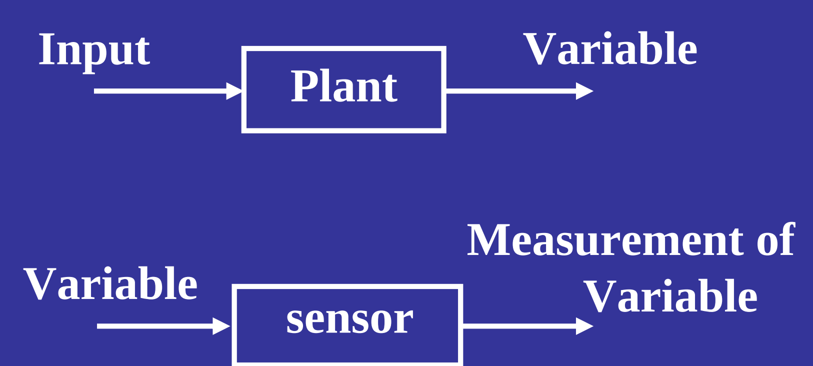
1.Control Theory

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.

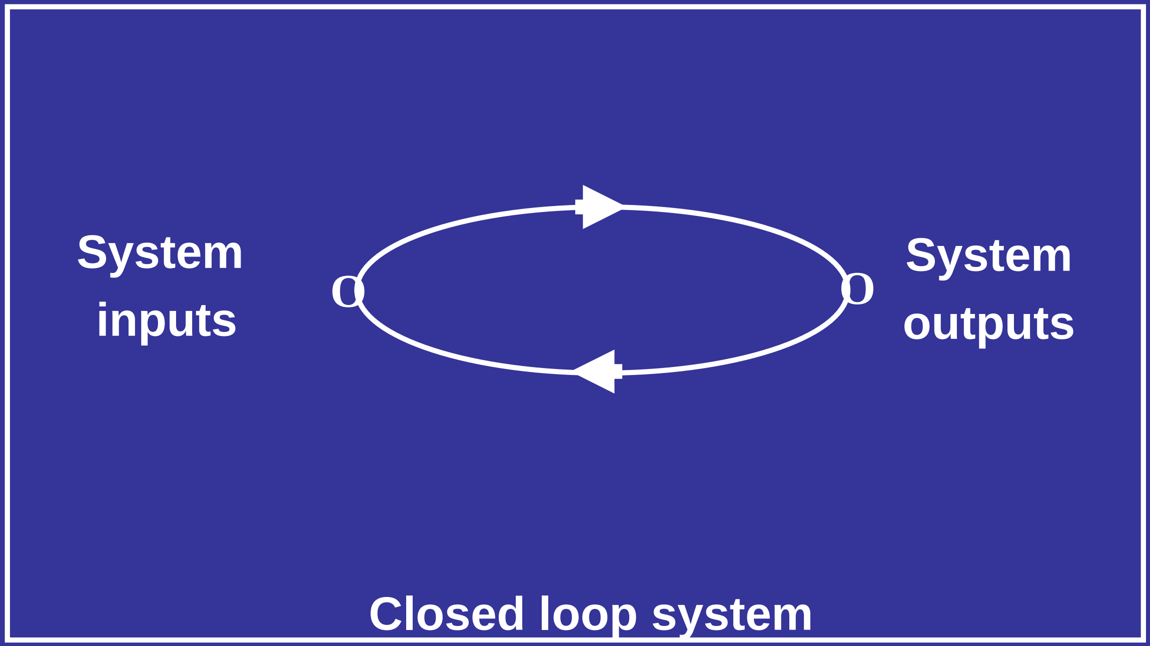
Design:
Desired system characteristics or behavior are specified ⟶ configure or synthesize closed loop system.



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).



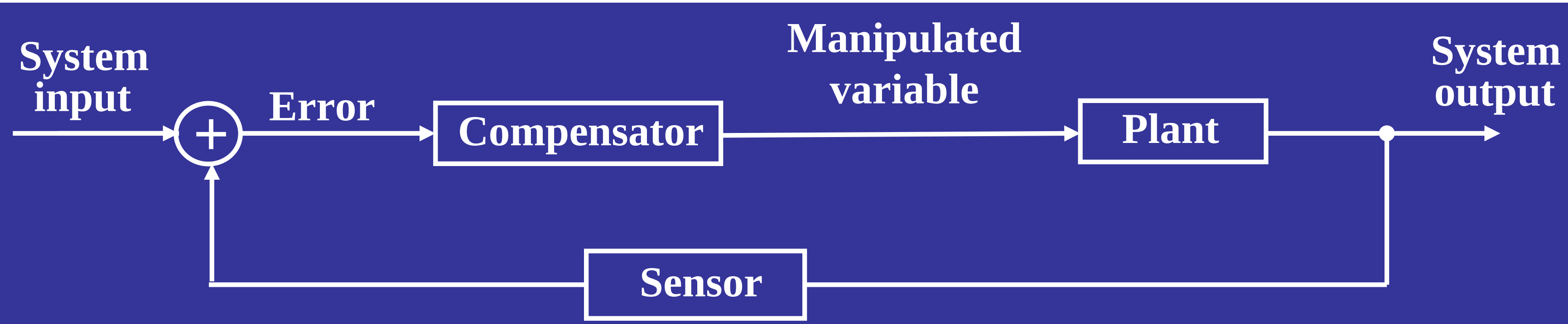
1.Introduction

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.



Closed loop control system

Two types of control systems:

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

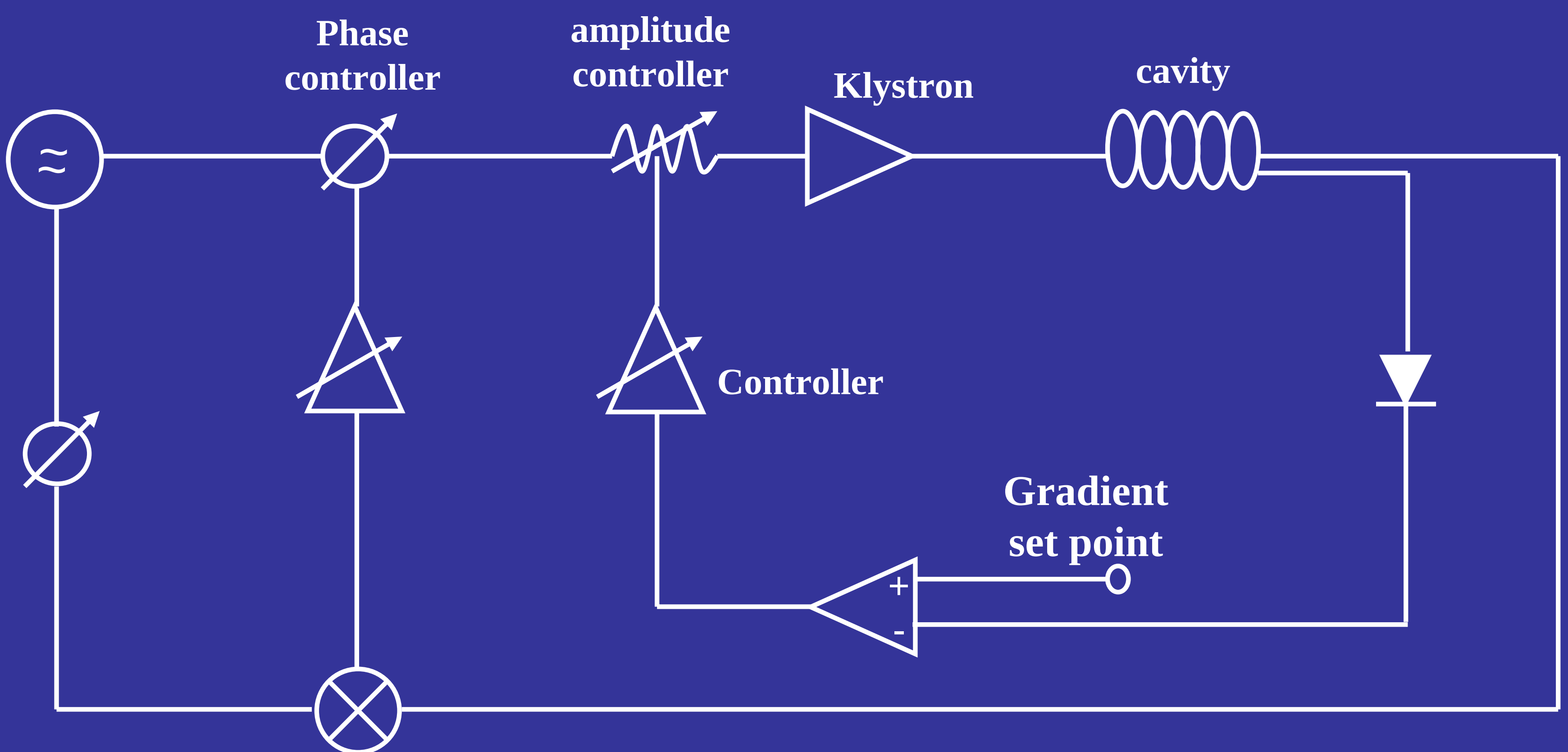
❖A ***servomechanism*** describes a control system in which a physical variable is required to follow, or track some desired time function (originally applied in order to control a mechanical position or motion).

1.Introduction

Example 1: RF control system

Goal:
Maintain stable gradient and phase.

Solution:
Feedback for gradient amplitude and phase.



