

# Controlli Automatici - laboratorio

Lezione 3 - LQR

Lezione 4 - Sistemi di primo e secondo ordine

Sistemi del primo ordine

Sistemi del secondo ordine

Lezione 5 - Sistemi del secondo ordine a poli reali

Sistemi del secondo ordine: dalla fdt allo spazio degli stati

Sistemi del secondo ordine con poli reali

Sistemi del primo ordine con uno zero

Sistemi a fase non minima (sottoelongazione)

Sistemi a fase minima con sovraelongazione

Sistemi a fase minima (code di assestamento)

Interconnessione di sistemi

Lezione 6 - esempio di un progetto di controlli automatici

Primo punto

Lezione 7 - diagrammi di Bode

Scala logaritmica

Guadagno statico - sistemi del primo ordine

Zeri nell'origine - sistemi del primo ordine

Poli nell'origine - sistemi del primo ordine

Zero reale - sistemi del primo ordine

Polo reale - sistemi del primo ordine

Zeri complessi coniugati - sistemi del secondo ordine

Poli complessi coniugati - sistemi del secondo ordine

Ritardo temporale

Risonanza

Lezione 8 - diagrammi di Bode in MATLAB

Sistemi del primo ordine - ingresso sinusoidale

Diagramma di Bode e risposta ad ingresso sinusoidale per un sistema complesso

Sistemi del secondo ordine - ingresso sinusoidale

Esempio di risonanza (carrello)

Esempio - onda quadra con disturbo in alta frequenza

Lezione 9 - sistemi in retroazione ad anello chiuso

Casi patologici

Funzioni di sensitività

Graficare schemi a blocchi in retroazione su MATLAB

Risposta a sistemi in retroazione in MATLAB con funzioni di sensitività

Lezione 10 - Sistemi di controllo: progetto del regolatore

Schema di controllo in retroazione: specifiche

Riepilogo specifiche

Loop shaping

Sintesi del regolatore

Lezione 11: specifiche in MATLAB

Lezione 13 - sintesi del regolatore

Regolatore statico

Regolatore dinamico

→ Appunti di teoria

[https://s3-us-west-2.amazonaws.com/secure.notion-static.com/d07a2f0c-c44a-4300-805d-44b35dbe0f3e/Comandi\\_Matlab.pdf](https://s3-us-west-2.amazonaws.com/secure.notion-static.com/d07a2f0c-c44a-4300-805d-44b35dbe0f3e/Comandi_Matlab.pdf)

## Lezione 3 - LQR

@October 27, 2022

Linear quadratic regulation problem - LQR: problema di ottimizzazione

minimizzare  $x(\cdot), u(\cdot)$ :

$$\int_0^{+\infty} x(\tau)^T Q x(\tau) + u(\tau)^T R u(\tau) dz \quad \text{soggetto a } \dot{x}(t) = Ax(t) + Bu(t)$$

$$x \in \mathbb{R}^n \quad Q \in \mathbb{R}^{n \times n}$$

$$u \in \mathbb{R}^m \quad R \in \mathbb{R}^{m \times m} \quad \text{definite positive}$$

$$x^T I x = \|x\|^2$$

$$u(t) = kx(t)$$

$$P \in \mathbb{R}^{n \times m} : A^T P + P A - P B R^{-1} B^T P + Q = 0$$

$K = R^{-1} B^T P$  → equazione algebrica di Riccati

```
k = lqr(A,B,Q,R)
```

```
%  
% Esempio Carrello: principio di sovrapposizione degli effetti  
%  
close all; clear all; clc;
```

```
% parametri fisici del sistema  
mm = 0.5; % kg  
kk = 1; % costante elastica [N/m]
```

```
% condizione iniziale del carrello  
pos_init = 5; % [m]  
vel_init = 1; % [m/s]  
x0 = [pos_init; vel_init];
```

```
% intervallo di tempo  
interv = 0:0.1:15; % da 0 a 15 secondi con passo 0.1
```

```
%% creazione oggetto sistema e soluzione equazione differenziale
```

```
% matrici del sistema
```

```
A = [0 1; -kk/mm 0];
```

```
B = [0; 1/mm];
```

```
C = [1 0];
```

```
D = 0;
```

```
Q = [10 0 ; 0 7];
```

```
Q = [5 0; 2 3];
```

```
R = [10];
```

```
K = lqr(A, B, Q, R);
```

```
x1_e = 0;
```

```
u_e = kk*x1_e;
```

```
% input applicato: costante pari ad 1  
inp = @(t) u_e*ones(size(t));
```

```
% state-space model
```

```
modello = ss(A-B*K,[], C, D);
```

```
% evoluzione sistema
```

```

uu = inp(interv);
[YY, TT, XX] = lsim(modello, uu, interv, x0);

%% plot con animazione

% parametri del disegno
r=0.25; %raggio
Wrad = 0:.02:2*pi;
Wx = r*cos(Wrad);
Wy = r*sin(Wrad);

% inizializza figura
figure(1);
clf;
min_T=0;
max_T=max(TT);
min_Y=min(YY);
max_Y=max(YY);
subplot(2,1,1)
axis([min_T max_T (min_Y-0.2) (max_Y+0.2)])
subplot(2,1,2)
axis([-10 10 0 5])

for i=1:5:length(TT)
    figure(1);
    clf;

    % aggiorna figura superiore
    subplot(2,1,1)
    plot(TT(1:i), YY(1:i),'b', TT(1:i), uu(1:i), 'r');
    axis([min_T max_T (min_Y-0.2) (max_Y+0.2)])
    title(['Evoluzione del sistema al tempo: ',num2str(TT(i)), ' [s]']);
    legend('Uscita y(t)', 'Ingresso u(t)')
    grid on; box on;

    % aggiorna figura inferiore
    subplot(2,1,2)
    axis([-10 10 0 5])
    patch(YY(i)+[-2 2 2 -2] ,2+[1.5 1.5 -1.5 -1.5],'y') % cart
    hold on;
    patch(Wx+YY(i)-1.8, Wy+.25,'r'); % wheel
    patch(Wx+YY(i)+1.8, Wy+.25,'r'); % wheel
    plot([-10 -10 20],[5 0 0],'k','LineWidth',4) % ground and wall.
    plot([-10, -9, -9:(+9 +YY(i)-4)/9]:YY(i)-4, YY(i)-4, YY(i)-2],...
        2+1+[0 0 0 .5 -.5 .5 -.5 .5 -.5 .5 -.5 0 0 0 ],'r','LineWidth',2) % spring
    title('Risposta del sistema')

    % pausa prima del prossimo frame
    pause(0.0001);
end

```

#### Funzione di trasferimento:

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

$$y(t) = Cx(t)$$

$$G(s) = C(sI - A)^{-1}B$$

$$A = -\frac{1}{M} \quad B = \frac{\mu}{M} \quad C = 1$$

```

%% importante: definizione della variabile di Laplace s in MATLAB
s = tf('s')

G = 3/(s+2);
G = C*inv(s*eye(1)-A)*B;

```

$$G(s) = \frac{k}{s + \alpha}$$

$$G(s) = \frac{\mu}{\frac{s}{\rho} + 1}$$

## Lezione 4 - Sistemi di primo e secondo ordine

@November 3, 2022

### Sistemi del primo ordine

sempre sulla funzione di trasferimento:

$$\lim_{t \rightarrow \infty} f(t) = \lim_{s \rightarrow 0} sF(s)$$

$$G(s) = \frac{\mu}{1+Ts} \quad U(s) = \frac{k}{s} \quad \mu > 0, k > 0, T > 0$$

$$Y(s) = G(s)U(s) = \frac{\mu k}{(1+Ts)s}$$

$$y(t) = \mu k (1 - e^{-\frac{t}{T}}) u(t)$$

Tempo di assennamento:

$$T_{a,\epsilon} : (1 - 0,01\epsilon)y_\infty \leq y(t) \leq (1 + 0,01\epsilon)y_\infty \quad \forall t \geq T_{a,\epsilon}$$

$$T_{a,\epsilon} = Tlu(\frac{1}{0,01\epsilon})$$

Esempio:

$$k = 1, \mu = 1$$

$$Y_\infty = \mu k = 1$$

$$\epsilon = 5\%$$

$$T_{a,\epsilon} = Tlu(\frac{1}{0,01\epsilon})$$

$$T_{a,5} \approx sT$$

$$T_{a,1} \approx 4,6T$$

$$G(s) = \frac{1}{1+sT} \quad T = 1;$$

in MATLAB:

```
s = tf('s')
G(s) = mu/(1+T*s)

step(G); oppure Y = step(G); oppure Y = step(G,T); T = 0:0.01:5;
```

codice completo:

```
clc; clear all; close all;
```

```
mu = 1;
epsilon = 5;
s = tf('s');

figure;
hold on; box on; zoom on; grid on;
TT = 0:0.1:15;
for T = 0.1:0.5:3.1
```

```
%definizione del sistema
G = mu/(1+T*s);
```

```
%elenco dei poli
% pole(G);
%diagramma poli/zeri
% pzmap(G);
```

```

%risposta al gradino
YY = step(G,TT);

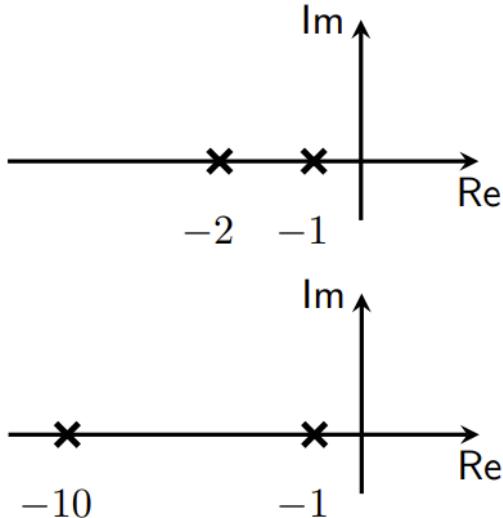
%tempo di assestamento
sinfo = stepinfo(G,'SettlingTimeThreshold',epsilon/100);
settling_time = sinfo.SettlingTime;
fprintf('T = %.1f, Ta,5 = %.1f secondi\n', T, settling_time);

%plot
plot(TT, YY, 'DisplayName', ['T = ' num2str(T)], 'LineWidth', 1.3);
ylim([-0.1 1.1]);

end
legend;

```

Ricapitolando:



$T$  = costante di tempo associata al polo;

$\mu$  = guadagno;

$$e_\infty = |1 - \mu|k$$

errore a regime;

$$T_{a,\epsilon} = T \ln\left(\frac{1}{0,01\epsilon}\right)$$

tempo di assestamento;

$$Y_\infty = \lim_{t \rightarrow \infty} y(t) = \mu k$$

parte asintotica;

## Sistemi del secondo ordine

Esempio: variante del carrello

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

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

$$G(s) = C(sI - A)^{-1}B = \frac{\frac{1}{M}}{s^2 + \frac{b}{M}s + \frac{k}{M}}$$

$$\mu = \frac{1}{k} \quad \omega_n = \sqrt{\frac{k}{M}} \quad \xi = \frac{b}{2\sqrt{kM}}$$

$$G(s) = \mu \frac{\omega_n^2}{s^2 + 2\xi\omega_n + \omega_n^2}$$

```

%% Sistema del secondo ordine

close all; clear all; clc;

% parametri del sistema
k = 1.0;
b = 5.0;
M = 2;

mu = 1/k;
xi = b/(2*sqrt(k*M));
omegan = sqrt(k/M);

s = tf('s');

%% definizione del sistema
fprintf("Funzione di trasferimento:\n");
if 0
    % fdt
    G = mu * omegan^2/(s^2 + 2*xi*omegan*s + omegan^2)
else
    % spazio degli stati
    A = [0 1; -k/M, -b/M];
    B = [0; 1/M];
    C = [1 0];
    D = 0;
    G = ss(A,B,C,D);
    tf(G);
end

%% elenco dei poli
poles = pole(G);
fprintf("Poli del sistema:\n");
disp(poles);

% diagramma poli/zeri
figure;
pzmap(G);

figure;
step(G); grid on;

```

$$G(s) = \mu \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad U(s) = \frac{k}{s}$$

$$|\xi| < 1 \quad \text{coefficiente di smorzamento;} \\ \omega_n > 0 \quad \text{pulsazione naturale;} \\$$

$$y(t) = \mu k(1 - A e^{-\xi\omega_n t} \sin(\omega t + \varphi)) \mathbf{1}(t)$$

$$A = \frac{1}{\sqrt{1-\xi^2}} \quad \omega = \omega_n \sqrt{1-\xi^2} \quad \varphi = a \cos(\xi) \\ Y_\infty = \mu k$$

$$T_{a,5} \approx \frac{3}{\xi\omega_n} \\ T_{a,1} \approx \frac{4.6}{\xi\omega_n}$$

### Sovraelongazione percentuale:

$$s\% = \frac{Y_{max} - Y_\infty}{Y_\infty}$$

$$s\% = 100 \frac{Y_{max} - Y_\infty}{Y_\infty}$$

questa  $s\%$  dipende solo ed unicamente da  $\xi$  ed è una funzione monotona decrescente.

