"State Space" Models

The state space form of the discretized controller equations is mirrored in the form of the continuous implementation eqns:

$$u(t) = c_0 e(t) + \sum_{i=1}^{M} c_i x_i(t)$$

$$\dot{x}_i(t) = a_i x_i(t) + e(t) \qquad i = 1, \ldots, M$$

$\Rightarrow$

$$u(t) = C\underline{x}(t) + De(t)$$
$$\dot{\underline{x}}(t) = A\underline{x}(t) + Be(t)$$

where $C = [c_1 \; c_2 \; \ldots \; c_M] \qquad D = c_0$

$$A = \text{diag}\{a_1, a_2, \ldots, a_m\} \quad \text{and}$$

$$B = [1 \;\; 1 \;\; 1 \; \cdots \; 1]^T \quad \text{(column vector)}$$

In fact, this is only one of many possible ways we could write the equations in this form.

Suppose generally:

$$U(s) = \underbrace{\left[c_0 + \frac{c'_{M-1}s^{M-1} + \cdots + c'_1 s + c'_0}{s^M + a_{M-1}s^{M-1} + \cdots + a_1 s + a_0} \right]}_{H(s)} E(s)$$

Let

$$Z(s) = \left[\frac{1}{s^M + a_{M-1}s^{M-1} + \cdots + a_1 s + a_0} \right] E(s)$$

Then $\boxed{U(s) = c_0 E(s) + c'_{M-1} s^{M-1} Z(s) + \cdots + c'_1 s Z(s) + c'_0 Z(s)}$

Now let $X_1(s) = Z(s)$

$$X_2(s) = s Z(s) = s X_1(s)$$

$$\vdots$$

$$X_M(s) = s^{M-1} Z(s) = s X_{M-1}(s)$$

Now look at $Z(s)$

$$\underbrace{s^m Z(s)}_{= s\left[\underbrace{s^{m-1}Z(s)}_{X_m(s)}\right]} = E(s) - a_{m-1}\underbrace{s^{m-1}Z(s)}_{X_m(s)} - \cdots - a_1 \underbrace{sZ(s)}_{X_2(s)} - a_0 \underbrace{Z(s)}_{X_1(s)}$$

Inverse transform:

$$\dot{x}_m(t) = e(t) - a_{m-1}x_m(t) - \cdots - a_1 x_2(t) - a_0 x_1(t)$$

Where

$$x_2(t) = \dot{x}_1(t)$$

$$x_3(t) = \dot{x}_2(t)$$

$$\vdots$$

$$x_m(t) = \dot{x}_{m-1}(t)$$

Then again

$$u(t) = c_0 e(t) + \sum_{i=1}^{m} c'_{i-1} x_i(t)$$

But now

$$\dot{x}_1(t) = x_2(t)$$
$$\dot{x}_2(t) = x_3(t)$$
$$\vdots$$
$$\dot{x}_m(t) = -a_0 x_1(t) - a_1 x_2(t) - \cdots a_{m-1} x_m(t) + e(t)$$

$$C = [c'_0 \ c'_1 \ \cdots \ c'_{m-1}]$$

(coefs of num after division)

Which is a state space model with same C, D but

$$A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 \cdots & 0 \\ & & & \ddots & \vdots \\ & & & & 0 \\ & & & & 1 \\ -a_0 & -a_1 & \cdots & & -a_{m-1} \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

coefs of denom of $H(s)$

So, there are actually many different state space models we could use for $H(s)$. in fact, infinitely many

The original model, with diagonal A, is known as the "modal" form, 2nd is "companion" form

Moreover: there is nothing special about $H(s)$. We can also apply the same ideas to $G(s)$.

$$\dot{\underline{x}}(t) = A\,\underline{x}(t) + B\,u(t)$$

$$y(t) = C\,\underline{x}(t) + D\,u(t)$$

State-space model for $G(s)$ itself!

(If confusion arises, we use (A_p, B_p, C_p, D_p) for $G(s)$ with state $\underline{x}_p(t)$, and (A_c, B_c, C_c, D_c) for $H(s)$ with state $\underline{x}_c(t)$

Further connections between Laplace and State Space

Laplace works fine on vectors too; just Laplace each component

$$\mathcal{L}\{\underline{x}(t)\} = \begin{bmatrix} \mathcal{L}\{x_1(t)\} \\ \mathcal{L}\{x_2(t)\} \\ \vdots \\ \mathcal{L}\{x_n(t)\} \end{bmatrix} = \underline{X}(s)$$

Usual linearity rule applies: (A: $n \times n$ constant matrix, $\underline{b}$ const vector)

$$\mathcal{L}\{A\underline{x}(t) + \underline{b}\} = A\underline{X}(s) + \underline{b}$$

Derivative rule is:

$$\underline{x}_0 = \begin{bmatrix} x_1(0) \\ x_2(0) \\ \vdots \\ x_n(0) \end{bmatrix}$$

$$\mathcal{L}\{\dot{\underline{x}}(t)\} = s\underline{X}(s) - \underline{x}_0$$

Apply Laplace to State space model

$$\dot{\underline{x}}(t) = A\underline{x}(t) + B u(t)$$
$$y(t) = C\underline{x}(t) + D u(t)$$

$$\Rightarrow \quad s\underline{x}(s) - \underline{x}_0 = A\underline{x}(s) + B u(s)$$
$$y(s) = C\underline{x}(s) + D u(s)$$

