

# Systems & control basics

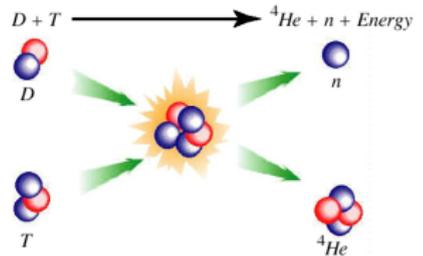
Gianmaria DE TOMMASI  
Email: detommas@unina.it

IST - Lisboa, Dec, 5th 2017

- 1 Brief introduction to plasma magnetic control in tokamaks
- 2 Preliminaries
  - Linear Systems
  - Transfer function
  - Block diagrams
  - Frequency response  $G(j\omega)$
  - Graphical representations of  $G(j\omega)$
- 3 Feedback Control Systems
  - The control problem
  - Stability margins
  - Nyquist Criterion
  - Root locus
- 4 Digital Control systems

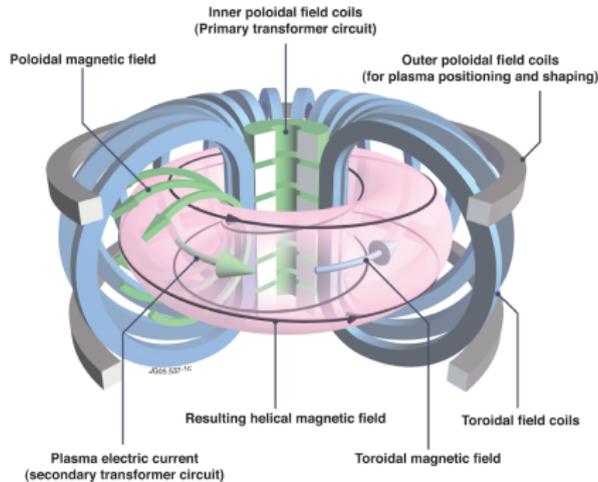
## Main Aim

Production of energy by means of a fusion reaction

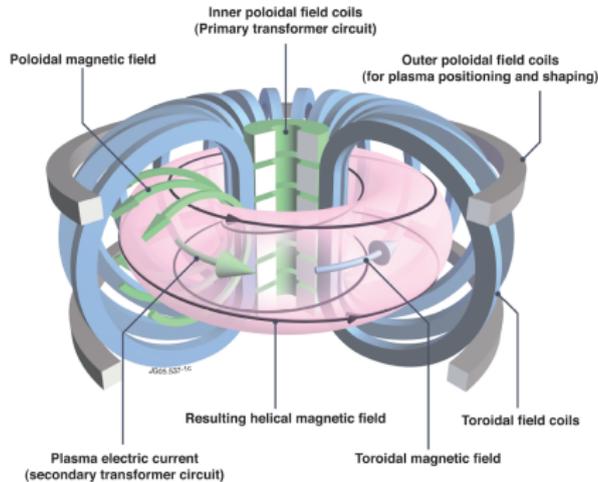


## Plasma

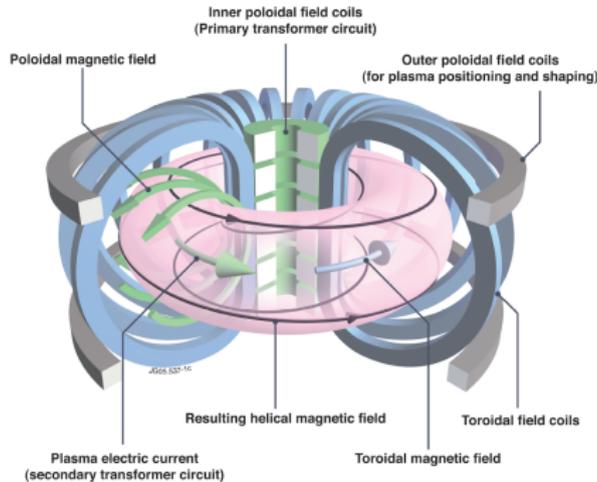
- High temperature and pressure are needed
- Fully ionised gas  $\mapsto$  Plasma
- Magnetic field is needed to confine the plasma



- In tokamaks, **magnetic control of the plasma is obtained by means of magnetic fields produced by the external active coils**



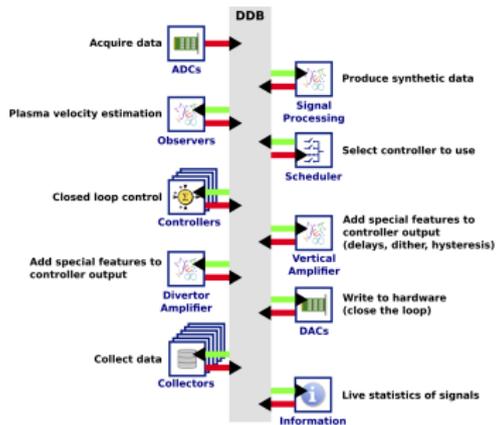
- In tokamaks, **magnetic control of the plasma is obtained by means of magnetic fields produced by the external active coils**
- In order to obtain good performance, it is necessary to have a plasma with **vertically elongated cross section**  $\Rightarrow$  **vertically unstable plasmas**



- In tokamaks, **magnetic control of the plasma is obtained by means of magnetic fields produced by the external active coils**
- In order to obtain good performance, it is necessary to have a plasma with **vertically elongated cross section**  $\Rightarrow$  **vertically unstable plasmas**
- It is important to **maintain adequate plasma-wall clearance during operation**

# The final objective

## Build a control system



- 1 The plasma and the surrounding conductive structures are assumed to be axisymmetric

# Main (basic) assumptions

- 1 The plasma and the surrounding conductive structures are assumed to be axisymmetric
- 2 The inertial effects can be neglected at the time scale of interest, since plasma mass density is low

- 1 The plasma and the surrounding conductive structures are assumed to be axisymmetric
- 2 The inertial effects can be neglected at the time scale of interest, since plasma mass density is low
- 3 The magnetic permeability  $\mu$  is homogeneous, and equal to  $\mu_0$  everywhere

- 1 The plasma and the surrounding conductive structures are assumed to be axisymmetric
- 2 The inertial effects can be neglected at the time scale of interest, since plasma mass density is low
- 3 The magnetic permeability  $\mu$  is homogeneous, and equal to  $\mu_0$  everywhere

## Mass vs Massless plasma

It has been proven that neglecting plasma mass may lead to erroneous conclusion on closed-loop stability.



M. L. Walker, D. A. Humphreys

On feedback stabilization of the tokamak plasma vertical instability  
*Automatica*, vol. 45, pp. 665–674, 2009.



J. W. Helton, K. J. McGown, M. L. Walker,

Conditions for stabilization of the tokamak plasma vertical instability using only a massless plasma analysis  
*Automatica*, vol. 46, pp. 1762.-1772, 2010.

The *input variables* are:

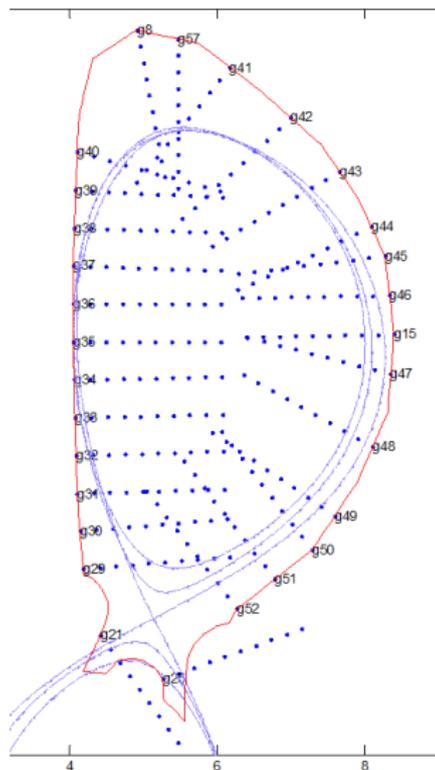
- The voltage applied to the active coils  $v$
- The plasma current  $I_p$
- The poloidal beta  $\beta_p$
- The internal inductance  $I_i$

$I_p$ ,  $\beta_p$  and  $I_i$

$I_p$ ,  $\beta_p$  and  $I_i$  are used to specify the plasma internal profiles for a given shape of these profiles (e.g., *bell-shaped* profiles)

Different outputs can be chosen:

- fluxes and fields where the magnetic sensors are located
- currents in the active and passive structures
- plasma radial and vertical position (1st and 2nd moment of the plasma current density)
- geometrical descriptors describing the plasma shape (gaps, x-point and strike point positions)



- Grad-Shafranov PDE can be solved by using finite-elements methods

- Grad-Shafranov PDE can be solved by using finite-elements methods
- From these numerical approximations it is possible retrieve a **nonlinear lumped parameters** model

$$\frac{d}{dt} [\mathcal{M}(\mathbf{y}(t), \beta_p(t), I_i(t)) \mathbf{I}(t)] + \mathbf{R} \mathbf{I}(t) = \mathbf{U}(t),$$
$$\mathbf{y}(t) = \mathcal{Y}(\mathbf{I}(t), \beta_p(t), I_i(t)).$$

where:

