

# Progetto Fondamenti di automatica

Francesco Zumpano matricola:209396  
traccia 117\_2021

## Quesito a.

Si consideri un sistema lineare e stazionario a tempo continuo descritto dalla seguente risposta al segnale  $u(t) = e^{-\frac{1}{2}t}$

$$y(t) = \left( -\frac{16}{9}e^{-\frac{t}{2}} \cdot t - 6 \cdot e^{-t} + \frac{496 \cdot e^{-\frac{t}{2}}}{81} - \frac{52}{81} \cdot \sqrt{2} \cdot e^{-t} \cdot \sin(\sqrt{2} \cdot t) - \frac{10}{81} \cdot e^{-t} \cdot \cos(\sqrt{2} \cdot t) \right) l(t)$$

Determinare:

1. la funzione di trasferimento del sistema ed i suoi poli e zeri;
2. i modi di evoluzione libera del sistema;
3. la risposta all'impulso del sistema;
4. la risposta al gradino ed il suo grafico;
5. la risposta alla rampa;
6. un possibile modello ARMA la cui funzione di trasferimento è quella ottenuta nel primo punto dell'esercizio;
7. tenendo conto del modello determinato al punto precedente valutare la risposta all'ingresso

$$u(t) = -1(-t)$$

Iniziamo il progetto inizializzando il foglio Maple e importando i toolbox che ci serviranno

```
restart :  
with(inttrans) :  
with(DynamicSystems) :
```

1. **Determinare la funzione di trasferimento del sistema ed i suoi poli e zeri;**

Per arrivare alla funzione di trasferimento analizziamo i dati che abbiamo a disposizione: una risposta forzata ed un segnale di un sistema LTI-TC.

La funzione di trasferimento  $G(s)$ , la riposta forzata  $y(t)$  e l'ingresso  $u(t)$  sono legati algebricamente nel dominio della trasformata di Laplace dalla relazione:  $Y(s)=G(s)*U(s)$  dunque  $G(s)=Y(s)/U(s)$ .

Riportiamo i dati su Maple:

$$y_1 := t \rightarrow \left( -\frac{16}{9} e^{-\frac{t}{2}} \cdot t - 6 \cdot e^{-t} + \frac{496 \cdot e^{-\frac{t}{2}}}{81} - \frac{52}{81} \cdot \sqrt{2} \cdot e^{-t} \cdot \sin(\sqrt{2} \cdot t) - \frac{10}{81} \cdot e^{-t} \cdot \cos(\sqrt{2} \cdot t) \right) \cdot \text{Heaviside}(t) :$$

$$u_1 := t \rightarrow e^{-\frac{1}{2}t} \cdot \text{Heaviside}(t) :$$

Per prima cosa spostiamo la risposta forzata e il segnale d'ingresso nel **dominio di Laplace** calcolando le rispettive L-trasformate

$$Y := s \rightarrow \text{laplace}(y_1(t), t, s) :$$

$$Y(s) = \frac{4(2s-1)}{(1+2s)^2(1+s)(s^2+2s+3)}$$

$$U := s \rightarrow \text{laplace}(u_1(t), t, s) :$$

$$U(s) = \frac{2}{1+2s}$$

Ora abbiamo tutto quello che ci serve per ricavare la funzione di trasferimento!

$$G := s \rightarrow \frac{Y(s)}{U(s)} :$$

$$G(s) = \frac{2(2s-1)}{(1+2s)(1+s)(s^2+2s+3)}$$

La nostra f.d.t ci suggerisce che **il sistema è proprio** infatti num. di poli=4 > num. di zeri=1

Calcoliamo ora i poli e gli zeri di G(s)

$$poli := \text{solve}(\text{denom}(G(s)))$$

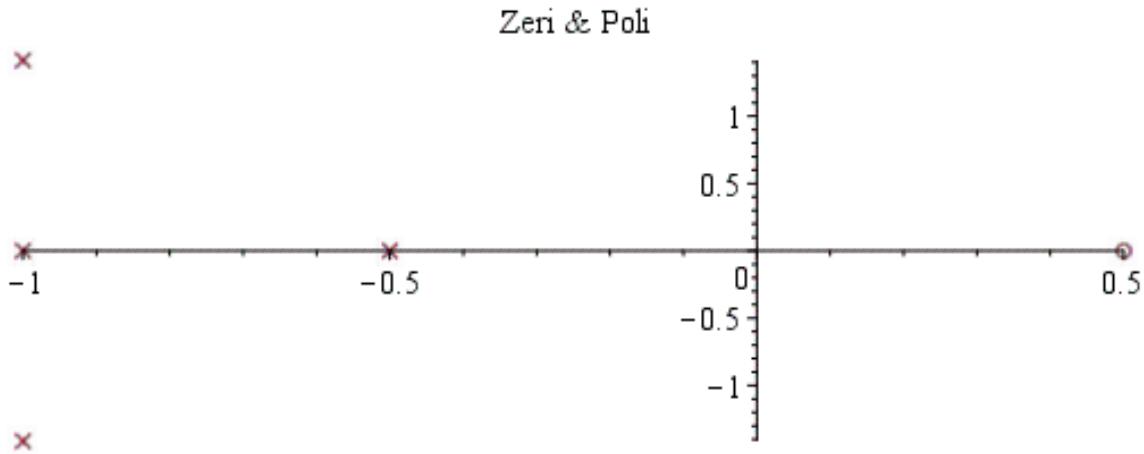
$$poli := -1, -\frac{1}{2}, I\sqrt{2} - 1, -I\sqrt{2} - 1$$

$$zeri := \text{solve}(\text{numer}(G(s)))$$

$$zeri := \frac{1}{2}$$

Vediamo dove si trovano sul piano di Gauss

```
sys := TransferFunction(G(s)) :
ZeroPolePlot(sys, title = "Zeri & Poli")
```



Abbiamo 2 poli reali, 2 poli complessi coniugati e uno zero reale

Notiamo come il sistema sia **BIBO stabile** in quanto i poli si trovano tutti sul semipiano sx.

## 2. Determinare i modi di evoluzione libera del sistema;

Teniamo presente, dalla teoria, che i modi di evoluzione libera in un sistema LTI-TC sono quelle componenti della risposta presenti nella Risposta transitoria, nella Risposta all'impulso e nella Risposta libera e sono della forma

$$\frac{t^k}{k!} e^{p[i]*t}$$

Inoltre, sappiamo che i modi di evoluzione libera sono strettamente legati al denominatore della funzione di trasferimento, infatti, **n.polí = n. modi di evoluzione libera**

Nel nostro caso abbiamo 4 poli  $\rightarrow$  4 modi

I primi due sono poli reali e generano modi esponenziali

$$m_1 := t \rightarrow e^{poli[1]*t} :$$

$$m_2 := t \rightarrow e^{poli[2]*t} :$$

I secondi due sono poli complessi e coniugati e generano due modi pseudo-oscillatori

$$m_3 := t \rightarrow e^{\operatorname{Re}(poli[3])*t} \cdot \cos(\operatorname{Im}(poli[3])*t) :$$

$$m_4 := t \rightarrow e^{\operatorname{Re}(poli[3])*t} \cdot \sin(\operatorname{Im}(poli[3])*t) :$$

$$modi := [m_1(t), m_2(t), m_3(t), m_4(t)]$$

$$modi := \left[ e^{-t}, e^{-\frac{t}{2}}, e^{-t} \cos(\sqrt{2} t), e^{-t} \sin(\sqrt{2} t) \right]$$

Possiamo verificarne la correttezza calcolando la risposta all'impulso (più dettagli nel prossimo punto) della funzione di trasferimento, questa si ottiene antitrasformando la fdt.

$$g := t \rightarrow invlaplace(G(s), s, t) :$$

$$expand(g(t), 'exp')$$

$$-\frac{16 e^{-\frac{t}{2}}}{9} - \frac{11 e^{-t} \cos(\sqrt{2} t)}{9} + 3 e^{-t} + \frac{4 \sqrt{2} e^{-t} \sin(\sqrt{2} t)}{9}$$

Notiamo che escludendo le costanti moltiplicative, ovvero i coefficienti dell'espansione nella risposta all'impulso, saltano subito all'occhio i modi in evoluzione libera calcolati precedentemente

### 3. Determinare la risposta all'impulso del sistema

Per determinare la risposta all'impulso dobbiamo conoscere la FDT e l'ingresso, infatti ricordiamo che la L-Trasformata della risposta all'impulso (che è a tutti gli effetti una risposta forzata) si calcola  $Y_{\text{impulso}}(s) = G(s) * U(s)$ . **U(s) è la L-trasformata dell'impulso ed ha valore unitario  $U(s)=1$** , inoltre l'impulso è un segnale particolare in quanto non rispetta certe proprietà come gli altri segnali, per esempio per  $s \rightarrow \infty$ ,  $U(s)$  non converge a zero. In conclusione,  $Y(s) = G(s)$  (la trasformata di Laplace della risposta all'impulso è la funzione di trasferimento stessa) dunque **otteniamo agilmente la risposta all'impulso antitrasformando la funzione di trasferimento**

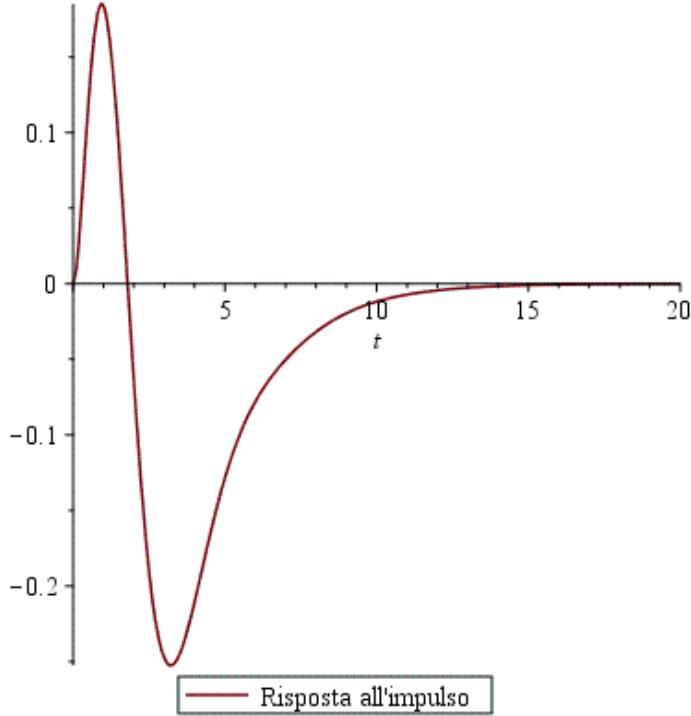
$$g := t \rightarrow invlaplace(G(s), s, t) :$$

$$g(t)$$

$$-\frac{16 e^{-\frac{t}{2}}}{9} + \frac{e^{-t} (4 \sqrt{2} \sin(\sqrt{2} t) - 11 \cos(\sqrt{2} t) + 27)}{9}$$

Per una maggiore chiarezza costruiamo il grafico della risposta impulsiva:

```
plot(g(t), t = 0 ..20, legend = [ "Risposta all'impulso" ])
```



Il sistema viene perturbato inizialmente dall'impulso per poi tornare stabilmente allo stato iniziale (conseguenza BIBO stabilità)

#### 4. Determinare la risposta al gradino ed il suo grafico

Il gradino (in questo caso consideriamo un gradino di ampiezza unitaria), anche **chiamato funzione di Heaviside**, è una funzione che vale  $u(t)=1(t)$  ovvero, esprimendola col costrutto piecewise è del tipo:

$$1(t) = \begin{cases} 1 & t \geq 0 \\ 0 & t < 0 \end{cases}$$

(Questo segnale, data la sua natura, viene spesso utilizzato come un **interruttore** per gli altri segnali)

Per trovare la risposta al gradino (dunque quella particolare risposta forzata che ha come ingresso un gradino) dobbiamo portare l'ingresso nel dominio di Laplace.