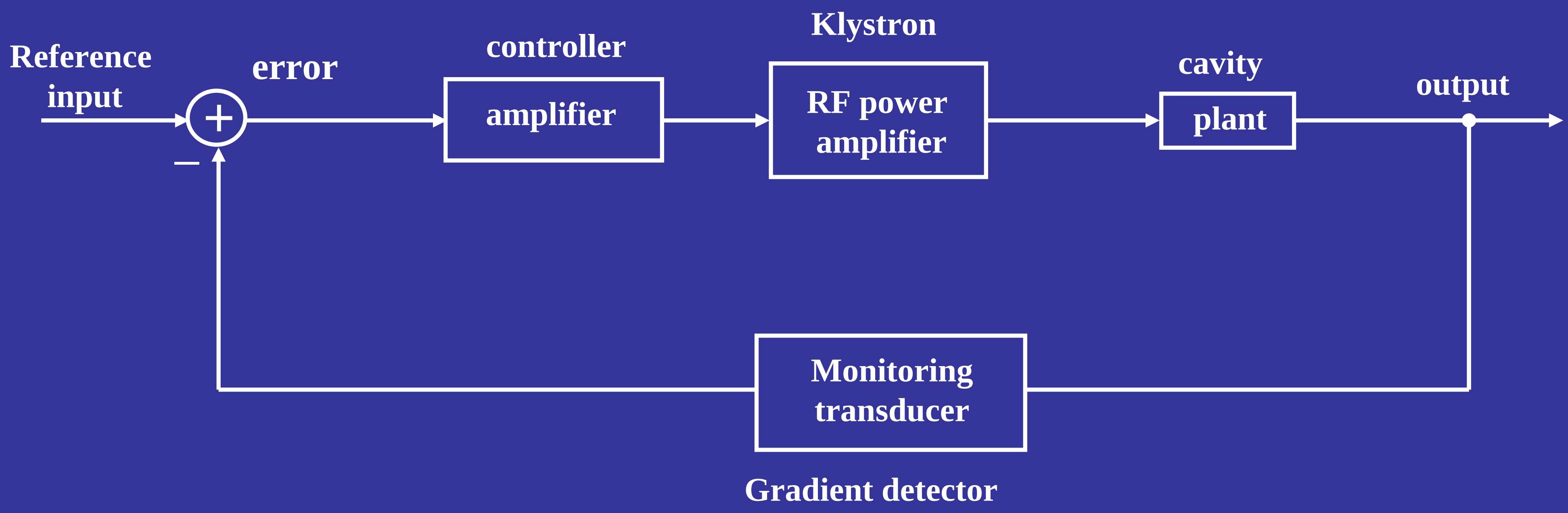
continued..

1.Control Systems

Model:

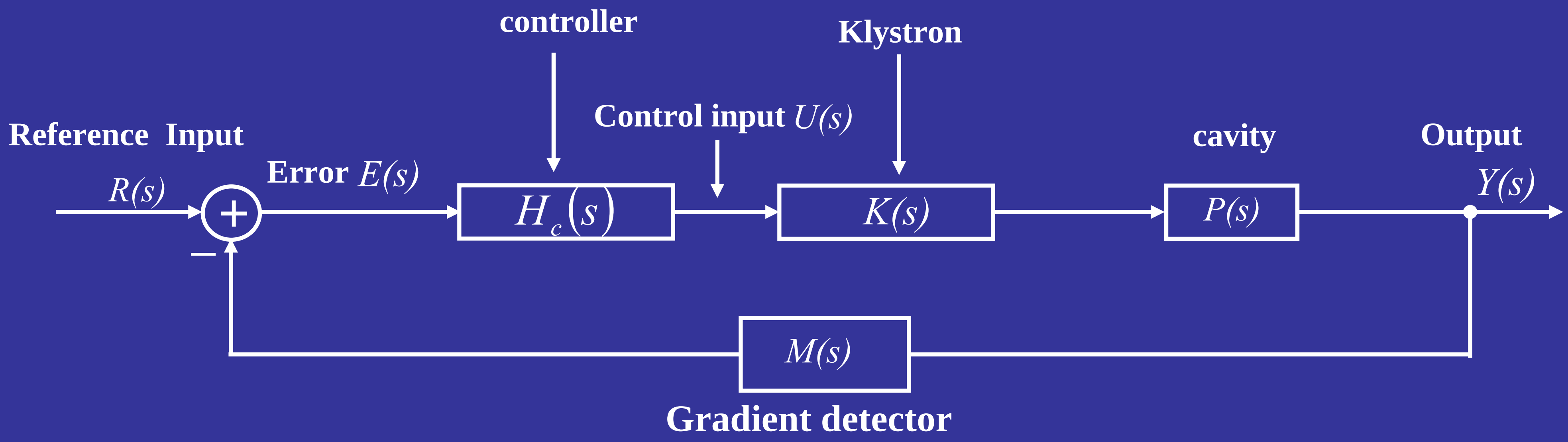
Mathematical description of input-output relation of components combined with block diagram.

Amplitude loop (general form):



1.Introduction

RF control model using *"**transfer functions**"*



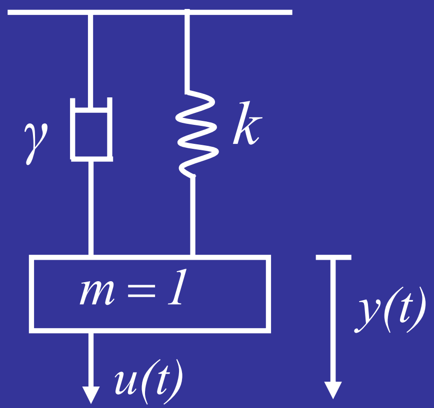
A transfer function of a **linear** system is defined as the ratio of the Laplace transform of the output and the Laplace transform of the input with I. C .'s =zero.

Input-Output Relations

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

2. 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. $= \frac{\partial}{\partial x}\frac{\partial}{\partial t} f(x,t) = 0$)

-All coefficients constant: $k(t) = \kappa\ , \gamma(t) = \gamma$ for all t

2. Model of Dynamic System

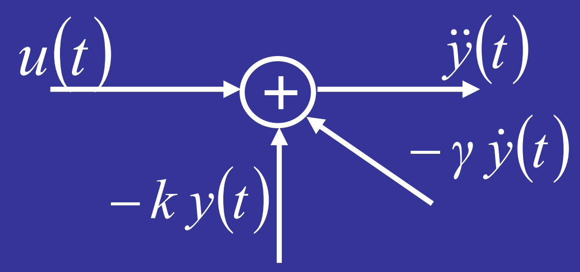
Stop calculating, let's paint!!!

Picture to visualize differential equation

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

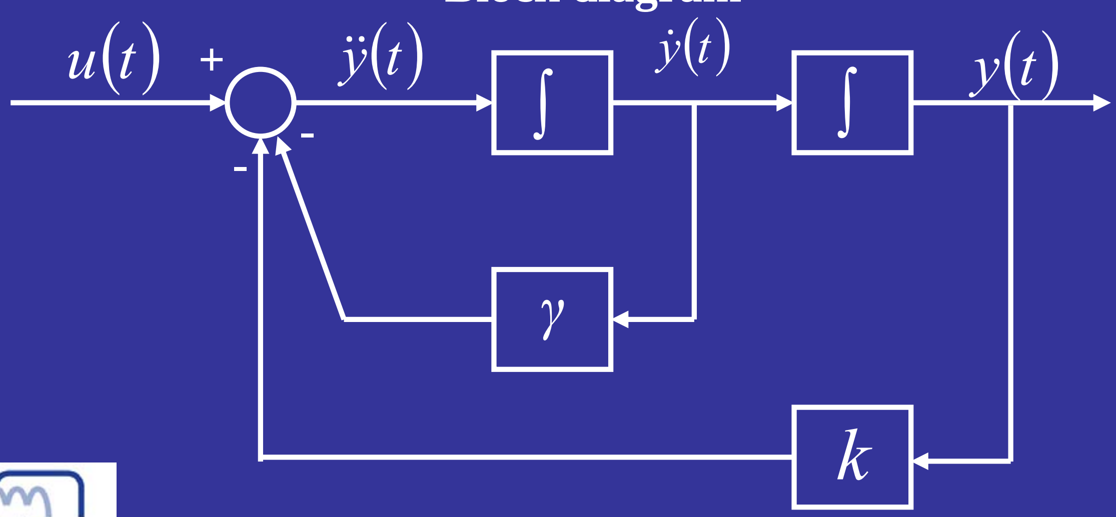
$$\ddot{y}(t) = -k\,y(t) - \gamma\,\dot{y}(t) + u(t)$$

2.Putt adder in front



3.Synthesize all other terms using integrators!

Block diagram



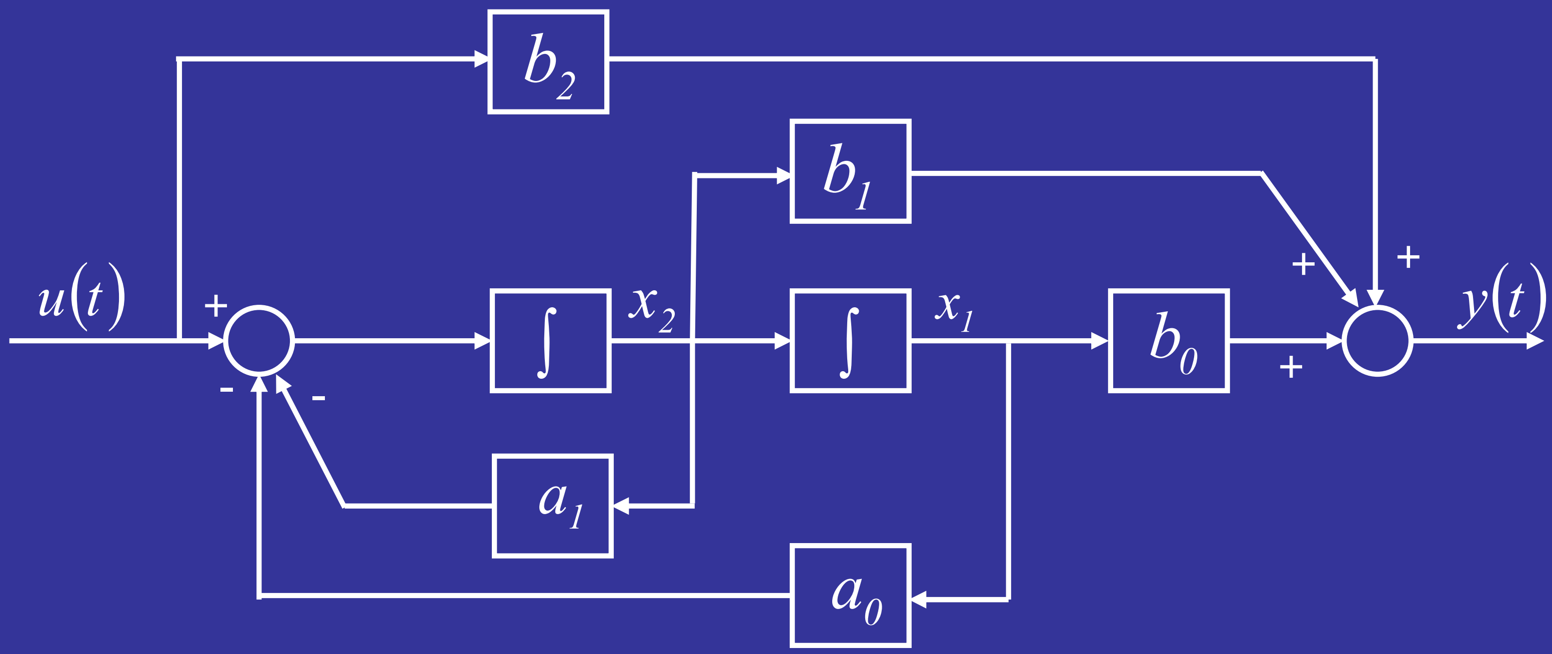
2.1Linear Ordinary Differential Equation (LODE)

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 *interaction* 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).