- $\mathbf{y}(t)$  are the output to be controlled
- $\mathbf{I}(t) = [\mathbf{I}_{PF}^T(t) \mathbf{I}_e^T(t) I_p(t)]^T$  is the currents vector, which includes the currents in the active coils  $\mathbf{I}_{PF}(t)$ , the eddy currents in the passive structures  $\mathbf{I}_e(t)$ , and the plasma current  $I_p(t)$
- $\mathbf{U}(t) = [\mathbf{U}_{PF}^T(t) \mathbf{0}^T 0]^T$  is the input voltages vector
- $\mathcal{M}(\cdot)$  is the mutual inductance nonlinear function
- $\mathbf{R}$  is the resistance matrix
- $\mathcal{Y}(\cdot)$  is the output nonlinear function

Starting from the nonlinear lumped parameters model, the following plasma linearized state space model can be easily obtained:

$$\delta\dot{\mathbf{x}}(t) = \mathbf{A}\delta\mathbf{x}(t) + \mathbf{B}\delta\mathbf{u}(t) + \mathbf{E}\delta\dot{\mathbf{w}}(t), \quad (1)$$

$$\delta\mathbf{y}(t) = \mathbf{C}\delta\mathbf{l}_{PF}(t) + \mathbf{F}\delta\mathbf{w}(t), \quad (2)$$

where:

- **A**, **B**, **E**, **C** and **F** are the model matrices
- $\delta\mathbf{x}(t) = [\delta\mathbf{l}_{PF}^T(t) \delta\mathbf{l}_e^T(t) \delta l_p(t)]^T$  is the state space vector
- $\delta\mathbf{u}(t) = [\delta\mathbf{U}_{PF}^T(t) \mathbf{0}^T 0]^T$  are the input voltages variations
- $\delta\mathbf{w}(t) = [\delta\beta_p(t) \delta l_i(t)]^T$  are the  $\beta_p$  and  $l_i$  variations
- $\delta\mathbf{y}(t)$  are the output variations

The model (1)–(2) relates the variations of the PF currents to the variations of the outputs around a given equilibrium

## State space model

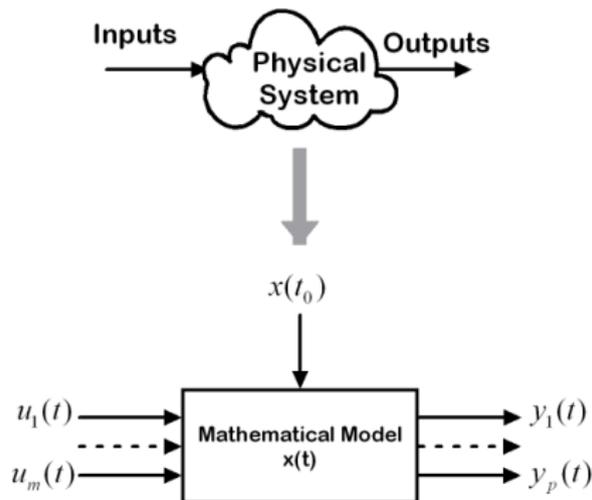
A **finite dimensional continuous-time** dynamical system can be described by the following differential equations:

$$\dot{x}(t) = f(x(t), u(t), t, t_0), x(t_0) = x_0 \quad (3a)$$

$$y(t) = h(x(t), u(t), t, t_0) \quad (3b)$$

where:

- $x(t) \in \mathbb{R}^n$  is the system **state**
- $x(t_0) \in \mathbb{R}^n$  is the **initial condition**
- $u(t) \in \mathbb{R}^m$  is the **input vector**
- $y(t) \in \mathbb{R}^p$  is the **output vector**
- $n$  is the **order** of the system



The state space model for a **linear time-invariant (LTI)** continuous-time system can be written as

$$\dot{x}(t) = Ax(t) + Bu(t), \quad x(0) = x_0 \quad (4a)$$

$$y(t) = Cx(t) + Du(t) \quad (4b)$$

where  $A \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times m}$ ,  $C \in \mathbb{R}^{p \times n}$  and  $D \in \mathbb{R}^{p \times m}$ .

A dynamical system with single-input ( $m = 1$ ) and single-output ( $p = 1$ ) is called *SISO*, otherwise it is called *MIMO*.

The state space model for a **linear time-invariant (LTI)** continuous-time system can be written as

$$\dot{x}(t) = Ax(t) + Bu(t), \quad x(0) = x_0 \quad (4a)$$

$$y(t) = Cx(t) + Du(t) \quad (4b)$$

where  $A \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times m}$ ,  $C \in \mathbb{R}^{p \times n}$  and  $D \in \mathbb{R}^{p \times m}$ .

A dynamical system with single-input ( $m = 1$ ) and single-output ( $p = 1$ ) is called *SISO*, otherwise it is called *MIMO*.

## Matlab commands

`sys = ss(A, B, C, D)` creates a state space model object.  
`y = lsim(sys, u, t)` simulates the time response of the LTI system `sys`.

Consider a nonlinear and **time-invariant** system

$$\dot{x}(t) = f(x(t), u(t)), \quad x(0) = x_0 \quad (5a)$$

$$y(t) = h(x(t), u(t)) \quad (5b)$$

If the input is constant, i.e.  $u(t) = \bar{u}$ , then the equilibrium states  $x_{e_1}, x_{e_2}, \dots, x_{e_q}$  of such a system can be computed as solutions of the homogeneous equation

$$f(x_e, \bar{u}) = 0,$$

Given an equilibrium state  $x_{e_i}$  the correspondent output is given by

$$y_{e_i} = h(x_{e_i}, \bar{u}).$$

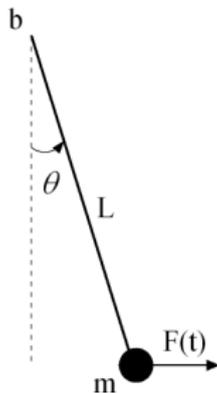
If  $x_0 = x_e + \delta x_0$  and  $u(t) = \bar{u} + \delta u(t)$ , with  $\delta x_0, \delta u(t)$  **sufficiently small**, then the behaviour of (5) around a given equilibrium point  $(\bar{u}, x_e)$  is well described by the linear system

$$\delta \dot{x}(t) = \left. \frac{\partial f}{\partial x} \right|_{\substack{x = x_e \\ u = \bar{u}}} \delta x(t) + \left. \frac{\partial f}{\partial u} \right|_{\substack{x = x_e \\ u = \bar{u}}} \delta u(t), \quad \delta x(0) = \delta x_0 \quad (6a)$$

$$\delta y(t) = \left. \frac{\partial h}{\partial x} \right|_{\substack{x = x_e \\ u = \bar{u}}} \delta x(t) + \left. \frac{\partial h}{\partial u} \right|_{\substack{x = x_e \\ u = \bar{u}}} \delta u(t) \quad (6b)$$

The total output can be computed as

$$y(t) = h(x_e, \bar{u}) + \delta y(t).$$



- Mass  $m$
- Length  $L$
- Friction coefficient  $b$

## Nonlinear model

Let

$$x(t) = \begin{pmatrix} \theta(t) \\ \dot{\theta}(t) \end{pmatrix} \quad u(t) = F(t) \quad y(t) = \theta(t)$$

then

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

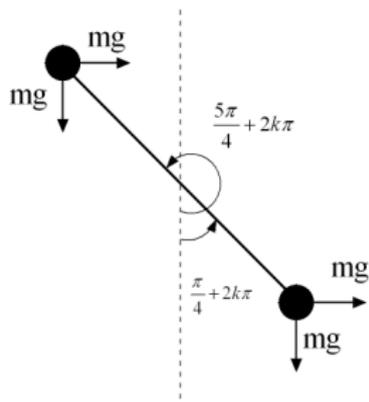
$$\dot{x}_2(t) = -\frac{g}{L} \sin x_1(t) - \frac{b}{mL^2} x_2(t) + \frac{1}{mL} \cos x_1(t) u(t)$$

$$y(t) = x_1(t)$$

## Equilibria

If  $\bar{u} = mg$ , solving  $f(x_e, \bar{u}) = 0$  we get

$$x_{e_k} = \begin{pmatrix} \frac{\pi}{4} + k\pi \\ 0 \end{pmatrix}$$



Around the equilibria  $x_e = (\frac{\pi}{4} \ 0)^T$  the behaviour of the pendulum is well described by the linear system

