[R^{4\times1}]$

$\Rightarrow \dot q = \begin{matrix} -0.133 \\ -0.067 \\ -1.132 \end{matrix}$

check! $\begin{bmatrix} J_1 \\ J_2 \end{bmatrix}\dot q = \begin{bmatrix} 1.039 \\ 0.993 \\ -0.133 \\ -0.067 \end{bmatrix} \approx \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$

(3) w/ priority, task 1 most important! "$w = J\dot q$"

$\Rightarrow \dot q = J_1^{+} w_1 + N_1 \dot q_0$ $\quad \to I - J_1^{+} J_1$

$\Rightarrow J_2 \dot q = J_2 J_1^{+} w_1 + J_2 N_1 \dot q_0 = w_2$

$\Rightarrow \dot q_0 = (J_2 N_1)^{+} w_2 - J_2 J_1^{+} w_1$

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

$= \begin{bmatrix} -0.169 \\ -0.085 \\ -1.070 \end{bmatrix}$

check! $\begin{bmatrix} J_1 \\ J_2 \end{bmatrix}\dot q = \begin{bmatrix} 1 \\ 1 \\ -0.169 \\ -0.085 \end{bmatrix}$

$J_1 [I - J_1^{+} J_1]$

$= J_1 - J_1 J_1^{+} J_1$

$= 0$

功 work 力 ⇒ force × distance

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

$A = \begin{bmatrix} 400 & -201 \\ -801 & 401 \end{bmatrix}$

$A^{-1} = \begin{bmatrix} -1.003 & -5.01 \\ -2 & 1 \end{bmatrix}$

$\Rightarrow \|A\|_\infty = |-801| + |401| = 1201$

$\|A^{-1}\|_\infty = |-2| + |-1| = 3$

$\Rightarrow$ cond$[A] = 3603$

△ more on matrix norm

- recall vector norms
  - L_1-norm $\quad \|u\|_1 = |u_1| + |u_2|$
  - L_2-norm $\quad \|u\|_2 = \sqrt{u_1^2 + u_2^2}$
- matrix norms "based on" vector norm

matrix → linear transformation

∴ matrix norm: "HOW BIG MY MATRIX CAN GET MY OUTPUT VECTOR".

$\|A\|_2 = \max_{\|u\|_2 = 1} \|A u\|_2$

- $\|AB\|_p \le \|A\|_p \|B\|_p$
- $\|A^k\|_p = \max_{\|u\|_p = 1} \|A^k u\|_p$

$A[u_1 \; u_2 \cdots u_i] = \lambda_i [u_1 \cdots u_i]$

$AU = \lambda U$

$\Rightarrow \|A u_i\|_p = |\lambda_i| \|u_i\|_p \le \|A\|_p \|u_i\|_p$

$\Rightarrow \|\lambda_i\| \le \|A\|_p$

$\Rightarrow \max|\lambda_i| \le \|A\|_p$

$\Rightarrow \max|\lambda_i|^k \le \|A^k\|_p$

△ Dynamics

this chapter basically talks about EoM; description of motion

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

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

- Newton's law for every particle

force $F = ma$

$\Rightarrow F = \frac{d}{dt}(mv) = \dot P$ — 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 body

- from rigid body kinematics

$r = r_s + P \qquad [\Omega]_\times P = -[P]_\times \Omega$

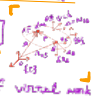
$\dot r = v_s + \Omega \times P = [I_{3\times3} \;\; -[P]_\times] \begin{bmatrix} v_s \\ \Omega \end{bmatrix}$

$\ddot r = a_s + \psi \times P + \Omega \times (\Omega \times P)$

$= [I_{3\times3} \;\; -[P]_\times]\begin{bmatrix} a_s \\ \psi \end{bmatrix} + [\Omega]_\times [\Omega]_\times P$

$\delta r = \delta r_s + \delta\Phi \times P$

$= [I_{3\times3} \;\; -[P]_\times]\begin{bmatrix} \delta r_s \\ \delta\Phi \end{bmatrix}$



- Apply principle of virtual work

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

$= \int_B \begin{pmatrix} \delta r \\ \delta\Phi \end{pmatrix}^T \begin{pmatrix} I_{3\times3} \\ [P]_\times \end{pmatrix}$ $\quad \delta r$