What do we have to do ???

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 : $\dot{y}(t)$:

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

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

<u>One LODE of order *n* transformed into *n* LODEs of order 1</u>

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

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

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

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

Declaration of variables

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

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!)

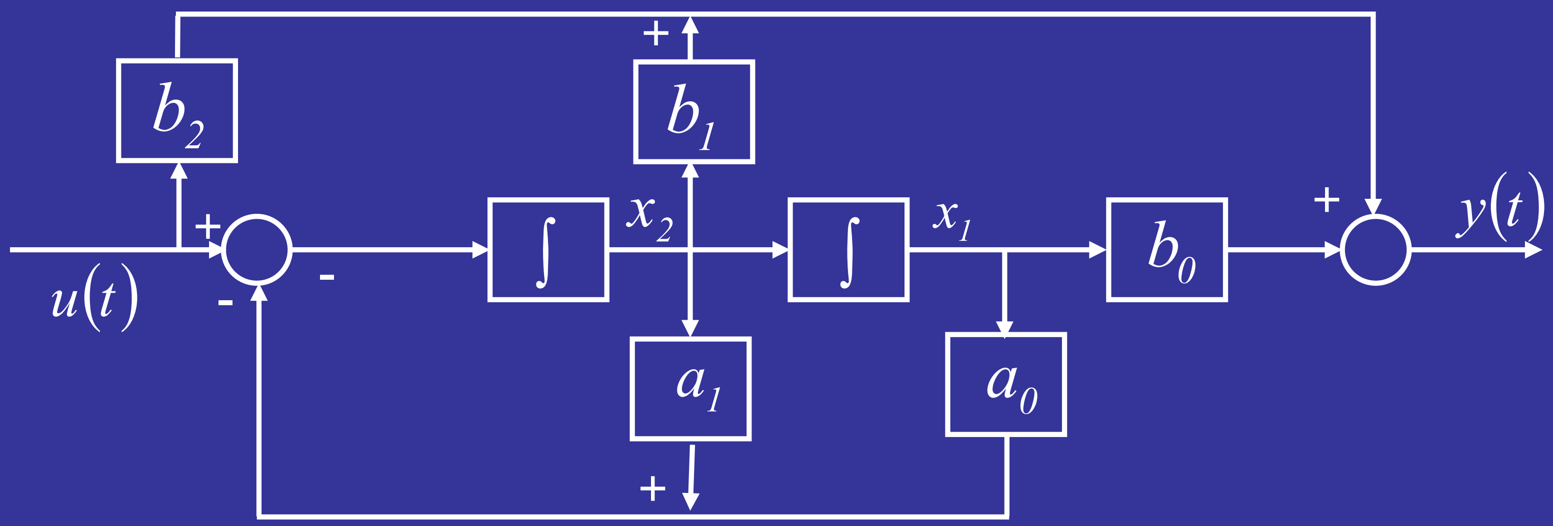
Modern control (1960-now) uses state space equation.

General (vector) block diagram for easy visualization.

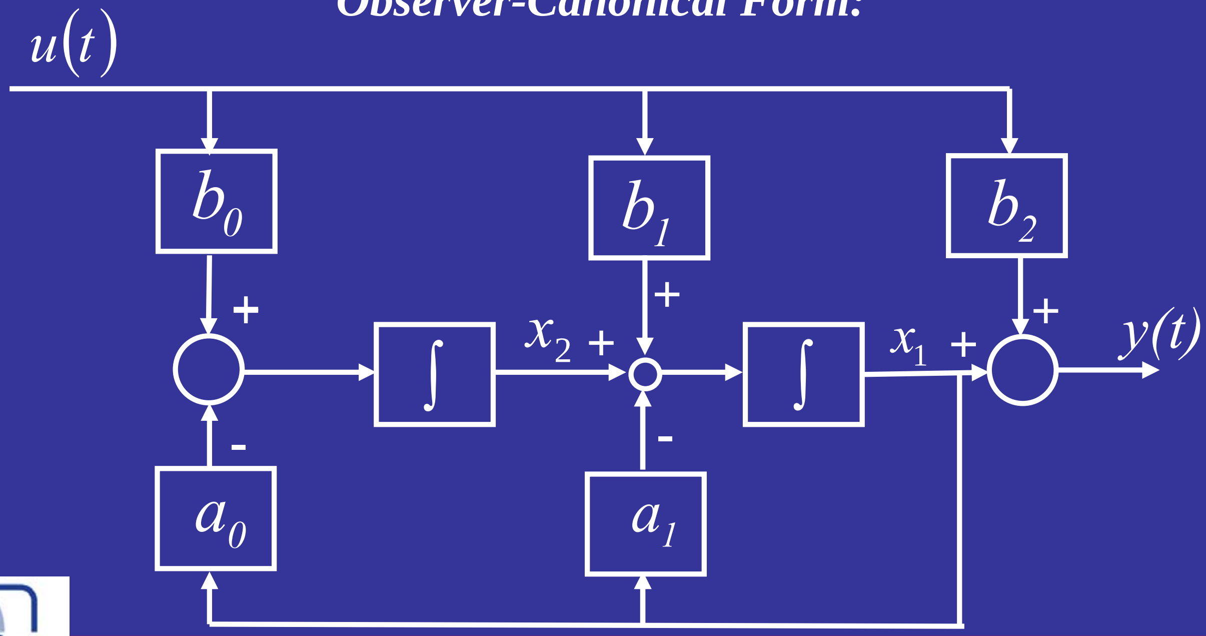
2.2 State Space Equation

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***

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

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

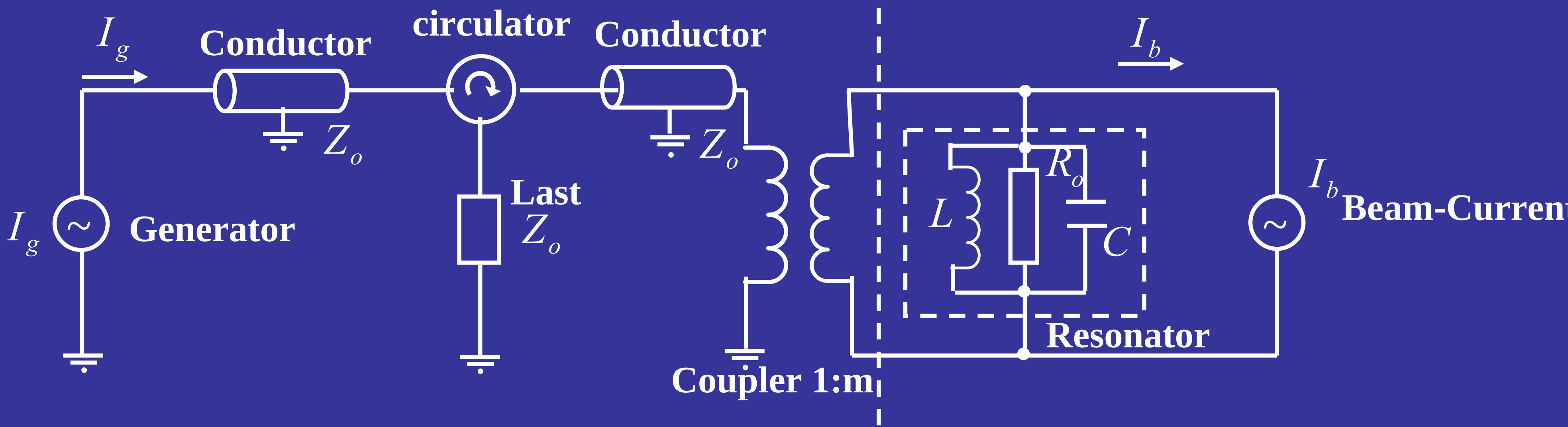
1. $\dfrac{d\Phi(t)}{dt} = A\,\Phi(t)$
2. $\Phi(0) = I$
3. $\Phi(t_1 + t_2) = \Phi(t_1)\cdot\Phi(t_2)$
4. $\Phi^{-1}(t) = \Phi(-t)$

Example:

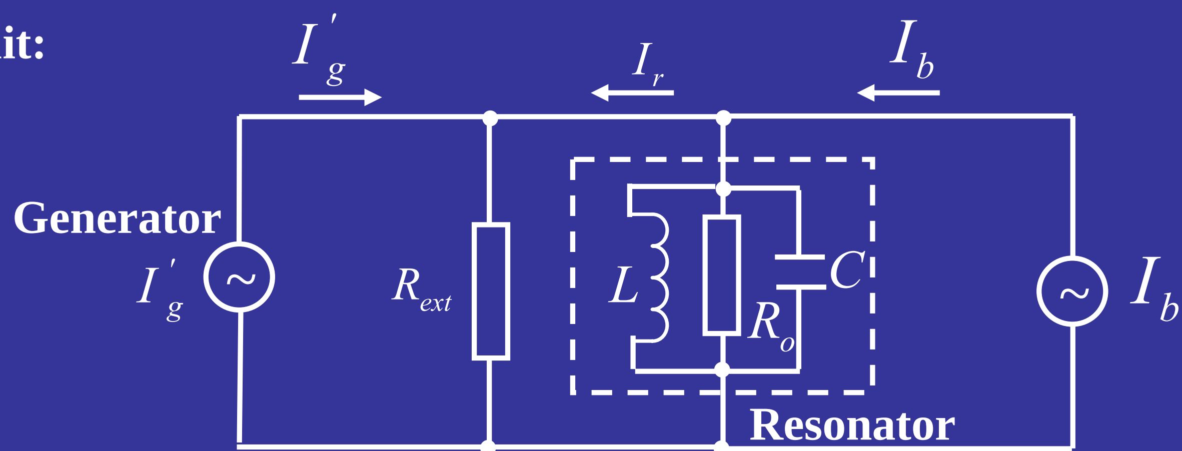
$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \Rightarrow A^2 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix},\; \Phi(t) = I + At = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix} = e^{At}$$

Matrix A is a nilpotent matrix.

