

Progetto Laboratorio di Automatica

Paolo Pangallo

2023

1 Regolatore digitale

Data $G(s) = \frac{-s+0.1}{(s+0.6)(s+8)}$, la prima parte del progetto consiste nella realizzazione di un regolatore standard digitale in grado di garantire le seguenti specifiche:

1. errore nullo rispetto ad un riferimento a gradino.
2. Sovra-elongazione percentuale massima pari al 20% e tempo di assestamento minore di 20sec.

1.1 Cos è un controllore digitale?

Un regolatore digitale è un sistema il cui compito è quello della trasformazione del segnale di errore campionato in una sequenza di campioni che, opportunamente ricostruita, è il segnale di comando del processo. Tale operazione di trasformazione avviene secondo un determinato algoritmo.

1.2 Perché scegliere un controllore digitale?

La scelta di un controllore digitale offre vari vantaggi rispetto ad un controllore analogico. Riassumendo i più importanti:

1.2.1 Flessibilità Algoritmica:

L'adozione di controllori digitali consente l'utilizzo di algoritmi più sofisticati, garantendo la flessibilità necessaria per adattarsi alle complesse dinamiche del sistema.

1.2.2 Stabilità e Coerenza:

La stabilità intrinseca e la coerenza dell'elaborazione numerica dei segnali nei controllori digitali offrono una maggiore affidabilità operativa, particolarmente critica nelle applicazioni dove la consistenza è essenziale.

1.2.3 Facile Modificabilità:

La possibilità di intervenire sul software senza dover apportare modifiche fisiche ai regolatori semplifica notevolmente il processo di aggiornamento e adattamento del sistema nel corso del progetto.

1.2.4 Precisione e Sensibilità:

La sensibilità concentrata nel codificatore nei controllori digitali contribuisce a migliorare la precisione del sistema, un aspetto cruciale per il successo delle nostre applicazioni.

1.2.5 Soglie e Saturazioni:

L'assenza di soglie e saturazioni durante l'elaborazione digitale dei segnali assicura una maggiore robustezza operativa, riducendo al minimo gli effetti di eventuali condizioni limite.

1.2.6 Trasmissione a Distanza:

La trasmissione digitale non solo migliora la resistenza al rumore, ma permette anche l'implementazione di tecniche avanzate di correzione degli errori, un aspetto cruciale per le applicazioni che richiedono trasmissioni a distanza.

1.3 Sintonizzazione dei controllori PID:

Esistono vari metodi per sintonizzare un controllore PID, ognuno dei quali presenta i propri vantaggi e un grado di difficoltà implementativo differente. Vedremo di seguito una possibile implementazione con il metodo Ziegler-Nichols.

1.3.1 Il metodo Ziegler-Nichols per un sistema a ciclo chiuso

Il metodo di ZN a ciclo chiuso è applicabile unicamente a quei processi che ammettono un guadagno critico finito, cioè quei processi tali che il corrispondente luogo delle radici entra nel semipiano destro per valori sufficientemente elevati del guadagno. Per contro, il metodo di ZN a ciclo chiuso può essere applicato sia a processi con poli nell'origine, la cui risposta al gradino a ciclo aperto diverge, che a processi aventi poli complessi coniugati e quindi risposta oscillatoria, due scenari in cui il metodo di ZN a ciclo aperto non può essere applicato (esso richiede infatti che la risposta al gradino a ciclo aperto converga senza oscillazioni verso un valore costante).

Nel nostro caso, notiamo che il sistema a è compatibile con le ipotesi di lavoro di Ziegler-Nichols e quindi possiamo procedere con la prima fase.

1.3.2 Fase 1: Guadagno e tempo di oscillazione

Una volta implementato, grazie all'aiuto di Simulink, il sistema a ciclo chiuso con gli opportuni blocchi, la prima fase consiste nell'introdurre un guadagno molto basso (generalmente si sceglie $k=1$) attraverso un blocco gain. Con un pò di pazienza si incrementa volta per volta il "gain" fino ad ottenere oscillazioni sostenute. Nel nostro caso ci si accorge che per valori compresi tra 8 e 9 si riesce ad ottenere delle oscillazioni grossomodo stabili e, con qualche tentativo aggiuntivo, si ottiene il valore "ottimale" 8.6. Da questo valore si simula il processo e si valuta geometricamente il tempo di oscillazione che in questo caso è di 2.735 secondi.