$\cdot \left(\left\{ [I_{3\times3} \;\; -[P]_\times]\begin{bmatrix} a_s \\ \psi \end{bmatrix} + [\Omega]_\times[\Omega]_\times P \right\} dm - dF_{ext} \right)$

$= \begin{pmatrix} \delta r \\ \delta\Phi \end{pmatrix}^T \int_B \left(\begin{bmatrix} I_{3\times3}\,dm & [P]_\times^T dm \\ [P]_\times dm & -[P]_\times^2 dm \end{bmatrix}\begin{pmatrix} a_s \\ \psi \end{pmatrix} + \begin{pmatrix} [\Omega]_\times^2 P\,dm \\ [P]_\times [\Omega]_\times^2 P\,dm \end{pmatrix} - \begin{pmatrix} dF_{ext} \\ [P]_\times dF_{ext} \end{pmatrix} \right)$

w/ $\int_B dm = m$

$\int_B P\,dm = 0 \quad S = COG$ moment.

$\int_B -[P]_\times^2 dm = \int_B [P]_\times [P]_\times^T dm = \Theta_s$ — moment of inertial

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

$\forall \begin{pmatrix} \delta r_s \\ \delta\Phi \end{pmatrix}$ ——→ ※ 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

$$\begin{cases} P_s = m v_s & \text{momentum} \\ \dot P_s = m a_s \;(f) & \text{change in momentum} \end{cases}$$

$$\begin{cases} N_s = \Theta_s \Omega_s & \text{angular momentum} \\ \dot N_s = \Theta_s \psi + \Omega \times \Theta_s \Omega & \text{change in angular momentum} \end{cases}$$

$$\Rightarrow \begin{cases} \dot P_s = F_{ext} & \text{force} & \text{newton} \\ \dot N_s = T_{ext} & \text{torque} & \text{euler} \end{cases}$$

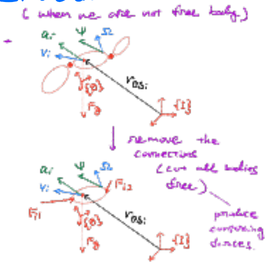
"change in linear / angular momentum"

△ some reminder

	linear	rotational	
@	x	θ	
v	v	ω	
a	a	α	
motion	$x = vt$	$\theta = \omega t$	
	$v = v_0 + at$	$\omega = \omega_0 + \alpha t$	
	$x = v_0 t + \frac12 a t^2$	$\theta = \omega_0 t + \frac12 \alpha t^2$	
	$v^2 = v_0^2 + 2ax$	$\omega^2 = \omega_0^2 + 2\alpha\theta$	
mass	m	I	moment of inertia
	$F = ma$	$\tau = I\alpha$	torque
	$P = mv$	$L = I\omega$	angular momentum
work	Fd	$\tau\theta$	
	$\frac12 m v^2$	$\frac12 I \omega^2$	
power	Fv	$\tau\omega$	

△ More on Newton-Euler method

(when we are not free body)



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

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

一般表達式 → 以 q, generalized coordinates 來表達.

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

△ projected Newton-Euler (in generalized motion direction)

- recall

$0 = \delta W = \sum_{i=1}^{n} \begin{pmatrix} \delta r_{s_i} \\ \delta\Phi_{s_i} \end{pmatrix}^T \left(\begin{bmatrix} \dot P_{s_i} \\ \dot N_{s_i} \end{bmatrix} - \begin{bmatrix} F_{ext,i} \\ T_{ext,i} \end{bmatrix} \right) \quad \forall \begin{pmatrix} \delta r_{s_i} \\ \delta\Phi_{s_i} \end{pmatrix}$

now, this is limited due to linkages!