2.3 Cavity model



Equivalent circuit:



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

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

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

2.3 Cavity model

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

$$U(t)=(U_r(t)+i\,U_i(t))\cdot exp(i\,\omega_{HF}t)$$
$$I_g(t)=(I_{gr}(t)+i\,I_{gi}(t))\cdot exp(i\,\omega_{HF}t)$$
$$I_b(t)=(I_{b\omega r}(t)+i\,I_{b\omega i}(t))\cdot exp(i\,\omega_{HF}t)=2(I_{b0r}(t)+i\,I_{b0i}(t))\cdot exp(i\,\omega_{HF}\,t)$$

Neglect small terms in derivatives for U and I

$$\ddot{U}_r+i\ddot{U}_i(t)<<\omega_{HF}^2(U_r(t)+iU_i(t))$$
$$2\omega_{1/2}(\dot{U}_r+i\dot{U}_r(t))<<\omega_{HF}^2(U_r(t)+iU_i(t))$$
$$\int_{t1}^{t2}(\dot{I}_r(t)+i\dot{I}_i(t))\,dt<<\int_{t1}^{t2}\omega_{HF}(I_r(t)+iI_i(t))\,dt$$

Envelope equations for real and imaginary component.

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

2.3 Cavity model

Matrix equations:

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

With system Matrices:

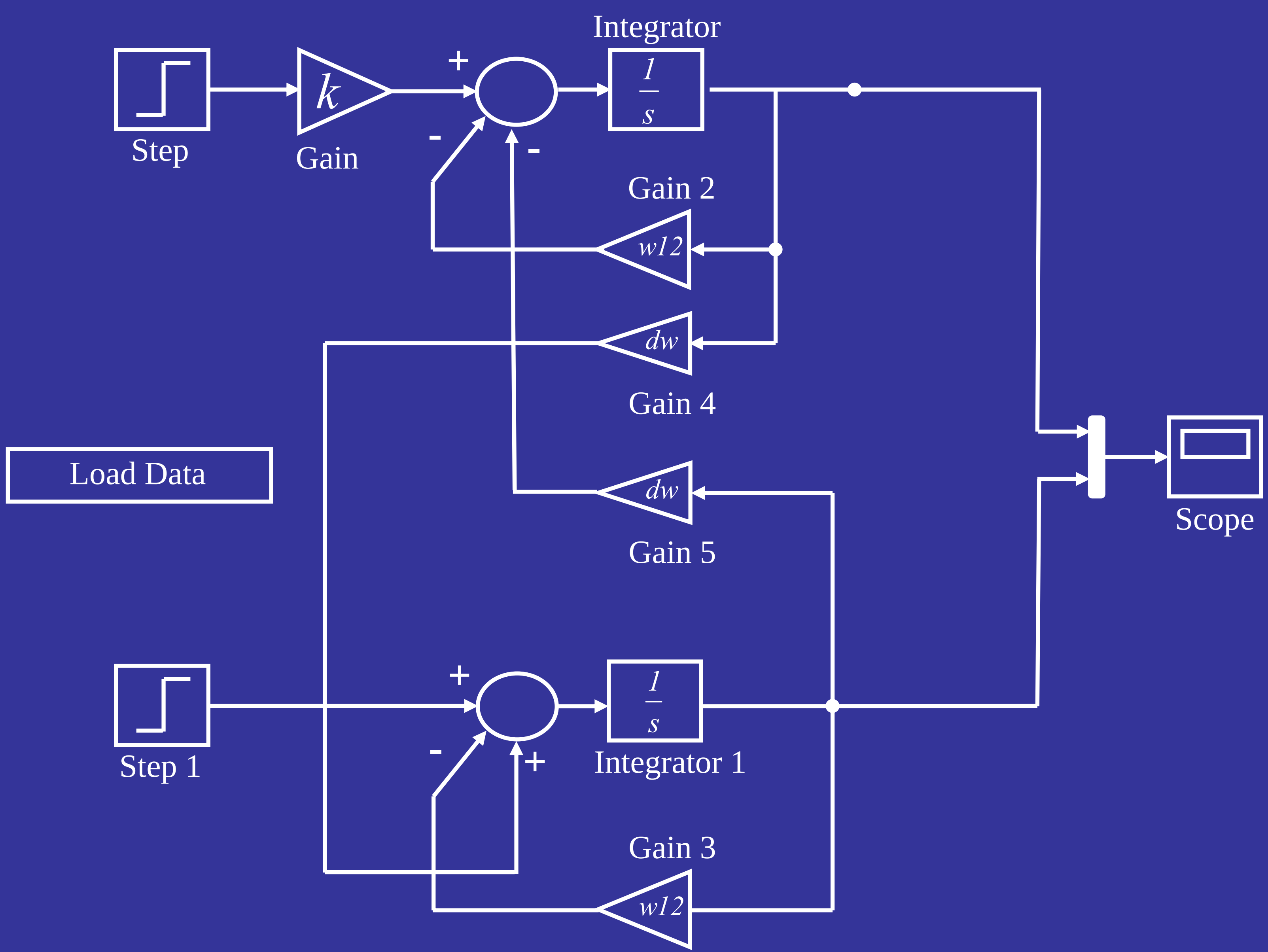
$$A = \begin{bmatrix} -\omega_{1/2} & -\Delta\omega \\ \Delta\omega & -\omega_{1/2} \end{bmatrix} \qquad B = \omega_{HF} \left(\frac{r}{Q} \right) \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\vec{x}(t) = \begin{bmatrix} U_r(t) \\ U_i(t) \end{bmatrix} \qquad \vec{u}(t) = \begin{bmatrix} \frac{1}{m} I_{g\,r}(t) + I_{b0r}(t) \\ \frac{1}{m} I_{g\,i}(t) + I_{b0i}(t) \end{bmatrix}$$

General Form:

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

2.3 Cavity Model



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.

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

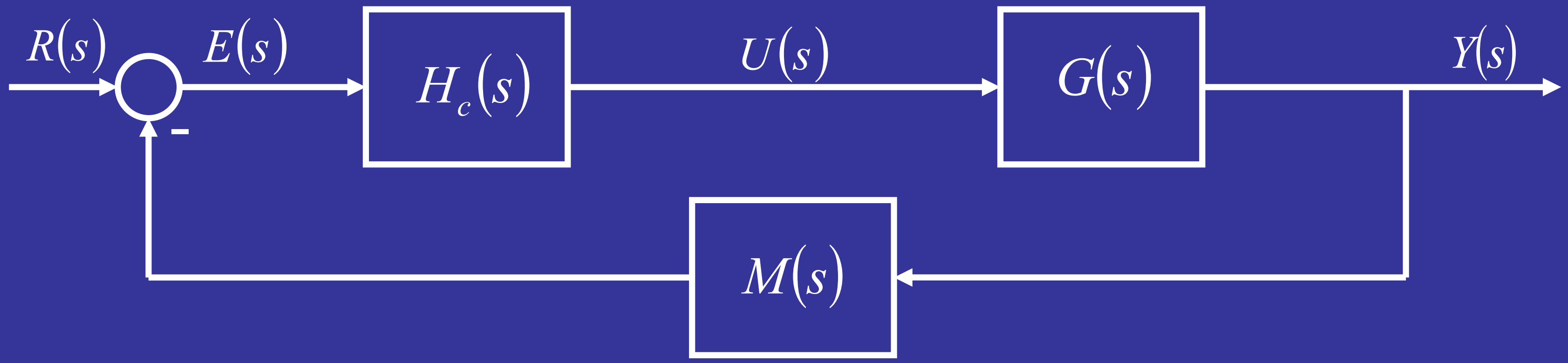
$$X(s) = (sI - A)^{-1} x(0) + (sI - A)^{-1} B\,U(s)$$
$$= \Phi(s)\,x(0) + \Phi(s)\,B\,U(s)$$

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

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

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

2.5 Transfer Function of a Closed Loop System



We can deduce for the output of the system.

$$
\begin{aligned}
Y(s) &= G(s)\,U(s) = G(s)\,H_c(s)\,E(s) \\
&= G(s)\,H_c(s)\left[R(s) - M(s)\,Y(s)\right] \\
&= L(s)\,R(s) - L(s)\,M(s)\,Y(s)
\end{aligned}
$$

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

$$
\left(I + L(s)\,M(s)\right) Y(s) = L(s)\,R(s)
$$

$$
\begin{aligned}
Y(s) &= \left(I + L(s)\,M(s)\right)^{-1} L(s)\,R(s) \\
&= T(s)\,R(s)
\end{aligned}
$$

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

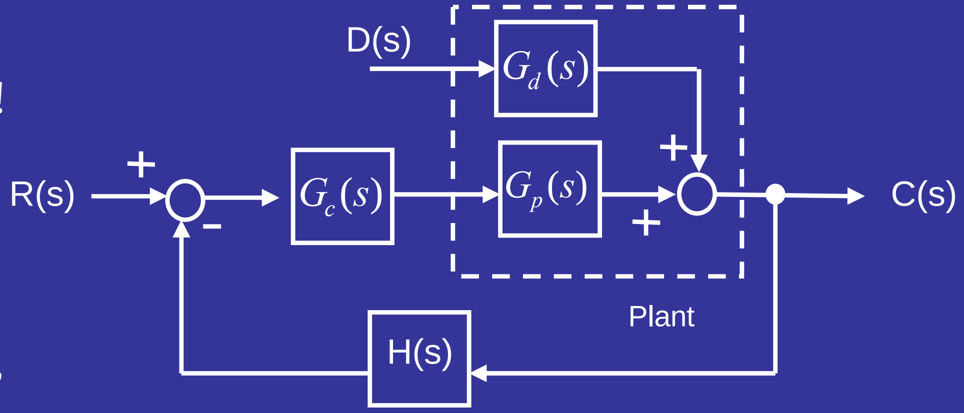
2.5 Disturbance Rejection

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 $T \cdot d(s) \cdot D(s)$ small!



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

- Reduce the Gain $Gd(jw)$ between dist. Input and output
- Increase the loop gain $GcGp(jw)$ without increasing the gain $Gd(jw)$. Usually accomplished by the compensator choice $Gc(jw)$
- Reduce the disturbance magnitude $d(t)$ Should always be attempted if reasonable
- Use feedforward compensation, if disturbance can be measured.