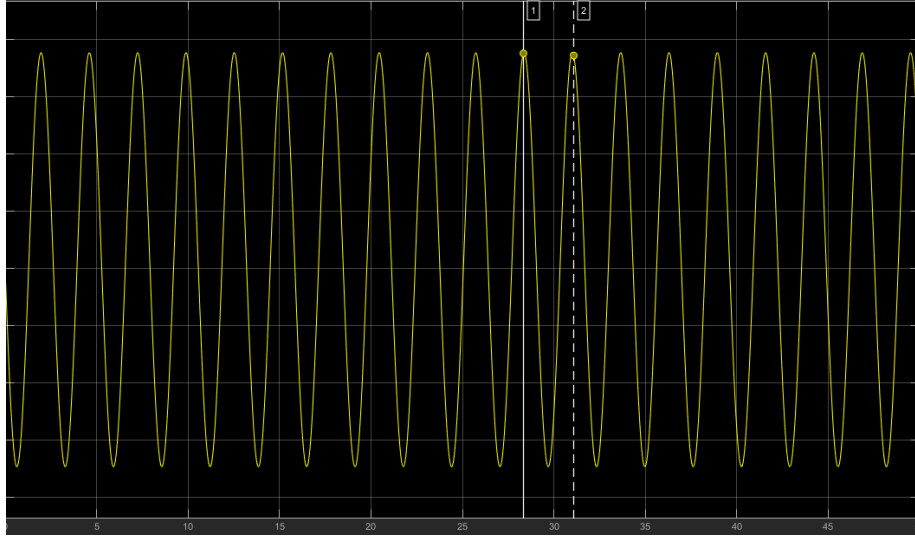


Figure 1: Valutazione geometrica tempo di oscillazione tramite simulink

1.3.3 Fase 2: Implementazione del Controllore

La scelta di un controllore adeguato non è triviale, e in alcuni casi bisogna effettuare alcuni test prima di determinare il controllore che meglio si adatta all'impianto. In questo caso le specifiche richiedono un errore nullo a regime e un tempo di assestamento non proibitivo, quindi un controllore PI potrebbe essere una buona scelta. Il metodo Ziegler-Nichols offre dei parametri iniziali per K_p , K_i (e K_d) in grado di ottenere una prima stima del controllore basandosi su guadagno e tempo di oscillazione precedente calcolati. Per un controllore di tipo Pi abbiamo:

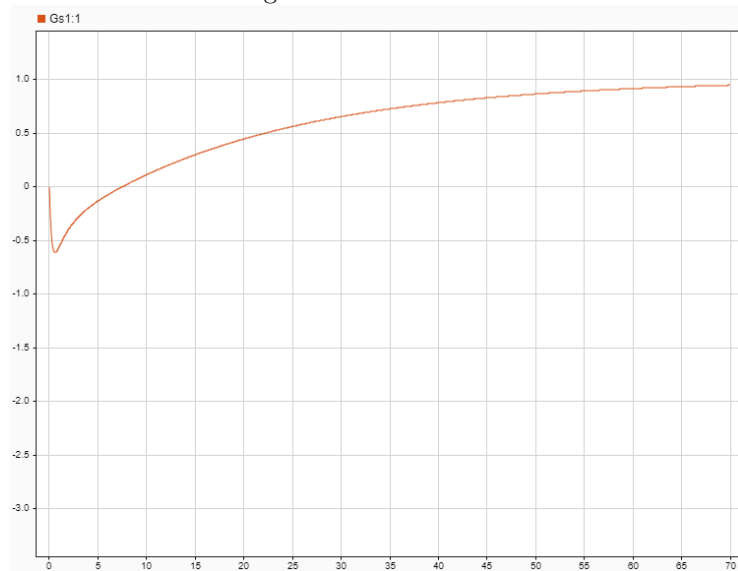
1. $k_p = 0.4k_c = 3.44$
2. $k_i = k_p/0.8t_c = 1.57$

Implementiamo il controllore in Simulink e verifichiamo come reagisce il sistema a ciclo chiuso

Notiamo come il sistema sia stabile ma abbia una risposta pessima che si assesta dopo molto tempo. In questo caso i parametri vanno rivisti "al rialzo" senza esagerare, poich' altrimenti si rischia di alterarne la stabilit . Otteniamo i nuovi parametri come segue:

1. $k_p = 1.5k_p = 5.16$

Figure 2: Taratura Pid.



2. $k_i = 1.5k_i = 4.07$

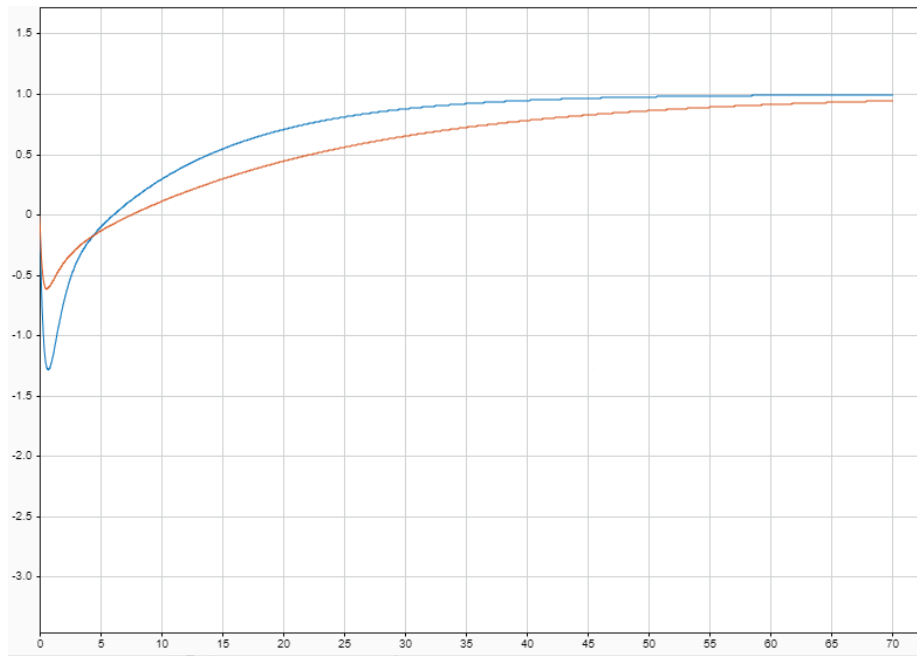


Figure 3: Comparazione controllori

Notiamo come il nuovo sistema abbia complessivamente delle prestazioni dinamiche più stabili, abbia mantenuto la sua stabilità e l'errore nullo a regime siano ancora un po' lontani sulle specifiche di sovraelongazione e tempo di assestamento, quindi la taratura va modificata ulteriormente, anche stavolta senza esagerare.