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

$$\begin{cases} \begin{bmatrix} v_s \\ \Omega \end{bmatrix} = \begin{bmatrix} J_P \\ J_R \end{bmatrix}\dot q \\ \begin{bmatrix} a_s \\ \psi \end{bmatrix} = \begin{bmatrix} J_P \\ J_R \end{bmatrix}\ddot q + \begin{bmatrix} \dot J_P \\ \dot J_R \end{bmatrix}\dot q \end{cases}$$

$\Rightarrow \begin{bmatrix} \dot P_{s_i} \\ \dot N_{s_i} \end{bmatrix} = \begin{bmatrix} m\, a_{s_i} \\ \Theta_{s_i}\psi_{s_i} + \Omega_{s_i} \times \Theta_{s_i}\Omega_{s_i} \end{bmatrix}$

$= \begin{pmatrix} m J_{s_i} \\ \Theta_{s_i} J_{R_i} \end{pmatrix}\ddot q + \begin{pmatrix} m \dot J_{s_i}\dot q \\ \Theta_{s_i}\dot J_{R_i}\dot q + J_{R_i}\dot q \times \Theta_{s_i} J_{R_i}\dot q \end{pmatrix}$

- virtual displacement in generalized coordinates

$\begin{pmatrix} \delta r_s \\ \delta\Phi \end{pmatrix} = \begin{bmatrix} J_P \\ J_R \end{bmatrix}\delta q$ — 心中有 雅克比 multi-body 都能

$\therefore 0 = \delta W = \sum_{i=1}^{n} \begin{pmatrix} \delta r_{s_i} \\ \delta\Phi_{s_i} \end{pmatrix}^T \left(\begin{bmatrix} \dot P_{s_i} \\ \dot N_{s_i} \end{bmatrix} - \begin{bmatrix} F_{ext,i} \\ T_{ext,i} \end{bmatrix} \right)$

$= \delta q^T \sum_{i=1}^{n} \Big\{ \underbrace{\begin{pmatrix} J_{S_i} \\ J_{R_i} \end{pmatrix}^T \begin{pmatrix} m J_{S_i} \\ \Theta_{S_i} J_{R_i} \end{pmatrix}\ddot q}_{M(q)}$

$+ \underbrace{\begin{pmatrix} J_{S_i} \\ J_{R_i} \end{pmatrix}^T \begin{pmatrix} m \dot J_{S_i}\dot q \\ \Theta_{S_i}\dot J_{R_i}\dot q + J_{R_i}\dot q \times \Theta_{S_i} J_{R_i}\dot q \end{pmatrix}}_{b(q,\dot q)}$

$- \underbrace{\begin{pmatrix} J_{S_i} \\ J_{R_i} \end{pmatrix}^T \begin{pmatrix} F_{ext,i} \\ T_{ext,i} \end{pmatrix}}_{g(q)} \Big\} \quad \forall \delta q$

搞了这麼久 我 是為了 用 "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 $\mathcal{L} = T - U$ (kinetic, potential)
- Lagrangian Equation

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

$\because U = U(q)$

$T = T(q, \dot q)$

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

△ Kinetic energy

- joint space $\quad T = \frac12 \dot q^T M \dot q$
- bodies space

$T = \sum_{i=1}^{n_b} \left(\frac12 m_i\, {}_A\dot r_{S_i}^T\, {}_A\dot r_{S_i} + \frac12\, {}_B\Omega_{S_i}^T\, {}_B\Theta_{S_i}\, {}_B\Omega_{S_i} \right)$

w/ $\dot r_{S_i} = J_{S_i}\dot q$

$\Omega_{S_i} = J_{R_i}\dot q$

$\therefore T(q, \dot q) = \frac12 \dot q^T \underbrace{\left(\sum_{i=1}^{n_b} (J_{S_i}^T m J_{S_i} + J_{R_i}^T \Theta_{S_i} J_{R_i}) \right)}_{M(q)} \dot q$

△ potential energy

- gravity

$F_{g_i} = m_i\, g_z\, e_z \Rightarrow U_g = -\sum_{i=1}^{n_b} r_{S_i}^T F_{g_i}$

- spring

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

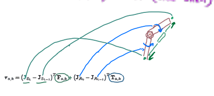
$\Rightarrow U_{E_j} = \frac12 k_j (d(q) - d_0)^2$

△ External Force

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

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 are we E.O.M, how do we get this?