Nel codice matlab, continuando quello precedente:

```
% proprietà della risposta
epsilon = 5; % percentuale per trovare il tempo di assestamento
sinfo = stepinfo(G,'SettlingTimeThreshold',epsilon/100);
settling_time = sinfo.SettlingTime;
percentage_overshoot = 100 * (sinfo.Peak - mu)/mu;
peak_time = sinfo.PeakTime;
fprintf('Il tempo di assestamento ad epsilon = %i%% è di %.2f secondi\n', epsilon, settling_time);
fprintf('La massima sovraelongazione percentuale è di %.2f%%\n', percentage_overshoot);
fprintf('Il tempo di massima sovraelongazione è di %.2f secondi\n', peak_time);
```

## Lezione 5 - Sistemi del secondo ordine a poli reali

@November 7, 2022

### Sistemi del secondo ordine: dalla fdt allo spazio degli stati

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = -\omega_n^2 x_1 - 2\xi\omega_n x_2 + \mu\omega_n^2 u$$

$$y = x_1$$

$$G(s) = C(sI - A)^{-1}B = \mu \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2}$$

1.  $\xi$  coefficiente di smorzamento;
2.  $\omega_n$  pulsazione naturale;
3.  $\mu$  guadagno;

### Sistemi del secondo ordine con poli reali

- Caso  $T_1 \neq T_2, T_1 > T_2$

$$G(s) = \frac{\mu}{(1+T_1s)(1+T_2s)} \quad \mu > 0, k > 0, T_1 > 0, T_2 > 0$$

$$U(s) = \frac{k}{s} \quad k > 0$$

$$e^{-\frac{t}{T_1}} 1(t) \quad e^{-\frac{t}{T_2}} \quad \text{modi del sistema}; \quad 1(t) \quad \text{modo di ingresso};$$

$$Y(s) = G(s)U(s) = \frac{\mu k}{T_1 T_2} \frac{1}{s(s + \frac{1}{T_1})(s + \frac{1}{T_2})}$$

$$Y(s) = \frac{k_1}{s} + \frac{k_2}{s + 1/T_1} + \frac{k_2}{s + 1/T_2}$$

$$\begin{aligned} k_1 &= \mu k & k_2 &= -\mu k \frac{T_1}{T_1 - T_2} & k_3 &= \mu k \frac{T_2}{T_1 - T_2} \\ y(t) &= \mu k \left(1 - \frac{T_1}{T_1 - T_2} e^{-\frac{t}{T_1}} + \frac{T_2}{T_1 - T_2} e^{-\frac{t}{T_2}}\right) 1(t) \\ y(0) &= 0 & \dot{y}(0) &= 0 & \ddot{y}(0) &= \frac{\mu k}{T_1 T_2} & y_\infty &= \mu k \end{aligned}$$

- Caso  $T_1 \gg T_2$  sistemi a polo (reale) dominante

Uno dei due modi ( $e^{-\frac{t}{T_2}}$ ) si spegne molto velocemente.

$$y(t) \approx \mu k \left(1 - e^{-\frac{t}{T_1}}\right) 1(t) \quad \frac{T_2}{T_1 - T_2} \ll \frac{T_1}{T_1 - T_2} \approx 1$$



- Caso  $T_1 = T_2$  poli reali coincidenti

$$G(s) = \frac{\mu}{(1 + T_1 s)^2} = \frac{\mu}{T_1^2} \frac{1}{(s + 1/T_1)^2}$$

$$Y(s) = \frac{\mu k}{T_1^2 s (s + 1/T_1)^2} = \frac{k_{11}}{s} + \frac{k_{21}}{s + 1/T_1} + \frac{k_{22}}{(s + 1/T_1)^2}$$

$$k_{11} = s Y(s)|_{s=0} = \frac{\mu k}{T_1^2} \cdot \frac{1}{(s + 1/T_1)^2} \Big|_{s=0} = \mu k$$

$$k_{22} = (s + \frac{1}{T_1})^2 Y(s) \Big|_{s=-\frac{1}{T_1}} = \frac{\mu k}{T_1^2} \cdot \frac{1}{s} \Big|_{s=-\frac{1}{T_1}} = -\frac{\mu k}{T_1}$$

$$k_{21} = \left[ \frac{ds}{ds} \left( s + \frac{1}{T_1} \right)^2 Y(s) \right]_{s=-\frac{1}{T_1}} = \frac{d}{ds} \left[ \frac{\mu k}{T_1^2} \cdot \frac{1}{s} \right]_{s=-\frac{1}{T_1}} = -\frac{\mu k}{T_1^2} \cdot \frac{1}{s^2} \Big|_{s=-\frac{1}{T_1}} = -\mu k$$

$$y(t) = \mu k \left(1 - e^{-\frac{t}{T_1}} - \frac{t}{T_1} e^{-\frac{t}{T_1}}\right) 1(t)$$

### Sistemi del primo ordine con uno zero

$$G(s) = \mu \frac{1 + \alpha T_s}{1 + T_s} \quad U(s) = \frac{k}{s} \quad \mu, k, T > 0$$

$$y(t) = \mu k(1 + (\alpha - 1)e^{-\frac{t}{T}})1(t)$$

$y(0) = \mu \alpha k \rightarrow$  particolarità dei sistemi con zero  $\rightarrow$  risposta in 0 diversa da 0

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

nei sistemi di primo ordine senza zero di solito  $Du = 0 \rightarrow y = Cx \rightarrow x(0) = 0 \rightarrow y(0) = 0$

```
clear all; close all; clc;

mu = 1;
T = 2;
alfa = -3;
k = 1;

s = tf('s');

G_1 = mu/(1+T*s)
modello_1 = ss(G_1);
fprintf("A=%1f, B=%1f, C=%1f, D=%1f\n",modello_1.A,modello_1.B,modello_1.C,modello_1.D)

pause
G_2 = mu*(1+alfa*T*s)/(1+T*s)
modello_2 = ss(G_2);
fprintf("A=%1f, B=%1f, C=%1f, D=%1f\n",modello_2.A,modello_2.B,modello_2.C,modello_2.D)

pause
step(G_2);
```

### Sistemi a fase non minima (sottoelongazione)

$$y(t) = \mu k \left( 1 - \frac{T_1 - \tau}{T_1 - T_2} e^{-\frac{t}{T_1}} + \frac{T_2 - \tau}{T_1 - T_2} e^{-\frac{t}{T_2}} \right) 1(t)$$

$$y(0) = 0 \quad y_\infty = \mu k \quad \dot{y}(0) = \frac{\mu k(\tau)}{T_1 T_2} \quad T_1 > T_2 > 0, \quad \tau > 0$$

```
clear all; close all; clc;

mu = 1;
T1 = 1;
T2 = 3;
s = tf('s');

TT = 0:0.01:15; % intervallo temporale

figure;
hold on; box on; zoom on; grid on;

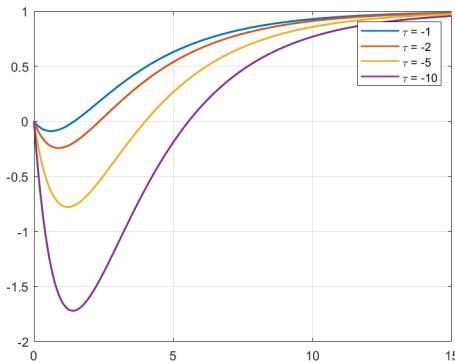
% ciclo su valori di tau
for tau=[-1, -2, -5, -10]

    % definizione del sistema
    G = mu*(1+tau*s)/((1+T1*s)*(1+T2*s));
    % risposta al gradino
    YY = step(G, TT);

    % plot
    plot(TT, YY,'DisplayName', ['\tau = ' num2str(tau)], 'LineWidth', 1.3);

end
```

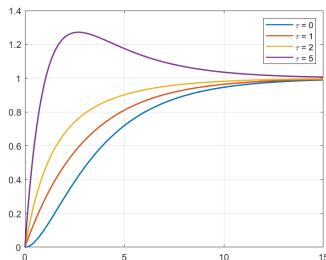
```
end
legend;
```



### Sistemi a fase minima con sovraelongazione

$$\tau > T_1 > T_2 > 0$$

$$\dot{y}(0) = \frac{\mu k \tau}{T_1 T_2} \quad \text{più } \tau \text{ è grande più abbiamo sovraelongazione;}$$



```
clear all; close all; clc;

mu = 1;
T1 = 1;
T2 = 3;
s = tf('s');

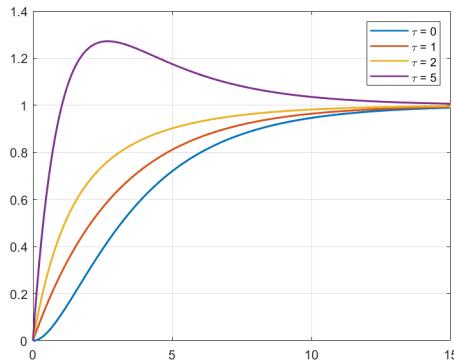
TT = 0:0.01:15; % intervallo temporale

figure;
hold on; box on; zoom on; grid on;

% ciclo su valori di tau
for tau=[0, 1, 2, 5]

    % definizione del sistema
    G = mu*(1+tau*s)/((1+T1*s)*(1+T2*s));
    % risposta al gradino
    YY = step(G, TT);

    % plot
    plot(TT, YY,'DisplayName', ['\tau = ' num2str(tau)], 'LineWidth', 1.3);
end
legend;
```

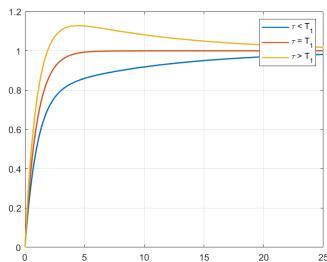


### Sistemi a fase minima (code di assestamento)

$$\tau \approx T_1 \gg T_2 \quad \tau, T_1, T_2 > 0$$

$$y(t) = \mu k \left( 1 - \frac{T_1 - \tau}{T_1 - T_2} e^{-\frac{t}{T_1}} + \frac{T_2 - \tau}{T_1 - T_2} e^{-\frac{t}{T_2}} \right) 1(t)$$

il sistema supera il valore asintotico e molto lentamente ci ritorna; questo effetto si chiama coda di assestamento.



```

clear all; close all; clc;

mu = 1;
T1 = 10;
T2 = 1;
s = tf('s');

TT = 0:0.01:25; % intervallo temporale

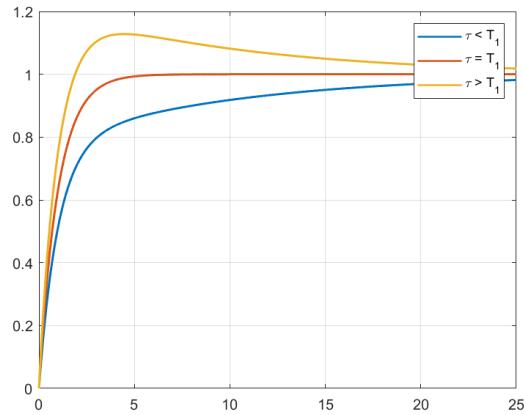
figure;
hold on; box on; zoom on; grid on;

% ciclo su valori di tau
for tau=[T1-2, T1, T1+2]

    % definizione del sistema
    G = mu*(1+tau*s)/((1+T1*s)*(1+T2*s));
    % risposta al gradino
    YY = step(G, TT);

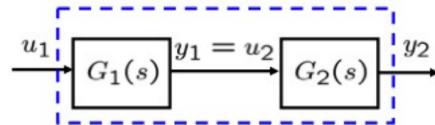
    % plot
    plot(TT, YY, 'LineWidth', 1.3);
end
legend('\tau < T_1', '\tau = T_1', '\tau > T_1');

```



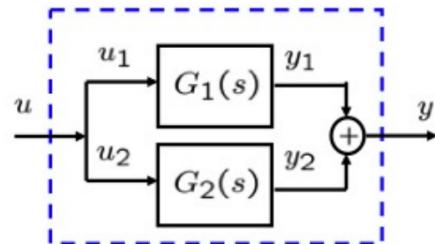
## Interconnessione di sistemi

- **Serie**



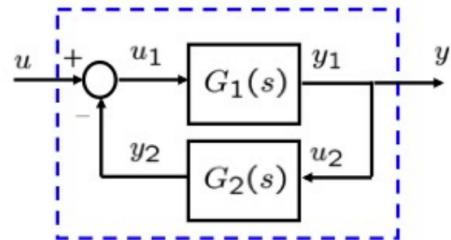
$$G_{serie}(s) = \frac{Y_2(s)}{U_1(s)} = G_1(s)G_2(s)$$

- **Parallelo**



$$G_{par}(s) = G_1(s) + G_2(s)$$

- **Retroazione**



$$G_{ret}(s) = \frac{G_1(s)}{1 + G_1(s)G_2(s)}$$

## Lezione 6 - esempio di un progetto di controlli automatici

@November 14, 2022

Esempio di progetto di tipologia b traccia 1:

[https://s3-us-west-2.amazonaws.com/secure.notion-static.com/779b305e-093e-43f8-b700-4b8666748d22/Progetto\\_b1.pdf](https://s3-us-west-2.amazonaws.com/secure.notion-static.com/779b305e-093e-43f8-b700-4b8666748d22/Progetto_b1.pdf)

Si consideri un satellite in orbita intorno alla Terra. Si supponga che la sua dinamica radiale ed angolare sia descritta dalle seguenti equazioni differenziali:

$$m\ddot{p} = mp\omega^2 - \frac{mK_G M}{p^2} - \beta_1 \dot{p} + F \quad \text{con} \quad F = mk\left(\frac{K_G M}{p^2} - p\omega^2\right) \quad \text{primo controllore già progettato}$$

$$\dot{\omega} = -\frac{2\omega\dot{p}}{p} - \frac{\beta_2\omega}{m} + \frac{\tau}{mp}$$

$p(t)$  = distanza satellite-terra

$\omega(t)$  velocità angolare → si suppone di poterla misurare

$F(t)$  forza radiale → ingresso del sistema

$\tau(t)$  forza tangenziale → ingresso del sistema

$\beta_1, \beta_2$  coefficienti di attrito

### Primo punto

1. riportare il sistema nella forma di stato

$$\begin{aligned} \dot{x} &= f(x, u) & x \in \mathbb{R}^n & u \in \mathbb{R}^m \\ y &= h(x, u) & y \in \mathbb{R}^m \end{aligned}$$

le variabili di stato →  $x = \begin{bmatrix} p \\ \dot{p} \\ \omega \end{bmatrix}$  rispettivamente  $x_1 = p$ ,  $x_2 = \dot{p}$ ,  $x_3 = \omega$

$$\dot{x}_1 = \dot{p}$$

$$\dot{x}_2 = -\beta_1 p + m(k-1)\left(\frac{K_G M}{p^2} - p\omega^2\right)$$

$$\dot{x}_3 = -\frac{2\omega\dot{p}}{p} - \frac{\beta_2\omega}{m} + \frac{\tau}{mp}$$

$$f(x, u) = \begin{bmatrix} x_2 \\ -\beta_1 x_2 + m(k-1)\left(\frac{K_G M}{x_1^2} - x_1 x_3^2\right) \\ -\frac{2x_3 x_2}{x_1} - \frac{\beta_2 x_3}{m} + \frac{u}{mx_1} \end{bmatrix} \quad (x_2 = \dot{p})$$

2. trovare la coppia di equazioni  $(x_e, \mu_e)$  a partire da  $p_e$

$$0 = f(x_e, u_e)$$

$$y_e = h(x_e, u_e) \quad x_e = \begin{bmatrix} x_{1e} \\ x_{2e} \\ x_{3e} \end{bmatrix}$$

$$\boxed{x_{1e} = p_e} \quad \text{da testo}$$

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} x_{2e} \\ -\beta_1 x_{2e} + m(k-1)\left(\frac{K_G M}{p_{1e}} - p_e x_{3e}^2\right) \\ -\frac{2x_{3e}x_{2e}}{p_e} - \frac{\beta_2 x_3}{m} + \frac{u_e}{mp_e} \end{bmatrix}$$

$$x_{2e} = 0$$

$$0 = (k-1)\left(\frac{K_G M}{p_e} - p_e x_3^2\right)$$

$$0 = -\frac{\beta_2 x_3}{m} + \frac{u_e}{mp_e}$$

$$x_{3e} = \sqrt{\frac{K_G M}{p_e^3}} \quad u_e = \beta_2 p_e \sqrt{\frac{K_G M}{p_e^3}}$$

$$x_e \begin{bmatrix} p_e \\ 0 \\ \sqrt{\frac{K_G M}{p_e^3}} \end{bmatrix} u_e$$

### 3. linearizzare il sistema nell'equilibrio (non finito)

$$\begin{aligned} \delta \dot{x} &= A \delta x + \\ B \delta u \\ \delta y &= C \delta x + D \delta u \end{aligned}$$

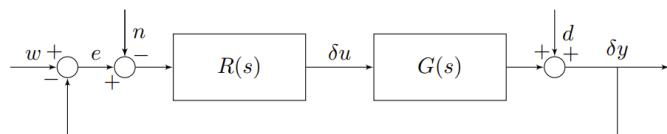


Figura 2: Schema di controllo.

