△ Inverse Kinematics

- Recall: forward kinematics

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

parametrize

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

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

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

△ sol^n

- Analytic :
  - for 3 intersecting neighbouring axes
- Geometric :
  - use length, theorem... "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 ⇒ $J_{e0}(q_s)$ 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

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

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

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

- Redundancy

$w = J\dot q$

- $q \in R^n$
- $w \in R^m$
- $J \in R^{m\times n}$ — $m > n$ → 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 & orientation, ensure kinematics
- break down tasks: $\{J_i, w_i^*\}$

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

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

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

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

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

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

- Prioritization

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

⇒ $w_2 = J_2 \dot q$

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

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

⇒ $\dot q = J_1^{+} w_1^* + N_1 (J_2 N_1)^{+} (w_2^* - J_2 J_1^{+} w_1^*)$ — $\dot q_0$ ↳ $J_2\dot q \cong w_2$

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

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

△ 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 ← 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-conditioned

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

- problem 2

Jacobian is rank deficient → bad condition

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

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

△ Orientation

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

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

⇒ $q \leftarrow q + k_{pr} J_{e0R}^{+} \Delta\varphi$ — geometric one, use rotation vector

△ Traj. ctrl.

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

△ L.A. revisit

△ determinants

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

$Ax = b$

- if $\det(A) = 0$:
  - A transform x to "lower dimension" cannot unpack, i.e., 无法 transform back
  - sometimes can! already "live" on the new "lower dim"
- if $\det(A) \neq 0$:
  - then can reverse back!

△ Rank

$Ax = b$ — DoF of the output

rank$(A) = n$

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

△ Nullspace (kernel)

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

↳ a set of vectors, which form a space, whose DoFs get "squeeze" to the origin;

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

null space $\in R^{n-m}$ if $n \geq m$; reduced rank

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

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

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

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

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

△ inverse kinematics case study

(1)

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

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

find $\dot q$ (generalized velocity)

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

△ $r_E = \begin{bmatrix} S(q_1) + S(q_1+q_2) + S(q_1+q_2+q_3) \\ 1 + C(q_1) + C(q_1+q_2) + C(q_1+q_2+q_3) \end{bmatrix}$

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

⇒ w/ $q = [\pi/6, \pi/3, \pi/3]^T$

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

⇒ $\dot q = J_E^{+} w$

$= J^T (JJ^T)^{-1} [1; 1]$

$= \begin{bmatrix} 0.069 \\ -0.56 \\ -0.535 \end{bmatrix}$

check! $\frac{1}{2}\begin{bmatrix} 0 & -\sqrt3 & -\sqrt3 \\ -4 & -3 & -1 \end{bmatrix}\begin{bmatrix} 0.069 \\ -0.56 \\ -0.535 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

(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 priority)

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

$R^{3\times 1}$ $[R^{4\times 3}]^{+} [R^{4\times 1}]$

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

check! $\begin{bmatrix} J_1 \\ J_2 \end{bmatrix}\dot q = \begin{bmatrix} 1.057 \\ 0.983 \\ -0.133 \\ -0.067 \end{bmatrix} \approx \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$

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

⇒ $\dot q = J_1^{+} w_1 + N_1 \dot q_0$ — $I - J_1^{+}J_1$

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

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

⇒ $\dot q = J_1^{+} w_1 + N_1 \ldots$

$= \begin{bmatrix} -0.164 \\ -0.085 \\ -1.070 \end{bmatrix}$

check! $\begin{bmatrix} J_1 \\ J_2 \end{bmatrix}\dot q = \begin{bmatrix} 1 \\ 1 \\ -0.164 \\ -0.085 \end{bmatrix}$