2.7 Poles and Zeroes

Stability directly from state-space

$$Recall: H(s) = C(sI - A)^{-1} B + D$$

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

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

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 $$b(s) = C(sI - A)_{adj} B$$

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

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

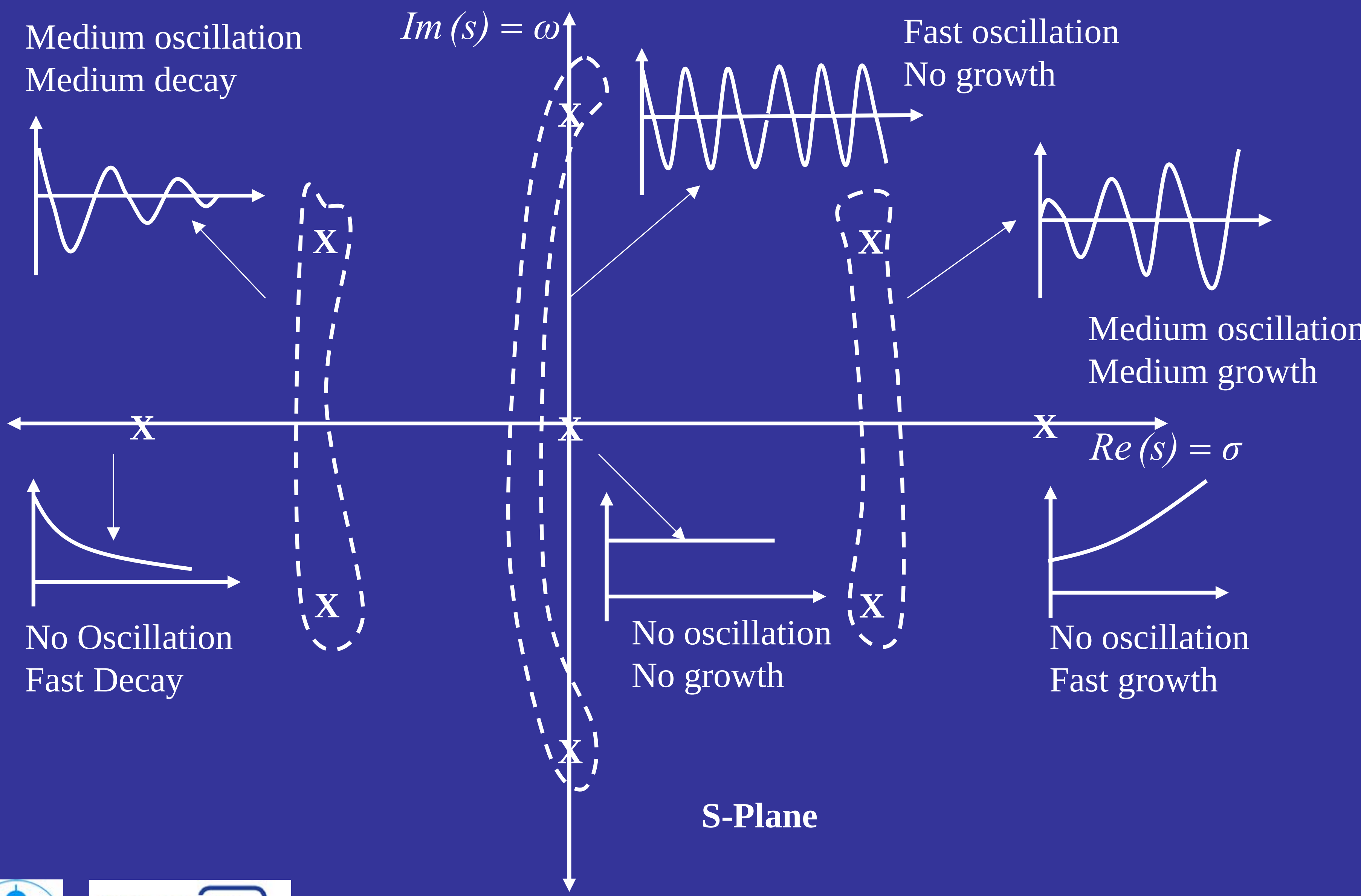
Let λ_i be the i^{th} eigenvalue of A

if $\mathrm{Re}\{\lambda_i\} \leq 0$ for all i => System stable

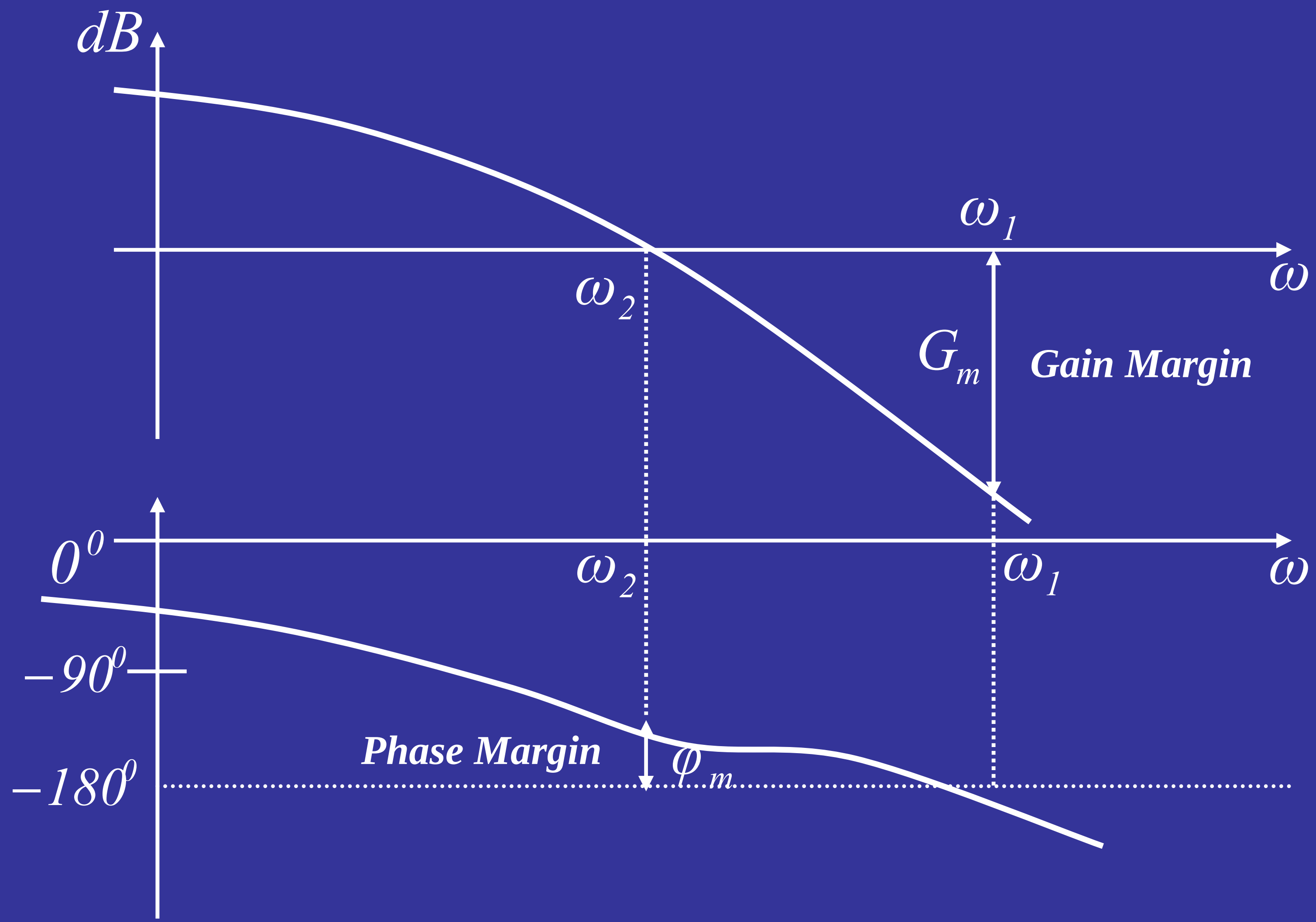
So with computer, with eigenvalue solver, can determine system stability directly from coupling matrix A.

2.8 Poles and Zeroes

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



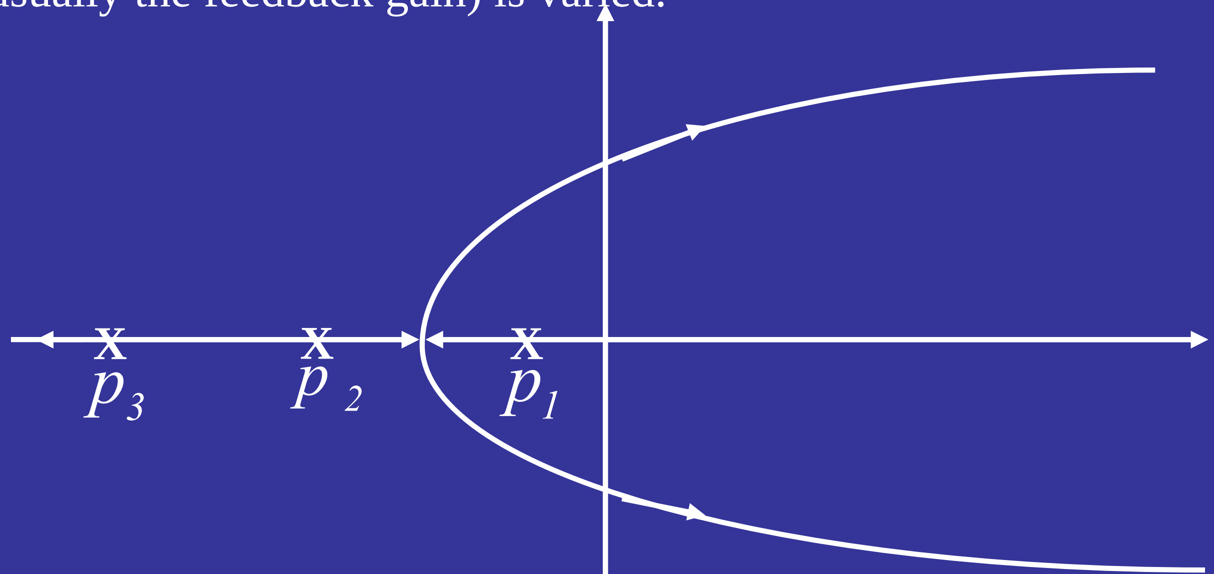
2.8 Bode Diagram



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.



