

**Università degli Studi di Padova**

---

FACOLTÀ DI INGEGNERIA  
Corso di Laurea triennale in Ingegneria dell'informazione

TESI DI LAUREA TRIENNALE

**Studio in ambiente Matlab/Simulink di un sistema di controllo  
per sospensioni attive**

Candidato:

**Andrea Aldegheri**

Matricola inf593053

Relatore:

**Alessandro Beghi**



# Indice

<b>1</b>	<b>Introduzione</b>	<b>5</b>
<b>2</b>	<b>Analisi del sistema</b>	<b>7</b>
2.1	Il modello . . . . .	7
2.2	Parametri del modello . . . . .	7
2.3	Equazioni del moto e funzioni di trasferimento . . . . .	8
2.4	Specifiche di controllo . . . . .	10
<b>3</b>	<b>Analisi del sistema in catena aperta</b>	<b>11</b>
3.1	Risposta in catena aperta . . . . .	11
3.2	Implementazione Simulink . . . . .	13
3.3	Risultati . . . . .	15
<b>4</b>	<b>Controllo in catena chiusa</b>	<b>17</b>
4.1	Luogo delle radici . . . . .	19
4.2	Diagrammi di Bode . . . . .	24
4.3	PID . . . . .	31
4.4	Implementazione Simulink . . . . .	33
<b>5</b>	<b>Osservazioni conclusive</b>	<b>37</b>
	<b>Bibliografia</b>	<b>39</b>



# Capitolo 1

## Introduzione

Gli autoveicoli sono dotati di un sistema di sospensioni che, oltre a sorreggere lo chassis del veicolo, deve isolarlo dalle irregolarità del terreno per migliorare il comfort di marcia. Le sospensioni tendono a limitare le accelerazioni verticali della massa sospesa quando il veicolo transita su una sede stradale che presenta delle irregolarità.

Molti veicoli impiegano sospensioni di tipo passivo, che sono le più semplici ed economiche e per questo sono le più usate nei veicoli commerciali. Sono composte da un ammortizzatore (molla  $K_1$ ) e uno smorzatore viscoso ( $b_1$ ) in parallelo i cui parametri sono fissi e scelti dalla casa costruttrice per ottenere un valore di compromesso tra le esigenze di comfort di marcia e quelle associate alla manovrabilità del veicolo (handling). Sospensioni troppo “morbide” migliorano il comfort in quanto si deformano molto rapidamente assorbendo (e quindi compensando) le asperità e le brusche variazioni di quota della sede stradale, ma rischiano di ridurre la tenuta di strada a causa delle ampie oscillazioni verticali del veicolo e delle conseguenti ampie fluttuazioni della forza di contatto tra pneumatico e strada. Viceversa una taratura troppo rigida garantisce una migliore aderenza ma provoca un aumento delle sollecitazioni verticali sul telaio del veicolo (basso comfort di marcia). Si vorrebbe simultaneamente che le sospensioni fossero “morbide” verso le asperità del terreno, e “rigide” nei confronti di forze esterne e/o inerziali. Appare chiaro come siano due specifiche contrastanti. Una sintesi efficace si traduce pertanto nell’identificazione di un buon compromesso tra le due specifiche.

Le prestazioni ottenibili con sospensioni passive sono limitate dalla semplicità dei dispositivi di attuazione e dall’insufficiente numero di gradi di libertà progettuali. Per avere prestazioni migliori si utilizzano tipologie alternative di sospensioni: le sospensioni attive. Nelle sospensioni attive, oltre alla molla e allo smorzatore vi è in più un terzo elemento, un attuatore in grado di generare una forza interna  $F(t)$  variabile nel tempo tra la cassa del veicolo e

la ruota. Tali sistemi consentono, attraverso una opportuna “modulazione” di tale forza, di stabilizzare il movimento e ottenere prestazioni nettamente superiori a quelle di un sistema passivo.

Il problema di come “calibrare” tale forza sulla base delle misure disponibili è il problema di controllo che verrà trattato in questo elaborato. Lo studio si concentrerà nell’analisi in ambiente MATLAB del sistema sospensione di un veicolo dove i parametri del sistema saranno all’incirca quelli di un bus. Dopo aver definito il problema con il suo modello matematico, si passerà all’analisi del sistema in catena aperta dove l’ingresso del sistema è deciso a priori e non viene modificato a seconda del valore reale dell’uscita. In questo capitolo verranno studiate le risposte indiciali e l’analisi armonica del sistema. Nel capitolo successivo verrà studiato il controllo in catena chiusa del sistema dove il controllo sarà effettuato comparando istante per istante l’ingresso del sistema con l’uscita. In questo modo l’uscita del sistema sarà dipendente sia dagli ingressi che dall’uscita stessa e, come si vedrà, si otterranno risultati migliori rispetto al controllo in catena aperta. Il controllore sarà progettato con diversi metodi: PID, luogo delle radici e reti correttive.

Il sistema verrà simulato attraverso lo strumento Simulink: prima in catena aperta e successivamente con un controllore PID.

# Capitolo 2

## Analisi del sistema

### 2.1 Il modello

Nella creazione del modello si suppone che la sospensione debba sostenere 1/4 del peso del veicolo (Body Mass).

Nella figura 2.1 si vede come la sospensione attiva agisce tramite la forza  $u$ .

Si noti che nel modello si suppone nullo il peso della ruota: quest'approssimazione influisce minimamente sulla fedeltà del modello al sistema originale.

### 2.2 Parametri del modello

I parametri sono dimensionati per la sospensione di un bus quindi la massa  $M1$  sarà 1/4 della massa totale del mezzo.

