Weighted Pseudo-Inverse.

- want to find $\dot{\vec{\theta}}$ such that $\xi^{b} = J_{body}(\theta)\,\dot{\theta}$ and $\dot{\vec{\theta}}^{T} W \dot{\vec{\theta}}$ is minimized, where W is a positive definite symmetric matrix.

solution to this problem is:

$$\dot{\vec{\theta}} = W^{-1} J^{T} (J W^{-1} J^{T})^{-1} \xi^{b}$$

in redundant case.

- now we can give different joint angle rates different priorities.
- done by minimizing $\mathcal{L}(\dot{\vec{\theta}}, \lambda) = \frac{1}{2} \dot{\vec{\theta}}^{T} W \dot{\vec{\theta}} + \lambda^{T} (\xi^{b} - J_{body}(\vec{\theta}) \dot{\vec{\theta}})$

Lagrange multiplier problem.

$J^{\#} = W^{-1} J (J W^{-1} J^{T})^{-1}$ is the weighted pseudo-inverse.

Damped Pseudo-Inverse

- want to avoid inverse blow-up near singularities.

consider $A = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \longrightarrow A^{\#} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$

but $B = \begin{bmatrix} 1 & 0 \\ 0 & \epsilon \end{bmatrix} \longrightarrow B^{\#} = \begin{bmatrix} 1 & 0 \\ 0 & 1/\epsilon \end{bmatrix}$

crazy stuff happens when ϵ goes from negative to positive.

singularities really are no good.

find $\dot{\theta}$ that minimizes the weighted sum

$$\mathcal{L}(\dot{\vec{\theta}}) = \frac{1}{2} \| \xi^b - J_{body}(\vec{\theta})\dot{\vec{\theta}} \|^2 + \frac{1}{2}\rho^2 \|\dot{\vec{\theta}}\|^2$$

the solution is:

$\cancel{\dot{\theta} = J^{\#}_{\rho}(\vec{\theta})\xi^b}$, where $J_{\rho}^{\#} = J^T$

$$\dot{\vec{\theta}} = J^{\#}_{body}(\vec{\theta};\rho)\,\xi^b, \quad \text{where } J^{\#}_{body}(\cdot;\rho) = J^T_{body}(J_{body}J^T_{body} + \rho^2 I)^{-1}$$

eigenvalues of system change to $\dfrac{\sigma_i}{\sigma_i^2 + \rho^2}$ ← no longer blow up at $\sigma_i = 0$.

- but, also bounds the joint velocities; $\|\dot{\vec{\theta}}\| / \|\xi\| < \frac{1}{2\rho}$. trajectory may deviate.

Recall: technique to connect two configurations by following a given trajectory was to use waypoints along trajectory & solve for the inverse kinematics:

$$\vec{\Theta}_i = \text{inv. kin. of } (g_e(t_i))$$

Problems

- multiple solutions to inverse can cause problems.
- redundant arms do not have unique inverses. ← this can really cause ambiguity.

Let's consider a different approach, called RESOLVED RATE TRAJECTORY GENERATION.

$$\dot{g}(t) = J(\vec{\theta}(t))\,\dot{\vec{\theta}}(t) \quad \text{OR} \quad \xi^b(t) = J_{body}(\vec{\theta}(t))\,\dot{\vec{\theta}}(t)$$

$$\xi^s(t) = J_{spatial}(\vec{\theta}(t))\,\dot{\vec{\theta}}(t)$$

$$\xi^h(t) = J_{hybrid}(\vec{\theta}(t))\,\dot{\vec{\theta}}(t)$$

$$\dot{\vec{\theta}}(t) = J^{\#}_{body}(\vec{\theta}(t))\,\xi^b(t)$$

↑ resolves ambiguity by picking solution w/smallest norm ⇒ minimizes joint velocities.

integrate:

$$\vec{\theta}(t) = \vec{\theta}(t_0) + \int_{t_0}^{t_f} J^{\#}_{body}(\vec{\theta}(\tau))\,\xi^b(\tau)\,d\tau$$

but, how do we really compute integral?

if done numerically on computer, need to discretize the problem.

$$\vec{\dot\theta}(t_j) = J^{\#}_{body}(\vec{\theta}(t_j))\, \xi^b(t_j)$$

$t_0, t_1, \dots, t_n, t_f$

$t_{i+1} - t_i = \Delta t$

should be pretty small.