$1^{\underline{st}}$ eq'n is equivalent to:

$$(sI - A)\underline{x}(s) = \underline{x}_0 + B u(s) \qquad (I = n \times n \text{ identity})$$

$$\Rightarrow \quad \underline{x}(s) = [sI - A]^{-1}[\underline{x}_0 + B u(s)]$$

Substitute into 2^{nd} eq'n:

$$\boxed{y(s) = C[sI-A]^{-1}\underline{x}_0 + [C(sI-A)^{-1}B + D]\,U(s)}$$

$$y(s) = C[sI-A]^{-1}\underline{x}_0 + [C(sI-A)^{-1}B+D]U(s)$$

Recall: TF derived assuming ICs = 0 $\Rightarrow$ $\underline{x}_0 = \underline{0}$

Then

$$y(s) = \overbrace{\boxed{[C(sI-A)^{-1}B+D]}}^{G(s)}U(s)$$

Hence, for any (A,B,C,D) state space representation the corresponding transfer function is:

$$G(s) = [C\overbrace{(sI-A)^{-1}}^{n\times n \text{ matrix inverse}}B+D]$$

Now recall for arbitrary matrix $\underline{M}$

$$\underline{M}^{-1} = \frac{Adj(\underline{M})}{Det(\underline{M})}$$

Adj = $n\times n$ Matrix of cofactors

Det = <u>Scalar</u> Determinant

Thus

$$(sI - A)^{-1} = \frac{Q(s)}{r(s)}$$

where

$$\begin{cases} Q(s) = \mathrm{Adj}(sI - A) & (n \times n \text{ matrix.}) \\ r(s) = \mathrm{Det}(sI - A) & \underline{\text{polynomial}} \text{ in } s. \end{cases}$$

and

$$G(s) = \frac{CQ(s)B}{r(s)} + D = \frac{CQ(s)B + Dr(s)}{r(s)}$$

where both $CQ(s)B$ and $r(s)$ are polynomials

$\Rightarrow$ zeros where $CQ(s)B + Dr(s) = 0$

$\Rightarrow$ poles where $r(s) = 0$.

So the poles of $G(s)$ will satisfy

$$r(s) = \emptyset = \mathrm{Det}(s\mathbb{I} - A)$$

at these values of s

$\Rightarrow (s\mathbb{I} - A)$ is <u>singular</u>, i.e. there exists nonzero $\underline{v}$

so that

$$(s\mathbb{I} - A)\underline{v} = \underline{0}$$

or:

$$A\underline{v} = s\underline{v} \quad \text{for any } s \text{ with } r(s) = 0$$

$\Rightarrow$ poles of $G(s)$ are $\boxed{\underline{\text{eigenvalues}}}$ of A!!

In fact, any A matrices with same eigenvalues have same $r(s)$ polynomial, and thus give rise to $G(s)$ with same denominator

$\Rightarrow$ for each such A: B, C can be chosen to preserve the numerator of $G(s)$, hence its zeros.

Thus, there are in fact infinitely many (A, B, C, D) combinations that give rise to the same $G(s)$.

All such matrices are related by a similarity transform

$$A' = P^{-1} A P$$

for any nonsingular matrix P

Now, suppose we have state space models of both plant + comp

$$\dot{\underline{x}}_p = A_p \underline{x}_p + B_p u$$
$$y = C_p \underline{x}_p$$

$$\dot{\underline{x}}_c = A_c \underline{x}_c + B_c e \qquad e = y_d - y$$
$$u = C_c \underline{x}_c + D_c e \qquad\quad = y_d - C_p \underline{x}_p$$

Substituting for e:

$$\dot{\underline{x}}_c = A_c \underline{x}_c + B_c y_d - B_c C_p \underline{x}_p$$

$$u = C_c \underline{x}_c + D_c y_d - D_c C_p \underline{x}_p$$

Substitute u into plant:

$$\dot{\underline{x}}_p = A_p \underline{x}_p - B_p D_c C_p \underline{x}_p + B_p C_c \underline{x}_c + B_p D_c y_d$$

Collect both sets of equations together:

$$\frac{d}{dt}\begin{bmatrix} \underline{x}_p \\ \underline{x}_c \end{bmatrix} = \underbrace{\left[\begin{array}{c:c} A_p - B_p D_c C_p & B_p C_c \\ \hdashline -B_c C_p & A_c \end{array}\right]}_{A_{CL}} \begin{bmatrix} \underline{x}_p \\ \underline{x}_c \end{bmatrix} + \underbrace{\begin{bmatrix} B_p D_c \\ B_c \end{bmatrix}}_{B_{CL}} y_d$$

$$y = \underbrace{\begin{bmatrix} C_p & 0 \end{bmatrix}}_{C_{CL}} \begin{bmatrix} \underline{x}_p \\ \underline{x}_c \end{bmatrix}$$

where $(A_{CL}, B_{CL}, C_{CL}, \emptyset)$ is a state space model for the closed-loop dynamics $\Rightarrow T(s)$

$\Rightarrow$ CL poles are <u>eigenvalues</u> of A_{CL}!

Problem in numerical linear algebra:

Find (A_c, B_c, C_c, D_c) so Eigenvalues of A_{CL} (poles of $T(s)$) have specified "nice" values.

⇒ Several algorithms exist to solve this problem

⇒ "Modern" control ⇒ requires computer to solve

⇒ Can make the base task of getting stable $T(s)$ easier, but still have to worry about

- tracking + dist. rejection
- robustness
- Delay/Discretization effects
- Control Saturation

"Tuning" to also address these is Not nearly so easy!

- There is NO "one true way" to design a controller
- All methods have their advantages, and their weaknesses
- More advanced analysis gives additional perspective + insight, but not "magic bullets"
- It *always* comes down to an engineer's judgment to weigh all the tradeoffs and settle on a final design.