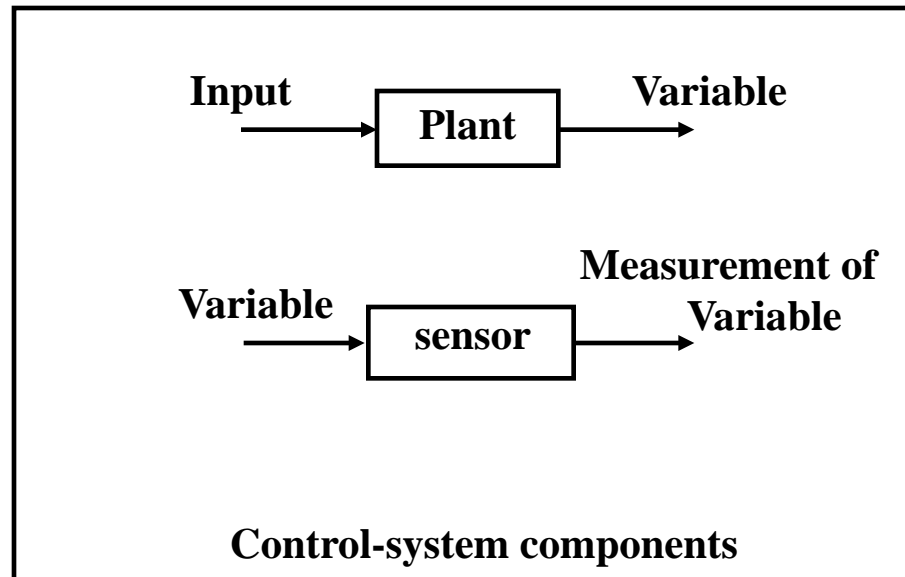# 1.Control Theory

Objective:
The course on control theory is concerned with the analysis and design of closed loop control systems.

Analysis:
Closed loop system is given ⟶ determine characteristics or behavior

Design:
Desired system characteristics or behavior are specified ⟶ configure or synthesize closed loop system

Control-system components
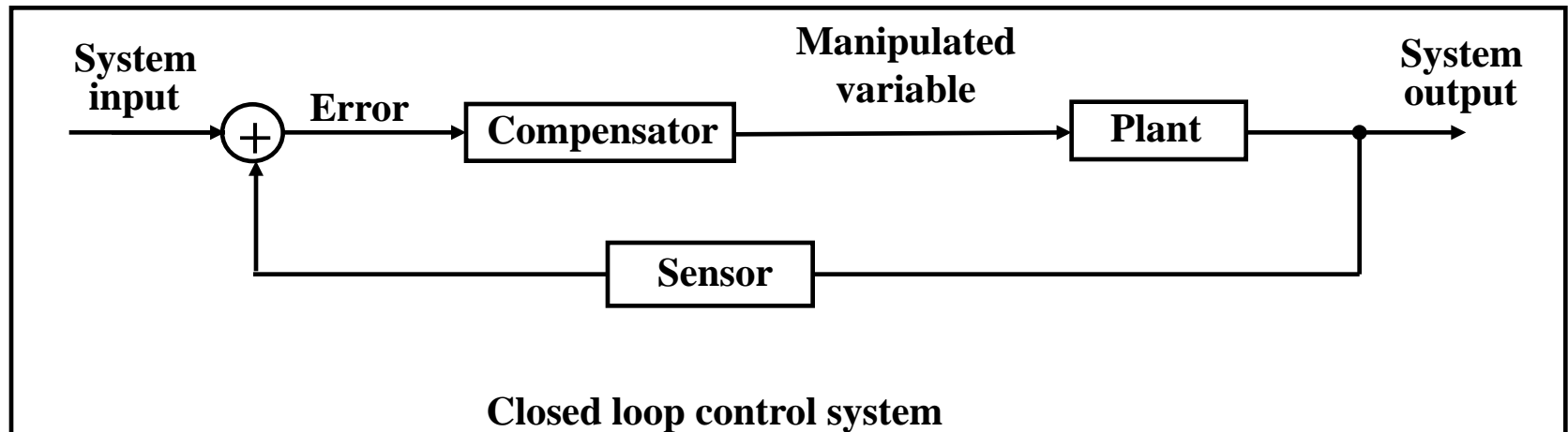
1.Introduction

Definition:

A closed-loop system is a system in which certain forces (we call these inputs) are determined, at least in part, by certain responses of the system (we call these outputs).

System inputs   **System outputs**

Closed loop system
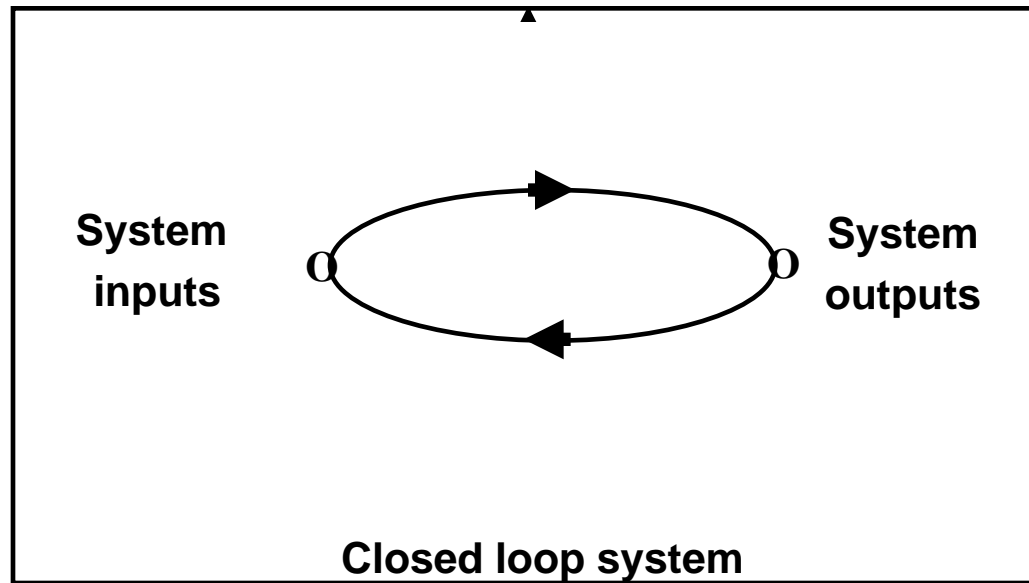
1.Introduction

Definitions:

❖The system for measurement of a variable (or signal) is called a ***sensor.***

❖A ***plant*** of a control system is the part of the system to be controlled.

❖The ***compensator*** (or controller or simply filter) provides satisfactory characteristics for the total system.

System input ──► (**+**) **Error** ──► | **Compensator** | ──**Manipulated variable**──► | **Plant** | ──► **System output**

| **Sensor** |

Closed loop control system

Two types of control systems:

❖A ***regulator*** maintains a physical variable at some constant value in the presence of perturbances.

❖A ***servomechanism*** describes a control system in which a physical variable is required to follow, or track some desired time function (originally applied in order to control a mechanical position or motion).
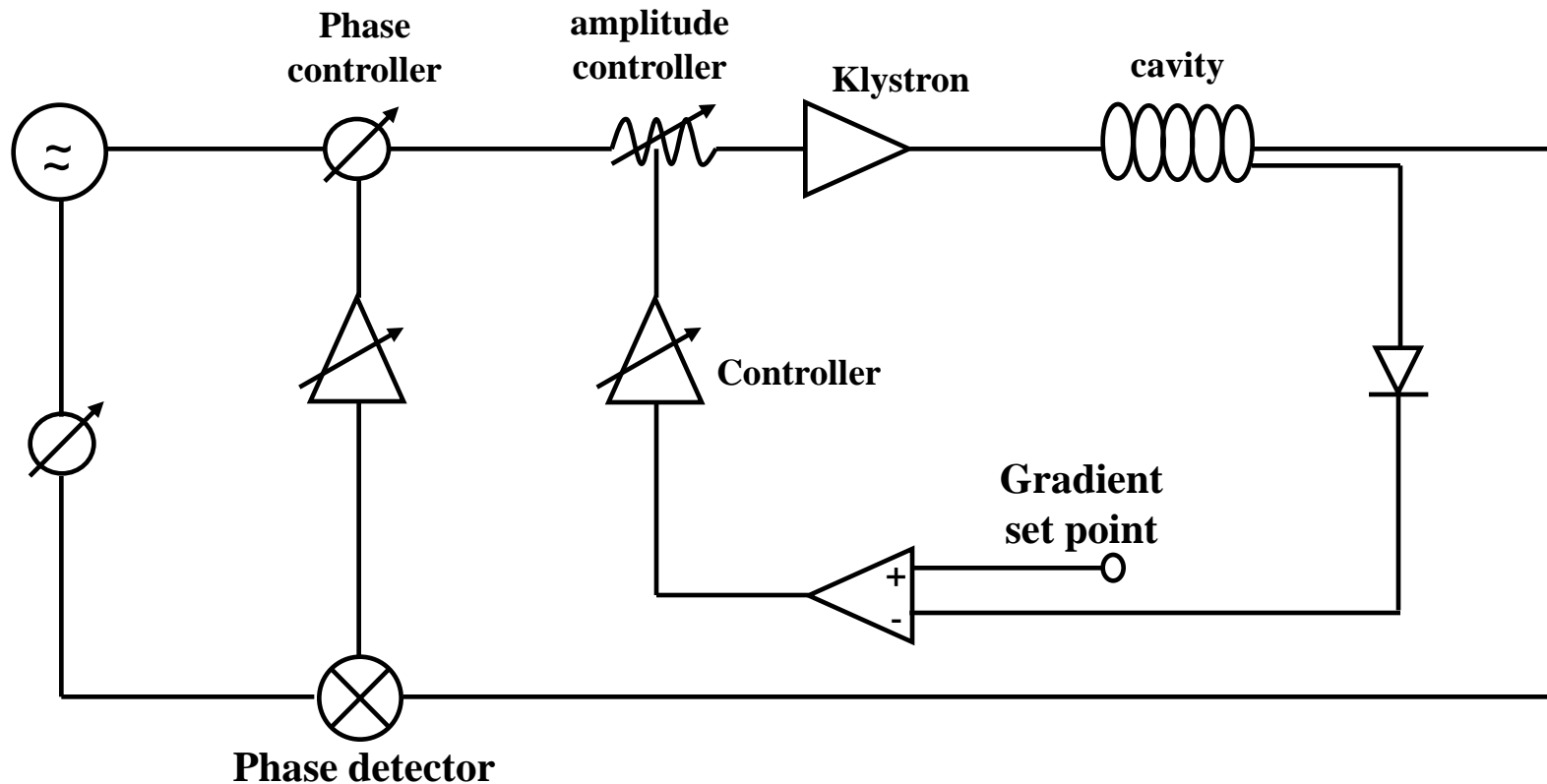
1.Introduction

Example 1: RF control system

Goal:
Maintain stable gradient and phase.
Solution:
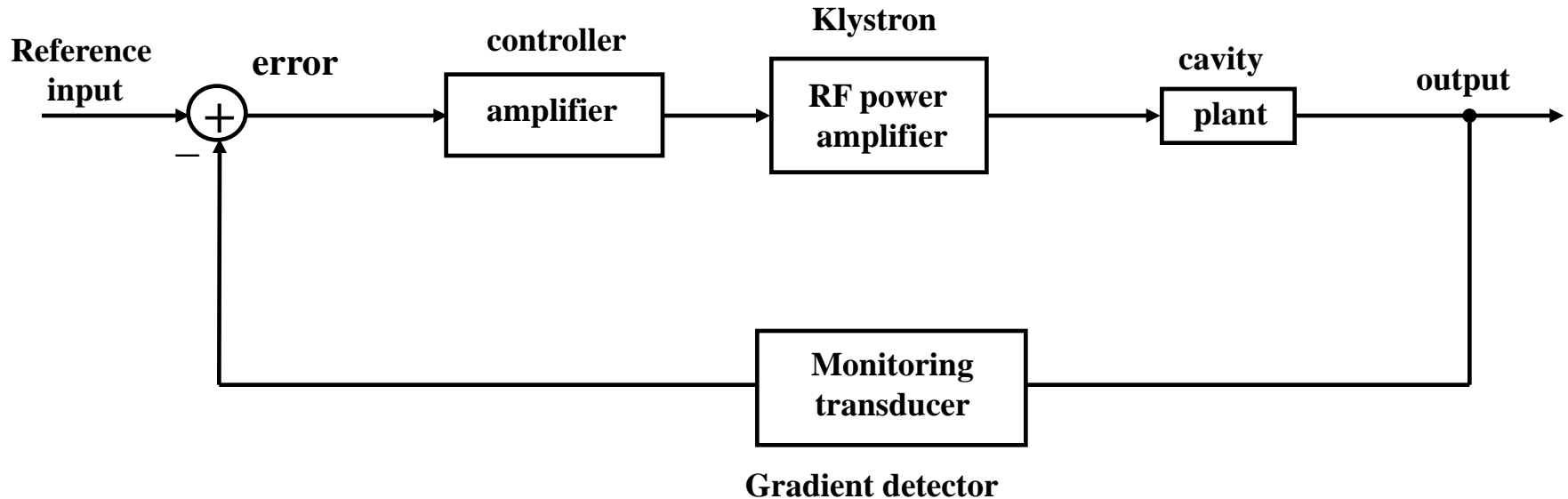Feedback for gradient amplitude and phase.
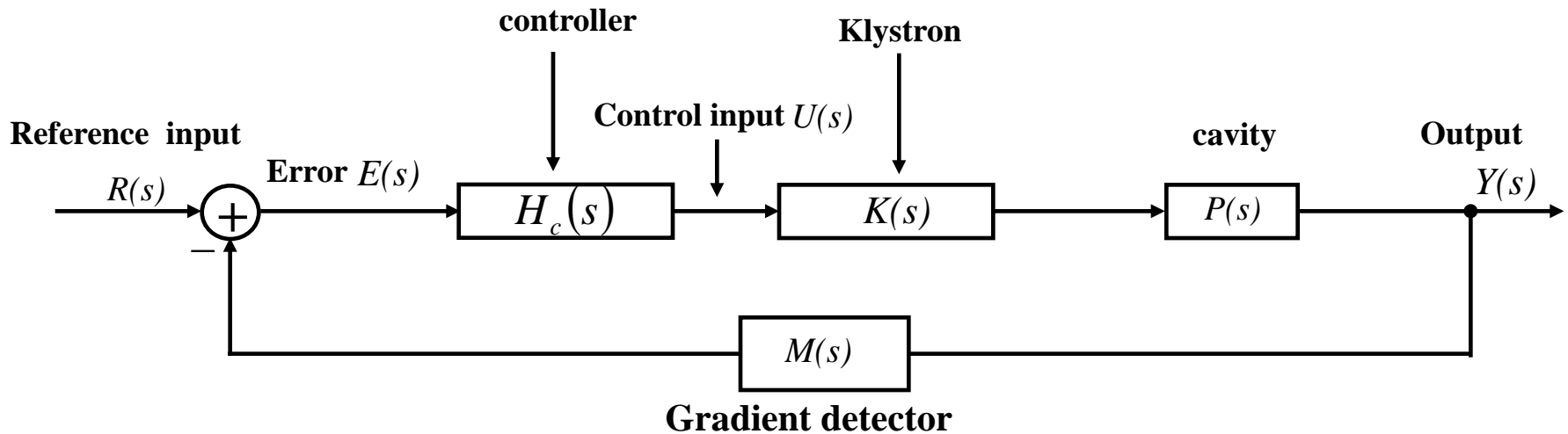
1.Introduction

Model:
Mathematical description of input-output relation of components combined with block diagram.

Amplitude loop (general form):

1.Introduction

RF control model using *"transfer functions"*

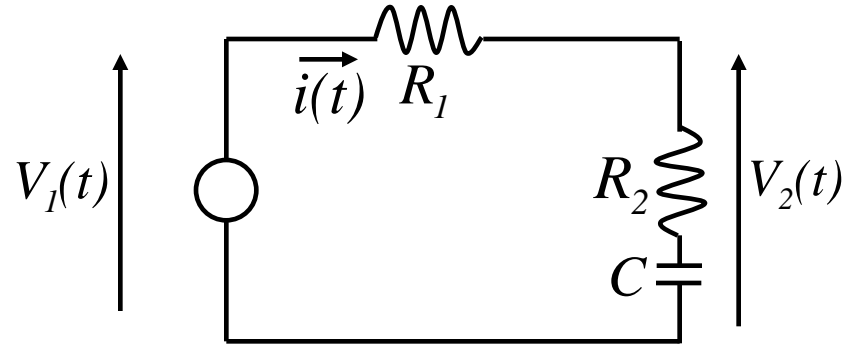A transfer function of a **linear** system is defined as the ratio of the Laplace transform of the output and the Laplace transform of the input with I. C .'s =zero.

Input-Output Relations

Input	Output	Transfer Function
$U(s)$	$Y(s)$	$G(s) = P(s)K(s)$
$E(s)$	$Y(s)$	$L(s) = G(s)H_c(s)$
$R(s)$	$Y(s)$	$T(s) = (1 + L(s)M(s))^{-1}L(s)$

1.Introduction

Example2: Electrical circuit

Differential equations:

$$R_1\ i(t) + R_2\ i(t) + \frac{1}{C}\int_0^t i(\tau)\ d\tau = V_1(t)$$

$$R_2\ i(t) + \frac{1}{C}\int_0^t i(\tau)\ d\tau = V_2(t)$$
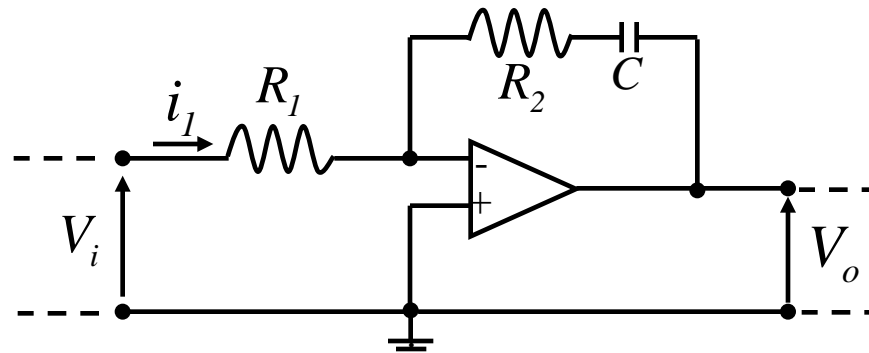
Laplace Transform:

$$R_1\ I(s) + R_2\ I(s) + \frac{1}{s\cdot C}I(s) = V_1(s)$$

$$R_2\ I(s) + \frac{1}{s\cdot C}I(s) = V_2(s)$$

Transfer function:

$$G(s) = \frac{V_2(s)}{V_1(s)} = \frac{R_2\cdot C\cdot s + 1}{(R_1 + R_2\ )C\cdot s + 1}$$

Input V_1 ,output V_2
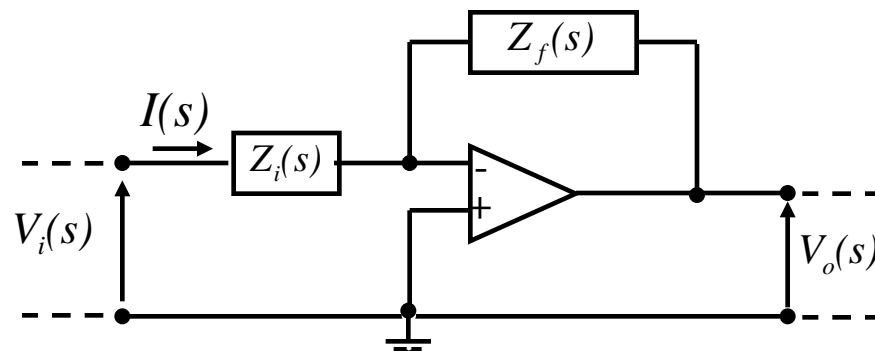
1.Introduction

Example 3: Circuit with operational amplifier

$$V_i(s) = R_1 I_1(s) \quad \text{and} \quad V_o(s) = -\left(R_2 + \frac{1}{s \cdot C} \right) I_1(s)$$

$$G(s) = \frac{V_o(s)}{V_i(s)} = -\frac{R_2 \cdot C \cdot s + 1}{R_1 \cdot C \cdot s}$$
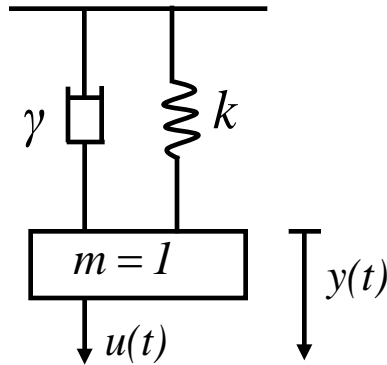
It is convenient to derive a transfer function for a circuit with a single operational amplifier that contains input and feedback impedance:

$$V_i(s) = Z_i(s)\, I(s) \quad \text{and} \quad V_o(s) = -Z_f(s)\, I(s) \quad \longrightarrow \quad G(s) = \frac{V_o(s)}{V_i(s)} = -\frac{Z_f(s)}{Z_i(s)}$$

Model of Dynamic System

We will study the following dynamic system:

Parameters:

k : spring constant

γ : damping constant

$u(t)$: force

Quantity of interest:

$y(t)$: displacement from equilibrium

Differential equation: Newton's third law $(m = 1)$

$$\ddot{y}(t) = \sum F_{ext} = -k\, y(t) - \gamma\, \dot{y}(t) + u(t)$$

$$\ddot{y}(t) + \gamma\, \dot{y}(t) + k\, y(t) = u(t)$$

$$y(0) = y_0 \, , \; \dot{y}(0) = \dot{y}_0$$

➤Equation is linear (i.e. no $\dot{y}^2$ like terms).