Cambio di variabile:

$$\delta x = x - x_e$$

$$\delta u = u - u_e$$

$$\delta y = y - y_e$$

Calcoliamo le matrici: (non finito)

$$A = \frac{\delta f(x, u)}{\delta x} \Big|_{x=x_e, u=u_e} = \begin{bmatrix} \frac{\delta f_1(x, u)}{\delta x_1} & \dots & \frac{\delta f_1(x, u)}{\delta x_u} \\ \dots & \dots & \dots \\ \frac{\delta f_u(x, u)}{\delta x_1} & \dots & \frac{\delta f_u(x, u)}{\delta x_u} \end{bmatrix}_{x=x_e, u=u_e}$$

$$= \begin{bmatrix} 0 & 1 & 0 \\ -x_3^2(k-1) \frac{-3(k-1)k_G M}{x_1^3} & \frac{-\beta_1}{m} & -2(k-1) - 2x_1 x_3 \\ \frac{-u}{mx_1^2} + \frac{2x_3 x_2}{x_1^2} & \frac{-2x_3}{x_1} & \frac{-2x_2}{x_1} - \frac{\beta_2}{m} \end{bmatrix}_{x=x_e, u=u_e}$$

$$= \begin{bmatrix} 0 & 1 & 0 \\ -x_{3e}^2(k-1) \frac{-3(k-1)k_G M}{x_{1e}^3} & \frac{-\beta_1}{m} & -2(k-1) - 2x_{1e} x_{3e} \\ \frac{-u_e}{mx_{1e}^2} + \frac{2x_{3e} x_{2e}}{x_1^2} & \frac{-2x_{3e}}{x_{1e}} & \frac{-2x_{2e}}{x_{1e}} - \frac{\beta_2}{m} \end{bmatrix}_{x=x_e, u=u_e}$$

## Lezione 7 - diagrammi di Bode

@November 15, 2022 → November 17, 2022

Per un certo valore di  $\omega$ ,  $G(\omega)$  ovvero la risposta in frequenza è un numero complesso, ed uno dei modi più usati per rappresentare la risposta in frequenza sono i diagrammi di Bode in cui si rappresentano separatamente  $|G(j\omega)|$  e  $\arg(G(j\omega))$  in funzione di  $\omega$ .

La funzione di trasferimento in forma fattorizzata si esprime così:

$$G(s) = \mu \frac{\Pi_i (1 + \tau_i s) \Pi_i (1 + 2 \frac{\zeta_i}{\alpha_{n,i}} s + \frac{s^2}{\alpha_{n,i}^2})}{s^g \Pi_i (1 + T_i s) \Pi_i (1 + 2j\xi_i \frac{\omega}{\omega_{n,i}} - \frac{\omega^2}{\omega_{n,i}^2})}$$

E quindi la risposta in frequenza associata:

$$G(j\omega) = \mu \frac{\Pi_i (1 + j\omega \tau_i) \Pi_i (1 + 2j\xi_i \frac{\omega}{\omega_{n,i}} - \frac{\omega^2}{\alpha_{n,i}^2})}{(j\omega)^g \Pi_i (1 + j\omega T_i) \Pi_i (1 + 2j\xi_i \frac{\omega}{\omega_{n,i}} - \frac{\omega^2}{\omega_{n,i}^2})}$$

Assegnando i valori, ad esempio si otterrà qualcosa del genere:

$$G(s) = \frac{10(1 + 0.1s)}{s(1 + 10s)(1 + 2 \frac{0.1s}{10^2} + \frac{s^2}{10^2})}$$

## Scala logaritmica

Per i diagrammi di Bode andreamo a considerare gli assi su una scala logaritmica.

**Aampiezza:**  $|G(j\omega)|_{dB} = 20 \log |G(j\omega)| = 20 \log |\mu| - 20g \log |\jmath\omega|$

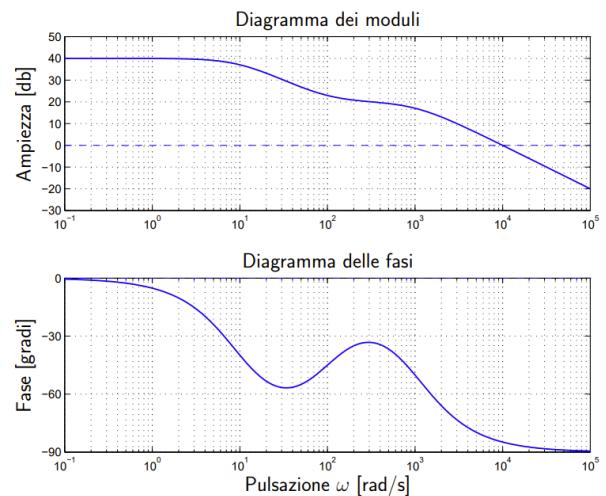
L'ampiezza o modulo è espresso sulle ordinate in decibel e rappresenta la parte reale del numero complesso.

**Fase:**  $\arg(G(j\omega)) = \arg \mu - g \arg(j\omega)$

La fase è espressa sulle ordinate in gradi o radianti e rappresenta la parte immaginaria del numero complesso.

In entrambi i diagrammi l'ascisse non misura semplicemente la pulsazione  $\omega$ , bensì il suo logaritmo. Quindi le frequenze sull'asse delle ascisse non sono in scala lineare e non sono equidistanti.

Si dice **decade** il rapporto tra due frequenze  $\omega_2/\omega_1$  pari a 10.



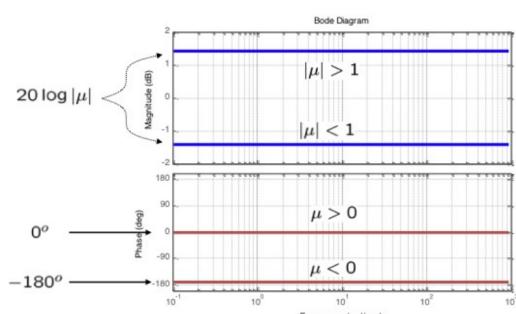
Nota:  $20 \log_0 \zeta$ , se  $\zeta = 0$ , in dB equivale a  $-\infty$ .

### Guadagno statico - sistemi del primo ordine

$$G_a(j\omega) = \mu \quad |G_a(j\omega)|_{dB} = 20 \log |\mu| \\ \arg(G_a(j\omega)) = \arg(\mu)$$

**Aampiezza:** se  $|\mu| \geq 1$  allora  $20 \log |\mu| \geq 0$ , se  $|\mu| < 1$  allora  $20 \log |\mu| < 0$ .

**Fase:** se  $\mu > 0$  allora  $\arg(\mu) = 0$ , se  $\mu < 0$  allora  $\arg(\mu) = 180^\circ$ .



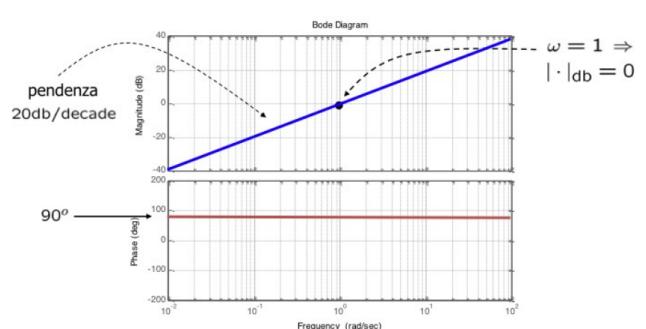
### Zeri nell'origine - sistemi del primo ordine

Consideriamo uno zero nell'origine (ovvero  $g = -1$ )

$$G_b(j\omega) = \frac{1}{(j\omega)^g} = j\omega \quad |G_b(j\omega)|_{dB} = 20 \log \omega \\ \arg(G_b(j\omega)) = \arg(j\omega)$$

**Aampiezza:** scala logaritmica quindi  $\log \omega \mapsto 20 \log \omega$  è una retta con pendenza 20dB/dec (20 dB su decade).

Esempio:



$$G(j\omega) = (j\omega)^4$$

$$|G(j\omega)|_{dB} = 4 \cdot 20 \log(|j\omega|)$$

Se ho  $g$  zeri nell'origine allora pendenza  
 $= 20g$  dB/dec.

**Fase:**  $j\omega$  punto sul semiasse immaginario positivo  $\forall \omega > 0$ , quindi fase  $90^\circ \forall \omega > 0$ .

### Poli nell'origine - sistemi del primo ordine

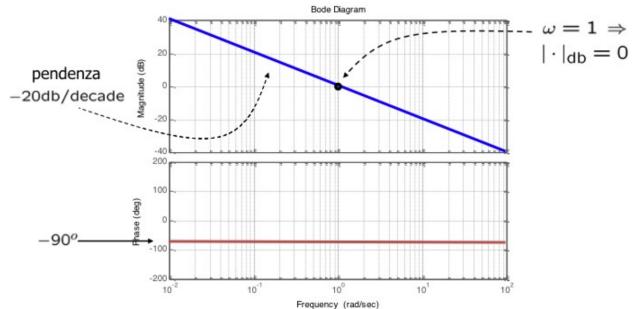
Consideriamo un polo nell'origine, ovvero

$$g = 1$$

$$G_b(j\omega) = \frac{1}{(j\omega)^g} = \frac{1}{j\omega}$$

$$|G_b(j\omega)|_{dB} = -20 \log \omega$$

$$\arg(G_b(j\omega)) = -\arg(j\omega)$$



**Ampiezza:** scala logaritmica quindi  
 $\log \omega \mapsto -20 \log \omega$  è una retta con pendenza  
 $-20$  dB/dec. (negativo!!)

Se ho  $g$  poli nell'origine allora pendenza  
 $-20g$  dB/dec.

**Fase:**  $-j\omega$  punto sul semiasse immaginario  
negativo  $\forall \omega > 0$ , quindi fase  $-90^\circ \forall \omega > 0$ .

### Zero reale - sistemi del primo ordine

Consideriamo  $G_c(j\omega) = 1 + j\omega\tau$  (zero reale)

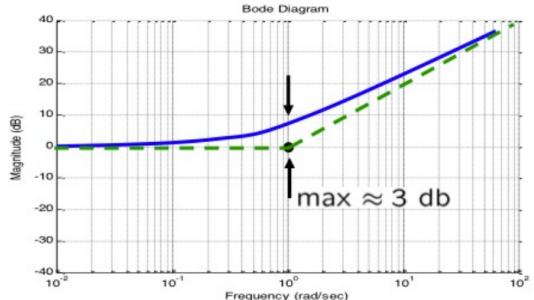
**Ampiezza:**

$$|G_c(j\omega)|_{dB} = 20 \log \sqrt{1 + \omega^2\tau^2} =$$

$$\begin{cases} 20 \log 1 = 0 & \omega \ll \frac{1}{|\tau|} \\ 20 \log \omega |\tau| = -20 \log \frac{1}{|\tau|} + 20 \log \omega & \omega \gg \frac{1}{|\tau|} \end{cases}$$

Nota (diagramma asintotico) → fino a  $\omega = \frac{1}{|\tau|}$  pulsazione di taglio, andamento

costante a 0 dB. A partire dalla  
 pulsazione di taglio, retta  $\log \omega \mapsto$   
 $-20 \log \frac{1}{|\tau|} + 20 \log \omega$  con pendenza 20  
 dB/dec.



