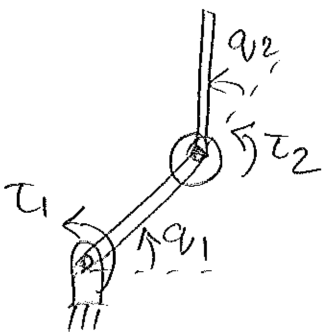
Lecture # 9

Adaptive Robot Control

(Chapter 9)



General form of eqns

$$\left.\begin{array}{l} H(\underline{q})\ddot{\underline{q}} + C(\underline{q},\dot{\underline{q}})\dot{\underline{q}} + D(\underline{q},\dot{\underline{q}})\dot{\underline{q}} \\ \quad + g(\underline{q}) \end{array}\right\} = \tau$$

$\underline{q}$ = vector of joint angles

τ = torques

H : inertia matrix

symm. p.d.

$$\exists \alpha > 0, \ \forall \underline{q}, \ H(\underline{q}) \geq \alpha I$$

$C(\underline{q},\dot{\underline{q}})\dot{\underline{q}}$: depends on $\underline{q}$
Quadratic on $\dot{\underline{q}}$ $\Big\}$ centripetal & Coriolis torques.

$D(\underline{q},\dot{\underline{q}})\dot{\underline{q}}$: viscous friction

$$D(\underline{q},\dot{\underline{q}}) \geq 0$$

p.s.d.

$g(\underline{q})$: gravitational torque.

Let's look @ position control first.

Constant q_d (desired position) in the presence of a load.

At the time (80s), it was difficult.

- nonlinear
- uncertainty
- multi-dimensional

[In 60s, industry used just P.D. control and it worked well.]

But they had large gear ratios G_i (transmission)

⇒ nonlinearities divided by G_i, or G_i^2 depending on the parameter.

⇒ mostly linear; less sensitivity to outside world

In mid-1980s, direct drive motors came into robotics.

⇒ direct contact w/ external world.

But PD control still worked! ⇒ in industry.

So this "frustrated" the researchers.

This forced robotics researchers to look into control aspects again.

Consider $\tau_i = \overbrace{-k_{P_i} \tilde{q}_i}^{P} - \overbrace{k_D \dot{q}_i}^{D}$ for the robot in the horizontal plane (no gravity)

Why does it work?

It behaves like a virtual spring + virtual friction/damping forces

"Virtual Physics"

+ its programmable!

This is now starting to look like our previously tried Lyapunov methods. Furthermore, it is autonomous.

what should V be? (assume $g = 0$)

$$KE = \frac{1}{2}\dot{q}^T H \dot{q}$$

But we shd consider virtual physics energy also.

$$V = \frac{1}{2}\underline{\dot{q}}^T H \underline{\dot{q}} + \frac{1}{2}\underline{\tilde{q}}^T K_p \underline{\tilde{q}}, \quad K_p > 0$$

more generally any s.p.d. matrix is fine also.

$$V \to \infty \text{ as } \begin{pmatrix}\tilde{q}\\ \dot{q}\end{pmatrix} \to \infty \qquad (\dot{q}_d = 0)$$

$$\dot{V} = \underline{\dot{q}}^T(\tau - D\underline{\dot{q}} - \underline{g}) + \underline{\dot{q}}^T K_p \underline{\tilde{q}}$$

$$= \underline{\dot{q}}^T(-\cancel{K_p\tilde{q}} - K_D\underline{\dot{q}} - D\dot{q}) + \cancel{\underline{\dot{q}}^T K_p \underline{\tilde{q}}}$$

$$= -\underline{\dot{q}}^T(K_D + D)\dot{q}$$

= virtual energy dissipated!

Could have avoided all the math!

Note conservation of energy $\frac{d}{dt}(KE)$ = external power input

= gravity + friction + torques

ignore for now

$$= \underline{\dot{q}}^T(\tau - D\dot{q} - \underline{g})$$

Which theorem to use? $V \to \infty$, $\dot{V} \leq 0$ } $\to$ use global invariant set theorem

$\to$ sys converges to largest invariant set inside $R = \{\dot{V} = 0\}$

Note $\dot{V} = 0 \Rightarrow \dot{\underline{q}} = 0$ } This does not mean $\tilde{q} \to 0$ though

$\downarrow$

because $K_D + D > 0$

System dynamics reduces to:

$$\Rightarrow \quad H\ddot{\underline{q}} = -K_P \tilde{\underline{q}}$$

$$\ddot{\underline{q}} = -H^{-1} K_P \tilde{\underline{q}} \neq 0 \quad \text{unless } \tilde{\underline{q}} = 0 \text{ or } \underline{q} = \underline{q}_d$$

$\to$ system does not get stuck @ $\dot{q} = 0$ except if $\tilde{q} = 0$

This type of controller works well in most cases, but does not say much about transients.