➤Ordinary (as opposed to partial e.g. $= \dfrac{\partial}{\partial x}\dfrac{\partial}{\partial t} f(x,t) = 0$)

➤All coefficients constant: $k(t) = \kappa$, $\gamma(t) = \gamma$ for all t
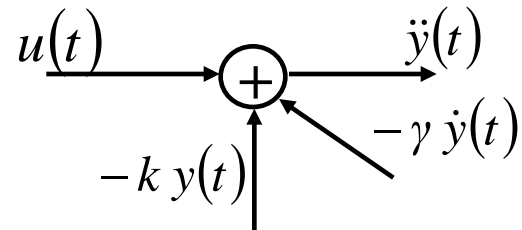
Model of Dynamic System

Stop calculating, let's paint!!!
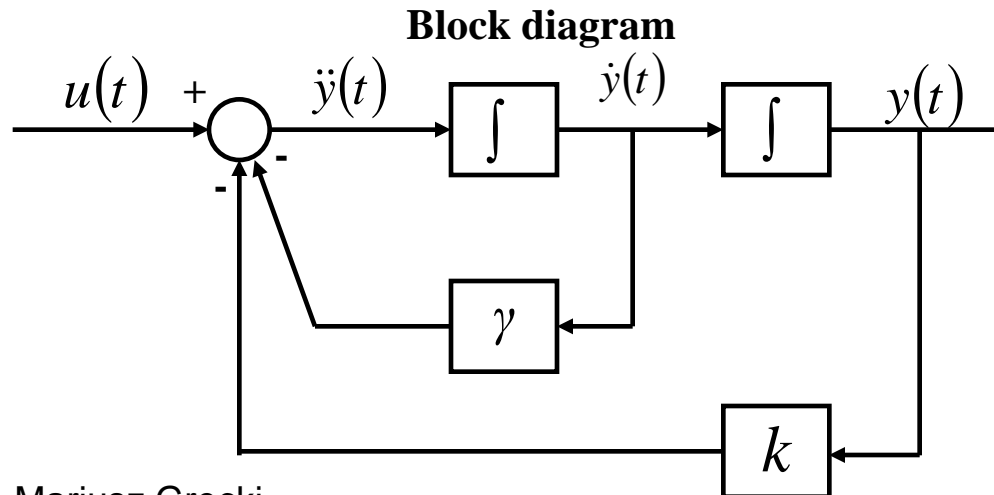
Picture to visualize differential equation

1. Express highest order term (put it to one side)

$$\ddot{y}(t) = -k\,y(t) - \gamma\,\dot{y}(t) + u(t)$$

2. Putt adder in front

3. Synthesize all other terms using integrators!

Block diagram

2.1 Linear Ordinary Differential Equation (LODE)

General form of LODE:

$$y^{(n)}(t) + a_{n-1} \, y^{(n-1)}(t) + \ldots + a_1 \, \dot{y}(t) + a_0 \, y(t) = b_m \, u^{(m)}(t) + \ldots + b_1 \, \dot{u}(t) + b_0 \, u(t)$$

m, n Positive integers, $m \leq n$;   coefficients $a_0, a_1, \ldots, a_{n-1}, \; b_0, \ldots, b_m$ real numbers.

__Mathematical solution: hopefully you know it__

Solution of LODE: $y(t) = y_h(t) + y_p(t)$,

Sum of homogeneous solution $y_h(t)$ (natural response) solving

$$y^{(n)}(t) + a_{n-1} \, y^{(n-1)}(t) + \ldots + a_1 \, \dot{y}(t) + a_0 \, y(t) = 0$$

And particular solution $y_p(t)$.

How to get natural response $y_h(t)$? Characteristic polynomial

$$\chi(\lambda) = \lambda^n + a_{n-1}\lambda^{n-1} + a_1\lambda + a_0 = 0$$

$$(\lambda - \lambda_1)^r \cdot (\lambda - \lambda_{r+1}) \cdot \ldots \cdot (\lambda - \lambda_n) = 0$$

$$y_h(t) = (c_1 + c_2 \, t + \ldots + c_r \, t^{r-1}) e^{\lambda_1 t} + c_{r+1}e^{\lambda_{r+1} t} + \ldots + c_n e^{\lambda_n t}$$

Determination of $y_p(t)$ relatively simple, if input $u(t)$ yields only a finite number of independent derivatives. E.g.: $u(t) \cong e^{\xi t}, \; \beta_r t^r$.

2.1Linear Ordinary Differential Equation (LODE)

Most important for control system/feedback design:

$$y^{(n)}(t) + a_{n-1}\, y^{(n-1)}(t) + \ldots + a_1\, \dot{y}(t) + a_0\, y(t) = b_m\, u^{(m)}(t) + \ldots + b_1\, \dot{u}(t) + b_0\, u(t)$$

In general: given any linear time invariant system described by LODE can be realized/simulated/easily visualized in a block diagram $(n = 2,\ m = 2)$

Control-canonical form
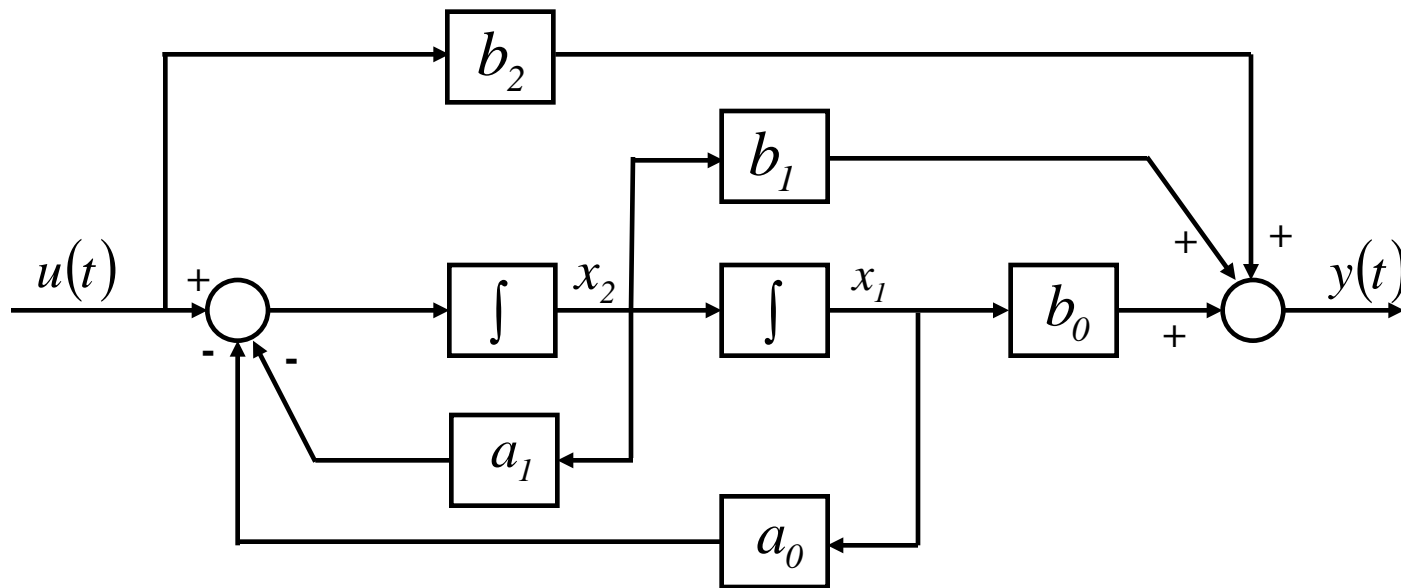
Very useful to visualize <u>*interaction*</u> between variables!
What are x_1 and x_2 ????

More explanation later, for now: please simply accept it!

2.2 State Space Equation

Any system which can be presented by LODE can be represented in *State space form* (matrix differential equation).

<div align="center">What do we have to do ???</div>

Let's go back to our first example (Newton's law):

$$\ddot{y}(t) + \gamma\,\dot{y}(t) + k\,y(t) = u(t)$$

1. STEP: Deduce set off first order differential equation in variables

$x_j(t)$ (so-called states of system)

$x_1(t) \cong$ Position : $y(t)$

$x_2(t) \cong$ Velocity : $\dot{y}(t)$:

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

$$\dot{x}_2(t) = \ddot{y}(t) = -k\,y(t) - \gamma\,\dot{y}(t) + u(t)$$
$$= -k\,x_1(t) - \gamma\,x_2(t) + u(t)$$

<div align="center"><u>**One LODE of order *n* transformed into *n* LODEs of order 1**</u></div>

2.2 State Space Equation

2. STEP:

Put everything together in a matrix differential equation:

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k & -\gamma \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t)$$

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

State equation

$$y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$$

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

Measurement equation

Definition:

The **system state** x of a system at any time t_0 is the "amount of information" that, together with all inputs for $t \geq t_0$, uniquely determines the behaviour of the system for all $t \geq t_0$.

2.2 State Space Equation

The linear time-invariant (LTI) analog system is described via
Standard form of the State Space Equation

$$\dot{x}(t) = A\, x(t) + B\, u(t) \qquad \text{State equation}$$

$$y(t) = C\, x(t) + D\, u(t) \qquad \text{State equation}$$

Where $\dot{x}(t)$ is the time derivative of the vector $\quad x(t) = \begin{bmatrix} x_1(t) \\ \cdots \\ x_n(t) \end{bmatrix}$. And starting conditions $x(t_0)$

System completely described by state space matrixes $A,\ B,\ C,\ D$ (in the most cases $D = 0$).

Declaration of variables

Variable	Dimension	Name
$X(t)$	$n \times 1$	state vector
A	$n \times n$	system matrix
B	$n \times r$	input matrix
$u(t)$	$r \times 1$	input vector
$y(t)$	$p \times 1$	output vector
C	$p \times n$	output matrix
D	$p \times r$	matrix representing direct coupling between input and output

2.2 State Space Equation

Why all this work with state space equation? Why bother with?

BECAUSE: Given any system of the LODE form

$$y^{(n)}(t) + a_{n-1}\, y^{(n-1)}(t) + \ldots + a_1\, \dot{y}(t) + a_0\, y(t) = b_m\, u^{(m)}(t) + \ldots + b_1\, \dot{u}(t) + b_0\, u(t)$$

Can be represented as

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

with e.g. ***Control-Canonical Form*** (case $n = 3$, $m = 3$):

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_0 & -a_1 & -a_2 \end{bmatrix},\; B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix},\; C = \begin{bmatrix} b_0 & b_1 & b_2 \end{bmatrix},\; D = b_3$$

or ***Observer-Canonical Form:***

$$A = \begin{bmatrix} 0 & 0 & -a_0 \\ 1 & 0 & -a_1 \\ 0 & 1 & -a_2 \end{bmatrix},\; B = \begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix},\; C = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix},\; D = b_3$$

Notation is very compact, But: not unique!!!
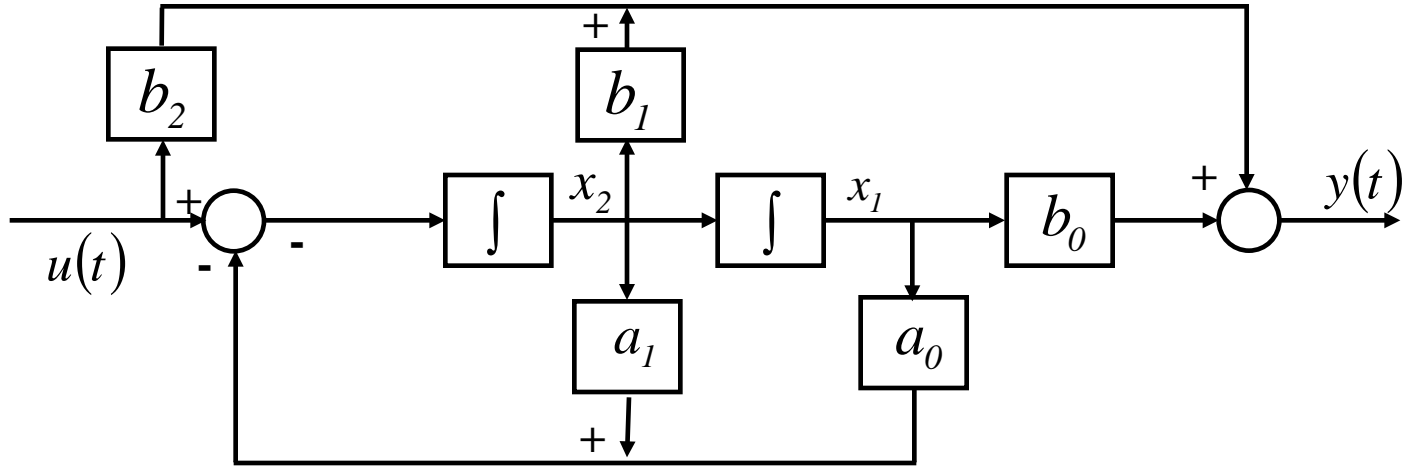Computers love state space equation! (Trust us!)
Modern control (1960-now) uses state space equation.
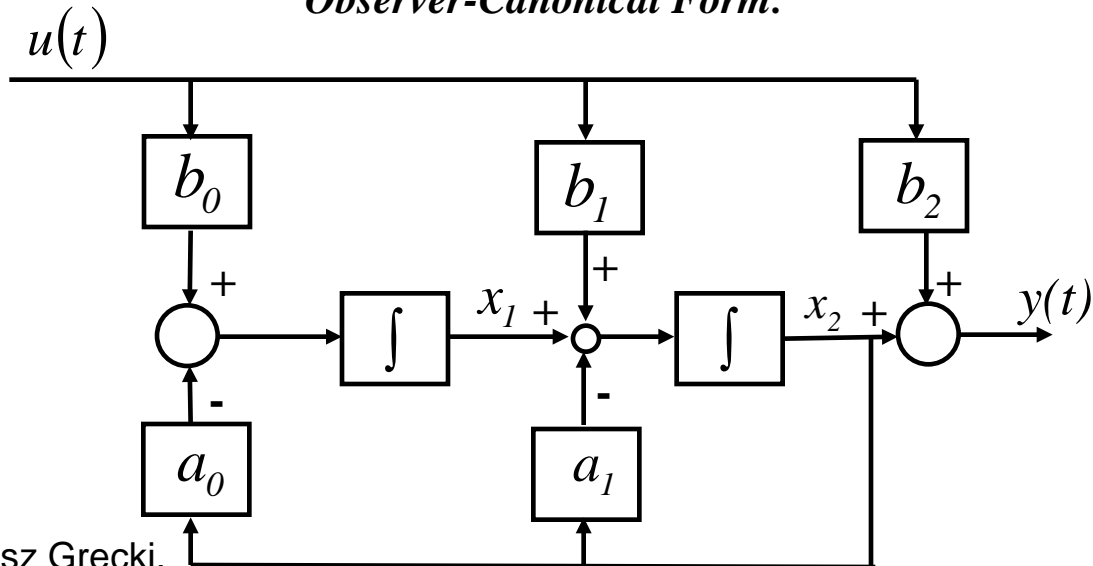General (vector) block diagram for easy visualization.

2.2 State Space Equation

Block diagrams:

Control-canonical Form:

Observer-Canonical Form:

2.2 State Space Equation

Now: Solution of State Space Equation in the time domain. Out of the hat…et voila:

$$x(t) = \Phi(t)\, x(0) + \int_0^t \Phi(\tau)\, B\, u(t-\tau)\, d\tau$$

Natural Response + Particular Solution

$$y(t) = C\, x(t) + D\, u(t)$$
$$= C\, \Phi(t)\, x(0) + C \int_0^t \Phi(\tau)\, B\, u(t-\tau)\, d\tau + D\, u(t)$$

With the **_state transition matrix_**

$$\Phi(t) = I + At + \frac{A^2}{2!}t^2 + \frac{A^3}{3!}t^3 + \ldots = e^{A\,t}$$

Exponential series in the matrix A (time evolution operator) properties of $\Phi(t)$ (state transition matrix).

$$1.\frac{d\Phi(t)}{dt} = A\, \Phi(t)$$
$$2.\Phi(0) = I$$
$$3.\Phi(t_1 + t_2) = \Phi(t_1)\cdot \Phi(t_2)$$
$$4.\Phi^{-1}(t) = \Phi(-t)$$

Example:

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \Rightarrow A^2 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \ \Phi(t) = I + At = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix} = e^{A\,t}$$

Matrix A is a nilpotent matrix.

2.3 Examples

Example:

It is given the following differential equation:

$$\frac{d^2}{dt^2} y(t) + 4 \frac{d}{dt} y(t) + 3 \, y(t) = 2 \, u(t)$$

➤State equations of differential equation:

Let $x_1(t) = y(t)$ and $x_2(t) = \dot{y}(t)$. It is:

$$\dot{x}_1(t) = \dot{y}(t) = x_2(t)$$
$$\dot{x}_2(t) + 4 \, x_2(t) + 3 \, x_1(t) = 2 \, u(t)$$
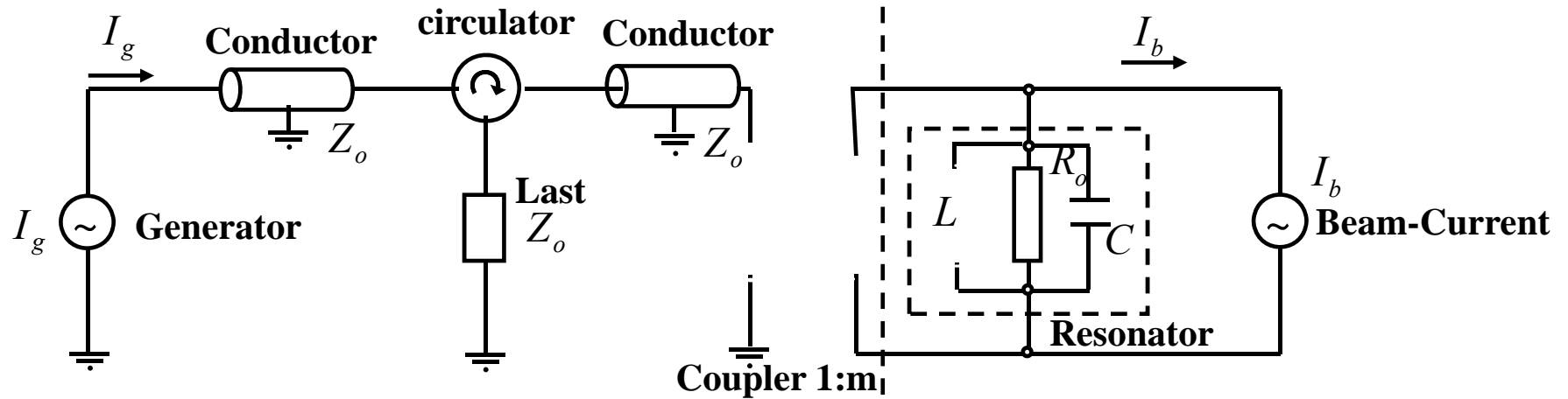$$\dot{x}_2(t) = -3 \, x_1(t) - 4 \, x_2(t) + 2 \, u(t)$$

➤Write the state equations in matrix form:

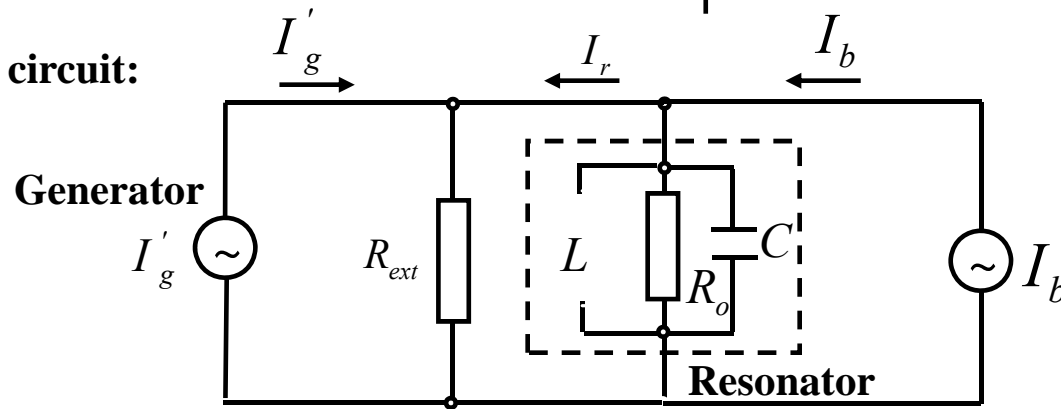Define system state $x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$. Then it follows:

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 \\ -3 & -4 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 2 \end{bmatrix} u(t)$$
$$y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} x(t)$$

2.3 Cavity Model

$$C \cdot \ddot{U} + \frac{1}{R_L} \cdot \dot{U} + \frac{1}{L} \cdot U = \dot{I}'_g + \dot{I}_b$$

$$\omega_{1/2} := \frac{1}{2R_L C} = \frac{\omega_0}{2Q_L}$$

$$\ddot{U} + 2\omega_{1/2} \cdot \dot{U} + \omega_0^2 \cdot U = 2R_L \omega_{1/2} \cdot \left(\frac{2}{m} \dot{I}_g + \dot{I}_b \right)$$

2.3 Cavity Model

Only envelope of **rf** (real and imaginary part) is of interest:

$$U(t) = \left(U_r(t) + i\, U_i(t)\right) \cdot exp(i\, \omega_{HF} t)$$

$$I_g(t) = \left(I_{gr}(t) + i\, I_{gi}(t)\right) \cdot exp(i\, \omega_{HF} t)$$

$$I_b(t) = \left(I_{b\omega r}(t) + i\, I_{b\omega i}(t)\right) \cdot exp(i\, \omega_{HF} t) = 2\left(I_{b0r}(t) + i\, I_{b0i}(t)\right) \cdot exp(i\, \omega_{HF}\, t)$$

Neglect small terms in derivatives for U and I

$$\ddot{U}_r + i\ddot{U}_i(t) << \omega_{HF}^2 \left(U_r(t) + iU_i(t)\right)$$

$$2\omega_{1/2}\left(\dot{U}_r + i\dot{U}_r(t)\right) << \omega_{HF}^2 \left(U_r(t) + iU_i(t)\right)$$

$$\int_{t1}^{t2} \left(\dot{I}_r(t) + i\dot{I}_i(t)\right) dt << \int_{t1}^{t2} \omega_{HF}\left(I_r(t) + iI_i(t)\right) dt$$

Envelope equations for real and imaginary component.

$$\boxed{\begin{aligned}
\dot{U}_r(t) + \omega_{1/2} \cdot U_r + \Delta\omega \cdot U_i &= \omega_{HF}\left(\frac{r}{Q}\right) \cdot \left(\frac{1}{m} I_{gr} + I_{b0r}\right) \\
\dot{U}_i(t) + \omega_{1/2} \cdot U_i - \Delta\omega \cdot U_r &= \omega_{HF}\left(\frac{r}{Q}\right) \cdot \left(\frac{1}{m} I_{gi} + I_{b0i}\right)
\end{aligned}}$$

Stefan Simrock & Mariusz Grecki, 5th LC School, Switzerland 2010

2.3 Cavity Model

Matrix equations:

$$\begin{bmatrix} \dot{U}_r(t) \\ \dot{U}_i(t) \end{bmatrix} = \begin{bmatrix} -\omega_{1/2} & -\Delta\omega \\ \Delta\Delta\omega & -\omega_{1/2} \end{bmatrix} \cdot \begin{bmatrix} U_r(t) \\ U_i(t) \end{bmatrix} + \omega_{HF}\left(\frac{r}{Q}\right) \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \dfrac{1}{m}I_{gr}(t) + I_{b0r}(t) \\ \dfrac{1}{m}I_{gi}(t) + I_{b0i}(t) \end{bmatrix}$$