Nota (diagramma reale) → scostamento  
 massimo per  $\omega = \frac{1}{|\tau|}$  dove  $|G_e(j\omega)|_{dB} =$   
 $20 \log \sqrt{1 + 1} = 20 \log \sqrt{2} \approx 3$

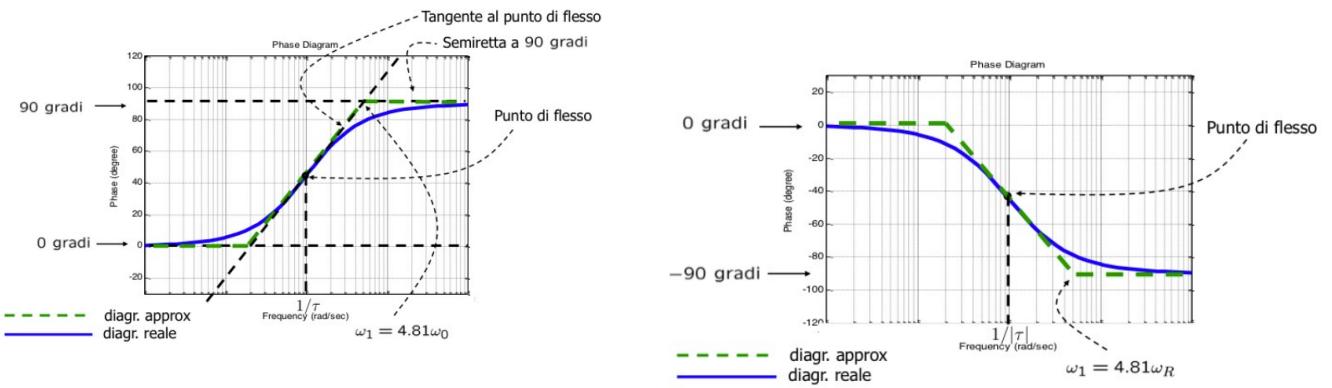
**Fase:**

Consideriamo  $G_e(j\omega) = 1 + j\omega\tau, \quad \tau < 0$   
 (zero reale negativo)

$$\arg(G_e(j\omega)) = \arg(1 + j\omega\tau) \approx \begin{cases} 0 & \omega \ll \frac{1}{\tau} \\ 90^\circ & \omega \gg \frac{1}{\tau} \end{cases}$$

Consideriamo  $G_e(j\omega) = 1 + j\omega\tau, \quad \tau > 0$  (zero  
 reale positivo)

$$\arg(G_e(j\omega)) = \arg(1 + j\omega\tau) \approx \begin{cases} 0 & \omega \ll \frac{1}{\tau} \\ -90^\circ & \omega \gg \frac{1}{\tau} \end{cases}$$

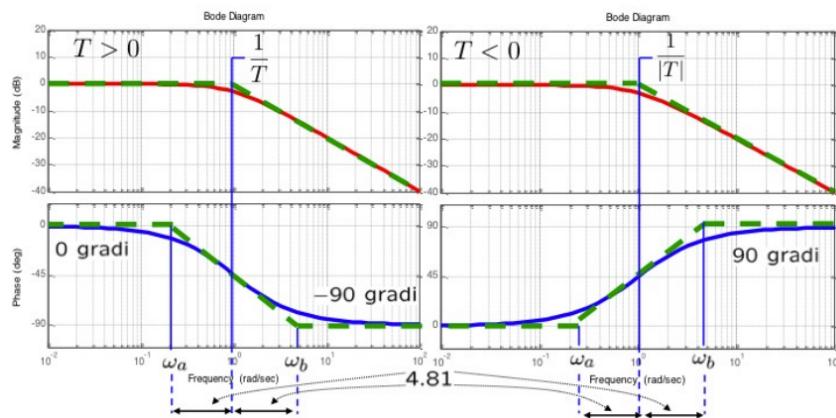


**Nota (diagramma reale) importante:** il cambio di fase inizia circa una decade prima e finisce circa una decade dopo la pulsazione di taglio.

### Polo reale - sistemi del primo ordine

Consideriamo  $G_e(j\omega) = \frac{1}{1+j\omega T}$  (polo reale)

$$|G_e(j\omega)|_{dB} = -20 \log \sqrt{1 + \omega^2 T^2} \quad \arg(G_e(j\omega)) = -\arg(1 + j\omega T)$$



**Nota:** diagramma ottenuto da quello dello zero ribaltando rispetto all'asse reale (consistente con il segno di  $T$ ).

Considerando solo il caso  $T > 0$  (polo reale negativo)  $\rightarrow$  (diagramma asintotico) fino alla pulsazione di taglio, andamento costante a 0 dB; a partire dalla pulsazione di taglio retta  $\log \omega \mapsto 20 \log \frac{1}{T} - 20 \log \omega$  con pendenza -20 dB/dec.

(diagramma reale)  $\rightarrow$  scostamento massimo per  $\omega = \frac{1}{T}$  dove  $|G_e(j\omega)|_{dB} = -20 \log \sqrt{1+1} = -20 \log \sqrt{2} \approx -3$ ; il cambio di fase inizia come al solito una decade prima e finisce circa una decade dopo.

### Zeri complessi coniugati - sistemi del secondo ordine

Consideriamo  $G_d(j\omega) = 1 + 2j7\zeta \frac{\omega}{\alpha_n} - \frac{\omega^2}{\alpha_n^2}$  (coppia di zeri complessi coniugati)

**Ampiezza:**

$$|G_d(j\omega)|_{dB} = 20 \log \sqrt{\left(1 - \frac{\omega^2}{\alpha_n^2}\right) + 4\zeta^2 \frac{\omega^2}{\alpha_n^2}} \approx \begin{cases} 20 \log(1) = 0 & \omega \ll \alpha_n \\ 20 \log \frac{\omega^2}{\alpha_n^2} = -40 \log \alpha_n + 40 \log \omega & \omega \gg \alpha_n \end{cases}$$

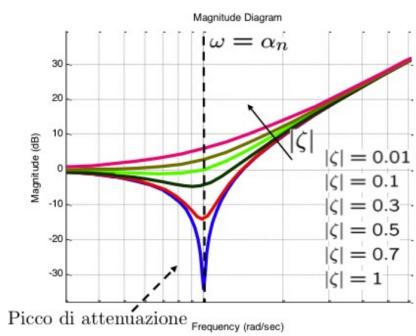
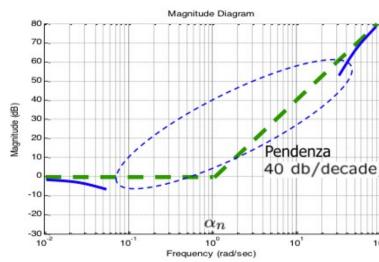
**Nota (diagramma asintotico)**

→ a partire da  $\omega = \alpha_n$ , pulsazione di taglio, retta  $\log \omega \mapsto -40 \log \alpha_n + 40 \log \omega$  con pendenza 40 dB/dec.

**Nota (diagramma reale)** → scostamento significativo nella zona tratteggiata dipendente dal valore di  $\zeta$ .

$$\text{Minimo a pulsazione } \omega_r = \alpha_n \sqrt{1 - 2\zeta^2} \text{ con } |G_d(j\omega_r)| = 2|\zeta| \sqrt{1 - \zeta^2}$$

Il picco di attenuazione dipende strettamente dal valore di  $\zeta$ , tanto più e piccolo tanto più sarà accentuato.

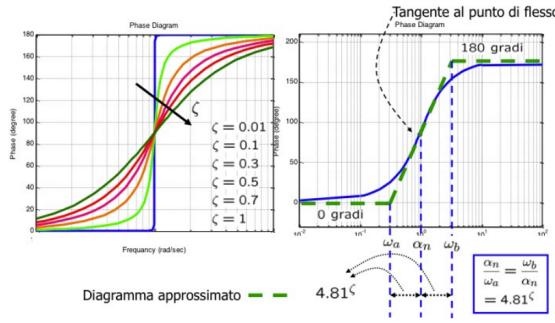


### Fase:

**Nota (diagramma reale)** → il cambio di fase dipende in modo significativo dal valore di  $\zeta$ .

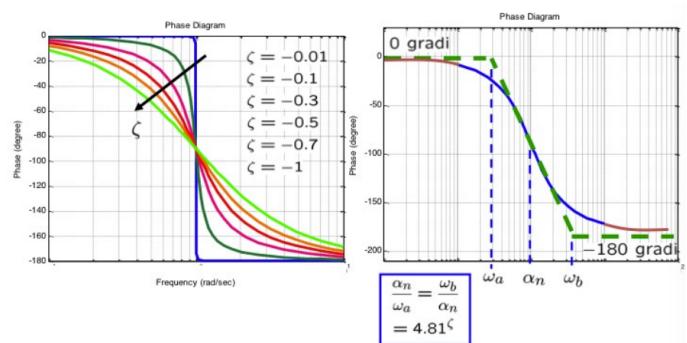
- Parte reale negativa:

$$\arg(G_c(j\omega)) \approx \begin{cases} 0 & \omega \ll \alpha_n \\ 180^\circ & \omega \gg \alpha_n \end{cases}$$



- Parte reale positiva:

$$\arg(G_c(j\omega)) \approx \begin{cases} 0 & \omega \ll \alpha_n \\ -180^\circ & \omega \gg \alpha_n \end{cases}$$



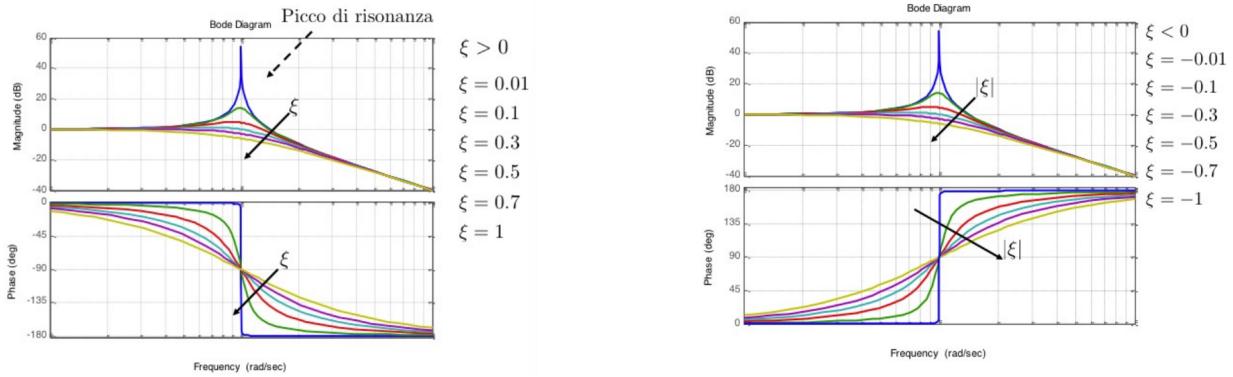
### Poli complessi coniugati - sistemi del secondo ordine

Consideriamo  $G_d(j\omega) = \frac{1}{1 + 2j\xi \frac{\omega}{\omega_n} - \frac{\omega^2}{\omega_n^2}}$  (coppia di poli complessi coniugati)

Diagrammi ottenuti da quello degli zeri ribaltando rispetto all'asse reale (caso  $\xi > 0$  corrispondente a  $\zeta > 0$  e viceversa).

Parte reale negativa: ( $\xi > 0$ )

Parte reale positiva: ( $\xi < 0$ )



Picco di risonanza alla pulsazione di risonanza  $\omega_r = \omega_n \sqrt{1 - 2\xi^2}$  con  $|G_d(j\omega_r)| = \frac{1}{2|\xi|\sqrt{1-\xi^2}}$

Alla frequenza  $\omega_n$  si ha  $|G_d(j\omega_n)| = \frac{1}{2|\xi|}$

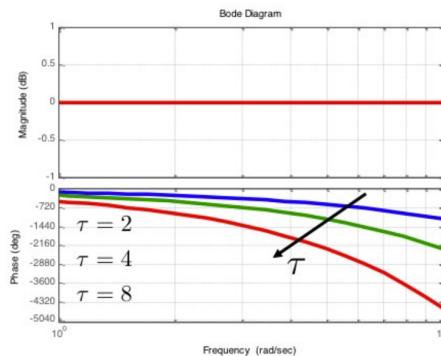
### Ritardo temporale

Consideriamo  $G(s) = e^{-\tau s}$  con

$$G(j\omega) = e^{-j\omega\tau}$$

$$|G(j\omega)|_{dB} = 20 \log |e^{-j\omega\tau}| = 20 \log 1 = 0$$

$$\arg G(j\omega) = \arg e^{-j\omega\tau} = -\omega\tau$$



### Risonanza

Supponiamo di avere un sistema con poli immaginari coniugati  $\pm j\omega_n$ , ovvero  $G(s) = \mu \frac{\omega_n^2}{s^2 + \omega_n^2}$ . Il diagramma di Bode ha un picco di risonanza infinito corrispondente alla pulsazione  $\omega_n$ . Verifichiamolo calcolando l'uscita del sistema in corrispondenza dell'ingresso  $u(t) = U \cos(\omega_n t)$ . La trasformata dell'ingresso è  $U(s) = U \frac{s}{s^2 + \omega_n^2}$ , quindi quella dell'uscita è:

$$Y(s) = G(s)U(s) = \mu \frac{U \omega_n^2 s}{(s^2 + \omega_n^2)^2} = \frac{k_1}{s - j\omega_n} + \frac{\bar{k}_1}{s + j\omega_n} + \frac{k_2}{(s - j\omega_n)^2} + \frac{\bar{k}_2}{(s + j\omega_n)^2}$$

dove nell'ultima uguaglianza abbiamo descritto lo sviluppo in fratti semplici per poli multipli. A causa della presenza dei termini al quadrato del denominatore, avremo un'uscita del tipo

$$y(t) = A_1 t \cos(\omega_n t + \phi_1) + A_2 \cos(\omega_n t + \phi_2)$$

il cui valore assoluto tende ad infinito per  $t \rightarrow \infty$ .

