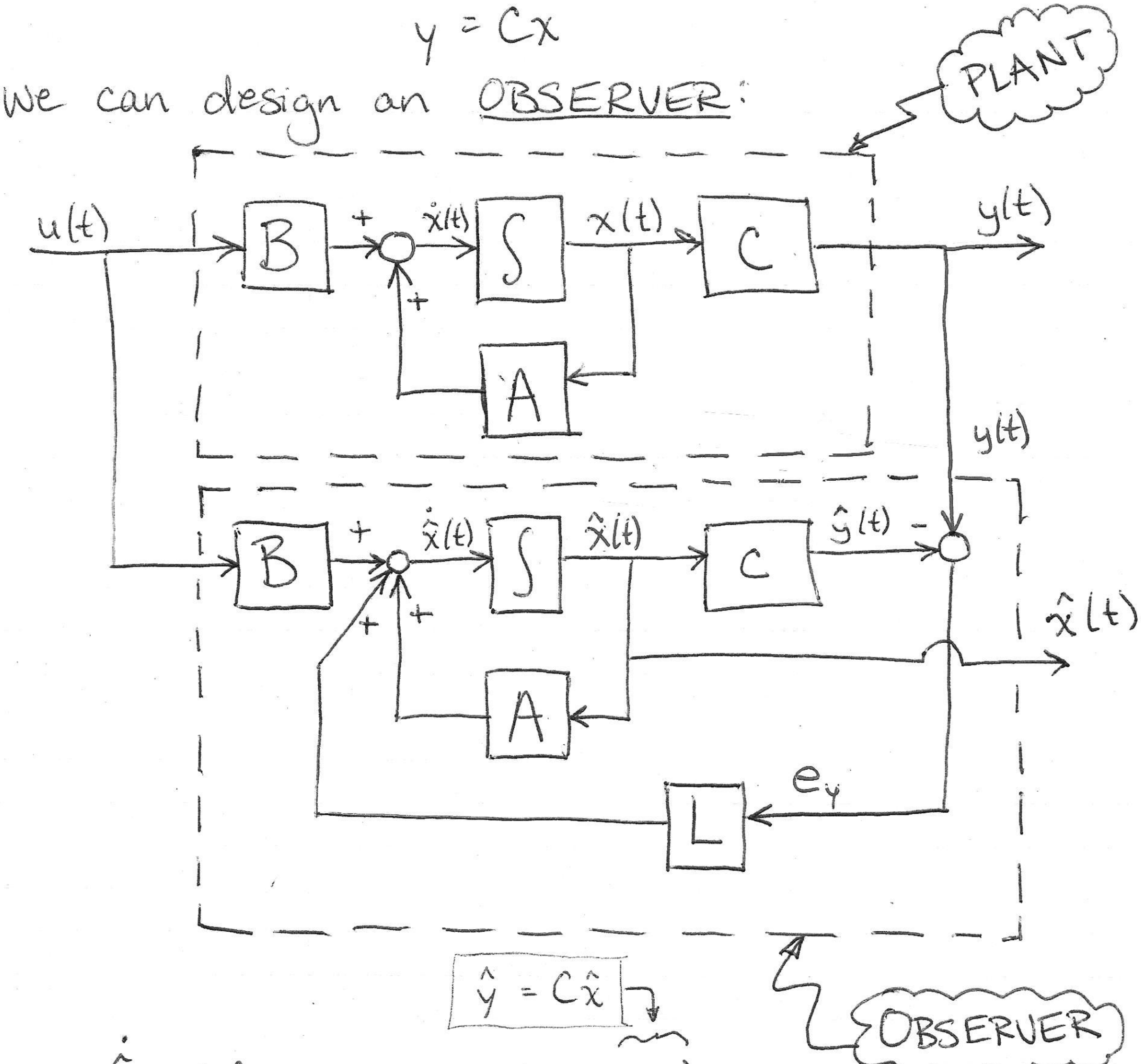
Estimators + Kalman Filters

Recall, for a system with no noise:

$$\dot{x} = Ax + Bu$$
$$y = Cx$$

We can design an OBSERVER:



$$\boxed{\hat{y} = C\hat{x}}$$

$$\underbrace{\dot{\hat{x}}}_{\text{estimate of } x(t)} = \underbrace{A\hat{x} + Bu}_{\text{model of PLANT dynamics}} + \underbrace{L(y - C\hat{x})}_{\text{feedback term}}$$

feedback term: Observer gain, L, times the output error: $\boxed{e_y = y - \hat{y}}$

{e_y sometimes written as "$\tilde{y}$".}

We saw that the error in the estimate:

e_x sometimes written as "$\tilde{x}$"

$$\boxed{e_x = x - \hat{x}}$$

has dynamics described by:

$$\boxed{\dot{e}_x = (A - LC)\, e_x}$$

So, selection of the observer gains, L, sets the poles of the observer dynamics, described by $A_o = A - LC$ ....which we want to be asymptotically stable and notably "faster" than the dynamics of the actually plant: $\dot{x} = (A - BK)\,x$ $\left\{ \begin{array}{l} \text{closed-loop} \\ \text{dynamics} \end{array} \boxed{u = -Kx} \right\}$

We can set the poles of $(A - LC)$ to arbitrary desired locations IF AND ONLY IF (C, A) is OBSERVABLE.