- Body mass ( $M_1$ ) = 2500 kg
- Suspension mass ( $M_2$ ) = 320 kg
- costante della molla nella sospensione ( $K_1$ ) = 80 000 N/m
- costante della molla nella ruota e nel pneumatico ( $K_2$ ) = 500 000 N/m
- costante di attrito del sistema di sospensione ( $b_1$ ) = 350 Ns/m
- costante di attrito di ruota e pneumatico ( $b_2$ ) = 15 020 Ns/m
- forza di controllo ( $u$ ) = forza applicata dal controllore da progettare

### Model of Bus Suspension System (1/4 Bus)

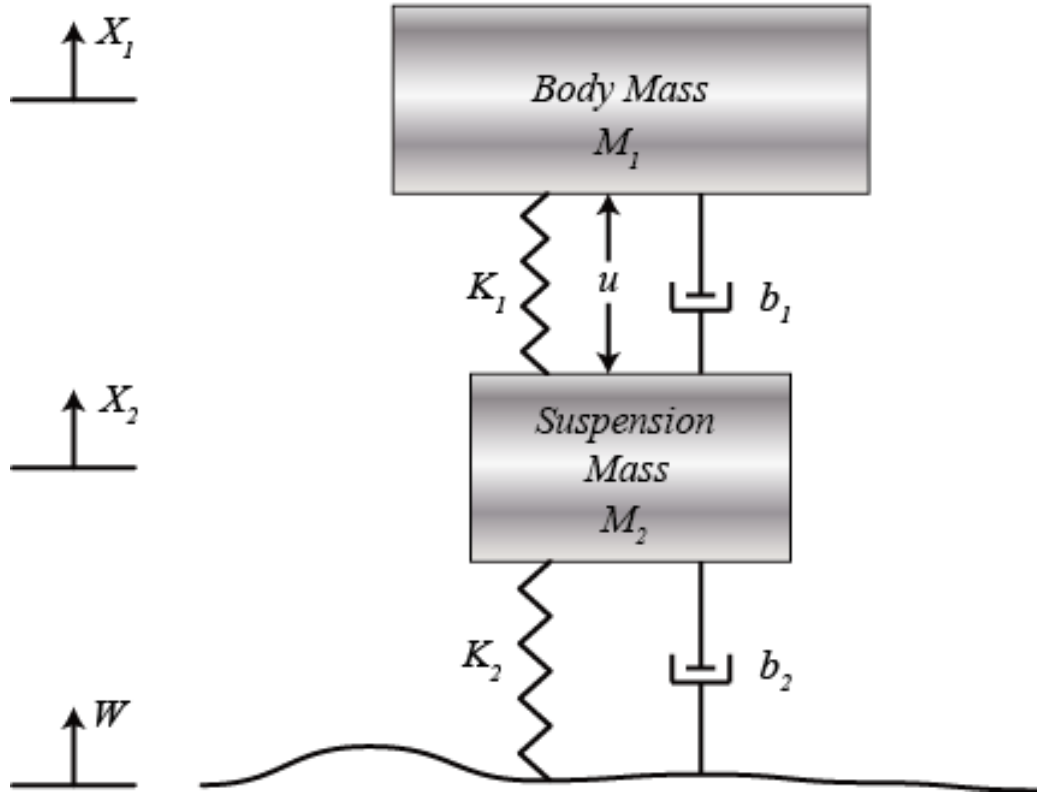


Figura 2.1: Modello della sospensione attiva (1/4)

## 2.3 Equazioni del moto e funzioni di trasferimento

Dalla figura 2.1 e dalla legge di Newton si trovano le equazioni del moto:

$$M_1 \ddot{X}_1 = -b_1(\dot{X}_1 - \dot{X}_2) - K_1(X_1 - X_2) + U \quad (2.1)$$

$$M_2 \ddot{X}_2 = b_1(\dot{X}_1 - \dot{X}_2) + K_1(X_1 - X_2) + b_2(\dot{W} - \dot{X}_2) + K_2(W - X_2) - U \quad (2.2)$$

Supponendo condizioni iniziali nulle, si possono scrivere le equazioni che descrivono il funzionamento del sistema quando l'autobus incontra una bu-



ca. Le equazioni sono espresse nella forma della trasformata di Laplace.

$$(M_1s^2 + b_1s + K_1)X_1(s) - (b_1s + K_1)X_2(s) = U(s) \quad (2.3)$$

$$(b_1s + K_1)X_1(s) + (M_2s^2 + (b_1 + b_2)s + (K_1 + K_2))X_2(s) = (b_2s + K_2)W(s) - U(s) \quad (2.4)$$

$$\begin{bmatrix} (M_1s^2 + b_1s + K_1) & -(b_1s + K_1) \\ -(b_1s + K_1) & (M_2s^2 + (b_1 + b_2)s + (K_1 + K_2)) \end{bmatrix} \begin{bmatrix} X_1(s) \\ X_2(s) \end{bmatrix} = \begin{bmatrix} U(s) \\ (b_2s + K_2)W(s) - U(s) \end{bmatrix} \quad (2.5)$$

$$A = \begin{bmatrix} (M_1s^2 + b_1s + K_1) & -(b_1s + K_1) \\ -(b_1s + K_1) & (M_2s^2 + (b_1 + b_2)s + (K_1 + K_2)) \end{bmatrix} \quad (2.6)$$

$$\begin{aligned} \Delta &= \det \begin{bmatrix} (M_1s^2 + b_1s + K_1) & -(b_1s + K_1) \\ -(b_1s + K_1) & (M_2s^2 + (b_1 + b_2)s + (K_1 + K_2)) \end{bmatrix} \\ &= (M_1s^2 + b_1s + K_1)(M_2s^2 + (b_1 + b_2)s + (K_1 + K_2)) - (b_1s + K_1)(b_1s + K_1) \end{aligned} \quad (2.7)$$