$U_{gr} := s \rightarrow \text{laplace}(1, t, s) :$

$$U_{gr}(s)$$

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

quindi calcoliamo la **L-Trasformata della Risposta al gradino**

$$Y_{gr} := s \rightarrow G(s) \cdot U_{gr}(s) :$$

e ora ritorniamo nel dominio del tempo

$$y_{gr} := t \rightarrow \text{invlaplace}(Y_{gr}(s), s, t) :$$

$$y_{gr}(t) \cdot \text{Heaviside}$$

$$\left( -\frac{2}{3} + \frac{32 e^{-\frac{t}{2}}}{9} + \frac{(-5\sqrt{2} \sin(\sqrt{2}t) + \cos(\sqrt{2}t) - 27)e^{-t}}{9} \right) \cdot \text{Heaviside}$$

$$\text{expand}(y_{gr}(t), \text{'exp'})$$

$$-\frac{2}{3} + \frac{32 e^{-\frac{t}{2}}}{9} - \frac{5\sqrt{2} e^{-t} \sin(\sqrt{2}t)}{9} + \frac{e^{-t} \cos(\sqrt{2}t)}{9} - 3e^{-t}$$

Dove **il primo addendo** è un gradino (risposta al gradino  $\rightarrow$  primo addendo gradino) nonché **il valore di regime** della risposta e gli altri sono i modi di evoluzione libera!

Possiamo provare il calcolo, conoscendo la risposta all'impulso, come la quadratura di questa

$$y_{gr2} := t \rightarrow \int_0^t g(\tau) d\tau :$$

$$y_{gr2}(t) = -\frac{2}{3} + \frac{32 e^{-\frac{t}{2}}}{9} - \frac{5\sqrt{2} e^{-t} \sin(\sqrt{2}t)}{9} + \frac{e^{-t} \cos(\sqrt{2}t)}{9} - 3e^{-t}$$

Stesso risultato

possiamo provare anche **il valore di regime** ovvero il valore che si ottiene a transitorio esaurito

$$y_{regime} := t \rightarrow \lim_{t \rightarrow \infty} y_{gr}(t) :$$

che si ottiene anche per  $G(s)$  valutato in 0 ( $G(0)$  guadagno statico) moltiplicato l'ampiezza del gradino,

oppure con il teorema del valore finale

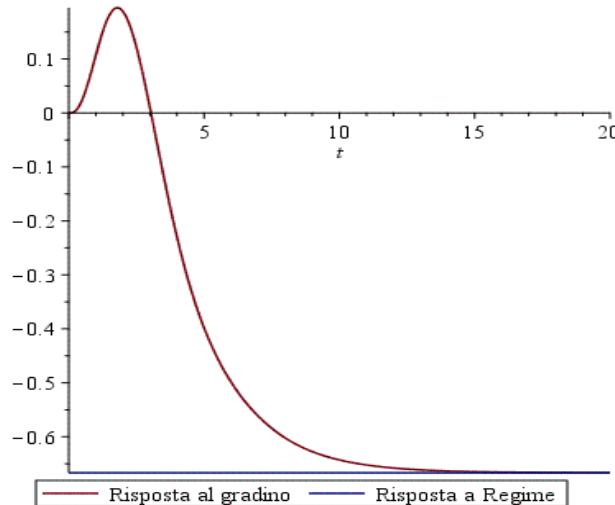
$$y_{regime2} := t \rightarrow \lim_{s \rightarrow 0} s \cdot Y_{gr}(s) :$$

$$RispRegime := \left[ \lim_{s \rightarrow 0} s \cdot Y_{gr}(s), \lim_{t \rightarrow \infty} y_{gr}(t), G(0) \right]$$

$$RispRegime := \left[ -\frac{2}{3}, -\frac{2}{3}, -\frac{2}{3} \right] \Rightarrow \text{come vediamo i tre calcoli portano allo stesso valore}$$

Visualizziamo la risposta al gradino

`plot([ygr(t),yregime2(t)],t=0..20,legend=[ "Risposta al gradino", "Risposta a Regime"])`



Notiamo un fenomeno di **sovraelongazione** (il picco) dovuto ai poli complessi coniugati.

### 5. Determinare la Risposta alla rampa

La rampa (in questo caso di ampiezza unitaria) è una funzione che vale  $u(t)=1*t$  ovvero, esprimendola col costrutto piecewise è del tipo:

$$u(t) = \begin{cases} t & t \geq 0 \\ 0 & t < 0 \end{cases}$$

È un segnale che riconosce altri segnali che hanno derivata costante, infatti è anche chiamato con l'appellativo segnale a deriva costante

Dunque, al solito portiamo l'ingresso nel dominio di Laplace

$$U_{rampa} := s \rightarrow \text{laplace}(t, t, s) :$$

$$U_{rampa}(s)$$

$$\frac{1}{s^2}$$

troviamo la risposta alla rampa

$$y_{rampa} := t \rightarrow \text{invlaplace}\left(\left(G(s) \cdot U_{rampa}(s)\right), s, t\right) :$$

$$y_{rampa}(t)$$

$$\frac{34}{9} - \frac{64 e^{-\frac{t}{2}}}{9} - \frac{2 t}{3} + \frac{e^{-t} (2 \sqrt{2} \sin(\sqrt{2} t) + 3 \cos(\sqrt{2} t) + 27)}{9}$$

Notiamo che i primi due addendi formano la risposta a regime.

Possiamo provare il risultato, conoscendo la risposta al gradino, come la **quadratura** di questa

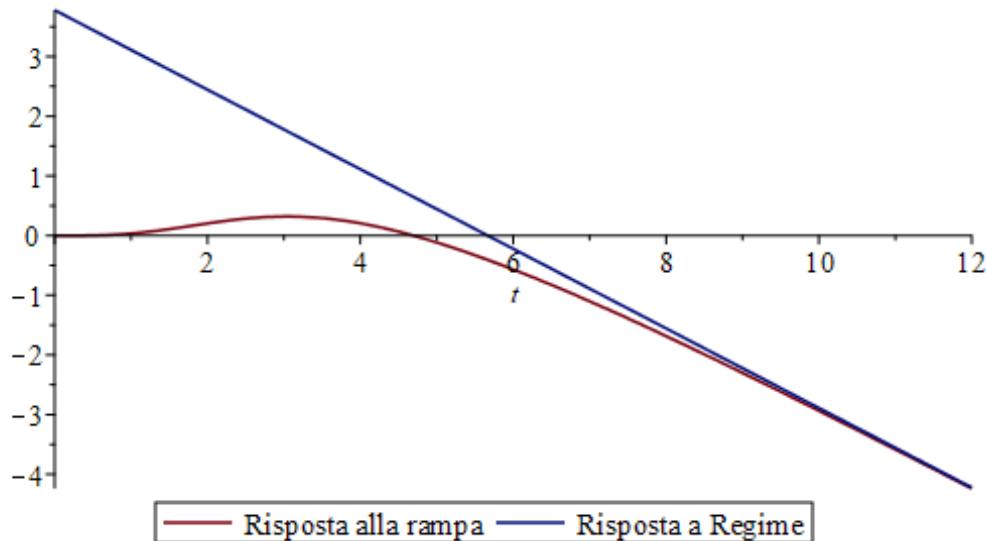
$$y_{rampa2} := t \rightarrow \int_0^t y_{gr}(\tau) d\tau :$$

$$y_{rampa2}(t)$$

$$\frac{34}{9} - \frac{2 t}{3} + \frac{e^{-t} \cos(\sqrt{2} t)}{3} + \frac{2 \sqrt{2} e^{-t} \sin(\sqrt{2} t)}{9} + 3 e^{-t} - \frac{64 e^{-\frac{t}{2}}}{9}$$

Vediamo la risposta su un grafico affiancata dalla risposta a regime

$$\text{plot}\left(\left[y_{rampa}(t), \frac{34}{9} - \frac{2 t}{3}\right], t = 0 .. 12, \text{legend} = ["\text{Risposta alla rampa}", "\text{Risposta a Regime"}]\right)$$



**6. Determinare un possibile modello ARMA la cui funzione di trasferimento è quella ottenuta nel primo punto dell'esercizio**

Possiamo ottenere il modello **ARMA** a partire dalla FDT

$$G(s) = \frac{\text{num}(s)}{\text{denom}(s)} = \frac{Y(s)}{U(s)} \quad \text{quindi } Y(s) * \text{denom}(s) = U(s) * \text{num}(s)$$

Ricordiamo la nostra funzione di trasferimento:

$$G(s) = \frac{2(2s - 1)}{(1 + 2s)(1 + s)(s^2 + 2s + 3)}$$

Eseguiamo le moltiplicazioni al denominatore e al numeratore

$$\begin{aligned} \text{denomG} &:= \text{convert}(\text{denom}(G(s)), \text{parfrac}) \\ \text{denomG} &:= 2s^4 + 7s^3 + 13s^2 + 11s + 3 \end{aligned}$$

$$\begin{aligned} \text{numG} &:= \text{convert}(\text{numer}(G(s)), \text{parfrac}) \\ \text{numG} &:= 4s - 2 \end{aligned}$$

$$Y(s) * (2s^4 + 7s^3 + 13s^2 + 11s + 3) = U(s) * (4s - 2)$$

Eseguendo i calcoli arriviamo a un passo dalla conclusione

$$2Y(s)s^4 + 7Y(s)s^3 + 13Y(s)s^2 + 11Y(s)s + 3Y(s) = 4U(s)s - 2U(s) :$$

$\frac{-----}{y'''(t)}$	$\frac{-----}{y''(t)}$	$\frac{-----}{y'(t)}$	$\frac{-----}{y(t)}$	$\frac{-----}{u'(t)}$	$\frac{-----}{u(t)}$
-------------------------	------------------------	-----------------------	----------------------	-----------------------	----------------------

Ritorniamo nel dominio del tempo (un addendo del tipo  $k^*Y(s)^*s^n$  si traduce, nel dominio del tempo con un addendo del tipo  $k^*y^{(n)}(t)$ , il grado di ‘s’ è l’ordine di derivazione di y)

Seguendo questa logica otteniamo il nostro modello ARMA a partire dalla funzione di trasferimento.

$$eq := 2 \cdot y_{ar}'''(t) + 7 \cdot y_{ar}''(t) + 13 \cdot y_{ar}'(t) + 11 \cdot y_{ar}(t) \\ + 3 y_{ar}(t) = 4 \cdot u_{ar}'(t) - 2 \cdot u_{ar}(t)$$

$$eq := 2 D^{(4)}(y_{ar})(t) + 7 D^{(3)}(y_{ar})(t) + 13 D^{(2)}(y_{ar})(t) \\ + 11 D(y_{ar})(t) + 3 y_{ar}(t) = 4 D(u_{ar})(t) - 2 u_{ar}(t)$$

-ar sta per arma

Per comodità, mi calcolo la risposta totale del sistema in quanto ci servirà per il prossimo punto  
 $eq\_laplace := laplace(eq, t, s) :$

$$u_{ar}(0) := 0 :$$

$$eq\_sub := subs(\{laplace(y_{ar}(t), t, s) = Y_{ar}(s), laplace(u_{ar}(t), t, s) \\ = U_{ar}(s)\}, eq\_laplace) :$$

$$eq\_solved := solve(eq\_sub, Y_{ar}(s)) :$$

$$risposta := collect(eq\_solved, U_{ar}(s))$$

$$risposta := \frac{(4s - 2) U_{ar}(s)}{2s^4 + 7s^3 + 13s^2 + 11s + 3} \\ + \frac{1}{2s^4 + 7s^3 + 13s^2 + 11s + 3} (2s^3 y_{ar}(0) + 2s^2 D(y_{ar})(0) \\ + 7s^2 y_{ar}(0) + 2s D^{(2)}(y_{ar})(0) + 7s D(y_{ar})(0) + 13s y_{ar}(0) \\ + 2 D^{(3)}(y_{ar})(0) + 7 D^{(2)}(y_{ar})(0) + 13 D(y_{ar})(0) + 11 y_{ar}(0))$$