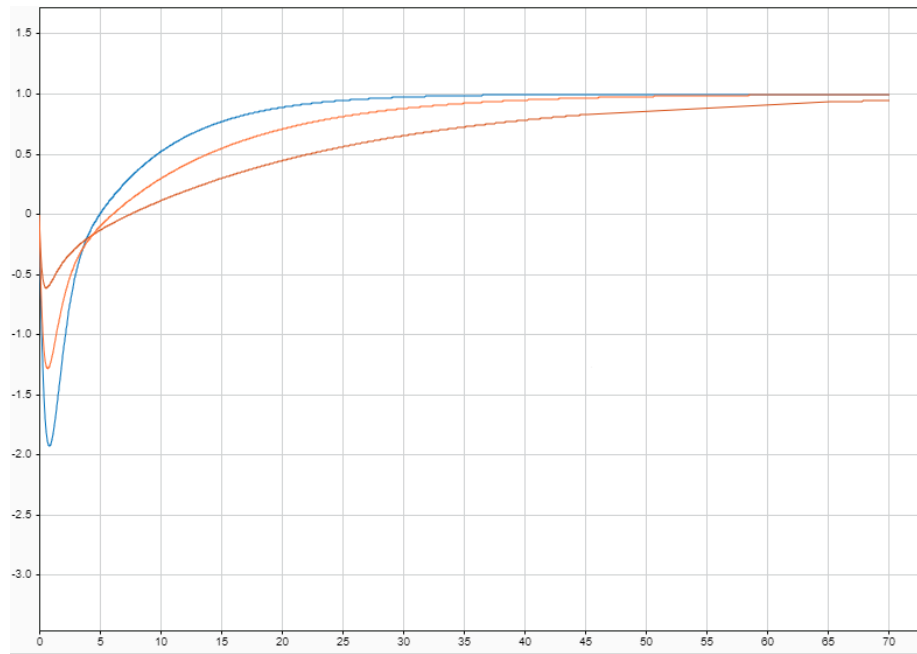


Figure 4: Comparazione controllori

Come si può notare il sistema sta "migliorando", quindi l'aumento di k_p e k_i è il modo migliore di procedere.

1.3.4 Fase 3: Verifiche e considerazioni conclusive

Sono stati resi necessari ulteriori test per tarare un controllore che rispettasse correttamente tutte le specifiche dinamiche e rappresentarle tutte appesantirebbe di molto la trattazione senza aggiungere nulla che già non si sappia.

Si conclude la parte sulla taratura del controllore mostrando come i valori $k_p = 6.66$ e $k_i = 4.07$ siano ottimali per il sistema proposto.