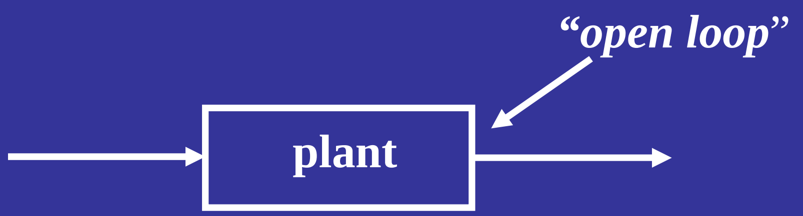
$$K\,H(s) = \frac{K}{(s-p_1)\,(s-p_2)\,(s-p_3)}$$

$$G_{CL}(s) = \frac{K\,H(s)}{1+K\,H(s)}\; roots\ at\ 1+K\,H(s)=0.$$

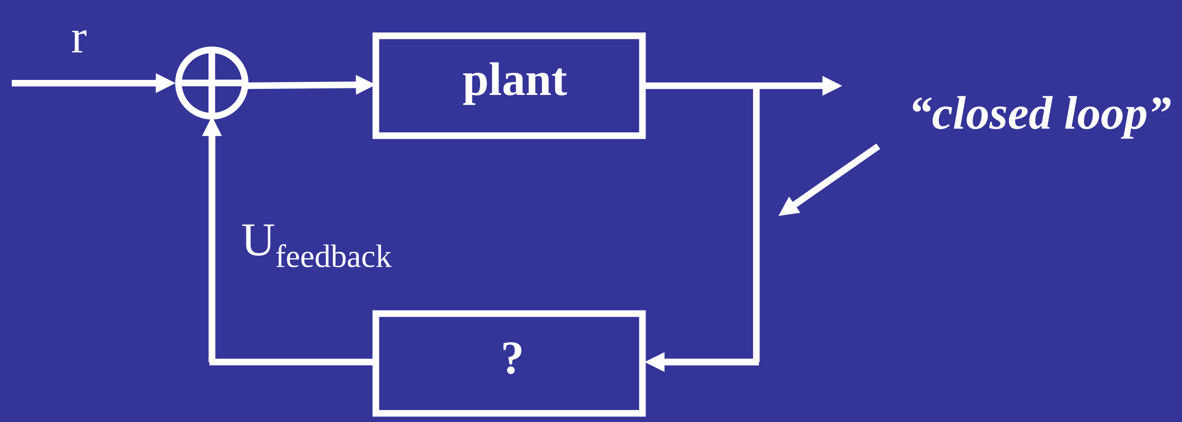
How do we move the poles by varying the constant gain K?

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***



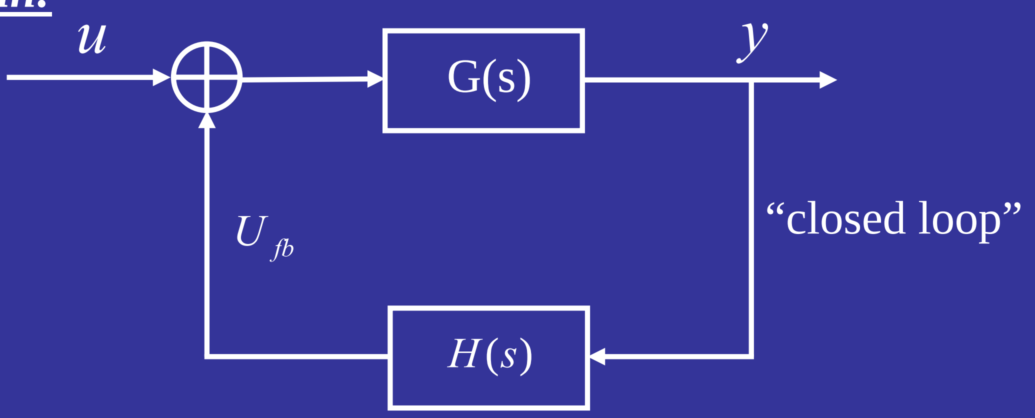
Question: What should this be?

3.Feedback

Open loop gain:

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

Closed-loop gain:



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

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

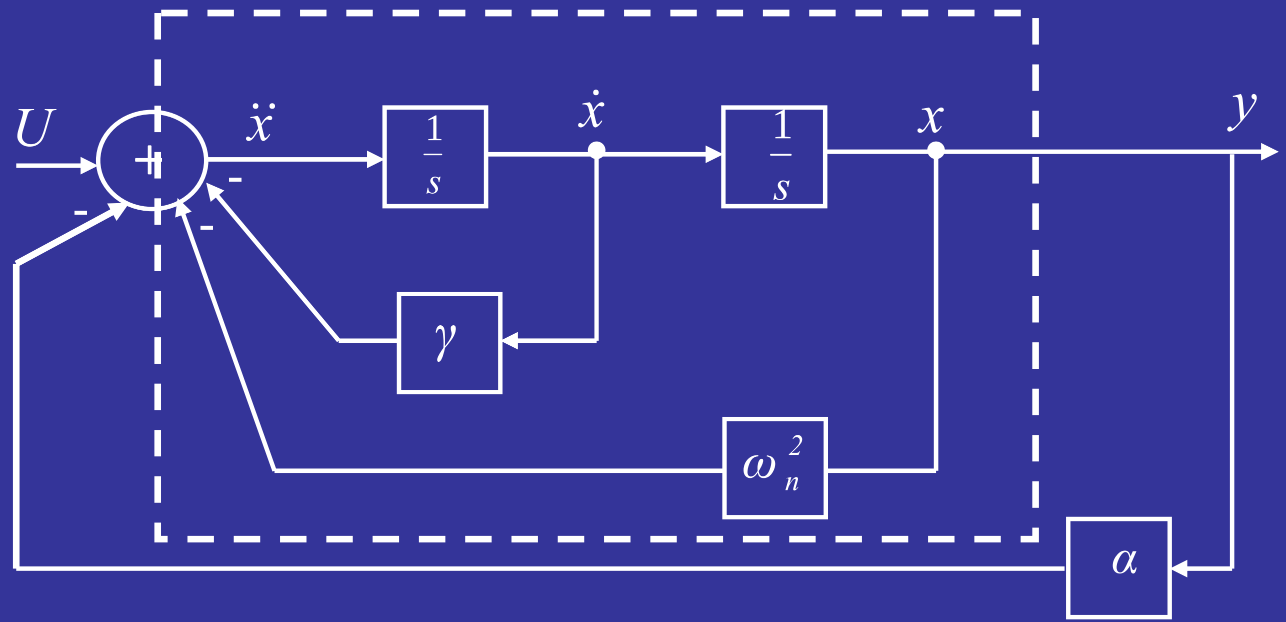
$$= G\,u - G\,u_{fb} \qquad \Rightarrow y + G\,H_y = G\,u$$

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

3.1 Feedback-Example 1

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

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

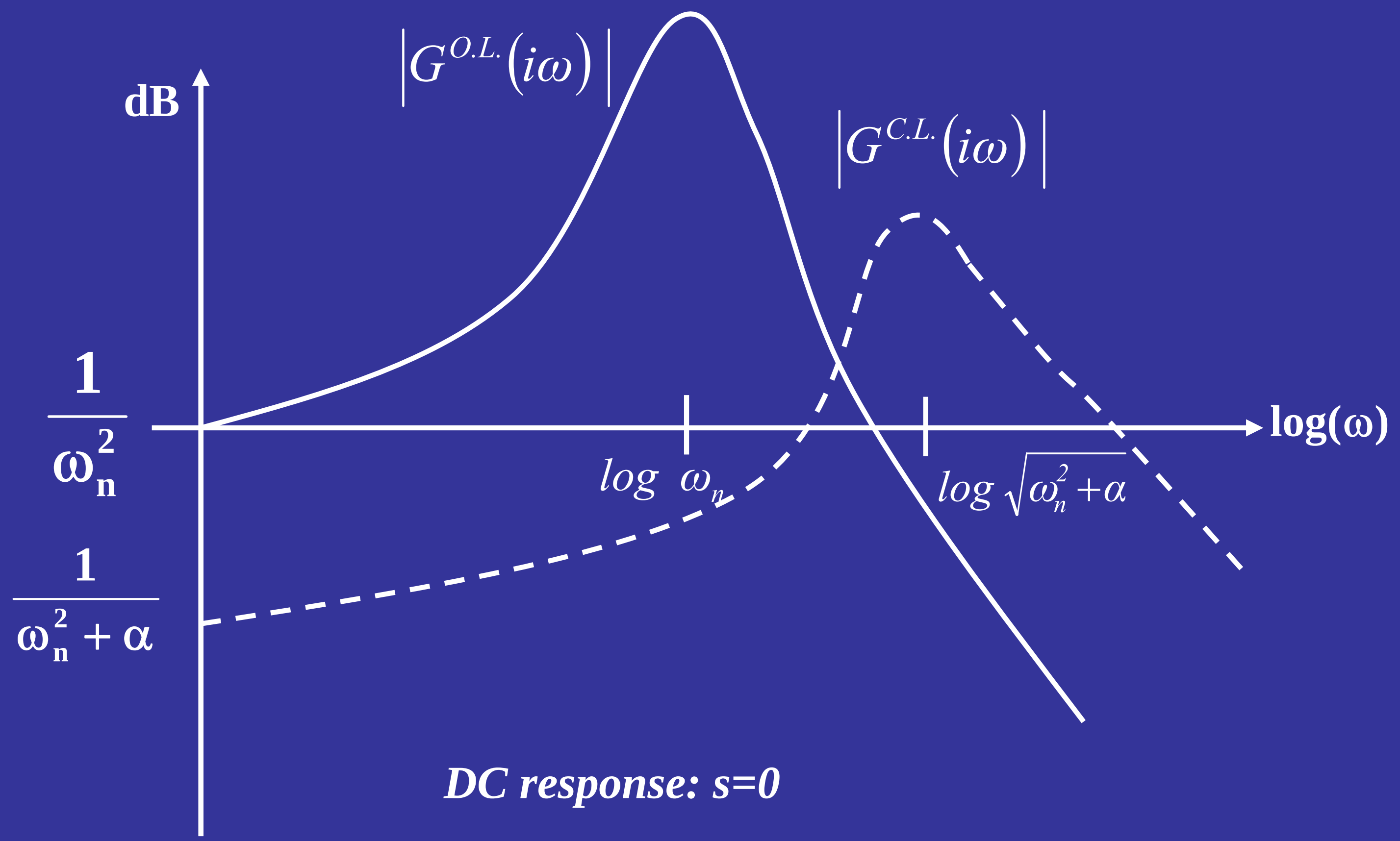


Then
$$\ddot{x} + \gamma\,\dot{x} + \omega_n^2 x = u - \alpha\,x$$
$$==> \ddot{x} + \gamma\,\dot{x} + \left(\omega_n^2 + \alpha\right)x = u$$

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

3.1 Feedback-Example 1

Now the closed loop T.F. is: $G^{C.L.}(s) = \dfrac{1}{s^2 + \gamma s + (\omega_n^2 + \alpha)}$



