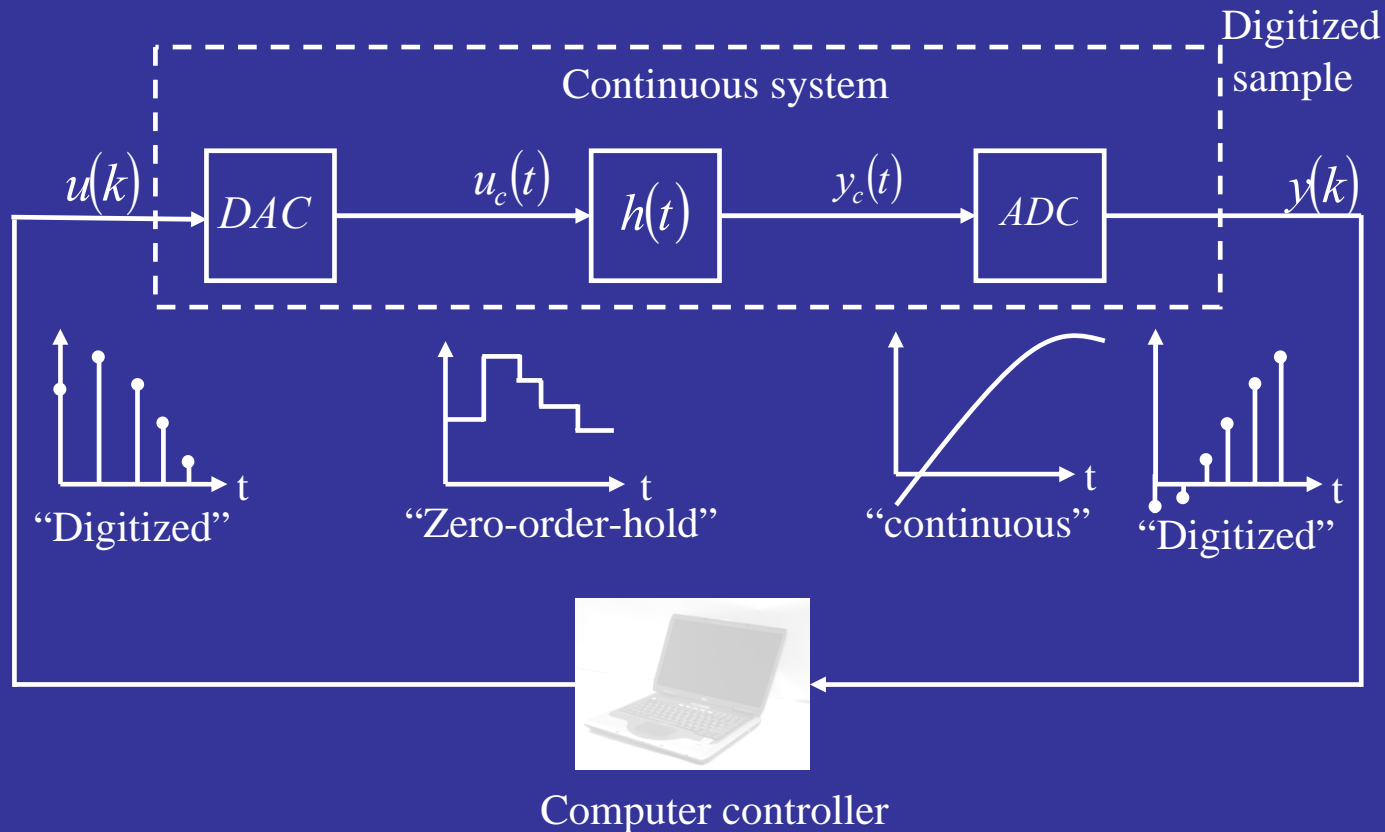


4. Discrete Systems

Where do discrete systems arise?

Typical control engineering example:



Assume the DAC+ADC are clocked at sampling period T .

4. Discrete Systems (Cnt'd)

Then $u(t)$ is given by:

$$u(k) \equiv u_c(t); kT \leq t < (k+1)T$$

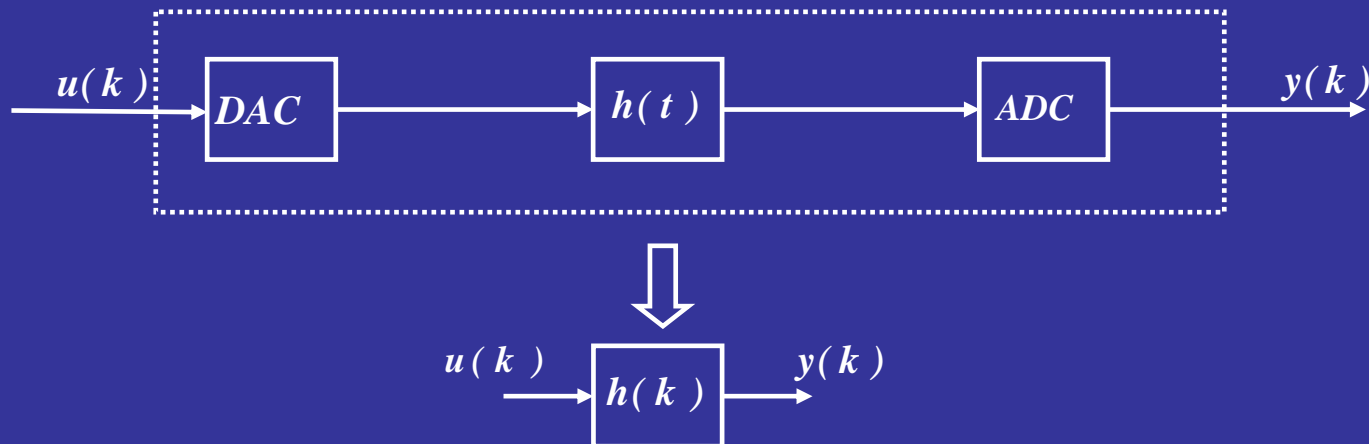
$$y(k) \equiv y_c(kT); k = 0, 1, 2, \dots$$

Suppose: time continuous system is given by state-Space

$$\dot{x}_c(t) = A x_c(t) + B u_c(t); x_c(0) = x_0$$

$$y_c(t) = C x_c(t) + D u_c(t)$$

Can we obtain direct relationship between $u(k)$ and $y(k)$? i.e. want Equivalent discrete system:



4. Discrete Systems

Yes! We can obtain equivalent discrete system.

Recall
$$x_c(t) = e^{At} x_c(0) + \int_0^t e^{A\tau} \cdot B u_c(t - \tau) d\tau$$

From this
$$x_c(kT + T) = e^{AT} x_c(kT) + \int_0^T e^{A\tau} \cdot B u_c(kT - \tau) d\tau$$

Observe that $u(kT + T - \tau) = u(kT)$ for $\tau \in [0, T]$
i.e. $u(kT + T - \tau)$ is constant $u(kT)$ over $\tau \in [0, T]$

i.e. can pull out of integral.

$$\implies x_c(kT + T) = e^{AT} x_c(kT) + \left(\int_0^T e^{A\tau} \cdot B d\tau \right) u_c(kT)$$

$$x(k+1) = A_d x(k) + B_d u(k)$$

$$y(k) = C_d x(k) + D_d u(k)$$

$$x(0) = x_c(0)$$

$$\text{So } A_d = e^{AT}, B_d = \int_0^T e^{A\tau} \cdot B d\tau \quad C = C_d, D_d = D$$

So we have an exact (note: $x(k+1) = x(k) + \dot{x}(k) T + O(\cdot)$) discrete time equivalent to the time Continuous system at sample times $t=kT$ - no numerical approximation!



4.1 Linear Ordinary Difference Equation

A linear ordinary difference equation looks similar to a LODE

$$y(k+n) + a_{n-1}y(k+n-1) + \dots + a_1y(k+1) + a_0y(k) = b_m u(k+m) + \dots + b_1 u(k+1) + b_0 u(k)$$

$n \geq m$; Assumes initial values $y(n-1), \dots, y(1), y(0) = 0$.