Actuator control $\xrightarrow{V,A}$ Actuator $\xrightarrow{\tau}$ Robot Dynamics

torque sensor

$q, \dot{q}, \ddot{q}$

$(q^*, \dot{q}^*)$

△ position-based control
- don't care about dynamics
- high gain PID : good joint-level tracking
- 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 or damping

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

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

△ impedance control & gravity compensation

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

configuration dependent, eg. CoG shifts — estimated

△ 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 EoD, and get the final τ.

based on inverse kinematics → inverse dynamics

result in
- assume no dynamic modelling error

$\ddot{q} - \ddot{q}^* = k_p(q^* - q) + k_d(\dot{q}^* - \dot{q})$

$\omega = \sqrt{k_p}$ $D = \frac{k_d}{2\sqrt{k_p}}$

- describe from task space

$\dot{w}_e = \begin{pmatrix} \dot{v} \\ \dot{\omega} \end{pmatrix}_e = J_e\ddot{q} + \dot{J}_e\dot{q}$

∴ $\ddot{q} = J_e^+(\dot{w}_e - \dot{J}_e\dot{q})$

& 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/

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

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

linking the two

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

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 $\quad p \quad$ Λ^{-1}

⇒ $\Lambda \dot{w}_e + \mu + p = F_e$

geometric interpretation: inertial ellipsoid (depend on curr. configuration)

∴ get $\dot{w}_e^* = k_p E(\chi_e^* - \chi_e)$ + rotation

$+ k_d(w_e^* - w_e)$

$+ \dot{w}_e(t)$ feed-forward

determine desired acceleration → task-level force

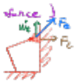
joint-space torques ← Jacobi transformation

$\tau^* = \hat{J}^T(\hat{\Lambda} \dot{w}_e^* + \hat{\mu} + \hat{p})$

△ interaction control

operation space control (generalized framework to control motion & force)

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



△ selection force

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

⇒ $\tau^* = \hat{J}^T F_e$

$\begin{cases} \Sigma_p = \begin{bmatrix} \sigma_{px} & & \\ & \sigma_{py} & \\ & & \sigma_{pz} \end{bmatrix} \quad \sigma \in \{0, 1\} \\ \Sigma_F = I_3 - \Sigma_p \end{cases}$

1 → can move, 0 → can apply force

$\begin{cases} \Sigma_R = \begin{bmatrix} \sigma_{rx} & & \\ & \sigma_{ry} & \\ & & \sigma_{rz} \end{bmatrix} \\ \Sigma_{RF} = I_3 - \Sigma_R \end{cases}$

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

$\tau^* = J_t^T F_e$

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

Not only sol^n

∴ $\tau^* = J_e^T(\Lambda \dot{w}_e^* + \mu + p) + N(J_e^T)\tau_0$

null space — can change my torque

$N(J_e^T)$

$= (I - J^T[(JM^{-1}J^T)^{-1}JM^{-1}])$ (w.o. changing desired $\dot{w}$)

Mass weighted pseudo inverse

proof

$\dot{w} = J\ddot{q} - \dot{J}\dot{q}$

$= JM^{-1}(\tau^* - b - g) + \dot{J}\dot{q}$

$= JM^{-1}(J^T(\Lambda \dot{w}_e^* + \mu + p)$

$+ (I - J^T(JM^{-1}J^T)^{-1}JM^{-1})\tau_0$ = 0

$- b - g)$

$+ \dot{J}\dot{q}$

- we can also see:

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

⇒ $\ddot{q} = M^{-1}(\tau^* - b - g)$

$= M^{-1}(J_e^T(J_e M^{-1} J_e)^{-1}\dot{w} + \mu + p) - b - g)$

some sort of mass-matrix weighted pseudo-inverse

3. QP

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

- $[M \; -I]\begin{bmatrix} \ddot{q} \\ \tau \end{bmatrix} + b + g = 0$ — x

$[J_e \; 0]\begin{bmatrix} \ddot{q} \\ \tau \end{bmatrix} + \dot{J}\dot{q} = \dot{w}_e^*$ — 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. $Ax = b$

- $A_1 x_1 + A_2 x_2 = b$

⇒ $[A_1 \; A_2]\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = b \Rightarrow \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = [A_1 \; A_2]^+ b$

⇕

$\min_{x_1, x_2} \left\| [A_1 \; A_2]\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - b \right\|_2$

or

$\min_{x_1, x_2} \left\| \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \right\|_2$ s.t. $[A_1 \; A_2]\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = b$

- $A_1 x - b_1 = 0$
  $A_2 x - b_2 = 0$

- Equal priority

$x = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}^+ \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$