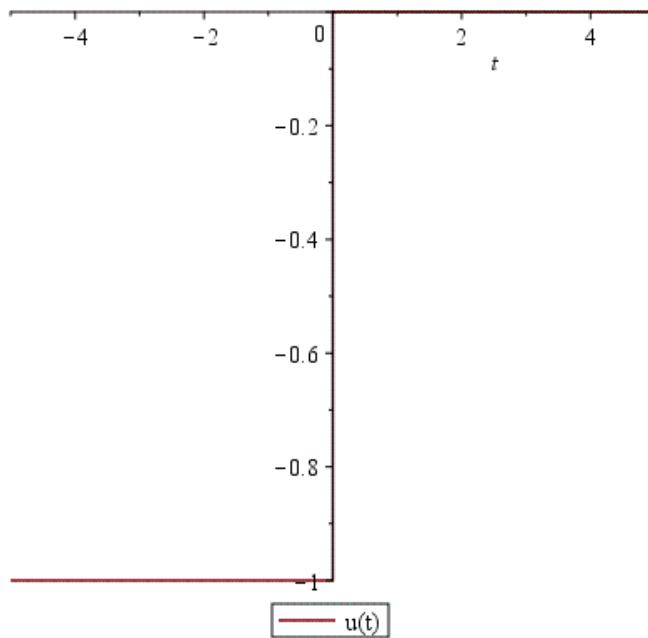
Risposta = Risposta forzata + Risposta libera

### 7. Determinare la risposta all'ingresso $u(t) = -1(-t)$

L'ingresso in questione è un segnale **left-sided** che vale -1 sul semiasse negativo e 0 sul positivo, vediamolo dunque con il costrutto piecewise poi sul piano cartesiano:

$$u_{segna} := t \rightarrow \begin{cases} -1 & t < 0 \\ 0 & t \geq 0 \end{cases} :$$

`plot(u_segna(t), t = -5 .. 5, legend = "u(t)")`



Ora iniziamo a costruire l'uscita  $y(t)$  ricordando che **vale per ogni  $t$**  quindi da  $-\infty$  a  $+\infty$

Possiamo vedere l'uscita come una funzione piecewise quindi come una **composizione di due uscite**, una per  $t$  negativo una per  $t$  positivo

$$y_{segna} := t \rightarrow \begin{cases} y_-(t) & t < 0 \\ y_+(t) & t \geq 0 \end{cases} :$$

## **t < 0:**

Essendo il sistema Bibo stabile, l'ingresso  $-1(-t)$  è applicato in un istante di tempo nel passato remoto rispetto a  $t$ , quindi per  $t < 0$  l'uscita conterrà solo le **componenti della risposta a regime**. In questo caso, avendo un ingresso di tipo gradino di ampiezza  $-1$ , l'uscita per  $t < 0$  è il guadagno statico ( $G(0)$ ) moltiplicato per l'ampiezza del gradino.

$$y_- := t \rightarrow G(0) \cdot (-1) :$$

Quindi per  $t$  negativo l'uscita vale

$$y_-(t)$$

$$\frac{2}{3}$$

## **t > 0:**

Per  $t > 0$  non ho nessuno ingresso quindi  $u(t) = 0$  ne consegue che  $y_+(t>0)$  è la **risposta in evoluzione libera!** (la risposta non influenzata da ingressi)

Per trovarla ci servono le condizioni iniziali, in numero pari al n.pol = grado del sistema = 4:

La prima condizione iniziale è proprio la risposta negativa per la **continuità** che c'è tra le **condizioni iniziali** ( $y(0^-) = y(0^+)$ )

$$y_{ar}(0) := y_-(0)$$

$$y_{ar}(0) := \frac{2}{3}$$

Definiamo le condizioni delle derivate

$$D(y_{ar})(0) := eval\left(\frac{d}{dt}(y_-(t)), t=0\right)$$

$$D(y_{ar})(0) := 0$$

$$D^{(2)}(y_{ar})(0) := eval\left(\frac{d^2}{dt^2}(y_-(t)), t=0\right); D^{(3)}(y_{ar})(0) := eval\left(\frac{d^3}{dt^3}(y_-(t)), t=0\right);$$

$$D^{(2)}(y_{ar})(0) := 0$$

$$D^{(3)}(y_{ar})(0) := 0$$

Troviamoci la risposta libera secondo queste condizioni (Maple lo calcola in automatico basta solo estrarre la risposta libera dalla risposta complessiva calcolata alla fine del punto precedente)

$$Y_{libera} := op(2, risposta)$$

$$Y_{libera} := \frac{\frac{4}{3}s^3 + \frac{22}{3} + \frac{14}{3}s^2 + \frac{26}{3}s}{2s^4 + 7s^3 + 13s^2 + 11s + 3}$$

Ritornando nel dominio del tempo antitrasformando la risposta libera otterremo la risposta al segnale dato per  $t > 0$

$$y_+ := t \rightarrow invlaplace(Y_{libera}, s, t) :$$

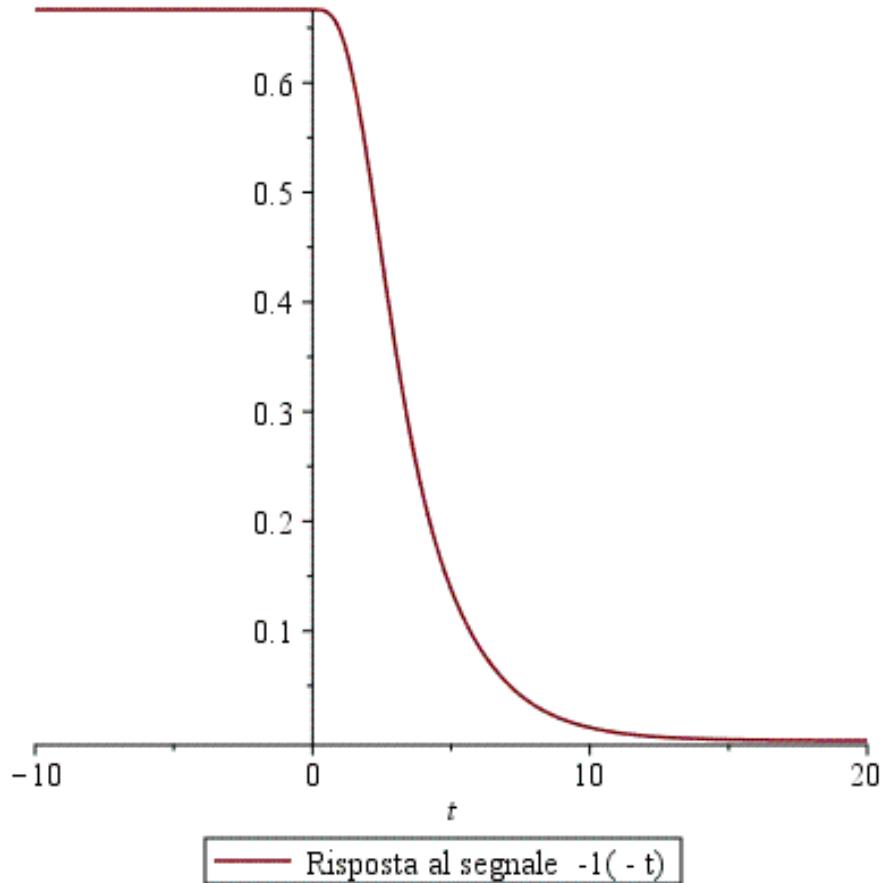
$$y_+(t)$$

$$\frac{16 e^{-\frac{t}{2}}}{9} - \frac{e^{-t} (\sqrt{2} \sin(\sqrt{2} t) + \cos(\sqrt{2} t) + 9)}{9}$$

Riscriviamo complessivamente la risposta

$$y_{segna}e(t) = \begin{cases} \frac{2}{3} & t < 0 \\ \frac{16 e^{-\frac{t}{2}} - e^{-t} (\sqrt{2} \sin(\sqrt{2} t) + \cos(\sqrt{2} t) + 9)}{9} & 0 \leq t \end{cases}$$

`plot(ysegnae(t), t=-10..20, legend = [ "Risposta al segnale -1(-t)" ])`



Per  $t$  negativo il livello è costante ed è la risposta a regime per  $t$  positivo abbiamo la risposta libera che converge a 0 in quanto il sistema è BIBO stabile.

## Quesito b.

Disegnare e discutere il Diagramma di Bode per la seguente funzione di trasferimento

$$G(s) = \frac{9(s + 4)}{40s^4(s^2 + 2s + 9)}$$

Introduciamo il discorso ricordando la definizione di risposta in frequenza: la risposta in frequenza è una particolare risposta forzata di un sistema LTI-TC che ha come ingresso un segnale periodico, **viene indicata con  $G(j\omega)$**  ed è anche fondamentale per il calcolo del fattore di distorsione della risposta a regime.

Il diagramma di Bode è un modo per **figurare la risposta in frequenza**. È costituito da due diagrammi cartesiani: il diagramma dei **moduli** ( $\omega, |G(j\omega)|$ ) e diagramma delle **fasi** ( $\omega, \angle G(j\omega)$ ).

- Nel diagramma dei moduli rappresentiamo le pulsazioni, sulle ascisse, in **scala logaritmica** (a decadi), mentre il modulo, sulle ordinate, in **decibel**  $|G(j\omega)|\text{dB} = 20 \log(|G(j\omega)|)$ , dunque c'è
- Nel diagramma delle fasi rappresentiamo le pulsazioni a decadi e gli angoli sulle ordinate in gradi (radiani o sessagesimali)

OSS: le fasi sono di natura in scala logaritmica

Queste assunzioni ci permettono di disegnare in maniere esaustiva e sintetica sia le basse che le alte frequenze (evitiamo il problema della compressione delle alte frequenze e di espansione delle basse frequenze).

*restart :*

```
with(DynamicSystems) :  
with(plots) :
```

- Costruiamo il sistema su cui basarci per la costruzione del diagramma di Bode:

$$G := s \rightarrow \frac{9(s + 4)}{40 \cdot s^2 \cdot (s^2 + 2s + 9)}$$

Troviamo gli elementi che costituiscono la funzione di trasferimento in **forma di Bode**:

### 1. Analizzare i poli e gli zeri della funzione

```
zG := solve(numer(G(s)))
```

*zG := -4*

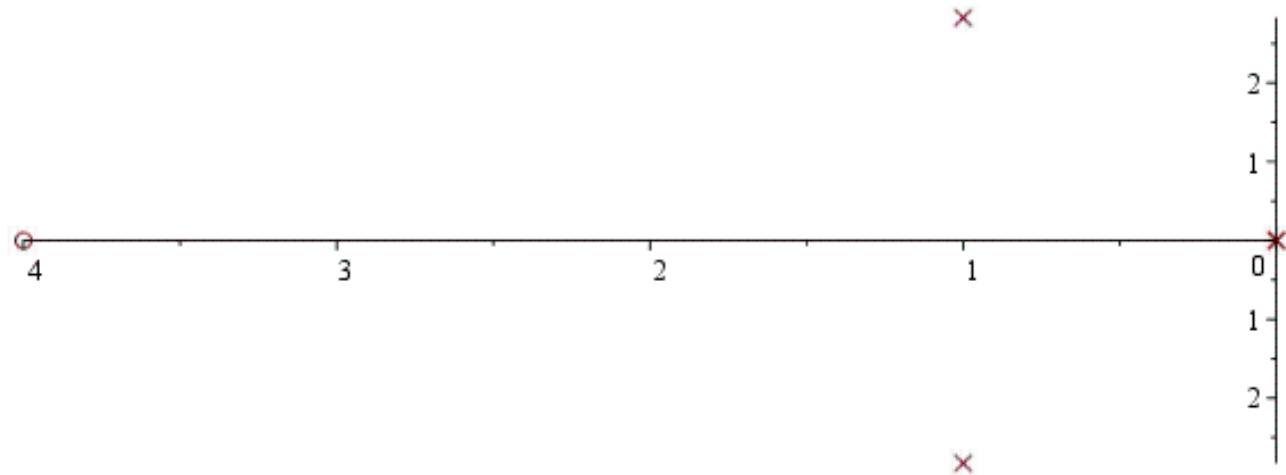
```

pG := [solve(denom(G(s)))]
      pG := [0, 0, -1 + 2 I √2, -1 - 2 I √2]

