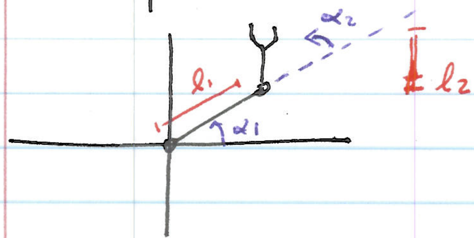
ECE 4560 11/05/07

Example:



$$g_1(\alpha_1) = \left[\begin{array}{c|c} R(\alpha_1) & 0 \\ \hline 0 & 1 \end{array}\right]$$

$$g_2(\alpha_2) = \left[\begin{array}{c|c} R(\alpha_2) & d_2 \\ \hline 0 & 1 \end{array}\right] \qquad d_2 = \begin{Bmatrix} l_1 \\ 0 \end{Bmatrix}$$

$$g_3(\alpha_3) = \left[\begin{array}{c|c} \mathbb{1} & d_3 \\ \hline 0 & 1 \end{array}\right] \qquad d_3 = \begin{Bmatrix} l_2 \\ 0 \end{Bmatrix}$$

$$\Rightarrow g_e(\alpha) = g_1(\alpha_1)\, g_2(\alpha_2)\, g_3$$
$$= \left[\begin{array}{c|c} R(\alpha_1+\alpha_2) & \begin{Bmatrix} l_1\cos(\alpha_1) + l_2\cos(\alpha_1+\alpha_2) \\ l_1\sin(\alpha_1) + l_2\sin(\alpha_1+\alpha_2) \end{Bmatrix} \\ \hline 0 & 1 \end{array}\right]$$

This is a kinematically insufficient arm $(m < n)$ $(2 < 3)$ $\Rightarrow$ can't expect total control.

Perhaps, should consider positioning only then. For analysis, let's reduce forward kinematics to translation/position only.

$$p_e(\vec{\alpha}) = \begin{Bmatrix} l_1\cos\alpha_1 + l_2\cos(\alpha_1+\alpha_2) \\ l_1\sin\alpha_1 + l_2\sin(\alpha_1+\alpha_2) \end{Bmatrix}$$

Since we deal with $p_e(\alpha)$ only and not $g_e(\alpha)$, there is no orientation to worry about, therefore standard Jacobian will be enough.

$$\Rightarrow J_{pos}(\alpha) = \frac{\partial p_e(\alpha)}{\partial \alpha} = \left[\begin{array}{c|c} -l_1\sin(\alpha_1) - l_2\sin(\alpha_1+\alpha_2) & -l_2\sin(\alpha_1+\alpha_2) \\ l_1\cos(\alpha_1) + l_2\cos(\alpha_1+\alpha_2) & l_2\cos(\alpha_1+\alpha_2) \end{array}\right]$$

Consider the following two cases

1) $\alpha = (\alpha_1, 0)$; $J(\alpha) = \begin{bmatrix} -(l_1+l_2)\sin\alpha_1 & -l_2 \cos\alpha_1 \\ (l_1+l_2)\cos(\alpha_1) & -l_2 \sin\alpha_1 \end{bmatrix}$

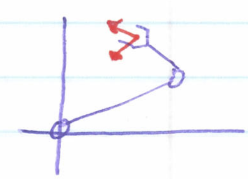
$\downarrow$ gives two parallel translation vectors

$\Rightarrow J(\alpha_1, 0)$ does not have full rank.

2) $\alpha = (\alpha_1, \pi/2)$; $J(\alpha) = \begin{bmatrix} -l_1 \sin(\alpha_1) - l_2\cos(\alpha_1) & -l_2 \sin(\alpha_1) \\ l_1 \cos(\alpha_1) - l_2 \sin(\alpha_1) & l_2 \cos(\alpha_1) \end{bmatrix}$

$\downarrow$ two non-parallel translation vectors.

$\Rightarrow J(\alpha_1, \pi/2)$ has full rank.



<u>Inverting the Manipulator Jacobian</u>:

- Jacobian gives end-effector velocity given joint configuration and joint velocity.
- the inverse <u>should</u> give a joint velocity given joint configuration and end-effector velocity.
  - there may be multiple solutions.
  - just pick one, whose selection can be picked based on some criteria.

The manipulator Jacobian in body/spatial form is:

$$\xi_e^b = J^b(\alpha)\cdot\dot{\alpha}\ ,\quad \xi_e^s = J^s(\alpha)\cdot\dot{\alpha}$$

Suppose that for our task, we are trying to control the end-effector velocity. We will need to convert these desired end-effector velocities into joint velocities to actually control the manipulator.

We will need (in body frame)

$$\dot{\alpha} = (\text{inverse of } J^b(\alpha))\cdot\xi^b$$

Can this really be done?

↳ Yes, and one such method is called the Moore-Penrose psuedo-inverse.

The psuedo-inverse computation varies according to the matrix and its properties:

Suppose we had $Ax=b$, and we want $A^{\#}$ such that $x = A^{\#}b$.

What is $A^{\#}$?

- this step gives the right-inverse to A.