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

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

$= 0$

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

$A = \begin{bmatrix} 400 & -201 \\ -800 & 401 \end{bmatrix}$

$A^{-1} = \begin{bmatrix} -2.00 & -5.01 \\ \ldots & \ldots \end{bmatrix}$

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

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

⇒ cond$[A] = 3603$

△ more on matrix norm

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

matrix → linear transformation

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

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

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

$A[u_1 \cdots u_n] = \lambda_i [u_1 \cdots u_n]$

$AU = \lambda U$

⇒ $\|AU_i\|_p = |\lambda_i| \|U_i\|_p \leq \|A\|_p \|U_i\|_p$

⇒ $\|\lambda_i\| \leq \|A\|_p$

⇒ $\max|\lambda_i| \leq \|A\|_p$

⇒ $\max|\lambda_i|^k \leq \|A^k\|_p$

△ Dynamics

this chapter basically talks about both description of motion — source of motion

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

τ → $\ddot q$ — cause of motion → description

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

- 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 $F = ma$

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

$r \times F = r \times ma$

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

功: work 功 ⇒ force × distance

△ Derivation of virtual displacements of Single Rigid body

- from rigid body kinematics

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

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

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

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

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

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

- Apply principle of virtual work

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

$= \int_B \begin{pmatrix} \delta r \\ \delta\Phi \end{pmatrix}^T \begin{bmatrix} I_{3\times3} \\ [p]_\times \end{bmatrix}$ — δr

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

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

w/ $\int_B dm = m$

$\int_B p\, dm = 0$ $S = COG$ moment

$\int_B -[p]_\times^2 dm = \int_B [p]_\times [p]_\times^T dm = \Theta_S$ — moment of inertia

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

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

⇒ means that

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

note that

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

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

⇒ $\begin{cases} \dot P_S = F_{ext} & \text{force — newton} \\ \dot N_S = T_{ext} & \text{torque — euler} \end{cases}$

"change in linear/angular momentum"

△ some reminder

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

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

- remove the connections (cut all bodies free) — produce constraint forces
- based on this, w/ n_b rigid bodies
  - $6n_b$ equation
  - $5n_b$ constrained force

△ projected Newton-Euler (in generalized motion direction)

一般通过 → 以 q, generalized coordinates 来表达。

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

- recall

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

now, this is limited due to linkages!

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

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

⇒ $\begin{bmatrix} \dot P_{S_i} \\ \dot N_{S_i} \end{bmatrix} = \begin{bmatrix} m a_{S_i} \\ \Theta_{S_i}\dot\psi_i + \Omega_{S_i} \times \Theta_{S_i} \Omega_{S_i} \end{bmatrix}$

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

- virtual displacement in generalized coordinates

$\begin{pmatrix} \delta r_S \\ \delta\Phi \end{pmatrix} = \begin{bmatrix} J_P \\ J_R \end{bmatrix} \delta q$ — 也叫有 雅克比 multi-body 都能推

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

$= \delta q^T \sum_{i=1}^{n_b} \Bigg\{ \underbrace{\begin{pmatrix} J_{S_i} \\ J_{R_i} \end{pmatrix}^T \begin{pmatrix} m J_{S_i} \\ \Theta_S J_{R_i} \end{pmatrix}}_{M(q)} \ddot q + \underbrace{\begin{pmatrix} J_{S_i} \\ J_{R_i} \end{pmatrix}^T \begin{pmatrix} m \dot J_{S_i} \dot q \\ \Theta_S \dot J_{R_i} \dot q + J_{R_i}\dot q \times \Theta_S J_{R_i} \dot q \end{pmatrix}}_{b(q,\dot q)} - \underbrace{\begin{pmatrix} J_{S_i} \\ J_{R_i} \end{pmatrix}^T \begin{pmatrix} F_{ext,i} \\ T_{ext,i} \end{pmatrix}}_{g(q)} \Bigg\}$

$\forall \delta q$

搞了这么久 就是为了用 "q"!

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

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

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

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

△ Lagrange II

- Lagrangian $\mathcal{L} = T - U$ — kinetic, potential
- Lagrangian Equation

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

- ∵ $U = U(q)$

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

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

△ Kinetic energy

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

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

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

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

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

△ potential energy

- gravity

$F_{g_i} = m_i g_I e_z$ ⇒ $U_g = -\sum_{i=1}^{n_b} r_{S_i}^T F_{g_i}$

- spring

$F_S = k_j (\|r_1 - r_0\| - d_0) \frac{r_1 - r_0}{\|r_1 - r_0\|}$

⇒ $U_S = \frac12 k_j (d(q) - d_0)^2$

△ External Force

- given r_{ext}, F_j, T_k

$\tau_{F_{ext}} = \sum_{j=1}^{n_f} J_{P,j}^T F_j$ — 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 we hv EoM, how do we control?