```

Vediamoli sul piano di Gauss per un'interpretazione più agevole

```
ZeroPolePlot(TransferFunction(G(s)))
```



- Abbiamo un solo zero (- 4) a **fase minima** in quanto minore di zero e quattro poli di cui due nell'origine, generati dal fattore integrale di ordine 2, e due complessi e coniugati **stabili** (parte reale negativa), generati dal trinomio non scomponibile
- Particolare attenzione va riservata al trinomio al denominatore, come abbiamo visto è un **trinomio non scomponibile** nel campo dei numeri reali (delta negativo) quindi le radici sono complesse e coniugate e hanno parte reale negativa quindi possiamo scriverlo sotto la forma di trinomio campione  $s^2 + 2\delta\omega_n s + \omega_n^2$  con  $\omega_n$  la pulsazione naturale e  $\delta$  lo smorzamento queste due quantità si ricavano risolvendo il sistema

$$\begin{cases} 2\delta\omega_n = a_1 \\ \omega_n^2 = a_2 \end{cases}$$

che su maple risolviamo così:

```

sol := solve( { 2·δ·ω_n^2 = 2, ω_n^2 = 9, ω_n > 0 }, { δ, ω_n } )
      sol := { δ = 1/3, ω_n = 3 }

```

$$\delta := \frac{1}{3} :$$

$$\omega_n := 3 :$$

Notiamo che lo smorzamento  $\delta$  è un numero compreso tra 0 e  $\frac{1}{\sqrt{2}}$  questo vuol dire che i nostri poli complessi e coniugati sono a **basso smorzamento**. Ciò significa che avremo una pulsazione di risonanza  $\omega_r$  nell'intorno della pulsazione naturale  $\omega_n$  dove vi troveremo un picco, che renderà il diagramma dei moduli non più monotono decrescente.

Troviamolo con la relazione tra pulsazione e picco di risonanza:

$$\omega_r := evalf\left(\omega_n \cdot \sqrt{1 - 2 \cdot \delta^2}\right);$$

$$\omega_r = 2.645751311$$

$$M_r := evalf\left(\frac{1}{2 \cdot \delta \cdot \sqrt{1 - \delta^2}}\right);$$

$$M_r = 1.590990257$$

quindi in corrispondenza della pulsazione  $\omega_r$  avremo un picco dal valore  $\sim 1.6$

teniamoci in una variabile  $\mu$  il numero di occorrenze dei poli nell'origine

$$\mu := numoccur(pG, 0)$$

$$\mu := 2$$

## 2. Calcoliamo il guadagno di Bode

$$k_b := \lim_{s \rightarrow 0} s^2 \cdot G(s)$$

$$k_b := \frac{1}{10}$$

Indichiamo con  $\Omega$  la frequenza di taglio dello zero

$$\Omega := |zG|$$

$$\Omega := 4$$

**La forma di Bode è la seguente**

$$G_b := s \rightarrow k_b \cdot \frac{\left(1 + \frac{sign(-zG) \cdot s}{\Omega}\right)}{s^2 \cdot \left(1 + \frac{2 \cdot \delta}{\omega_n} \cdot s + \frac{s^2}{\omega_n}\right)}$$

$$G_b(s)$$

$$\frac{1 + \frac{s}{4}}{10 s^2 \left(1 + \frac{2}{9} s + \frac{1}{3} s^2\right)}$$

## Diagramma dei moduli

Per tracciare il diagramma analizziamo tutti i contributi delle varie componenti:

- Iniziamo con il guadagno di Bode in decibel:

$$k_{b, dB} := \text{evalf}(20 \cdot \log10(|k_b|))$$

$$k_{b, dB} := -20.$$

ha un valore negativo, quindi ci sarà un fenomeno di attenuazione del modulo, ovvero quest'ultimo subirà uno slittamento di 20 decibel verso sinistra

- Vediamo i poli nell'origine, sono due quindi danno un contributo sulla pendenza di **-40 dB/dec** per tutto il diagramma (-20 \* n.polì nell'origine)
- Lo zero reale (valore -4) è a fase minima: in bassa frequenza non dà alcun apporto al diagramma mentre in alta frequenza, a partire dalla pulsazione di taglio  $\Omega$ , amplifica il modulo di **+20 dB/dec** (portando la retta a una pendenza di -20db/dec).

Definiamo la funzione che calcola precisamente la pendenza degli zeri/poli reali:

$$f_{modRe} := (\omega, \omega_c) \rightarrow 20 \cdot \log10 \left( \sqrt{1 + \left( \frac{\omega}{\omega_c} \right)^2} \right):$$

- La coppia di poli c.c: in bassa frequenza non da nessun contributo, in alta frequenza aggiunge, a partire dalla pulsazione  $\omega_n$ , una pendenza di **-40 db/dec**(portando la pendenza totale a -60 dB/dec).

Definiamo la funzione che calcola precisamente la pendenza di poli/zeri complessi e coniugati

$$f_{modCC} := (\omega, \omega_n, \delta) \rightarrow 20 \cdot \log10 \left( \left| \left( 1 - \frac{\omega^2}{\omega_n^2} \right) + 2 \cdot I \cdot \delta \cdot \frac{\omega}{\omega_n} \right| \right):$$

Possiamo finalmente definire la **funzione modulo**:

$$modulo := \omega \rightarrow k_{b,dB} - \mu \cdot 20 \cdot \log 10(\omega) + f_{modRe}(\omega, \Omega) - f_{modCC}(\omega, \omega_n, \delta) :$$

Mettiamola a confronto con la sua approssimazione **asintotica**:

- Definiamo la funzione che approssima i moduli dei poli/zeri reali

$$f_{modAsintotica} := (\omega, \omega_c) \rightarrow \begin{cases} 0 & \frac{\omega}{\omega_c} < 1 \\ 20 \cdot \log 10 \left( \frac{\omega}{\omega_c} \right) & \frac{\omega}{\omega_c} \geq 1 \end{cases} :$$

- La funzione complessiva modulo approssimato sarà:

$$modulo_{Asintotico} := \omega \rightarrow k_{b,dB} - \mu \cdot 20 \cdot \log 10(\omega) + f_{modAsintotica}(\omega, \Omega) - f_{modCC}(\omega, \omega_n, \delta) :$$

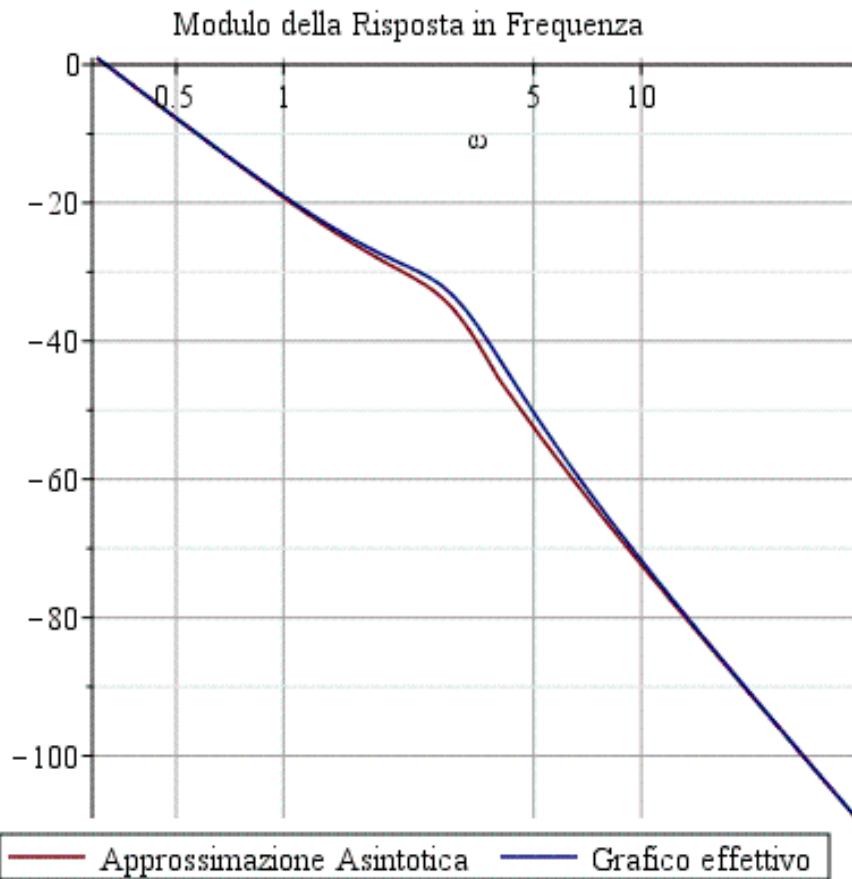
- Determiniamo la pulsazione di taglio più a bassa frequenza ( $\omega_{sx}$ ) e la più ad alta frequenza sui moduli ( $\omega_{dx}$ )

$$\omega_{sx} := \min(\omega_n, \Omega) :$$

$$\omega_{dx} := \max(\omega_n, \Omega) :$$

Vediamo le due funzioni su un grafico:

```
semilogplot([modulo_Asintotico(ω), modulo(ω)], ω = 0.1·ωsx..10·ωdx, axis = [gridlines], title = "Modulo della Risposta in Frequenza", legend = ["Approssimazione Asintotica", "Grafico effettivo"])
```



Nell'approssimazione asintotica il picco di risonanza è meno evidente

Recap:

- in BF ho una pendenza di  $-20 \cdot (\text{n.pol} \text{ origine}) = -40 \text{ dB/dec}$
- in AF ho una pendenza di  $20 \cdot (-\text{n.pol}_0 + \text{n.zeri} - \text{n.pol}) = 20 \cdot (-2 + 1 - 2) = -60 \text{ dB/dec}$

## Diagramma delle fasi

Per tracciare il diagramma analizziamo tutti i contributi delle varie componenti:

- Iniziamo con il guadagno di Bode in decibel:

$$k_{b, dB} := \text{evalf}\left(20 \cdot \log10(|k_b|)\right)$$

$$k_{b, dB} := -20.$$

ha un valore negativo, quindi seguendo la formula  $\frac{(1 - \text{sign}(k_b))}{2} \cdot \pi$  non aggiunge niente alla fase

- Vediamo i poli nell'origine, sono due quindi danno un contributo sulla fase di  $-180^\circ$
- Lo zero reale (-4) è a fase minima: in bassa frequenza non dà alcun apporto, in alta frequenza, a partire dalla pulsazione di taglio  $\Omega$  anticipa la fase di  $90^\circ$  ( $+90^\circ$ )

Definiamo la funzione che calcola precisamente la fase di zeri/poli reali:

$$g_{faseRe} := (\omega, \omega_c) \rightarrow \arg\left(1 + I \cdot \left(\frac{\omega}{\omega_c}\right)\right) :$$

- La coppia di poli c.c: in bassa frequenza non da nessun contributo, in alta frequenza aggiunge, a partire dalla pulsazione  $\omega_n$ , alla fase  $-180^\circ$

Definiamo la funzione che calcola precisamente la fase di zeri/poli CC

$$g_{faseCC} := (\omega, \omega_n, \delta) \rightarrow \text{argument}\left(\left(1 - \frac{\omega^2}{\omega_n^2}\right) + 2 \cdot I \cdot \delta \cdot \frac{\omega}{\omega_n}\right) :$$

Definiamo la funzione fase complessiva:

$$fase := \omega \rightarrow \frac{(1 - \text{sign}(k_b))}{2} \cdot \pi - \frac{\mu \cdot \pi}{2} + \text{sign}(-zG) \cdot g_{faseRe}(\omega, \Omega) - \text{sign}(\delta) \cdot g_{faseCC}(\omega, \omega_n, \delta) :$$

