ECE 4560

continued from last time

$$p^0(0) = a_0^0 \qquad = p_0$$
$$p^0(T) = a_0^0 + a_1^0 T + a_2^0 T^2 + a_3^0 T^3 \qquad = p_1$$
$$\dot{p}^0(0) = a_1^0 \qquad = 0$$
$$\dot{p}^0(T) - \dot{p}^1(0) = a_1^0 + 2a_2^0 T + + 3a_3^0 T^2 - a_1^1 \qquad = 0$$
$$\ddot{p}^0(T) - \ddot{p}^1(0) = 2a_2^0 + 6a_3^0 T \underbrace{- 2a_2^1}_{\text{coeffecients from neighboring polynomials}} = 0$$

$\vdots$

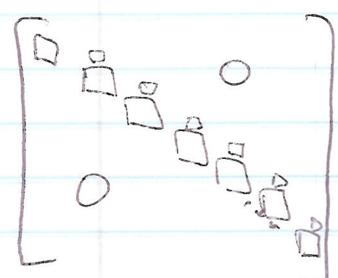
$$p^k(0) = a_0^k \qquad = p_k$$
$$p^k(T) = a_0^k + a_1^k T + a_2^k T^2 + a_3^k T^3 \qquad = p_{k+1}$$
$$\dot{p}^k(T) - \dot{p}^{k+1}(0) = a_1^k + 2a_2^k + 3a_3^k T^2 - a_1^{k+1} \qquad = 0$$
$$\ddot{p}^k(T) - \ddot{p}^{k+1}(0) = 2a_2^k + 6a_3^k T - 2a_2^{k+1} \qquad = 0$$

$\vdots$

$$p^n(0) = a_0^n \qquad = p_n$$
$$p^n(T) = a_0^n + a_1^n T + a_2^n T^2 + a_3^n T^3 \qquad = p_{n+1}$$
$$\dot{p}^n(T) = a_1^n + 2a_2^n T + 3a_3^n T^2 \qquad = 0$$

Leads to

$A(T)\vec{a} = p_0$ where $\vec{a} = \begin{pmatrix} a_0^0 \\ a_1^0 \\ a_2^0 \\ a_3^0 \\ \vdots \\ a_0^n \\ a_1^n \\ a_2^n \\ a_3^n \end{pmatrix}$



* if done correctly, "spline" function may do this.

# Resolved Rate Trajectory Generation:

- Just finished covering techniques to connect initial and desired end-effector configuration with a trajectory using splines & inverse kinematics.

Problems: 1) multiple solns to inverse kinematics can cause problems
2) redundant arms do have unique/finite inverses

So, to avoid these problems, let's consider an alternative technique called resolved rate trajectory generation.

Given a desired end-effector trajectory $g_e^*(t)$ and its associated velocity

$(\xi_e^*)^b(t)$, we consider

$$(\xi_e^*)^b(t) = J^b(\alpha(t))\dot{\alpha}(t)$$

in order to solve for $\alpha(t)$.

$\Rightarrow$ leads to differential equation:

$$\dot{\alpha}(t) = (J^b(\alpha(t)))^{\#}(\xi_e^*)^b$$

with initial condition:

$\alpha(0) = \text{inv. kin. } g_e^*(0) = \alpha_0$ → usually we know this upfront.

* at this point, we just need to integrate to get the solution.

Ideally, let a computational program deal with this, but if the option is not available, here is one option...

$$\dot{\alpha}(t) = \left(J^b(\alpha(t))\right)^{\#} \left(\xi_e^*\right)^b$$ } — discretize time

$\downarrow$

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

where $t_{i+1} - t_i = \Delta t$ ← should be small.

discretize derivative

$$\dot{\alpha}(t_k) = \frac{\alpha(t_{k+1}) - \alpha(t_k)}{\Delta t}$$

$$\Rightarrow \frac{\alpha(t_{k+1}) - \alpha(t_k)}{\Delta t} = \cancel{\left(J^b(\alpha(t))\right)^{\#}\left(\xi_e^*\right)^b}$$

$$= \left(J^b(\alpha(t_k))\right)^{\#} \left(\xi_e^*(t_k)\right)^b$$

$$\Rightarrow \alpha(t_{k+1}) = \alpha(t_k) + \Delta t \left(J^b(\alpha(t_k))\right)^{\#} \left(\xi_e^*(t_k)\right)^b$$

$\rightarrow$ repeat for all $t_k$, $k=0, \dots, n+1$

- This is a first-order accurate integration technique; there are other more accurate methods.
- Assumes we can get $(\xi_e^*(t))^b$ from a continuous trajectory $g_e^*(t)$

So, suppose that $g_e^*(t)$ is really just a collection of closely spaced way-points for which $(\xi_e^*(t))^b$ can't be found, e.g. we just have $g_e^*(t_k)$.