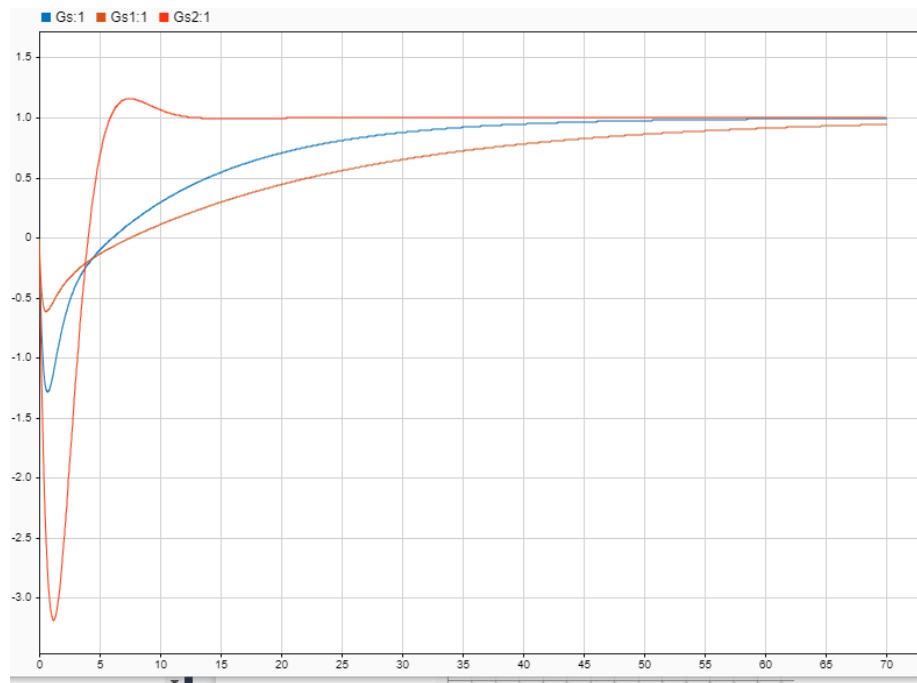


Figure 5: Comparazione controllori

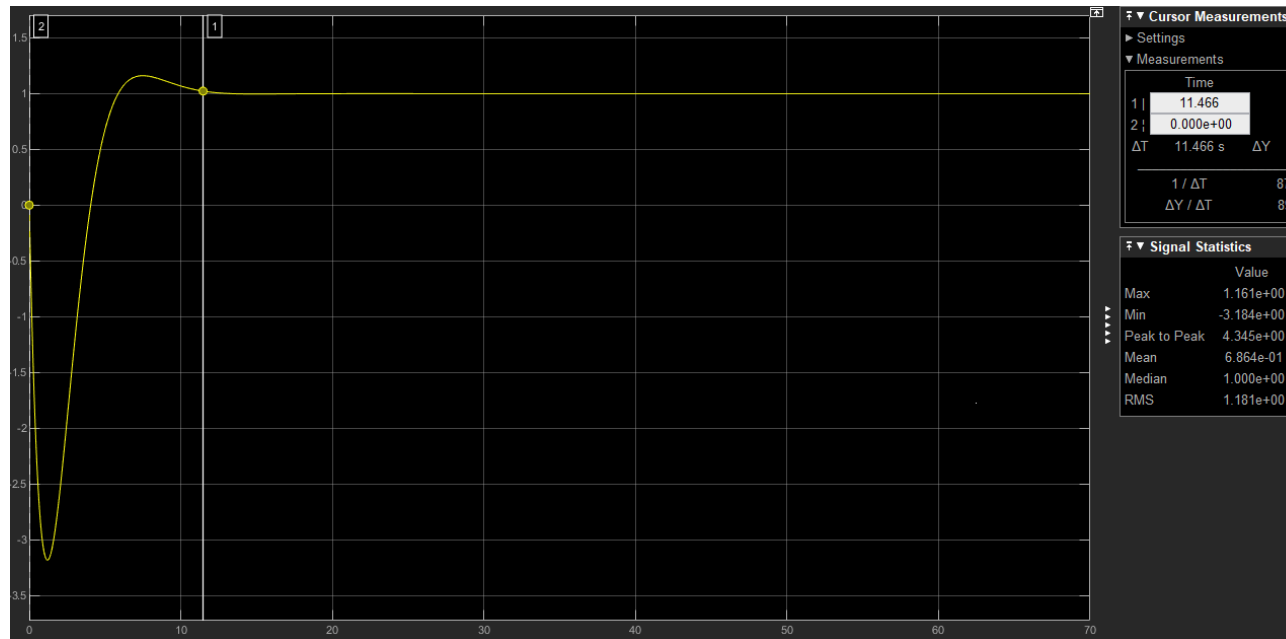


Figure 6: Sistema su blocco scope

L'ultima immagine consiste nella rappresentazione grafica del sistema vista dal blocco Scope di simulink, che ci permette di visualizzare alcune "statistiche" del sistema. Il valore 11.466 indica il tempo di assestamento del sistema mentre il max indica il picco della funzione che coincide in questo caso con la sovra-elongazione pari quindi all'11.61 %.

Il sistema rispetta tutte le specifiche ed è pronto per essere discretizzato.

1.4 Discretizzazione PI

La discretizzazione di un controllore, una volta nota la sua forma matematica al tempo continuo, non è un procedimento complesso, soprattutto grazie all'ausilio di strumenti come Matlab e Simulink. Bisogna tuttavia fare alcune considerazioni di carattere teorico per evitare che la discretizzazione descriva il controllore in maniera troppo poco "accurata"

1.4.1 Dal tempo continuo al tempo discreto

La conversione da tempo continuo a tempo discreto passa innanzitutto dalla scelta del tempo di campionamento e sull'assunzione secondo la quale una cattiva scelta di quest'ultimo possa produrre situazioni spiacevoli differenti.

1.4.2 Tempo di campionamento troppo basso: problemi e soluzioni

Problema Aliasing: Se il tempo di campionamento è troppo basso rispetto alla frequenza del segnale, può verificarsi aliasing. Questo si traduce nella distorsione del segnale e nella perdita di informazioni.

Soluzione: Teorema di Nyquist-Shannon suggerisce che il tempo di campionamento debba essere almeno il doppio della frequenza massima del segnale.

Problema Requisiti computazionali: Tempi di campionamento troppo bassi possono richiedere risorse computazionali significative, portando a problemi di prestazioni e complessità nell'implementazione.

Soluzione: Ottimizzare gli algoritmi di campionamento e le risorse di calcolo per mantenere un equilibrio tra accuratezza e efficienza

1.4.3 Tempo di campionamento troppo alto: problemi e soluzioni

Problema Perdita di informazioni: Un tempo di campionamento troppo alto può causare la perdita di informazioni dettagliate e veloci nei segnali o nei dati dinamici.

Soluzione: Ridurre il tempo di campionamento per catturare meglio le variazioni rapide nei segnali.

Problema Spreco di risorse: Un campionamento troppo alto può comportare un uso inefficiente delle risorse, specialmente in sistemi embedded o con vincoli di risorse.

Soluzione: Ottimizzare il tempo di campionamento in base alle esigenze specifiche del sistema, evitando sprechi di risorse.

1.5 Scelta del tempo di campionamento

Tralasciando queste assunzioni pratiche, la bontà di un controllore va valutata a posteriori, non è quindi necessario ricavarne uno in maniera definitiva già da subito, ma come orientarsi?

Generalmente si prendono come riferimento due caratteristiche dinamiche del sistema: tempo di assestamento e tempo di salita secondo questi due criteri.

$$\frac{t_r}{20} \leq t_c \leq \frac{t_r}{10} \quad \text{e} \quad t_c \leq \frac{t_s}{40}$$

con

$$T_r$$

Time Rising (tempo di salita) e

$$T_s$$

Time settling (tempo di assestamento).

Oppure una strategia alternativa è considerare il polo più veloce del sistema:

$$\tau = \frac{1}{\text{polo più veloce}} = \frac{1}{8}. \quad \text{Quindi:}$$

$$T_c = \frac{\tau}{10} = \frac{1/8}{10} = 0.0125$$

nel nostro caso le specifiche garantiscono un tempo di assestamento al più di 20s, mentre il tempo di salita sarà intuibile dal grafico a tempo continuo, di circa 6 secondi.

Quindi otteniamo:

1.

$$t_{c1} = 0.6$$

2.

$$t_{c2} = 0.5$$

3.

$$t_{c3} = 0.0125$$

Dati alla mano generalmente il T_c preferibile è il minore tra i tre, in questo caso quello relativo al polo più veloce. Di seguito verranno testati tutti e 3 per poterne valutare e apprezzare le differenze.

1.5.1 Conversione e test

Generalmente per passare dal tempo continuo al tempo discreto si valutano 3 algoritmi

1. Eulero in avanti (o Eulero standard) = $s \rightarrow \frac{1}{T_c} \cdot (z - 1)$
2. Eulero all'indietro = $s \rightarrow \frac{1}{T_c} \cdot \frac{z-1}{z}$
3. Tustin = $s \rightarrow \frac{2}{T_c} \cdot \frac{z-1}{z+1}$

Matlab permette, tramite il comando `c2d` di convertire un segnale continuo in un segnale discreto utilizzando Tustin, mentre per i metodi di eulero è necessario sostituire i valori manualmente.