Mettiamola a confronto con la sua approssimazione asintotica:

- Definiamo la funzione che approssima le fasi

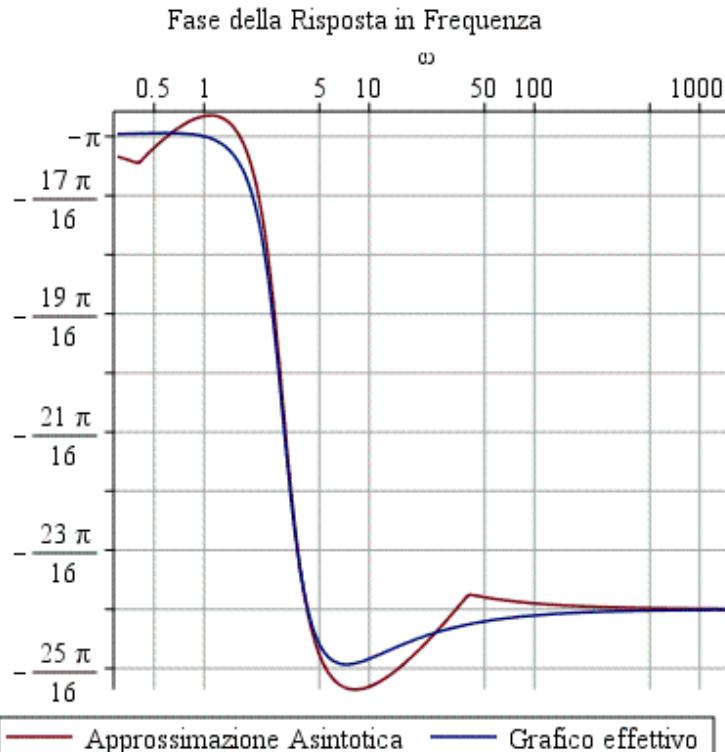
$$g_{faseAsintotica} := (\omega, \omega_c) \rightarrow \begin{cases} 0 & \frac{\omega}{\omega_c} < \frac{1}{10} \\ \frac{\pi}{4} \cdot \log_{10} \left( \frac{\omega}{\omega_c} \right) & \frac{1}{10} \leq \frac{\omega}{\omega_c} < 10 \\ \frac{\pi}{2} & \frac{\omega}{\omega_c} \geq 10 \end{cases} :$$

- La funzione complessiva fase approssimata sarà:

$$fase_{Asintotica} := \omega \rightarrow \frac{(1 - sign(k_b))}{2} \cdot \pi - \frac{\mu \cdot \pi}{2} + sign(-zG) \cdot g_{faseAsintotica}(\omega, \Omega) - sign(\delta) \cdot g_{faseCC}(\omega, \omega_n, \delta) :$$

Vediamo le due funzioni su un grafico:

```
semilogplot([faseAsintotica(omega), fase(omega)], omega=0.1*omega_sx..400*omega_dx, axis=[gridlines], tickmarks=[default, piticks], title="Fase della Risposta in Frequenza", legend=["Approssimazione Asintotica", "Grafico effettivo"])
```

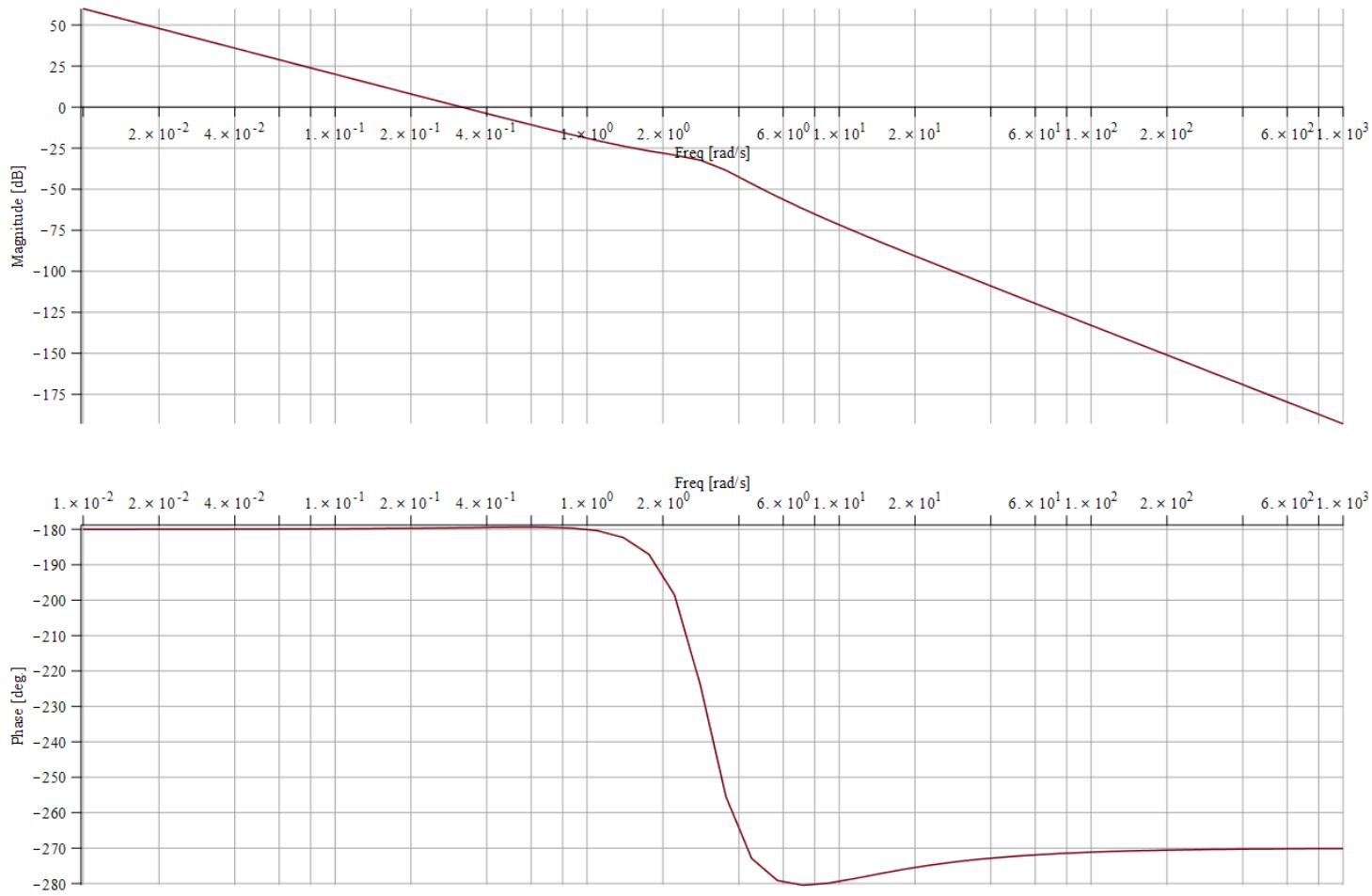


Recap:

- in BF la fase è di  $\frac{(1 - \text{sign}(k_b))}{2} \cdot \pi - \frac{2 \cdot \pi}{2} : = -\pi = -180^\circ$
- in AF la fase è di  $\frac{(1 - \text{sign}(k_b))}{2} \cdot \pi - \frac{2 \cdot \pi}{2} + \frac{1 \cdot \pi}{2} - \frac{2 \cdot \pi}{2} : = -3/2 \pi = -270^\circ$

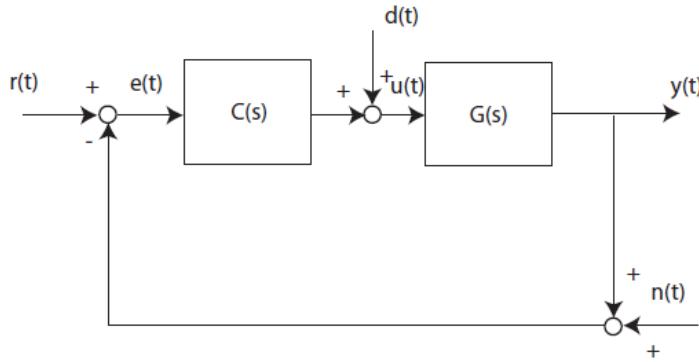
Confrontiamo i grafici che abbiamo ottenuto con i grafici che Maple calcola in automatico e vediamo che sono pressoché coincidenti

*BodePlot(TransferFunction(G(s)))*



### Quesito c.

Si consideri il seguente schema di controllo in retroazione algebrica ed unitaria.



dove  $G(s)$  rappresenta la f.d.t. del processo

$$G(s) = \frac{s + 1}{s(s + \frac{1}{3})^2}$$

Si chiede di determinare un regolatore  $C(s)$  di struttura semplice che garantisca il soddisfacimento delle seguenti specifiche:

1. errore di inseguimento alla rampa inferiore al 25 %;
2. massima sovraelongazione  $S\% \leq 25\%$ , tempo di assestamento  $t_s \leq 70$  sec.

Abbiamo uno schema di controllo in retroazione algebrica e unitaria quindi abbiamo un **problema di sintesi**. Il dato che abbiamo è  $G(s)$  ovvero la funzione di trasferimento del sistema che modella il nostro impianto.

Il quesito in esame ci chiede di determinare un regolatore  $C(s)$  da mettere in serie con la funzione di trasferimento in modo che la nuova funzione  $L(s) = C(s)*G(s)$  soddisfi due specifiche:

- errore di inseguimento alla rampa (anche detto **errore di velocità**) inferiore al 25% (specifiche che riguarda la cosiddetta **precisione statica**);
- massima **sovraelongazione**  $S\% \leq 25\%$  e **tempo di assestamento**  $t_{s,5\%} \leq 70$  secondi (specifiche che riguarda la **precisione dinamica**)

Ricordiamo che il regolatore  $C(s)$  deve garantire il **principio di parsimonia** ovvero deve aver il minimo numero di elementi dinamici con i quali soddisfare le specifiche.

Iniziamo analizzando la funzione di trasferimento, essa presenta uno zero a fase minima (-1), un polo stabile (-1/3) e **un polo nell'origine**. Di questi elementi quello che ci dà più informazioni per questo tipo di esercizi è il polo nell'origine, infatti questo ci suggerisce che il nostro sistema ha "di serie" un effetto integrale e che quindi **il sistema è di tipo 1**.

Vediamo la prima specifica, un errore di inseguimento, a fronte di un riferimento a rampa, inferiore al 25% (dunque avendo una percentuale il coefficiente angolare della rampa è indifferente quindi scegliamo una **rampa unitaria**).

Iniziamo a risolvere il quesito:

- dobbiamo trovare un errore di velocità FINITO quindi, dalla precisione statica ricordo che, la funzione di anello  $L(s)=C(s)*G(s)$  deve essere di tipo 1. Quindi a meno che non ho già un effetto integrale lo devo aggiungere nel regolatore, in questo caso sono fortunato e **l'integratore è già presente nella FDT** quindi  $C(s)$  sarà una costante  $k$ . ✓
- Troviamo il valore di  $C(s)$  che tara l'errore di velocità:
  - L'errore di inseguimento inferiore del 25% si calcola come:

$$e_{v,\infty} = \lim_{s \rightarrow 0} \frac{1}{s \cdot L(s)} \leq \frac{25}{100} = \frac{1}{4}$$

$$\lim_{s \rightarrow 0} \frac{1}{s \cdot k \cdot \frac{(s+1)}{s \cdot \left(s + \frac{1}{3}\right)^2}} = \frac{1}{9 \cdot k} \leq \frac{1}{4}$$

$$9k \geq 4 \quad k \geq \frac{4}{9} \rightarrow k \geq 0.444$$

**Scelgo  $k=0.5$**

- Il controllore  $C(s) = 0.5$  ✓
- Il primo punto ce lo siamo calcolati agilmente.