## Lezione 8 - diagrammi di Bode in MATLAB

Prendiamo come esempio la funzione di trasferimento:  $G(s) = \frac{10}{1 + 0.1s}$  e la sua risposta in frequenza  $G(j\omega) = \frac{10}{1 + 0.1j\omega}$

$G(s) = \mu \rightarrow$  in Matlab  $\rightarrow$  `G = tf(mu)`

E per quanto riguarda il diagramma di Bode si utilizza la funzione `bode(G)` (e per colorarla, ad esempio di rosso, `bode(G, 'r')`)

```
clear all; close all; clc;

mu = 10;
T = 0.1;

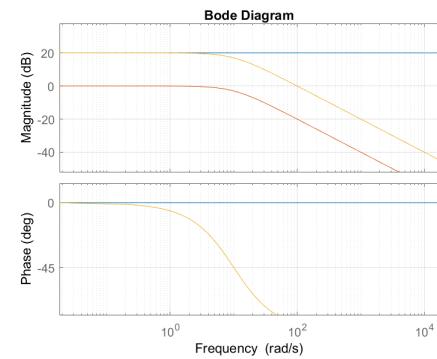
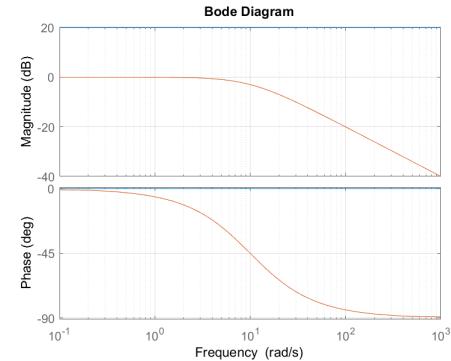
% prima funzione, per il guadagno statico
G1 = tf(mu);

% diagramma di Bode
bode(G1); grid on; zoom on; hold on;

% definisco la variabile di Laplace (da fare per forza)
s = tf('s');
G2 = 1/(1+T*s);
bode(G2);

% prima immagine - G1 in blu, G2 in rosso

% scriviamo la funzione definitiva - seconda immagine
% G in giallo
G = mu/(1+T*s);
bode(G);
```



## Sistemi del primo ordine - ingresso sinusoidale

```
% Sistemi del primo ordine - ingresso sinusoidale
%
close all; clear all; clc;

% parametri del sistema
mu = 1; % guadagno
T = 0.1; % costante di tempo

% parametri della sinusoide in ingresso
omega = 1e2; % pulsazione: 100 rad/s
tt = 0:1e-3:0.3; % intervallo temporale: da 0 ad 0.3 con passo 0.001

%% simulazione

fprintf('Pulsazione sinusoide in ingresso: %.2f rad/s (%.2f Hz)\n', omega, omega/(2*pi));
fprintf('Pulsazione di taglio a -3dB: %.2f rad/s (%.2f Hz)\n\n', 1/T, 1/(T^2*pi));

fprintf('Azione filtrante del sistema:\n');
fprintf('Guadagno: %.2f dB (cioè fattore moltiplicativo: %.2f)\n', -20*log10(sqrt(1+omega^2*T^2)), 1/sqrt(1+omega^2*T^2));
fprintf('Sfasamento: %.2f gradi\n', rad2deg(-atan(omega*T)));

% definizione del sistema
s = tf('s');
G = mu/(1+T*s);

% definizione dell'ingresso
uu = cos(omega*tt);
```

```
% simulazione sistema
yy = lsim(G, uu, tt);

%% plot

% diagramma di bode
figure;
bode(G);
box on; zoom on; grid on;

% plot ingresso-uscita
figure;
hold on; box on; zoom on; grid on;
plot(tt,uu,'LineWidth',1,'DisplayName','u(t)');
plot(tt,yy,'LineWidth',1,'DisplayName','y(t)');
xlabel('t');
legend;
```

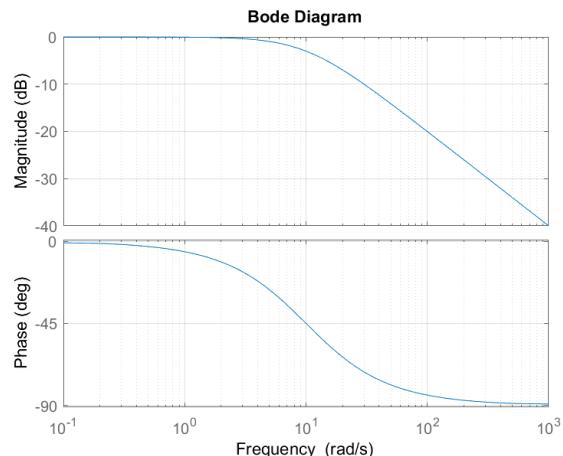
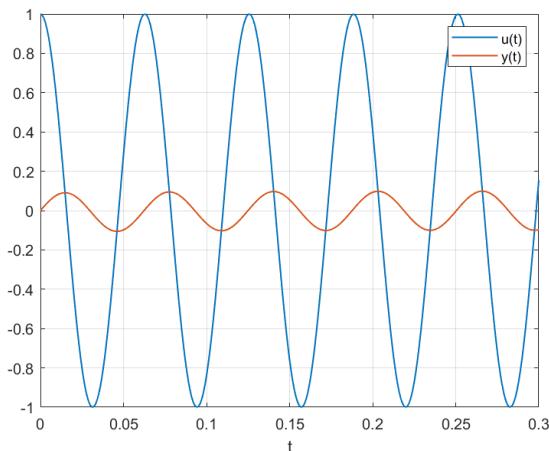
Pulsazione sinusoidale in ingresso: 100.00 rad/s (15.92 Hz)

Pulsazione di taglio a -3dB: 10.00 rad/s (1.59 Hz)

Azione filtrante del sistema:

Guadagno: -20.04 dB (cioè fattore moltiplicativo: 0.10)

Sfasamento: -84.29 gradi



### Diagramma di Bode e risposta ad ingresso sinusoidale per un sistema complesso

```
% Diagramma di bode e risposta ad ingresso sinusoidale per un sistema complesso
%
close all; clear all; clc;

% funzione di trasferimento
s = tf('s');
G = 50*(5-s)^2/(s*(s^2+3*s+900)*(10*s+5));

% parametri della sinusode in ingresso
omega = 0.1;    % pulsazione: 0.1 rad/s
tt = 0:0.01:500; % intervallo temporale: da 0 a 500 con passo 0.01

%% simulazione

risp_freq = evalfr(G, 1j*omega); % risposta in frequenza G(s) valutata per s = j*omega

fprintf('Pulsazione sinusoidale in ingresso: %.2f rad/s (%.2f Hz)\n', omega, omega/(2*pi));
fprintf('Guadagno attraverso sistema: %.2f dB (cioè fattore moltiplicativo: %.2f)\n', 20*log10(abs(risp_freq)), abs(risp_freq));
```

```
% definizione dell'ingresso
uu = cos(omega*tt);

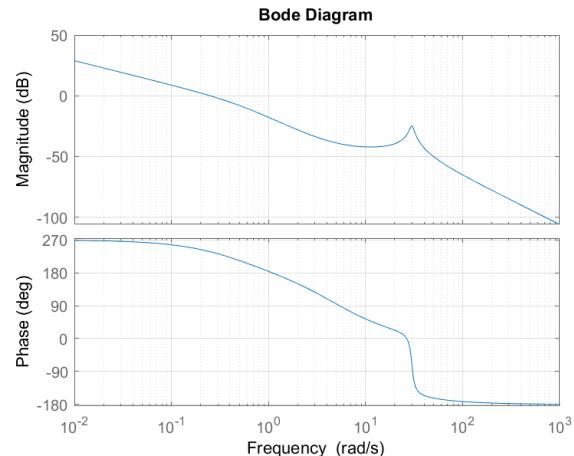
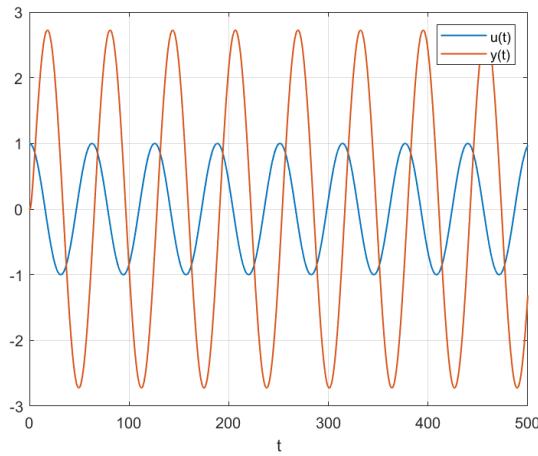
% simulazione sistema
yy = lsim(G, uu, tt);

%% plot

% diagramma di bode
figure;
bode(G);
box on; zoom on; grid on;

% plot ingresso-uscita
figure;
hold on; box on; zoom on; grid on;
plot(tt,uu,'LineWidth',1,'DisplayName','u(t)');
plot(tt,yy,'LineWidth',1,'DisplayName','y(t)');
xlabel('t');
legend;
```

Pulsazione sinusoidale in ingresso: 0.10 rad/s (0.02 Hz)  
Guadagno attraverso sistema: 8.71 dB (cioè fattore moltiplicativo: 2.72)



## Sistemi del secondo ordine - ingresso sinusoidale

```
%  

% Sistema del secondo ordine: risposta ad ingresso sinusoidale  

%  

close all; clear all; clc;  

% parametri del sistema  

mu = 1; % guadagno  

xi = 0.2; % smorzamento  

omegan = 130; % pulsazione naturale: 130 rad/s  

% parametri della sinusode in ingresso  

omega = 1e2; % pulsazione: 100 rad/s  

tt = 0:1e-3:0.5; % intervallo temporale: da 0 ad 0.5 con passo 0.001  

%% simulazione  

fprintf('Pulsazione sinusoidale in ingresso: %.2f rad/s (%.2f Hz)\n', omega, omega/(2*pi));  

fprintf('Pulsazione di taglio: %.2f rad/s (%.2f Hz)\n', omegan, omegan/(2*pi));  

fprintf('Picco di risonanza: %.2f dB in corrispondenza della pulsazione %.2f rad/s (%.2f Hz)\n\n',
-20*log10(2*abs(xi)*sqrt(1-xi^2)), omegan*sqrt(1-2*xi^2), omegan*sqrt(1-2*xi^2)/(2*pi));  

fprintf('Azione filtrante del sistema:\n');  

fprintf('Guadagno: %.2f dB (cioè fattore moltiplicativo: %.2f)\n',
-20*log10(sqrt((1-omega^2/omegan^2)^2 + 4*xi^2*omega^2/omegan^2)),
1/sqrt((1-omega^2/omegan^2)^2 + 4*xi^2*omega^2/omegan^2));
```

```

fprintf('Sfasamento: %.2f gradi\n', rad2deg(-atan2(2*xi*omega/omegan, 1-omega^2/omegan^2)));

% definizione del sistema
s = tf('s');
G = mu*omegan^2/(s^2 + 2*xi*omegan*s + omega^2);

% definizione dell'ingresso
uu = cos(omega*t);

% simulazione sistema
yy = lsim(G, uu, tt);

%% plot

% diagramma di bode
figure;
bode(G);
box on; zoom on; grid on;

% plot ingresso-uscita
figure;
hold on; box on; zoom on; grid on;
plot(tt,uu,'LineWidth',1,'DisplayName','u(t)');
plot(tt,yy,'LineWidth',1,'DisplayName','y(t)');
xlabel('t');
legend;

```

Pulsazione sinusoide in ingresso: 100.00 rad/s (15.92 Hz)

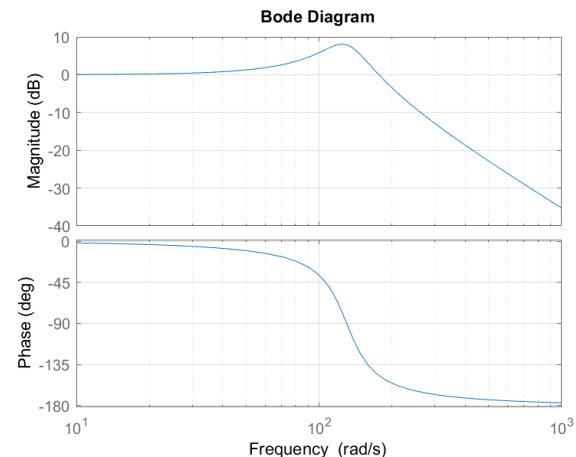
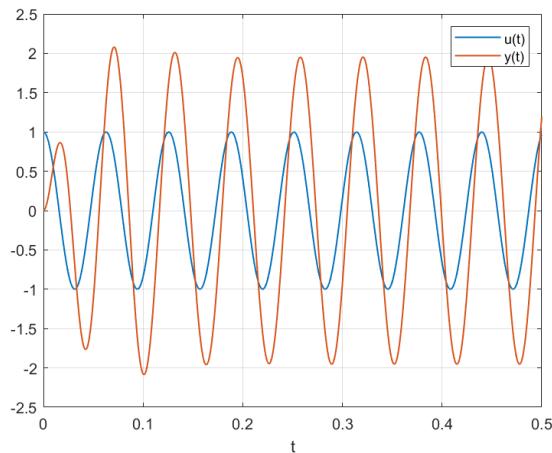
Pulsazione di taglio: 130.00 rad/s (20.69 Hz)

Picco di risonanza: 8.14 dB in corrispondenza della pulsazione 124.69 rad/s (19.85 Hz)

Azione filtrante del sistema:

Guadagno: 5.83 dB (cioè fattore moltiplicativo: 1.96)

Sfasamento: -37.00 gradi



## Esempio di risonanza (carrello)

```

%
% Esempio Carrello: risonanza
%
close all; clear all; clc;

% parametri fisici del sistema
mass = 0.5; % kg (parametro M)
elas = 2; % costante elastica [N/m] (parametro k)
smorz = 0.1; % costante dello smorzatore (parametro b) - NON coincide con smorzamento xi

% condizione iniziale pari a 0 (evoluzione libera)
pos_init = 0; % [m]

```

```

vel_init = 0; % [m/s]

% calcolo di guadagno, pulsazione naturale e coefficiente di smorzamento
mu = 1/elas; % 1/k
omegan = sqrt(elas/mass); % sqrt(k/M)
xi = smorz/(2*sqrt(elas*mass)); % b/(2 sqrt(kM))

fprintf('Coeffienti sistema:\n');
fprintf('\tGuadagno: %.2f\n', mu);
fprintf('\tPulsazione naturale: %.2f\n', omegan);
fprintf('\tSmorzamento: %.2f\n', xi);

% input applicato: sinusoidale con pulsazione omega_n
inp = @(t) sin(omegan.*t);

% intervallo di tempo
interv = 0:0.1:20; % da 0 a 20 secondi con passo 0.1

%% creazione oggetto sistema e soluzione equazione differenziale

s = tf('s');
G = mu * omegan^2 / (s^2 + 2*xi*omegan*s + omegan^2);

% evoluzione sistema
uu = inp(interv);
x0 = [pos_init; vel_init];
[YY, TT, XX] = lsim(G, uu, interv, x0);

%% plot con animazione

% parametri del disegno
r=0.25; %raggio
Wrad = 0:.02:2*pi;
Wx = r*cos(Wrad);
Wy = r*sin(Wrad);

% inizializza figura
figure(1);
clf;
min_T=0;
max_T=max(TT);
min_Y=min(YY);
max_Y=max(YY);
subplot(2,1,1)
axis([min_T max_T (min_Y-0.2) (max_Y+0.2)])
subplot(2,1,2)
axis([-10 10 0 5])

for i=1:5:length(TT)
    figure(1);
    clf;

    % aggiorna figura superiore
    subplot(2,1,1)
    plot(TT(1:i), YY(1:i),'b', TT(1:i), uu(1:i), 'r');
    axis([min_T max_T (min_Y-0.2) (max_Y+0.2)])
    title(['Evoluzione del sistema al tempo: ',num2str(TT(i)), ' [s]']);
    legend('Uscita y(t)', 'Ingresso u(t)');
    grid on; box on;

    % aggiorna figura inferiore
    subplot(2,1,2)
    axis([-10 10 0 5])
    patch(YY(i)+[-2 2 2 -2], 2+[1.5 1.5 -1.5 -1.5], 'y') % cart
    hold on;
    patch(Wx+YY(i)-1.8, Wy+.25, 'r'); % wheel
    patch(Wx+YY(i)+1.8, Wy+.25, 'r'); % wheel
    plot([-10 -10 20],[5 0 0], 'k', 'LineWidth', 4) % ground and wall.
    plot([-10, -9, -9:(+9+YY(i)-4)/9:YY(i)-4, YY(i)-4, YY(i)-2],...
        2+1+[0 0 .5 -.5 .5 -.5 .5 -.5 .5 -.5 0 0 0], 'r', 'LineWidth', 2) % spring
    title('Risposta del sistema')

    % pausa prima del prossimo frame
    pause(0.0001);
end

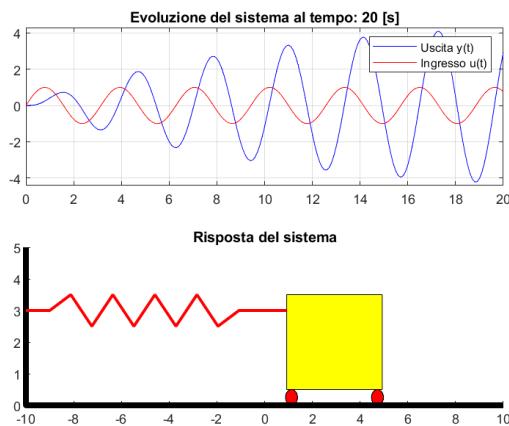
```