Z-Transform of the LODE yields (linearity of Z-Transform):

$$z^n Y(z) + z^{n-1}a_{n-1} Y(z) + \dots + za_1 Y(z) + a_0 Y(z) = z^m b_m U(z) + \dots + zb_1 U(z) + b_0 U(z)$$

It follows the input-output relation:

$$(z^n + z^{n-1}a_{n-1} + \dots + za_1 + a_0) Y(z) = (z^m b_m + \dots + zb_1 + b_0) U(z)$$

$$Y(z) = \frac{z^m b_m + \dots + zb_1 + b_0}{z^n + \dots + za_1 + a_0} U(z)$$

$$Y(z) = G(z) U(z)$$

Once again: if $U(z) = 1$, ($u(k) = \delta(k)$), then $Y(z) = G(z)$.

Transfer Function of system is the Z-Transform of its pulse response!



4.1 Z-Transform of Discrete State Space Equation

$$\begin{aligned}x(k+1) &= A_d x(k) + B_d u(k) \\ y(k) &= C_d x(k) + D_d u(k)\end{aligned}$$

Applying Z-Transform on first equation:

$$\begin{aligned}z \cdot X(z) - z x(0) &= A_d X(z) + B_d U(z) \\ (zI - A_d) X(z) &= z x(0) + B_d U(z)\end{aligned}$$

$$X(z) = (zI - A_d)^{-1} z x(0) + (zI - A_d)^{-1} B_d U(z)$$

Homogeneous solution Particular solution

NOW:

$$\begin{aligned}Y(z) &= C_d X(z) + D_d U(z) \\ &= C_d (zI - A_d)^{-1} z x(0) + (C_d (zI - A_d)^{-1} B_d + D_d) U(z)\end{aligned}$$

If $x(0)=0$ then we get the input-output relation:

$$\begin{aligned}Y(z) &= G(z) U(z) \text{ with} \\ G(z) &= C_d (zI - A_d)^{-1} B_d + D_d\end{aligned}$$

Exactly like for the continuous systems!!!!!!



4.2 Frequency Domain/Z-Transform

For analyzing discrete-time systems:

Z-Transform

(analogue to Laplace Transform for time-continuous system)

It converts linear ordinary difference equation into algebraic equations: easier to find a solution of the system!

It gives the frequency response for free!

Z-Transform ==generalized discrete-time Fourier Transform

Given any sequence $f(k)$ the discrete Fourier transform is

$$\tilde{F}(\omega) = \sum_{k=-\infty}^{\infty} f(k) e^{-i\omega k}$$

with $\omega = 2\pi f$, $f = \frac{1}{T}$ The sampling frequency in Hz,

T difference / Time between two samples.

In the same spirit: $F(z) = Z[f(k)] = \sum_{k=0}^{\infty} f(k) z^{-k}$.

With z a complex variable

Note:

if $f(k) = 0$ for $k = -1, -2, \dots$ then $\tilde{F}(\omega) = F(z = e^{i\omega})$.



4.3 stability (z-domain)

A discrete LTI system is BIBO stable if

$$|u(k)| < M; \forall k = |y(k)| < K; \forall k$$

Condition for BIBO stability:

$$|y(k)| = \left| \sum_0^k u(k-i) h(i) \right| \leq \sum_0^k |u(k-i)| |h(i)| \leq M \sum_0^k |h(i)| \leq M \sum_0^{\infty} |h(i)|$$

$$\therefore \sum_0^{\infty} |h(i)| < \infty \quad \rightarrow \text{BIBO stable.}$$

For L.O.D.E State space system:

$$H(z) = \alpha \cdot \frac{\prod_{i=1}^k (z - z_i)}{\prod_{i=1}^n (z - p_i)} = \sum_{i=1}^k \beta_i T_i(z)$$

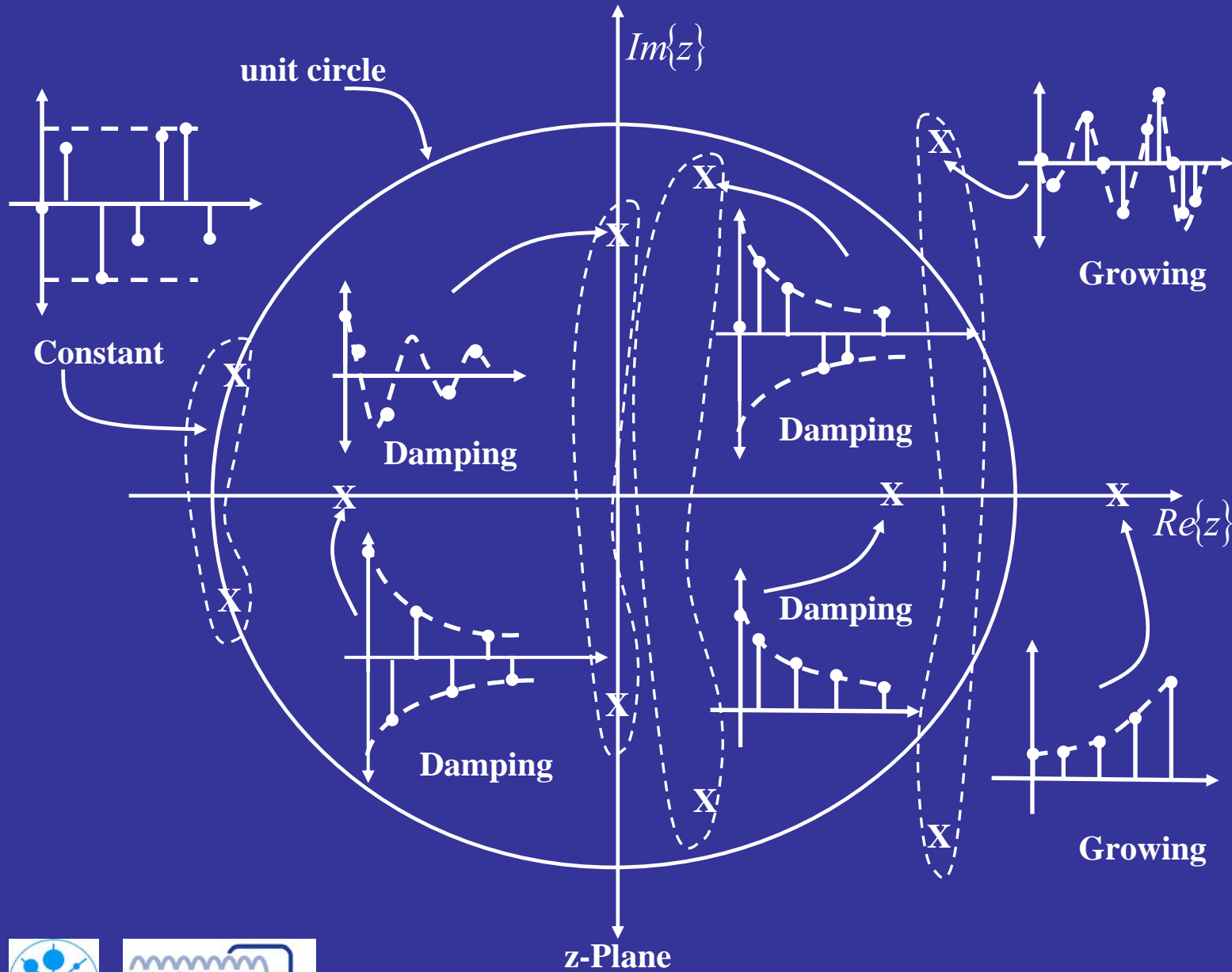
With partial fraction of the rational function:

Once again pole locations tell a lot about shape of pulse response.