Ora calcolando  $A^{-1}$  si trova:

$$\begin{bmatrix} X_1(s) \\ X_2(s) \end{bmatrix} = \frac{1}{\Delta} \begin{bmatrix} (M_2s^2 + (b_1 + b_2)s + (K_1 + K_2)) & (b_1s + K_1) \\ (b_1s + K_1) & (M_1s^2 + b_1s + K_1) \end{bmatrix} \begin{bmatrix} U(s) \\ (b_2s + K_2)W(s) - U(s) \end{bmatrix} \quad (2.8)$$

Ora evidenziando gli ingressi  $U(s)$  e  $W(s)$  si trova:

$$\begin{aligned} & \begin{bmatrix} (M_2s^2 + b_2s + K_2) & (b_1b_2s^2 + (b_1K_2 + b_2K_1)s + K_1K_2) \\ -M_1s^2 & (M_1b_2s^3 + (M_1K_2 + b_1b_2)s^2 + (b_1K_2 + b_2K_1)s + K_1K_2) \end{bmatrix} \begin{bmatrix} X_1(s) \\ X_2(s) \end{bmatrix} = \frac{1}{\Delta} \\ & \begin{bmatrix} U(s) \\ W(s) \end{bmatrix} \end{aligned} \quad (2.9)$$

Posto  $W(s) = 0$  si calcola la funzione di trasferimento  $G_1(s)$  relativa al solo ingresso  $U(s)$ :

$$G_1(s) = \frac{X_1(s) - X_2(s)}{U(s)} = \frac{(M_1 + M_2)s^2 + b_2s + K_2}{\Delta} \quad (2.10)$$

Se si considera solo il disturbo  $W(s)$  allora ponendo  $U(s) = 0$  si ha:

$$G_2(s) = \frac{X_1(s) - X_2(s)}{W(s)} = \frac{-M_1b_2s^3 - M_1K_2s^2}{\Delta} \quad (2.11)$$

## 2.4 Specifiche di controllo

Un sistema di sospensioni deve assicurare un buon comfort al mezzo in presenza buche o dossi sulla strada limitando al minimo le oscillazioni.

Poichè la distanza  $X_1 - W$  è difficile da misurare e la deformazione della ruota  $X_2 - W$  è trascurabile, si preferisce usare come output la distanza  $X_1 - X_2$ . Il disturbo stradale  $W$  verrà simulato da un segnale a gradino, che rappresenta il momento in cui la ruota del bus esce da una buca. Si vuole progettare un controllore in retroazione che rispetti le seguenti specifiche:

- sovraelongazione inferiore al 5%.
- tempo di assestamento inferiore a 5 secondi.

# Capitolo 3

## Analisi del sistema in catena aperta

### 3.1 Risposta in catena aperta

Per valutare la risposta in catena aperta si utilizza lo schema in figura 3.1. In questo caso l'ingresso  $u(t)$  è deciso a priori e non viene modificato a seconda del valore reale di  $y(t)$ .

Verrà calcolata la risposta al gradino delle due funzioni di trasferimento  $G_1(s)$  e  $G_2(s)$  per provare che con questo schema il sistema non rispetta le specifiche di progetto.

La funzione di trasferimento  $G_1(s)$  rappresenta il sistema sospensione che non incontra buche o dossi sulla strada:  $W = 0$ . Di seguito il codice MATLAB utilizzato per visualizzare il grafico della risposta in catena aperta.

---

```
M1 = 2500;  
M2 = 320;  
K1 = 80000;  
K2 = 500000;  
b1 = 350;  
b2 = 15020;  
  
s = tf('s');  
G1 = ((M1+M2)*s^2+b2*s+K2)/((M1*s^2+b1*s+K1)*(M2*s^2+(b1+b2)  
    *s+(K1+K2))-(b1*s+K1)*(b1*s+K1));  
  
step(100*G1)
```

---

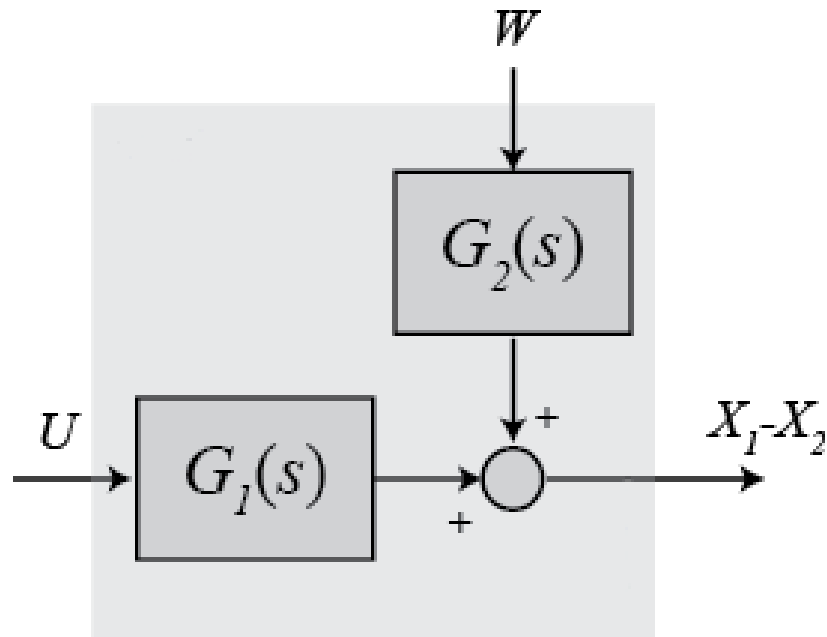


Figura 3.1: Schema controllo in catena aperta

Inizialmente vengono definiti i parametri del sistema. La funzione `tf()` crea una funzione di trasferimento che, in questo caso, serve per comporre  $G_1(s)$  che viene definita nell'istruzione successiva. La funzione `step()` calcola e visualizza la risposta del sistema al gradino. Per rendere il calcolo più realistico, la risposta al gradino di  $G_1(s)$  viene calcolata con un gradino di altezza 100 (Newton). Il risultato è visualizzato in figura 3.2.