$\boxed{1}$ Case 1, A is square and it's full rank, then $A^{\#} = A^{-1}$

$\boxed{2}$ Case 2, A is $n\times m$ with $n \le m$ and it's full (row) rank then

$$A^{\#} = A^T(AA^T)^{-1}$$

Note: $x = A^{\#}b$

$$\Rightarrow Ax = AA^{\#}b = AA^T(AA^T)^{-1}b = b$$

ECE4560

11/07/07
33-1

Psuedo-inverse continued...

(kin. sufficient) $\boxed{1}$ Case 1. A is $n \times m$ with $n = m$, and has full rank. then

$$A^{\#} = A^{-1}$$

(kin. redundant) $\boxed{2}$ Case 2. A is $n \times m$ with $n < m$ and has full (row) rank then

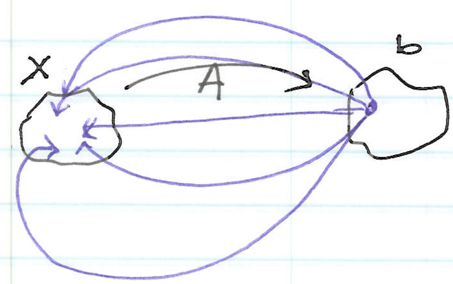
$$A^{\#} = A^T (AA^T)^{-1} \qquad \text{Right inverse}$$

(kin. insufficient) $\boxed{3}$ Case 3. A is $n \times m$ with $n > m$ and has full (column) rank then

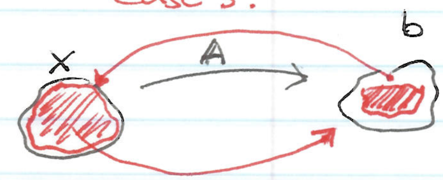
$$A^{\#} = (A^T A)^{-1} A^T \qquad \text{Left inverse}$$

↑ this third case should be avoided for (kinematically insufficient) manipulators, unless one is very careful about keeping ξ in the range space of $J / J^b / J^s$

case 2:



Case 3:



picks solution that gives the smallest magnitude

picks solution that minimizes the error $\|b - A^{\#}x\|$

So, basically we can do the following

$$\dot{\alpha} = (\text{inverse of } J^b(\alpha)) \cdot \xi^b$$

and what we do is use the Moore-Penrose psuedo-inverse:

$$\dot{\alpha} = (J^b(\alpha))^{\#} \cdot \xi^b$$

What good is it?

- it can convert velocity requirements/constraints on the group space to velocity requirements/constraints on the joints.

- Or if we have constraints on $\dot{\alpha}$, but we are given $\xi^{wb}(t)$, we can figure out $\dot{\alpha}$ for specific α.

- and, it can be used to generate joint trajectories given desired end-effector velocity (think feedback!)

TRAJECTORY DESIGN:

Goal: to design realistic/feasible trajectories connecting different configurations.

- joint configurations $(\alpha, \dot{\alpha})$
- group/end-effector configurations (g, ξ)
- both levels?

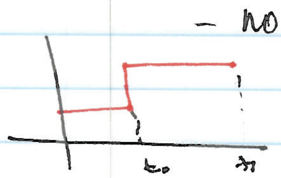
sometimes not possible using only one type of configuration space. May need to utilize both.

When dealing with trajectory design

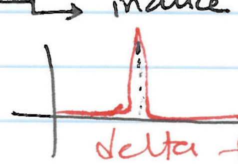
* we now consider the time history: position, velocity, acceleration

* we want to ensure sufficiently smooth trajectories (continuous derivatives)

- no jerky motion → cause wear → induce vibrations (excite resonance)



time-derivative →



← excites all frequencies

delta function

ECE 4560

continued...

- need to manage velocity better

let's add velocity constraints on initial and final conditions.

Cubic Polynomials / Splines

(*) $$p(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$ (scalar case)

Why cubic? → because of the necessary extrema (eg., # places where zero velocity is required)

→ because of the number of constraints being imposed.

4 constraints on the polynomial:

1) initial point p_i
2) final point p_f
3) initial velocity $\dot{p}_i$
4) final velocity $\dot{p}_f$

↳ so, minimally, a cubic polynomial is required

The constraints on p(t) work out to be

$$p(0) = p_i \qquad \dot{p}(0) = 0$$
$$p(t_f) = p_f \qquad \dot{p}(t_f) = 0$$

Given the form of p(t) from (*) we have

$$\dot{p}(t) = a_1 + 2a_2 t + 3a_3 t^2$$
$$\ddot{p}(t) = 2a_2 + 6a_3 t$$

$$\Rightarrow \begin{aligned} p(0) &= a_0 && = p_i \\ p(t_f) &= a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^2 && = p_f \\ \dot{p}(0) &= a_1 && = \dot{p}_i = 0 \\ \dot{p}(t_f) &= a_1 + 2a_2 t_f + 3a_3 t_f^2 && = \dot{p}_f = 0 \end{aligned}$$