Zeros determine the size of β_i



4.3 Stability (z- domain)



4.3 Stability (Z- domain)

In General

Complex pair → oscillatory growth / damping

Real pole → exponential growth / decay but maybe oscillatory too (e.g: $r^n 1(n)$ where $r < 0$)

The farther inside unit circle poles are

→ The faster the damping → the higher stability

i.e $|p_i| \leq 1$ → system stable



4.3 Stability (z-domain)

Stability directly from State Space:

Exactly as for cts systems, assuming no pole-zero cancellations and $D=0$

$$\begin{aligned} H(z) &= \frac{b(z)}{a(z)} = C_d d (zI - A_d)^{-1} B_d \\ &= \frac{C_d (zI - A_d)_{adj} B_d}{\det(zI - A_d)} \end{aligned}$$

$$\begin{aligned} b(z) &= C_d \text{adj}(zI - A_d) B_d \\ a(z) &= \det(zI - A_d) \end{aligned}$$

→ Poles are eigenvalues of A_d

So check stability, use eigenvalue solver to get e-values of the matrix A_d , then

If $|\lambda_i| < 1$ for all i → system stable

Where λ_i is the i th e-value of A_d .



4.4 Discrete Cavity Model

Converting the transfer function from the continuous cavity model to the discrete model:

$$H(s) = \frac{\omega_{12}}{\Delta\omega^2 + (s + \omega_{12})^2} \begin{bmatrix} s + \omega_{12} & \Delta\omega \\ \Delta\omega & s + \omega_{12} \end{bmatrix}$$

The discretization of the model is represented by the z-transform:

$$H(z) = \left(1 - \frac{1}{z}\right) Z\left(\frac{H(s)}{s}\right) = \frac{z-1}{z} \cdot Z\left\{L^{-1}\left\{\frac{H(s)}{s}\right\}\right\}_{t=kT_s}$$

$$H(z) = \frac{\omega_{12}}{\Delta\omega^2 + \omega_{12}^2} \cdot \begin{bmatrix} \omega_{12} & -\Delta\omega \\ \Delta\omega & \omega_{12} \end{bmatrix} - \left(\frac{\omega_{12}}{\Delta\omega^2 + \omega_{12}^2} \cdot \frac{z-1}{z^2 - 2ze^{\omega_{12}T_s} \cdot \cos(\Delta\omega T_s) + e^{2\omega_{12}T_s}} \right) \cdot \left\{ \left(z - e^{\omega_{12}T_s} \cdot \cos(\Delta\omega T_s) \right) \cdot \begin{bmatrix} \omega_{12} & -\Delta\omega \\ \Delta\omega & \omega_{12} \end{bmatrix} - e^{\omega_{12}T_s} \cdot \sin(\Delta\omega T_s) \cdot \begin{bmatrix} \Delta\omega & \omega_{12} \\ -\omega_{12} & \Delta\omega \end{bmatrix} \right\}$$

4.5 Linear Quadratic Regulator

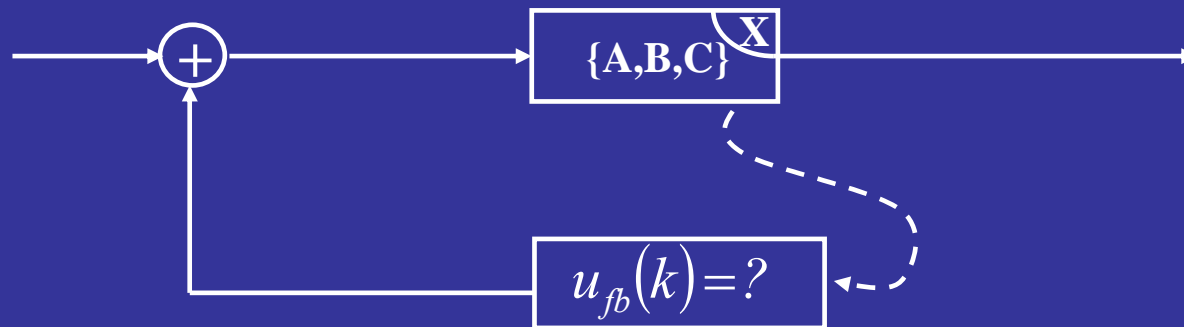
Given:

$$x(k+1) = Ax(k) + Bu(k)$$
$$z(k) = Cx(k)$$

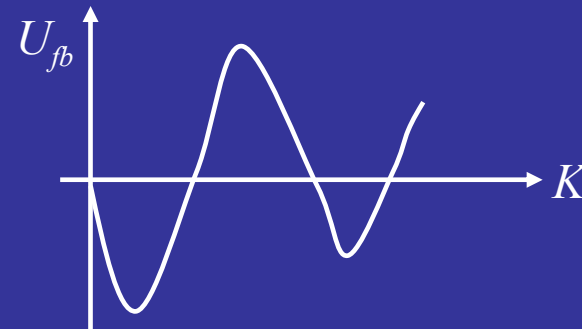
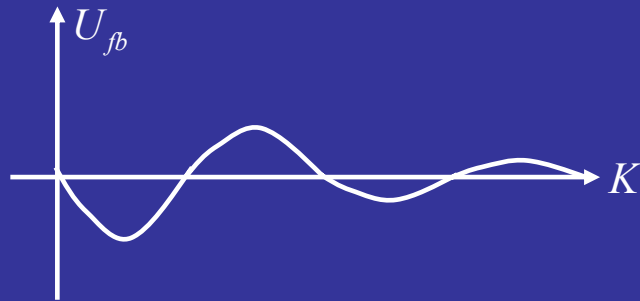
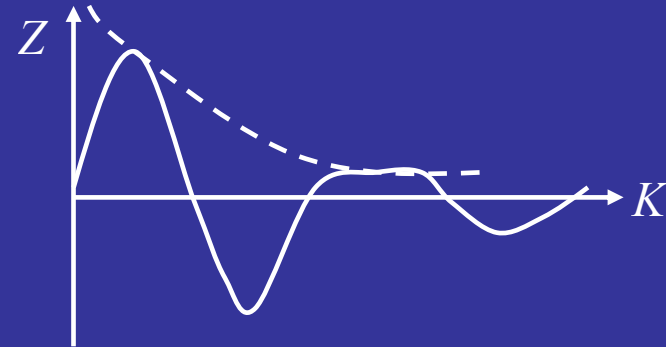
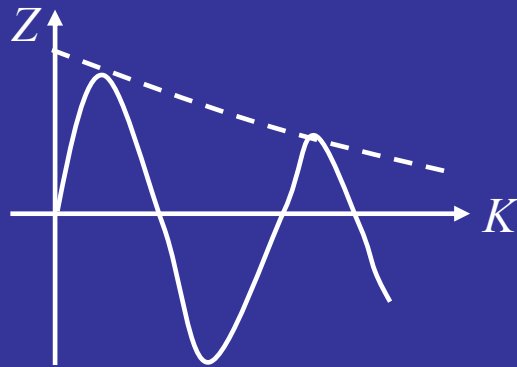
(Assume $D=0$ for simplicity)

Suppose the system is unstable or almost unstable. We want to find $u_{fb}(k)$ which will bring $x(k)$ to Zero, quickly, from any Initial condition.

i.e.



4.5 Trade Off



(1) "Bad" damping
→ **Large** Output excursions

(1) "Good" damping
→ **Small** Output excursions

(2) But "Cheap" control i.e. u_{fb} **Small**

(2) But "expensive" control i.e. u_{fb} **large.**

4.5 Quadratic Forms

A quadratic form is a quadratic function of the components of a vector:

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in R^2$$

$$\begin{aligned} f(x) &= f(x_1, x_2) \\ &= ax_1^2 + bx_1x_2 + cx_1 + dx_2^2 \\ &= [x_1 \ x_2] \underbrace{\begin{bmatrix} a & \frac{1}{2}b \\ \frac{1}{2}b & d \end{bmatrix}}_Q \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \underbrace{[c \ 0]}_{P^T} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \end{aligned}$$

$$f(x) = \underbrace{x^T Q x}_{\text{Quadratic Part}} + \underbrace{P^T x}_{\text{Linear Part}} + \underbrace{e}_{\text{Constant}}$$



4.5 Quadratic Cost for Regulator

What do we mean by “bad“ damping and “cheap“ control? We now define precisely What we mean. Consider:

$$J \equiv \sum_{i=0}^{\infty} \{x_i^T Q x_i + u_i^T R u_i\}$$

The first term penalizes large state excursions, the second penalizes large control.

$$Q \geq 0, R > 0$$

Can tradeoff between state excursions and control by varying Q and R.