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

$$\delta \dot{x}_2(t) = -\frac{\sqrt{2}g}{L}\delta x_1(t) - \frac{b}{mL^2}\delta x_2(t) + \frac{1}{\sqrt{2}mL}\delta u(t)$$

$$\delta y(t) = \delta x_1(t)$$



```
% System parameters
m = 10;
b = 15;
g = 9.81;
L = 1;
% Equilibria
xbar = [pi/4 0];
ubar = m*g;
% Output at the equilibria
ybar = xbar(1);
% Variation of the initial conditions
dx0 = [0.3 3];
% Input variation: Am*sin(t)
Am = 10;
% Simulation time interval
tfin = 15;
```



```
% Linearized model matrices
A = [ 0 1 ; -sqrt(2)*g/L -b/(m*L^2) ];
B = [ 0 ; 1/(sqrt(2)*m*L) ];
C = [ 1 0 ];
D = 0;
% Linearized model
sys_l = ss(A,B,C,D);
% Time vector
tlin = 0:.001:tfin;
% Linear simulation
du = Am*sin(tlin);
ylin= lsim(sys_l,du,tlin,dx0);
```



```
% Nonlinear simulation
[ t,x_nl ]=ode45('p_nl',[0
tfin],xbar+dx0,[],m,b,g,L,ubar,Am);
y_nl = x_nl(:,1);
% Plots
figure(1)
plot(t,y_nl*180/pi,'- ',tlin,(ylin+ybar)*180/pi,'- ')
grid on
ylabel('[deg]')
xlabel('tempo [s]')
title('theta')
legend('NL','L')
```

```
function xdot = p_nl(t,x,flag,m,b,g,L,ubar,Am)
%
% Returns the state derivative
%
u = ubar + Am*sin(t);
xdot = [x(2); -g/L*sin(x(1)) - b/(m*L^2)*x(2) + ...
u/(m*L)*cos(x(1))];
```

[▶ Download Matlab example](#)

## Asymptotic stability

This property roughly asserts that every solution of  $\dot{x}(t) = Ax(t)$  tends to zero as  $t \rightarrow \infty$ .

**Note that for LTI systems the stability property is related to the system and not to a specific equilibrium.**

## Asymptotic stability

This property roughly asserts that every solution of  $\dot{x}(t) = Ax(t)$  tends to zero as  $t \rightarrow \infty$ .

**Note that for LTI systems the stability property is related to the system and not to a specific equilibrium.**

**Theorem** - System (4) is **asymptotically stable** iff  $A$  is Hurwitz, that is if every eigenvalue  $\lambda_i$  of  $A$  has strictly negative real part

$$\Re(\lambda_i) < 0, \forall \lambda_i.$$

**Theorem** - System (4) is **unstable** if  $A$  has at least one eigenvalue  $\bar{\lambda}$  with strictly positive real part, that is

$$\exists \bar{\lambda} \text{ s.t. } \Re(\bar{\lambda}) > 0.$$

**For nonlinear system the stability property is related to the specific equilibrium**

### For nonlinear system the stability property is related to the specific equilibrium

**Theorem** - The equilibrium state  $x_e$  corresponding to the constant input  $\bar{u}$  a nonlinear system (5) is **asymptotically stable** if all the eigenvalues of the correspondent linearized system (6) have strictly negative real part.

**Theorem** - The equilibrium state  $x_e$  corresponding to the constant input  $\bar{u}$  a nonlinear system (5) is **unstable** if there exists at least one eigenvalue of the correspondent linearized system (6) which has strictly positive real part.

Given a LTI system (4) the corresponding *transfer matrix* from  $u$  to  $y$  is defined as

$$G(s) = C(sI - A)^{-1}B + D, \quad (7)$$

where  $s \in \mathbb{C}$ . If we denote with  $U(s)$  and  $Y(s)$  the Laplace transforms of  $u(t)$  and  $y(t)$ , then it is

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

when the initial condition of system (4) is  $x(0) = 0$ .

For SISO system (7) is called *transfer function* and it is equal to the Laplace transform of the **impulsive response** of system (4) with zero initial condition.

## Matlab commands

`sys = tf(num, den)` creates a transfer function object.

Given the transfer function  $G(s)$  and the Laplace transform of the input  $U(s)$  the time response of the system can be computed as the inverse transform of  $G(s)U(s)$ , without solving differential equations.

As an example, the **step response** of a system can be computed as:

$$y(t) = \mathcal{L}^{-1} \left[ G(s) \frac{1}{s} \right].$$

## Matlab commands

`[y,t] = step(sys)` computes the step response of the LTI system `sys`.

`[y,t] = impulse(sys)` computes the impulse response of the LTI system `sys`.

Given a SISO LTI system, its transfer function is a rational function of  $s$

$$G(s) = \frac{N(s)}{D(s)} = \rho \frac{\prod_i (s - z_i)}{\prod_j (s - p_j)},$$

where  $N(s)$  and  $D(s)$  are polynomial in  $s$ , with  $\deg(N(s)) \leq \deg(D(s))$ .

We call

- $p_j$  **poles** of  $G(s) \rightarrow$  roots of  $D(s)$
- $z_i$  **zeros** of  $G(s) \rightarrow$  roots of  $N(s)$

## Matlab commands

**sys = zpk(z,p,k)** creates a zeros-poles-gain object.

**p = eig(sys)** or **p = pole(sys)** return the poles of the LTI system **sys**.

**z = zero(sys)** returns the zeros of the LTI system **sys**.



**Each pole of  $G(s)$  is an eigenvalue of the system matrix  $A$ , while the converse is not necessarily true**

**Each pole of  $G(s)$  is an eigenvalue of the system matrix  $A$ , while the converse is not necessarily true**

If all the poles of  $G(s)$  have strictly negative real part – i.e. they are located in the left half of the  $s$ -plane (LHP) – the SISO system is said to be **Bounded–Input Bounded–Output stable (BIBO)**

A system is BIBO stable if bounded input to the system results in a bounded output over the time interval  $[0, +\infty)$

# Time constants, natural frequencies and damping factors

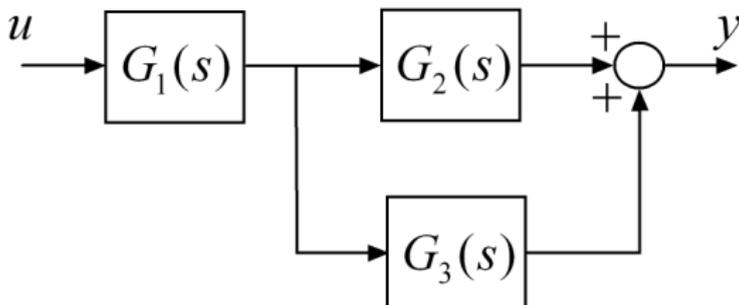


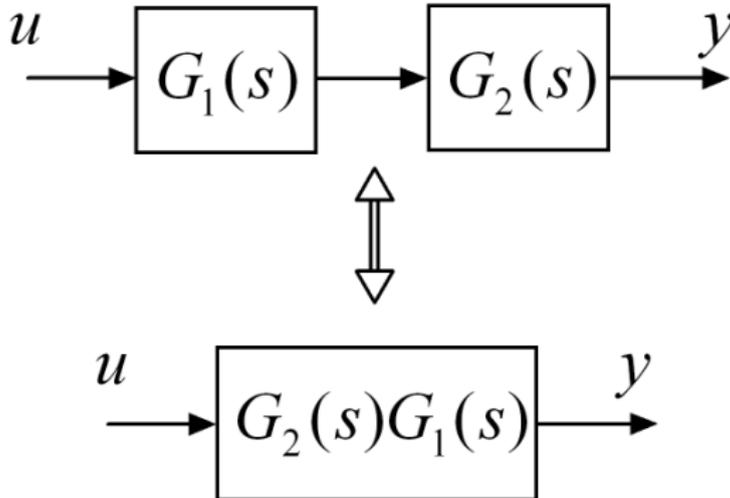
A transfer function can be also specified in terms of

- time constants ( $\tau, T$ )
- natural frequencies ( $\omega_n, \alpha_n$ )
- damping factors ( $\xi, \zeta$ )
- gain ( $\mu$ )
- system type (i.e. number of poles/zeros in 0,  $g$ )

$$G(s) = \mu \frac{\prod_i (1 + T_i s) \prod_j \left( 1 + 2 \frac{\zeta_j}{\alpha_{n_j}} s + \frac{s^2}{\alpha_{n_j}} \right)}{s^g \prod_k (1 + \tau_k s) \prod_l \left( 1 + 2 \frac{\xi_l}{\omega_{n_l}} s + \frac{s^2}{\omega_{n_l}} \right)}$$

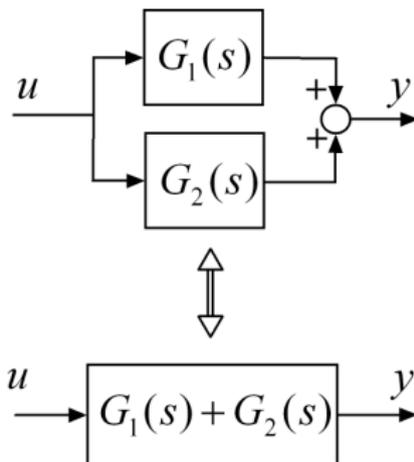
When dealing with transfer functions, it is usual to resort to **block diagrams** which permit to graphically represent the interconnections between systems in a convenient way





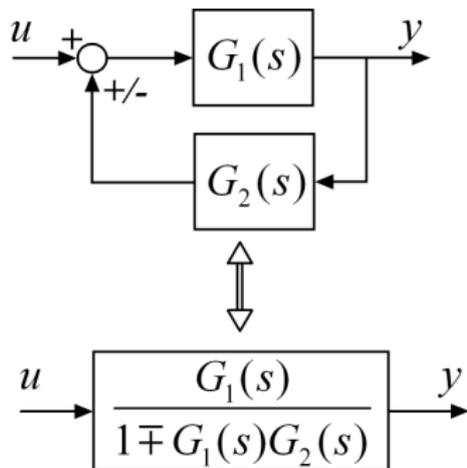
## Matlab commands

**sys = series(sys1,sys2)** or **sys = sys2\*sys1** make the series interconnection between **sys1** and **sys2**.



## Matlab commands

**sys = parallel(sys1,sys2)** or **sys = sys1+sys2**  
make the parallel interconnection between **sys1** and **sys2**.



## Matlab commands

`sys = feedback(sys1, sys2, [+1])` makes the feedback interconnection between `sys1` and `sys2`. Negative feedback is the default. If the third parameter is equal to `+1` positive feedback is applied.