- Hierarchy

$x = A_1^+ b_1 + N(A_1)x_0$

$A_1(A_1^+ b_1 + N(A_1)x_0)$

$= A_1 A_1^+ b_1 + 0 = b_1$

$\min_x \|A_2 x - b_2\|_2$

s.t. $\|A_1 x - b_1\| = c_1$

⇒ $A_2 x - b_2 = 0$

⇒ $A_2(A_1^+ b_1 + N(A_1)x_0) - b_2 = 0$

⇒ $x_0 = [A_2 N(A_1)]^+ [b_2 - A_2 A_1^+ b_1]$

try to fulfill $A_2 x = b_2$

△ case study

- $q = (\varphi_1 \; \varphi_2 \; x \; \varphi_3)^T$
- $M\ddot{q} + b + g = \tau$

- $J_P \in \mathbb{R}^{2 \times 4}$
  $J_R \in \mathbb{R}^{1 \times 4}$

(1) static gravity compensation joint torque?
i.e., $\cancel{M\ddot{q}} + \cancel{b} + g = \tau$

(2) lifting 100 kg, joint torque?

$M\ddot{q} + b + g + J_P^T \begin{pmatrix} 0 \\ 100 \end{pmatrix} = \tau$

⇒ $\tau = g + J_P^T \begin{pmatrix} 0 \\ 100 \end{pmatrix}$

(3) accelerate w/ 1 m/s² upward
φ, x = const, joint torque?

- $J_t = \begin{bmatrix} J_P \\ J_R \end{bmatrix}$ $\quad \dot{w}_e^* = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$ might subject to feedback law

my desired acceleration.

- we know that $\quad w = J\dot{q}$
  $\dot{w} = J\ddot{q} + \dot{J}\dot{q}$

⇒ $\ddot{q}^* = J^+(\dot{w} - \dot{J}\dot{q}) + N\ddot{q}_0$

⇒ $\tau = M\ddot{q}^* + b + g$

$= MJ^+(\dot{w} - \dot{J}\dot{q}) + b + g$

(1) task space:

1 m/s² in y, x, φ = const.
joint torque?
is the sol^n the same?

- $\dot{w}_t^* = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$ $\quad \Lambda = (J_e M^{-1} J_e^T)^{-1}$

$J_e = J_t = \begin{bmatrix} J_P \\ J_R \end{bmatrix}$ $\quad \mu = \Lambda J_e M^{-1} b - \Lambda \dot{J}_e \dot{q}$
$p = \Lambda J_e M^{-1} g$

⇒ $\tau^* = J_t^T(\underbrace{\Lambda \dot{w}_t^* + \mu + p}_{F_e})$

(1) interaction control

$\underbrace{\dot{w}_t^* = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}}_{\text{desired}}$ $\quad J_t = \begin{bmatrix} J_P \\ J_R \end{bmatrix}$

contact force

$\tau^* = J_e^T \underbrace{\left[\Lambda \dot{w}_t^* + \mu + p + \underbrace{\begin{bmatrix} -100 \\ -1000 \\ 0 \end{bmatrix}}_{F_c}\right]}_{F_e}$

(1) QP problems.

(a) least-square minimal-torque to get $\dot{w}^* = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$?

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

- $[M \; -I]x - (-b-g) = 0$ (1)
  $A_1 \; x - b_1 = 0$

$\begin{cases} w = J\dot{q} \\ \dot{w} = J\ddot{q} + \dot{J}\dot{q} \end{cases}$

- $\left[\begin{matrix} J_P \\ J_R \end{matrix} \; 0\right]x - \left(\dot{w}^* - \begin{bmatrix} \dot{J}_P \\ \dot{J}_R \end{bmatrix}\dot{q}\right) = 0$ (2)

$A_2 x - b_2 = 0$

- $[0 \; I]x = 0$ (minimize torque) $\min \|\tau\|$
  $A_3 x - b_3 = 0$ (3)

as good as possible

then solve.