With system Matrices:

$$A = \begin{bmatrix} -\omega_{1/2} & -\Delta\omega \\ \Delta\omega & -\omega_{1/2} \end{bmatrix} \qquad B = \omega_{HF}\left(\frac{r}{Q}\right) \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\vec{x}(t) = \begin{bmatrix} U_r(t) \\ U_i(t) \end{bmatrix} \qquad \vec{u}(t) = \begin{bmatrix} \dfrac{1}{m}I_{gr}(t) + I_{b0r}(t) \\ \dfrac{1}{m}I_{gi}(t) + I_{b0i}(t) \end{bmatrix}$$

General Form:

$$\dot{\vec{x}}(t) = A \cdot \vec{x}(t) + B \cdot \vec{u}(t)$$

Stefan Simrock & Mariusz Grecki,
5th LC School, Switzerland 2010

2.3 Cavity Model

Solution:

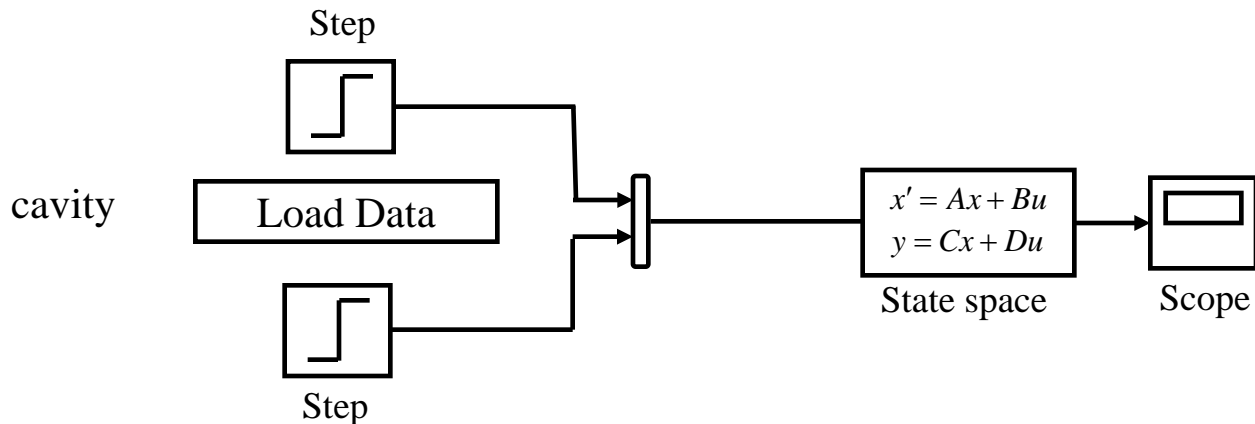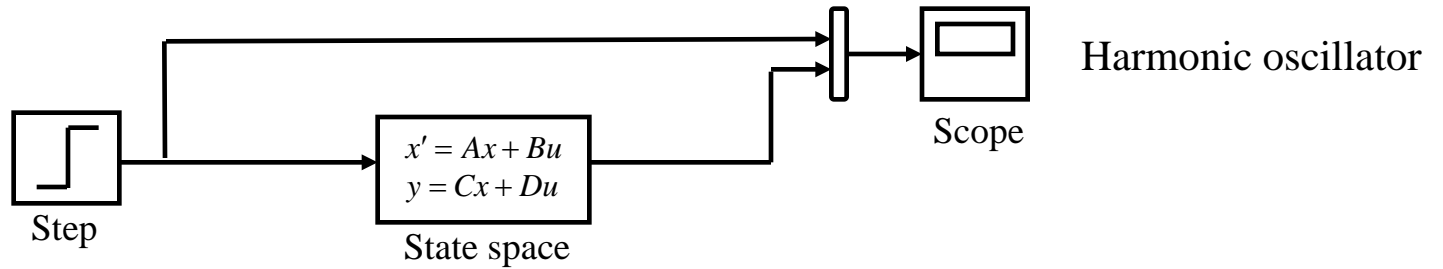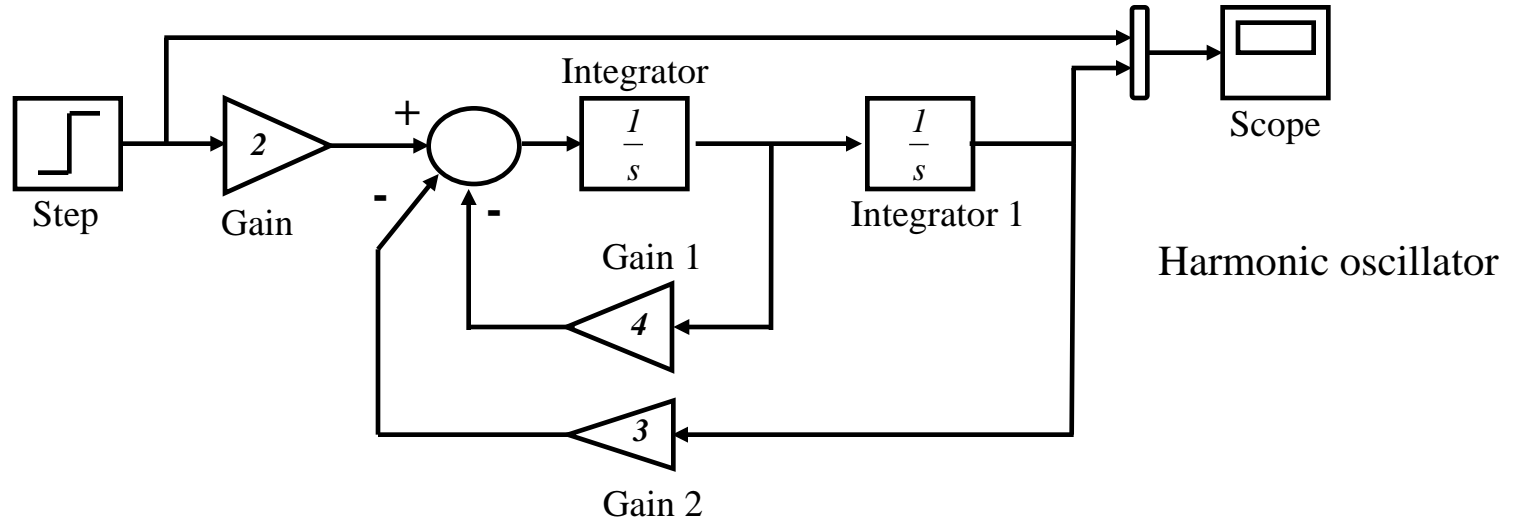$$\vec{x}(t) = \Phi(t) \cdot \vec{x}(0) + \int_0^t \Phi(t - t') \cdot B \cdot \vec{u}(t') \, dt'$$

$$\Phi(t) = e^{-\omega_{1/2} \, t} \begin{bmatrix} cos(\Delta\omega t) & -sin(\Delta\omega t) \\ sin(\Delta\omega t) & cos(\Delta\omega t) \end{bmatrix}$$
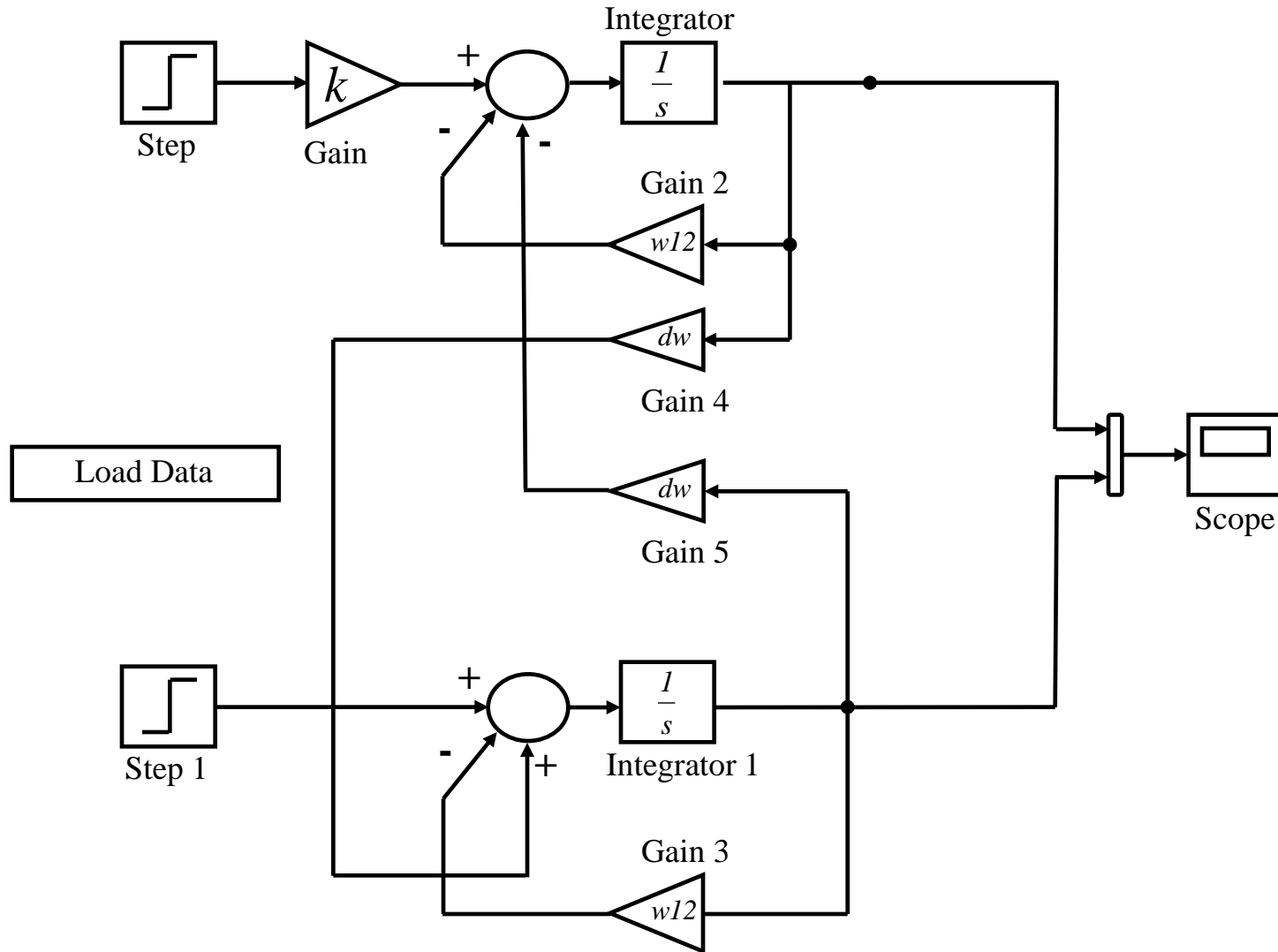
Special Case:

$$\vec{u}(t) = \begin{bmatrix} \dfrac{1}{m} I_{gr}(t) + I_{b0r}(t) \\ \dfrac{1}{m} I_{gi}(t) + I_{b0i}(t) \end{bmatrix} =: \begin{bmatrix} I_r \\ I_i \end{bmatrix}$$

$$\begin{bmatrix} U_r(t) \\ U_i(t) \end{bmatrix} = \frac{\omega_{HF}\left(\dfrac{r}{Q}\right)}{\omega_{1/2}^2 + \Delta\omega^2} \cdot \begin{bmatrix} \omega_{1/2} & -\Delta\omega \\ \Delta\omega & \omega_{1/2} \end{bmatrix} \cdot \left\{ 1 - \begin{bmatrix} cos(\Delta\omega t) & -sin(\Delta\omega t) \\ sin(\Delta\omega t) & cos(\Delta\omega t) \end{bmatrix} e^{-\omega_{1/2} \, t} \right\} \cdot \begin{bmatrix} I_r \\ I_i \end{bmatrix}$$

2.3 Cavity Model

Integrator

$$\frac{1}{s}$$

Integrator 1

$$\frac{1}{s}$$

Step

Gain

2

Gain 1

4

Gain 2

3

Scope

Harmonic oscillator

Harmonic oscillator

$$x' = Ax + Bu$$
$$y = Cx + Du$$

Step

State space

Scope

Step

cavity

Load Data

$$x' = Ax + Bu$$
$$y = Cx + Du$$

State space

Scope

Step

24

Stefan Simrock & Mariusz Grecki, 5th LC School, Switzerland 2010
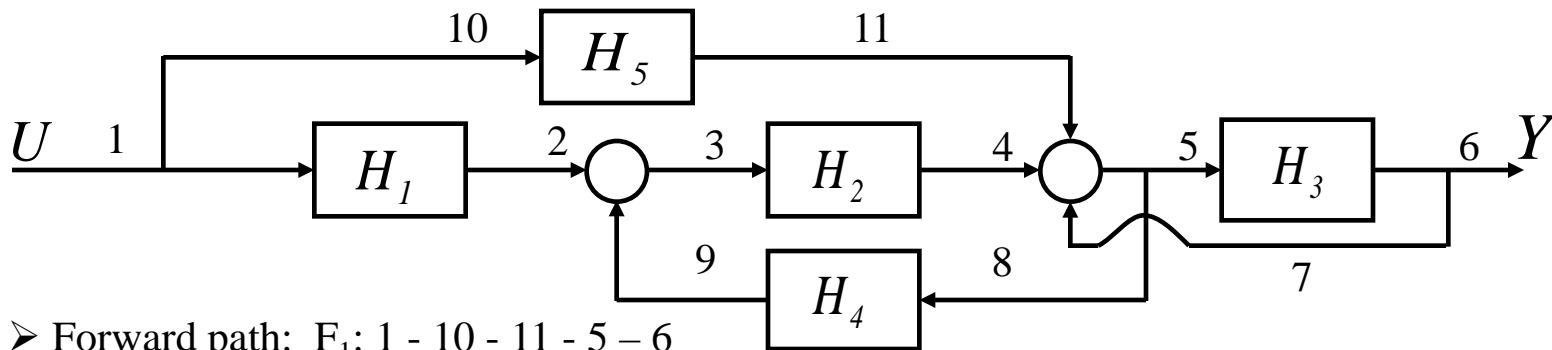
2.3 Cavity Model

2.4 Masons Rule

Mason's Rule is a simple formula for reducing block diagrams. Works on continuous and discrete. In its most general form it is messy, but **For <u>special case</u> when <u>all path touch</u>**

$$H(s) = \frac{\sum(forward\ path\ gains)}{1 - \sum(loop\ path\ gains)}$$

Two path are said to <u>touch</u> if they have a component in common, e.g. an adder.

➤ Forward path: F_1: 1 - 10 - 11 - 5 – 6
  $\quad\quad\quad\quad\quad$ F_2: 1 - 2 - 3 - 4 - 5 – 6

$$G(f_1) = H_5\ H_3$$
$$G(f_2) = H_1\ H_2\ H_3$$

➤ Loop path : $\quad$ I_1: 3 - 4 - 5 - 8 – 9
  $\quad\quad\quad\quad\quad$ I_2 : 5 - 6 - 7

$$G(I_1) = H_2\ H_4$$
$$G(I_2) = H_3$$

Check: all path touch (contain adder between 4 and 5)

➤ By Mason's rule: $\quad H = \dfrac{G(f_1) + G(f_2)}{1 - G(l_1) - G(l_2)} = \dfrac{H_5 H_3 + H_1 H_2 H_3}{1 - H_2 H_4 - H_3} = \dfrac{H_3(H_5 + H_1 H_2)}{1 - H_2 H_4 - H_3}$

2.5 Transfer Function G (s)

Continuous-time state space model

$$\dot{x}(t) = A\,x(t) + B\,u(t) \qquad \text{State equation}$$
$$y(t) = C\,x(t) + D\,u(t) \qquad \text{Measurement equation}$$

Transfer function describes input-output relation of system.

$$U(s) \longrightarrow \boxed{\textbf{System}} \longrightarrow Y(s)$$

$$s\,X(s) - x(0) = A\,X(s) + B\,U(s)$$

$$X(s) = (sI - A)^{-1}\,x(0) + (sI - A)^{-1}\,B\,U(s)$$
$$= \varphi(s)\,x(0) + \varphi(s)\,B\,U(s)$$

$$Y(s) = C\,X(s) + D\,U(s)$$
$$= C[\,(sI - A)^{-1}\,]x(0) + [c(sI - A)^{-1}\,B + D]U(s)$$
$$= C\,\varphi(s)\,x(0) + C\,\varphi(s)\,B\,U(s) + D\,U(s)$$

Transfer function $G(s)$ (pxr) (case: x(0)=0):

$$G(s) = C(sI - A)^{-1}\,B + D = C\,\varphi(s)\,B + D$$

2.5 Transfer Function

Transfer function of TESLA cavity including 8/9-pi mode

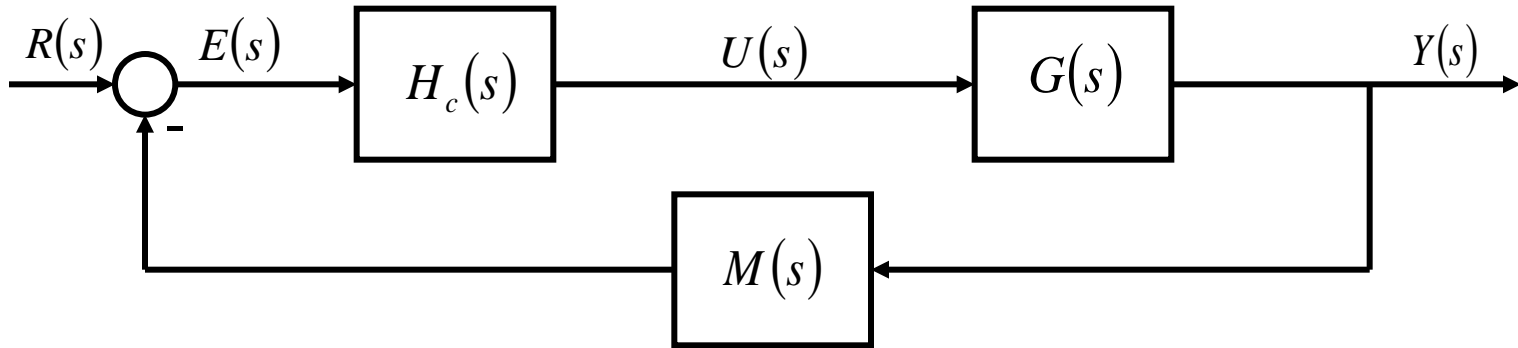$$H_{cont}(s) \approx H_{cav}(s) = H_\pi(s) + H_{\frac{8}{9}\pi}(s)$$

π − mode

$$H_\pi(s) = \frac{(\omega_{1/2})_\pi}{\Delta\omega_\pi^2 + (s + (\omega_{1/2})_\pi)^2} \begin{pmatrix} s + (\omega_{1/2})_\pi & -\Delta\omega_\pi \\ -\Delta\omega_\pi & s + (\omega_{1/2})_\pi \end{pmatrix}$$

$\dfrac{8}{9}\pi$ − mode

$$H_{\frac{8}{9}\pi}(s) = -\frac{(\omega_{1/2})_{\frac{8}{9}\pi}}{\Delta\omega_{\frac{8}{9}\pi}^2 + \left(s + (\omega_{1/2})_{\frac{8}{9}\pi}\right)^2} \begin{pmatrix} s + (\omega_{1/2})_{\frac{8}{9}\pi} & -\Delta\omega_{\frac{8}{9}\pi} \\ \Delta\omega_{\frac{8}{9}\pi} & s + (\omega_{1/2})_{\frac{8}{9}\pi} \end{pmatrix}$$

2.5 Transfer Function of a Closed Loop System

We can deduce for the output of the system.

$$Y(s) = G(s)\, U(s) = G(s)\, H_c(s)\, E(s)$$
$$= G(s)\, H_c(s)[R(s) - M(s)\, Y(s)]$$
$$= L(s)\, R(s) - L(s)\, M(s)\, Y(s)$$

With $L(s)$ the transfer function of the open loop system (controller plus plant).

$$(I + L(s)\, M(s))\, Y(s) = L(s)\, R(s)$$
$$Y(s) = (I + L(s)\, M(s))^{-1}\, L(s)\, R(s)$$
$$= T(s)\, R(s)$$

$T(s)$ is called : Reference Transfer Function

2.5 Sensitivity

The ratio of change in Transferfunction T(s) by the parameter b can be defined as:

System characteristics change with system parameter variations

$$S = \frac{\Delta T(s)}{T(s)} \frac{b}{\Delta b}$$

The sensitivity function is defined as:

$$S_b^T = \lim_{\Delta b \to 0} \frac{\Delta T(s)}{\Delta b} \frac{b}{T(s)} = \frac{\partial T(s)}{\partial b} \frac{b}{T(s)}$$

Or in General sensitivity function of a characteristics W with respect to the parameter b:

$$\boxed{S_b^W = \frac{\partial W}{\partial b} \frac{b}{W}}$$
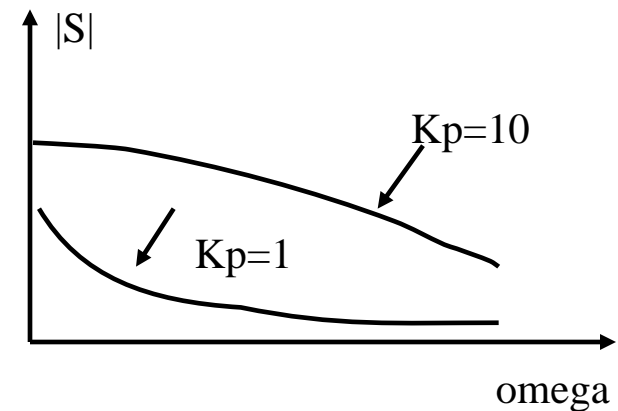
Example: plant with proportional feedback given by

$$G_c(s) = K_p \qquad G_p(s) = \frac{K}{s + 0.1}$$

Plant transfer function T(s):

$$T(s) \frac{K_p G_p(s)}{1 + K_p G_p(s) H_k}$$

$$S_H^T(j\omega) = \frac{-K_p G_p(j\omega) H_k}{1 + K_p G_p(j\omega) H_k} = \frac{-0.25 K_p}{0.1 + 0.25 K_p + j\omega}$$

Increase of H results in decrease of T
→ System cant be insensitive to both H,T

|S|

Kp=10

Kp=1

omega

2.5 Disturbance Rejection

Disturbances are system influences we do not control and want to minimize its impact on the system.

$$C(s) = \frac{G_c(s) \cdot G_p(s)}{1 + G_c(s) \cdot G_p(s) \cdot H(s)} R(s) + \frac{G_d(s)}{1 + G_c(s) \cdot G_p(s) \cdot H(s)} D(s)$$

$$= T(s) \cdot R(s) + T_d(s) \cdot D(s)$$

To Reject disturbances, make $T \cdot D(s)$ small!