$\Rightarrow$

$$\frac{\vec{\theta}(t_j) - \vec{\theta}(t_{j-1})}{\Delta t} = J^{\#}_{body}(\vec{\theta}(t_j))\, \xi^b(t_j)$$

$\Rightarrow$

$$\vec{\theta}(t_j) = \vec{\theta}(t_{j-1}) + \Delta t\, J^{\#}_{body}(\vec{\theta}(t_j))\, \xi^b(t_j)$$

- Assumes we can get $\xi^b(t_j)$ from cts. trajectory $g(t)$.
- This is a first-order accurate integration technique; can use other more accurate techniques.

If $\xi^b(t_j)$ can't be found & $g(\cdot)$ is vectorizable, then we can use the standard Jacobian:

$$\dot\theta(t_j) = J^{\#}(\theta(t_j))\, \dot g(t_j)$$

$\Rightarrow$ first order approximation: $\dot g(t_j) = \dfrac{g(t_j) - g(t_{j-1})}{\Delta t}$

$$\theta(t_j) = \theta(t_{j-1}) + J^{\#}(\theta(t_j))\, \underbrace{(g(t_j) - g(t_{j-1}))}_{\text{must be careful here since } g \text{ treated like a vector in } \mathbb{E}}$$

Or, if can't do that, but have a collection of sufficiently close points, then use logarithm,

$$\xi_j = \log_{\Delta t}\left(g^{-1}(t_{j-1})\, g(t_j) \right) \quad \text{so that} \quad \exp(\xi_j \Delta t) = g^{-1}(t_{j-1})\, g(t_j)$$

requires scaling to get timing right.

and integrate

$$\vec{\theta}(t_j) = \vec{\theta}(t_{j-1}) + \Delta t \, J^{\#}_{body}(\vec{\theta}(t_j)) \cdot \xi_j$$

- Ultimately, idea is to use Jacobian to relate velocities and integrate up to final configuration on trajectory.

- what if trajectory crosses, or passes nearby, a singularity?

  at singularity, Jacobian loses rank.

  what happens to $J^{\#}$?

  - main part of $J^{\#}$ is $(J J^T)^{-1}$

    $$\det(A) = \prod_i^n \lambda_i$$

    if loses rank, $\lambda_i \to 0$ for some i

    $\Rightarrow$ matrix blows up (inverse that is)

    $\Rightarrow$ get ∞ velocities.

- when minimizing joint velocities, ~~tries to~~ gives all joint velocities the same weighting. May not be desirable.

Redundant Manipulators & Internal Motions

redundant manipulator $\Rightarrow$ ∞ many configurations to inv. kin.

we can design joint-trajectories that satisfy,

$$g_e(\theta(t)) = g^* \qquad (*)$$

for g^* constant.

The set of all θ that satisfy $(*)$ form the self-motion manifold for the configuration g^*.

$\Rightarrow$ differentiate

$$J_{body}(\theta)\,\dot{\theta} = 0$$

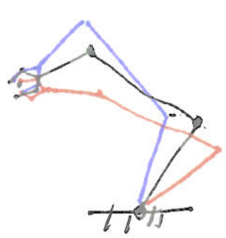
Joint-Trajectories in self-motion manifold have joint velocities which lie in the null-space of the manipulator Jacobian.

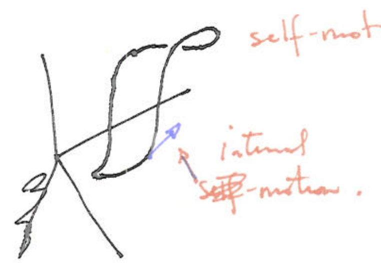
Motion along the self-motion manifold is called an internal motion.

→ pseudo-inverse picks $\dot{\theta}$ to have no internal motions.

↳ there are other possible inverses

↑ may even be preferred / desired.





Null Space Projection

$$\xi^b = J_{body}(\vec{\theta})\,\dot\theta$$

$\Rightarrow$

$$\dot\theta = J^{\#}_{body}(\theta)\,\xi^b$$

where $\dot\theta = \dot\theta_{row}$ not null-space component.