Large Q → “good“ damping important

Large R → actuator effort “expensive“



4.5 LQR Problem Statement

(Linear quadratic regulator)

Given: $x_{i+1} = Ax_i + Bu_i$; x_0 given:

Find control sequence $\{u_0, u_1, u_2, \dots\}$ such that

$$J = \sum_{i=0}^{\infty} x_i^T \{ Q x_i + u_i^T R u_i \}$$

Answer: = minimum

The optimal control sequence is a state feedback sequence $\{u_i\}_0^{\infty}$

$$u_i = -K_{opt} x_i$$

$$K_{opt} = (R + B^T S B)^{-1} B^T S A$$

$$S = A^T S A + Q - A^T A B (R + B^T S B)^{-1} B^T S A$$

Algebraic Riccati Equation (A.R.E) for discrete-time systems.

Note: Since u_i = state feedback, it works for any initial state x_0



4.5 LQR Problem Statement

Remarks:

- (1) So optimal control, $u_i = -K_{opt} x_i$ is state feedback! This is why we are interested in state feedback.
- (2) Equation A.R.E. is matrix quadratic equation. Looks pretty intimidating but Computer can solve in a second.
- (3) No tweaking ! Just specify {A,B,C,D} and Q and R, press return button, LQR Routine Spits out K_{opt} - Done
- (Of course picking Q and R is tricky sometimes but that's another story).
- (4) Design is guaranteed optimal in the sense that it minimizes.

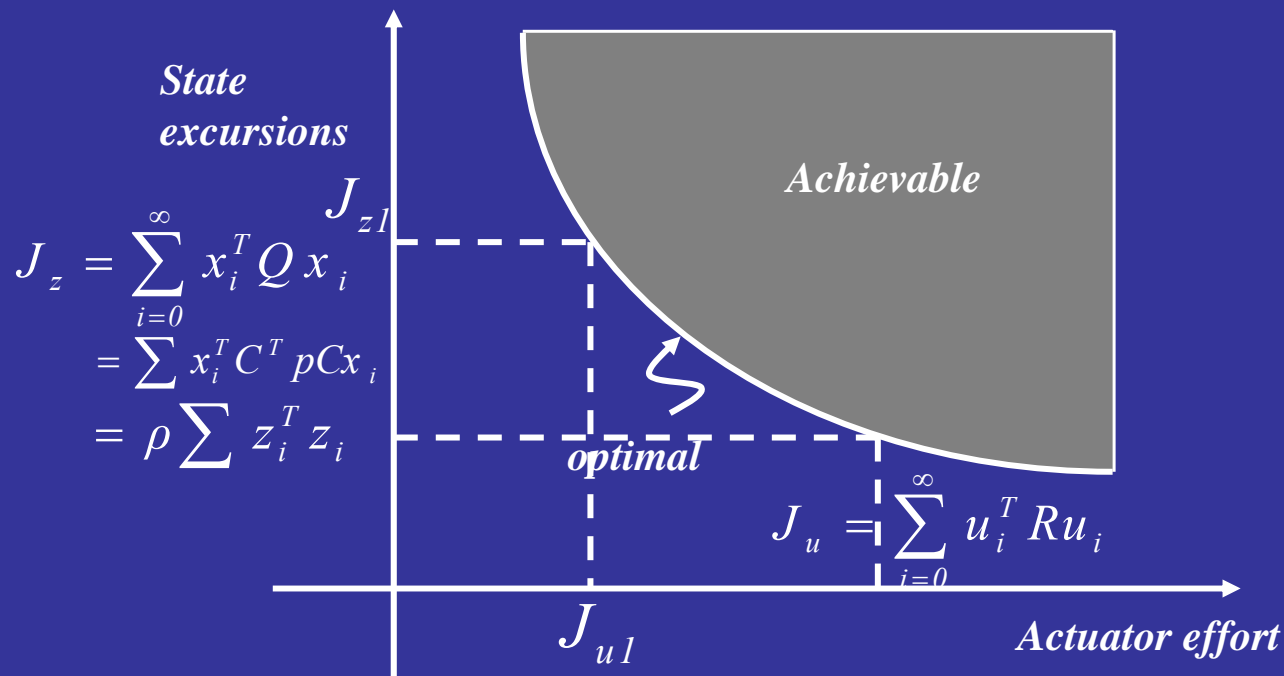
$$J_{lqr}(x_0, \{u_i\}_0^\infty) = \sum_{i=0}^{\infty} \{x_i^T Q x_i + u_i^T R u_i\}$$

(Of course that doesn't mean its "best" in the absolute sense .-)



4.5 LQR Problem Statement - Remarks

- (5) As vary Q/R Ratio we get whole family of K_{lqr} 's, i.e. can Trade-off between state excursion (Damping) Vs actuator effort (Control)



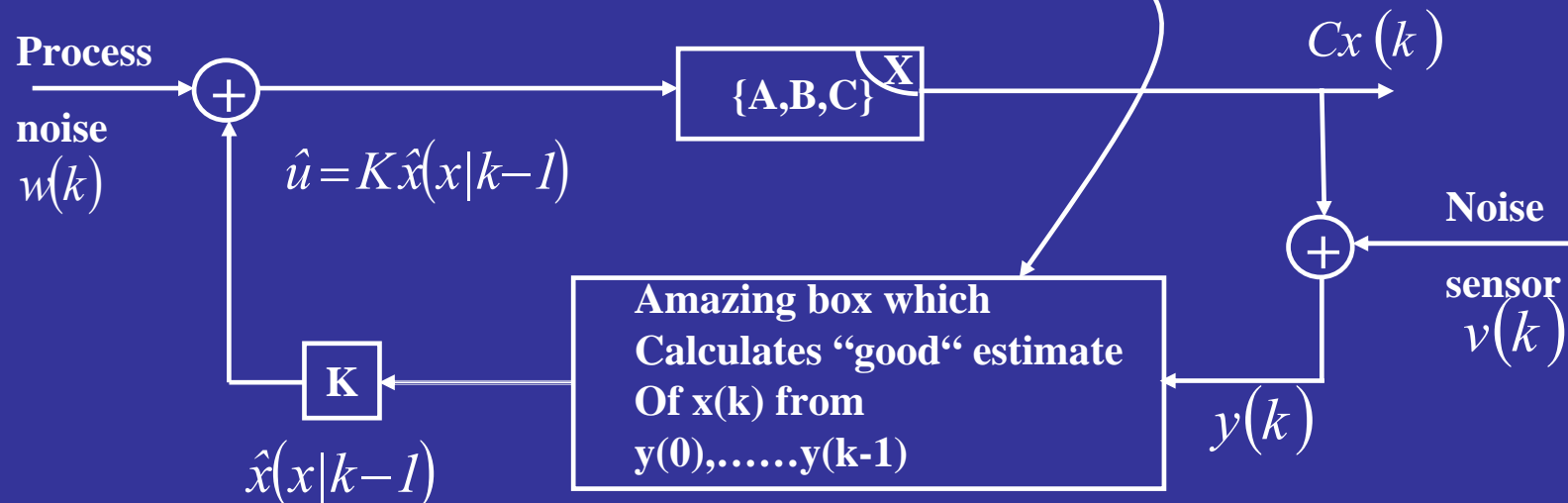
4.6 Optimal Linear Estimation

Our optimal control has the form $u_{opt}(k) = -K(k)x_{opt}(k)$

This assumes that we have complete state Information $x_{opt}(k)$ -not actually true!.
e.g: in SHO, we might have only a Position sensor but Not a velocity sensor.

How can we obtain “good“ estimates of the velocity state from just observing the position state?