Coefficienti sistema:

Guadagno: 0.50

Pulsazione naturale: 2.00

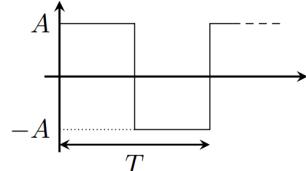
Smorzamento: 0.05



### Esempio - onda quadra con disturbo in alta frequenza

Considerando un segnale  $s(t)$  ad onda quadra con periodo  $T$  e ampiezza  $A$ :

$$s(t) = A \operatorname{sign} \left( \sin \left( \frac{2\pi t}{T} \right) \right) =$$
$$\begin{cases} A & \text{se } \sin \left( \frac{2\pi t}{T} \right) \geq 0 \\ -A & \text{se } \sin \left( \frac{2\pi t}{T} \right) < 0 \end{cases}$$



Supponiamo che l'ingresso consista nell'onda quadra più un disturbo sinusoidale ad alta frequenza  $\omega_N \gg \frac{2\pi}{T}$  con ampiezza  $A_N$ :

»

Per filtrare il disturbo in alta frequenza possiamo usare un filtro passabasso, cioè un sistema dinamico del primo ordine con un polo reale in  $-1/T_p$ :  $G(s) = \frac{1}{1+sT_p}$ . Scegliendo opportunatamente la costante di tempo  $T_p$  il segnale in uscita sarà una versione filtrata dell'onda quadra, con il disturbo quasi completamente attenuato.

```
%  
% Azione filtrante dei sistemi dinamici: esempio onda quadra  
%  
close all; clear all; clc;  
  
% Sistema: filtro PASSA-BASSO con pulsazione di taglio in a 10 rad/s  
mu = 1; % guadagno  
T = 0.05; % costante di tempo  
  
% parametri dell'onda quadra in ingresso  
omega = 2; % pulsazione: 2 rad/s  
tt = 0:1e-3:10; % intervallo temporale: da 0 ad 10 con passo 0.01  
  
% parametri del rumore  
enable_noise = false;  
omega_noise = 1000;  
amplitude_noise = 1;  
  
%% simulazione  
  
fprintf('Pulsazione segnale in ingresso: %.2f rad/s (%.2f Hz)\n', omega, omega/(2*pi));  
fprintf('Pulsazione di taglio a -3dB: %.2f rad/s (%.2f Hz)\n\n', 1/T, 1/(T*2*pi));
```

```
% definizione del sistema
s = tf('s');
G = mu/(1+T*s);

% definizione dell'ingresso
uu = sign(sin(omega*tt));

if enable_noise
    uu = uu + amplitude_noise*sin(omega_noise*tt);
end

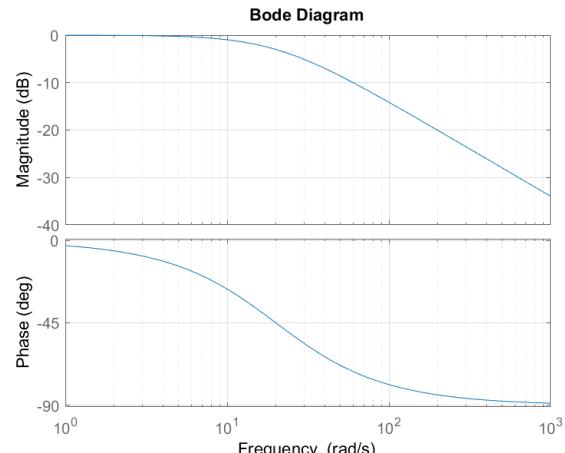
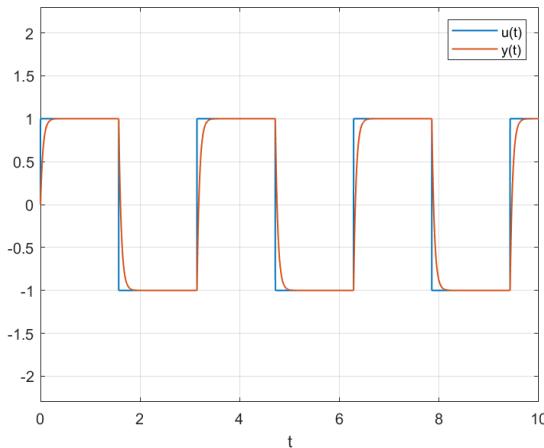
% simulazione sistema
yy = lsim(G, uu, tt);

%% plot

% diagramma di bode
figure;
bode(G);
box on; zoom on; grid on;

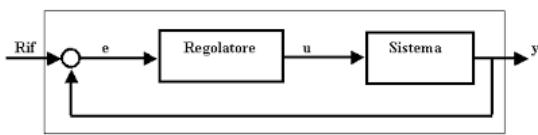
% plot ingresso-uscita
figure;
hold on; box on; zoom on; grid on;
plot(tt,uu,'LineWidth',1,'DisplayName','u(t)');
plot(tt,yy,'LineWidth',1,'DisplayName','y(t)');
ylim([-2.3,2.3]);
xlabel('t');
legend;
```

Pulsazione segnale in ingresso: 2.00 rad/s (0.32 Hz)  
Pulsazione di taglio a -3dB: 20.00 rad/s (3.18 Hz)



## Lezione 9 - sistemi in retroazione ad anello chiuso

@November 24, 2022



In ingresso  $w$ , in uscita  $y$ ; ( $L = R(s) + G(s)$ )

$$\mu = 100; \quad T_1 = 1; \quad T_2 = 1 \cdot 10^{-2}; \quad T_3 = 1 \cdot 10^{-3}$$

```
clear all; close all; clc;

s = tf('s');
```

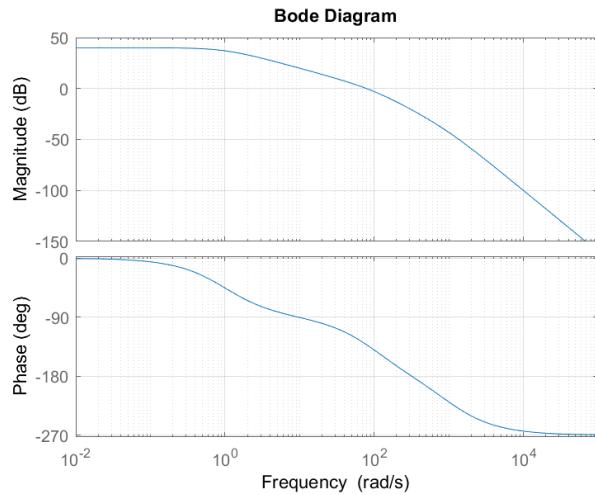
```

mu = 100;
T1 = 1;
T2 = 10^-2;
T3 = 10^-3;

L = mu/((1+T1*s)*(1+T2*s)*(1+T3*s));

bode(L); grid on;

```



**Margine di fase:**  $M_f = 180^\circ + \arg(L(j\omega_c))$  con  $\omega_c$  t.c.  $|L(j\omega_c)|_{dB} = 0$

**Margine di ampiezza:**  $M_a = -|L(j\omega_\pi)|_{dB}$  con  $\omega_\pi$  t.c.  $\arg(L(j\omega_\pi)) = -180^\circ$

Nella L non devono esserci poli reali a parte strettamente positiva, deve esserci un solo attraversamento, e se a quel punto il margine di fase è positivo il sistema è bilanciato.

In MATLAB la funzione che ci permette di calcolare margini di fase e di ampiezza si chiama `margin(L)` e da in uscita 4 output:

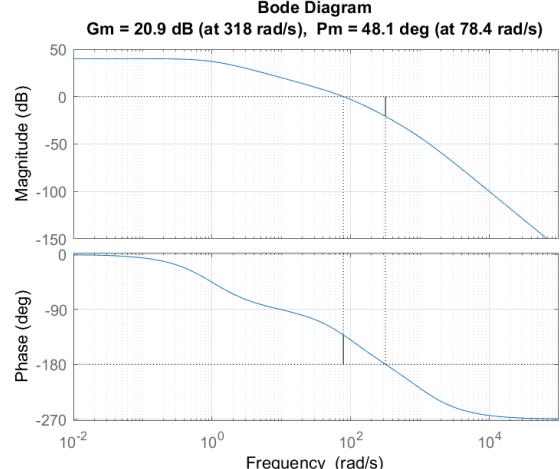
- margine di ampiezza  $M_a$ ;
- margine di fase  $M_f$ ;
- la frequenza alla quale si attraversa il  $\pi$  ovvero  $\omega_\pi$ ;
- la frequenza alla quale si attraversa la pulsazione critica  $\omega_c$ .

% con le stesse variabili di prima

```

[M_a, M_f, omega_pi, omega_c] = margin(L);
figure; margin(L); grid on;
% margin fa in automatico il diagramma di Bode

```



## Casi patologici

(cambiare 0 in 1 a seconda del caso che si vuole testare)

```

clear all; close all; clc;

s = tf('s');

%% margini ben definiti
if 1
    mu = 100;

```

```

T1 = 1;
T2 = 10^-2;
T3 = 10^-3;
L = mu/((1+T1*s)*(1+T2*s)*(1+T3*s));
end

%% intersezioni multiple
if 0
    mu = 10;
    xi = 0.01;
    omega_n = 100;
    L = mu/(s*(1+2*xi/omega_n*s + s^2/omega_n^2))
end

%% zero intersezioni
if 0
    mu = 0.1;
    tau = 0.1;
    TT = 1;
    L = mu*(1+tau*s)/(1+TT*s)
end

%% ritardo in ingresso
if 0
    mu = 0.1;
    TT = 1;
    TT2 = 0.1;

    tau = 14;
    L = exp(-tau*s)*mu/(s*(1+TT*s)*(1+TT2*s))
end

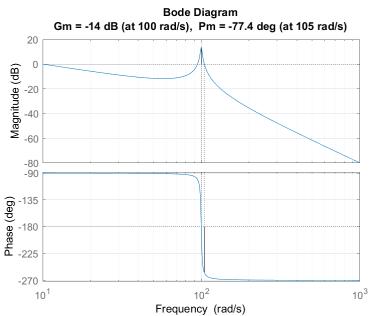
%% incertezze sul guadagno
if 1
    % mu = 0.1 sarebbe corretto
    % sbagliamo mu e lo mettiamo un'ordine di grandezza + grande -> va ancora bene ma il margine è molto piccolo
    mu = 0.1*150;
    TT = 1;
    TT2 = 0.1;

    L = mu/((1+TT*s)*(1+TT2*s))
end

[M_a, M_f, omega_pi, omega_c] = margin(L);
figure; margin(L); grid on;

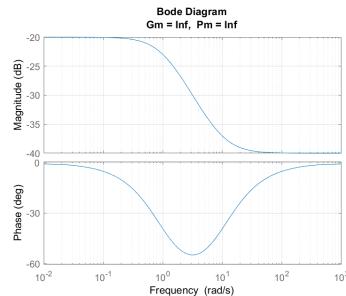
```

Caso intersezioni multiple:

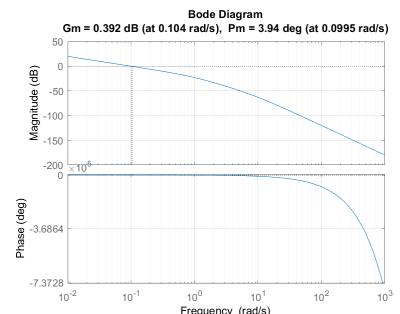


**Warning:** The closed-loop system is unstable.

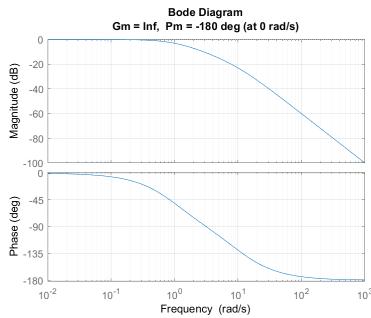
Caso zero intersezioni:



Caso di ritardo in ingresso:



Caso incertezze sul guadagno:



