Lecture # 7

Adaptive Control

Last time: $x^{(n)} = f(x,t) + u$

$$\tilde{x}(t) = x(t) - x_d(t)$$

↑ track trajectory.

Reduce nth order system to first order system

$$s = \left(\frac{d}{dt} + \lambda\right)^{n-1} \tilde{x}$$

properties: $\dot{s}$ contains u

$s \to 0 \Rightarrow \tilde{x} \to 0$

Of course, there is uncertainty in f

$$\left|\hat{f}(x,t) - f(x,t)\right| \leq F(x,t)$$

Use $u = \hat{u} - k\,\mathrm{sgn}(s) \Rightarrow$ achieve zero tracking error w/ $k = F + \eta$

But this leads to chattering.

Replace w/ $u = \hat{u} - (k - \dot{\phi})\,\mathrm{sat}\left(\frac{s}{\phi}\right)$ boundary layer interpolation

Today, we will study what we can do if

$|f-\hat{f}|$ has constants? Can we exploit such constants? (to get better tracking)

For example,

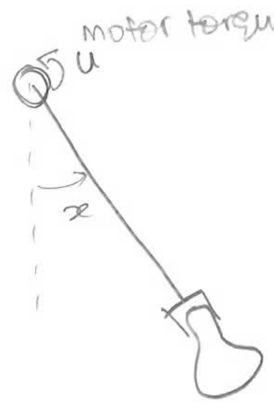
$$f = a\cos x + \cdots$$

$\downarrow$

unknown.
Can we control system as though we know the constant?

For example, we do this in linear systems. PID control to reject constant disturbance.

<u>For example</u> One-link robot (pendulum)



Eqn's of motion

$$J\ddot{x} + \underbrace{b\dot{x}|\dot{x}|}_{\text{nonlinear}} + mgl\sin x = u$$

J, b, m, l are unknown

But we want sys. to follow arbitrary trajectory.

Note: There are only three unknowns really in the eqn

$a_1 = J$

$a_2 = b$

$a_3 = ml.$

Again use the same idea as in robust control (replace n^{th} order sys. w/ first-order system)

Define

$$s = \dot{\tilde{x}} + \lambda \tilde{x}$$

$$= \dot{x} - \underbrace{\dot{x}_r}_{\dot{x}_d - \lambda \tilde{x}}$$

$$\ddot{x}_r = \ddot{x}_d - \lambda \dot{\tilde{x}}$$

Since we have a non-autonomous theorem,

use Barbalat's theorem ($V(x,t)$ lower bounded
$\dot{V} \leq 0$
$\ddot{V}$ bounded)
$\Rightarrow \dot{V} \to 0$

Let's see what happens if

$$V = \frac{1}{2} J s^2$$

$$\dot{V} = s J \dot{s}$$

What is $J\dot{s}$?

$$J\dot{s} = J(\ddot{x} - \ddot{x}_r)$$

$$= u - J\ddot{x}_r - b\dot{x}|\dot{x}| - mgl \sin x$$

(unknown qty) × (known qty)

$$= u - \underbrace{\begin{bmatrix} \ddot{x}_r & \dot{x}|\dot{x}| & \sin x \end{bmatrix}}_{Y} \underbrace{\begin{bmatrix} J \\ b \\ -mgl \end{bmatrix}}_{a}$$

$$J\dot{s} = u - Y\underline{a}$$

$$\mathring{V} = s(u - Y\underline{a})$$

u: known; $\underline{a}$: unknown

Assume, for a moment, that $\underline{a}$ is known.

Then we can set $u = Y\underline{a} - ks$, with k constant > 0

$$\Rightarrow \mathring{V} = -ks^2$$

Barbalat theorem: $\mathring{V} \to 0 \Rightarrow s \to 0 \Rightarrow \tilde{x} \to 0$ and $\mathring{\tilde{x}} \to 0$

Rough work

$s(Y\hat{a} - ks - Ya)$

$-ks^2 - sY(\hat{a} - a)$

But of course we do not know a.

Let's use an estimate $\underline{\hat{a}}$ so that $u = Y\hat{a} - ks$.

Then we have

$$\mathring{V} = -ks^2 + \underbrace{sY\tilde{a}}_{\text{note: this can be any sign, which is a problem.}}, \quad \text{where } \underbrace{\underline{\tilde{a}} = \underline{\hat{a}} - \underline{a}}_{\text{parameter estimation error}}$$

Can we add something to V so that we get rid of $sY\tilde{a}$?

Let's try $V = \frac{1}{2}Js^2 + \frac{1}{2}\tilde{a}^T \underbrace{P^{-1}}_{\text{constant symm. p.d}} \tilde{a}$ ($\rightarrow$ even w/ this additional term V must be lower bounded)

$\rightarrow$ for weighting, may be even diagonal

For this to work,

$\tilde{a}$ shd be time varying.

So $\hat{a}$ must " " also.