Vedremo di seguito alcuni test svolti con tutti e 3 i metodi per capirne le differenze; alla fine verrà proposta una "scorciatoia", ovvero l'implementazione di un PI già discretizzato da Matlab a cui sarà necessario solamente inserire uno dei tempi di campionamento precedentemente calcolati.

Scegliamo inizialmente di discretizzare il controllore tramite Tustin e lo tarriamo con tutti e 3 i tempi di campionamento precedentemente calcolati.

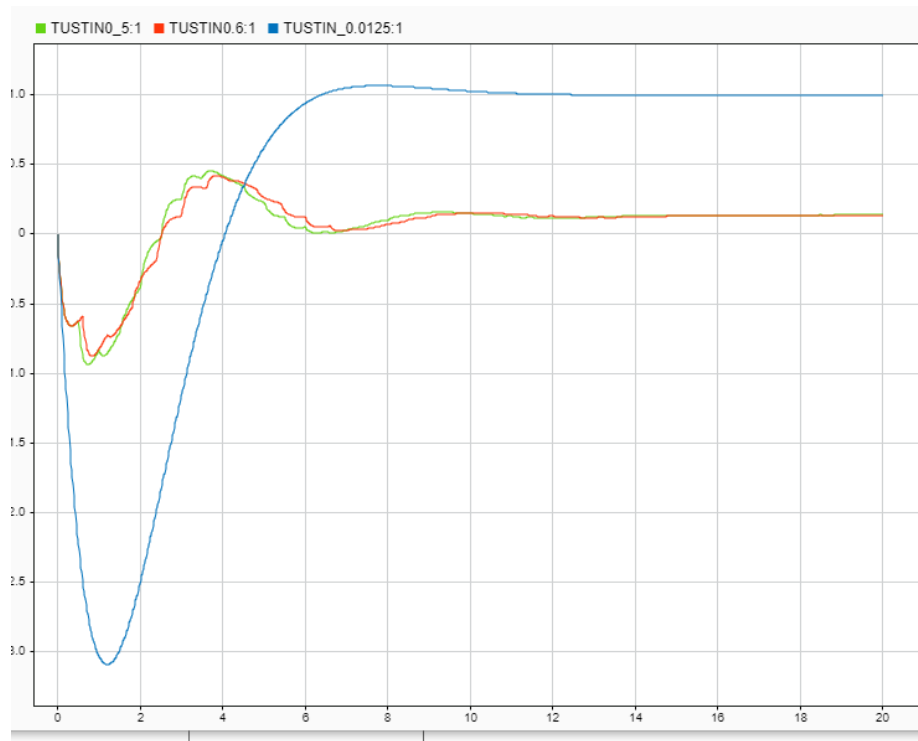


Figure 7: Confronto tra I vari tempi di campionamento

Notiamo che il tempo di campionamento adeguato è 0.0125, utilizzeremo questo valore per valutare le differenze tra Tustin e i metodi di eulero.

Implementando i tre metodi tramite il blocco 'Discrete Transfer fnc' di Simulink otteniamo le 3 discretizzazioni confrontate con il sistema a tempo continuo

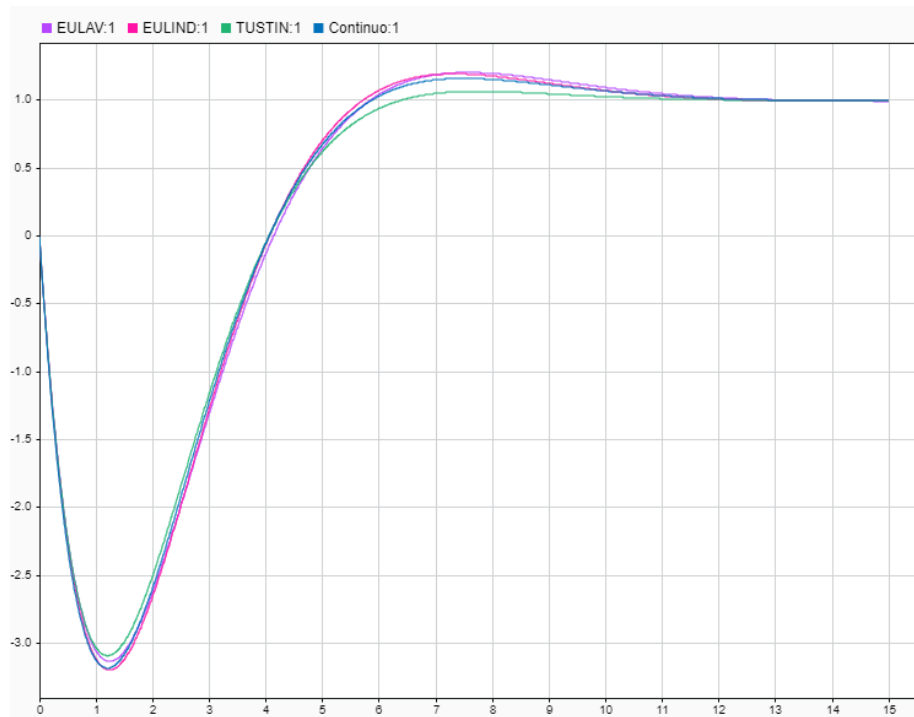


Figure 8: Eulero e Tustin a confronto

Notiamo come i metodi di eulero siano entrambi molto fedeli al controllore a tempo continuo e forniscano entrambi una buona rappresentazione della sua versione discreta senza una perdita eccessiva di prestazioni, avremo in fatti per Eulero in avanti ed Eulero all'indietro un tempo di assestamento pressoché identico alla loro versione in continuo, e una sovraelongazione al 'limite' delle specifiche, corrispettivamente del 20.5% e del 19.5%.

Prima di terminare la trattazione sulla discretizzazione forniamo un'alternativa ai metodi di discretizzazione affrontati precedentemente.