* note $(A - LC)$ & $(A - LC)^T$ have the same poles... Therefore, we can exploit "duality" to solve:

"Duality":

Controller	Observer
$A - BK$	$(A - LC)^T = A^T - C^T L^T$

Form puts "free matrix" to solve for (e.g. in MATLAB), to the RIGHT! In this form, we can treat the pole placement problems just the same for each.

Kalman filter:

> Acts like a "low-pass filter", with good noise rejection...

Real world systems include:

- modeling inaccuracies
- disturbances (to the plant dynamics)
- noise (in the measurement outputs)

A popular type of observer for reconstructing the state in noisy situations is a Kalman filter.

Stochastic dynamic system is given by:

$$\dot{x} = Ax + Bu + Gw \quad \leftarrow \text{process noise}$$

$$y = Cx + v \quad \leftarrow \text{measurement noise}$$

Assumptions:

"white noise"

$w(t)$ and $v(t)$ have zero mean (no bias) and are white noise processes:

The cross-correlation matrix for each is:

$$R_w(\tau) = E\{w(t+\tau)w^T(t)\} = Q\,\delta(\tau)$$
$$R_v(\tau) = E\{v(t+\tau)v^T(t)\} = R\,\delta(\tau)$$

$w(t) \quad (0, Q) \quad Q \geq 0 \leftarrow$ Can have zero process noise

$v(t) \quad (0, R) \quad R > 0 \leftarrow$ Always assume non-zero measurement noise

(No Kalman solution exists if $R=0$...)

HOWEVER:

There are "tricks" to augment the true system if $w(t)$ [and/or $v(t)$] is NOT white...

Say $w(t)$ is <u>NOT</u> white, then we can define:

"noise-shaping filter"
$$\left\{\begin{aligned} \dot{x}_w &= A_w x_w + B_w n \\ w &= C_w x_w + D_w n \end{aligned}\right.$$
where "n" <u>is</u> white noise

Now, augment the <u>true</u> state, x, w/ new state, x_w. (This technique uses a noise-shaping filter.)

$$\begin{bmatrix} \dot{x} \\ \dot{x}_w \end{bmatrix} = \begin{bmatrix} A & GC_w \\ 0 & A_w \end{bmatrix} \begin{bmatrix} x \\ x_w \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u + \begin{bmatrix} CD_w \\ B_w \end{bmatrix} n$$

$$y = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} x \\ x_w \end{bmatrix} + v$$

white noise

(Back to Kalman filtering...)

We now want to select <u>OBSERVER</u> gains. <u>As before</u>, our estimated <u>state is</u>:

$\boxed{\hat{y} = C\hat{x}}$

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - \hat{y})$$
$$= (A - LC)\hat{x} + Bu + Ly$$
$$\hat{y} = E\{Cx + v\} = C\hat{x}$$

estimation error
$$\left[\begin{aligned} e_x &= x - \hat{x} \\ &= (A - LC)e_x + Gw - Lv \\ &= A_o e_x + Gw - Lv \end{aligned}\right.$$

$\boxed{A_o \equiv A - LC}$

output error
$$\left[\begin{aligned} e_y &= y - \hat{y} \\ &= Cx - C\hat{x} \\ &= Ce_x \end{aligned}\right.$$

Now, the ERROR COVARIANCE is:

$$\boxed{P(t) = E\{e_x e_x^T\}}$$ ← time-varying

↑ measures uncertainty of our estimate

$P(t)$ small → estimate is good.

---

* The <u>optimal gain</u>, L, minimizing the steady-state error covariance, P.

(∴ The optimal gain L is a constant matrix of observer gains.)

(Sorry to skip a lot of math here...)

Error covariance for a particular L goes as:

Recall, R is σ for measurement noise, v

$$\dot{P} = A_o P + P A_o^T + L R L^T + G Q G^T$$

@ steady state, $\dot{P} = 0$ ($P(t) \to$ constant...)

$$0 = A_o P + P A_o^T + L R L^T + G Q G^T$$

It turns out, the optimal L is:

$$\boxed{L = P C^T R^{-1}}$$

plug in for L...

- Plugging in for "L":

$$0 = \underbrace{(A - PC^T R^{-1} C)}_{A_o \equiv A - LC} P + P \underbrace{(A - PC^T R^{-1} C)}_{A_o \equiv A - LC}{}^T + PC^T R^{-1} C P + G Q G^T$$

$$\boxed{0 = AP + PA^T + GQG^T - PC^T R^{-1} CP}$$

So, first solve for P (eqn above), <u>then</u> solve for L:

"Predictor" & "Corrector"

$$(1)\quad \hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1}$$

$$(2)\quad P_k^- = AP_{k-1}A^T + Q$$

---

or "K_k"

$$(3)\quad L_k = P_k^- C^T(CP_k^- C^T + R)^{-1}$$

$$(4)\quad \hat{x}_k = \hat{x}_k^- + L_k(y - C\hat{x}_k^-)$$

$$(5)\quad P_k = P_k^- - L_k C P_k^-$$

$$= (I - L_k C)P_k^-$$

$$\left.\begin{array}{l} \text{Usually, } L_k \to K_k \\ \qquad C \to H \\ \qquad y \to z \\ \text{in this iterative form...} \end{array}\right\}$$

Notation varies in the literature...