Dal grafico 3.2 si vede che il sistema non è abbastanza smorzato infatti, anche se la sovraelongazione è accettabile, il tempo di assestamento va ben oltre le specifiche del sistema.

La funzione di trasferimento  $G_2(s)$  rappresenta il sistema sospensione passiva in cui la forza  $u$  di controllo è nulla. L'ingresso del sistema è, quindi, solo il disturbo  $W$ . Analogamente a quanto fatto per  $G_1(s)$ , la risposta al gradino della funzione  $G_2(s)$  si calcola con il codice:

---


$$G2 = (-M1*b2*s^3 - M1*K2*s^2) / ((M1*s^2 + b1*s + K1) * (M2*s^2 + (b1 + b2)*s + (K1 + K2)) - (b1*s + K1) * (b1*s + K1));$$

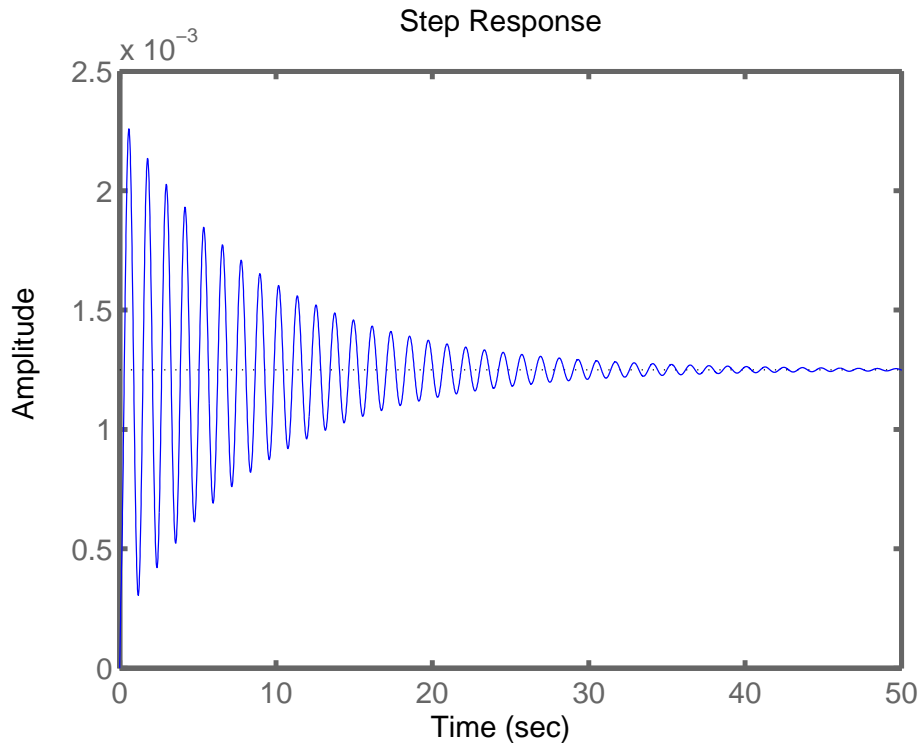


Figura 3.2: Risposta in catena aperta  $100 * G_1(s)$

```
step(0.1*G2)
```

---

Con l'istruzione `step(0.1*G2)` viene simulato un gradino di 10 centimetri. Anche in questo caso (figura 3.3), la sovranelongazione e il tempo di assestamento vanno ben oltre le specifiche di progetto.

## 3.2 Implementazione Simulink

Con Simulink, è molto semplice rappresentare e simulare un modello matematico che rappresenta un sistema fisico. Uno dei principali vantaggi di Simulink (e la simulazione in generale) per l'analisi dei sistemi dinamici è che permette di analizzare rapidamente la risposta dei sistemi complicati che possono essere difficili da analizzare analiticamente. Simulink è in grado di approssimare numericamente le soluzioni dei modelli matematici che

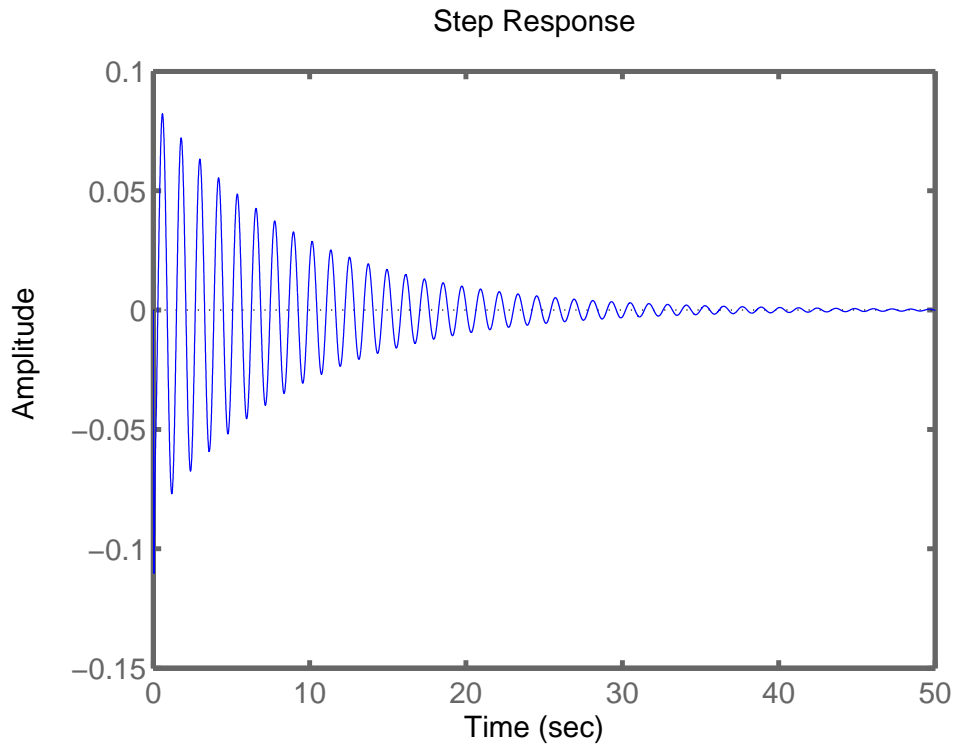


Figura 3.3: Risposta in catena aperta  $G_2(s)$

non siamo in grado, o non si vogliono, risolvere a mano. Il sistema viene modellizzato sommando le forze che agiscono su entrambe le masse (corpo e sospensione) e integrando le accelerazioni di ciascuna massa due volte per ottenere velocità e posizioni istantanee.