Furthermore the Sensors may be noisy and the plant itself maybe subject to outside disturbances (process noise) i.e. we are looking for **this:**

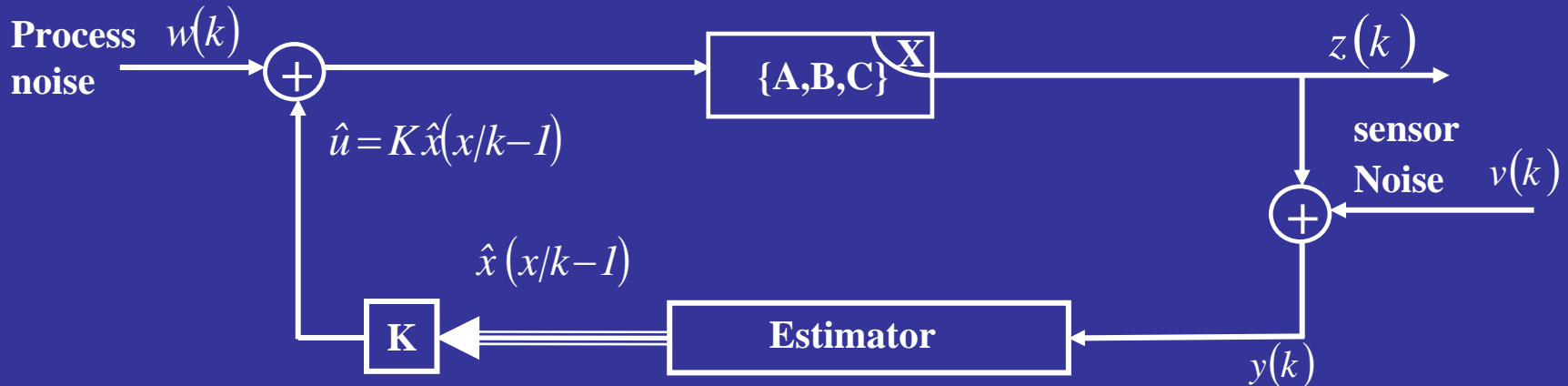


4.6 Problem Statement :

$$x(k+1) = A x(k) + B w(k)$$

$$z(k) = C x(k)$$

$$y(k) = C x(k) + v(k)$$



Assume also $x(0)$ is Random & Gaussian and that $x(k), w(k), v(k)$ are all mutually Independent for all k .

Find : $\hat{x}(k/k-1)$ Optimal estimate of $x(k)$ given y_0, \dots, y_{k-1}

Such that “mean squared error“

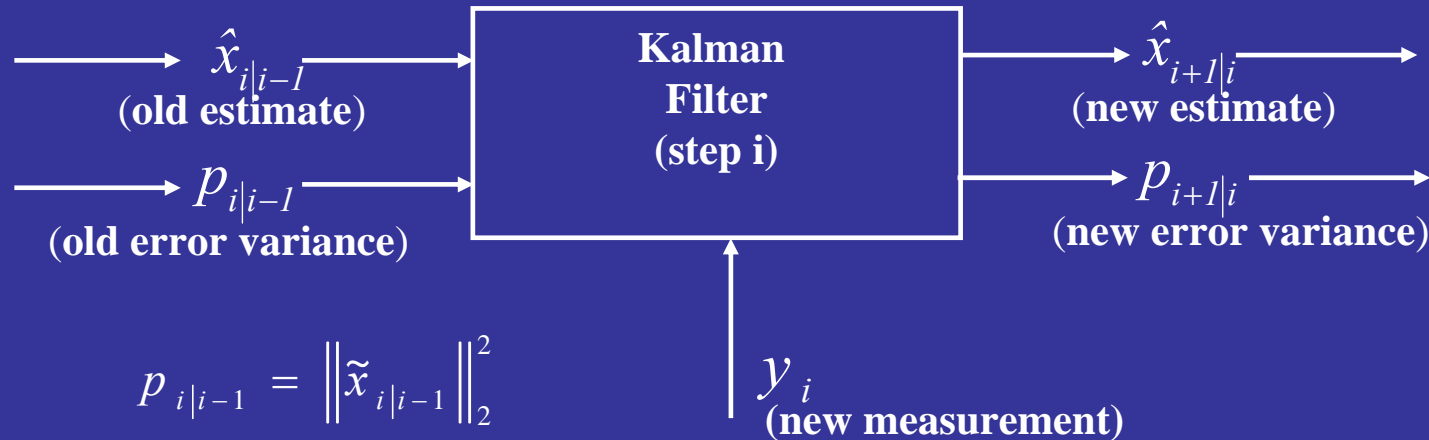
$$E \left[\left\| x(k) - \hat{x}(k/k-1) \right\|_2^2 \right] = \text{minimal}$$

Fact from statistics: $\hat{x}(k|k-1) = E[x(k) | (y_0, \dots, y_{k-1})]$



4.6 Kalman Filter

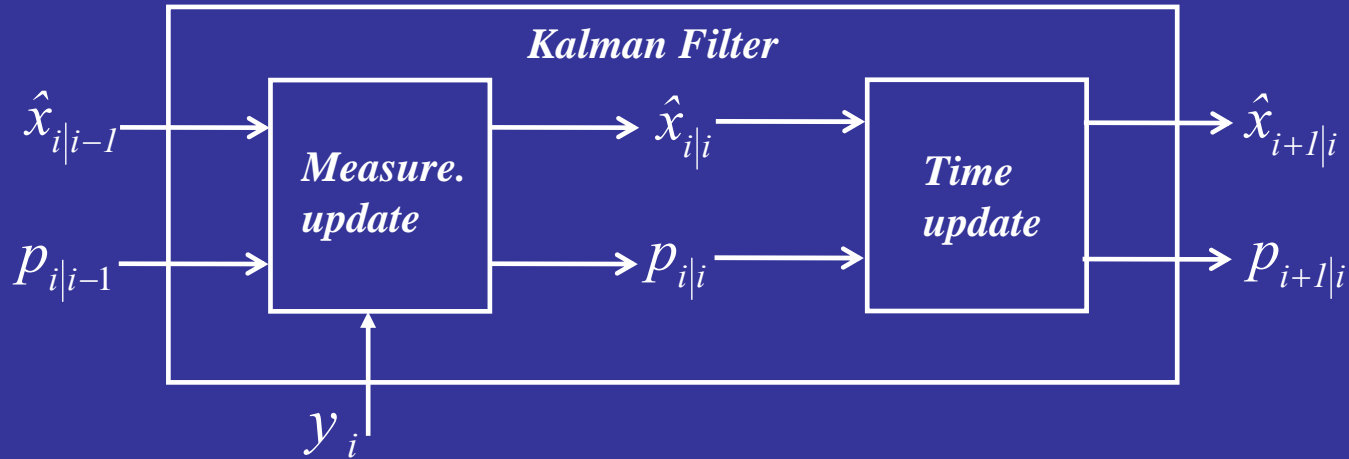
The Kalman filter is an efficient algorithm that computes the new $\hat{x}_{i+1|i}$ (the linear-least-mean (square estimate) of the system state vector x_{i+1} , given $\{y_0, \dots, y_i\}$, by updating the old estimate $\hat{x}_{i|i-1}$ and old $\tilde{x}_{i|i-1}$ (error).



The Kalman Filter produces $\hat{x}_{i+1|i}$ from $\hat{x}_{i|i-1}$ (rather than $\hat{x}_{i|i}$), because it “tracks” the system “dynamics”. By the time we compute $\hat{x}_{i|i}$ from $\hat{x}_{i|i-1}$, the system state has changed from x_i to $x_{i+1} = Ax_i + Bw_i$

4.6 Kalman Filter

The Kalman Filter algorithm can be divided in a measurement update and a time update:



Measurement update (M.U.):

$$\hat{x}_{i|i} = \hat{x}_{i|i-1} + p_{i|i-1} C^T (C p_{i|i-1} C^T + V)^{-1} (y_i - C \hat{x}_{i|i-1})$$

$$p_{i|i} = p_{i|i-1} - p_{i|i-1} C^T (C p_{i|i-1} C^T + V)^{-1} C p_{i|i-1}$$

Time Update (T.U.):

$$\hat{x}_{i+1|i} = A \hat{x}_{i|i}$$

$$p_{i+1|i} = A p_{i|i} A^T + B W B^T$$

With initial conditions:

$$\hat{x}_{0|-1} = 0$$

$$\hat{p}_{0|-1} = X_0$$



4.6 Kalman Filter

By plugging M.U. equations into T.U. equations. One can do both steps at once:

$$\begin{aligned}\hat{x}_{i+1|i} &= A\hat{x}_{i|i} \\ &= A\hat{x}_{i|i-1} + Ap_{i|i-1}C^T(Cp_{i|i-1}C^T + v)^{-1}(y_i - C\hat{x}_{i|i-1})\end{aligned}$$

$$\begin{aligned}\hat{x}_{i+1|i} &= A\hat{x}_{i|i-1} + L_i(y_i - C\hat{x}_{i|i-1}) \\ \text{where } L_i &\equiv A(p_{i|i-1}C^T(Cp_{i|i-1}C^T + V)^{-1})\end{aligned}$$