- Riportiamo i risultati ottenuti su Matlab in un nuovo script:

```

C:\Users\chicco\Desktop\Uni\2 anno\2 semestre\Fondamenti di automatica\Matlab ex\Reti_Correttive\progettino3eserc.m

EDITOR PUBLISH VIEW
1 % Inizializziamo il nuovo script pulendo le eventuali variabili già
2 % caricate su matlab e chiudendo le eventuali finestre aperte da un altro
3 % script
4
5 - clear; close all;
6
7 % Usiamo il comando zpk('s') per far sì che la lettera 's' diventi una
8 % variabile speciale in modo da poter esprimere il nostro sistema T-C in
9 % modo semplice e più intuitivo rispetto a quello di specificare i
10 % coefficienti polinomiali
11 %
12 - s=zpk('s');
13
14 - G=(s+1)/(s*(s+1/3)^2);
15
16 % Scriviamo il controllore appena calcolato
17
18 - C_s=0.5; % come detto ho già un polo nell'origine nella FDT e
19 % non devo aggiungere un integratore
20

```

La seconda specifica invece ci chiede di tarare la massima sovraelognazione(che indichiamo con **S**). affinché sia al di sotto del 25% e contemporaneamente il tempo di assestamento minore di 70 sec.

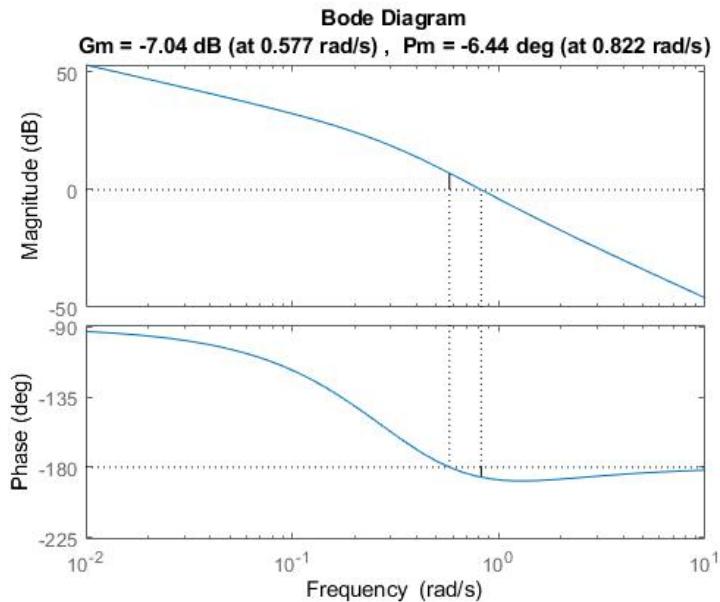
La seconda specifica è quella più ostica, andiamo per gradi:

- Vediamo innanzitutto con solo l'aggiunta del controllore quanto vale il margine di fase e la pulsazione di attraversamento della funzione di anello

```

25 - L_old=series(C_s,G);
26 -
27 - margin(L_old);

```



Come possiamo vedere abbiamo un margine di fase negativo, dunque salta la condizione di BIBO stabilità, non disperiamoci perché ancora l'esercizio non è finito.

- Un modo per risolvere questo punto è quello di **convertire le specifiche dinamiche in requisiti approssimati legati alla pulsazione di attraversamento e al margine di fase** sulla funzione di anello
- Vediamo come legare la massima sovraelongazione col margine di fase:
  - Ipotizziamo che la FDT del sistema retroazionato approssimato sia del tipo:

$$T(s) = \frac{\omega_n^2}{s^2 + 2\delta\omega_n s + \omega_n^2}$$

(Dove il guadagno statico di questa fdt è 1 in quanto il sistema è di tipo 1)

- Quindi avrò che la massima sovraelongazione ( $S$ ) è legata allo smorzamento dei poli complessi e coniugati secondo questa relazione:

$$S = e^{-\frac{\pi\delta}{\sqrt{1-\delta^2}}}$$

- Io devo trovare quello smorzamento critico  $\delta_{cr}$  in corrispondenza del quale ottengo la massima sovraelongazione del 25%

$$e^{-\frac{\pi \delta_{cr}}{\sqrt{1-\delta_{cr}^2}}} = 0.25$$

- Inoltre, so che **S e  $\delta$  sono tra di loro inversamente proporzionali** quindi, condizione necessaria, per far sì che la massima sovraelongazione sia minore del 25% è quella di scegliere un  $\delta > \delta_{cr}$ . Sfruttiamo la funzione, messa a disposizione dal professore su matlab, *smorz\_S* per ricavare lo smorzamento critico a partire dalla massima sovraelongazione.

Nel nostro caso il delta critico vale 0.4037

```
>> delta_cr=smorz_S(0.25)
```

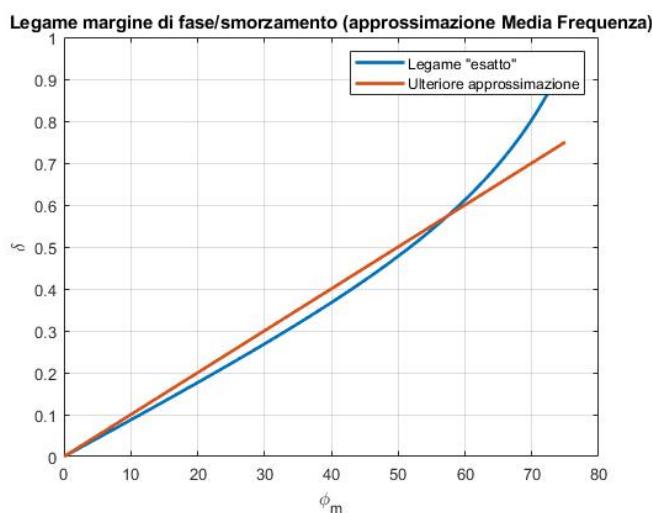
```
delta_cr =
```

```
0.4037
```

- Ora sfruttiamo un ulteriore approssimazione per ricavare dallo smorzamento il margine di fase  $\Phi_m$

$$\phi_m = 100 \cdot \delta$$

Questa approssimazione non è valida sempre ma possiamo prenderla per buona fino a un margine di fase di circa 70° come possiamo vedere dal grafico:



(Speriamo di non dover andar oltre ai 65-70 gradi per risolvere il quesito.)

Quindi al nostro smorzamento critico corrisponde un margine di fase critico di 40.37°.

Questo implica che un **margine di fase maggiore di 40.37° mi darà una massima sovraelongazione sotto il 25%** scelgo un margine di fase di 50° (spoiler: meno di questo non mi basta per abbassare S)

- Adesso vediamo come legare tempo di assestamento con la pulsazione di attraversamento  $\omega_c$ 
  - sfruttiamo il legame tra tempo di assestamento e pulsazione naturale:

$$t_{s, 5\%} = \frac{3}{\delta \omega_n}$$

- Mi ricavo la pulsazione naturale critica dallo smorzamento critico

$$\omega_{n, cr} = \frac{3}{t_{s, 5\%} \cdot \delta_{cr}} = \frac{3}{70 \cdot 0.4037} = 0.1061$$

- Calcolo la pulsazione di banda passante servandomi della funzione su matlab `wBwn` messa a disposizione dal professore che sfrutta la relazione tra banda passante e pulsazione naturale

```
>> wn_cr=3/(70*delta_cr);
>> w_Bw=wBwn(delta_cr)*wn_cr
```

`w_Bw =`

**0.1456**

- Ora so che una  $\omega_c$  è un **minorante di  $\omega_{bw}$**  quindi prendendo una pulsazione di attraversamento maggiore della pulsazione della banda passante fa sì che il nostro tempo di assestamento sia minore di 70 sec

- Quindi ricapitolando, devo scegliere un **margine di fase maggiore di 40.37°** e una **pulsazione di attraversamento maggiore di 0.1456**

- Scelgo una pulsazione di attraversamento pari a  $\omega_{c,new} = 1.5$  (spoiler: la prendo abbastanza più grande di  $\omega_{bw}$  in quanto facendo dei test con valori più vicini a 0.1456, taup usciva negativa costringendomi a fare due reti anticipatrici)
- Scelgo un margine di fase obiettivo pari a 50° (spoiler: più piccolo di così non basta per ridurre S sotto il 25%)

Una volta convertito il problema da massima sovraelongazione e tempo di assestamento in un problema di margine di fase e pulsazione di attraversamento andiamo avanti.

Troviamoci su matlab il modulo e l'argomento valutato in corrispondenza di  $\omega_{c,new} = 1.5$  della funzione di anello soltanto con il controllore per la precisione statica

```
wc_new=1.5;

[modulo,argomento]=bode(L_old,wc_new);

>> modulo      >> argomento

modulo =          argomento =

0.2545          -188.6325
```

Come possiamo notare il **modulo è minore dell'unità** (quindi **devo amplificare**) e la distanza goneometrica

`distanza=180-abs(argomento);` = - 8.6325° è **minore del margine di fase obiettivo di 50° (devo anticipare)**, dunque, dobbiamo aggiungere **in serie a C(s) una rete anticipatrice** tale che

$$\begin{cases} |L(j\omega_c)| = 1 \\ 180 - |\angle L(j\omega_c)| = \phi_m \end{cases}$$

Costruiamo la rete anticipatrice:

- Innanzitutto, ricordiamo la forma della rete anticipatrice:

$$\frac{1 + s\tau_z}{1 + s\tau_p} \quad \tau_z > \tau_p > 0$$

- Troviamoci  $\tau_z$  e  $\tau_p$  con la funzione matlab messa a disposizione nella cartella delle reti correttive. Questa funzione ha bisogno della pulsazione di attraversamento obiettivo  $\omega_{c,new}$ , del reciproco del modulo  $m=1/modulo$ , e di theta  $\theta = \Phi_m - (180 - |\argomento|)$

`m=1/modulo;`

`theta=50-(180-abs(argomento));`

`[tauz,taup]=generica(wc_new,m,theta);`

Assicuriamoci che la condizione  $\tau_z > \tau_p > 0$  sia verificata:

 taup	0.2077
 tauz	2.6614 come vediamo è rispettata

- Costruiamo la rete correttrice vera e propria:

```
C_lead=(1+s*tauz)/(1+s*taup);
```

- Mettiamola in serie con il controllore statico

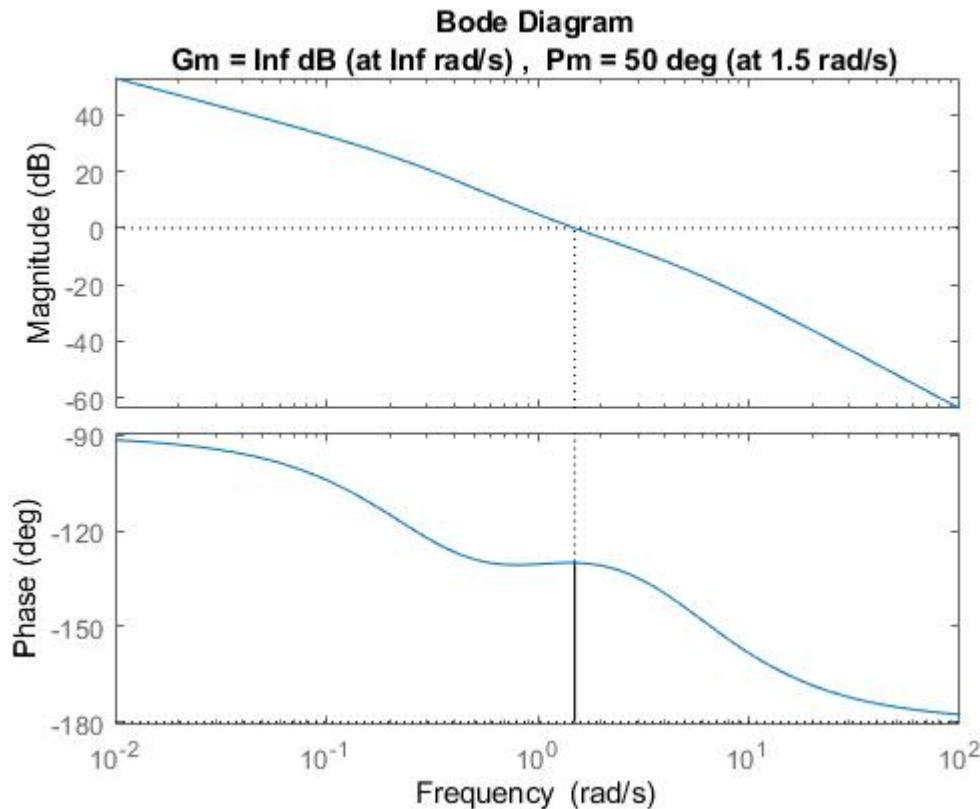
```
C=series(C_s,C_lead);
```

- Costruiamo la funzione di anello definitiva

```
L=series(C,G);
```

- E studiamo il diagramma dei moduli e delle fasi

```
margin(L);
```



Come vediamo abbiamo raggiunto grazie alla rete anticipatrice i parametri imposti di pulsazione e margine di fase. Inoltre, come anticipato in precedenza abbiamo portato il sistema ad essere **BIBO stabile**.