Given two **asymptotically stable** LTI systems  $G_1(s)$  and  $G_2(s)$

- the **series** connection  $G_2(s)G_1(s)$  is asymptotically stable
- the **parallel** connection  $G_1(s) + G_2(s)$  is asymptotically stable
- the **feedback** connection  $\frac{G_1(s)}{1 \pm G_1(s)G_2(s)}$  **is not necessarily stable**

- Given two **asymptotically stable** LTI systems  $G_1(s)$  and  $G_2(s)$
- the **series** connection  $G_2(s)G_1(s)$  is asymptotically stable
  - the **parallel** connection  $G_1(s) + G_2(s)$  is asymptotically stable
  - the **feedback** connection  $\frac{G_1(s)}{1 \pm G_1(s)G_2(s)}$  **is not necessarily stable**

## THE CURSE OF FEEDBACK!

Given a LTI system the complex function

$$G(j\omega) = C(j\omega I - A)^{-1}B + D,$$

with  $\omega \in \mathbb{R}^+$  is called *frequency response* of the system.

$G(j\omega)$  permits to evaluate the system steady-state response to a sinusoidal input. In particular if

$$u(t) = A \sin(\bar{\omega}t + \varphi),$$

then the steady-state response of a LTI system is given by

$$y(t) = |G(j\bar{\omega})|A \sin(\bar{\omega}t + \varphi + \angle G(j\bar{\omega})).$$

Given a LTI system  $G(s)$  the Bode diagrams plot

- the magnitude of  $G(j\omega)$  (in dB,  $|G(j\omega)|_{dB} = 20 \log_{10}|G(j\omega)|$ )
  - and the phase of  $G(j\omega)$  (in degree)
- as a function of  $\omega$  (in rad/s) in a semi-log scale (base 10).

**Bode plots are used for both analysis and synthesis of control systems.**

## Matlab commands

**bode (sys)** plots the the Bode diagrams of the LTI system **sys**.

**bodemag (sys)** plots the Bode magnitude diagram of the LTI system **sys**.

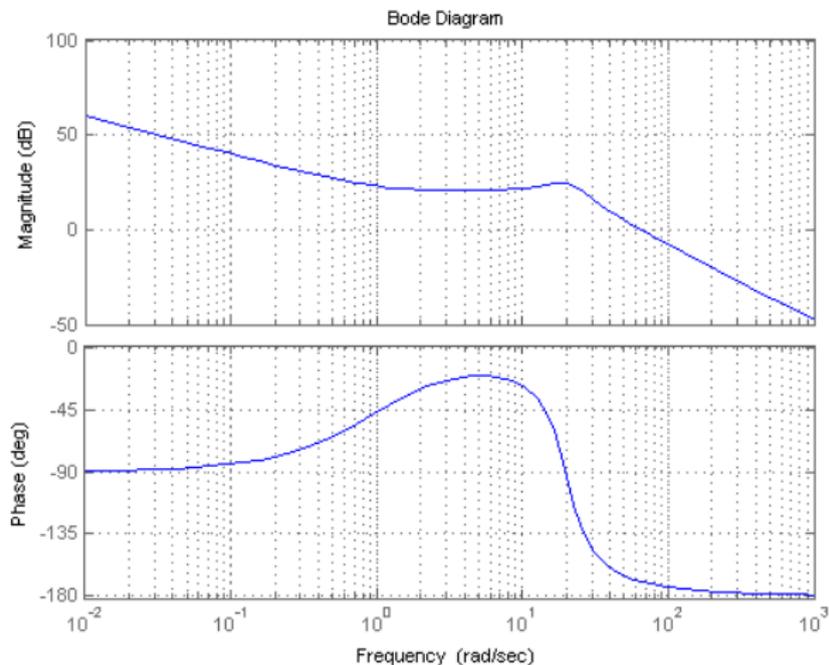
Let consider

$$G(s) = 10 \frac{1 + s}{s \left( \frac{s^2}{400} + 2 \frac{0.3}{20} s + 1 \right)} = 10 \frac{1 + s}{s(0.0025s^2 + 0.03s + 1)}$$

## Matlab commands

```
s = tf('s');  
sys = 10*(1+s)/(s*(s^2/400+0.6*s/20+1));  
bode(sys);  
grid
```

# Bode plot - Example



A stable system is said to be a **minimum phase** system if it **has not time delays or right-half plane (RHP) zeros**.

For minimum phase systems there is a unique relationship between the gain and phase of the frequency response  $G(j\omega)$ . This may be quantified by the *Bode's gain-phase relationship*

$$\angle G(j\bar{\omega}) = \frac{1}{\pi} \int_{-\infty}^{+\infty} \frac{d \ln |G(j\omega)|}{d \ln \omega} \ln \left| \frac{\omega + \bar{\omega}}{\omega - \bar{\omega}} \right| \frac{d\omega}{\omega}.$$

The name minimum phase refers to the fact that such a system has the minimum possible phase lag for the given magnitude response  $|G(j\omega)|$ .

The Nyquist plot is a polar plot of the frequency response  $G(j\omega)$  on the complex plane.

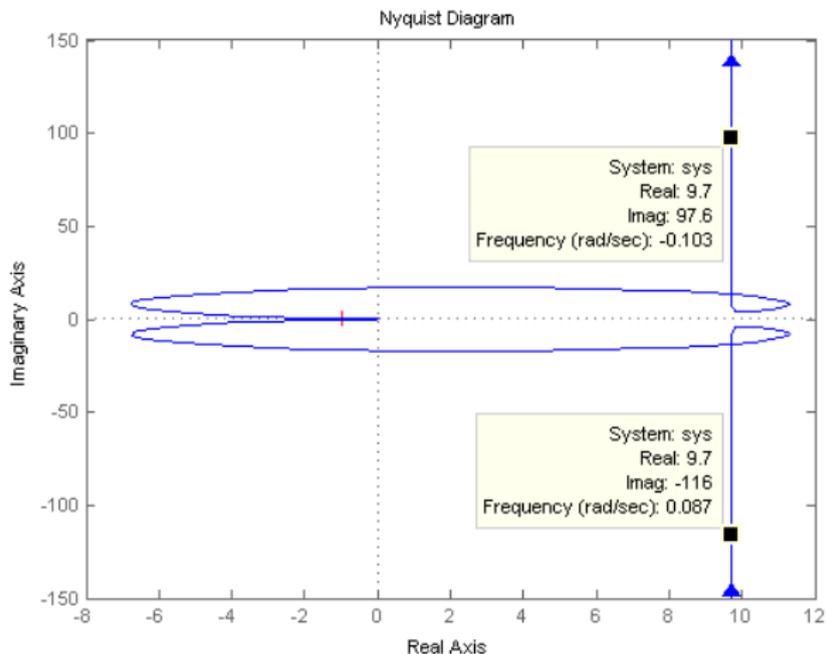
This plot combines the two Bode plots - magnitude and phase - on a single graph, with frequency  $\omega$ , which ranges in  $(-\infty, +\infty)$ , as a parameter along the curve.

**Nyquist plots are useful to check stability of closed-loop systems** (see [Nyquist stability criterion](#)).

## Matlab commands

`nyquist(sys)` plots the Nyquist plot of the LTI system `sys`.

# Nyquist plot - Example



It is similar to the Nyquist plot, since it plots both the magnitude and the phase of  $G(j\omega)$  on a single chart, with frequency  $\omega$  as a parameter along the curve.