DC response: s=0

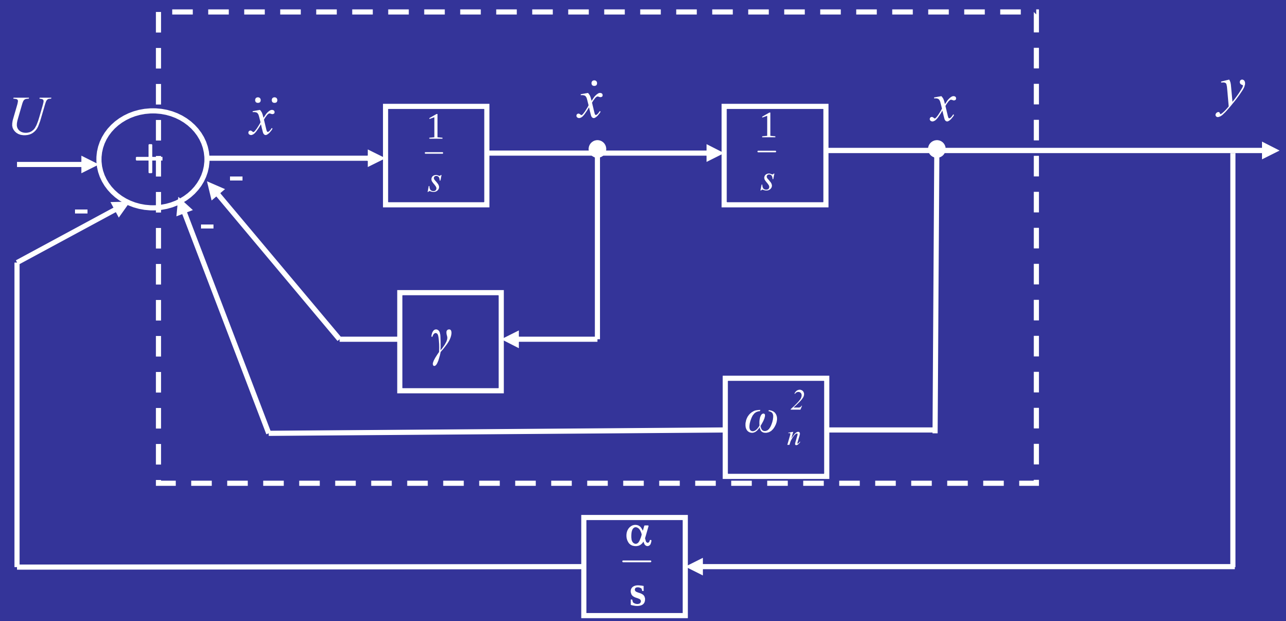
So the effect of the proportional feedback in this case is ***to increase the bandwidth of the system***
(and reduce gain slightly, but this can easily be compensated by adding a constant gain in front…)

3.1 Feedback-Example 2

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

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

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



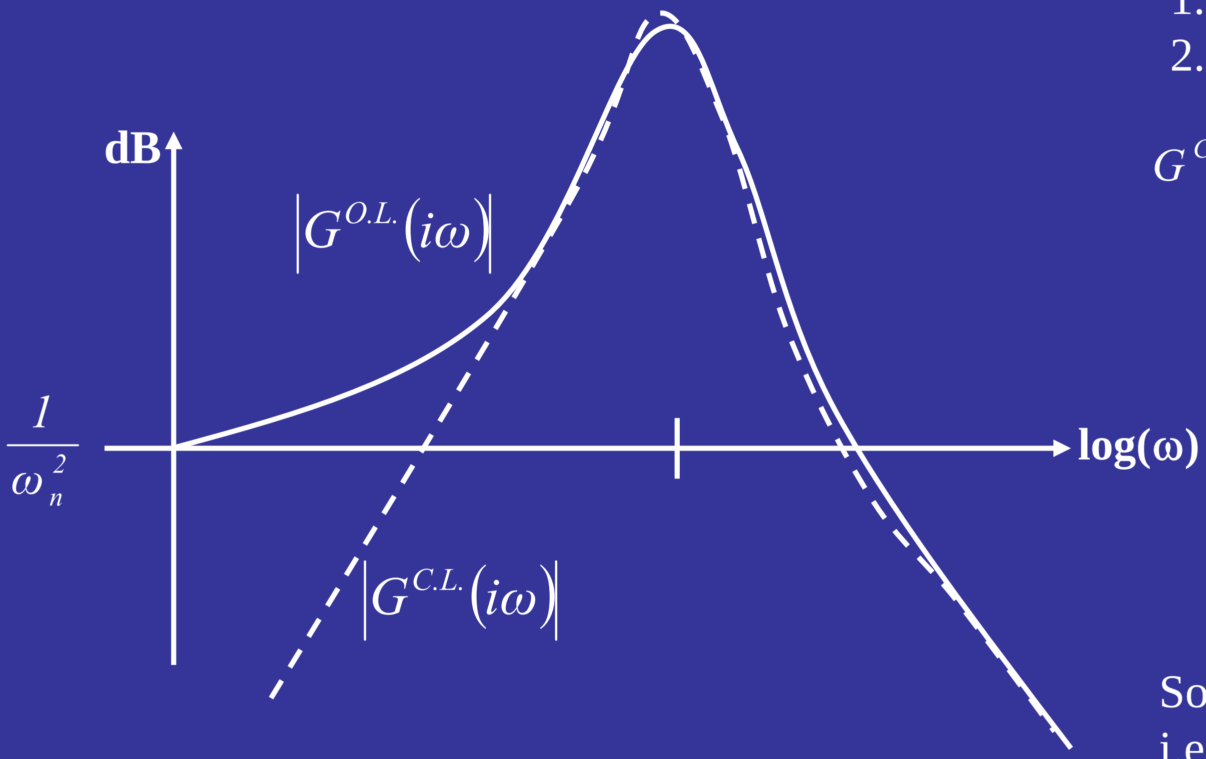
Differentiating once more yields: $\dddot{x} + \gamma \ddot{x} + \omega_n^2 \dot{x} + \alpha x = \dot{u}$

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)

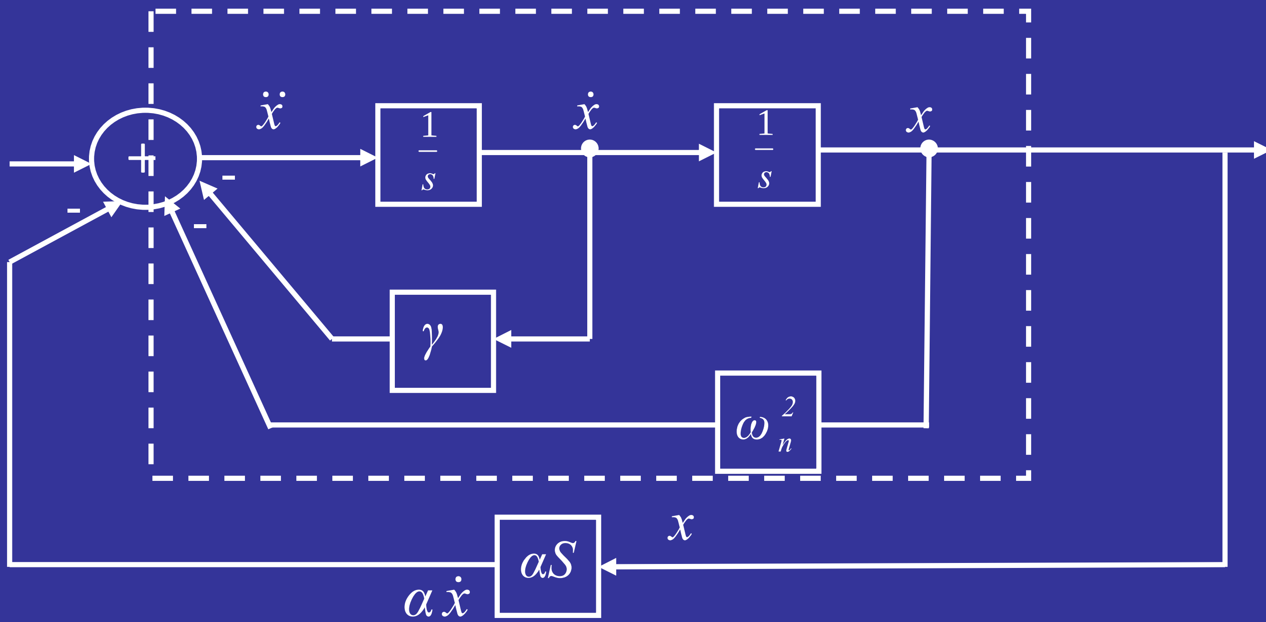
$$G^{C.L.}(s)\approx\frac{1}{\left(s^2+\gamma s+\omega_n^2\right)}\approx G^{O.L.}(s)$$

So integral feedback has killed DC gain i.e system *rejects constant* disturbances