- Using frequency response approach to investigate disturbance rejection
- In general $Td(j\omega)$ cant be small for all -ω Design $Td(j\omega)$ small for significant portion of system bandwidth

- Reduce the Gain $Gd(j\omega)$ between dist. Input and output
- Increase the loop gain $Gc(j\omega)Gp(j\omega)$ without increasing the gain $Gd(j\omega)$. Usually accomplished by the compensator $Gc(j\omega)$ choice
- Reduce the disturbance magnitude $d(t)$ should always be attempted if reasonable
- Use feed forward compensation, if disturbance can be measured.
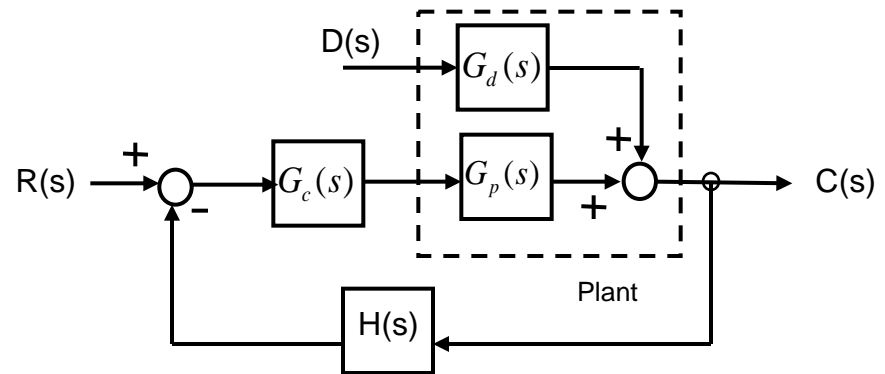
2.6 Stability

Now what do we know:
The impulse response tells us everything about the system response to any arbitrary
input signal *u(t)* .

what we have not learnt:
If we know the transfer function G(s), how can we deduce the systems behavior?
What can we say e.g. about the system stability?

Definition:

A linear time invariant system is called to be **BIBO** stable (Bounded-input-bounded-output)
For all bounded inputs $|u(t)| \le M_1$ (for all t) exists a boundary for the output signal M_2 ,
So that $|y(t)| \le M_2$. (for all t) with M_1 and M_2 , positive real numbers.

**Input never exceeds M_1 and output never exceeds M_2 , then we have BIBO
stability!**

Note: it has to be valid for ALL bounded input signals!

2.6 Stability

Example: $Y(s) = G(s)\, U(s)$, integrator $G(s) = \dfrac{1}{s}$

1.Case

$$u(t) = \delta(t), \quad U(s) = 1$$

$$\left| y(t) \right| = \left| L^{-1}[Y(s)] \right| = \left| L^{-1}\left[\frac{1}{s} \right] \right| = 1$$

The bounded input signal causes a bounded output signal.

2.Case

$$u(t) = 1, \quad U(s) = \frac{1}{s}$$

$$\left| y(t) \right| = \left| L^{-1}[Y(s)] \right| = \left| L^{-1}\left[\frac{1}{s^2} \right] \right| = t$$

BIBO-stability has to be shown/proved for any input. Is is not sufficient to show its validity for a single input signal!

2.6 Stability

Condition for BIBO stability:

We start from the input-output relation

$$Y(s) = G(s)\, U(s)$$

By means of the convolution theorem we get

$$\left| y(t) \right| = \left| \int_0^t g(\tau)\, u(t-\tau)\, d\tau \right| \leq \int_0^t \left| g(\tau) \right| \left| u(t-\tau) \right| d\tau \leq M_1 \int_0^\infty \left| g(\tau) \right| d\tau \leq M_2$$

Therefore it follows immediately:

If the impulse response is absolutely integrable

$$\int_0^\infty \left| g(t) \right| dt < \infty$$

Then the system is BIBO-stable.

2.7 Poles and Zeroes

Can stability be determined if we know the TF of a system?

$$G(s) = C\ \Phi(s)\ B + D = C\frac{[sI - A]_{adj}}{\chi(s)} B + D$$

Coefficients of Transfer function G(s) are rational functions in the complex variable**s**

$$g_{ij}(s) = \alpha \cdot \frac{\prod_{k=1}^{m}(s - z_k)}{\prod_{l=1}^{n}(s - p_l)} = \frac{N_{ij}(s)}{D_{ij}(s)}$$

z_k Zeroes, p_l Poles, α real constant, and it is $m \leq n$ (we assume common factors have already been canceled!)

What do we know about the zeros and the poles?

Since numerator $N(s)$ and denominator $D(s)$ are polynomials with real coefficients, Poles and zeroes must be real numbers or must arise as complex conjugated pairs!

2.7 Poles and Zeroes

Stability directly from state-space

$$Re\,call : H(s) = C(sI - A)^{-1}B + D$$

Assuming D=0 (D could change zeros but not poles)

$$H(s) = \frac{Cadj(sI - A)B}{det(sI - A)} = \frac{b(s)}{a(s)}$$

Assuming there are no common factors between the poly $Cadj(sI - A)B$ and $det(sI - A)$
i.e. no pole-zero cancellations (usually true, system called " minimal") then we can identify

and
$$b(s) = Cadj(sI - A)\,B$$

$$a(s) = det\,(sI - A)$$

i.e. poles are root of $det\,(sI - A)$

Let λ_i be the ith eigenvalue of A

$$if \quad Re\{\lambda_i\} \leq 0 \; forall \; i \Rightarrow \; \text{System stable}$$

So with computer, with eigenvalue solver, can determine system stability directly from coupling matrix A.

2.8 Stability Criteria

➢ A system is BIBO stable if, for every bounded input, the output remains bounded with Increasing time.

➢ For a LTI system, this definition requires that all poles of the closed-loop transfer-function (all roots of the system characteristic equation) lie in the left half of the complex plane.

Several methods are available for stability analysis:

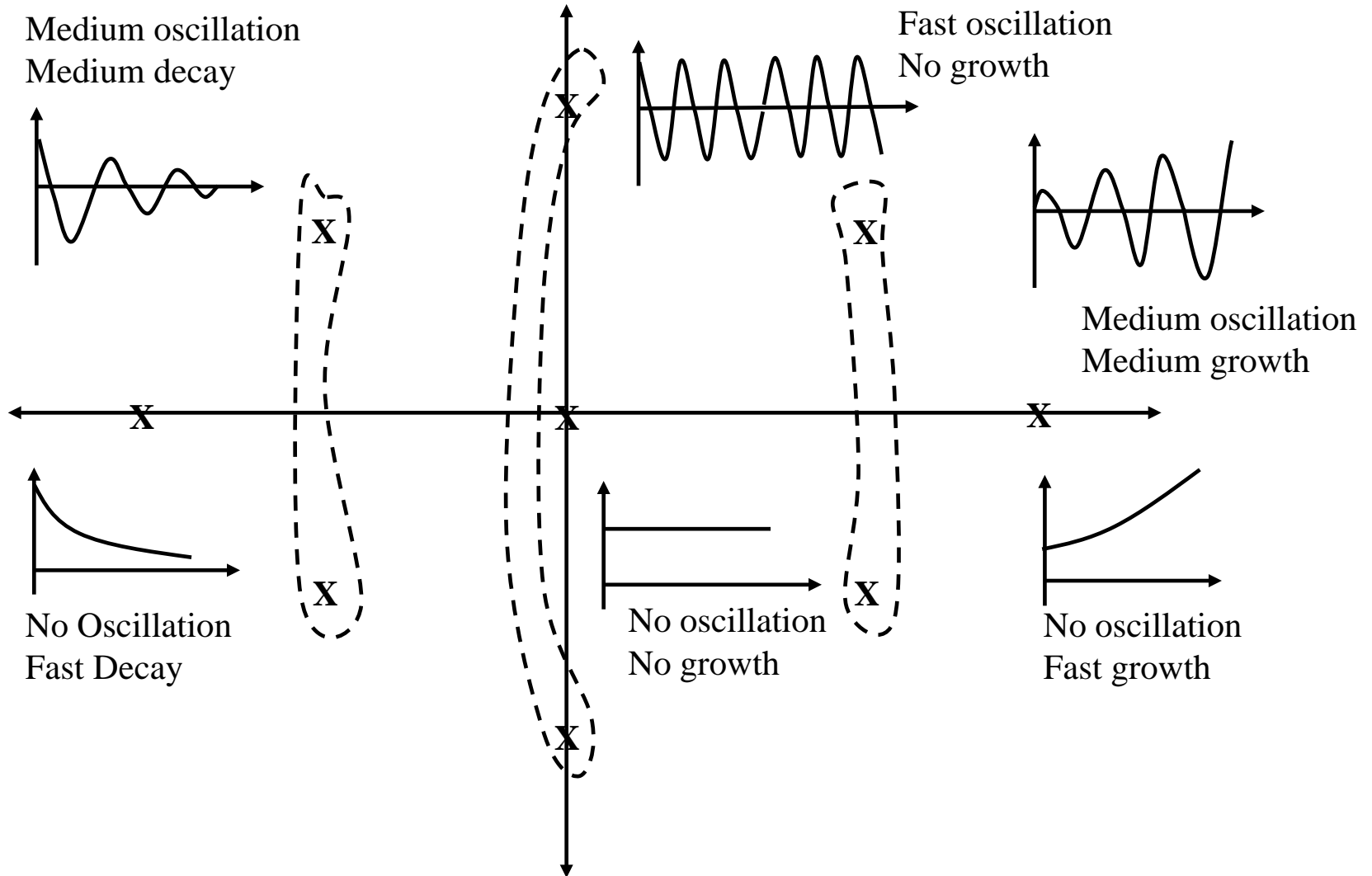1. Routh Hurwitz criterion

2. Calculation of exact locations of roots
   a. Root locus technique
   b. Nyquist criterion
   c. Bode plot

3. Simulation (only general procedures for nonlinear systems)

➢ While the first criterion proofs whether a feedback system is stable or unstable, the second Method also provides information about the setting time (damping term).

2.8 Poles and Zeroes

Pole locations tell us about impulse response i.e. also stability:

Medium oscillation
Medium decay

Fast oscillation
No growth

Medium oscillation
Medium growth

No Oscillation
Fast Decay

No oscillation
No growth

No oscillation
Fast growth

2.8 Poles and Zeroes

Furthermore: Keep in mind the following picture and facts!
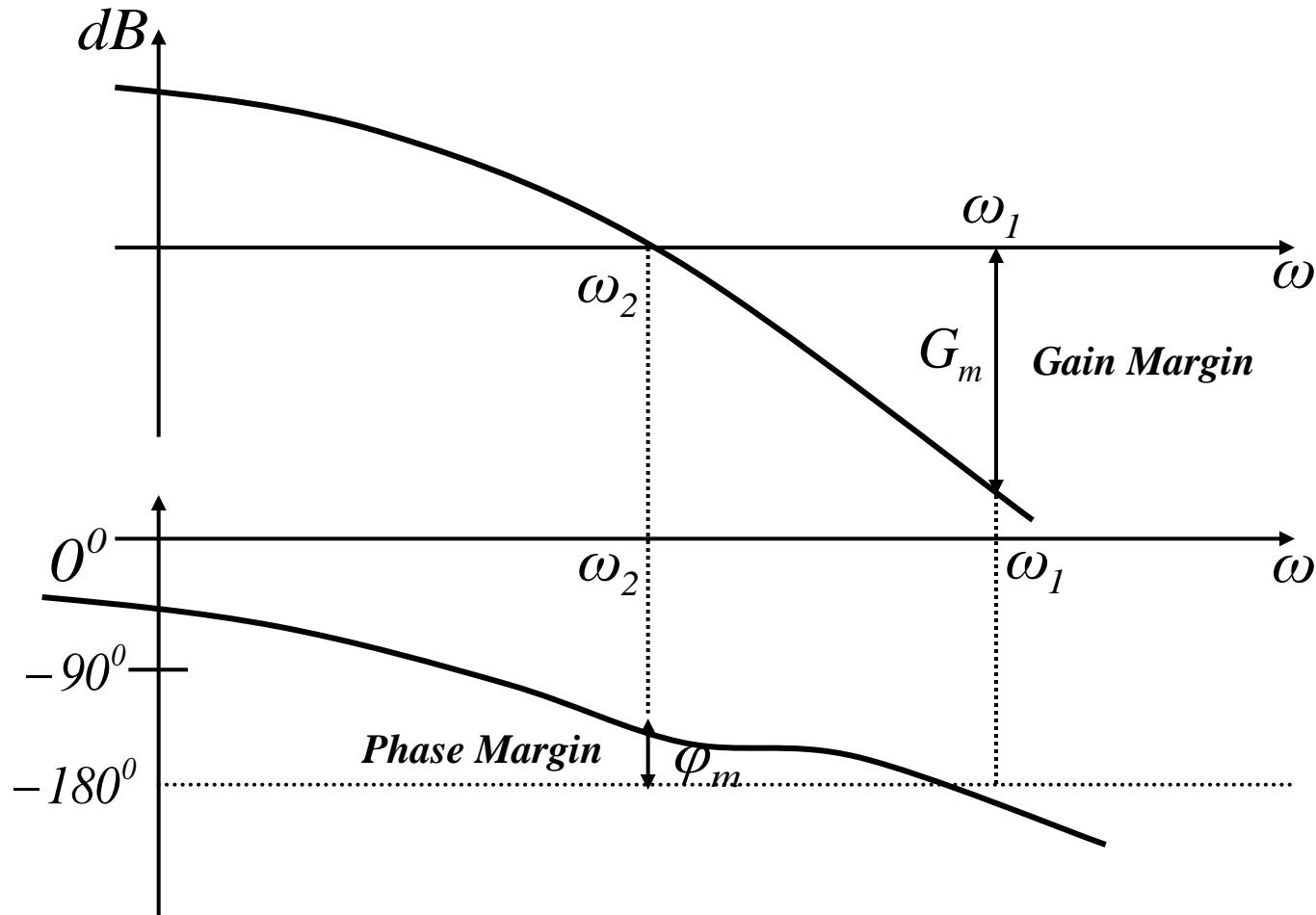
➢Complex pole pair: Oscillation with growth or decay.

➢Real pole: exponential growth or decay.

➢Poles are the Eigenvalues of the matrix A.

➢Position of zeros goes into the size of c_j....

✓ In general a complex root must have a corresponding conjugate root (N(s), D(S) polynomials with real coefficients.

2.8 Bode Diagram

The closed loop is stable if the phase of the unity crossover frequency of the OPEN LOOP
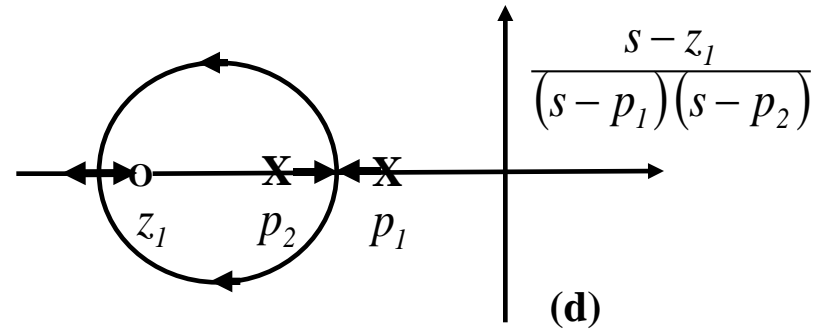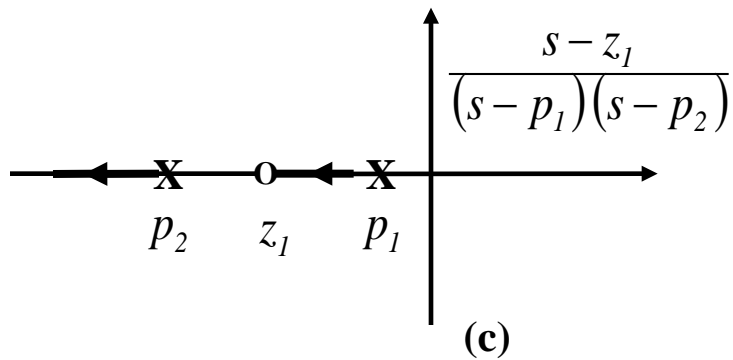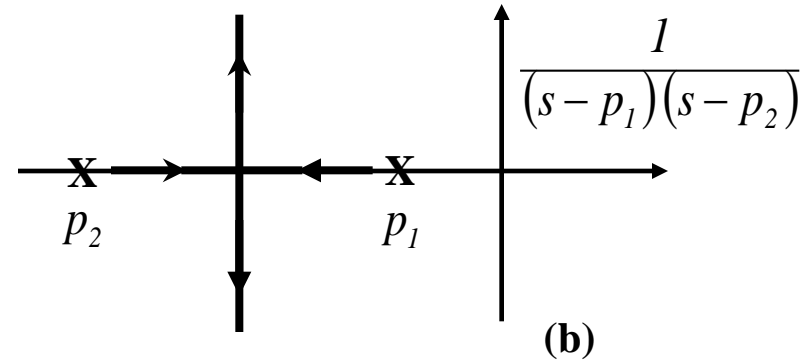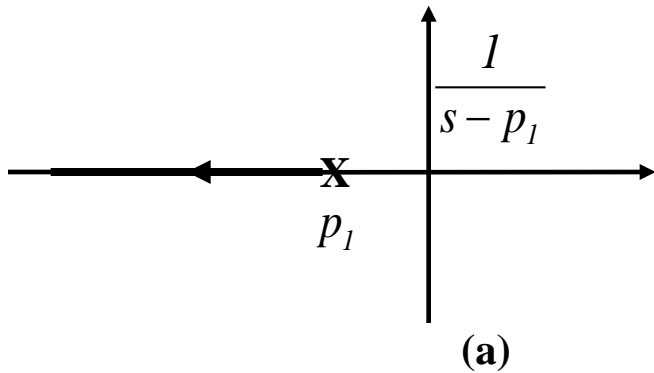Is larger than-180 degrees.

2.8 Root Locus Analysis

Definition: A root locus of a system is a plot of the roots of the system characteristic equation (the poles of the closed-loop transfer function) while some parameter of the system (usually the feedback gain) is varied.
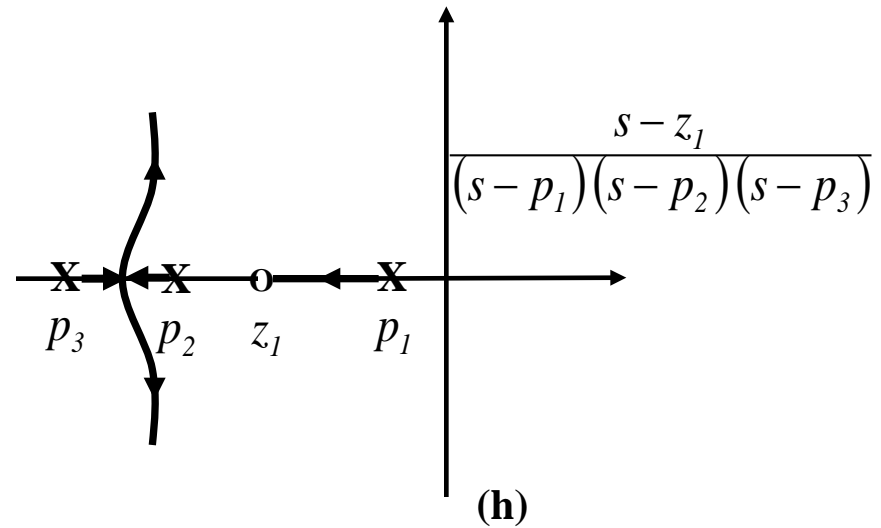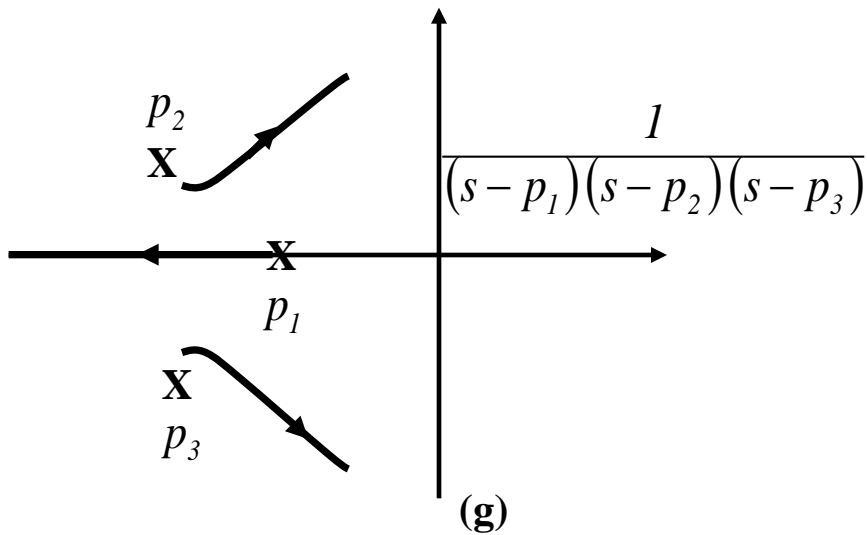

$$K\,H(s) = \frac{K}{(s - p_1)(s - p_2)(s - p_3)}$$

$$G_{CL}(s) = \frac{K\,H(s)}{1 + K\,H(s)} \; roots \; at \; 1 + K\,H(s) = 0.$$
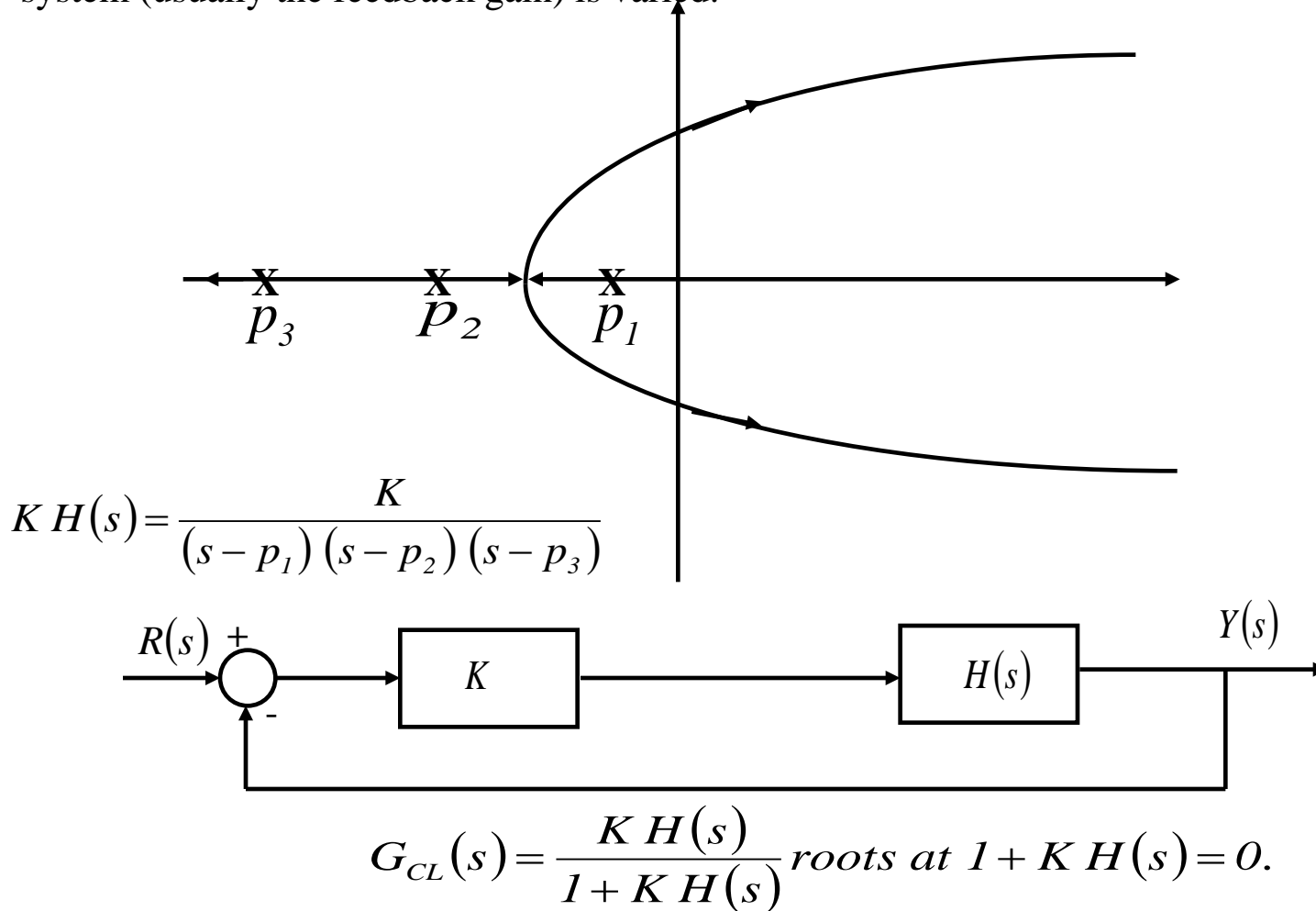
How do we move the poles by varying the constant gain K?