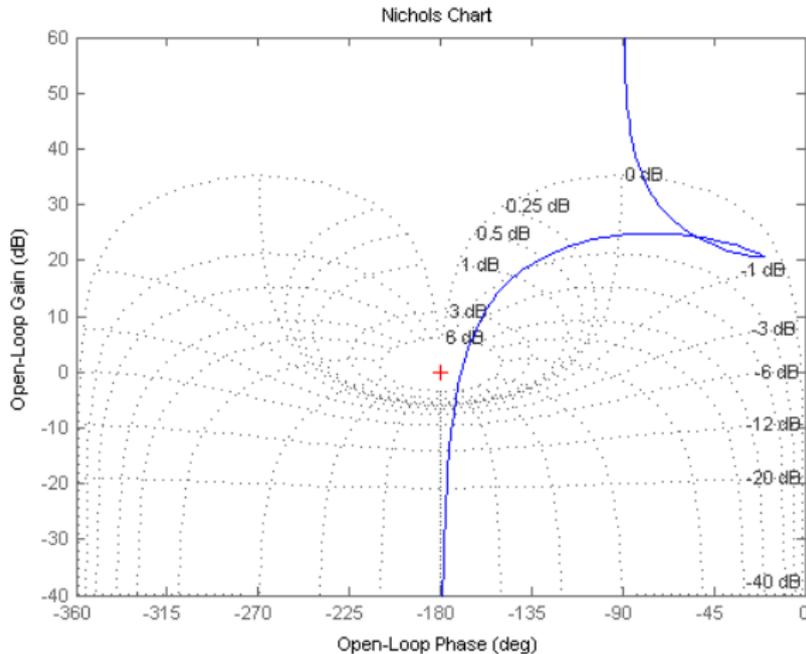
As for the Bode plot the magnitude  $|G(j\omega)|$  is expressed in dB and the phase  $\angle G(j\omega)$  in degree.

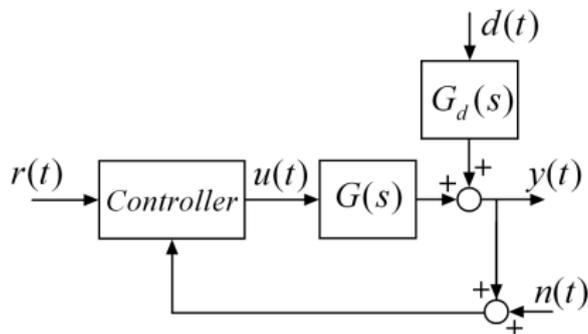
**Nichols charts are useful for the design of control systems, in particular for the design of lead, lag, lead-lag compensators.**

## Matlab commands

`nichols(sys)` plots the Nichols chart of the LTI system `sys`.

# Nichols plot - Example

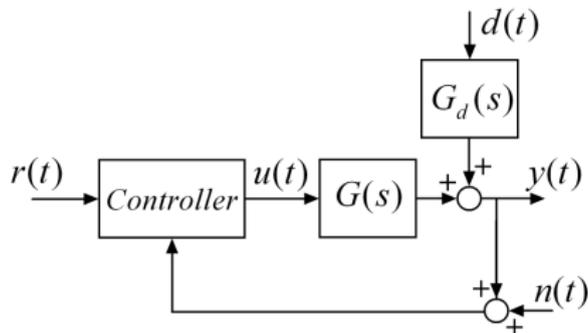




The objective of a control system is to make the output of a plant  $y(t)$  behave in a desired way by manipulating the plant input  $u(t)$

A good controller should manipulate  $u(t)$  so as to

- counteract the effect of a disturbance  $d(t)$  (**regulator problem**)
- keep the output close to a given reference input  $r(t)$  (**servo problem**)



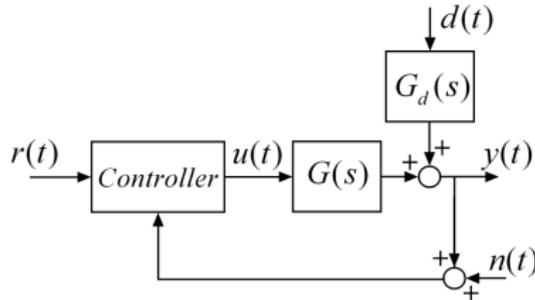
The objective of a control system is to make the output of a plant  $y(t)$  behave in a desired way by manipulating the plant input  $u(t)$

A good controller should manipulate  $u(t)$  so as to

- counteract the effect of a disturbance  $d(t)$  (**regulator problem**)
- keep the output close to a given reference input  $r(t)$  (**servo problem**)

**In both cases we want the control error  $e(t) = y(t) - r(t)$  to be small**

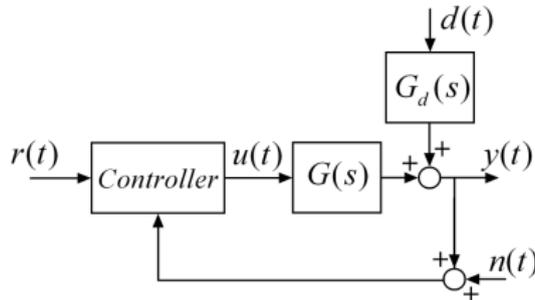
# Why feedback ?



The main sources of difficulty in achieving the control objectives are that

- 1 the plant model  $G(s)$  and the disturbance model  $G_d(s)$  may be affected by uncertainty and/or may change with time
- 2 the disturbance is not always measurable
- 3 the plant can be unstable

# Why feedback ?

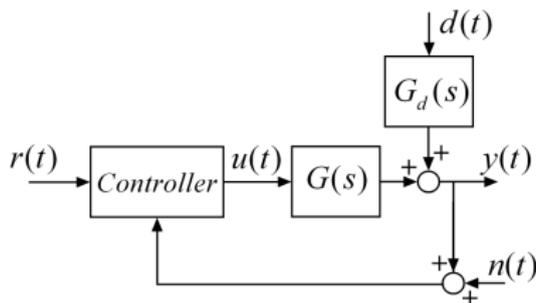


The main sources of difficulty in achieving the control objectives are that

- 1 the plant model  $G(s)$  and the disturbance model  $G_d(s)$  may be affected by uncertainty and/or may change with time
- 2 the disturbance is not always measurable
- 3 the plant can be unstable

**It turns out that open-loop and/or feed-forward approaches are not robust enough and/or are not always viable solutions**

A feedback approach can guarantee the desired degree of robustness. However design a feedback control system is not straightforward:



The main sources of difficulty in achieving the control objectives are that

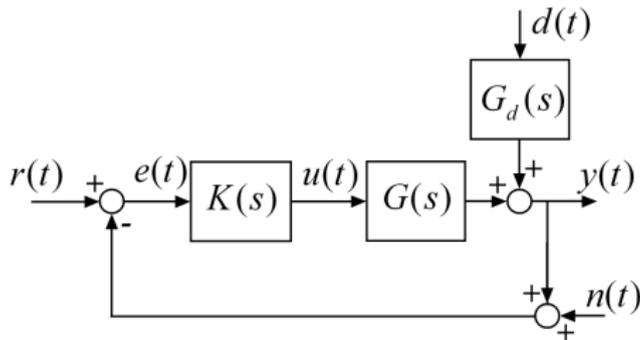
- 1 the plant model  $G(s)$  and the disturbance model  $G_d(s)$  may be affected by uncertainty and/or may change with time
- 2 the disturbance is not always measurable
- 3 the plant can be unstable

**It turns out that open-loop and/or feed-forward approaches are not robust enough and/or are not always viable solutions**

A feedback approach can guarantee the desired degree of robustness. However design a feedback control system is not straightforward: **instability is around the corner!**

A good controller must guarantee:

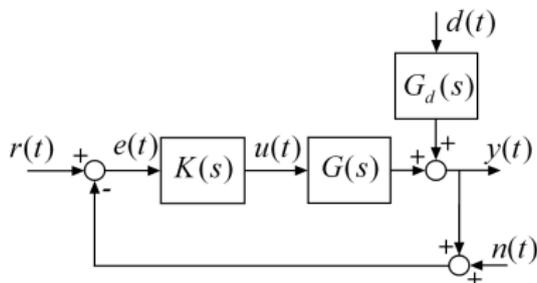
- **Nominal stability** - The closed loop system is stable when the nominal (without uncertainty) model is considered
- **Nominal Performance** - The closed loop system satisfies the performance specifications when the nominal model is considered
- **Robust stability** - The closed loop system is stable for all perturbed plants (i.e., taking account uncertainty)
- **Robust performance** - The closed loop system satisfies the performance specifications for all perturbed plants



The input to the plant is given by

$$U(s) = K(s)(R(s) - Y(s) - N(s)).$$

The objective of control is to design a *controller*  $K(s)$  such that the control error  $e(t) = r(t) - y(t)$  remains small in spite of the disturbance  $d(t)$ .



