Lecture #18

What really is "relative degree":

If you have o/p eqn., $y = h(\underline{x})$
$$= C\,\underline{x}$$

$$\Rightarrow \underbrace{\dot{y}}_{1\times 1} = C\,\dot{x}$$
$$= C(Ax + Bu)$$
$$= \underbrace{C}_{1\times n}\underbrace{A}_{n\times n}\underbrace{x}_{n\times 1} + \underbrace{C}_{1\times n}\underbrace{B}_{n\times 1}\underbrace{u}_{1\times 1}$$

CB — Scalar

$$\underbrace{\dot{x}}_{n\times 1} = Ax + \underbrace{B}_{n\times 1}\underbrace{u}_{1\times 1}$$

If $CB \neq 0$

then $r = 1$

relative degree

$\therefore$ Rel. deg. = # of times we need to take $\frac{d}{dt}$ of o/p eqn. so that "u" appears on the RHS.

If $CB = 0$,

$$\ddot{y} = CA\dot{x} + 0$$
$$= CA(Ax + Bu) = CA^2x + CABu$$

If $CAB \neq 0$, then $r = 2$.

for LTI:

$$y^{(r)} = CA^r x + \underbrace{CA^{r-1}B}_{\neq 0} u$$

Rel. deg. = r

Example 2 (nonlin)

$$\dot{x}_1 = x_3 - x_2^3$$
$$\dot{x}_2 = -x_2 - u$$
$$\dot{x}_3 = x_1^2 - x_3 + u$$

& $y = h(x) = x_1$

SISO

$$\dot{y} = \dot{x}_1 = x_3 - x_2^3$$

$$\ddot{y} = \dot{x}_3 - 3x_2^2 \dot{x}_2$$
$$= x_1^2 - x_3 + u - 3x_2^2(-x_2 - u)$$
$$= (x_1^2 - x_3 + 3x_2^3) + (1 + 3x_2^2)u$$

$\therefore$ Rel. deg. $r = 2$ globally

Example 3 : (Vander Pol Oscillator + Control)

$$\underline{\dot{x}} = \underline{f}(\underline{x}) + \underline{g}(\underline{x})\, u$$

$$= \begin{pmatrix} x_2 \\ 2\omega\zeta(1-\mu x_1^2)x_2 - \omega^2 x_1 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u$$

Again <u>SISO</u>

$$y = h(x_1, x_2) = x_1$$

$$L_g h(x) = \frac{\partial h}{\partial x} g(x) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 0$$

$$L_f h(x) = \frac{\partial h}{\partial x} f(x) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{pmatrix} x_2 \\ 2\omega\zeta(1-\mu x_1^2)x_2 \\ -\omega^2 x_1 \end{pmatrix}$$

$$= x_2$$

$$L_g L_f h(x) = \frac{\partial}{\partial x}(L_f h)\, g(x) = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 1$$

$$\therefore\ r = 2 \qquad \forall\ x = x_0 \in \mathbb{R}^2$$

exercise for prev example, if we do

$$y = h(\underline{x}) = \sin x_2$$

then $L_g h(x) = \cos x_2$

$\Rightarrow r = 1 \quad \forall \quad \underline{x}^0 = \begin{pmatrix} x_1^0 \\ x_2^0 \end{pmatrix}$ s.t.

$$x_2^0 \neq (2k+1)\pi/2$$

when violated then r is undefined.

In general:

$$\underline{\dot{x}} = f(\underline{x}) + g(\underline{x})u \qquad \underline{x} \in \mathbb{R}^4$$

$$y = h(\underline{x}) \qquad u, y \in \mathbb{R}$$

$$y = h(x)$$

$$\Rightarrow \dot{y} = \frac{\partial h}{\partial x}\dot{x} = \frac{\partial h}{\partial x}\left(f(\underline{x}) + g(\underline{x})u\right)$$

$$= L_f h(x) + L_g h(x)\, u$$

$\neq 0$, then $r = 1$

else if $L_g h(x) = 0$ then

$$\dot{y} = L_f h(x) \quad (\text{indep. of } u)$$

Then do

$$\ddot{y} = \frac{\partial}{\partial x}(L_f h)\,\dot{x} \qquad (f(x)+g(x)u)$$

$$= L_f^2 h(x) + L_g L_f h(x)$$

$\vdots$

continue

$$L_g L_f^k h(x) = 0 \quad \forall\, k < r-1 \quad \text{locally near } \underline{x} = \underline{x}^0$$

$$\& \quad L_g L_f^{r-1} h(\underline{x}^0) \neq 0$$

Then $y^{(r)} = L_f^r h(x) + \underbrace{L_g L_f^{r-1} h(x)}_{\neq 0}\, u$

~~Thm~~ Suppose rel. deg. $= r < n$

Then can do transformation:

new variable $z_i := \tau_i(\underline{x})$, $1 \le i \le n$

$$\tau_1(\underline{x}) = h(\underline{x})$$
$$\tau_2(\underline{x}) = L_f h(\underline{x})$$
$$\vdots$$
$$\tau_r(x) = L_f^{r-1} h(\underline{x})$$

will then define $\underbrace{\tau_{r+1}(\underline{x}), \ldots, \tau_n(\underline{x})}_{n-r \text{ more functions}}$

such that

$$\underline{\underline{z}} = \tau(\underline{x}) = \begin{pmatrix} \tau_1(\underline{x}) \\ \vdots \\ \tau_n(x) \end{pmatrix}$$

has Jacobian @ $\underline{x} = \underline{x}^0$ non-singular.