↳ there are other possible solutions that will move the joints but not affect the velocity of the end-effector.

$\Rightarrow$ What if we consider using them for additional control tasks?

* review null space and internal motions.

In general, for a vector z

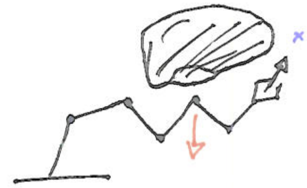
$$(id - J^{\#}J)\,\vec{z} \in \text{null}(J)$$

proof: $J(id - J^{\#}J)\vec{z} = (J - JJ^{\#}J)\vec{z} = (J - J)\vec{z} = 0$

Any solution $\dot\theta^*$ to $\xi^b = J_{body}(\theta)\,\dot\theta$ can be written in the form:

$$\dot\theta = J^{\#}_{body}(\theta)\,\xi^b + (id - J^{\#}_{body}(\theta)\,J_{body}(\theta))\,\cancel{z}\,\underset{\substack{\uparrow \\ \text{arbitrary}}}{\zeta}$$

Null-Space Projection algorithms are based on determining $\cancel{z}\,\zeta$ to use the redundancy to satisfy additional criteria besides ~~to~~ end-effector trajectory tracking.



Example : using redundancy to avoid obstacles

1) assume that we can determine the closest point on the robot to the obstacle.

2) to avoid the obstacle, cause the closest point to move away from the obstacle by specifying the velocity ξ_θ^b (body) of the closest point.

3) $\xi^b = J_{body}(\theta)\,\dot\theta$

$\xi_\theta^b = J_\theta(\theta)\,\dot\theta$ ← body Jacobian of closest point on robot

4) $\dot\theta = J^\#(\theta)\,\xi^b + \left(id - J^\#_{body}(\theta)\,J_{body}\right)\alpha$

choose α by solving,

$$\xi_\theta^b = J_\theta(\theta)\,J^\#(\theta)\,\xi^b \; \cancel{+ J_\theta\left(id - J^\#_{body}(\theta)\,J_{body}(\theta)\right)\xi}$$
$$+ J_\theta\left(id - J^\#_{body}(\theta)\,J_{body}(\theta)\right)\alpha$$

$\Rightarrow$

$$\alpha = \left[J_\theta\left(id - J^\#_{body}(\theta)\,J_{body}(\theta)\right)\right]^\# \cdot \left[\xi_\theta^b - J_\theta(\theta)\,J^\#(\theta)\,\xi^b\right]$$

$\Rightarrow$

$$\dot\theta = J^\#(\theta)\,\xi^b + \left(id - J^\#_{body}(\theta)\,J_{body}\right)\cdot\left[J_\theta\left(id - J^\#_{body}\cdot J_{body}(\theta)\right]^\# \cdot \left[\xi_\theta^b - J_\theta(\theta)\,J^\#(\theta)\,\xi^b\right]$$

$\Rightarrow B[CB]^\dagger = [CB]^\dagger$

$$= J^\#(\theta)\,\xi^b + \left[J_\theta\left(id - J^\#_{body}(\theta)\,J_{body}(\theta)\right)\right]^\# \cdot \left[\xi_\theta^b - J_\theta(\theta)\,J^\#(\theta)\,\xi^b\right]$$

↑ move tool frame along desired path w/ no internal motion

↑ joint motion corresponding to null action we want

↑ velocity of closest point P_0, due to motion of end-effector.

More generally, we have the following theorem.

Theorem. Given any generalized inverse, $J^{inv}(\theta)$, of $J(\theta)$, $\exists\, \alpha$ such that

$$\dot{\theta} = J^{\#}_{body}(\theta)\,\xi^{b} + \left(id - J^{\#}_{body}(\theta)\, J_{body}(\theta)\right)\alpha$$

and

$$\dot{\theta} = J^{inv}(\theta)\,\xi^{b}$$

are the same.

Up until now, we have considered that robot can follow a "direct" route from initial configuration to the goal configuration. What if such a trajectory is not possible due to obstacles and/or workspace ~~[illegible]~~ constraints?