3.1 Feedback-Example 3

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

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



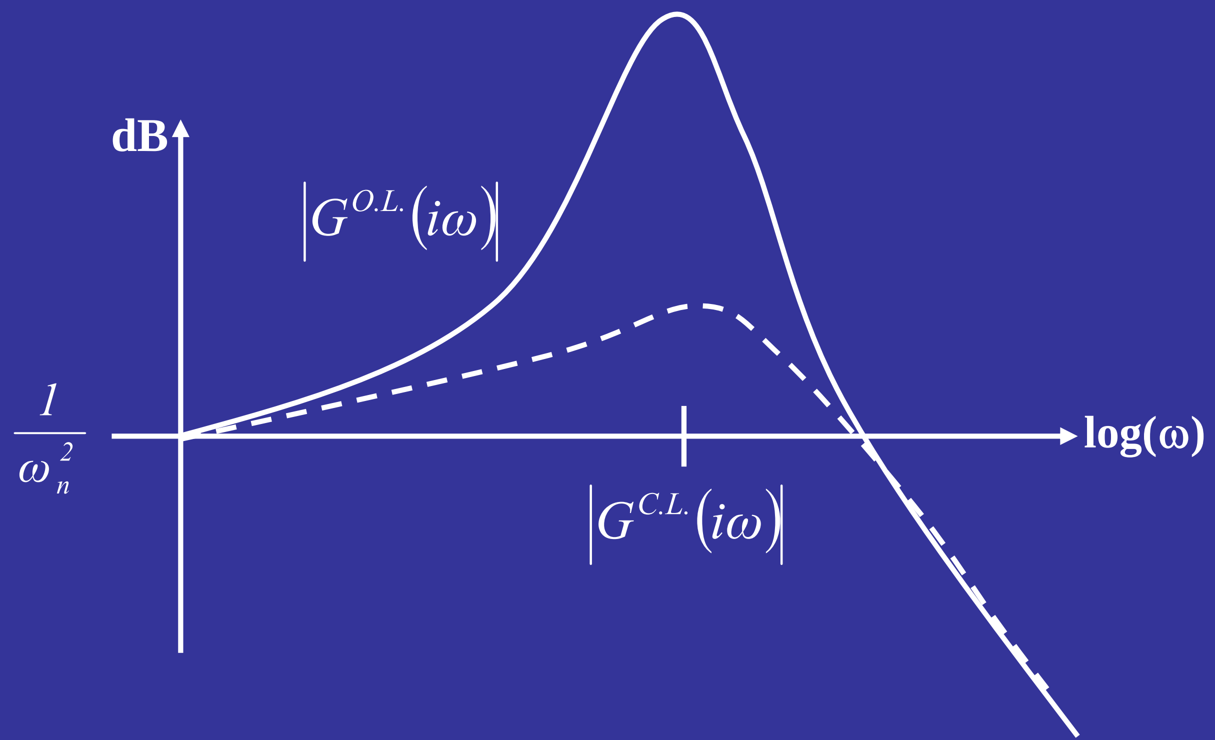
Now have

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

So effect off differential feedback is to increase damping

3.1 Feedback-Example 3

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

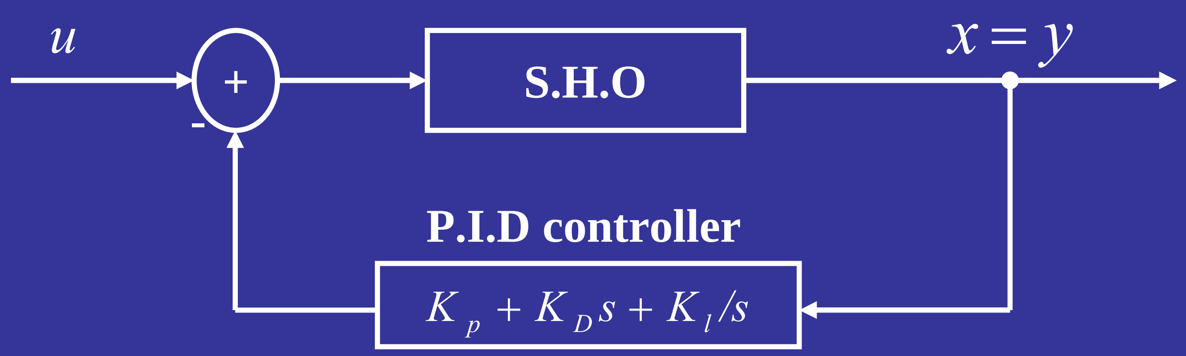


So the effect of differential feedback here is to "flatten the resonance" i.e. ***damping is increased***.

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).



$$u_{fb} = u_p + u_d + u_I$$

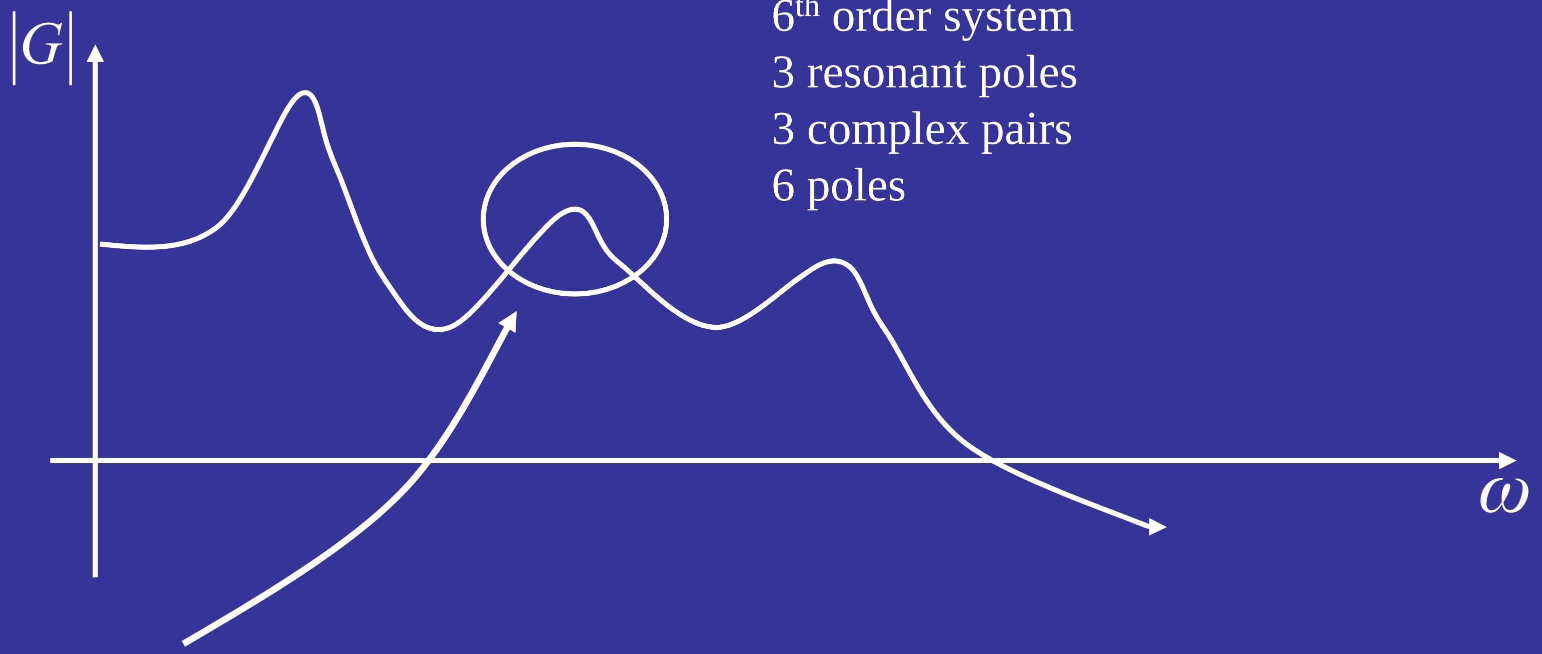
(2) In example above S.H.O. was a very simple system and it was clear what *physical interpretation* of P. or I. or D. did. But for *large complex systems* not obvious

==> **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.

$$u = -k_1 x_1 - \ldots.. - k_n x_n$$
$$= -k\,x$$

Where $k = [k_1 \ldots\ldots k_n]$ (row matrix)

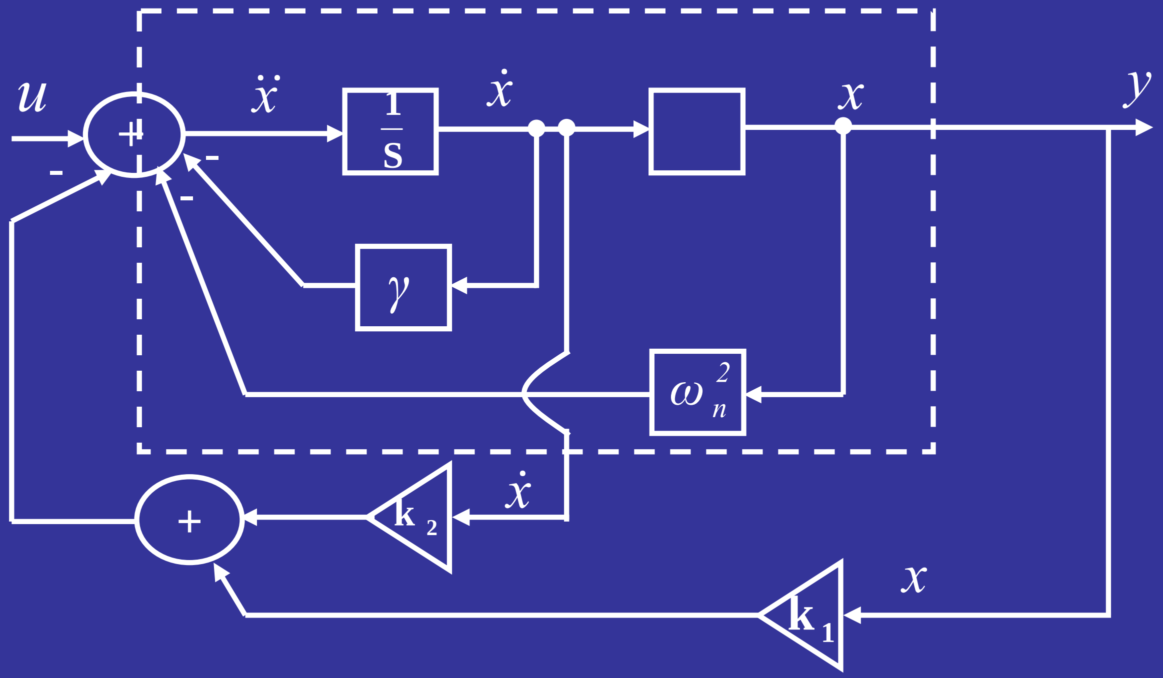
Where example: In S.H.O. examples

Proportional fbk : $u_p = -k_p x = -[k_p \;\; 0]$

Differential fbk : $u_D = -k_D \dot{x} = -[0 \;\; k_D]$

3.2 Full State Control

Example: Detailed block diagram of S.H.O with full-scale feedback

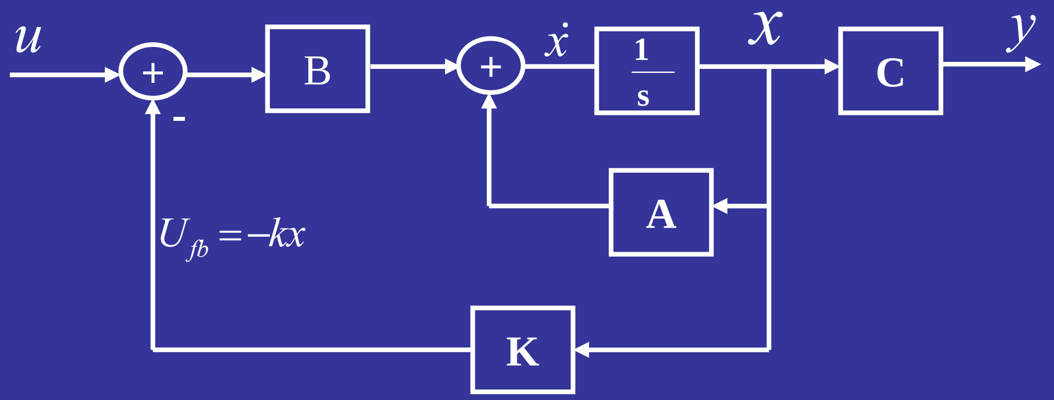


Of course this <u>assumes</u> we have access to the $\dot{x}$ state, which we actually *<u>Don't</u>* in practice.

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

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

$$= Ax + Bu + BKx$$

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

$$u_{fb} = -Kx$$

$$y = Cx$$

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