## Funzioni di sensitività

L'uscita del sistema dipenderà ovviamente dal sistema, ma anche dai disturbi in uscita. Per considerare anche questi disturbi sono necessarie le **funzioni di sensitività** → funzioni di trasferimento tra ingressi e uscite di interesse:

$$S(s) = \frac{1}{1 + R(s)G(s)} \quad \text{funzione di sensitività;}$$

$$F(s) = \frac{R(s)G(s)}{1 + R(s)G(s)} \quad \text{funzione di sensitività complementare;}$$

$$Q(s) = \frac{R(s)}{1 + R(s)G(s)} \quad \text{funzione di sensitività del controllo.}$$

```

clear all; close all; clc;

s = tf('s');

mu = 40;
TT = 10;
TT2 = 2;
TT3 = 0.2;

L = mu/((1+TT*s)*(1+TT2*s)*(1+TT3*s));

% funzione di sensitività complementare
F = L/(1+L);

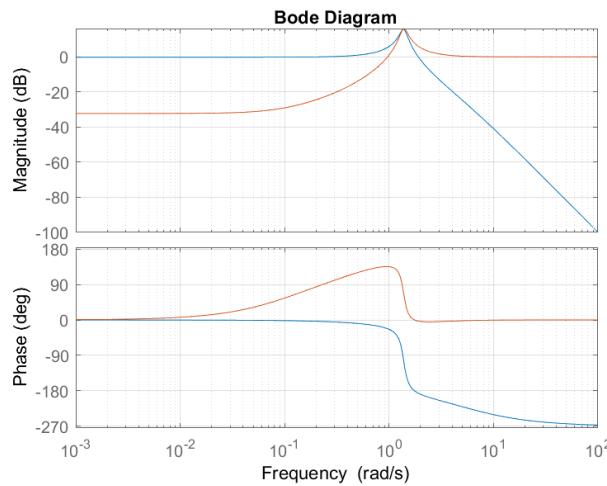
% per vedere la forma fattorizzata usiamo questa funzione
F_fattorizzata = zpk(F);

% F definitiva con cui lavoreremo
% minreal passa allo spazio degli stati
% dovremo rispostarci nelle frequenze
FF = tf(minreal(F));

%% funzione di sensitività semplic
S = 1/(1+L);
S_fattorizzata = zpk(S);
SS = tf(minreal(S));

figure; bode(FF); hold on; bode(SS); grid on;

```



Osserviamo i poli in retroazione con l'esempio di prima:

```

clear all; close all; clc;

s = tf('s');

%% margini ben definiti
if 1
    mu = 100;
    T1 = 1;
    T2 = 10^-2;
    T3 = 10^-3;
    L = mu/((1+T1*s)*(1+T2*s)*(1+T3*s));
end

%% incertezze sul guadagno
if 0
    mu = 0.1*150;
    TT = 1;
    TT2 = 0.1;
    L = mu/((1+TT*s)*(1+TT2*s))
end

FF = tf(minreal(L/(1+L)));
poli_retroazione = pole(FF);

[M_a, M_f, omega_pi, omega_c] = margin(L);
figure; margin(L); grid on;

```

Caso margini ben definiti:

```

poli_retroazione =
1.0e+03 *
1.0109 + 0.0000i
-0.0451 + 0.0892i
-0.0451 - 0.0892i

```

Caso incertezza sul guadagno:

```

poli_retroazione =
5.5000 +11.3908i
-5.5000 -11.3908i

```

## Graficare schemi a blocchi in retroazione su MATLAB

Da riga di comando per graficare su MATLAB sistemi in retroazione:

```

FF.InputName = 'ref';
FF.OutputName = 'output';

```

Oppure nel codice:

```

clear all; close all; clc;

s = tf('s');

mu = 40;
TT = 10;
TT2 = 2;
TT3 = 0.2;

L = mu/((1+TT*s)*(1+TT2*s)*(1+TT3*s));

%% graficare lo schema a blocchi in retroazione
% modo 1
if 0
    Sum = sumblk('error = ref - output');
    L.InputName = 'error';
    L.OutputName = 'output';
    FF = connect(L,Sum,'ref','output');
    FF = tf(minreal(FF));
end

% modo 2 - feedback
if 1
    FF = feedback(L,1);
    FF = tf(minreal(FF))
end

```

$FF =$

*From input "ref" to output "output":*

10

-----  
 $s^3 + 5.6 s^2 + 3.05 s + 10.25$

*Continuous-time transfer function.*

## Risposta a sistemi in retroazione in MATLAB con funzioni di sensitività

```

clear all; close all; clc;

s = tf('s');
mu = 40;
TT = 10;
TT2 = 2;
TT3 = 0.2;
L = mu/((1+TT*s)*(1+TT2*s)*(1+TT3*s));

% definiamo le due funzioni di sensitività
FF = L/(1+L);
SS = 1/(1+L);

% definisco un intervallo temporale
tt = (0:1e-2:2e3);
omega_w = 0.01; % ingresso sinusoidale
omega_n = 0.05; % rumore

```

```

ww = cos(omega_w*tt); % ingresso
dd = cos(omega_n*tt); % disturbo

% per fare le risposte usiamo la sovrapposizione degli effetti

% risposta al riferimento data dal sistema modellato con funzioni di sensitività complementare
% ha per ingresso w e intervallo tt

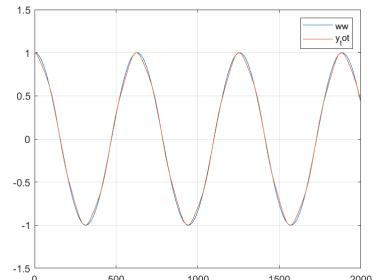
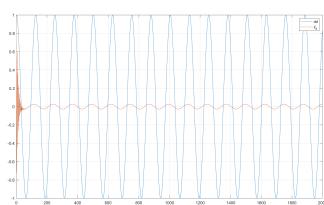
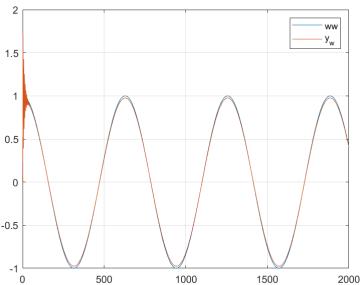
%% Risposta in anello chiuso
y_w = lsim(FF, ww, tt); % uscita dal primo sistema, ce l'aspettiamo deamplificata o sfasata
y_d = lsim(SS, dd, tt); % uscita dal secondo sistema, considerando il disturbo
y_tot = y_w + y_d; % uscita complessiva

%% plot della prima risposta
figure;
plot(tt,ww);
hold on; grid on;
plot(tt,y_w);
legend('ww','y_w')

%% plot del rumore
figure;
plot(tt,dd);
hold on; grid on;
plot(tt, y_d);
legend('dd','y_d');

%% plot della risposta complessiva
figure;
plot(tt,ww);
hold on; grid on;
plot(tt,y_tot);
legend('ww','y_{tot}');

```



A noi interessa che la nostra  $y$  finale sia più simile possibile alla  $w$ , come si vede dall'ultima figura, ci siamo quasi.

```

clear all, close all, clc

s = tf('s');

%% ESEMPIO FUNZIONI DI SENSITIVITÀ

% definizione sistema in anello aperto
mu = 40;
TT = 10;
TT2 = 2;
TT3 = 0.2;

RR = mu;
GG = 1/(1 + TT*s)/(1 + TT2*s)/(1 + TT3*s);
LL = mu * GG;
LL.Name = 'Sistema in anello aperto';
LL.OutputName = 'y';
LL.InputName = 'u';

```

```

% margini di stabilità
[M_a,M_f,omega_pi,omega_c] = margin(LL)
M_a_db = 20*log10(M_a)

% funzione di sensitività complementare
FF = LL / (1 + LL);
FF.OutputName = 'y';
FF.InputName = 'w';

% funzione di sensitività
SS = 1 / (1 + LL);
SS.OutputName = 'y';
SS.InputName = 'd';

% bode plot
figure(1)
bode(LL)
grid on, zoom on

figure(2)
bode(FF)
grid on, zoom on

figure(3)
bode(SS)
grid on, zoom on

%% ESEMPIO RISPOSTE ANELLO APERTO E CHIUSO
% Disturbo in uscita

omega_w = 0.01;
omega_d = 0.05;

tt=(0:1e-2:2e3)'; % 2000 secondi con passo 0.01

% segnali in ingresso: riferimento e disturbo di misura
WW = 1;
DD = 1;

ww = WW*cos(omega_w*tt);
dd = DD*cos(omega_d*tt);

% Risposte in anello APERTO
y_w_OPEN = lsim(GG,ww,tt);
y_tot_OPEN = y_w_OPEN + dd;

% Risposte in anello CHIUSO
y_w = lsim(FF,ww,tt);
y_d = lsim(SS,dd,tt);
y_tot = y_w + y_d;

%% RISPOSTE IN ANELLO APERTO

figure()
hold on, grid on, zoom on
plot(tt,ww,'m')
plot(tt,y_w_OPEN,'b')
grid on
legend('ww','y_{w\_OPEN}')

figure()
hold on, grid on, zoom on
plot(tt,ww,'m')
plot(tt,y_tot_OPEN,'b')
grid on
legend('ww','y_{tot\_OPEN}')

%% RISPOSTE IN ANELLO CHIUSO

figure()
hold on, grid on, zoom on
plot(tt,ww,'m')

```

```

plot(tt,y_w,'b')
grid on
legend('ww','y_w')

figure()
hold on, grid on, zoom on
plot(tt,dd,'m')
plot(tt,y_d,'b')
grid on
legend('dd','y_d')

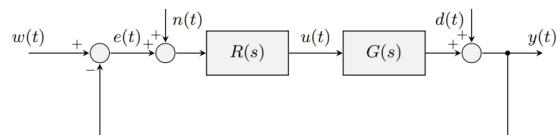
figure()
hold on, grid on, zoom on
plot(tt,ww,'m')
plot(tt,y_tot,'b')
grid on
legend('ww','y_{tot}')

```

## Lezione 10 - Sistemi di controllo: progetto del regolatore

@November 30, 2022 → December 1, 2022

### Schema di controllo in retroazione: specifiche



La specifica base è che il sistema sia stabile, perché altrimenti l'uscita  $y(t)$  sta divergendo, per di più vogliamo una stabilità robusta rispetto alle incertezze, quindi in presenza di errori di modello o incertezze di parametri (ad esempio massimo ritardo temporale  $\tau_{max}$  o massima incertezza sul guadagno statico  $\Delta\mu_{max}$ ).

#### Specifiche:

- Specifica su  $L(j\omega)$ :  $M_f \geq M_f^*$ ;
- Precisione statica: sia  $e_\infty = \lim_{t \rightarrow \infty} e(t)$  il valore a regime dell'errore in risposta a riferimenti  $w(t)$  o disturbi in uscita  $d(t)$  "canonici" (gradino  $w(t) = W1(t), d(t) = D1(t)$ , o rampa  $w(t) = t1(t), d(t) = t1(t)$ );

Specifiche:  $|e_\infty| \leq e^*$  oppure  $e_\infty = 0$

Polo nell'origine → errore nullo

Per soddisfare tali specifiche va considerata l'analisi statica effettuata sulla funzione di sensitività  $S(s)$ .

Esempi:

1.  $|e_\infty| \leq e^*$  in risposta a un gradino  $w(t) = W1(t), d(t) = D1(t)$  con  $|W| \leq W^*$  e  $|D| \leq D^*$ .

$$e_\infty = \frac{W}{1+\mu} + \frac{D}{1+\mu} = \frac{D+W}{1+\mu} \approx \frac{D+W}{\mu}$$

$$\mu = L(0) \geq \frac{D^* + W^*}{e^*}$$

2.  $e_\infty = 0$  in risposta a  $W(s) = \frac{W}{s^k}$  e/o  $D(s) = \frac{D}{s^k} \rightarrow L(s)$  deve avere  $k$  poli nell'origine.

Nota: se  $|e_\infty| \leq e^*$  in risposta a  $W(s) = \frac{W}{s^k}$  e  $D(s) = \frac{D}{s^k}$  allora  
 $k - 1$  poli in  $L(s)$  e  $\mu \geq \frac{D^* + W^*}{e^*}$

Nota: se  $e_\infty = 0$  in risposta a un disturbo sull'attuatore  $D_a(s) = \frac{D_a}{s^k}$ , allora  
 $D(s) = D_a(s)G(s)$  e  $E(s) = S(s)G(s)D_a(s)$

$k$  poli nell'origine in  $R(s)$

- Precisione dinamica:** tipicamente specifiche in termini di sovraelongazione e tempo di assestamento massimi.

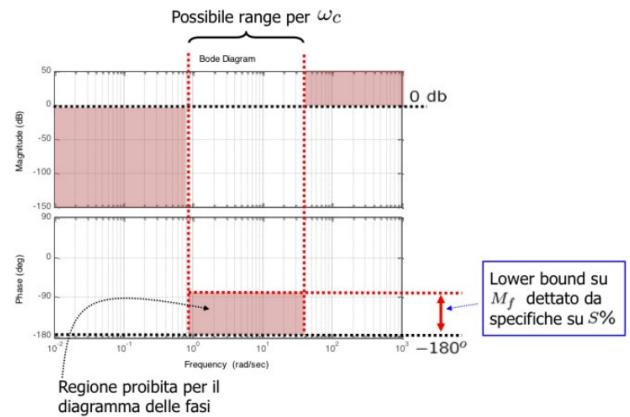
Specifiche:  $S\% \leq S^*$  e  $T_{a,\epsilon} \leq T^*$

Se progettiamo  $L(j\omega)$  in modo che  $F(j\omega)$  abbia una coppia di poli c.c. dominanti in  $\omega_n \approx \omega_c$  con coeff. smorzamento  $\xi$  allora:

$$\xi \approx \frac{M_f}{100}$$

perchè  $S\% \leq S^*$  allora  $\xi \geq \xi^*$  (con  $S^* = e^{\frac{-\pi\xi^*}{\sqrt{1-(\xi^*)^2}}}$ ) e quindi  $\rightarrow M_f \geq 100\xi^*$

perchè  $T_{a,1} \leq T^*$  allora  $\xi\omega_n \geq \frac{4.6}{T^*}$  e quindi  
 $\rightarrow M_f\omega_c \geq \frac{460}{T^*}$



La zona proibita per il diagramma di fase va evitata solo a  $\omega_c$

- Attenuazione disturbo in uscita  $d(t)$ :** il disturbo in uscita  $d(t)$ , con una banda limitata in un range di pulsazioni  $[\omega_{d,min}, \omega_{d,max}]$  deve essere attenuato di  $A_{dB}$  (con  $A_d > 0$ );