NAIVE APPROACH: If $g_e^*(t_k)$ is vectorizeable, then use standard Jacobian and first order approximation to end-effector derivative:

$$\alpha(t_{k+1}) = \alpha(t_k) + \Delta t\, J^{\#}(\alpha(t_k)) \frac{\left(g_e^*(t_{k+1}) - g_e^*(t_k)\right)}{\Delta t}$$

$$\Rightarrow \alpha(t_{k+1}) = \alpha(t_k) + J^{\#}(\alpha(t_k)) \left(g_e^*(t_{k+1}) - g_e^*(t_k)\right)$$

Geometric Approach: use logarithm,

$$\left(\xi_e^*(t_k)\right)^b = \ln_{\Delta t}\left(g^{-1}(t_k)\, g(t_{k+1})\right)$$

$$\parallel$$

$$\xi_k$$

need the time part to be $\Delta t$, eg., set $\tau = \Delta t$

$\Rightarrow$ leads to

$$\alpha(t_{k+1}) = \alpha(t_k) + \Delta t\, J^b(\alpha(t_k))\, \xi_k$$

ECE 4560

. ~~[illegible]~~ continued from last time...

. ultimately the idea is to use the Jacobian to relate velocities, then integrate up to final configuration of the trajectory (MATLAB & others can do this)

Problem 1: If true trajectory starts to deviate from desired, then end-effector may diverge from desired trajectory (open-loop problem)

Problem 2: What if the trajectory crosses, or passes nearby, a singularity? At singularity, Jacobian loses rank.

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

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

$$\det(A) = \prod_{i}^{n} \lambda_i$$

losing rank $\Rightarrow \lambda_i \rightarrow 0$ for some $i$

$\Rightarrow$ inverse matrix blows up!

$\Rightarrow$ get $\infty$ velocities.

. Observation: when minimizing joint velocities, the pseudo-inverse gives all joint velocities the same weighting. May not be desireable.

<u>Damped Pseudo-Inverse</u>

. GOAL: avoid blow-up near singularities.

the pseudo-inverse solved for the $\dot{\alpha}$ that minimized

$$\mathcal{L}(\dot{\alpha}) = \|\dot{\alpha}\| \quad \text{subject to} \quad \dot{g} = J(\alpha)\dot{\alpha}$$

(for redundant/sufficient)

leads to blow-up since $\xi^b = J^b(\alpha)\dot{\alpha}$ must hold.

Propose instead, to minimize

$$\mathcal{L}(\dot{\alpha}) = \frac{1}{2}\|\xi^b - J^b(\alpha)\dot{\alpha}\|^2 + \frac{1}{2}\rho^2\|\dot{\alpha}\|^2$$

The solution is called the damped pseudo-inverse and is

$$\dot{\alpha} = (J^b(\alpha))^{\#}\xi^b \quad \text{with}$$

$$(J^b(\alpha;\rho))^{\#} = (J^b)^T\left(J^b(J^b)^T + \rho^2 \mathbb{1}\right)^{-1}$$

eigenvalues of $(JJ^T)^{-1}$ change from $\frac{1}{\sigma_i}$ to $\frac{\sigma_i}{\sigma_i^2+\rho^2}$

the issue will be that now the joint velocities are bounded.

in fact, $\frac{\|\dot{\alpha}\|}{\|\xi\|} < \frac{1}{2\rho}$

trajectory may deviate.

## Weighted Psuedo-Inverse

- This considers an alternative formulation of the pseudo-inverse minimization problem.
  It is the minimization of

$\|a\|^2 = a \cdot a = a^T a$

$$\mathcal{L}(\dot{\alpha}) = \frac{1}{2}\|\xi - J^b(\alpha)\dot{\alpha}\|^2 + \frac{1}{2}\|\dot{\alpha}\|_W^2$$

where $\|v\|_W = v^T W v$
and $W$ is positive definite & symmetric.

Solution to this problem is:

$$\dot{\alpha} = W^{-1}(J^b)^T\left(J^b W (J^b)^T\right)^{-1}\xi^b$$

in redundant / sufficient case.

- now we can give different joint angle rates different ~~pe~~ priorities.

So, $J^{\#} = W^{-1}(J^b)^T\left(J^b W^{-1}(J^b)^T\right)^{-1}$ is the weighted psuedo-inverse.

Side note: Damped - Weighted Psuedo Inverse:

$$J^{\#} = W^{-1}(J^b)^T\left(J^b W^{-1}(J^b)^T + \rho^2 \mathbb{1}\right)^{-1}$$

---

HW info:

ode45 ( ______ , tspan, I.C.)

$$\dot{\alpha}(t) = \left(J^b(\alpha(t))\right)^{\#}\xi^b$$

function alphadot = diffeq(alpha, t)

$$\dot{\alpha} = \left(\underline{J^b(\alpha(t))}\right)^{\#}\xi^b(t)$$

end

---