(b) assume $\ddot{q}_i = 0$, and $\dot{w}^* = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

$[M \; -I]x - (-b-g) = 0$ (1)

$\begin{bmatrix} J_P & \\ J_R & 0 \\ [1, 0, 0, 0] & \end{bmatrix} x - \left(\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} \dot{J}_P \\ \dot{J}_R \\ [0, 0, 0, 0] \end{bmatrix}\dot{q} \right) = 0$ (2)

(c) assume $\tau_1 = 0$, $\dot{w}^* = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

$[M \; -I]x - (-b-g) = 0$ (1)

$[0 \; [1\,0\,0\,0]]\,x = 0$ (2)

$\left[\begin{matrix} J_P \\ J_R \end{matrix} \; 0\right]x - \left(\dot{w}^* - \begin{bmatrix} \dot{J}_P \\ \dot{J}_R \end{bmatrix}\dot{q}\right) = 0$ (3)

△ solving a set of QPs

- w/ different priorities
  methods:
  - exploit Nullspace of tasks w/ higher priority
  - every step = QP w/ constraints
  - iteratively nullspace projection

- pseudo-code

```
n_T = number of tasks
x = 0
N_1 = I
for i = 1 -> n_T do
    x_i = (A_i N_i)^+ (b_i - A_i x)
    x = x + N_i x_i
    N_{i+1} = N([A_1^T ... A_i^T]^T)
end for
```

eg. @ i = 1

$x = 0$

$N_1 = I$

$x_1 = (A_1 I)^+(b_1 - A \cdot 0)$

$= A_1^+ b_1$

$x := 0 + I A_1^+ b_1$

$N_2 = N(A_1)$

@ i = 2

$x = A_1^+ b_1$

$N_2 = N(A_1)$

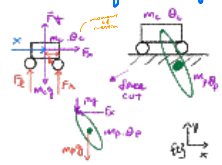
$x_2 = [A_2 N(A_1)]^+ [b_2 - A_2 A_1^+ b_1]$

$x := A_1^+ b_1 + N(A_1)[A_2 N(A_1)]^+ [b_2 - A_2 A_1^+ b_1]$

use → solver (most of the time)

△ Dynamic Modelling Case Study

(1) NE



△ 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)

$m_p\left(\ddot{\varphi} l\sin(\varphi) + \dot{\varphi}^2 l\cos(\varphi)\right) = -F_y - m_p g$ (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

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

(2) NE - projected

$q = \begin{bmatrix} x \\ \varphi \end{bmatrix}$

- $r_{OS_c} = \begin{bmatrix} x \\ 0 \end{bmatrix}$

$J_{Pc} = \frac{\partial r}{\partial q} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$

$\dot{J}_{Pc} = \frac{\partial J_{Pc}}{\partial t} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$

- $r_{OS_p} = \begin{bmatrix} x + l\sin\varphi \\ -l\cos\varphi \end{bmatrix}$

$J_{Pp} = \frac{\partial r}{\partial q} = \begin{bmatrix} 1 & l\cos(\varphi) \\ 0 & l\sin(\varphi) \end{bmatrix}$

$\dot{J}_{Pp} = \frac{\partial J_{Pp}}{dt} = \begin{bmatrix} 0 & -l\sin(\varphi)\dot{\varphi} \\ 0 & l\cos(\varphi)\dot{\varphi} \end{bmatrix}$

$w_p = \dot{\varphi}$

$J_{Rp} = \frac{\partial w_p}{\partial \dot{q}} = [0 \; 1]$

- E.O.M

$\mathbf{M} = \sum \mathbf{J}_{P_i}^T m_i \mathbf{J}_{P_i} + \mathbf{J}_{R_i}^T \Theta_i \mathbf{J}_{R_i} = \mathbf{J}_{P_c}^T m_c \mathbf{J}_{P_c} + \mathbf{J}_{P_p}^T m_p \mathbf{J}_{P_p} + \mathbf{J}_{R_p}^T \theta_p \mathbf{J}_{R_p} = \begin{bmatrix} m_c + m_p & l m_p \cos(\varphi) \\ l m_p \cos(\varphi) & m_p l^2 + \theta_p \end{bmatrix}$

$\mathbf{b} = \sum \mathbf{J}_{P_i}^T m_i \dot{\mathbf{J}}_{P_i} \dot{\mathbf{q}} + \mathbf{J}_{R_i}^T \Theta_i \dot{\mathbf{J}}_{R_i} \dot{\mathbf{q}} + \underbrace{(\mathbf{J}_{R_i} \dot{\mathbf{q}}) \times \Theta_i \mathbf{J}_{R_i} \dot{\mathbf{q}}}_{=0 \text{ (planar system)}} = \mathbf{J}_{P_p}^T m_p \dot{\mathbf{J}}_{P_p} \dot{\mathbf{q}} = \begin{pmatrix} -\dot{\varphi}^2 l m_p \sin(\varphi) \\ 0 \end{pmatrix}$