$$\int \int \frac{d^2 x_1}{dt^2} dt = \int \frac{dx_1}{dt} dt = x_1$$

$$\int \int \frac{d^2 x_2}{dt^2} dt = \int \frac{dx_2}{dt} dt = x_2$$

I coefficienti delle equazioni del modello vengono modellizzati con blocchi “gain” e i vari termini sommati con blocchi sommatore.

In figura 3.4 il modello Simulink dello schema di figura 3.1.

Lo schema in figura 3.4 rappresenta il modello in catena aperta del sistema. La risposta al gradino della fdt  $G_1(s)$  può essere calcolata imponendo il

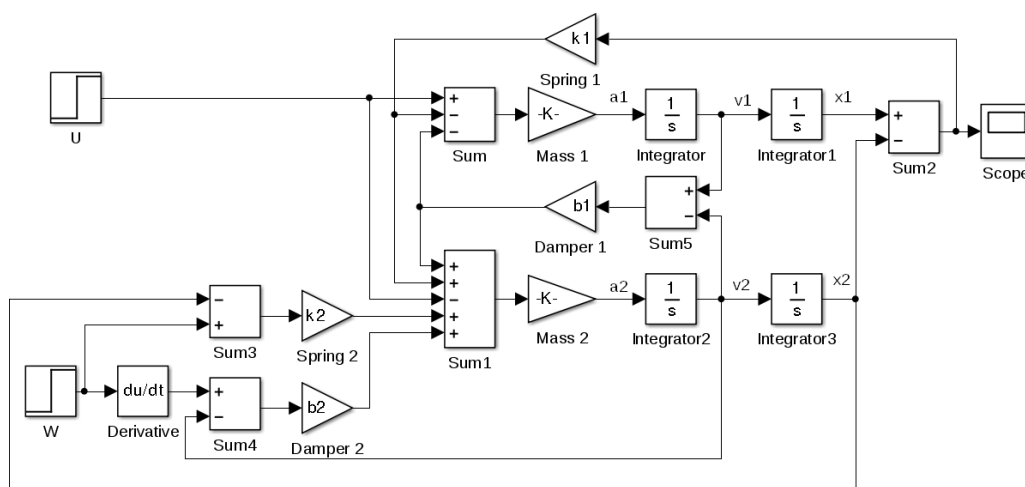


Figura 3.4: Schema Simulink del modello sospensione

segnale  $W = 0$  e impostando il valore finale di  $U$  pari a 100 per simulare lo stesso gradino del paragrafo precedente (più realistico).

Prima però, si definiscono i valori dei parametri del sistema nel prompt MATLAB:

---

```

m1 = 2500;
m2 = 320;
k1 = 80000;
k2 = 500000;
b1 = 350;
b2 = 15020;

```

---

La risposta al gradino può essere facilmente visualizzata facendo partire la simulazione.

Dal grafico in figura 3.5 si nota la somiglianza con la risposta al gradino calcolata nei paragrafi precedenti. Per controllare il sistema sarà necessario un controllore in catena chiusa che verrà sviluppato nei capitoli successivi.

### 3.3 Risultati

Dai calcoli eseguiti risulta evidente che il sistema in catena aperta non ha prestazioni sufficienti a soddisfare le specifiche richieste.

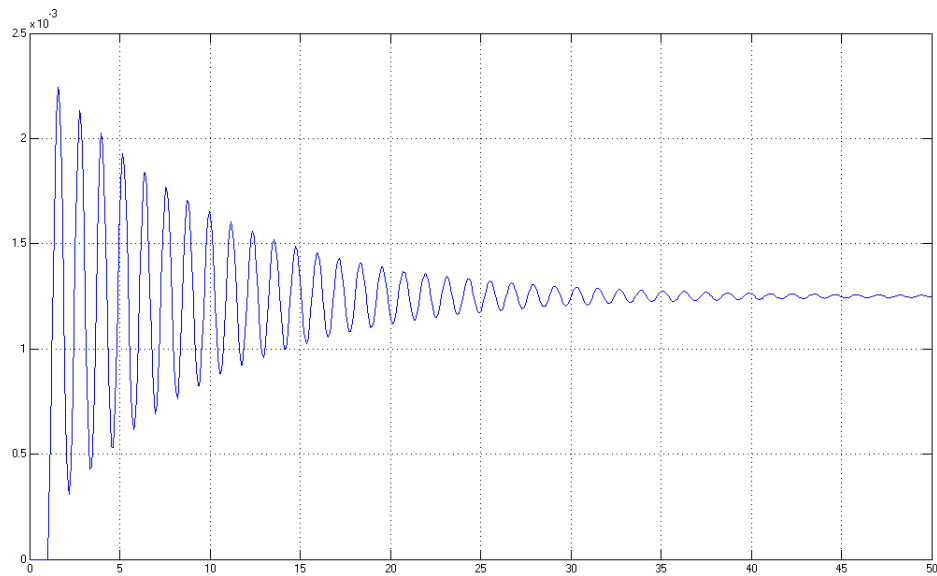


Figura 3.5: Simulink: risposta al gradino ( $100G_1(s)$ ) in catena aperta

Se l'output del sistema non è quello desiderato, il sistema di controllo non può in alcun modo accorgersi dell'errore, né tanto meno modificare l'input del sistema. In questo modo la garanzia del risultato è affidata alla precisione del modello matematico ed all'assenza di disturbi. Nei capitoli successivi si cercherà il metodo migliore per controllare il sistema ricorrendo a schemi in catena chiusa.