Moreover, it is possible to choose these additional fns $\tau_{r+1}, \ldots \tau_n$ s.t.

$$L_g \tau_i(x) = 0$$

$\forall\ r+1 \le i \le n$
$\forall\ x$ around $\underline{x_0}$.

In new variables $z_1, \ldots, z_n$:

$$\dot{z}_1 = \left(\frac{\partial \tau_1}{\partial \underline{x}}\right)^T \frac{d\underline{x}}{dt} = \left\langle \frac{\partial h}{\partial \underline{x}}, \frac{d\underline{x}}{dt} \right\rangle = L_{\underline{f}} h(x) = z_2$$

$$\dot{z}_2 = \frac{d}{dt}\left(L_{\underline{f}} h(\underline{x})\right) = z_3$$

$$\beta(x) := +\frac{1}{a(z)} = +\frac{1}{a(\tau(x))}$$

$$\vdots$$

$$\dot{z}_{r-1} = \tau_r(\underline{x}) = \underline{L_f^{r-1} h(\underline{x})} = z_r$$

$$\dot{z}_r = L_f^r h(x) + L_g L_f^{r-1} h(x)\, u$$

$$\boxed{\underline{x} = \tau^{-1}(\underline{z})}$$

$$= \underbrace{L_f^r h(\tau^{-1}(z))}_{b(z)} + \underbrace{L_g L_f^{r-1} h(\tau^{-1}(z))}_{a(z)}\, u$$

$$= \underbrace{b(z) + a(z)\, u}_{v}$$

where $\alpha(x) := -\frac{b(z)}{a(z)} = \frac{-b(\tau(x))}{a(\tau(x))}$

$$u = \frac{1}{a(z)} v - \frac{b(z)}{a(z)} = \alpha(x) + \beta(x)\, v$$

(contd.)

For $\dot{z}_{r+1}, \ldots, \dot{z}_n$, we cannot expect any spl. structure.

BUT. if we choose

$\tau_{r+1}(\underline{x}), \ldots, \tau_n(\underline{x})$ s.t.

$\boxed{L_g \tau_i(x) = 0}$ then

$$\frac{dz_i}{dt} = \frac{\partial \tau_i}{\partial \underline{x}} \left(f(\underline{x}) + g(\underline{x}) u \right)$$

$$= L_f \tau_i(x) + \cancel{L_g \tau_i(x)}^{0} u$$

$$= L_f \tau_i(x)$$

$$= L_f \tau_i(\underline{\tau}^{-1}(\underline{z})) =: q_i(z)$$

$\forall\ r+1 \leq i \leq n$

$$
\Rightarrow \quad \underbrace{\left.\begin{aligned}
\dot{z}_1 &= z_2 \\
\dot{z}_2 &= z_3 \\
&\vdots \\
\dot{z}_{r-1} &= z_r \\
\dot{z}_r &= \boxed{b(z) + a(z)\,u} \\
\dot{z}_{r+1} &= q_{r+1}(z) \\
&\vdots \\
\dot{z}_n &= q_n(z)
\end{aligned}\right\}}_{}
\qquad
\begin{aligned}
\dot{z} &= \hat{f}(z) + \hat{g}(z)\,u \\
y &= h(x) \\
&= z_1
\end{aligned}
$$

"Normal form" of SISO control system.

(Input-output linearization)

Example:

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{pmatrix} = \begin{pmatrix} -x_1 \\ x_1 x_2 \\ x_2 \end{pmatrix} + \begin{pmatrix} \exp(x_2) \\ 1 \\ 0 \end{pmatrix} u$$

$$y = h(x) = x_3$$

SISO.

Show that, can choose

$$z_1 = \tau_1(x) = h(x) = x_3$$

$$z_2 = \tau_2(x) = L_f h(x) = x_2$$

we need τ_3 s.t. $\frac{\partial \tau_3}{\partial x} g(x) = 0$

such a choice of τ_3 is: $\tau_3(x) = 1 + x_1 - \exp(x_2)$

Then show:

$$\dot{z}_1 = z_2$$

$$\dot{z}_2 = \left(-1 + z_3 + \exp(z_2)\right) z_2 + u$$

$$\dot{z}_3 = \left(1 - z_3 - \exp(z_2)\right)\left(1 + z_2 \exp(z_2)\right)$$

Feedback Linearization is then a spl. case of I/O. linearization :

If $r = n$ @ some $\underline{x} = \underline{x}^0$ then

rel. degree — state dim.

$$z = \tau(\underline{x}) = \begin{pmatrix} \tau_1(\underline{x}) \\ \vdots \\ \tau_n(\underline{x}) \end{pmatrix} = \begin{pmatrix} h(\underline{x}) \\ L_f h(\underline{x}) \\ L_f^2 h(\underline{x}) \\ \vdots \\ L_f^{n-1} h(\underline{x}) \end{pmatrix}$$

In the new co-ord :

$$\begin{aligned} \dot{z}_1 &= z_2 \\ \dot{z}_2 &= z_3 \\ &\vdots \\ \dot{z}_{n-1} &= z_n \\ \dot{z}_n &= \underbrace{b(z) + a(z)\, u}_{v} \end{aligned}$$

NOTE : @ $\underline{z}^0 = \tau(\underline{x}^0)$

$a(z^0) \neq 0$ by defn.