How do we get this

Actuator control $\xrightarrow{VA}$ Actuator torque generation $\xrightarrow{\tau}$ Robot Dynamics

$q, \dot{q}$

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

△ position-based control
- don't care about dynamics
- high gain PID: good 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 functions 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 $M\ddot{q}, b \to 0$, $q^* - q = 0$)

△ 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: $\ddot{q}^* = \ddot{q}^* + k_p(q^*-q) + k_d(\dot{q}^*-\dot{q})$ - assume w/ dynamic modelling error

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

- describe from task space

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

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

& similarly, multi-task

- $\ddot{q} = \begin{bmatrix} J_1 \\ \vdots \\ J_{nt} \end{bmatrix}^+ \left(\begin{bmatrix} \dot{w}_1 \\ \vdots \\ \dot{w}_{nt} \end{bmatrix} - \begin{bmatrix} \dot{J}_1 \\ \vdots \\ \dot{J}_{nt} \end{bmatrix} \dot{q} \right)$

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

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

derivation

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

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

⇒ $\dot{w}_e - \dot{J}_e\dot{q} + J_e M^{-1} b + J_e M^{-1} g = J_e M^{-1} J_e^T F_e$

$\quad\quad \mu \quad\quad\quad p \quad\quad \Lambda^{-1}$

⇒ $\Lambda\dot{w}_e + \mu + p = F_e$

↳ geometric interpretation: inertial ellipsoid (depend on our configuration)

$\therefore$ get $\dot{w}_e^* = k_p E(\chi_e^* - \chi_e)$ + $k_d(w_e^* - w_e)$ — $\dot{w}_e(t)$ — feed-forward

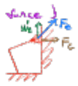
determine desired acceleration → task-level force

joint-space torques ← Jacobi transformation

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

△ interaction control

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



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

△ selection force

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

⇒ $\tau^* = J^T F_e$

$\begin{cases} \Sigma_p = \begin{bmatrix} \sigma_{px} & 0 & 0 \\ 0 & \sigma_{py} & 0 \\ 0 & 0 & \sigma_{pz} \end{bmatrix} & \sigma \in \{0,1\} \\ \Sigma_{pf} = I_3 - \Sigma_p \end{cases}$ 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-level frame & world frame

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

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

$\therefore$ 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

$w = J\dot{q}$

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

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

$= M(q)[J^+(\dot{w} - \dot{J}\dot{q})] + b(q,\dot{q}) + g$

2. operation space control

$\tau^* = J_e^T F_e$

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

Not only sol^n

$\therefore \tau^* = J_e^T(\Lambda\dot{w}_e^* + \mu + p) + N(J_e^T)\tau_0$ — null space — can change my torque

$N(J_e^T)$

$= (I - J_e^T[J_e M^{-1} J_e^T]^{-1} J_e M^{-1})$ — Mass-weighted pseudo-inverse — (change desired $\dot{w}$ w.o. $\dot{w}$) proof

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

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

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

$+ (I - J_e^T(J M^{-1} J_e^T)^{-1} J M^{-1})\tau_0$

$- b - g)$

$+ \dot{J}\dot{q}$ $= 0$

- we can also see:

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

⇒ $\ddot{q} = M^{-1}(\tau^* - b - g)$

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

some sort of mass-matrix weighted pseudo-inverse

3. QP

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

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

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

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

△ case study

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

- $J_P \in \mathbb{R}^{2\times4}$
  $J_R \in \mathbb{R}^{1\times4}$

(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?

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

my desired acceleration.

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

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

⇒ $\tau = M\ddot{q}^* + b + g$

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

(1) task space:

1 m/s² in y, x, φ = const.

joint torque?

is the sol^n the same?

- $\dot{w}_t^* = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$ $\Lambda = (J_e M^{-1} J_e^T)^{-1}$
  $J_e = J_t = \begin{bmatrix} J_P \\ J_R \end{bmatrix}$ $\mu = \Lambda J_e M^{-1} b - \Lambda \dot{J}_e\dot{q}$
  $p = \Lambda J_e M^{-1} g$

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

(1) interaction control

$\dot{w}_t^* = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}$ $J_t = \begin{bmatrix} J_P \\ J_R \end{bmatrix}$

desired

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

F_e

(1) QP problems.