É infatti possibile discretizzare un controllore di tipo PID (e quindi anche PI, PD ecc.), ipotizzando di conoscerne il modello matematico (in questo caso i valori di P, I, e D) direttamente attraverso il blocco simulink del controllore che abbiamo scelto modificando l'opzione 'time-domain' da 'continuos' a 'discrete' e impostando un tempo di campionamento a nostra scelta. Optando per $T_c=0.0125$ otteniamo

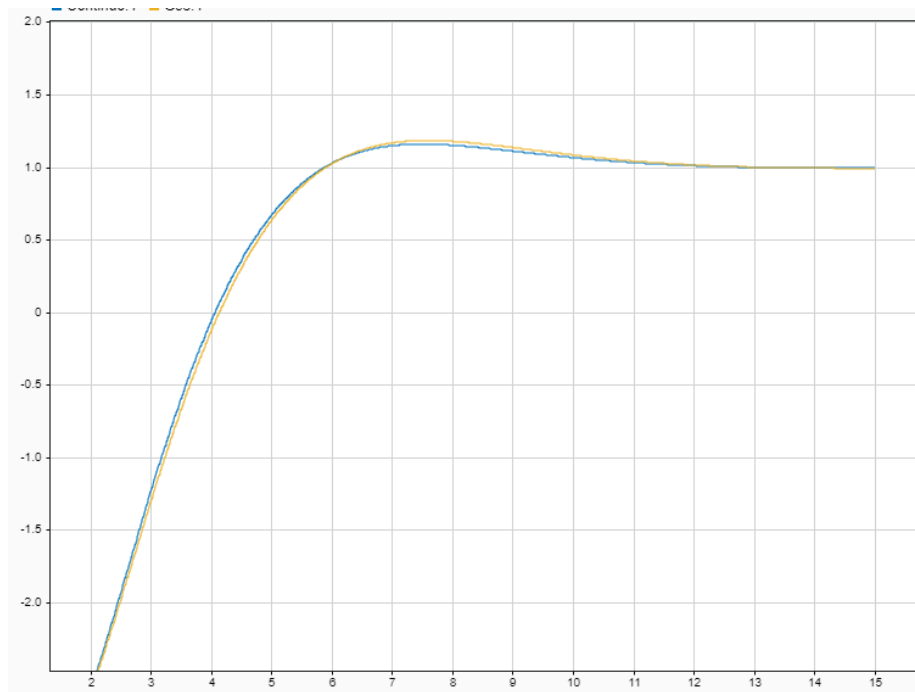


Figure 9: Confronto tra continuo e discreto tramite Simulink

Notiamo che anche in questo caso la discretizzazione è andata a buon fine.

2 Controllore dinamico

La seconda parte del progetto richiede di progettare un controllore dinamico data una funzione di trasferimento G :

$$G(s) = \frac{1}{s(s + 3.8)^3}$$

in grado di garantire:

1. errore massimo del 5% rispetto ad un riferimento a rampa con pendenza unitaria.
2. Sovra-elongazione percentuale massima pari al 20% e tempo di salita minore di 2 sec.

Prima di progettare il controllore assumiamo che il criterio di Nyquits sia rispettato e che non vi siano cancellazioni nel semipiano

destro fra regolatore e impianto. Queste assunzioni sono fondamentali per garantire la stabilità interna del sistema, assumemendo quindi che le uscite corrispondenti ai segnali di ingresso(riferimenti, disturbi e rumore) siano limitate.

Un ulteriore semplificazione per garantire le specifiche sulla precisione statica consiste nel presupporre rumore e disturbo nulli.

2.1 L'errore come parametro della precisione statica

Nell'analisi dell'errore a regime, e quindi nella scelta di un controllore adeguato che lo "tenga sotto controllo", è fondamentale scegliere correttamente il riferimento corretto, in base al tipo di funzione ad anello.

Distinguiamo alcuni casi tipici:

	<i>Gradino</i>	<i>Rampa</i>	<i>RampaQuadra</i>
<i>Tipo0</i>	$1/k_s$	∞	∞
<i>Tipo1</i>	0	$1/k_v$	∞
<i>Tipo2</i>	0	0	$1/k_a$

Table 1: Tabella degli errori

Nel nostro caso, le specifiche della funzione e del riferimento richiesto renderanno necessario l'utilizzo di un controllore k, senza l'aggiunta di ulteriori poli.

Il guadagno del controllore, per un sistema definito nel dominio di Laplace, viene calcolato dal teorema del valore finale: se $\lim_{t \rightarrow +\infty} e(t)$ esiste allora: $\lim_{t \rightarrow +\infty} e(t) = \lim_{s \rightarrow 0} s \cdot E(s)$.

Poiche $E(s) = \frac{1}{1+L(s)}R(s)$ con $R(s) = \frac{1}{s^2}$ e $C(s) = K$ determiniamo il valore di guadagno k a partire da:

$$\lim_{s \rightarrow 0} s \cdot \frac{1}{K \cdot G(s)} \cdot \frac{1}{s^2} < \frac{1}{20} = \lim_{s \rightarrow 0} s \frac{1}{K * \frac{1}{s(s+3.8)^3}} * \frac{1}{s^2} \implies$$

$$\frac{1}{K * \frac{1}{54.87}} < 0.05 \implies K > 1097,4$$

Ponendo $K=1100$ e notiamo come il sistema non sia stabile e diverga molto rapidamente a valori altissimi. Questo perché la scelta di un controllore adeguato non sempre permette di soddisfare le specifiche sulla precisione statica, in questo caso vediamo come il sistema non riesce a inseguire il riferimento proposto