Can show: this linear system is <u>controllable</u>.

Can now design your favorite lin controller.

<u>Exercise</u>:

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{pmatrix} = \begin{pmatrix} 0 \\ x_1 + x_2^2 \\ x_1 - x_2 \end{pmatrix} + \begin{pmatrix} \exp(x_2) \\ \exp(x_2) \\ 0 \end{pmatrix} u$$

$$y = x_3 \quad , \quad \underline{\text{SISO}}.$$

<u>Prove that</u>.

(1) new coord. transformⁿ

$$\begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} = \underbrace{\begin{pmatrix} h(x) \\ L_f h(x) \\ L_f^2 h(x) \end{pmatrix}}_{\underline{T}(\underline{x})} = \begin{pmatrix} x_3 \\ x_1 - x_2 \\ -x_1 - x_2^2 \end{pmatrix}$$

(2) $u(x) = \alpha(x) + \beta(x) v$

where $\alpha(x) = \dfrac{-2x_2(x_1 + x_2^2)}{(1+2x_2)\, e^{x_2}}$, $\beta(x) = \dfrac{-1}{(1+2x_2)e^{x_2}}$

& consequently,

$$\begin{pmatrix} \dot z_1 \\ \dot z_2 \\ \dot z_3 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} v$$

So far used $h(\underline{x})$ to make this work.
(i.e.) <u>sufficient cond</u> (Turns out it's also necessary).

Consider $\underline{\dot x} = f(\underline{x}) + g(\underline{x}) u$, $\quad \underline{x} \in \mathbb{R}^n$, $u \in \mathbb{R}$.

Given some $\underline{x} = \underline{x}^0$, find, if possible

$\rightarrow$ a domain $\mathcal{D}_0$ around x_0 s.t.

the feedback $u = \alpha(x) + \beta(x) v$ on $\mathcal{D}_0$

& state transform^{n} $z = \tau(x)$ (also defined on $\mathcal{D}_0$)

s.t. the new closed loop system

$$\dot{x} = f(x) + g(x)\,\alpha(x) + g(x)\,\beta(x)\,v$$

in new z-coord. is lin. & controllable.

Thm. The above problem is solvable iff $\boxed{\exists\ \lambda(\underline{x})}$ on $\mathcal{D}_0$ s.t.
(scalar field)

$$\dot{x} = f(x) + g(x)\,u$$
$$y = \lambda(x)$$

has $\rho = n$ @ $x = x_0$

$\Updownarrow$

This existence is equiv. to 2 constructive cond^{ns}.

(i) The matrix $M(\underline{x}) := \left[\, g(\underline{x}^0) \,\middle|\, ad_f\, g(x^0) \,\middle|\, \ldots \,\middle|\, ad_f^{n-1} g(x_0) \,\right]$ has rank n.

(ii) the distribution $\Delta := \operatorname{span}\{g, ad_f\, g, \ldots, ad_f^{n-2} g\}$ is involutive near x_0

This gives recipe

$\boxed{\text{Step 1}}$ Given f & g, construct the vector fields

$g(x), \; ad_f\, g(x), \; \ldots, \; ad_f^{n-1} g(x)$

& check conditions (i) & (ii) in prev. page.

(IF answer is Yes (conditions are satisfied) then the system is Feedback Linearizable)

IF YES

$\boxed{\text{STEP 2}}$ Solve for $\lambda(\underline{x})$ from the PDE:

$$L_g \lambda(\underline{x}) = L_{ad_f g} \lambda(\underline{x}) = \ldots = L_{ad_f^{n-2} g} \lambda(x) = 0$$

$$L_{\underbrace{[f,[f,[f,[\ldots,[f,g]]}_{n-2}} \qquad \left(\& \; L_{ad_f^{n-1} g} \lambda(x^0) \neq 0 \right)$$

STEP 3 : Set

$$\left.\begin{aligned} \alpha(x) &:= \frac{-L_f^n \lambda(\underline{x})}{L_g L_f^{n-1} \lambda(\underline{x})} \\ \beta(x) &:= \frac{1}{L_g L_f^{n-1} \lambda(\underline{x})} \end{aligned}\right\} u = \alpha(x) + \beta(x) v$$

STEP 4

$$\tau(\underline{x}) = \begin{pmatrix} \lambda(\underline{x}) \\ L_f \lambda(\underline{x}) \\ \vdots \\ L_f^{n-1} \lambda(\underline{x}) \end{pmatrix}$$

This $(\tau, (\alpha, \beta))$ will do the feedback linearization.