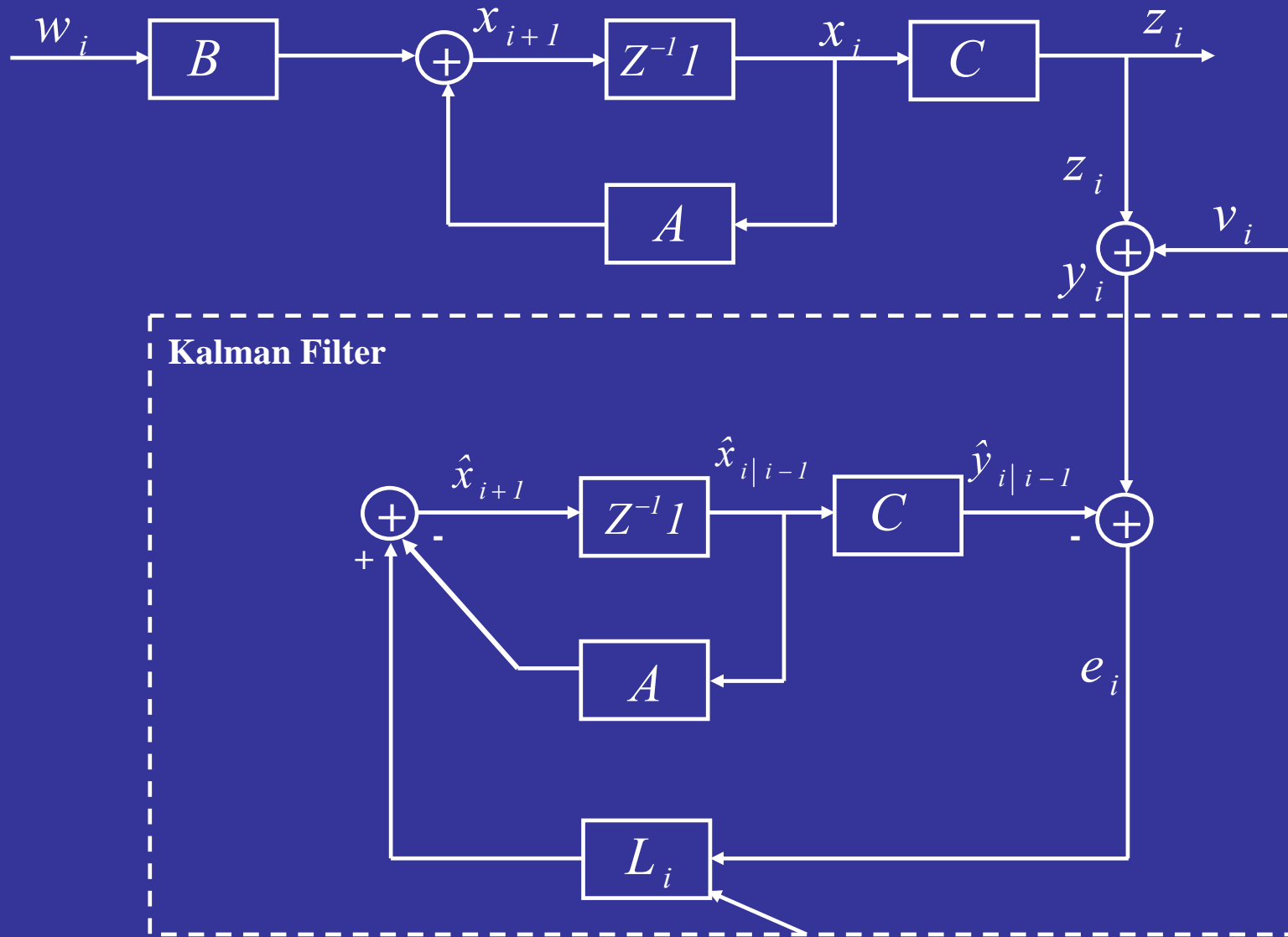
$$\begin{aligned}p_{i+1|i} &= Ap_{i|i}A^T + BwB^T \\ &= A\left[p_{i|i-1} - p_{i|i-1}C^T(Cp_{i|i-1}C^T + V)^{-1}Cp_{i|i-1}\right]A^T + BwB^T\end{aligned}$$

$$p_{i+1|i} = Ap_{i|i}A^T + BwB^T - Ap_{i|i-1}C^T(Cp_{i|i-1}C^T + V)^{-1}(Cp_{i|i-1} - 1)A^T$$

Known as discrete time Riccati Equation



4.6 Picture of Kalman Filter



Time varying gain



4.6 Picture of Kalman Filter

Plant Equations:

$$x_{i+1} = Ax_i + Bu_i$$

$$y_i = Cx_i + v_i$$

Kalman Filter:

$$\hat{x}_{i+1|i} = A\hat{x}_{i|i-1} + L_i (y_i - \hat{y}_{i|i-1})$$

$$y_{i|i-1} = C\hat{x}_{i|i-1}$$

If $v=w=0 \Rightarrow$ Kalman filter can estimate the state precisely in a finite number of steps.



4.6 Kalman Filter

Remarks:

(1) Since $y_i = Cx_i + v_i$ and $\hat{y}_{i|i-1} = C\hat{x}_i$ can write estimator equation as

$$\begin{aligned}\hat{x}_{i+1|i} &= A\hat{x}_{i|i-1} + L_i(Cx_i + v_i - C\hat{x}_{i|i-1}) \\ &= (A - L_iC)\hat{x}_{i|i-1} + L_iC x_i + v_i\end{aligned}$$

can combine this with equation for x_{i+1}

$$\begin{bmatrix} x_{i+1} \\ \hat{x}_{i+1|i} \end{bmatrix} = \begin{bmatrix} A & 0 \\ L_iC & A - L_iC \end{bmatrix} \begin{bmatrix} x_i \\ \hat{x}_{i|i-1} \end{bmatrix} + \begin{bmatrix} B & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} w_i \\ v_i \end{bmatrix}$$

$$\begin{bmatrix} z_i \\ \hat{y}_{i|i-1} \end{bmatrix} = \begin{bmatrix} C & 0 \\ 0 & C \end{bmatrix} \begin{bmatrix} x_i \\ \hat{x}_{i|i-1} \end{bmatrix}$$

(2) In practice, Riccati equation reaches steady state in few steps. People often run with steady-state K.F.i.e

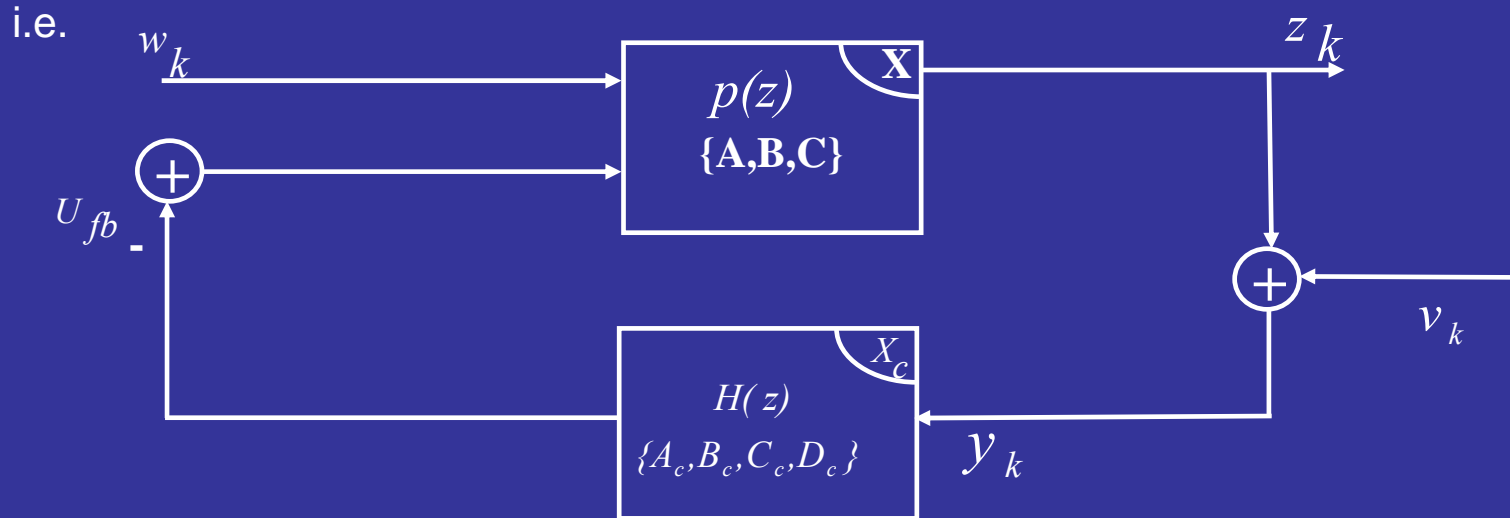
$$L_{ss} = Ap_{ss} C^T (Cp_{ss} C^T + v)^{-1}$$