Trajectory Tracking (section 9.2)

Now $q_d(t)$; may need to adapt to load.

What are the unknowns?

mass. (1 #)

inertia matrix: symm constant 3x3 matrix
6 params.

center of mass (3 #s)

$\Rightarrow$ Total of 10 params.

Main difference w/ what we have seen before $\to$ in adaptive control:

Now multidimensional and all ~~##~~ coupled.

Now

$$\frac{1}{2}\frac{d}{dt}\left(\dot{q}^T H \dot{q}\right) = \dot{q}^T\left(\overbrace{\tau - D\dot{q} - g}^{\text{cancel w/ below}}\right)$$

$$\Rightarrow \quad \dot{q}^T H(q)\ddot{q} + \frac{1}{2}\dot{q}^T \dot{H}\dot{q} = \text{RHS}$$

$$= \dot{q}^T\left(\underbrace{\tau - D\dot{q} - g}_{\text{cancel}} - C\dot{q}\right) + \frac{1}{2}\dot{q}^T\dot{H}\dot{q} = \text{RHS.}$$

$$\Rightarrow \quad \dot{q}^T\left(\dot{H} - 2C\right)\dot{q} = 0$$

for any $\dot{q}$

$\Rightarrow$ $\dot{H} - 2C$ is skew symmetric.

"Matrix Representation" of energy conservation that will be useful in adaptive control.

Now, when you derive the equations of motion

you get $C\dot{q}$, ... not C uniquely.

So you need some additional formula to compute C.

$$C_{ij} = \frac{1}{2}\dot{H}_{ij} + \frac{1}{2}\sum_{k=1}^{n}\left(\frac{\partial H_{ik}}{\partial q_j} - \frac{\partial H_{jk}}{q_i}\right)\dot{q}_k$$

skew symmetric.

Now lets do adaptive control: easy to expand to multidimensional case (p6)

$$\underline{s} = \dot{\tilde{\underline{q}}} + \underbrace{\lambda}_{\text{constant} > 0} \tilde{\underline{q}} \qquad \text{or} \qquad \underline{s} = \dot{\tilde{\underline{q}}} - \underbrace{\Lambda}_{\text{stable}} \tilde{\underline{q}}$$

$$s^2 \longrightarrow \underline{s}^T \underline{s}$$

$$J s^2 \longrightarrow \underline{s}^T H \underline{s}$$ energy like; exploit system physics.

$$\dot{V} = \underline{s}^T H \dot{\underline{s}} + \frac{1}{2} \underline{s}^T \dot{H} \underline{s}$$

[choose $s = \dot{q} - \dot{q}_r$]

$$= \underline{s}^T (H \ddot{\underline{q}} - H \ddot{\underline{q}}_r) + \frac{1}{2} \underline{s}^T \dot{H} \underline{s}$$

$$= \underline{s}^T (\underline{\tau} - H \ddot{\underline{q}}_r - C \dot{\underline{q}} - D \dot{\underline{q}} - \underline{g}) + \underbrace{\frac{1}{2} \underline{s}^T \dot{H} \underline{s}}_{\text{find "a } C\text{" to cancel this}}$$

$$\dot{\underline{q}} = \underline{s} + \dot{\underline{q}}_r$$

$$= \underline{s}^T (\underline{\tau} - H \ddot{\underline{q}}_r - C \dot{\underline{q}}_r - D \dot{\underline{q}} - \underline{g}) + \cancel{\frac{1}{2} \underline{s}^T (\dot{H} - 2C) \underline{s}} \rightarrow 0$$

Similar in form to what we have seen before:

[known matrix] [unknown]

$$Y(\underline{q}, \dot{\underline{q}}, \underline{q}_r, \dot{\underline{q}}_r)\, \underline{a}$$