- $L(s) = G(s)K(s)$  is called **loop transfer function**
- $S(s) = (I + L(s))^{-1}$  is called **sensitivity function**
- $T(s) = (I + L(s))^{-1} L(s)$  is called **complementary sensitivity function**

It is straightforward to note that

$$T(s) + S(s) = I.$$

Exploiting the composition rules for block diagrams, it turns out that

$$Y(s) = T \cdot R(s) + SG_d \cdot D(s) - T \cdot N(s) \quad (8a)$$

$$E(s) = -S \cdot R(s) + SG_d \cdot D(s) - T \cdot N(s) \quad (8b)$$

$$U(s) = KS \cdot R(s) - K(s)S(s)G_d \cdot D(s) - KS \cdot N(s) \quad (8c)$$

**Remark** -  $S(s)$  is called sensitivity because it gives the relative sensitivity of the closed-loop transfer function  $T(s)$  to the relative plant model error. In particular, **at a given frequency**  $\omega$  we have for a SISO plant that

$$\frac{\frac{dT}{T}}{\frac{dG}{G}} = S.$$

**Remark** - Equations (8) are written in matrix form because they apply to MIMO systems. For SISO systems we may write

$$S(s) = \frac{1}{1 + L(s)},$$

$$T(s) = \frac{L(s)}{1 + L(s)}.$$

Let consider

$$Y(s) = T \cdot R(s) + SG_d \cdot D(s) - T \cdot N(s).$$

- In order to reduce the effect of the disturbance  $d(t)$  on the output  $y(t)$ , the sensitivity function  $S(s)$  should be made small (particularly in the *low frequency* range)
- In order to reduce the effect of the measurement noise  $n(t)$  on the output  $y(t)$ , the complementary sensitivity function  $T(s)$  should be made small (particularly in the *high frequency* range)

However, for all frequencies it is

$$T + S = I.$$

Let consider

$$Y(s) = T \cdot R(s) + SG_d \cdot D(s) - T \cdot N(s).$$

- In order to reduce the effect of the disturbance  $d(t)$  on the output  $y(t)$ , the sensitivity function  $S(s)$  should be made small (particularly in the *low frequency* range)
  - In order to reduce the effect of the measurement noise  $n(t)$  on the output  $y(t)$ , the complementary sensitivity function  $T(s)$  should be made small (particularly in the *high frequency* range)
- However, for all frequencies it is

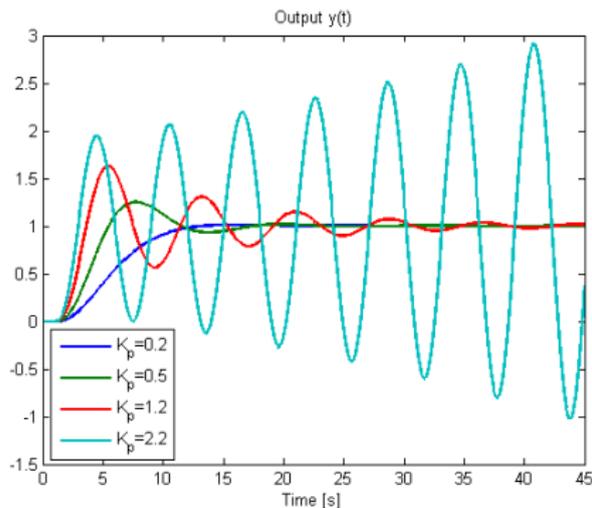
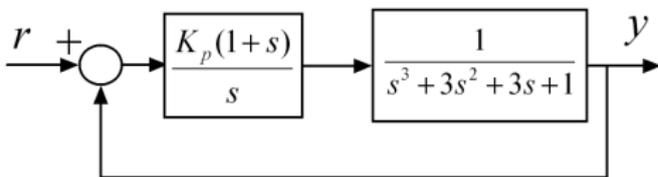
$$T + S = I.$$

**Thus a trade-off solution must be achieved.**

# Feedback may cause instability

One of the main issues in designing feedback controllers is stability.

If the feedback gain is too large then the controller may *overreact* and the closed-loop system becomes unstable.

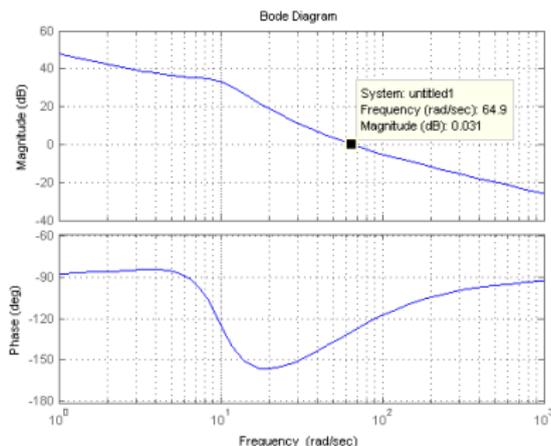


► Download Simulink example

Usually the frequency response of loop transfer function  $|L(j\omega)|$  has a low-pass behaviour.

The **crossover frequency**  $\omega_c$  is the frequency such that  $|L(j\omega_c)| = 1$ .

**In most of the cases the crossover frequency is a good estimation of the closed-loop bandwidth**



The frequency response of the loop transfer function  $L(j\omega)$  can be used to estimate the **stability margins**.

## Gain margin (GM)

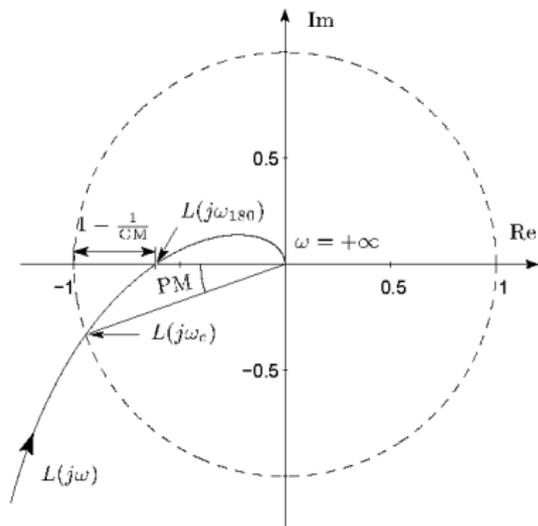
$$1/|L(j\omega_{180})|,$$

where  $\omega_{180}$  is the *phase crossover frequency*.

## Phase margin (PM)

$$\angle L(j\omega_c) + 180\text{deg},$$

where  $\omega_c$  is the *crossover frequency*.



The GM is the factor by which the loop gain  $|L(j\omega)|$  may be increased before the closed-loop system becomes unstable.

The GM is thus a direct safeguard against steady-state gain uncertainty.

The phase margin tells how much phase lag can be added to  $L(s)$  at frequency  $\omega_c$  before the phase at this frequency becomes 180 deg which corresponds to closed-loop instability (see [▶ Nyquist stability criterion](#)).

The PM is a direct safeguard against time delay uncertainty.

The Nyquist Criterion permits to check the stability of a closed loop system by using the Nyquist plot of the loop frequency response  $L(j\omega)$ .

The criterion is based on the fact the the *close-loop poles are equal to the zeros* of the transfer function

$$D(s) = 1 + L(s).$$

Hence, if  $D(s)$  has at least one zero  $\bar{z}$  such that  $\Re(\bar{z}) > 0$  the closed-loop system is unstable.

Consider a loop frequency response  $L(j\omega)$  and let

- $P$  be the number of poles of  $L(s)$  with strictly positive real part
- $Z$  be the number of zeros of  $L(s)$  with strictly positive real part

The Nyquist plot of  $L(j\omega)$  makes a number of encirclements  $N$  (**clockwise**) about the point  $(-1, j0)$  equal to

$$N = Z - P.$$

It turns out that the closed-loop system is asymptotically stable **if and only if** the Nyquist plot of  $L(j\omega)$  encircles (**counter clockwise**) the point  $(-1, j0)$  a number of times equal to  $P$ .

**The criterion is valid if the Nyquist plot of  $L(j\omega)$  do not intersect the point  $(-1, j0)$ .**



- 1 If the loop transfer function  $L(s)$  has a zero pole of multiplicity  $l$ , then the Nyquist plot has a discontinuity at  $\omega = 0$ . Further analysis indicates that the zero poles should be neglected, hence if there are no other unstable poles, then the loop transfer function  $L(s)$  should be considered stable, i.e.  $P = 0$ .

- 1 If the loop transfer function  $L(s)$  has a zero pole of multiplicity  $l$ , then the Nyquist plot has a discontinuity at  $\omega = 0$ . Further analysis indicates that the zero poles should be neglected, hence if there are no other unstable poles, then the loop transfer function  $L(s)$  should be considered stable, i.e.  $P = 0$ .
- 2 If the loop transfer function  $L(s)$  is stable, then the closed-loop system is unstable for any encirclement (clockwise) of the point  $-1$ .

- 1** If the loop transfer function  $L(s)$  has a zero pole of multiplicity  $l$ , then the Nyquist plot has a discontinuity at  $\omega = 0$ . Further analysis indicates that the zero poles should be neglected, hence if there are no other unstable poles, then the loop transfer function  $L(s)$  should be considered stable, i.e.  $P = 0$ .
- 2** If the loop transfer function  $L(s)$  is stable, then the closed-loop system is unstable for any encirclement (clockwise) of the point  $-1$ .
- 3** If the loop transfer function  $L(s)$  is unstable, then there must be one counter clockwise encirclement of  $-1$  for each pole of  $L(s)$  in the right-half of the complex plane.