$$\text{Where } p_{ss} = Ap_{ss} A^T + BwB^T - Ap_{ss} C^T (Cp_{ss} C^T + v)^{-1} Cp_{ss} A$$



4.7 LQG Problem

Now we are finally ready to solve the full control problem.



Given:

$$x_{k+1} = Ax_k + Bu_k + B_w w_k$$

$$z_k = Cx_k$$

$$y_k = Cx_k + v_k$$

$$\langle w_i, w_j \rangle = w \delta_{ij}, \quad \langle v_i, v_j \rangle = v \delta_{ij}$$

$$\langle w_i, v_j \rangle = 0$$

w_k, v_k both Gaussian

For Gaussian, K.L. gives the absolute best estimate

4.7 Problem Statement (in English)

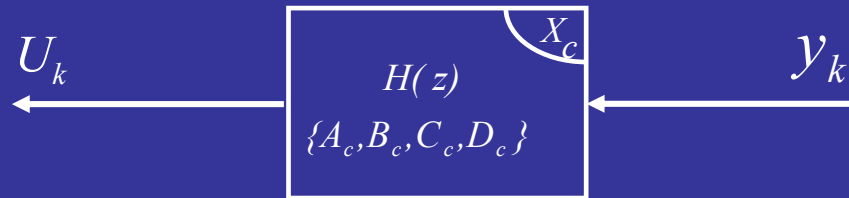
Want a controller which takes as **input** noisy measurements, y , and produces as **output** a feedback signal u , which will minimize excursions of the regulated plant outputs (if no pole-zero cancellation, then this is equivalent to minimizing state excursions.)

Also want to achieve “regulation” with as little actuator effort u , as possible.

Problem statement (Mathematically)

Find: Controller

$$H(z) = C_c (zI - A_c)^{-1} B_c + D_c$$



Controller:

$$x_c(k+1) = A_c x_c(k) + B_c y(k)$$

$$y_c(k) = C_c x_c(k)$$

Which will minimize the cost

Where

$$\begin{cases} x_{k+1} = Ax_k + (-Bu_k) + B_w w_k \\ z_k = C x_k \\ y_k = C x_k + v_k \end{cases}$$

Plant

$$J_{LQG} = \lim_{k \rightarrow \infty} E \left[\underbrace{x_k^T Q x_k}_{\text{Rms "state" excursions}} + \underbrace{u_k^T R u_k}_{\text{Rms "actuator" effort}} \right]$$



4.7 Problem Statement

Remarks:

- Q and R are weighing matrices that allow trading off of rms u and rms x.
- If $Q = C^T \rho C; \rho > 0$ then trade off rms z vs rms u
- In the stochastic LQR case, the only difference is that now we don't have complete state information $y_i = Cx_i + v_i$ we have only noisy observations

i.e. can't use full state feedback.

Idea: Could we use estimated state Feedback? (i.e. $-K \hat{x}_{k|k-1}$)



4.7 LQG problem

Separation principle: (we won't prove)

The separation principle states that the LQG optimal controller is obtained by:

(1) Using Kalman filter to obtain least squares optimal estimate of the plant state,

$$\text{i.e.: Let } x_c(k) = \hat{x}_{k|k-1}$$

(2) Feedback estimated LQR- optimal state feedback

$$\begin{aligned} u(k) &= -K_{LQR} x_c(k) \\ &= -K_{LQR} \hat{x}_{k|k-1} \end{aligned}$$

i.e. can treat problems of

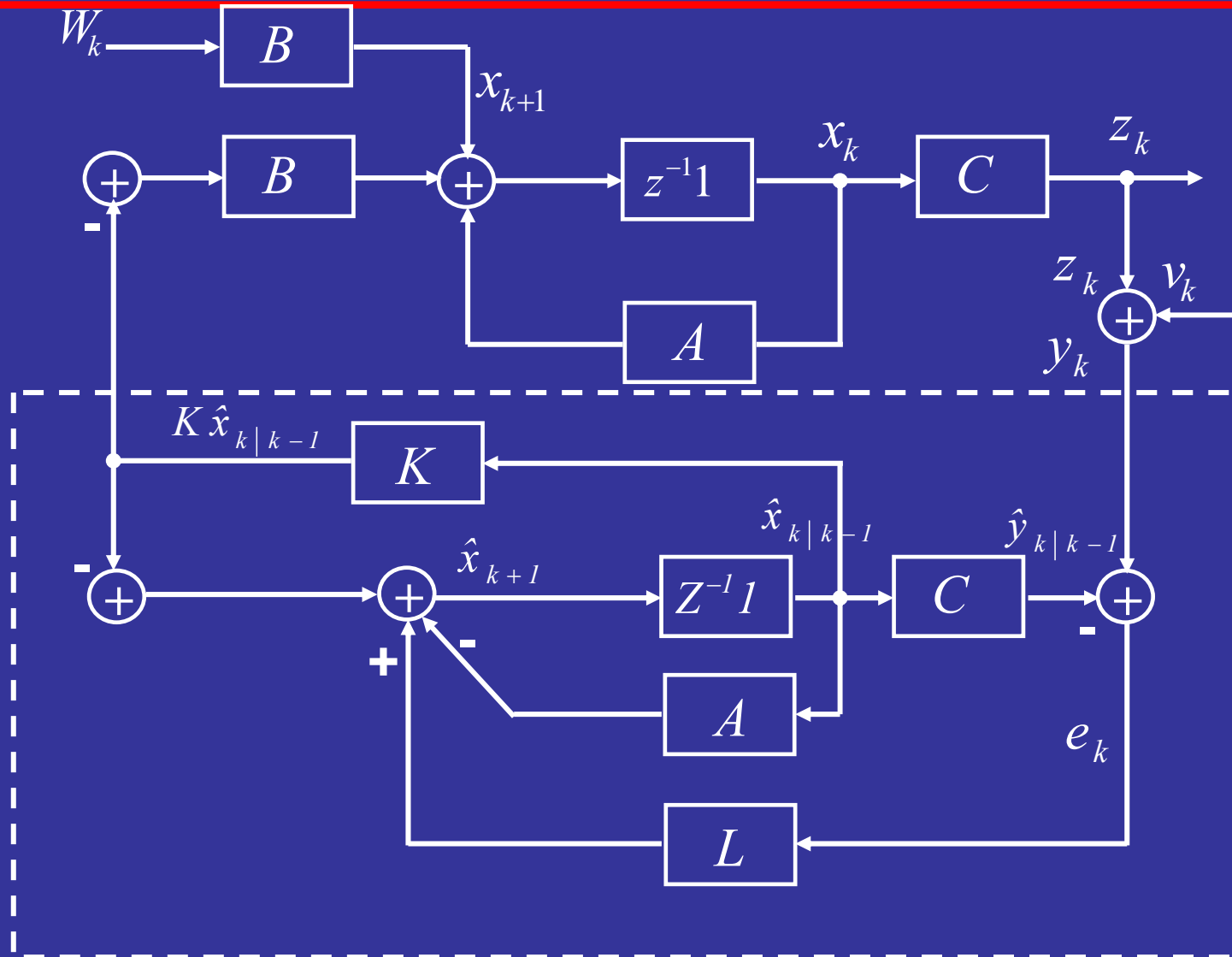
-Optimal feedback and

-state estimate

seperately.