41

2.8 Root Locus Examples

$$\frac{1}{s-p_1}$$

(a)

$$\frac{1}{(s-p_1)(s-p_2)}$$

(b)

$$\frac{s-z_1}{(s-p_1)(s-p_2)}$$

(c)

$$\frac{s-z_1}{(s-p_1)(s-p_2)}$$

(d)

2.8 Root Locus Examples (Cnt'd)

$$\frac{1}{(s-p_1)(s-p_2)(s-p_3)}$$

X p_3 **X** p_2 **X** p_1

(e)

$$\frac{1}{(s-p_1)(s-p_2)(s-p_3)}$$

p_2 **X**

X p_3

X p_1

(f)

$$\frac{1}{(s-p_1)(s-p_2)(s-p_3)}$$

p_2 **X**

X p_1

X p_3

(g)

$$\frac{s-z_1}{(s-p_1)(s-p_2)(s-p_3)}$$

X p_3 **X** p_2 **o** z_1 **X** p_1

(h)
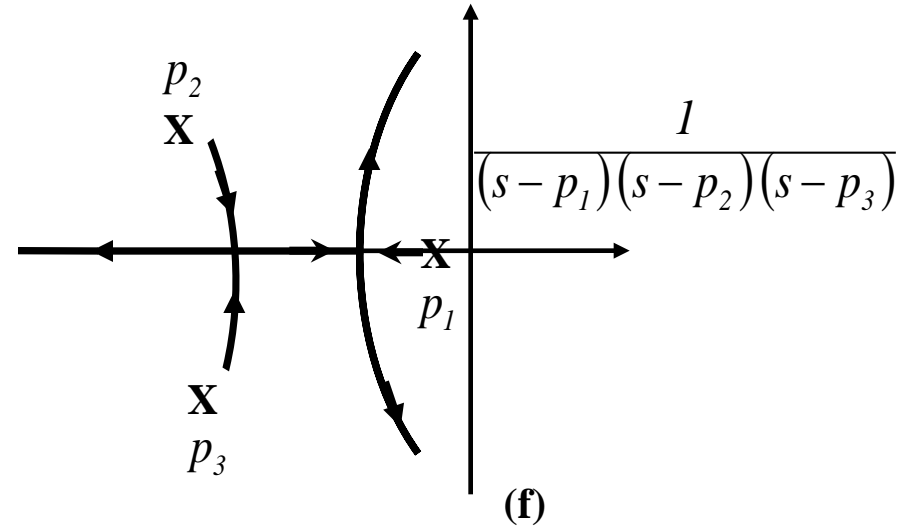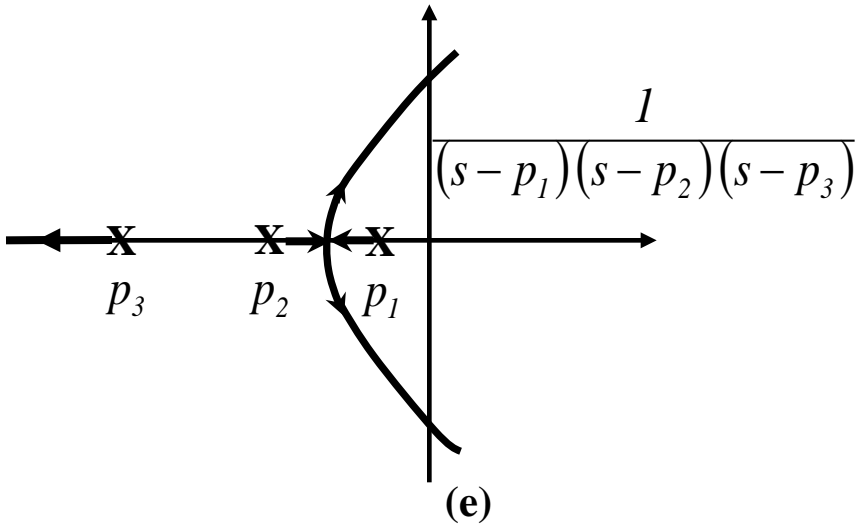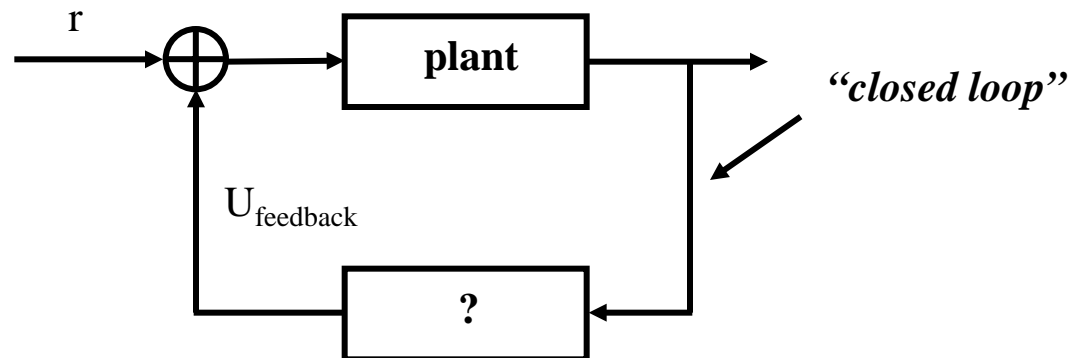
3.Feedback

The idea:
Suppose we have a system or "plant"
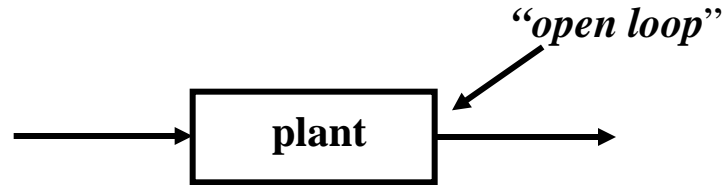
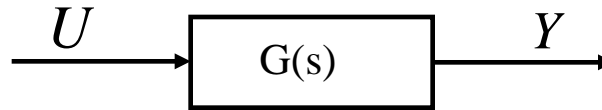We want to improve some aspect of plant's performance by observing the output and applying a appropriate "correction" signal. ***This is feedback***

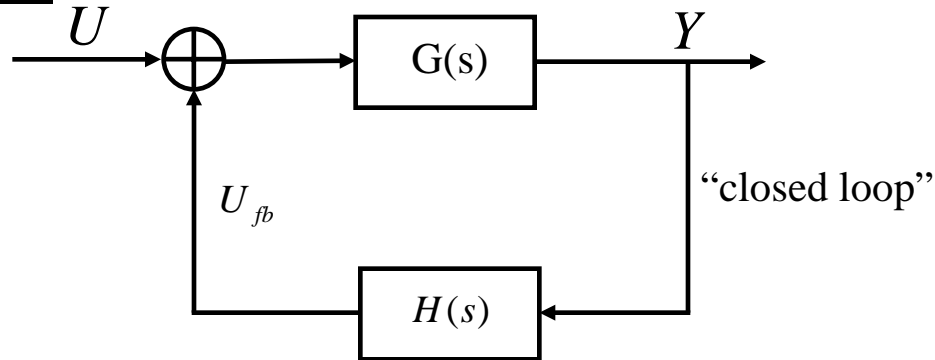***Question: What should this be?***

3.Feedback

Open loop gain:

$$G^{O.L}(s) = G(s) = \left(\frac{u}{y}\right)^{-1}$$

Closed-loop gain:

"closed loop"

$$G^{C.L}(s) = \frac{G(s)}{1 + G(s)\,H(s)}$$

$Proof:\ y = G(u - u_{fb})$

$\qquad\qquad = G\,u - G\,u_{fb} \qquad\qquad \Rightarrow y + G\,H_y = G\,u$

$\qquad\qquad = G\,u - G\,Hy \qquad\qquad \Rightarrow \dfrac{y}{u} = \dfrac{G}{(1 + G\,H)}$

3.1 Feedback-Example 1

Consider S.H.O with feedback proportional to x i.e.:

Where
$$\ddot{x} + \gamma\,\dot{x} + \omega_n^2 x = u + u_{fb}$$

$$u_{fb}(t) = -\alpha\,x\,(t)$$
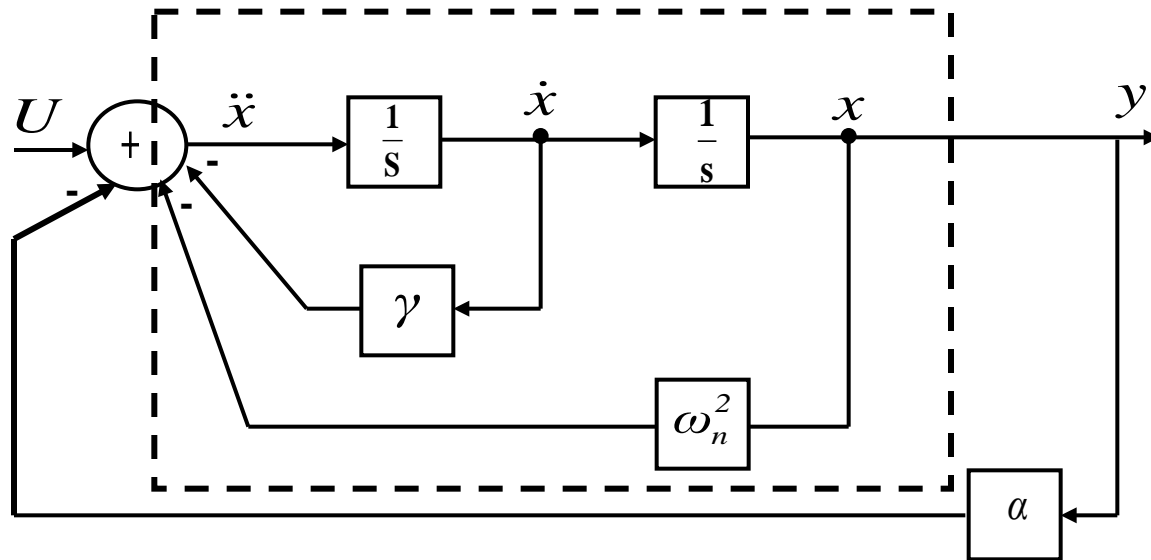
Then
$$\ddot{x} + \gamma\,\dot{x} + \omega_n^2 x = u - \alpha\,x$$

$$\Longrightarrow \ddot{x} + \gamma\,\dot{x} + \left(\omega_n^2 + \alpha\right)x = u$$

Same as before, except that new "natural" frequency $\omega_n^2 + \alpha$

Stefan Simrock & Mariusz Grecki,
5th LC School, Switzerland 2010

46

3.1 Feedback-Example 1

Now the closed loop T.F. is:

$$G^{C.L.}(s) = \frac{1}{s^2 + \gamma s + (\omega_n^2 + \alpha)}$$

DC response: s=0
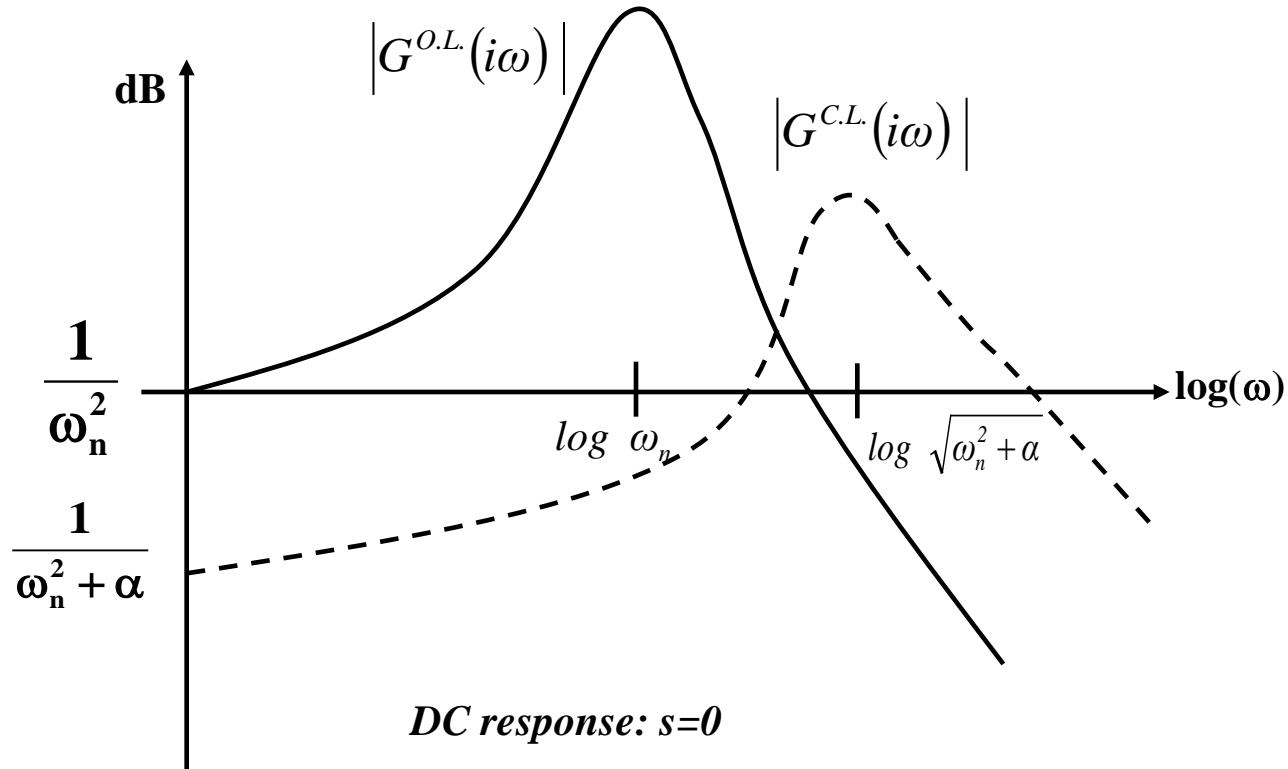
So the effect of the proportional feedback in this case is *to increase the bandwidth of the system*
(and reduce gain slightly, but this can easily be compensated by adding a constant gain in front…)

3.1 Feedback-Example 2

In S.H.O. suppose we use *integral feedback*:

$$u_{fb}(t) = -\alpha \int_0^t x(\tau)\, d\tau$$

$$i.e \quad \ddot{x} + \gamma\,\dot{x} + \omega_n^2\, x = u - \alpha \int_0^t x(\tau)\, d\tau$$
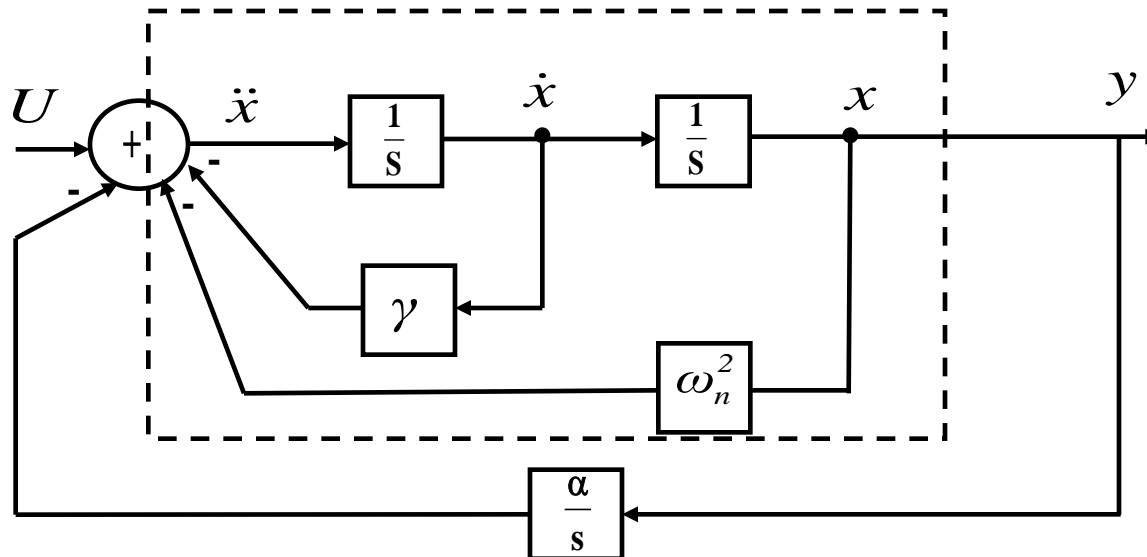
Differentiating once more yields: $\quad \dddot{x} + \gamma\,\ddot{x} + \omega_n^2\,\dot{x} + \alpha\, x = \dot{u}$

No longer just simple S.H.O., add another state

3.1 Feedback-Example 2

$$G^{C.L.}(s) = \frac{\dfrac{1}{s^2 + \gamma s + \omega_n^2}}{1 + \left(\dfrac{\alpha}{s}\right)\left(\dfrac{1}{s^2 + \gamma s + \left(\omega_n^2 + \alpha\right)}\right)}$$
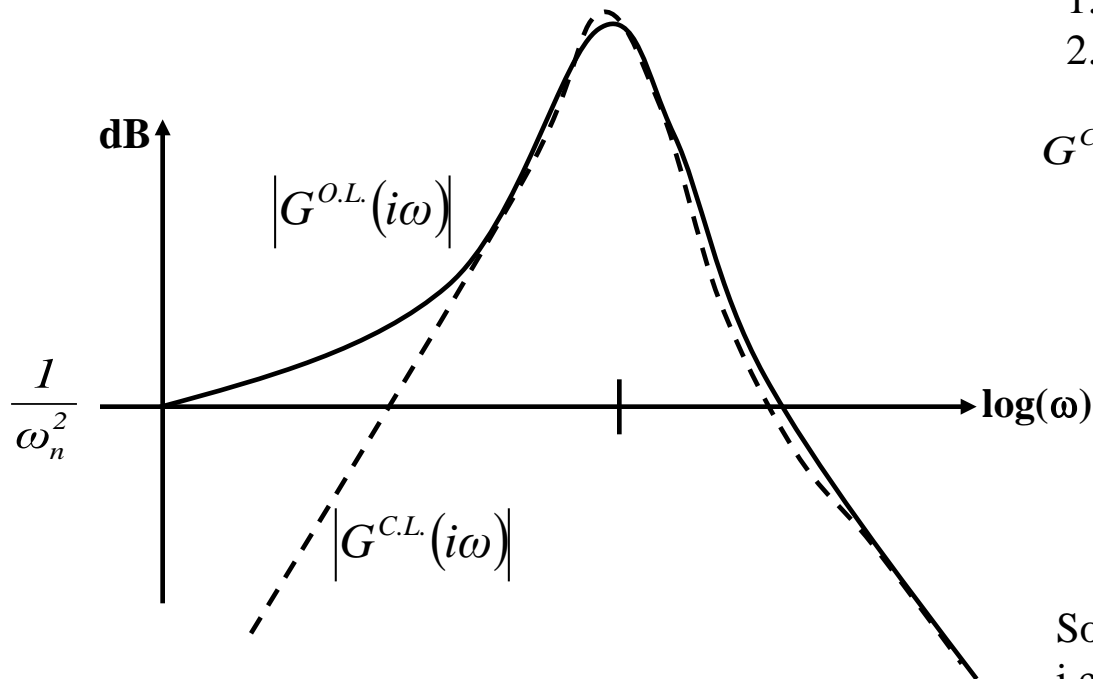
$$= \frac{s}{s\left(s^2 + \gamma s + \omega_n^2\right) + \alpha}$$

Observe that
1. $G^{C.L.}(0 = 0)$
2. For large s (and hence for large ω)

$$G^{C.L.}(s) \approx \frac{1}{\left(s^2 + \gamma\, s + \omega_n^2\right)} \approx G^{O.L.}(s)$$

So integral feedback has killed DC gain
i.e system _rejects constant_ disturbances

Stefan Simrock & Mariusz Grecki,
5th LC School, Switzerland 2010

49

3.1 Feedback-Example 3

Suppose S.H.O now apply <u>differential feedback</u> i.e.

$$u_{fb}(t) = -\alpha\, \dot{x}(t)$$
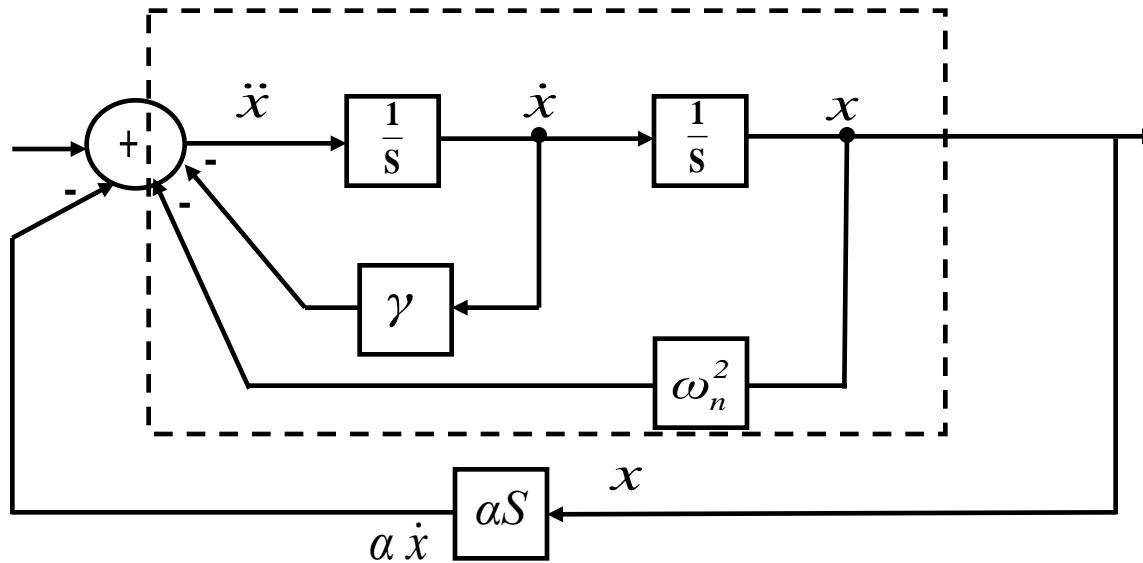
Now have

$$\ddot{x} + (\gamma + \alpha)\, \dot{x} + \omega_n^2 x = u$$

So effect off differential feedback is to increase damping

3.1 Feedback-Example 3

Now
$$G^{C.L.}(s) = \frac{1}{s^2 + (\gamma + \alpha)\, s + \omega_n^2}$$

So the effect of differential feedback here is to "flatten the resonance" i.e. ___*damping is increased*___.