- 1 If the loop transfer function  $L(s)$  has a zero pole of multiplicity  $l$ , then the Nyquist plot has a discontinuity at  $\omega = 0$ . Further analysis indicates that the zero poles should be neglected, hence if there are no other unstable poles, then the loop transfer function  $L(s)$  should be considered stable, i.e.  $P = 0$ .
- 2 If the loop transfer function  $L(s)$  is stable, then the closed-loop system is unstable for any encirclement (clockwise) of the point  $-1$ .
- 3 If the loop transfer function  $L(s)$  is unstable, then there must be one counter clockwise encirclement of  $-1$  for each pole of  $L(s)$  in the right-half of the complex plane.
- 4 If the Nyquist plot of  $L(j\omega)$  intersect the point  $(-1, j0)$ , then deciding upon even the marginal stability of the system becomes difficult and the only conclusion that can be drawn from the graph is that there exist poles on the imaginary axis.

The time behaviour of a closed-loop system is strictly related to the position of its poles on the complex plane.

For example, for a second order closed-loop system it is possible to relate the features of the step response such as

- rise time
- overshoot
- settling time

to the location of its poles.

The Root Locus design method permits to evaluate how changes in the loop transfer function  $L(s)$  affect the position of the closed-loop poles.

The closed-loop poles are given by the roots of

$$1 + L(s) = 0 \quad (9)$$

Assuming that  $L(s) = \rho L'(s)$  the Root Locus plots the locus of all possible roots of (9) as  $\rho$  varies in the range  $[0, \infty)$ .

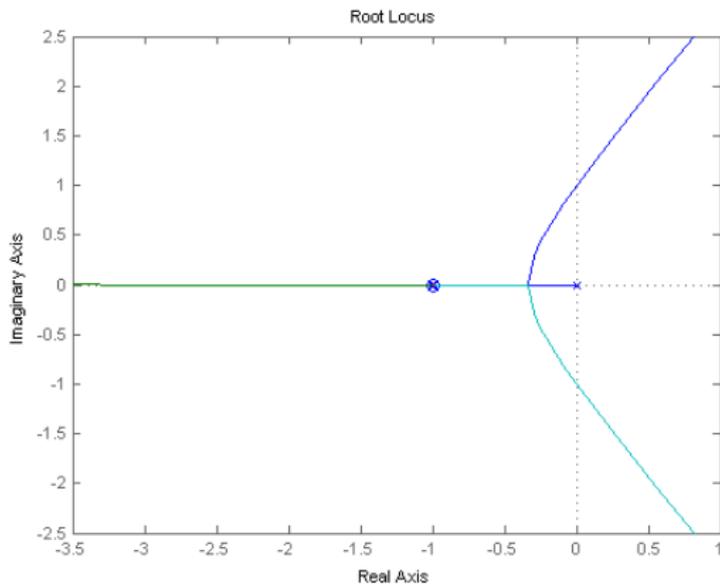
The Root Locus can be used to study the effect of additional poles and zeros in  $L'(s)$ , i.e. in the controller  $K(s)$ .

The Root Locus can be effectively used to design SISO controllers.

## Matlab commands

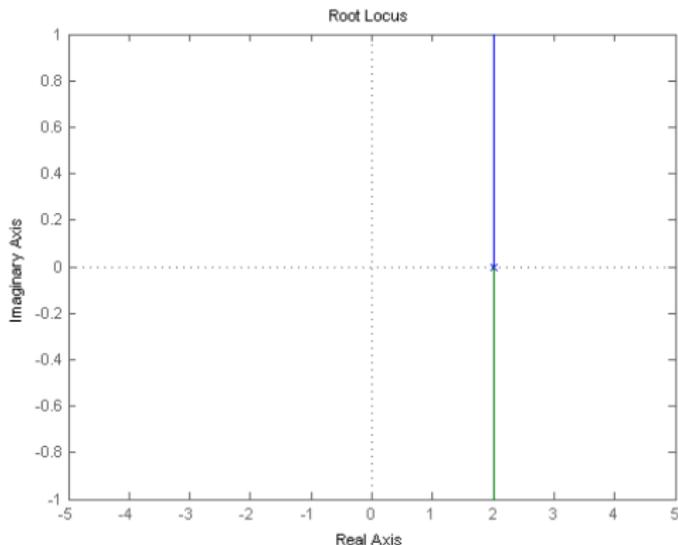
`rlocus(sys)` plots the root locus for the loop transfer function specified by `sys`.

$$L'(s) = \frac{1+s}{s} \cdot \frac{1}{s^3 + 3s^2 + 3s + 1} = \frac{1}{s(s+1)^2}.$$



$$L'(s) = \frac{1}{(s - 2)^2}$$

It is not possible to stabilize the system with a simple proportional controller.

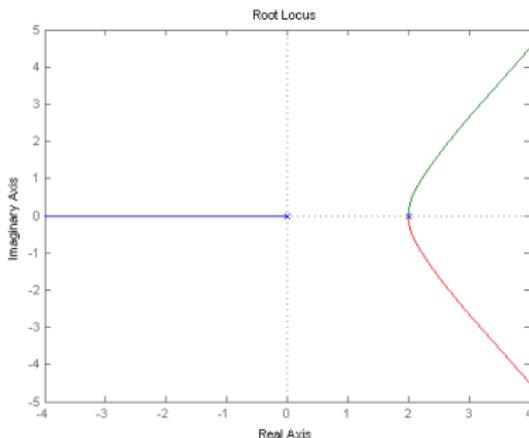


# Root Locus - Example

Add a pole in 0 to have zero steady-state error

$$L'(s) = \frac{1}{s(s-2)^2}$$

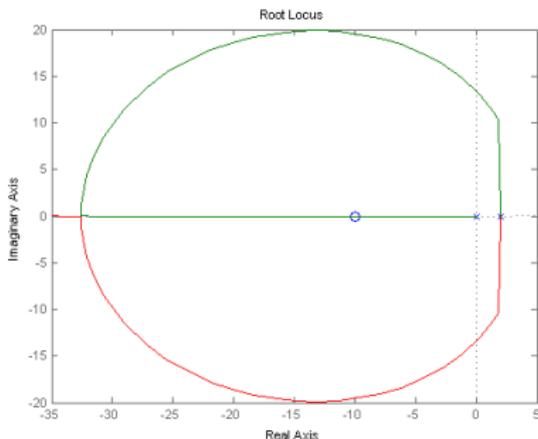
It is still not possible to stabilize the system with a simple proportional controller.



Add two zeros to *draw* the poles in the LHP

$$L'(s) = \frac{(s + 10)^2}{s(s - 2)^2}$$

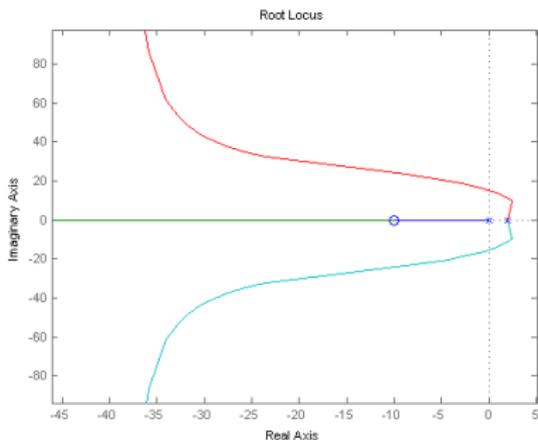
The controller  $K(s) = \rho \frac{(s+10)^2}{s}$  can stabilize the plant but is **not causal**.



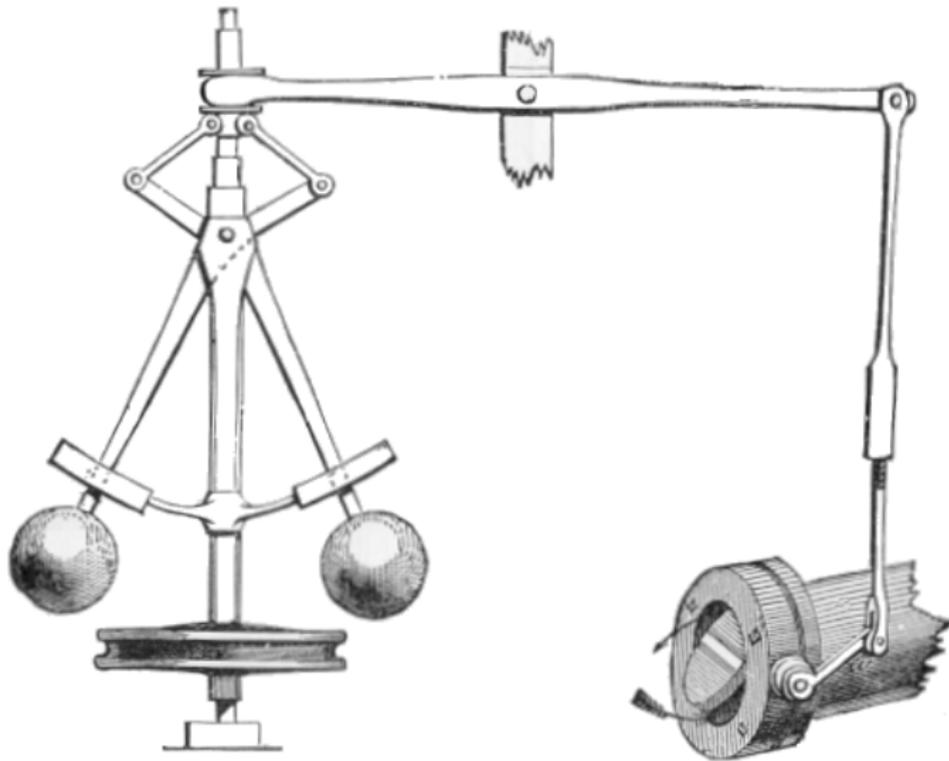
Add an *high frequency* pole to have a proper controller