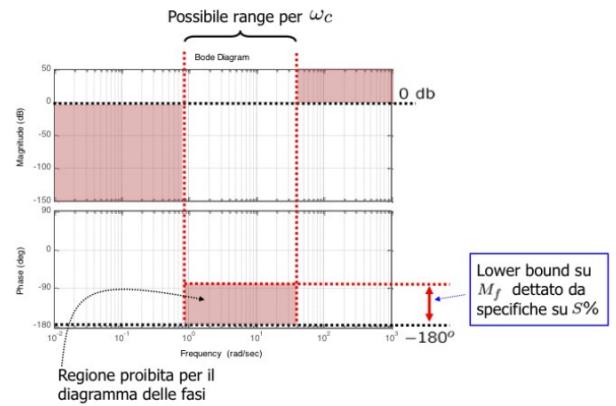
Ricordiamo che se  $d(t) = D \cos(\omega t + \varphi)$   
allora

$$y(t) = |S(j\omega)|D \cos(\omega t + \varphi + \arg(S(j\omega)))$$

e che  $|S(j\omega)|_{dB} \approx \begin{cases} -|L(j\omega)|_{dB} & \omega \leq \omega_c \\ 0 & \omega > \omega_c \end{cases}$

da specifica vogliamo  $|S(j\omega)|_{dB} \leq -A_d$  dB; poiché  $\omega_{d,max} \ll \omega_c$  si ha

$$|L(j\omega)|_{dB} \geq A_d$$



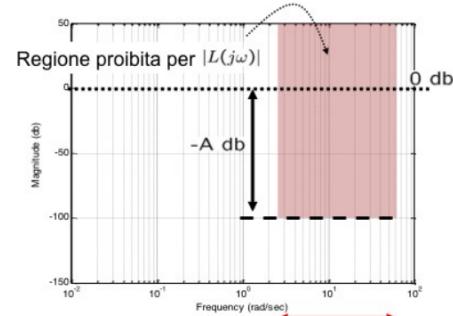
- Attenuazione disturbo di misura  $n(t)$ : il disturbo in uscita  $n(t)$ , con una banda limitata in un range di pulsazioni  $[\omega_{n,min}, \omega_{n,max}]$  deve essere attenuato di  $A_n$  dB (con  $A_n > 0$ , e in genere  $\omega_{d,max} \ll \omega_{n,min}$ );

Ricordiamo che se  $n(t) = N \cos(\omega t + \varphi)$   
allora

$$y_n(t) = |F(j\omega)|N \cos(\omega t + \varphi - \arg(F(j\omega)))$$

e che  $|F(j\omega)|_{dB} \approx \begin{cases} 0 & \omega \leq \omega_c \\ |L(j\omega)|_{dB} & \omega > \omega_c \end{cases}$

$n(t) = N \cos(\omega_n t + \varphi_n)$ ; per scriverci la  $y_n(t)$  dobbiamo porre a zero gli altri ingressi (sovraposizione degli effetti)  
 $\rightarrow y_n(s) = -F(s)N(s)$  e quindi per  $t \gg 0 \rightarrow y_n(t) \approx N|F(j\omega_n)| \cos(\omega_n t + \varphi_n - \arg(F(j\omega_n)))$



Esempio: attenuare  $n(t)$  di 100 volte  $\rightarrow 1/100$ ,  $A_n = 40$  dB  $\overline{A}_n = 100 \rightarrow |F(j\omega_n)| \leq \frac{1}{100} = \frac{1}{100}$

$$\forall \omega \in [\omega_{n,min}, \omega_{n,max}] \rightarrow |L(s\omega)|_{dB} \leq -A_n$$

- Moderazione variabile di controllo  $u(t)$ : contenimento dell'ampiezza della variabile di controllo  $u$  in ingresso al sistema fisico (impianto);

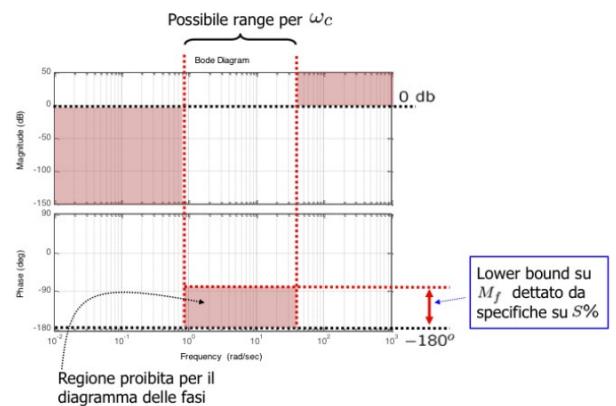
Ricordiamo che se  $w(t) = W \cos(\omega t + \varphi)$   
allora

$$u(t) = |Q(j\omega)|W \cos(\omega t + \varphi + \arg(Q(j\omega)))$$

e che

$$|Q(j\omega)|_{dB} \approx \begin{cases} -|G(j\omega)|_{dB} & \omega \leq \omega_c \\ |R(j\omega)|_{dB} & \omega > \omega_c \end{cases}$$

Poiché vogliamo contenere  $|Q(j\omega)|_{dB}$  e non abbiamo controllo su  $G(j\omega)$  dobbiamo:



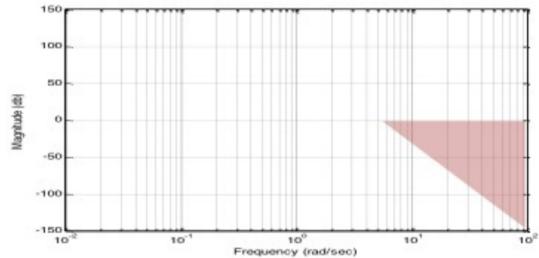
- limitare  $\omega_c$ ;
- realizzare  $R(j\omega)$  passa-basso.

- **Fisica realizzabilità del regolatore  $R(s)$ :** il regolatore deve essere un sistema proprio, quindi il grado relativo (differenza poli-zeri) deve essere maggiore o uguale a zero. (da non tenere troppo in considerazione, la diamo per scontata)

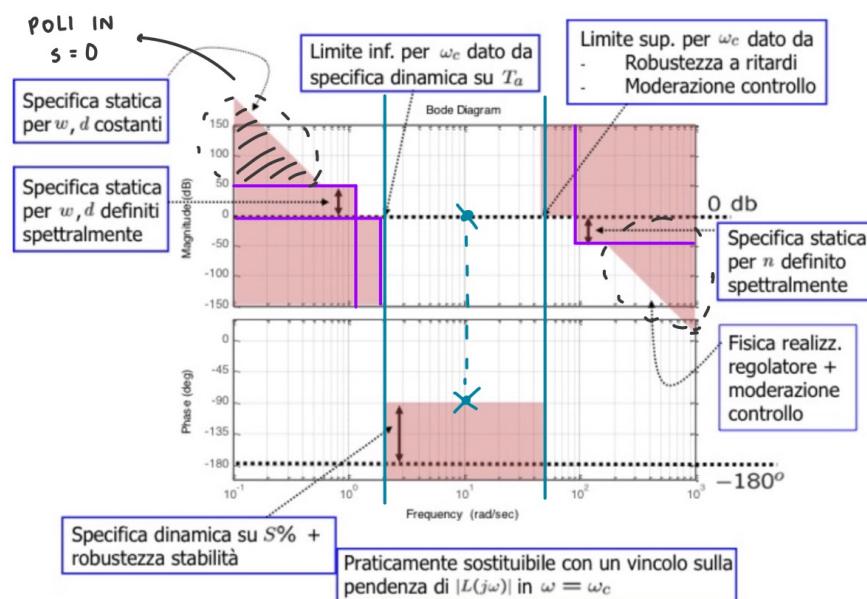
A pulsazioni elevate la pendenza  $-k_L$  dB/dec di  $|L(j\omega)|_{dB}$  è determinata dalla differenza tra poli (ciascuno contribuisce con pendenza -20dB/dec) e zeri (ciascuno contribuisce con pendenza 20dB/dec)

Se a pulsazioni elevate  $|G(j\omega)|_{dB}$  ha pendenza  $-k_G$  dB/dec allora:

$$-k_L \leq -k_G$$



## Riepilogo specifiche



## Loop shaping

O sintesi per tentativi, consiste nel "dare forma" alla  $L(j\omega)$  in modo che:

- il diagramma delle ampiezze non attraversi le "regioni proibite" in bassa e alta frequenza,
- per  $\omega = \omega_c$  rispetti il vincolo sul margine di fase,

procedendo per tentativi basati su opportune considerazioni.

## Sintesi del regolatore

Conviene dividere il progetto in due fasi fattorizzando  $R(s)$  come  $R(s) = R_s(s)R_d(s)$

**Regolatore statico:**

$R_s(s) = \frac{\mu_s}{s^k}$  progettato per soddisfare precisione statica e attenuazione disturbi  $d$ .

**Regolatore dinamico:**

$$R_d(s) = \mu_d \frac{\Pi_i(1 + \tau_i s) \Pi_i(1 + 2\frac{\zeta_i}{\alpha_{n,i}} s + \frac{s^2}{\alpha_{n,i}^2})}{\Pi_i(1 + T_i s) \Pi_i(1 + 2\frac{\xi_i}{\omega_{n,i}} s + \frac{s^2}{\omega_{n,i}^2})}$$

progettato per soddisfare stabilità robusta, precisione dinamica, attenuazione disturbi  $n$ , moderazione controllo e fisica realizzabilità.

Nota:  $\mu_d$  può essere scelto solo se  $\mu_s$  non è stato imposto.

## Lezione 11: specifiche in MATLAB

@December 1, 2022

$$G(s) = \frac{\mu}{(1 + T_1 s)(1 + T_2 s)} \quad \mu = 20, \quad T_1 = 0.1, \quad T_2 = 0.001$$

Specifiche da traccia:

1. errore a regime nullo se  $w(t) = d(t) = 1(t)$
2. attenuazione di almeno 20 dB per  $d(t) \rightarrow [\omega_{d,min}, \omega_{d,max}] = [0, 1]$
3. attenuazione di almeno 20 dB per  $n(t) \rightarrow [\omega_{n,min}, \omega_{n,max}] = [10^3, 10^4]$
4. massima sovraelongazione  $S\% < 20\%$
5. tempo di assestamento 1%  $\rightarrow T_{a,1} \leq 2s$

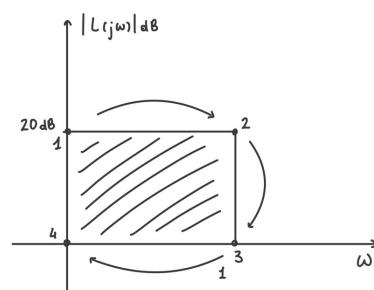
1.  $R_s(s) = \frac{1}{s} \rightarrow$  per avere l'errore a regime nullo il regolatore deve essere fatto così;

2. Sappiamo che  $y_d(s) = S(s)D(s)$  e quindi nel range di frequenze  $[0, 1]$  vorremmo che  $|S(j\omega)|_{dB} \leq -20dB$  e  $|L(j\omega)|_{dB} > 20dB$

Coordinate:

$$c_x = [\omega_{min}, \omega_{MAX}; \omega_{MAX}, \omega_{min}]$$

$c_y = [A_d; A_d; \dots; \dots] \rightarrow$  numeri a caso



3. I disturbi sull'uscita vengono mappati  $y_n(s) = -F(s)N(s)$  vogliamo attenuare di almeno 20 dB quindi  $|-F(s)|_{dB} > 20dB$  e  $|L(j\omega)|_{dB} \leq -20dB$  con  $\omega_n \in [10^3, 10^4]$
4. Massima sovraelongazione

in MATLAB:

```
clear all; close all; clc;

%% punti 2 e 3
A_d = 20; % attenuazione d
A_n = -20; % attenuazione n
```

```

omega_d_max = 1;
omega_d_min = 1e-5; % in matlab 0 non funziona

omega_n_max = 1e4;
omega_n_min = 1e3;

% coordinate
Bnd_d_x = [omega_d_min; omega_d_max; omega_d_max; omega_d_min];
Bnd_d_y = [A_d; A_d; -150; -150];

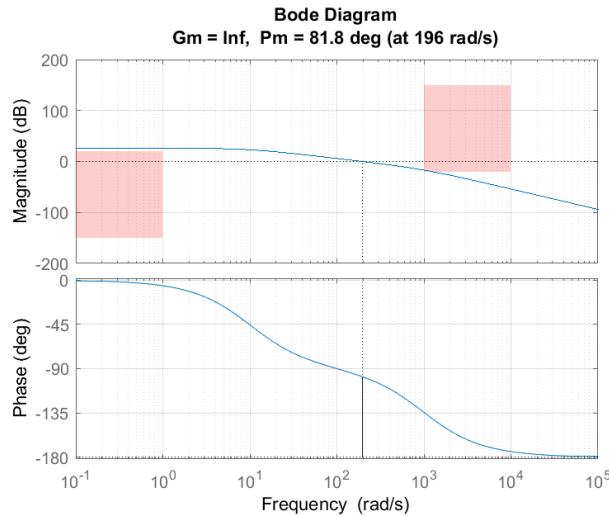
Bnd_n_x = [omega_n_min; omega_n_max; omega_n_max; omega_n_min];
Bnd_n_y = [A_n; A_n; 150; 150];

%% graficare le zone proibite
figure(1); hold on;
patch(Bnd_d_x, Bnd_d_y, 'r', 'FaceAlpha', 0.2, 'EdgeAlpha', 0);
patch(Bnd_n_x, Bnd_n_y, 'r', 'FaceAlpha', 0.2, 'EdgeAlpha', 0);

% dati del sistema
mu = 20;
T1 = 0.1;
T2 = 0.001;
s = tf('s');
GG = mu/((1+T1*s)*(1+T2*s));

margin(GG); grid on;
% MATLAB sovrascrive in automatico il diagramma di bode e le zone proibite

```



## Lezione 13 - sintesi del regolatore

@December 12, 2022

### Regolatore statico

Sappiamo che mettendo in cascata due sistemi, la loro funzione di trasferimento è il prodotto tra le loro due funzioni di trasferimento  $\rightarrow R(s) = R_s(s)R_d(s) \rightarrow R_s(s) = \frac{\mu s}{s^k}$

- se poli in  $s = 0$  (quanti?)
- se "aumentare"  $\mu s$  (quanto?)
- specifiche  $e_\infty$
- specifiche di  $d(t) \rightarrow$  volendo gestirle all'interno di  $R_d(s)$

Ad esempio  $|e_\infty| \leq e^*$  in risposta a gradino su  $w$  e  $d$ , con  $G(s)$  senza poli nell'origine:

per il progetto possiamo scegliere  $R(s) = \mu s \geq \mu^*$  oppure  $R(s) = \frac{\mu s}{s} \rightarrow \underline{\text{attenzione:}}$  poli nell'origine = sfasamento di  $-90^\circ$ !!

### Regolatore dinamico

La progettazione di  $R_d(s)$  mira a:

- imporre  $\omega_c$  in un certo intervallo
- garantire un dato margine di fase  $M_f$