- need to come up w/a alternative trajectory that gets from initial to goal configuration ~~and as~~ that is feasible.

known as MOTION PLANNING (or subset thereof).

There are lots of motion planning / trajectory generation algorithms. Will focus on ~~one of~~ method.

ONLINE!

POTENTIAL FIELD METHODS : POSITION ONLY

idea: consider the robot as a particle that is attracted to the goal by an attracting potential and pushed away from obstacles by a repulsive potential. global minima at goal.

The total potential acting on the robot at configuration q is

$$U(q) = U_{attractive}(q) + U_{repulsive}(q)$$

The robot follows the negated gradient of the potential

$$-\nabla U(q) = -\nabla U_{attractive}(q) - \nabla U_{repulsive}(q)$$

Implementation – for a point robot

a) attractive goal : $U_{attractive}(q) = \frac{1}{2}\mu\, d^2(q, q_f)$

$d(q, q_f)$ – distance from q_i to q_f.

· attains minimum at $q = q_f$.

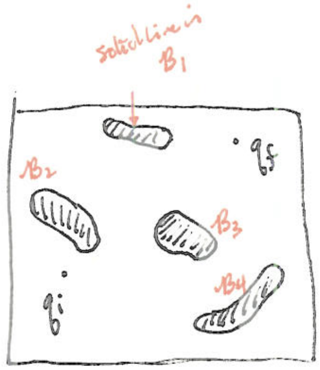
b) repulsive goal :   many choices, here's one

$$U_{repulsive} = \begin{cases} \frac{1}{2}\nu\left(\frac{1}{d(q,q_f)} - \frac{1}{\rho_o^i}\right) & \text{if } d(q, B_i) \le \rho_o^i \\ 0 & \text{otherwise} \end{cases}$$

boundary of obstacle i.

threshold distance

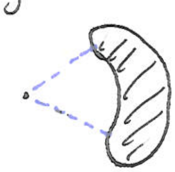
· if far from obstacle, repulsive potential not in effect



$$d(q, B_i) = \min_{(b \in B_i)} d(q, b)$$

usually, $d(q,b) = \|q - b\|^2$ (position only)

when B_i is convex, the closest point is unique, otherwise it may not be the case.



two closest points.

· gradient of $d_i(q)$

when closest point is unique: $\nabla d_i(q) = \frac{q-b}{\|q-b\|}$ (unit length vector)

· $\nabla d_i(q)$ is the direction that maximally increases distance to B_i.

when closest point is not unique; need generalized gradient

$$\partial d_i(q) = co(\vec{v}_1, \dots, \vec{v}_p)$$

where $v_j = \frac{q-b_j}{\|q-b_j\|}$

b_j = one of the closest points.

{co – co-convex hull}

· usually obtained by breaking up non-convex object into bunch of convex objects.

The Multi-Object Distance Function

What if there is more than one object close enough to be repelling?

a) here is one solution

let $D(q) = \min_{j \in J} d_j(q) = \min\{d_{j_1}(q), d_{j_2}(q), \dots, d_{j_k}(q)\}$

where J – index set of all active obstacles.

then,

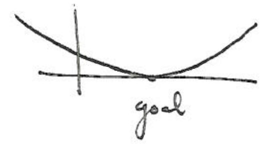
$$U_{repulsive}(q) = \begin{cases} \frac{1}{2}\left(\frac{1}{D(q)} - \frac{1}{\rho}\right)^2 & D(q) \le \rho \\ 0 & \text{otherwise.} \end{cases}$$

⇒ gradient:

$$\nabla D(q) = \begin{cases} \nabla d_j(q) & \text{if } B_j \text{ is unique closest obstacle} \\ co(\nabla d_{j_1}, \dots, \nabla d_{j_k}(q)) & \text{otherwise} \end{cases}$$

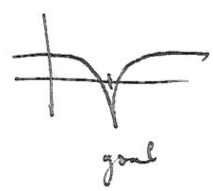
Problem w/ numerics

$U_{attractive}$



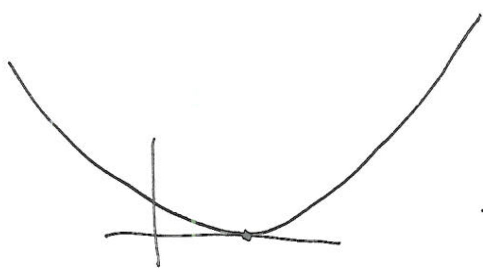
gradient may not be strong enough.