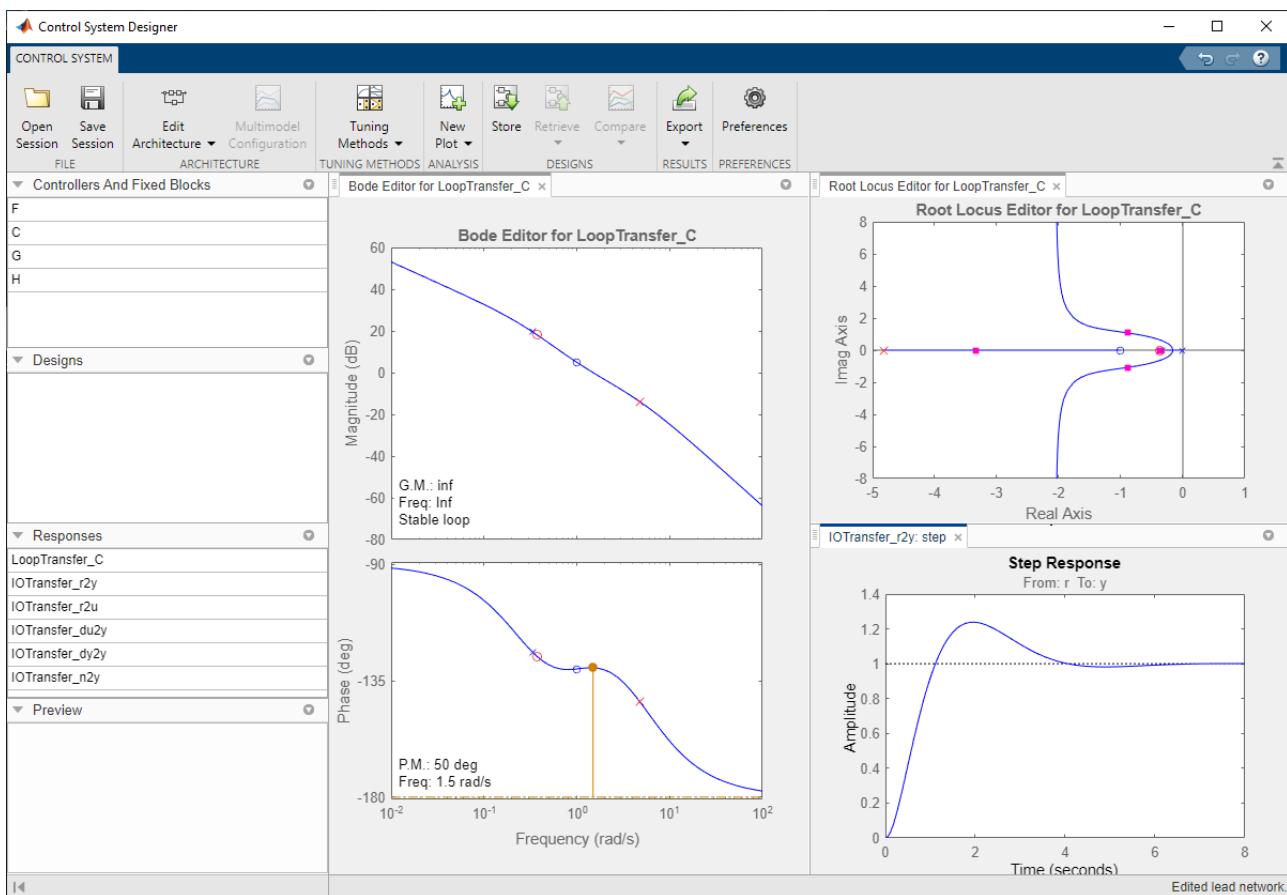
- Estraiamo lo zero il polo col comando:

```
[zero,polo,guadagno]=zpkdata(C_lead,'v');
```

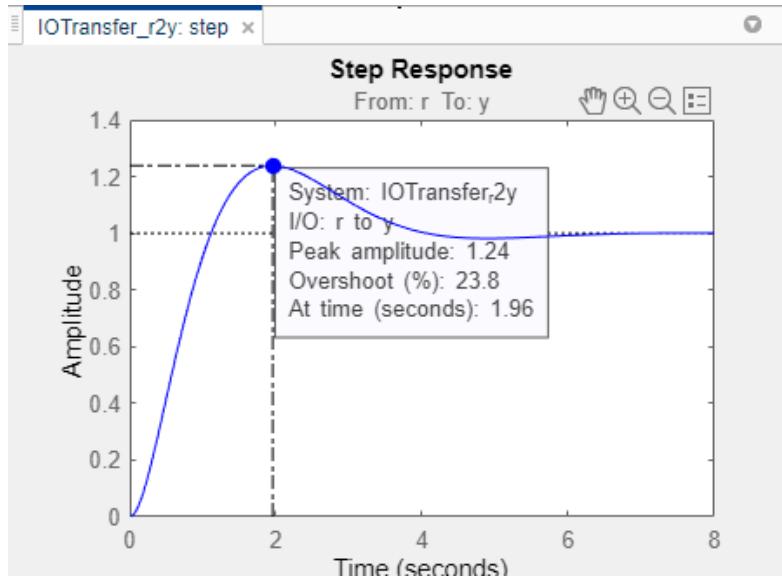
Ora verifichiamo, avvalendoci di sisotool, che ci troviamo con i conti:

- Carichiamo la nostra funzione di trasferimento dal pulsante edit Architecture
- **aggiungiamo il controllore statico e la rete attenuatrice dal menu edit compensator**
- impostiamo il tempo di assestamento al 5% e siamo pronti a verificare

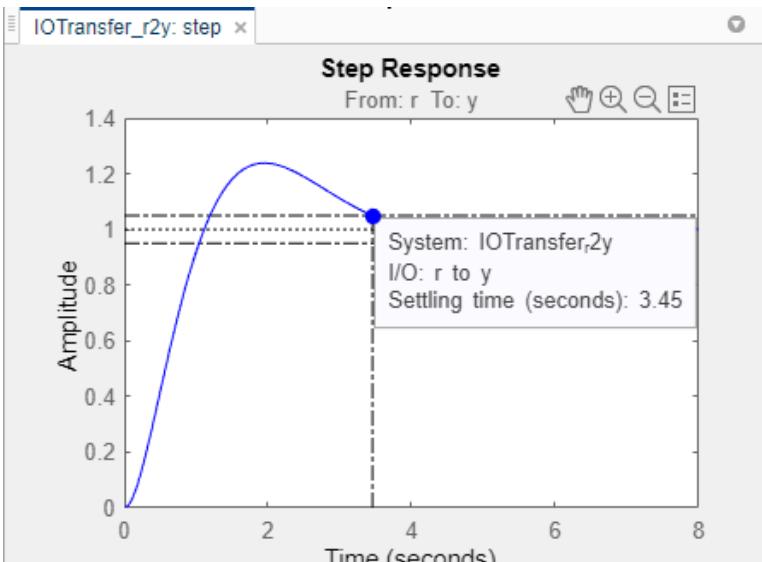
La nostra situazione è questa:



Dedichiamo maggiore attenzione alla step response (risposta al gradino):



La massima sovraelongazione è al 23.8% < 25% (ci siamo!)

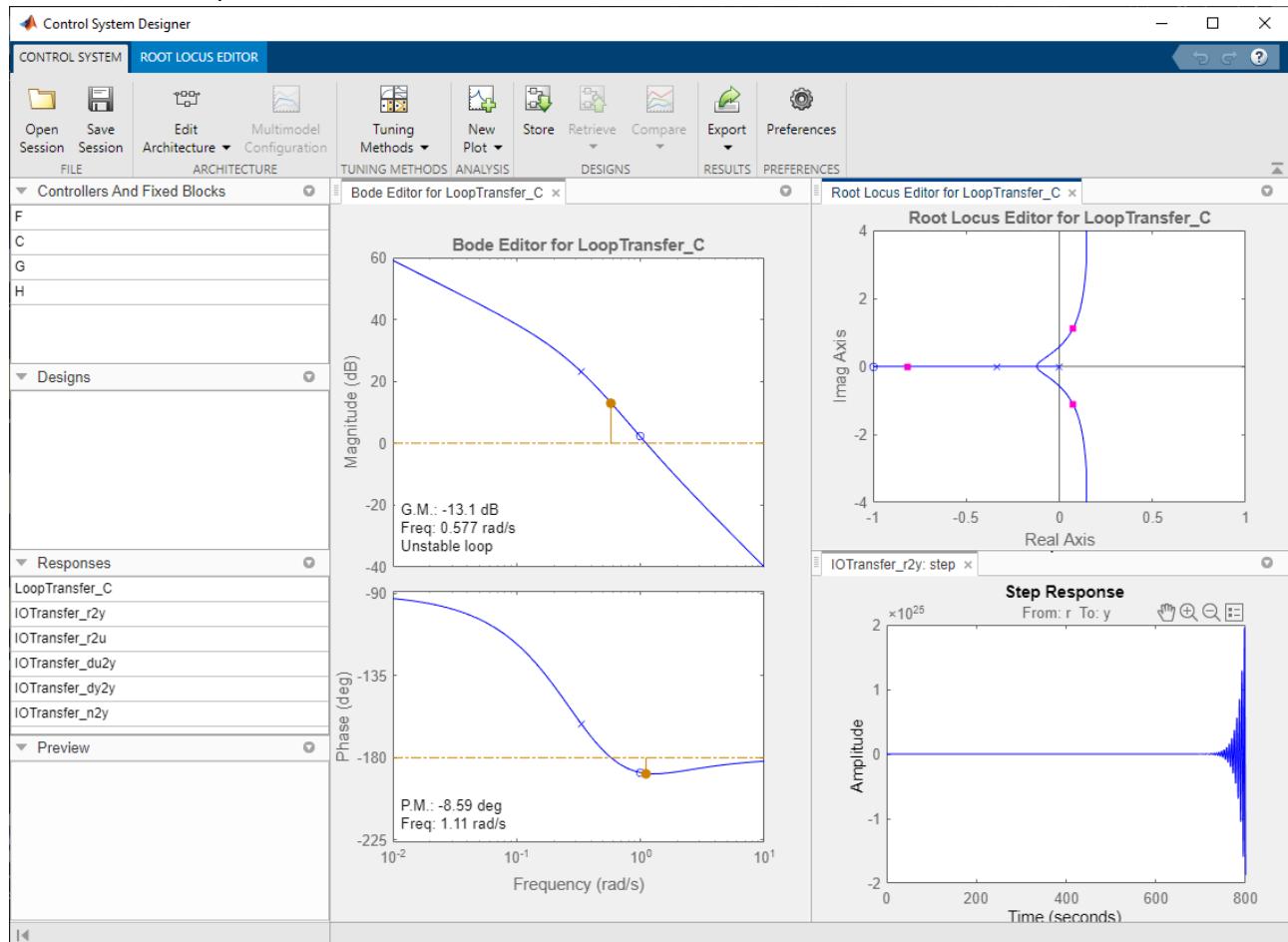


Il tempo di assestamento è 3.45 sec < 70 sec (perfetto!)