Note: Differentiators can never be built exactly, only approximately.
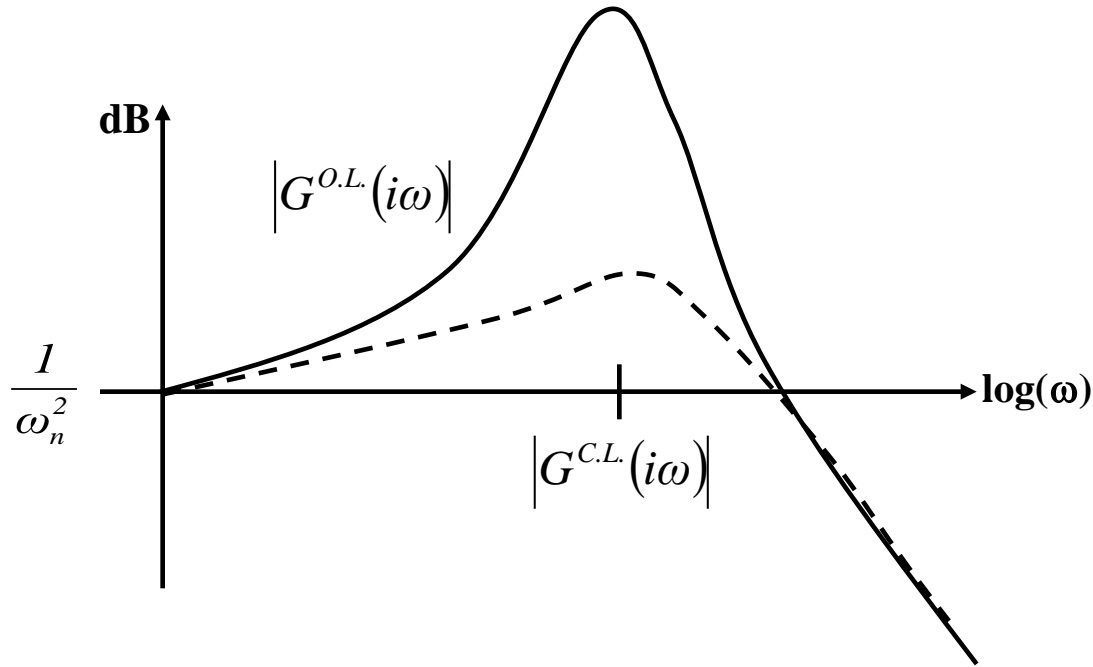
3.1 PID controller

(1) The latter 3 examples of feedback can all be combined to form a
    <u>P.I.D. controller</u> (prop.-integral-diff).
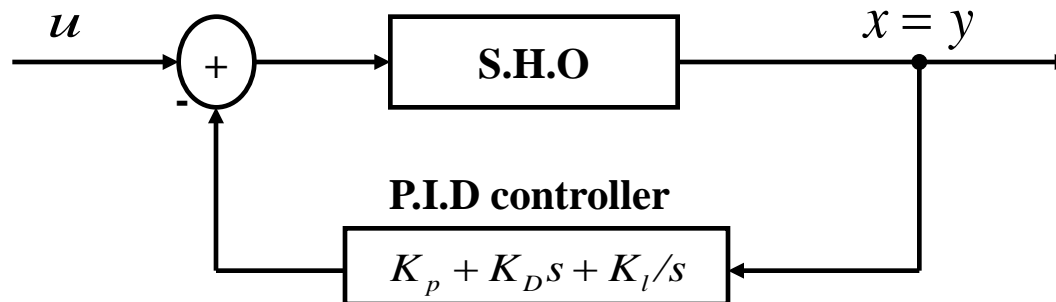
$$u_{fb} = u_p + u_d + u_l$$

(2) In example above S.H.O. was a very simple system and it was clear what
    physical interpretation of P. or I. or D. did. But for *large complex systems* not
    obvious

➔ **Require arbitrary " tweaking"**

That's what we're trying to avoid

3.1 PID controller

For example, if you are so smart let's see you do this with your P.I.D. controller:

6th order system
3 resonant poles
3 complex pairs
6 poles
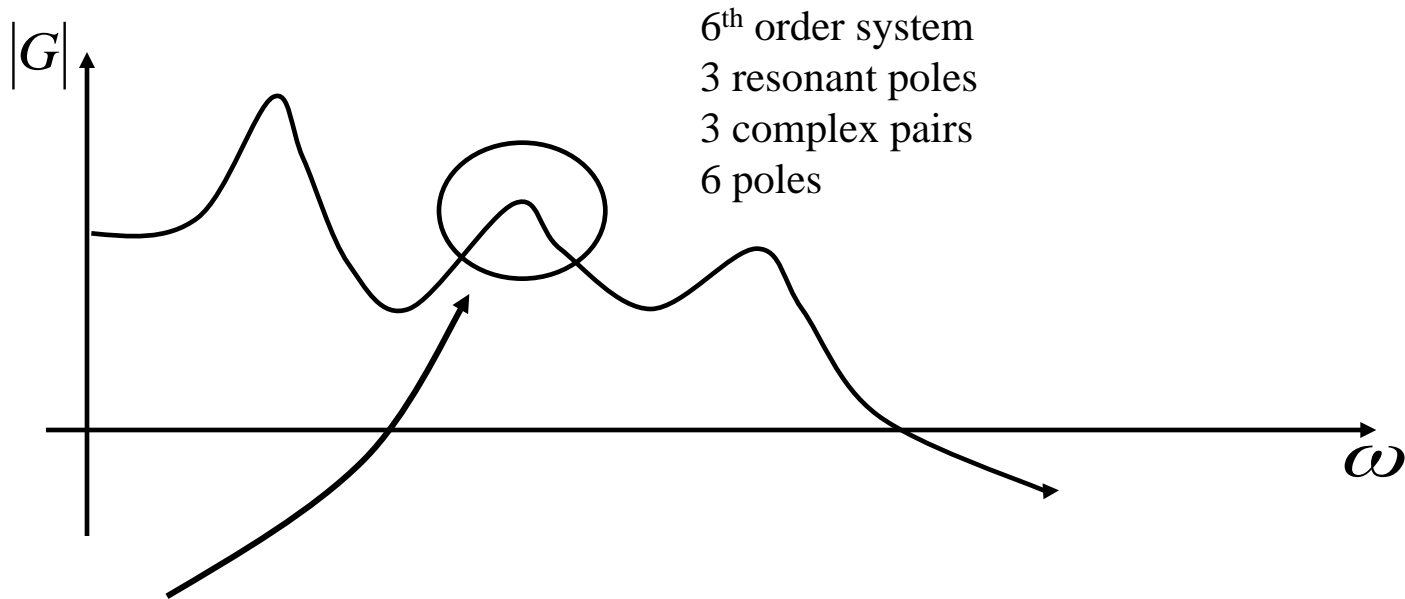
Damp this mode, but leave the other two modes undamped, just as they are.

This could turn out to be a tweaking nightmare that'll get you nowhere fast!

We'll see how this problem can be solved easily.

3.2 Full State Control

Suppose we have system

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

Since the state vector x(t) contains all current information about the system the most general feedback makes use of **all** the state info.

$$u_{fb} = -k_1 x_1 - \ldots - k_n x_n$$
$$= -k\,x$$

Where $\quad k = \begin{bmatrix} k_1 \ldots k_n \end{bmatrix} \quad$ (row matrix)

Where example: In S.H.O. examples

Proportional fbk : $\quad u_p = -k_p x = - \begin{bmatrix} k_p & 0 \end{bmatrix}$

Differential fbk : $\quad u_d = -k_d \dot{x} = - \begin{bmatrix} 0 & k_d \end{bmatrix}$

3.2 Full State Control

Theorem: If there are no poles cancellations in

$$G_{O.L.}(s) = \frac{b(s)}{a(s)} = C(sI - A)^{-1} B$$

Then can move eigen values of $A - BK$ anywhere we want using full state feedback.

Proof:

Given <u>any</u> system as L.O.D.E. or state space it can be written as:

$$\begin{bmatrix} x_1 \\ \dots \\ \dots \\ x_n \end{bmatrix} = \overbrace{\begin{bmatrix} 0 & 1 & \dots & 0 \\ 0 & \dots & \dots & \dots \\ 0 & \dots & \dots & 1 \\ -a_0 & \dots & \dots & -a_{n-1} \end{bmatrix}}^{A^{O.L.}} \begin{bmatrix} x_1 \\ \dots \\ \dots \\ x_n \end{bmatrix} + \overbrace{\begin{bmatrix} 0 \\ 0 \\ \dots \\ 1 \end{bmatrix}}^{B} u$$

$$y = \begin{bmatrix} b_0 & \dots & \dots & b_{n-1} \end{bmatrix} \begin{bmatrix} x_1 \\ \dots \\ \dots \\ x_n \end{bmatrix}$$

Where

$$G^{O.L.} = C(sI - A)^{-1} B = \frac{b_{n-1}s^{n-1} + \dots + b_0}{s^n + a_{n-1}s^{n-1} + \dots + a_0}$$

3.2 Full State Control

i.e. first row of $A^{O.L.}$ Gives the coefficients of the denominator

$$a^{O.L.}(s) = det(sI - A^{O.L.}) = s^n + a_{n-1}s^{n-1} + \ldots + a_0$$

Now

$$A^{C.L.} = A^{O.L.} - BK$$

$$= \begin{bmatrix} 0 & 1 & \ldots & 0 \\ 0 & \ldots & \ldots & \ldots \\ 0 & \ldots & \ldots & 1 \\ -a_0 & \ldots & \ldots & -a_{n-1} \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ \ldots \\ 1 \end{bmatrix} \begin{bmatrix} k_0 & \ldots & \ldots & k_{n-1} \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 1 & \ldots & 0 \\ 0 & \ldots & \ldots & \ldots \\ 0 & \ldots & \ldots & 1 \\ -(a_0 + k_0) & \ldots & \ldots & -(a_{n-1} + k_{n-1}) \end{bmatrix}$$

So closed loop denominator

$$a^{C.L.}(s) = det(sI - A^{C.L.})$$
$$= s^n + (a_0 + k_0)s^{n-1} + \ldots + (a_{n-1} + k_{n-1})$$

Using $u = -Kx$ have direct control over every closed-loop denominator coefficient

➔ can place root anywhere we want in s-plane.

3.2 Full State Control

Example: Detailed block diagram of S.H.O with full-scale feedback
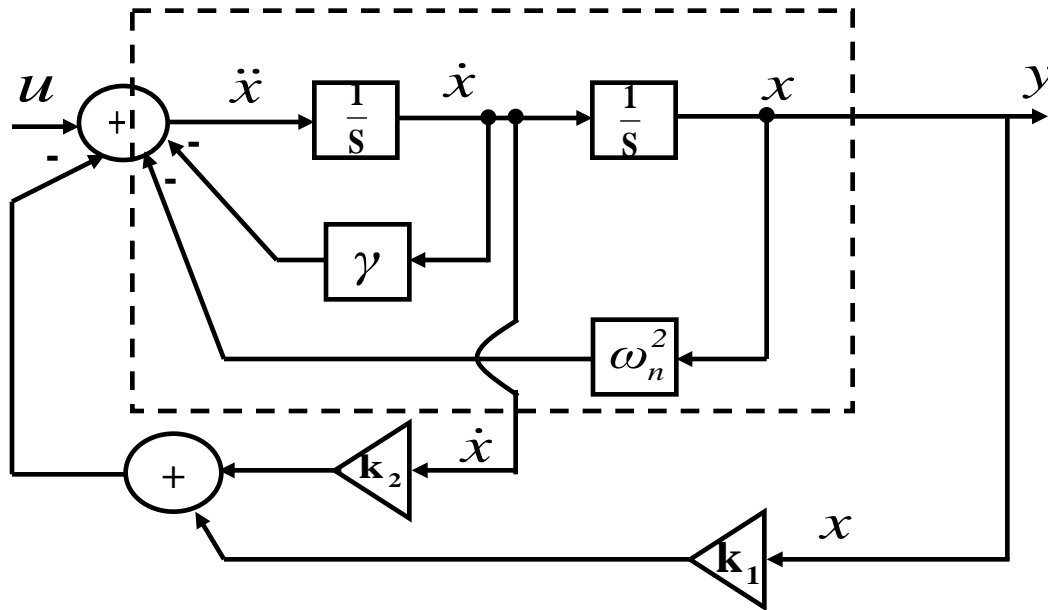
Of course this <u>assumes</u> we have access to the $\dot{x}$ *state, which we actually <u>Don't</u> in practice.*

However, let's ignore that " minor" practical detail for now.
(Kalman filter will show us how to get $\dot{x}$ from x).

3.2 Full State Control

With full state feedback have (assume D=0)

So
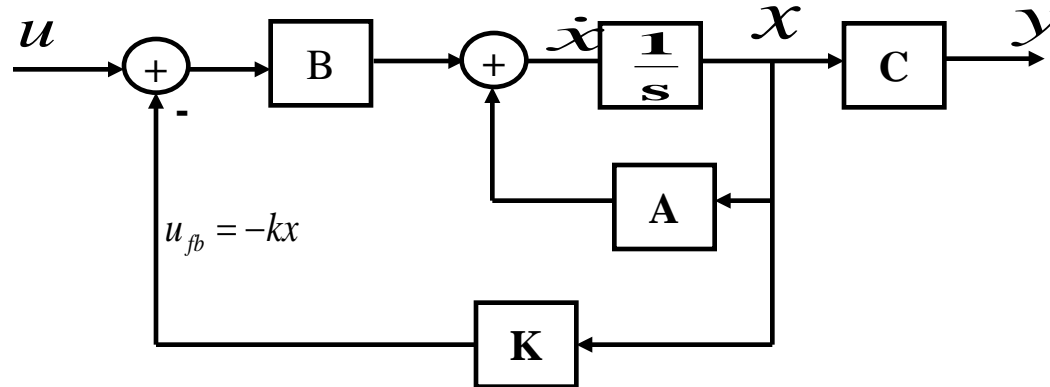
$$\dot{x} = A\,x + B[u + u_{fb}]$$

$$= Ax + Bu + BKu_{fb}$$

$$\dot{x} = (A - BK)\,x + B\,u$$

$$u_{fb} = -Kx$$

$$y = Cx$$

With full state feedback, get new closed loop matrix

$$A^{C.L.} = \left(A^{O.L.} - BK\right)$$

Now all stability info is now given by the eigen values of new A matrix

3.3 Controllability and Observability

The linear time-invariant system

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

Is said to be controllable if it is possible to find some input u(t) that will transfer the initial state x(0) to the origin of state-space, $x(t_0) = 0, with \ t_0 \ finite$

The solution of the state equation is:

$$x(t) = \varphi(t)x(0) + \int_0^t \varphi(\tau)B \ u(t - \tau) \ d\tau$$

For the system to be controllable, a function u(t) must exist that satisfies the equation:

$$0 = \varphi(t_0)x(0) + \int_0^{t_0} \varphi(\tau)Bu(t_0 - \tau) \ d\tau$$

With t_O finite. It can be shown that this condition is satisfied if the controllability matrix

$$C_M = [B \ \ AB \ \ A^2B \ ... \ A^{n-1}B]$$

Has inverse. This is equivalent to the matrix C_M having full rank (rank n for an n- th order differential equation).

3.3 Controllability and Observability

Observable:

➢ The linear time-invariant system is said to be observable if the initial conditions x(0) Can be determined from the output function y(t), $0 \leq t \leq t_1$ where t_1 is finite. With

$$y(t) = Cx = C\ \varphi(t)x0 + C\int_0^t \varphi(\tau)Bu(t - \tau)\ d\tau$$

➢ The system is observable if this equation can be solved for x(0). It can be shown that the system is observable if the matrix:
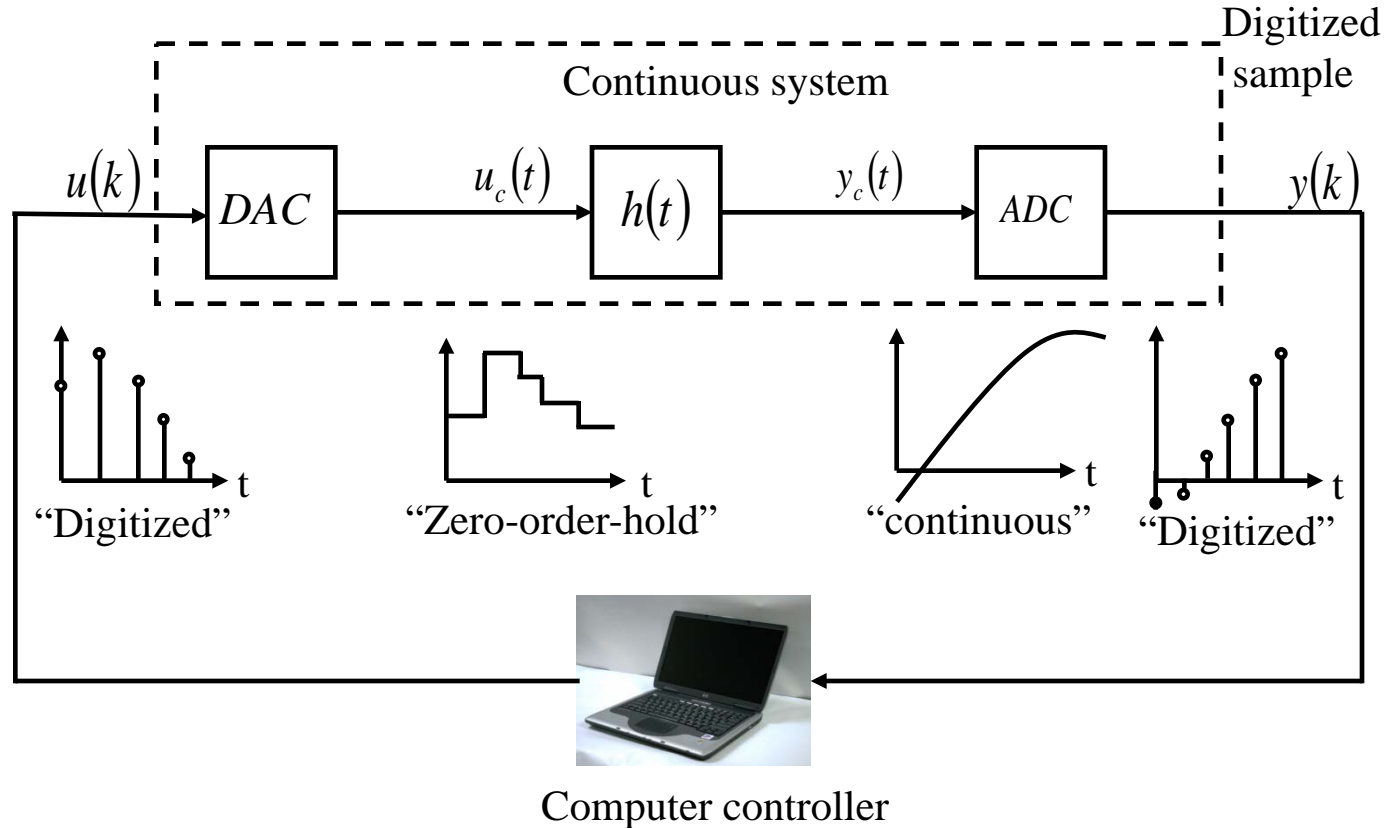
$$O_M = \begin{bmatrix} C \\ CA \\ ... \\ CA^{n-1} \end{bmatrix}$$

➢ Has inverse. This is equivalent to the matrix O_M having full rank (rank n for an n-th Order differential equation).

4.Discrete Systems

Where do discrete systems arise?