# Capitolo 4

## Controllo in catena chiusa

Nel controllo in catena chiusa il controllo viene effettuato comparando istante per istante l'ingresso con l'uscita del sistema quindi nel nostro caso (figura 4.1), l'ingresso del controllore sarà  $e = r - (X_1 - X_2) = X_2 - X_1$ . Il controllore, quindi, dovrà gestire la forza  $U$  che va ad agire sulla sospensione.

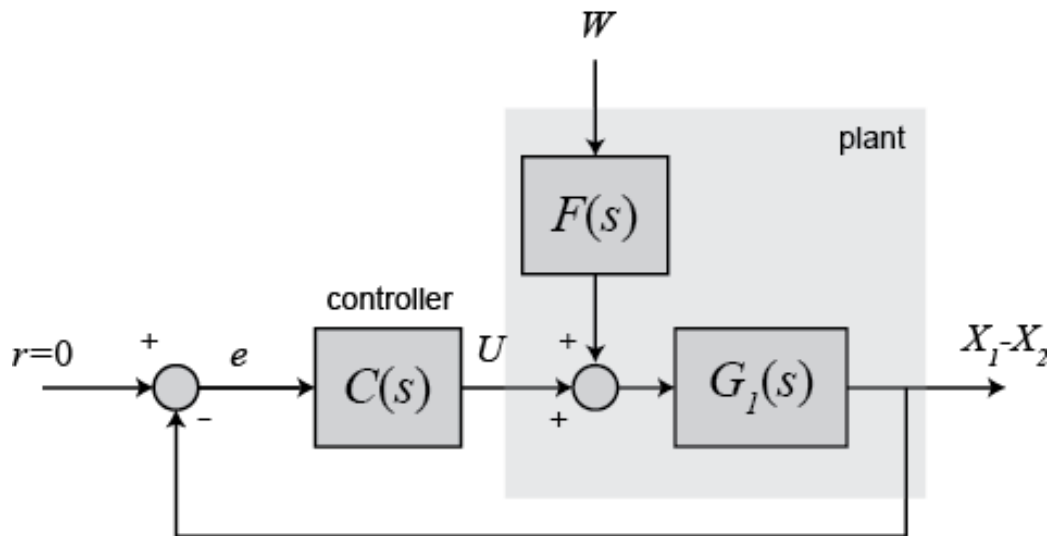


Figura 4.1: Schema controllo in catena chiusa

Il controllore sarà, quindi, un sistema che a partire da un errore presente tra  $r(t)$  e  $y(t)$  genera il segnale di controllo  $u(t)$  in modo tale da compensare l'errore. Dalla figura 4.1, con semplici calcoli, si vede che  $F(s)G_1(s) = G_2(s)$

quindi

$$F(s) = \frac{G_2(s)}{G_1(s)} \quad (4.1)$$

Il codice MATLAB per creare  $F(s)$  è il seguente:

---

```

m1 = 2500;
m2 = 320;
k1 = 80000;
k2 = 500000;
b1 = 350;
b2 = 15020;

nump=[(m1+m2) b2 k2];
denp=[(m1*m2) (m1*(b1+b2))+(m2*b1) (m1*(k1+k2))+(m2*k1)+
      +(b1*b2) (b1*k2)+(b2*k1) k1*k2];
G1=tf(nump,denp);

num1=[-(m1*b2) -(m1*k2) 0 0];
den1=[(m1*m2) (m1*(b1+b2))+(m2*b1) (m1*(k1+k2))+(m2*k1)+
      +(b1*b2) (b1*k2)+(b2*k1) k1*k2];
G2=tf(num1,den1);

numf=num1;
denf=nump;
F=tf(numf,denf);

```

---

In questo caso le funzioni di trasferimento vengono create specificando i coefficienti di  $s$ . La funzione `tf()` accetta in input il numeratore e il denominatore della funzione di trasferimento da creare quindi  $F(s)$  sarà composta dal numeratore di  $G_2(s)$  e dal denominatore di  $G_1(s)$ . Sarà quindi:

$$G_1(s) = \frac{nump}{denp}, \quad G_2(s) = \frac{num1}{den1}, \quad F(s) = \frac{num1}{nump} \quad (4.2)$$

La definizione di queste tre fdt sono la base di partenza per la sintesi analitica del controllore che avverrà, con vari metodi, nei paragrafi successivi. I metodi di sintesi utilizzati saranno:

- Luogo delle radici
- Diagrammi di Bode
- PID

## 4.1 Luogo delle radici

Il metodo del luogo delle radici è un metodo grafico che permette di studiare lo spostamento dei poli di una funzione di trasferimento ad anello chiuso al variare di un parametro reale  $K$ . I poli in catena aperta del sistema sono:

---

```
R = roots(denp)
```

```
R =
-23.9758 +35.1869i
-23.9758 -35.1869i
-0.1098 + 5.2504i
-0.1098 - 5.2504i
```

---

La funzione `roots()` calcola, appunto, le radici complesse di *denp*. Con questo metodo di sintesi si cercherà di stimare la risposta in catena chiusa attraverso il luogo delle radici in catena aperta.