$$L'(s) = \frac{(s + 10)^2}{s(s + 100)(s - 2)^2}$$

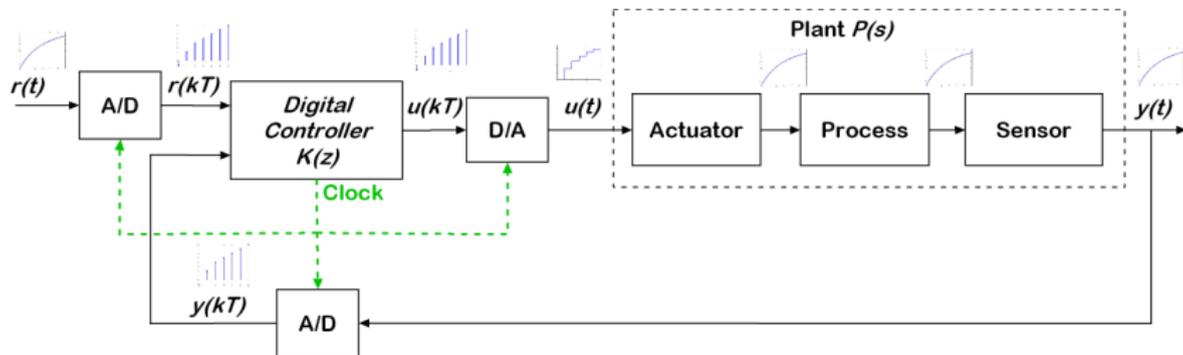
The controller  $K(s) = \rho \frac{(s+10)^2}{s(s+100)}$  can stabilize.



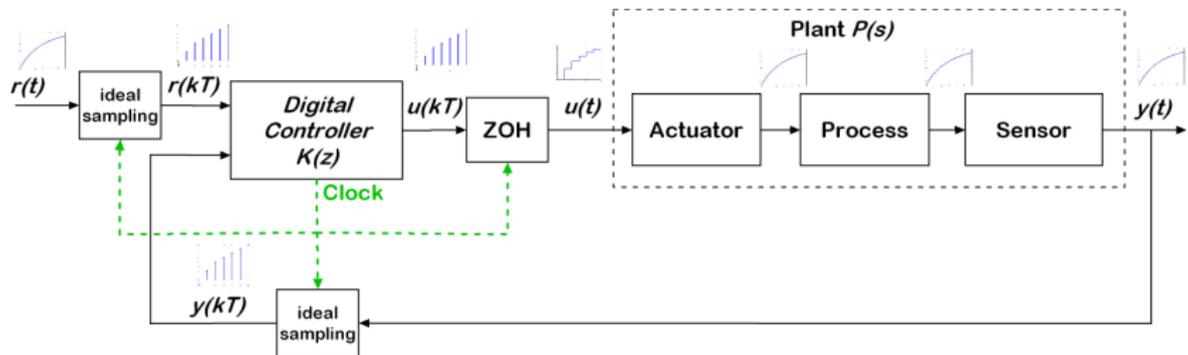
# Watt's governor



# Reference control scheme



# Reference control scheme



A linear time-invariant digital controller can be specified in the discrete-time domain by means of difference equations

$$\begin{aligned}x(k + 1) &= Ax(k) + Be(k) \\ u(k) &= Cx(k) + De(k)\end{aligned}$$

or by the transfer-function in the  $\mathcal{Z}$  domain

$$K(z) = C(zI - A)^{-1}B + D.$$

Given a sampled signal  $\{u(k)\}$  with sampling period  $T = 1/\omega_s$ , one possible way to reconstruct the signal is

$$u(t) = \sum_{k=-\infty}^{+\infty} u(k) \operatorname{sinc} \frac{\pi(t - kT)}{T}. \quad (10)$$

Equation (10) is *noncausal*.

In many communications problems the noncausality can be overcome by adding a phase lag, which adds a **delay** to the reconstructed signal.

**In feedback control systems, delays are disastrous for stability!** Therefore the *simpler* polynomial reconstruction is used. In particular a Zero-Order Hold (ZOH) is usually adopted.

The frequency response of a ZOH is

$$ZOH(j\omega) = e^{-\frac{\omega T}{2}} T \operatorname{sinc}\left(\frac{\omega T}{2}\right).$$

The scaling factor  $T$  is *absorbed* by the sampler.

If the bandwidth of the closed-loop system is smaller than the Nyquist frequency  $\omega_N = \frac{\omega_s}{2}$  (it is usual to work with  $\omega_B < \omega_N/8$ ), then

- the distortion in amplitude can be neglected

The frequency response of a ZOH is

$$ZOH(j\omega) = e^{-\frac{\omega T}{2}} T \operatorname{sinc}\left(\frac{\omega T}{2}\right).$$

The scaling factor  $T$  is *absorbed* by the sampler.

If the bandwidth of the closed-loop system is smaller than the Nyquist frequency  $\omega_N = \frac{\omega_s}{2}$  (it is usual to work with  $\omega_B < \omega_N/8$ ), then

- the distortion in amplitude can be neglected
- BUT a phase lag equal to  $e^{-j\frac{\omega T}{2}}$  MUST BE considered

- Design in the  $\mathcal{L}$  domain (continuous-time) and digitization the controller
- Synthesis in the  $\mathcal{Z}$  domain (discrete-time)

- The sampling frequency has to be chosen in the proper way, so as to avoid aliasing, hidden oscillation, *ringing* of the controller, etc.
- The controller  $K(s)$  is designed in the  $\mathcal{L}$  domain taking explicitly into account the delay due to the phase lag  $e^{-j\frac{\omega T}{2}}$  of the ZOH.
- Digitization of the controller  $K(s)$  so as to attain a discrete-time approximation  $K(z)$

Taking into account that

$$\int_{kT}^{(k+1)T} f(\tau) d\tau \cong Tx(kT), \quad \text{Forward method}$$

$$\int_{kT}^{(k+1)T} f(\tau) d\tau \cong Tx((k+1)T), \quad \text{Backward method}$$

$$\int_{kT}^{(k+1)T} f(\tau) d\tau \cong \frac{T}{2} (x(kT) + x((k+1)T)), \quad \text{Trapezoidal method}$$

A discrete-time approximation  $K(z)$  is obtained by simply replacing the argument  $s$  in  $K(s)$  by  $s'$ , where

$$s' = \frac{z-1}{T}, \quad \text{Forward method}$$

$$s' = \frac{z-1}{zT}, \quad \text{Backward method}$$

$$s' = \frac{2}{T} \frac{z-1}{z+1}, \quad \text{Trapezoidal method}$$

## Matlab commands

`sys_d = c2d(sys_c, T_s, method)` - produces a continuous-time model `sys_c` that is equivalent to the discrete-time LTI model `sys_d`. `method` is a string that selects the conversion method (example `'Tustin'`).

Given the discrete-time plant  $P(z)$ , the controller  $K(z)$  can be designed in the  $\mathcal{Z}$  domain by

- pole-placement techniques based on
  - root locus
  - state space approach
  - polynomial approach (Ragazzini's method, Diophantine equations)
- optimal control approaches (LQG, MPC, . . .)

Given the LTI plant

$$\dot{x}(t) = Ax(t) + Bu(t)$$

$$y(t) = Cx(t) + Du(t)$$

a *sampled equivalent* is computed

$$x(k+1) = A_s x(k) + B_s u(k)$$

$$y(k) = C_s x(k) + D_s u(k)$$

with

$$A_s = e^{AT}, \quad B_s = \int_0^T e^{A\sigma} d\sigma B,$$

and  $C_s = C$ ,  $D_s = D$ .

-  F. M. Callier and C. A. Desoer  
*Linear System Theory*  
Springer-Verlag, 1991
-  G. F. Franklin, J. D. Powell and A. Emami-Naeini  
*Feedback Control of Dynamic Systems*  
Pearson Prentice Hall, 2008
-  S. Skogestad and I. Postlethwaite  
*Multivariable Feedback Control - Analysis and Design*  
John Wiley and Sons, 2006
-  K. Zhou and J. C. Doyle  
*Essentials of Robust Control*  
Prentice Hall, 1998

-  K. J. Åström and B. Wittenmark  
*Computer-Controlled Systems - Theory and Design*  
Prentice Hall, 1997
-  G. F. Franklin, J. D. Powell and M. Workman  
*Digital Control of Dynamic Systems*  
Addison-Wesley, 1998

# Systems & control basics

Gianmaria DE TOMMASI  
Email: detommas@unina.it

IST - Lisboa, Dec, 5th 2017

Now we are ready to start :)