Typical control engineering example:

Computer controller

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

4. Discrete Systems

Then u(t) is given by:
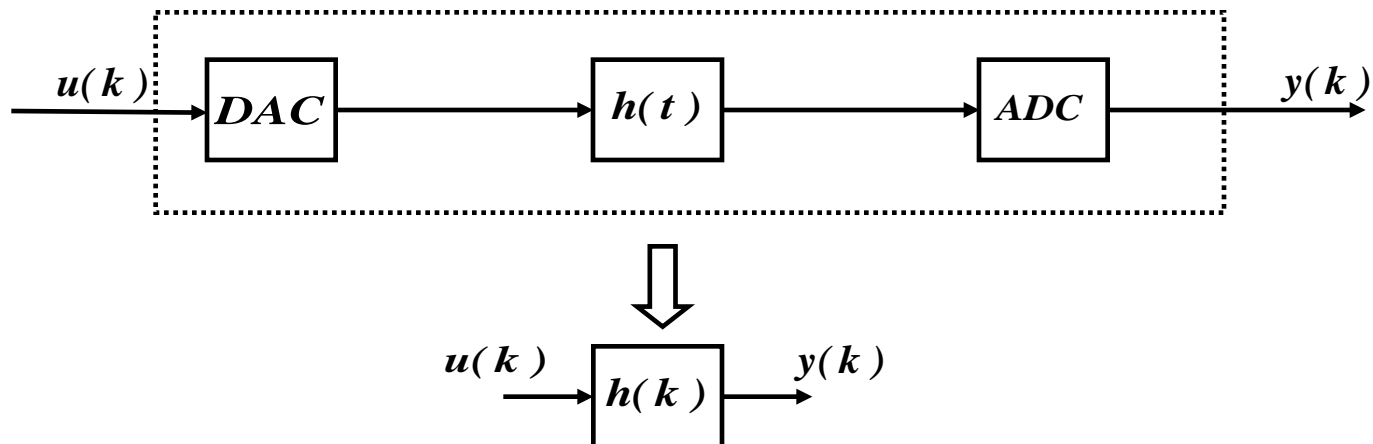
$$u(k) \equiv u_c(t); \quad kT \le t < (k+1)T$$
$$y(k) \equiv y_c(kT); \quad k = 0,1,2,...$$

Suppose: time continuous system is given by state-Space

$$\dot{x}_c(t) = A\, x_c(t) + B\, u_c(t); \quad x_c(0) = x_0$$
$$y_c(t) = C\, x_c(t) + D\, u_c(t)$$

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

4. Discrete Systems

Yes! We can obtain equivalent discrete system.

Recall
$$x_c(t) = e^{At}x_c(0) + \int_0^t e^{A\tau}.Bu_c(t-\tau)\, d\tau$$

From this
$$x_c(kT+T) = e^{AT}x_c(kT) + \int_0^T e^{A\tau}.Bu_c(kT-\tau)\, d\tau$$

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

i.e. can pull out of integral.

$$\Longrightarrow\ x_c(kT+T) = e^{At}x_c(kT) + \left(\int_0^t e^{A\tau}\cdot B\, d\tau\right)u_c(kT)$$

$$x(k+1) = A_d x(k) + B_d u(k)$$
$$y(k) = C_d x(k) + D_d u(k)$$
$$x(0) = x_c(0)$$

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

So we have an exact (note: $x(k+1) = x(k) + \dot{x}(k)\, T + O(.)$) 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

$$y(k+n) + a_{n-1} y(k+n-1) + ... + a_1 y(k+1) + a_0 y(k) = b_m u(k+m) + ... + b_1 u(k+1) + b_0 u(k)$$

$n \geq m;$ Assumes initial values $\quad y(n\text{-}1), \ ...., \ y(1), y(0) = 0.$

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

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

It follows the input-output relation:

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

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

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

Once again:

$$if \ U(z) = 1, \ \left(u(k) = \delta(k) \right), \ then \ Y(z) = G(z).$$

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

4.1 z-Transform of Discrete State Space Equation

$$x(k+1) = A_d\, x(k) + B_d\, u(k)$$
$$y(k) = C\, x(k) + D\, u(k)$$

Applying z-Transform on first equation:

$$z \cdot X(z) - z\, x(0) = A_d\, X(z) + B_d\, U(z)$$
$$(zI\text{-}A_d)\, X(z) = z\, x(0) + B\, U(z\ )$$

$$X(z) = (zI\text{-}A_d)^{-1} z\, x(0) + (zI - A_d)^{-1} B\, U(z)$$

Homogeneous solution

NOW: Particular solution

$$Y(z) = CX(z) + D\, U(z)$$
$$= C\,(zI\text{-}A_d)^{-1} z\, x(0) + \left(C(zI - A_d)^{-1} B + D\right) U(z)$$

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

$$Y(z) = G(z)\, U(z\ )\ with$$
$$G(z) = C(zI\text{-}A_d)^{-1} B + D$$

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

4.3 Stability (z-domain)

A discrete LTI system is <u>BIBO stable</u> if

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

Condition for BIBO stability:

$$| y(k) | = \left| \sum_{0}^{k} u(k-i) \, h(i) \right| \leq \sum_{0}^{k} |u(k-i)| \, |h(i)| \leq M \sum_{0}^{k} | h(i) | \leq M \sum_{0}^{\infty} | h(i) |$$

$$\therefore \sum_{0}^{\infty} |h(i)| < \infty \quad \blacktriangleright \quad \text{BIBO stable.}$$
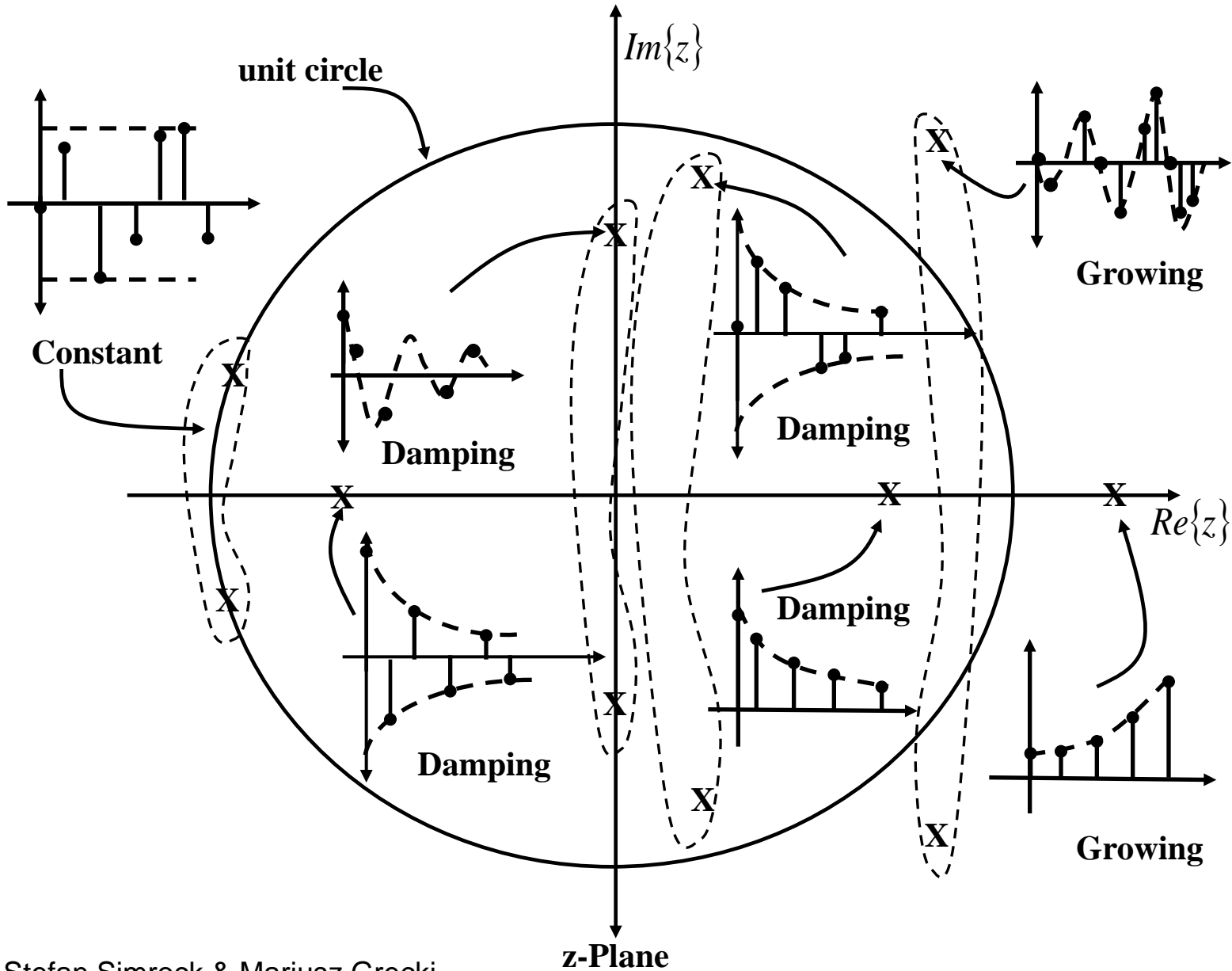
For L.O.D.E State space system:

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

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

Stefan Simrock & Mariusz Grecki,
5th LC School, Switzerland 2010

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

$$i.e \left| p_i \right| \leq 1 \quad ➔ \underline{\textbf{system stable}}$$

4.3 stability (z-domain)

Stability directly from State Space:

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

$$H(z) = \frac{b(z)}{a(z)} = C(zI - A_d)^{-1} B_d$$

$$= \frac{Cadj(zI - A_d)B_d}{det(zI - A_d)}$$

$$b(z) = Cadj(zI - A_d)B_d$$
$$a(z) = det(zI - A_d)$$

➔ 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 ith e-value of A_d .

4.4 Discrete Cavity Model

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

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

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

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

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

$$\cdot\left\{\left(\left(z - e^{\omega 12 T_S}\cdot\cos(\Delta\omega T_s)\right)\cdot\begin{bmatrix} \omega_{12} & -\Delta \\ \Delta\omega & \omega_{12} \end{bmatrix}\right) - e^{\omega 12 T_S}\cdot\sin(\Delta\omega T_s)\cdot\begin{bmatrix} \Delta\omega & \omega_{12} \\ -\omega_{12} & \Delta \end{bmatrix}\right\}$$
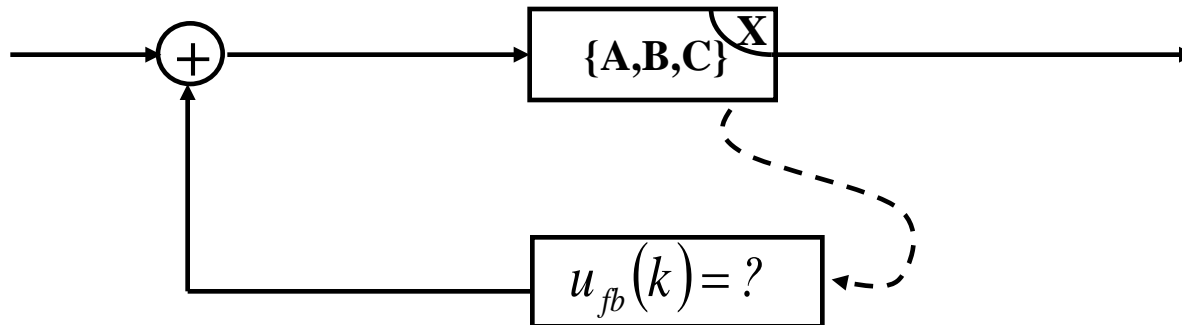
4.5 Linear Quadratic Regulator

Given:
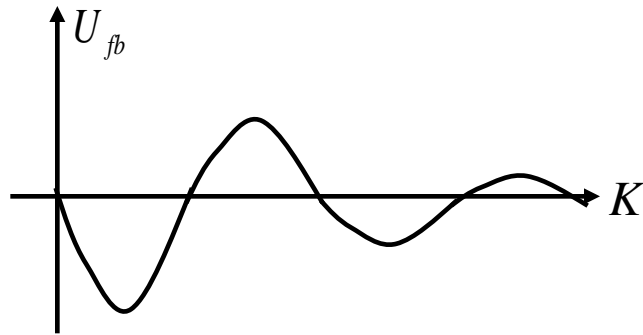$$x(k+1) = A\,x(k) + B\,u(k)$$
$$z(k) = C\,x(x)$$

(Assume D=0 for simplicity)
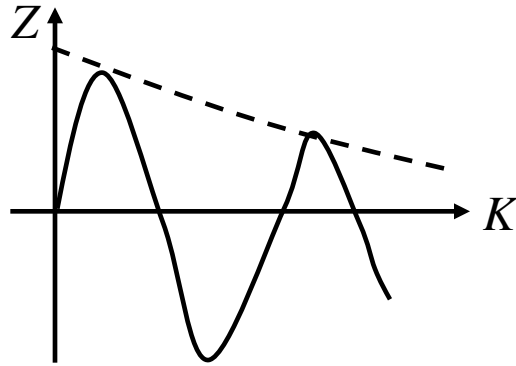
Suppose the system is <u>unstable</u> or almost unstable. We want to find $u_{fb}(k)$ which will bring x(k) to Zero, <u>quickly</u>, from any Initial condition.
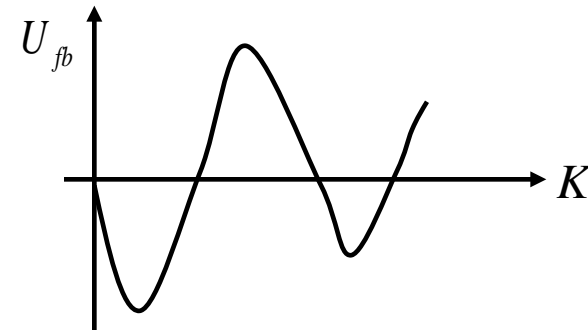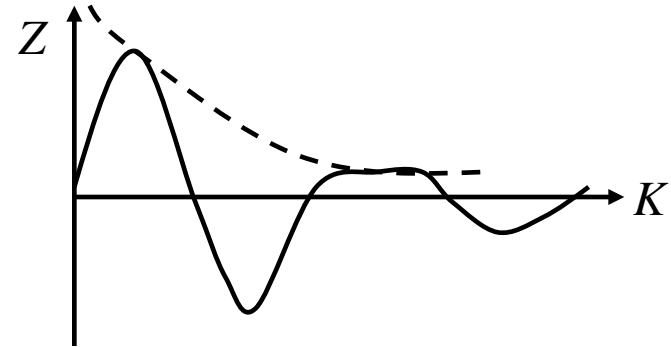
i.e.

4.5 Trade Off

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

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

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

(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:

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

$$f(x) = f(x_1, x_2)$$

$$= ax_1^2 + bx_1x_2 + cx_1 + dx_2^2$$

$$= [x_1 \ x_2] \underbrace{\begin{bmatrix} a & \frac{1}{2}b \\ \frac{1}{2}b & d \end{bmatrix}}_{Q} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \underbrace{[c \ 0]}_{P^T} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

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

4.5 Quadratic Cost for Regulator

What do we mean by "bad" damping and "cheap" control? We now define <u>precisely</u> 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)

<u>Given</u>: $x_{i+1} = Ax_i + Bu_i;$ x_0 given:

<u>Find</u> control sequence $\{u_0, u_1, u_2, ...\}$ such that

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

$$= \text{minimum}$$

Answer:

The optimal control sequence is a <u>*state feedback*</u> sequence $\{u_i\}_0^{\infty}$

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

$$K_{opt} = \left(R + B^T SB \right)^{-1} B^T SA$$

$$S = A^T SA + Q - A^T AB \left(R + B^T SB \right)^{-1} B^T SA$$

<u>Algebraic Riccati Equation</u> (A.R.E) for <u>discrete-time systems.</u>

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

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 feedbck.

(2) Equation A.R.E. is <u>matrix</u> quadratic equation. Looks pretty intimidating but Computer can solve in a second.

(3) <u>No tweaking</u> ! Just specify {A,B,C,D} and Q and R, press return button, LQR Routine Spits out K_{opt} - done
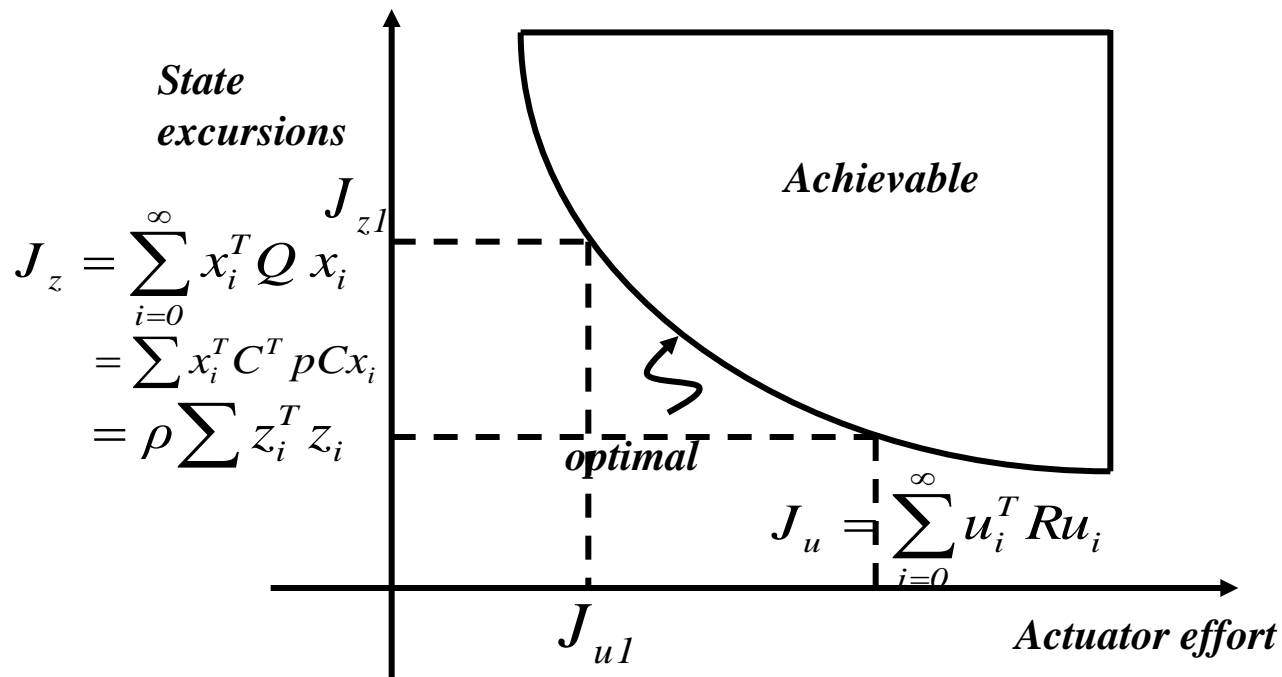
(Of course picking Q and R is tricky sometimes but that's another story).

(4) Design is guaranteed optimal in the sense that it minimizes.

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

(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)

$$J_z = \sum_{i=0}^{\infty} x_i^T Q x_i$$
$$= \sum x_i^T C^T p C x_i$$
$$= \rho \sum z_i^T z_i$$

State excursions

J_{z1}

Achievable

optimal

$$J_u = \sum_{i=0}^{\infty} u_i^T R u_i$$
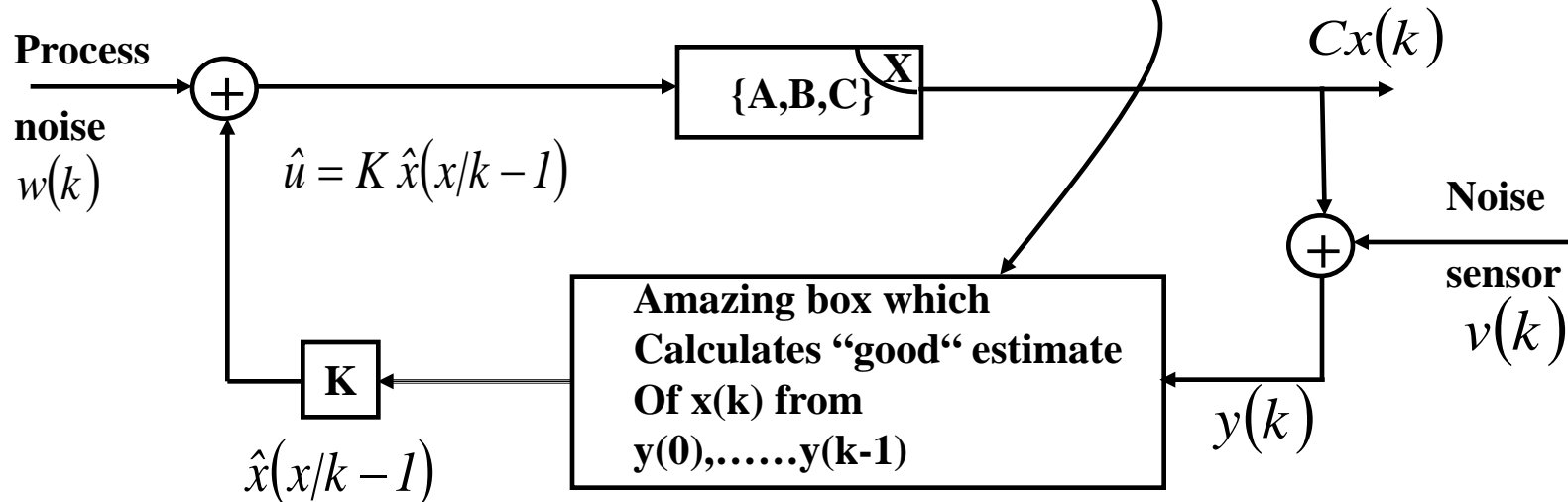
J_{u1}

Actuator effort

4.6 Optimal Linear Estimation

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

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

How can be 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:

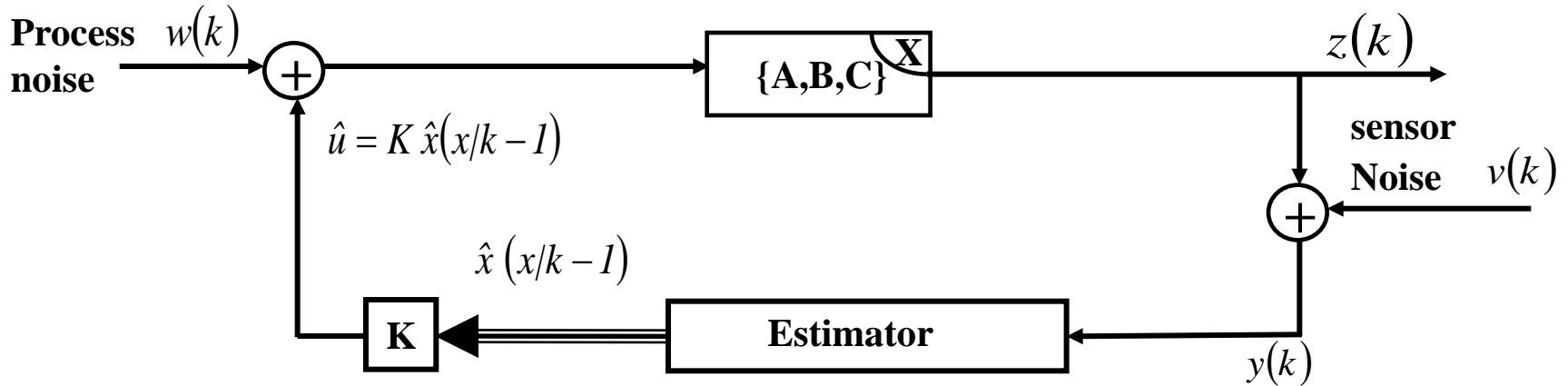**Process**

noise

$w(k)$

$\hat{u} = K\,\hat{x}(x/k-1)$

$\{A,B,C\}$ **X**

$Cx(k)$

Noise

sensor

$v(k)$

K

$\hat{x}(x/k-1)$

**Amazing box which
Calculates "good" estimate
Of x(k) from
y(0),……y(k-1)**

$y(k)$

4.6 Problem Statement :

$$x(k+1) = A\,x(k) + B\,w(k)$$
$$z(k) = C\,x(x)$$
$$y(k) = C\,x(k) + v(k)$$

Process noise $w(k)$

$\hat{u} = K\,\hat{x}(x/k-1)$

$\hat{x}(x/k-1)$

{A,B,C} X

$z(k)$

sensor Noise $v(k)$

K

Estimator

$y(k)$

Assume also $x(0)$ is Random & Gaussian and that $x(k),\,w(k)+V(k)$ are all <u>mutually Independent</u> for all k.