4.7 Picture of LQG Regulator



4.7 LQG Regulator

Plant

$$\left\| \begin{aligned} x_{k+1} &= A x_k + (-B u_k) + B_w w_k \\ z_k &= C x_k \\ y_k &= C x_k + v_k \end{aligned} \right\|$$

LQG Controller

$$\left\| \begin{aligned} \hat{x}_{k+1|k} &= A \hat{x}_{k|k-1} + B u_k + L \left(y_k - C \hat{x}_{k|k-1} \right) \\ u_k &= -K \hat{x}_{k|k-1} \end{aligned} \right\|$$

$$k = - \left[R + B^T S B \right]^{-1} + S = A^T S A + Q - A^T S B \left[R + B^T S B \right]^{-1} B^T S A$$

$$L = A P C^T \left[V + C P C^T \right]^{-1} + P$$

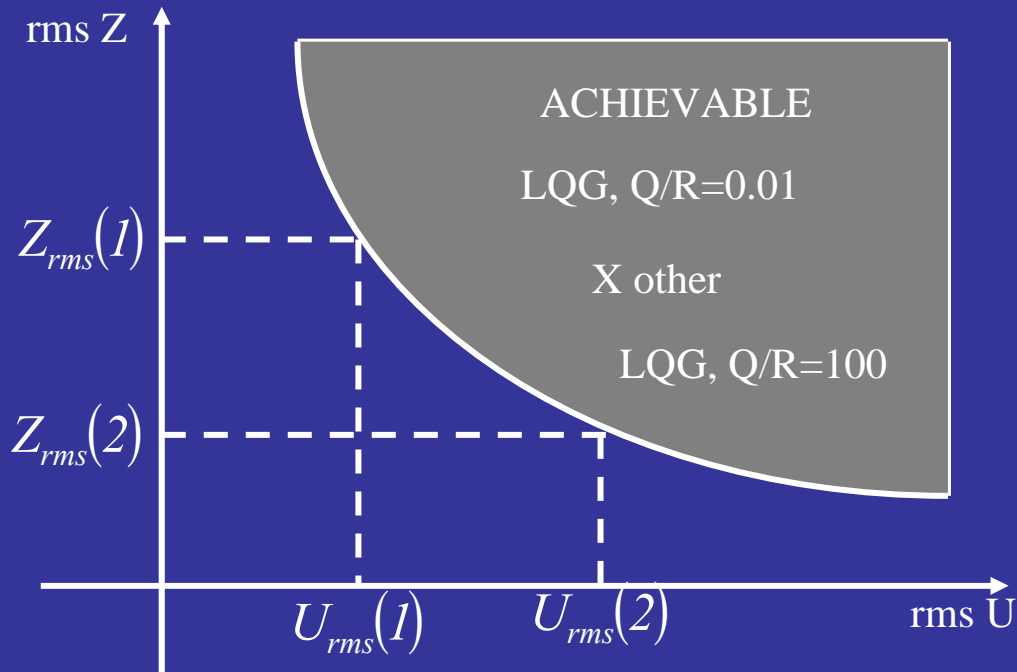
$$= A P A^T + B W B^T - A P C^T \left[V + C P C^T \right]^{-1} C P C^T$$



4.7 Problem Statement

- We can let Q/R ratio vary and we'll obtain family of LQG controllers. Can Plot rms z vs rms u for each one

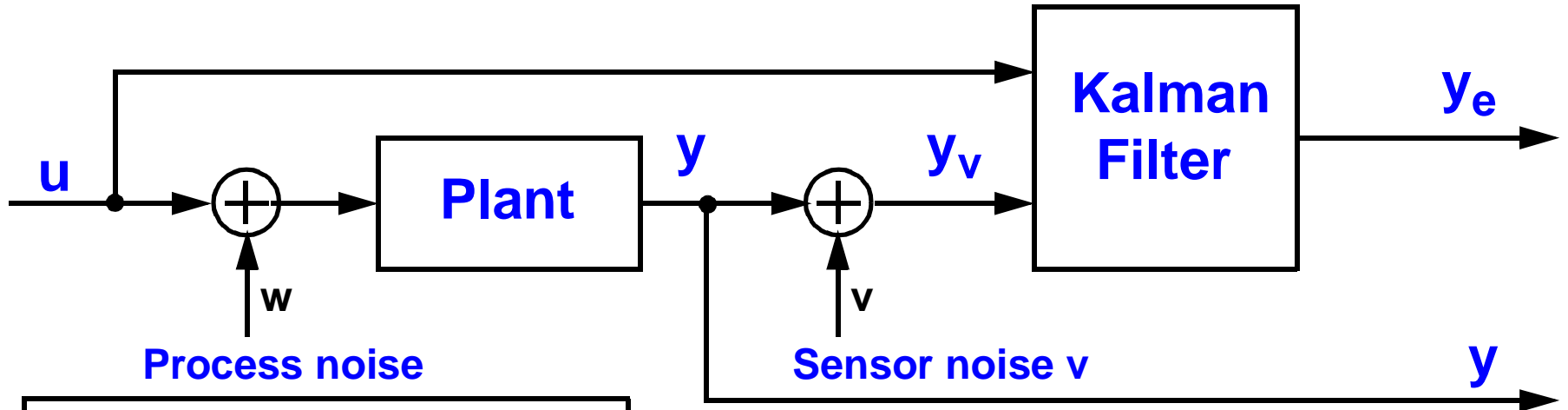
→ Trade-Off curves



So by specifying (1) system model, (2) noise variances, (3) optimally criterion J_{LQG} , and plotting trade off curve completely specifies limit of performance of

System i. e which combinations of (Z_{rms}, U_{rms}) are achievable by any controller -good "benchmark curve".

Principle Kalman Filter (steady state)



Discrete Plant:
 $x[n+1]=Ax[n]+B(u[n]+w[n])$
 $y[n]=Cx[n]$

Noisy output measurement: $y_v[n]=Cx[n]+v[n]$

Measurement update:

$$\hat{x}[n|n]=\hat{x}[n|n-1]+M(y_v[n]-C\hat{x}[n|n-1])$$

Time update: $\hat{x}[n+1|n]=A\hat{x}[n|n]+Bu[n]$

The correction term is a function of the innovation, i.e. the discrepancy

$$y_v[n+1]-C\hat{x}[n+1|n]=C(x[n+1]-\hat{x}[n+1|n])$$

The innovation gain matrix **M** is chosen to minimize steady-state covariance of the estimation error given the noise covariances $E(w[n]w[n]^T)=Q$ and $E(v[n]v[n]^T)=R$

Kalman Filter (Cnt'd)

where M is the solution of the Riccati Equation:

$$M = Q + A M A^T - A M C^T (R + C M C^T)^{-1} C M A^T$$

Combining time and measurement update into state space model (the kalman filter):

$$\hat{x}[n+1|n] = A(I-MC) \cdot \hat{x}[n|n-1] + \begin{bmatrix} B & AM \end{bmatrix} \begin{bmatrix} u[n] \\ y_v[n] \end{bmatrix}$$

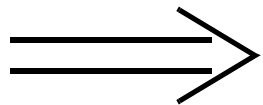
$$\hat{y}[n|n] = C(I-MC) \cdot \hat{x}[n|n-1] + C M y_v[n]$$

This filter generates an optimal estimate $\hat{y}[n|n]$ of $y[n]$. Note that filter state is $\hat{x}[n|n-1]$

Example: TTF Cavity $Q_L = 3 \cdot 10^6$
 $\omega_0 = 1.3 \cdot 10^9$ Hz

Beam noise : $\sigma(I_b)/I_b = 0.1$

Sensor noise : $\sigma(V_d)/V_d = 0.01$



$$\sigma_y / y = 0.0009$$