I poli dominanti sono  $-0,1098 \pm 5.2504i$ , che sono i più vicini all'asse immaginario. Ora, dopo aver creato le varie fdt, per trovare il grafico si usa il codice:

---

```
rlocus(G1)
z=-log(0.05)/sqrt(pi^2+(log(0.05)^2))
sgrid(z,0)
z =

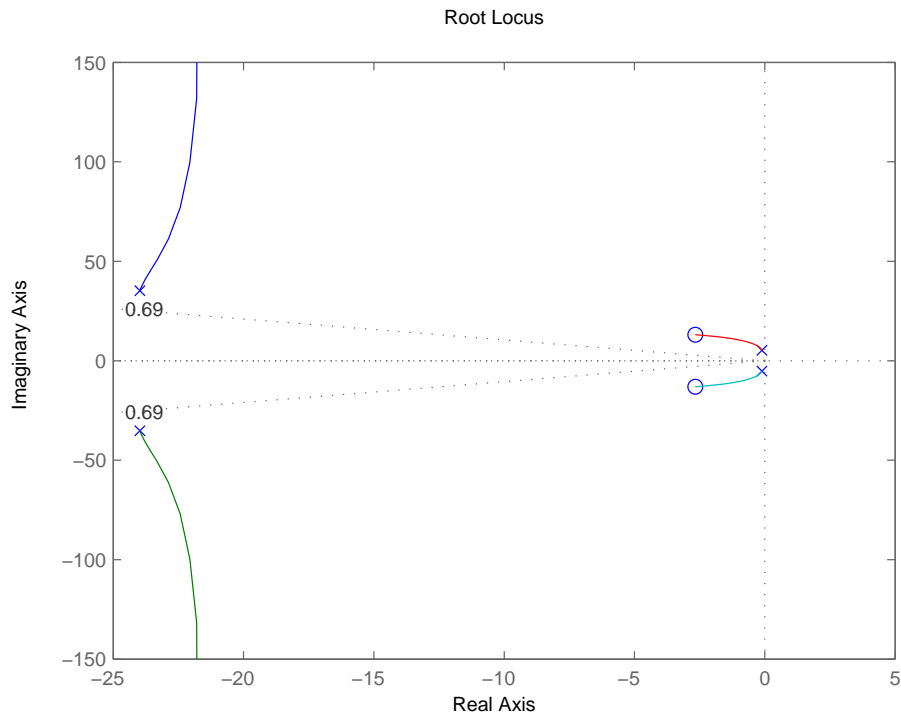
    0.6901
```

---

Il comando `rlocus()` genera il grafico del luogo delle radici (figura 4.2) mentre il comando `sgrid(z,0)` crea le linee tratteggiate che delimitano la porzione di piano d'interesse: all'interno della regione tratteggiata le specifiche di controllo del sistema sono verificate. Questo comando è molto utile perché permette di capire dove è necessario spostare il luogo per soddisfare le richieste progettuali. Il parametro  $z$  (rapporto di smorzamento) in ingresso a `sgrid()` viene calcolato con la formula:

$$\zeta = -\log(5/100)/\sqrt{\pi^2 + \log(5/100)^2}$$

All'interno delle linee tratteggiate si avrà  $\zeta > 0.6901$ . Si vede chiaramente (figura 4.2) che il luogo delle radici non sta nella regione di interesse quindi è necessario un controllore.

Figura 4.2: Luogo delle radici di  $G_1(s)$ 

Ci sono due coppie zero polo vicine all'asse immaginario e questo potrebbe causare problemi di stabilità. Il controllo del sistema consisterà nel aggiungere due zeri vicino ai poli dominanti sull'asse complesso per eliminarli e due poli a sinistra sull'asse reale per spostare il luogo all'interno della regione e per rendere il sistema più pronto. Proviamo a mettere i poli 30 e 60 e due zeri  $2 \pm i$ . La fdt del controllore sarà quindi:

$$C(s) = \frac{s^2 + 4s + 5}{s^2 + 90s + 1800} \quad (4.3)$$

Ora si definisce la fdt in MATLAB:

---

```
z1=2+i;
z2=2-i;
p1=30;
p2=60;
```

```

numc=conv([1 z1],[1 z2]);
denc=conv([1 p1],[1 p2]);
C=tf(numc,denc);

rlocus(C*G1)

```

---

La funzione `conv(A,B)` fa la moltiplicazione dei due polinomi in ingresso forniti con i loro coefficienti mentre la funzione `tf()`, nota, crea la fdt del controllore. Il luogo di  $C(s)G_1(s)$  in figura 4.3.

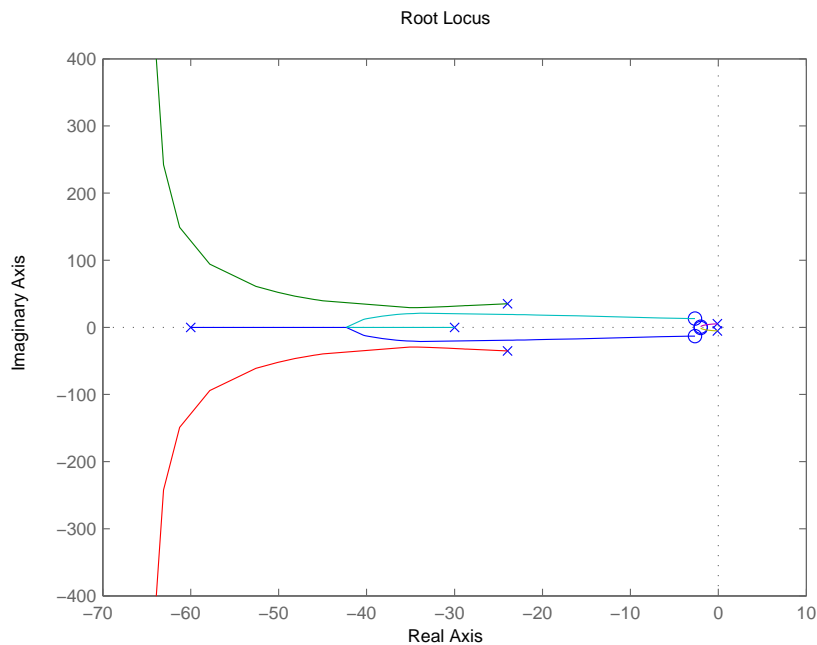


Figura 4.3: Luogo delle radici sistema controllato

Con l'aggiunta del controllore il luogo cambia notevolmente. A questo punto si può fare uno zoom della parte del piano interessata e inserire i delimitatori per vedere se almeno una porzione del luogo sta all'interno della porzione di piano che ci interessa.