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

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

- $[M \;\; -I]x - (-b-g) = 0$ (1)
  $A_1 \; x - b_1 = 0$ $\begin{cases} w = J\dot{q} \\ \dot{w} = J\ddot{q} + \dot{J}\dot{q} \end{cases}$

- $\begin{bmatrix} J_P & 0 \\ J_R & \end{bmatrix} x - \left(\dot{w}^* - \begin{bmatrix} \dot{J}_P \\ \dot{J}_R \end{bmatrix}\dot{q}\right) = 0$ (2)
  $A_2 x - b_2 = 0$

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

as good as possible — then solve.

(b) assume $\ddot{q}_1 = 0$, and $\dot{w}^* = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$

$[M \;\; -I]x - (-b-g) = 0$ (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 $\tau_1 = 0$, $\dot{w}^* = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$

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

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

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

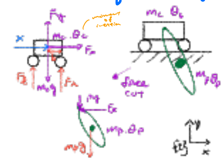
- pseudo-code
  n_T = number of tasks
  $x = 0$
  $N_1 = I$
  for $i = 1 \to n_T$ do
  $\quad x_i = (A_i N_i)^+(b_i - A_i x)$
  $\quad x = x + N_i x_i$
  $\quad N_{i+1} = N([A_1^T \cdots A_i^T]^T)$
  end for

e.g. @ $i = 1$
$x = 0$
$N_1 = I$
$x_1 = (A_1 I)^+(b_1 - A\,0)$
$\quad = A_1^+ b_1$
$x := 0 + I A_1^+ b_1$
$N_2 = N(A_1)$
@ $i = 2$
$x = A_1^+ b_1$
$N_2 = N(A_1)$
$x_2 = [A_2 N(A_1)]^+[b_2 - A_2 A_1^+ b_1]$
x is $A_1^+ b_1 + N(A_1)[A_2 N(A_1)]^+[b_2 - A_2 A_1^+ b_1]$

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

(2) NE-projected

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

- $r_{OSc} = \begin{bmatrix} x \\ 0 \end{bmatrix}$
  $J_{Pc} = \frac{\partial r}{\partial q} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$
  $\dot{J}_{Pc} = \frac{\partial J_{Pc}}{\partial t} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$

- $r_{OSp} = \begin{bmatrix} x + l\sin\varphi \\ -l\cos\varphi \end{bmatrix}$
  $J_{Pp} = \frac{\partial r}{\partial q} = \begin{bmatrix} 1 & l\cos(\varphi) \\ 0 & l\sin(\varphi) \end{bmatrix}$
  $\dot{J}_{Pp} = \frac{\partial J_{Pp}}{\partial t} = \begin{bmatrix} 0 & -l\sin(\varphi)\dot{\varphi} \\ 0 & l\cos(\varphi)\dot{\varphi} \end{bmatrix}$
  $w_p = \dot{\varphi}$
  $J_{Rp} = \frac{\partial w_p}{\partial \dot{q}} = [0 \;\; 1]$

- E.o.M

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

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

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

(3) Lagrange II

- Kinematics cart and pendulum

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

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

- 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

$\frac{d}{dt}\left(\frac{\partial\mathcal{T}}{\partial\dot{\mathbf{q}}}\right) - \frac{\partial\mathcal{T}}{\partial\mathbf{q}} + \frac{\partial\mathcal{U}}{\partial\mathbf{q}} = 0$

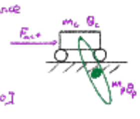
$\frac{\partial\mathcal{T}}{\partial\dot{\mathbf{q}}} = \begin{pmatrix} m_c\dot{x} + m_p\dot{x} + m_p l\cos(\varphi)\dot{\varphi} \\ m_p l^2\dot{\varphi} + m_p\dot{x}l\cos(\varphi) + \theta\dot{\varphi} \end{pmatrix}$

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

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

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

(4) External force



- $J_e^T F_e$
  where $J_e = [1, 0]$

$\therefore \tau = J^T F_e$

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

△ Eigendecomposition