$$\Rightarrow \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & t_f & t_f^2 & t_f^3 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 2t_f & 3t_f^2 \end{bmatrix}}_{A(t_f)} \vec{a} = \vec{p_0} \quad \text{where } \vec{a} = \begin{Bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{Bmatrix} \ \& \ \vec{p_0} = \begin{Bmatrix} p_i \\ p_f \\ 0 \\ 0 \end{Bmatrix}$$

$$\Rightarrow A(t_f)\vec{a} = \vec{p_0}$$

$$\Rightarrow \vec{a} = P(t_f)\vec{p_0} \quad \text{where } P(t_f) = A^{-1}(t_f)$$

$$P(t_f) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -3/t_f^2 & 3/t_f^2 & -2/t_f & -1/t_f \\ 2/t_f^3 & -2/t_f^3 & 1/t_f^2 & 1/t_f^2 \end{bmatrix} \leftarrow \text{general solution for a given } \vec{p_0}$$

- but, notice that $\vec{p_0}$ has 0's in the last two rows based on our setup. $\Rightarrow$ Last two columns of $P(t_f)$ are not needed.

$$\Rightarrow \vec{a} = P_{simp}(t_f)\begin{Bmatrix} p_i \\ p_f \end{Bmatrix} \quad \text{where } P_{simp}(t_f) = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ -3/t_f^2 & 3/t_f^2 \\ 2/t_f^3 & -2/t_f^3 \end{bmatrix}$$

worked out individually:

$$a_0 = p_i \qquad a_2 = 3/t_f^2\,(p_f - p_i)$$
$$a_1 = 0 \qquad a_3 = 2/t_f^3\,(p_i - p_f)$$

What value should t_f take?

- certainly should be short enough so that trajectory doesn't take "forever."
- not too short though: actuator limits, end-effector velocity/accel. limits
- have decent guess based on limits:

$$V_{lim} \cdot t = d$$

solving for t

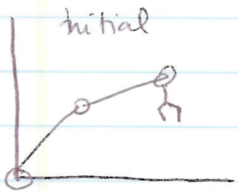
t — time; d — distance; V_{lim} — limit/nominal velocity

- want to add fudge factor

$$t_f = 1.2\, t$$

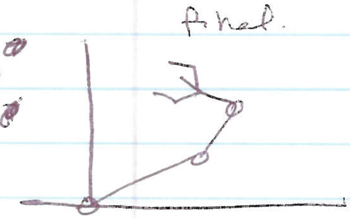
Example

initial



$$\vec{\alpha}_i = (\pi/3,\ -\pi/8,\ -\pi/2)$$
$$\vec{\alpha}_f = (\pi/8,\ \pi/6,\ \pi/2)$$

final



Goal: join initial & final joint configurations w/ trajectory whose velocity does not exceed $\frac{\pi}{4}$ rad./sec.

$$\left|\pi/8 - \pi/3\right| = \frac{5\pi}{24}$$
$$\left|\pi/6 - \left(-\frac{\pi}{8}\right)\right| = \frac{7\pi}{24}$$
$$\left|\frac{\pi}{2} - \left(-\frac{\pi}{2}\right)\right| = \pi$$

$\Rightarrow \pi = \max.$

$$\Rightarrow t_f = \frac{d}{V_{max}} = \frac{\pi}{\pi/4} = 4 \text{ sec.}$$

d — max. abs. joint difference

↳ solve for each polynomial

$$\alpha_1(t) = \frac{\pi}{3} - \frac{5\pi}{128}t^2 + \frac{5\pi}{768}t^3$$

$$\alpha_2(t) = -\frac{\pi}{8} + \frac{7\pi}{128}t^2 - \frac{7\pi}{768}t^3$$

$$\alpha_3(t) = -\frac{\pi}{2} + \frac{3\pi}{16}t^2 - \frac{\pi}{32}t^3$$

$$\ddot{\alpha}_3(t) = \frac{6\pi}{16} - \frac{6\pi}{32}t$$

$$\Rightarrow \ddot{\alpha}_3(t) = 0 \quad \text{for} \quad t^* = \frac{6\pi}{16} \cdot \frac{32}{6\pi} = 2$$

$$\Rightarrow \dot{\alpha}_3(t^*) = \frac{6\pi}{16}t^* - \frac{3\pi}{32}(t^*)^2 = \frac{6\pi}{16}\cdot 2 - \frac{3\pi}{32}\cdot 4$$

$$= \frac{3\pi}{8} > \frac{\pi}{4}$$

$\Rightarrow$ going too fast (past desired limit)

extending t_f to be 7 reduces velocity below $\pi/4$ rad./sec

$$\Rightarrow \alpha_1(t) = \frac{\pi}{3} - \frac{5\pi}{392}t^2 + \frac{5\pi}{4116}t^3$$

$$\alpha_2(t) = -\frac{\pi}{8} + \frac{\pi}{5}t^2 - \frac{\pi}{588}t^3 \qquad \text{for } t \in [0,7]$$

$$\alpha_3(t) = \frac{-\pi}{2} + \frac{3\pi}{49}t^2 - \frac{2\pi}{343}t^3$$