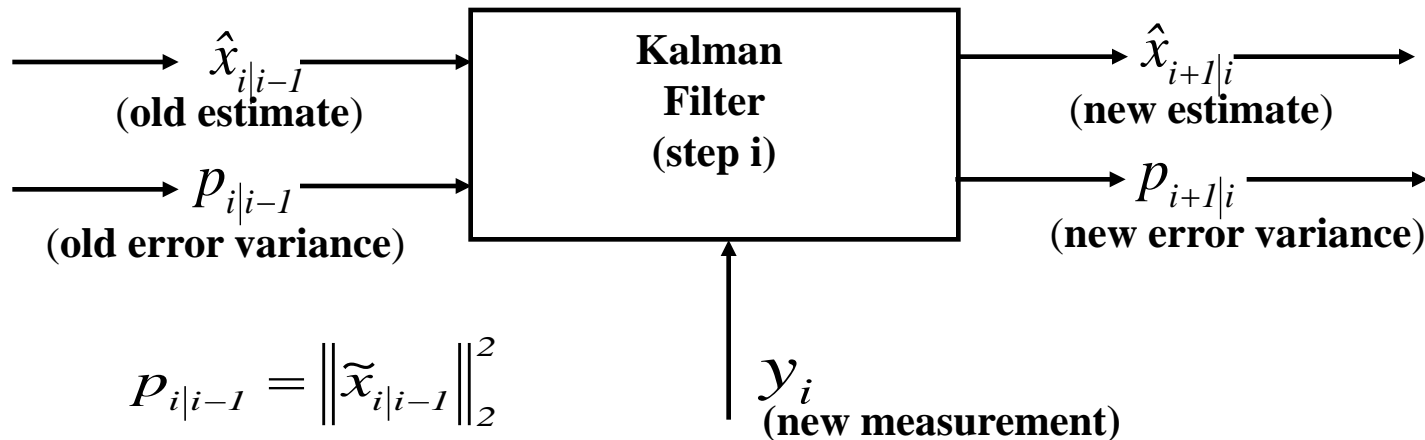*Find :* $\hat{x}\left(k/k-1\right)$ Optimal estimate of x(k) given $y_{0\,,..,}\,y_{k-1}$

Such that "mean squared error"

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

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

4.6 Kalman Filter

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

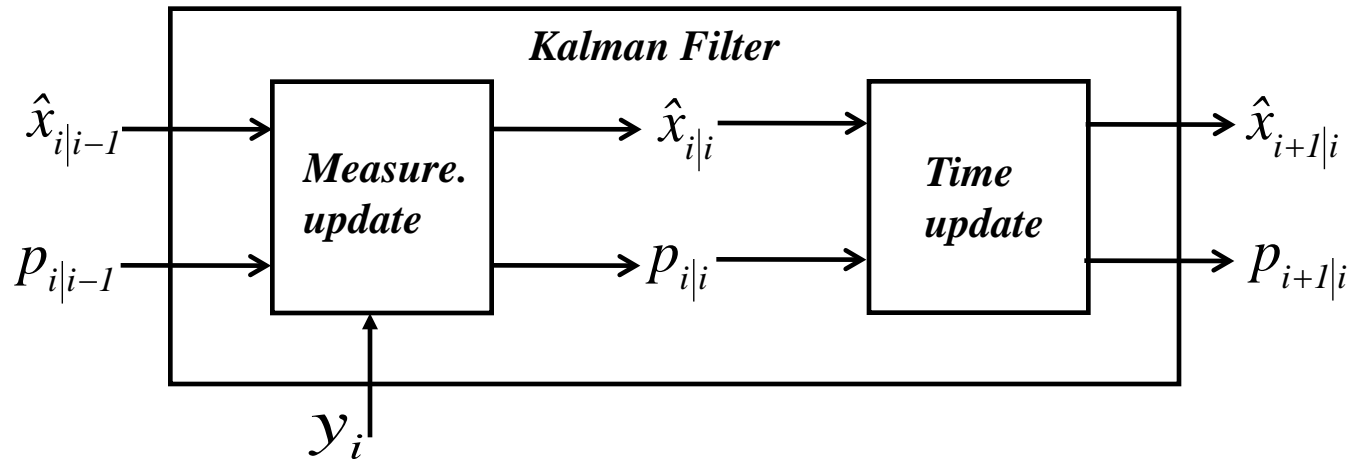$$P_{i|i-1} = \left\| \tilde{x}_{i|i-1} \right\|_2^2$$

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.):

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

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

Time Update (T.U.):

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

$$p_{i+1|i} = Ap_{i|i}A^T + BWB^T$$

With initial conditions:

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

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

4.6 Kalman Filter

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

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

$$= A\hat{x}_{i|i-1} + Ap_{i|i-1}C^T\left(Cp_{i|i-1}C^T + V\right)^{-1}\left(y_i - C\hat{x}_{i|i-1}\right)$$

$$\boxed{\begin{array}{l} \hat{x}_{i+1|i} = A\hat{x}_{i|i-1} + L_i\left(y_i - C\hat{x}_{i|i-1}\right) \\[2mm] \text{where} \quad L_i \equiv A\left(p_{i|i-1}C^T\left(Cp_{i|i-1}C^T + V\right)^{-1}\right) \end{array}}$$

$$p_{i+1|i} = Ap_{i|i}A^T + BWB^T$$

$$= A\left[p_{i|i-1} - p_{i|i-1}C^T\left(Cp_{i|i-1}C^T + V\right)^{-1}Cp_{i|i-1}\right]A^T + BWB^T$$

$$\boxed{p_{i+1|i} = Ap_{i|i}A^T + BWB^T - Ap_{i|i-1}C^T\left(Cp_{i|i-1}C^T + V\right)^{-1}\left(Cp_{i|i-1} - 1\right)A^T}$$

Known as discrete time Riccati Equation

4.6 Picture of Kalman Filter
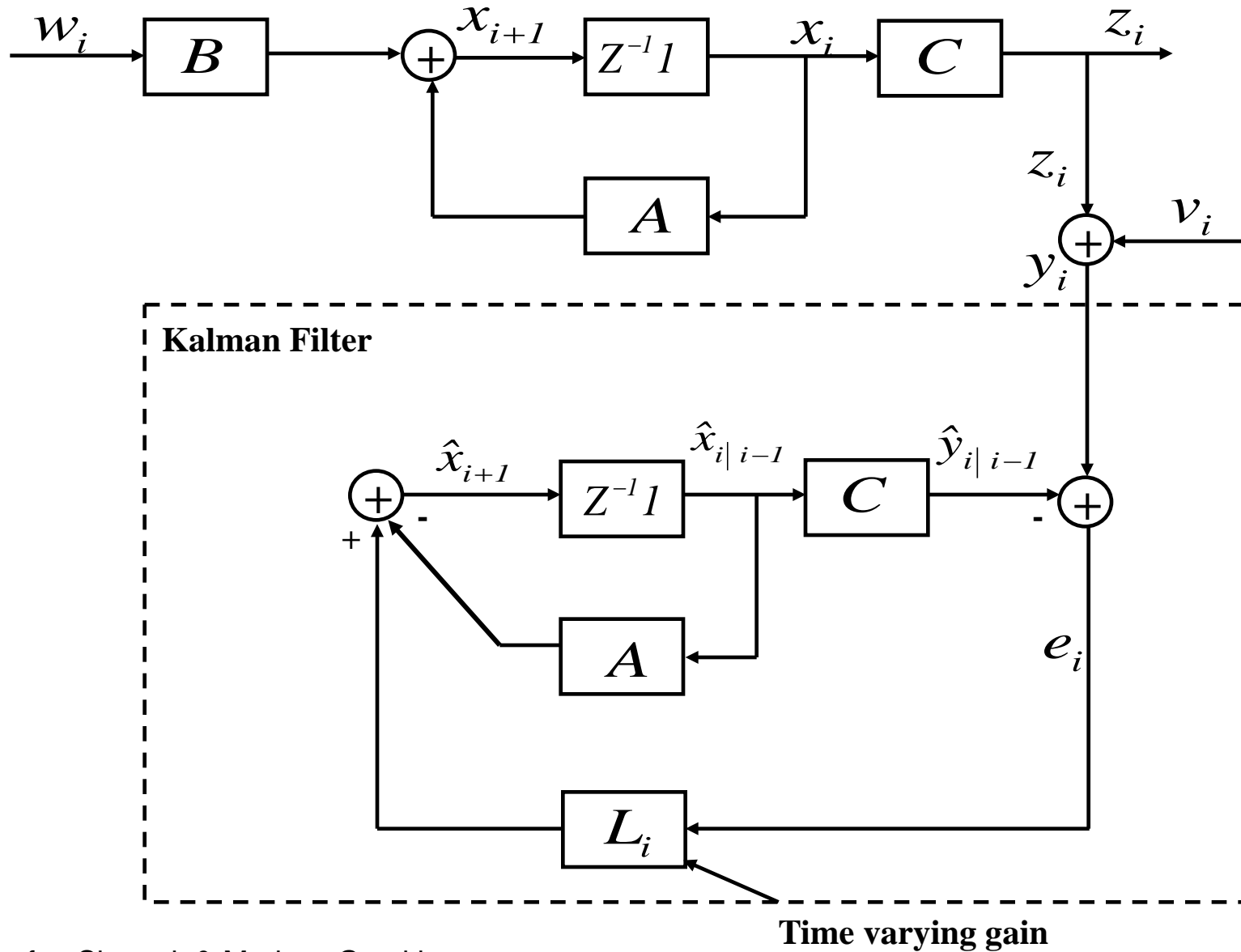
4.6 Picture of Kalman Filter

<u>Plant Equations:</u>

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

$$y_i = Cx_i + v_i$$

<u>Kalman Filter:</u>

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

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

If v=w=0=> 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

$$\hat{x}_{i+1|i} = A\,\hat{x}_{i|i-1} + L_i\left(C\,x_i + v_i - C\,\hat{x}_{i|i-1}\right)$$
$$= (A - L_iC)\,\hat{x}_{i|i-1} + L_iC\,x_i + v_i$$

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

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

$$\begin{bmatrix} z_i \\ \hat{y}_{i|i-1} \end{bmatrix} = \begin{bmatrix} C & 0 \\ 0 & C \end{bmatrix} \begin{bmatrix} x_i \\ \hat{x}_{i|i-1} \end{bmatrix}$$
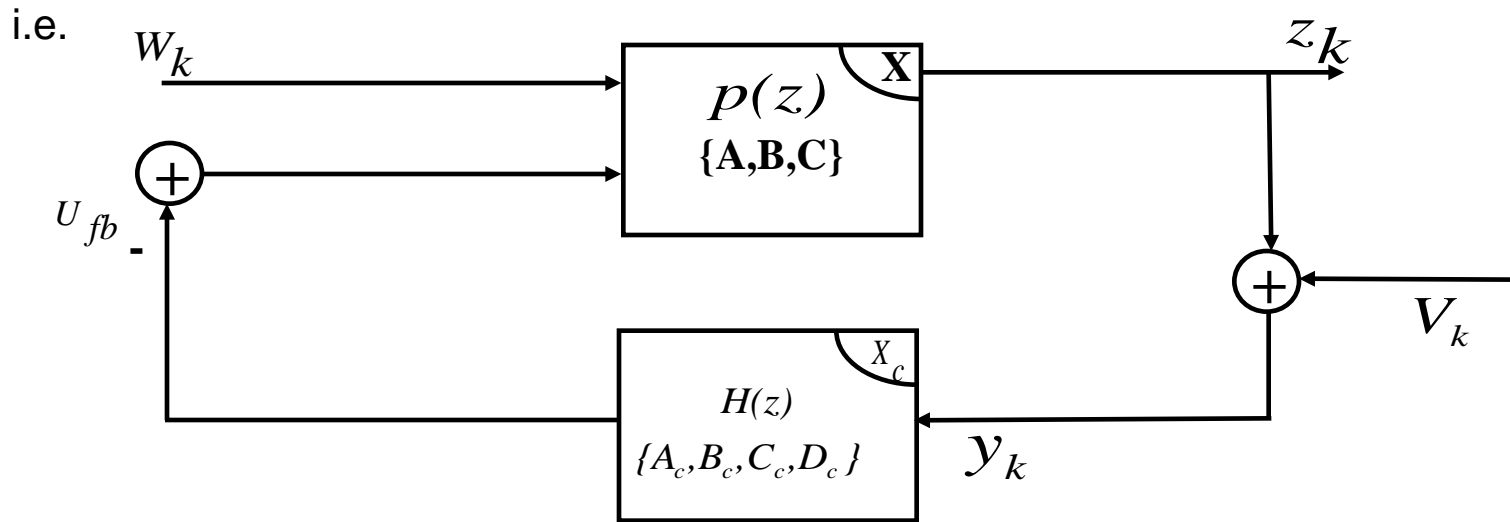
(2) In practice, the Riccati equation reaches <u>steady state</u> in a few steps. People Often run with steady-state K.F.i.e

Where
$$L_{ss} = Ap_{ss}\,C^T(CP_{ss}\,C^T + V)^{-1}$$
$$p_{ss} = Ap_{ss}\,A^T + BWB^T - Ap_{ss}\,C^T(CP_{ss}\,C^T + V)^{-1}CP_{ss}\,A$$

4.7 LQG Problem

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

i.e.

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 LQG problem

Separation principle: (we won't prove)

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

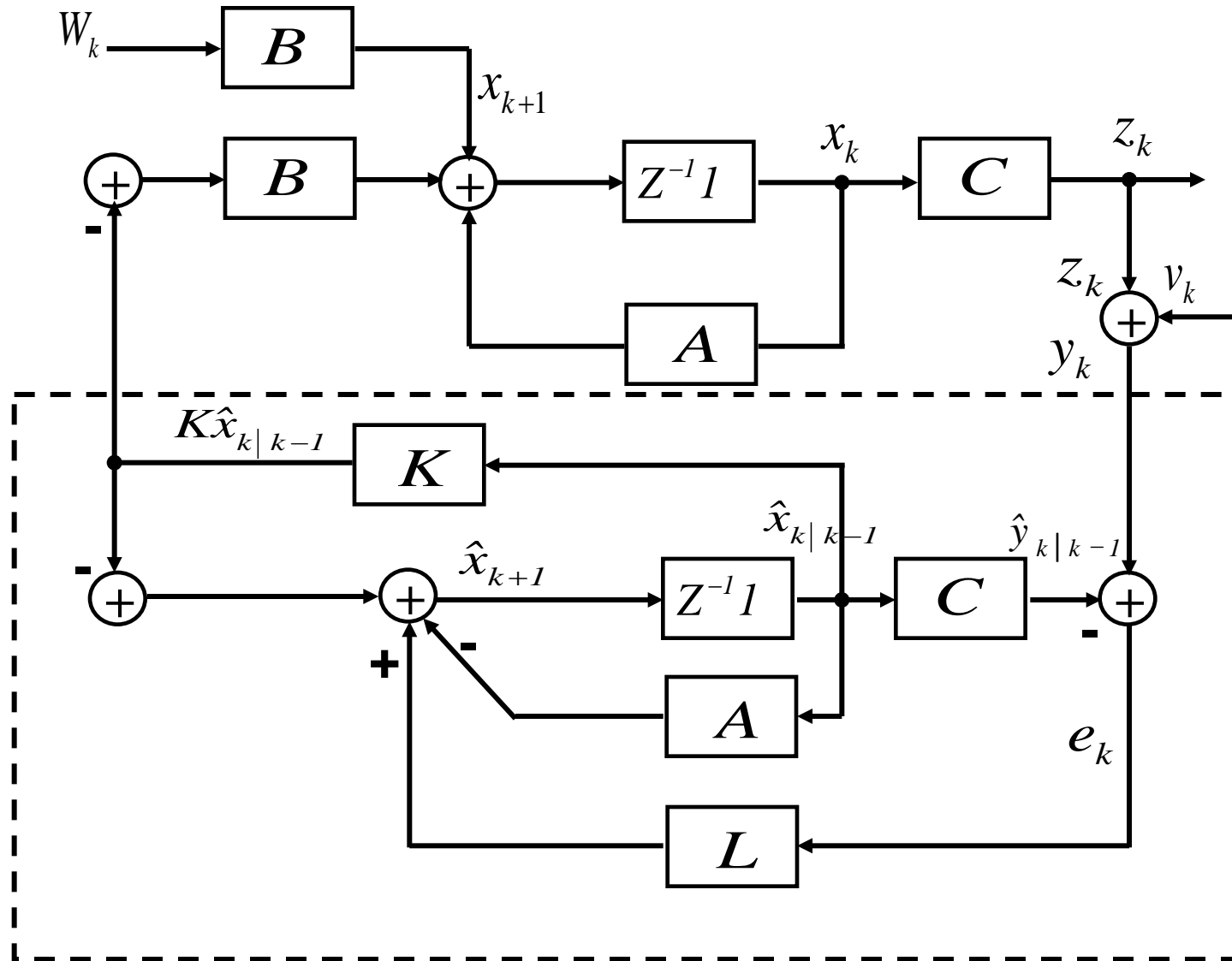(1) Using Kalman filter to obtain least squares optimal estimate of the plant state,

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

(2) Feedback <u>estimated</u> LQR- optimal state feedback

$$u(k) = -K_{LQR} \, x_c(k)$$
$$= -K_{LQR} \, \hat{x}_{k|k-1}$$

i.e. can treat problems of
- optimal feedback and
- state estimate separately.

4.7 Picture of LQG Regulator

4.7 LQG Regulator

Plant

$$\left\| \begin{matrix} x_{k+1} = A\,x_k + (-Bu_k) + B_w w_k \\ z_k = C\,x_k \\ y_k = C\,x_k + v_k \end{matrix} \right\|$$

LQG Controller

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

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

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

$$= APA^T + BWB^T - APC^T\left[V + CPC^T \right]^{-1} CPC^T$$
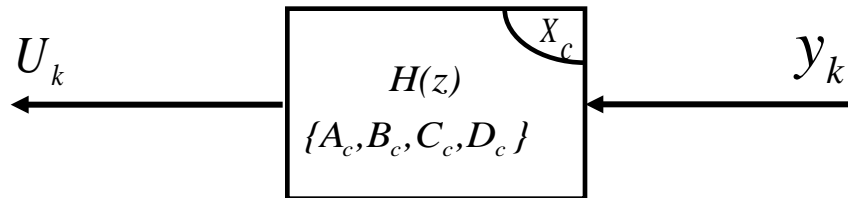
4.7 Problem Statement (in English)

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

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

$$U_k \longleftarrow \boxed{\begin{array}{c} X_c \\ H(z) \\ \{A_c, B_c, C_c, D_c\} \end{array}} \longleftarrow y_k$$

Controller:

$$x_c(k+1) = A_c \, x(k+1) + B_c \, y(k)$$

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

Which will minimize the cost

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

Where

$$\left\| \begin{array}{l} x_{k+1} = A x_k + \left(-B u_k\right) + B_w w_k \end{array} \right.$$

Plant

$$z_k = C \, x_k$$

$$y_k = C \, x_k + v_k$$

4.7 Problem Statement

Remarks:

(1). Q and R are weigthing matrices that allow trading off of rms u and rms x.

(2) if $Q = C^T \rho \, C; \, \rho > 0$ then trade off rms z VS rms u

(3). 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
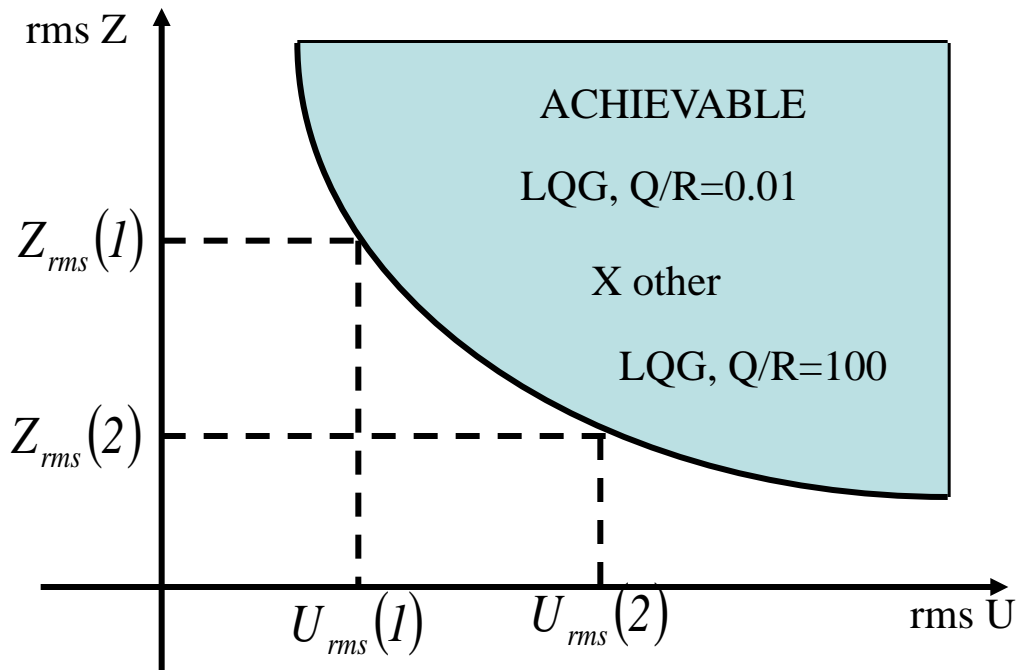
   i .e **can't** use full state feedback.

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

4.7 Problem Statement

(5) 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 <u>limit</u> of performance of

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