*Per onor di cronaca ho voluto provare a usare anche il metodo delle radici*

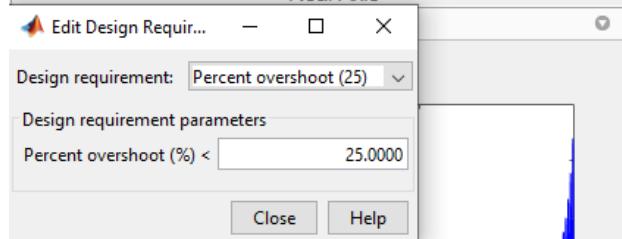
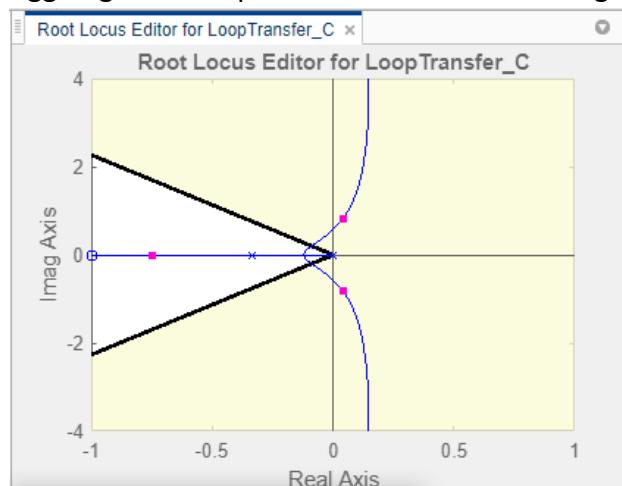
Un altro modo di procedere è quello di fare tutto direttamente con lo strumento sisotool:

- Carichiamo sempre la nostra funzione di trasferimento da Edit Architecture

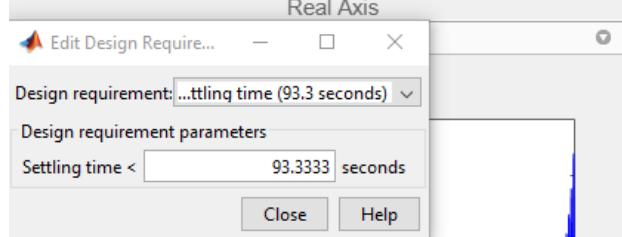
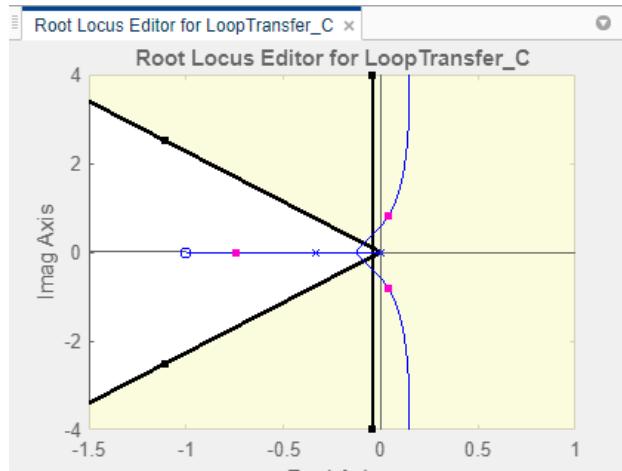


- Aggiungiamo il controllore statico ( $C(s) = 0.5$ )

- Aggiungiamo i requisiti di massima sovraelongazione



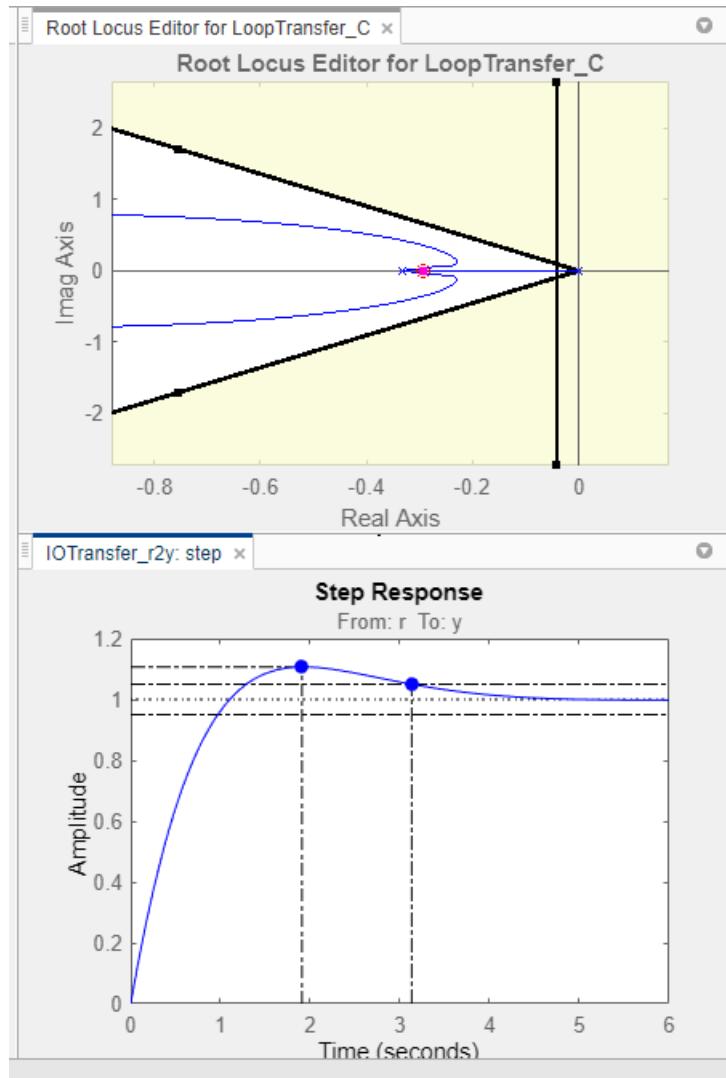
- E il tempo di assentamento ( convertiamo il tempo di assestamento al 5% nel tempo al 2% ,70 \* 4/3)



- Le specifiche sono rispettate se i poli si trovano nella zona bianca**
- Allora devo cercare di far convergere i nostri poli all'interno della zona bianca e al contempo lasciare il **controllore fisicamente realizzabile** ( $n$  poli =  $n$  zeri). Nel nostro caso quindi devo aggiungere sia un polo che uno zero (lo zero attrae i rami a se, quindi dovrò metterlo in un

punto dal quale i rami entrano nella zona obiettivo, il polo mi serve per bilanciare fisicamente il controllore)

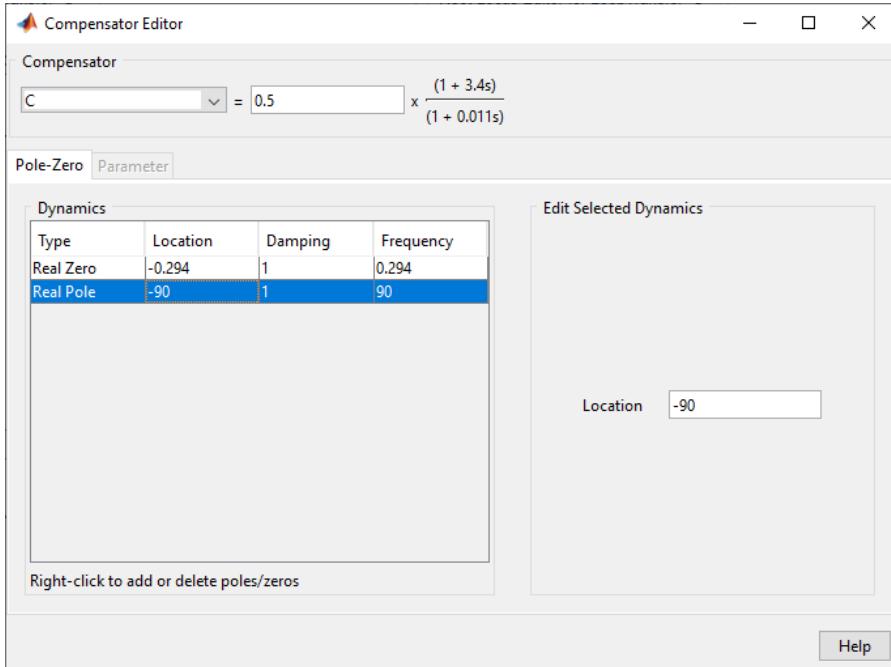
- Provo a inserire uno zero molto vicino al confine della “safe zone”



Ottengo una massima sovraelongazione del 10.7% e un tempo di assestamento di 3.13 sec

- Ora però il controllore ha uno zero quindi devo aggiungere un polo che lo renda fisicamente realizzabile. Devo scegliere un polo che non mi rovini il risultato ottimo appena ottenuto quindi scelgo **un polo abbastanza lontano** che non dia molto fastidio (Scelgo un polo reale in posizione -90)

- Questo sarà il nostro controllore finale



Le nostre specifiche sono rimaste pressoché invariate con l'aggiunta del polo, la massima sovraelongazione finale è di 10.9% e il tempo di assestamento al 5% è di 3.12 sec

L'ultima cosa possiamo analizzare il comportamento del sistema retroazionato ottenuto rispetto al sistema retroazionato approssimato.

Esportiamo la funzione di trasferimento che lega la funzione di trasferimento all'uscita regolata dentro matlab:

- Usiamo il comando export in modo da ritrovarci con la rappresentazione ISU del sistema retroazionato e usiamo il comando zpk per esplicitare la G(s)

```
>> zpk(T)

ans =
From input "r" to output "y":
153.24 (s+0.2937) (s+1)
-----
(s+88.28) (s+0.2933) (s^2 + 2.097s + 1.738)
```

- Costruiamo il sistema approssimato usando solo i poli complessi e coniugati

```
>> T_rid=1.738/(s^2 + 2.097*s + 1.738)
```

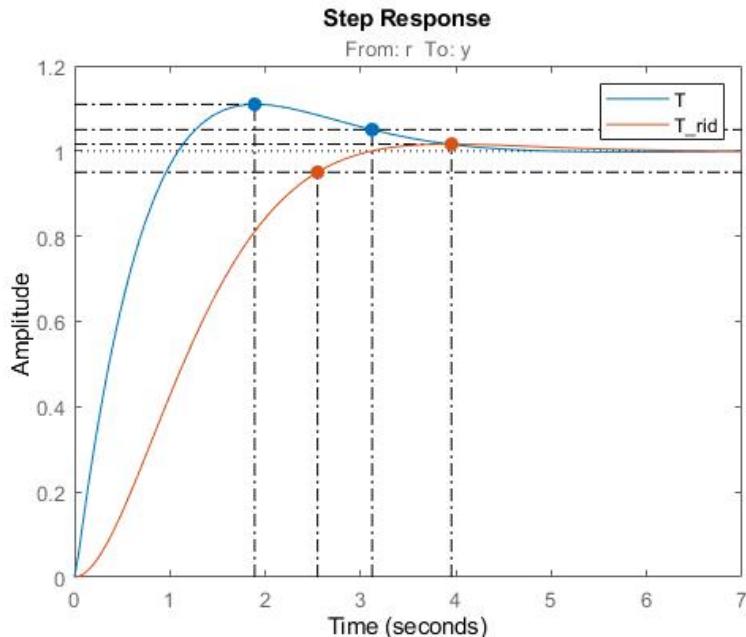
$T_{rid} =$

1.738

-----

( $s^2 + 2.097s + 1.738$ )

- Vediamo la risposta al gradino di entrambi



Abbiamo che la massima sovraelongazione del sistema effettivo è al 10.9% con un tempo di assestamento di 3.12 sec, mentre del sistema approssimato abbiamo  $S=1.62\%$  e  $t_{s,5}=2.55$  sec

Le differenze sono dovute agli zeri del sistema effettivo che sollevano la risposta al gradino