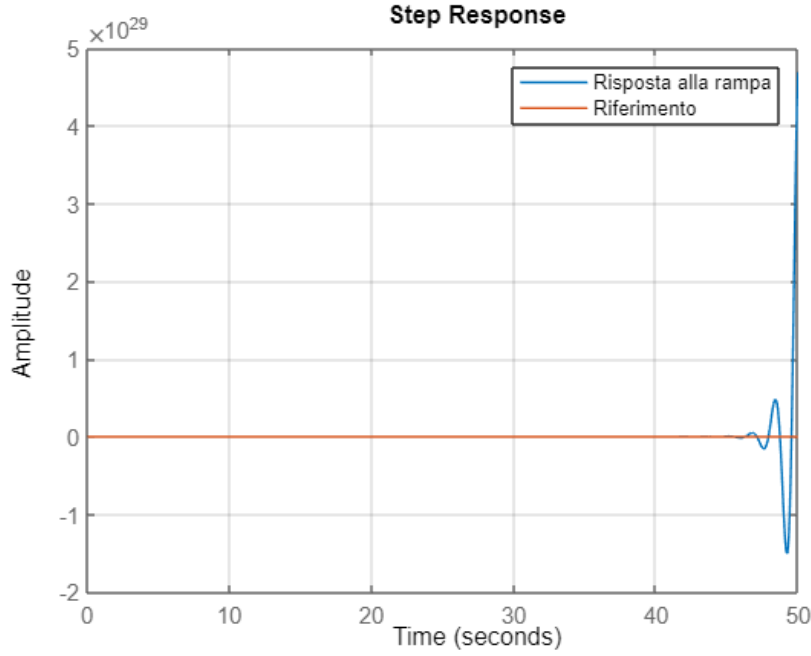


Figure 10: Confronto tra riferimento e risposta

2.2 Reti corretttrici

L'utilizzo di una rete corretttrice è necessario per soddisfare alcune specifiche dinamiche desiderabili (bassa Sovra-elongazione, buon margine di fase, basso tempo di assestamento ecc.). La scelta di una rete adeguata dipende dalle caratteristiche intrinseche del sistema a ciclo chiuso quali guadagno e margine di fase. Oltre al caso triviale

$$\begin{cases} |L(j\tilde{\omega}_c)| < 1 \\ \pi - |\angle L(j\omega_c)| > \tilde{\phi}_M \end{cases}$$

Nel quale andrà ritoccato il valore del controllore implementato per la precisione statica, è possibile introdurre una delle 3 seguenti reti corretttrici.

2.2.1 Rete anticipatrice

Una rete anticipatrice è richiesta quando è necessario recuperare il guadagno e migliorare il margine di fase, ovvero nel caso in cui:

$$\begin{cases} |L(j\tilde{\omega}_c)| < 1 \\ \pi - |\angle L(j\omega_c)| < \tilde{\phi}_M \end{cases}$$

2.2.2 Rete Ritardatrice

Una rete ritardatrice o attenuatrice è richiesta invece quando è necessario abbassare il guadagno cercando di non peggiorare il margine di fase. Ovvero nel caso in cui

$$\begin{cases} |L(j\tilde{\omega}_c)| > 1 \\ \pi - |\angle L(j\omega_c)| > \tilde{\phi}_M \end{cases}$$

2.2.3 Rete a sella

Una rete a sella è necessaria quando si deve sia attenuare il guadagno che recuperare il margine di fase, ovvero nel caso in cui:

$$\begin{cases} |L(j\tilde{\omega}_c)| > 1 \\ \pi - |\angle L(j\omega_c)| < \tilde{\phi}_M \end{cases}$$

Le strutture delle 3 reti sono le seguenti

Rete anticipatrice: $\frac{1+sT}{1+s\alpha T}$

Rete ritardatrice: $\frac{1+sT}{1+s\alpha T}$

Rete a sella: $\frac{(1+s\alpha T1)}{(1+sT1)} \cdot \frac{(1+sT2)}{(1+s\alpha T2)}$

Valutiamo i criteri con i quali scegliere le incognite

$$m, \quad \alpha, \quad \tau$$

.

2.3 Taratura dei parametri per una rete ritardatrice

Dalla specifica sul tempo di salita (massimo) richiesto: $w_b = 0.8 \frac{\pi}{T_r} = 1.25$, ricaviamo un lower bound per la pulsazione di attraversamento, ottimiamo quindi per un valore sulla banda passante che sia \geq , ad esempio 1.5.

Applichiamo la rete al sistema, che ricordiamo essere già stato precedentemente tarato da un controllore $G1(s)$ allo scopo di rispettare

i criteri della precisione statica. Quindi il nostro sistema $L(s)$ sarà (temporaneamente) dato da:

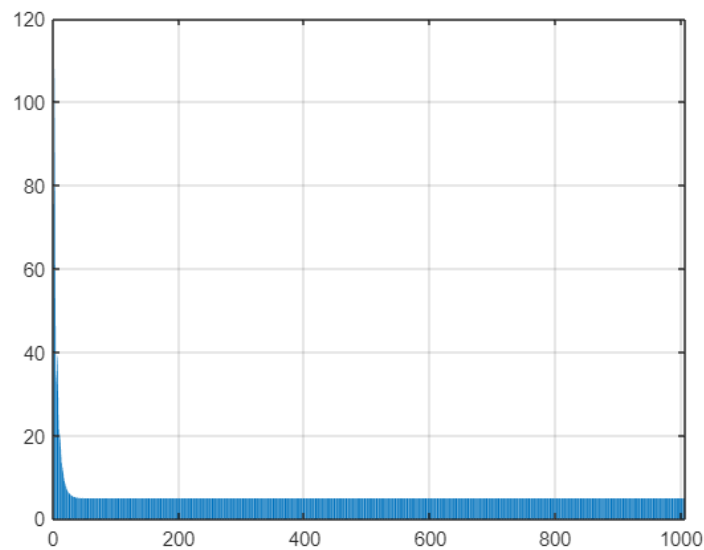
$$c_{\text{lead}} = \frac{1 + sT}{1 + s\alpha T}$$

$$C_1 = k = 1100$$

$$G(s) = \frac{1}{s(s + 3.8)^3}]$$

Valutiamo su Matlab come si comporta il nostro sistema a ciclo chiuso sia dal punto di vista grafico, che descritto nelle sue componenti rispetto ad un riferimento ad un gradino