So $\frac{d}{dt}\left(\frac{1}{2}\underline{\tilde{a}}^T P^{-1}\underline{\tilde{a}}\right) = \frac{1}{2}\underline{\mathring{\tilde{a}}}^T P^{-1}\underline{\tilde{a}} + \frac{1}{2}\underline{\tilde{a}}^T \overbrace{\cancel{\frac{d(P^{-1})}{dt}}}^{0 \text{ since } P^{-1} = \text{const}}\underline{\tilde{a}}$

$+ \frac{1}{2}\underline{\tilde{a}}^T P^{-1}\underline{\mathring{\tilde{a}}}$

(Note $P^{-1} = P^{-T}$ since P is symmetric)

$= \underline{\mathring{\tilde{a}}}^T P^{-1}\underline{\tilde{a}}$

$= \mathring{\hat{a}}^T P^{-1}\tilde{a}$ (since $\underline{a}$ = constant)

$\Rightarrow \mathring{V} = sJ\mathring{s} + \mathring{\hat{a}}^T P^{-1}\underline{\tilde{a}}$

$= -ks^2 + \underbrace{sY\tilde{a} + \mathring{\hat{a}}^T P^{-1}\tilde{a}}_{\text{we want this to go to zero}}$

$= (sY + \mathring{\hat{a}}^T P^{-1})\tilde{a}$

Solve for $\mathring{\hat{a}}^T$ such that $\left.\right\}$ $sY + \mathring{\hat{a}}^T P^{-1} = 0$

$\mathring{\hat{a}}^T = -sYP$

$\Rightarrow \boxed{\mathring{\hat{a}} = -P\,Y^T s}$ $\qquad P = P^T$

Therefore

Control Law $\boxed{u = Y\hat{a} - ks}$

Adaptation Law. $\boxed{\dot{\hat{\underline{a}}} = -P Y^T s}$

This is powerful!

Start w/ arbitrary trajectory
Start w/ any parameter
→ sys will end up on trajectory!

Note that $\dot{\hat{\underline{a}}} = -P Y^T s$ is a generalization of the PID controller.

Specifically, if $Y(3) = \text{constant} = 1$ (instead of sin x)

Then the corresponding term of $\dot{\hat{a}}$ for $\hat{a}$: $\dot{\hat{a}}(3) = \text{constant}$

$\Rightarrow \hat{a} = \text{constant}$

So $$u = \underbrace{-ks}_{\text{proportional term}} - Y \underbrace{\int_0^t P Y^T s \, dt}_{\text{Integral term.}}$$

So what will $\underline{\hat{a}}$ converge to?

$\hat{a}$ does not necessarily converge to the right value.

The algo only finds $\hat{a}$ to drive $\begin{pmatrix} \tilde{x} \to 0 \\ \dot{\tilde{x}} \to 0 \end{pmatrix}$ task is achieved.

In the case of robot arm movements, the algo may find the parameters to drive error to zero only for one plane of rotations. It does not know what params are needed for another plane

Adaptation on a "need to know" basis.

– "sufficient richness" (this is a feature. Not a bug!)

If trajectory is so complicated that the true params are reqd, then algo will converge.

We know $s \to 0$

we also know $\underbrace{\dot{s} \to 0}_{\text{Barbalat}}$ ($\ddot{s}$ is bounded)

$$\left.\begin{aligned} J\dot{s} &= u - Y\underline{a} \\ u &= Y\underline{\hat{a}} - ks \end{aligned}\right\} \Rightarrow J\dot{s} = Y(\underline{\hat{a}} - \underline{a}) - ks$$

$$\Rightarrow J\dot{s} + ks = \cancel{Y}\tilde{a}$$

Thus, $Y\tilde{a} \to 0$

Note $Y \to Y_d$ over time.

$\Rightarrow \quad Y_d \tilde{a} \to 0$

This makes sense because $J\dot{s} + Ks$ is a linear system

driven by input $Y\tilde{a}$

$Y_d \tilde{a} = 0$ one equation w/ 3 unknowns

Thus, the sol'n is not unique.

and $\tilde{a}$ not necessarily $= 0$

Now if $\|Y_d \tilde{a}\| = 0 = \tilde{a}^T \underbrace{Y_d^T Y_d}_{} \tilde{a}$

square matrix } positive s.d.

(it has rank utmost 1)

– not p.d.

So what is this "sufficient richness" condition?

At $t = \infty$

– Trajectory still moving, but parameters have converged

Even though $Y_d^T Y_d$ has rank 1, the time integral of $Y_d^T Y_d$

may have full rank.

$$\frac{1}{3}\left[\underbrace{\begin{pmatrix}1&0&0\\0&0&0\\0&0&0\end{pmatrix}}_{\text{rank } 1}+\underbrace{\begin{pmatrix}0&0&0\\0&1&0\\0&0&0\end{pmatrix}}_{\text{rank } 1}+\underbrace{\begin{pmatrix}0&0&0\\0&0&0\\0&0&1\end{pmatrix}}_{\text{rank}}\right]$$

$$= \frac{1}{3}\underbrace{I}_{\text{rank } 3}$$

So if the $\xrightarrow{}$ desired trajectory is rich enough, the params will converge.

$$\hat{\underline{a}} \to \underline{a} \quad \text{if} \quad \begin{array}{l}\exists\, t_0 \geq 0\\ \exists\, T > 0\\ \exists\, \alpha > 0\end{array}, \quad \forall t \geq t_0, \ \underbrace{\frac{1}{T}\int_t^{t+T} Y_d^T Y_d}_{\text{average over a time window}} \geq \alpha I$$

$\left(\begin{array}{l} M_1 \geq M_2 \\ M_1 - M_2 \ \text{p.s.d} \end{array} \right)$

In some applications, you only care about the task.

" " " , " care about the parameters also.