can add to attraction a local component



$$U_{attractive}(q) = \frac{1}{2}\mu_1 \|q - q_f\|^2 - \frac{1}{2}\mu_2 \frac{1}{\|q-q_f\|^2 + \epsilon^2}$$

some small number.



far away, gets to be too high.

can use

$$U_{attractive} = \frac{1}{2}\mu \|q - q_f\|$$

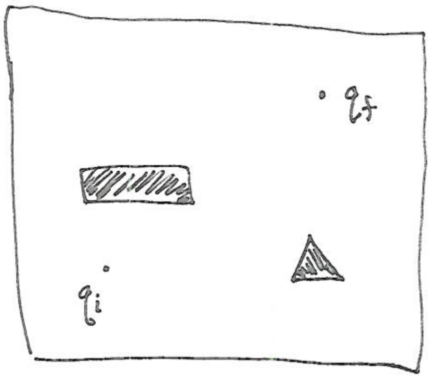
$\Rightarrow$

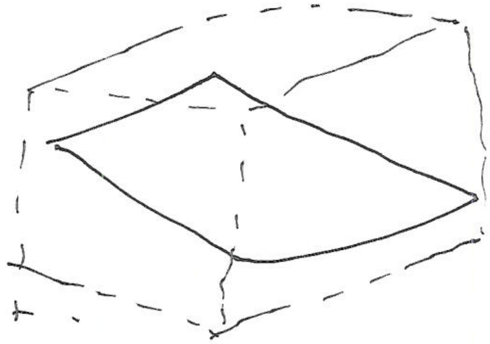
$$\nabla U_{attractive} = \cancel{\frac{1}{2}}\mu \frac{q - q_f}{\|q - q_f\|}$$

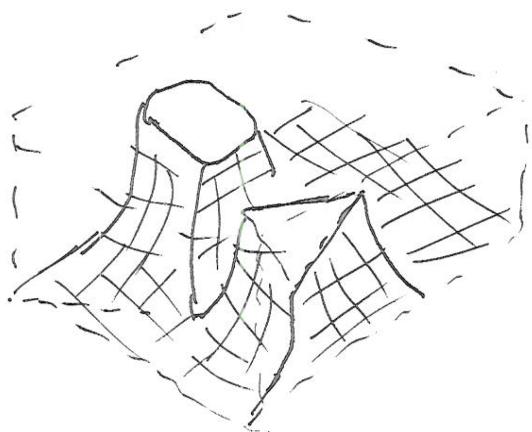
unit length.

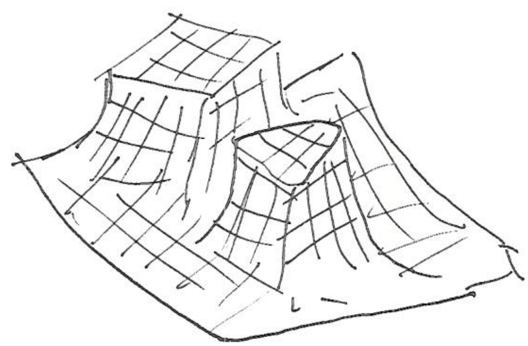
- nice because normalized.
- not nice because of numerics near minima.

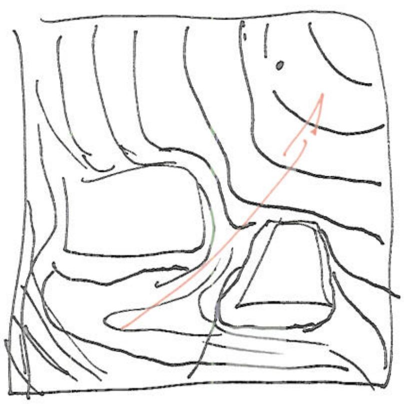
DRAWING











Pros: quick & easy

Cons: incomplete

local minima

~~fis~~ fields for circles functions

~~for circle~~