$\mathbf{g} = \sum_{i=1}^{n} -\mathbf{J}_{S_i}^T \mathbf{F}_i^g = -\mathbf{J}_{P_c}^T \begin{pmatrix} 0 \\ -m_c g \end{pmatrix} - \mathbf{J}_{P_p}^T \begin{pmatrix} 0 \\ -m_p g \end{pmatrix} = \begin{pmatrix} 0 \\ m_p g l \sin(\varphi) \end{pmatrix}$

(3) Lagrange II

- Kinematics cart and pendulum

$\mathbf{r}_{OS_c} = \begin{pmatrix} x \\ 0 \end{pmatrix}$ $\quad \mathbf{r}_{OS_p} = \begin{pmatrix} x + l\sin(\varphi) \\ -l\cos(\varphi) \end{pmatrix}$ $\quad \varphi_p = \varphi$

$\dot{\mathbf{r}}_{S_c} = \begin{pmatrix} \dot{x} \\ 0 \end{pmatrix}$ $\quad \dot{\mathbf{r}}_{S_p} = \begin{pmatrix} \dot{x} + l\cos(\varphi)\dot{\varphi} \\ l\sin(\varphi)\dot{\varphi} \end{pmatrix}$ $\quad \omega_p = \dot{\varphi}$

- Kinetic and potential energy

$\mathcal{T} = \sum \frac{1}{2}\dot{\mathbf{r}}_{S_i}^T m_i \dot{\mathbf{r}}_{S_i} + \frac{1}{2}\boldsymbol{\omega}_i^T \Theta_i \boldsymbol{\omega}_i = \frac{1}{2}m_c\dot{x}^2 + \frac{1}{2}m_p\dot{x}^2 + \frac{1}{2}m_p l^2\dot{\varphi}^2 + m_p\dot{x}l\cos(\varphi)\dot{\varphi} + \frac{1}{2}\theta\dot{\varphi}^2$

$\mathcal{U} = -m_p g l\cos(\varphi)$ ← 0-level can be chosen

- Equation of motion

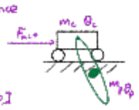
$\frac{\partial \mathcal{T}}{\partial \dot{\mathbf{q}}} = \begin{pmatrix} m_c\dot{x} + m_p\dot{x} + m_p l\cos(\varphi)\dot{\varphi} \\ m_p l^2\dot{\varphi} + m_p\dot{x}l\cos(\varphi) + \theta\dot{\varphi} \end{pmatrix}$

$\frac{d}{dt}\left(\frac{\partial \mathcal{T}}{\partial \dot{\mathbf{q}}}\right) - \frac{\partial \mathcal{T}}{\partial \mathbf{q}} + \frac{\partial \mathcal{U}}{\partial \mathbf{q}} = 0$ $\quad \frac{d}{dt}\left(\frac{\partial \mathcal{T}}{\partial \dot{\mathbf{q}}}\right) = \begin{pmatrix} m_c\ddot{x} + m_p\ddot{x} + m_p l\cos(\varphi)\ddot{\varphi} - m_p l\sin(\varphi)\dot{\varphi}^2 \\ m_p l^2\ddot{\varphi} + m_p\ddot{x}l\cos(\varphi) - m_p\dot{x}l\sin(\varphi)\dot{\varphi} + \theta\ddot{\varphi} \end{pmatrix}$

$-\frac{\partial \mathcal{T}}{\partial \mathbf{q}} = \begin{pmatrix} 0 \\ m_p\dot{x}l\sin(\varphi)\dot{\varphi} \end{pmatrix}$

$\frac{\partial \mathcal{U}}{\partial \mathbf{q}} = \begin{pmatrix} 0 \\ m_p g l\sin(\varphi) \end{pmatrix}$

(4) External force



- $J_e^T F_e$

where $J_e = [1, 0]$

∴ $\tau = J^T F_e$

$= \begin{bmatrix} 1 \\ 0 \end{bmatrix} F_{ext} = \begin{bmatrix} F_{ext} \\ 0 \end{bmatrix}$

△ Eigendecomposition