Figure 11: Errore



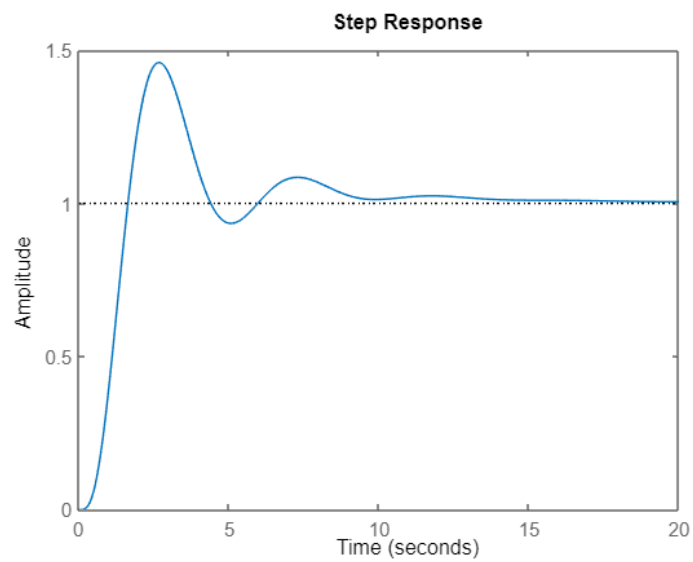


Figure 12: Sistema con rete attenuatrice

```
ans = struct with fields:
    RiseTime: 0.9403
    TransientTime: 12.7294
    SettlingTime: 12.7294
    SettlingMin: 0.9055
    SettlingMax: 1.4594
    Overshoot: 45.9426
    Undershoot: 0
    Peak: 1.4594
    PeakTime: 2.6759
```

Figure 13: Info sistema

Sulla base di quanto osservato, facciamo alcune considerazioni fondamentali:

1. Il sistema è stabile.

2. Le specifiche sulla precisione statica sono state rispettate (errore a regime intorno al 4.9%)
3. La Sovra-elongazione è "pessima" ovvero è più che raddoppiata rispetto alle specifiche richieste (45,6%)

È necessario quindi modificare ulteriormente il sistema senza snaturarlo, ricordando che l'obiettivo principale è risolvere i problemi sulla sovra-elongazione senza peggiorare le prestazioni della rete (per quanto possibile): optiamo per una rete anticipatrice.

2.4 Taratura dei parametri per una rete anticipatrice

Anche in questo caso calcoliamo

$$m, \quad \alpha, \quad \tau$$

considerando una sovra-elongazione massima del 20% e un tempo di salita massimo di 2 secondi.

Attraverso Matlab ricaviamo $m=1/M=4.99$, $\alpha=0.0149$ e $\tau=1.46$. La c_{lead} risultante, assumerà quindi il seguente valore:

$$C_{\text{lead}} = \frac{1.635s + 1}{0.02439s + 1}$$

Implementiamo adesso la rete ottenuta nel sistema che sarà quindi composto da tre controllori a cascata insieme al modello dell'impianto $G(s)$.

$$c_{\text{lead}} * c_{\text{lag}} * c_1 * G(s) = L(s) =$$

$$\frac{1.221 \times 10^4 s^2 + 9302s + 1120}{2.675s^6 + 140.2s^5 + 1368s^4 + 4911s^3 + 6063s^2 + 54.87s}$$

Valutiamo adesso il sistema a ciclo chiuso:

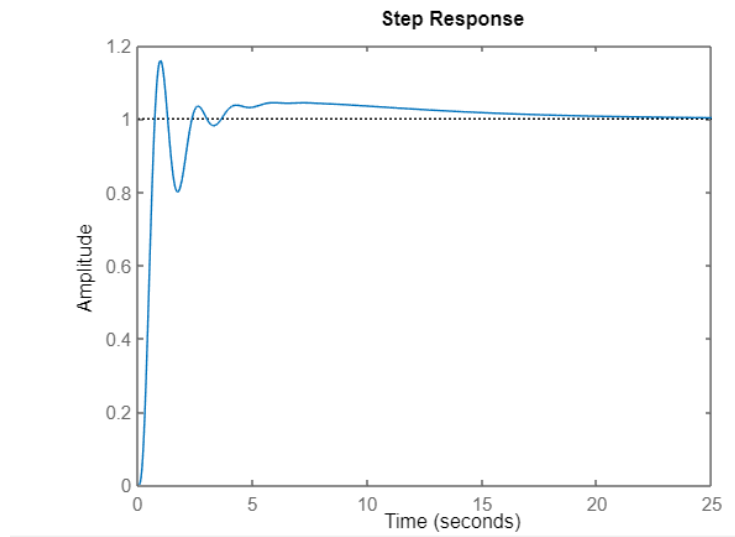


Figure 14: Grafico finale

```
ans = struct with fields:
    RiseTime: 0.4354
    TransientTime: 13.9710
    SettlingTime: 13.9710
    SettlingMin: 0.8005
    SettlingMax: 1.1592
    Overshoot: 15.9201
    Undershoot: 0
    Peak: 1.1592
    PeakTime: 0.9867
```

Figure 15: Info del sistema finale

Notiamo che la sovralongazione si è abbassata notevolmente e che tutte le specifiche sono rispettate con:

1. Errore a regime = $4.9\% \leq 5\%$
2. Sovralongazione = $15.92\% \leq 20\%$

3. Tempo di salita = $0.43 \text{ sec} \leq 2 \text{ sec}$