---

```
axis([-20 10 -15 15])
```

```
z=-log(0.05)/sqrt(pi^2+(log(0.05)^2))
sgrid(z,0)
```

---

L'istruzione `axis()` definisce la finestra di piano visualizzata: in ingresso vengono forniti gli estremi dell'intervallo di valori per i due assi. Il risultato in figura 4.4 mostra che una parte del luogo delle radici è all'interno della regione che ci interessa.

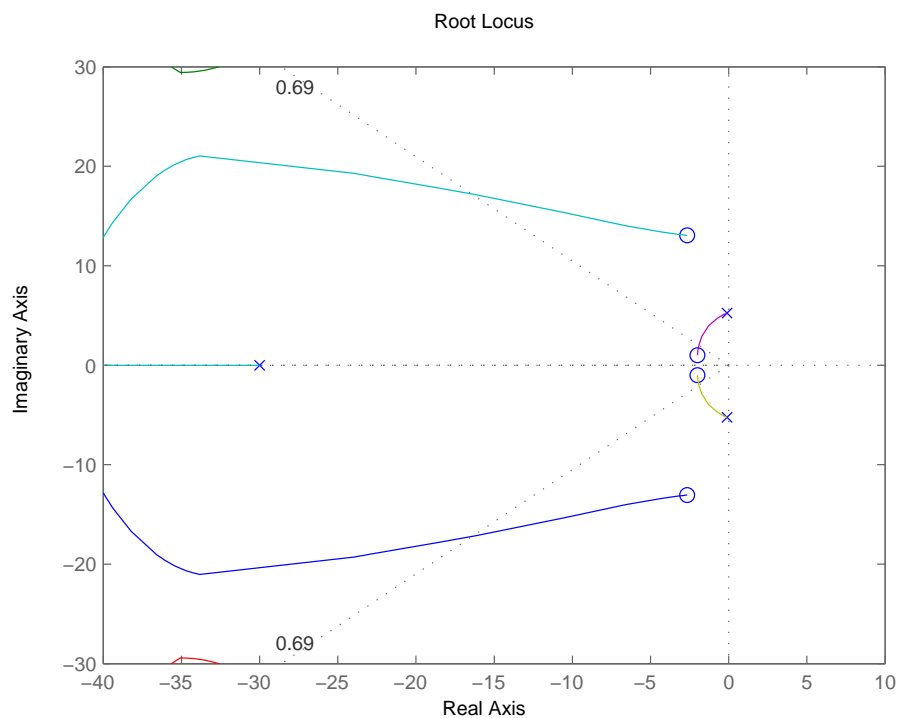


Figura 4.4: Zoom luogo delle radici sistema controllato

L'ultima cosa da fare per avere tutti i parametri del controllore è trovare il valore del parametro  $K$  in grado di soddisfare i requisiti di progettazione. Per un guadagno sufficientemente alto i poli del sistema entrano nella zona di interesse dove le specifiche vengono rispettate. MATLAB permette di scegliere il valore di  $K$  graficamente con la funzione `rlocfind()`.

---

```
[k,poles]=rlocfind(C*G1)
```

---

Dopo aver lanciato l'istruzione in MATLAB è possibile selezionare un punto del luogo e viene restituito il valore di  $K$  corrispondente. In figura 4.5 si vede come i poli dominanti scelti siano dentro la regione di interesse.

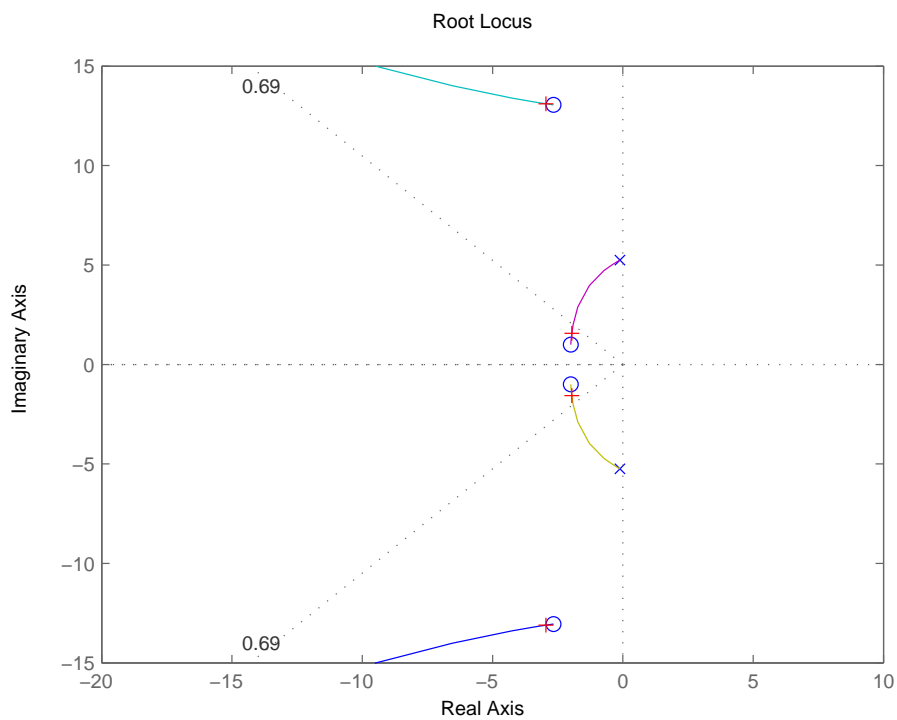


Figura 4.5: Scelta zero nel piano

---

```
selected_point =
```

```
-1.9133 + 1.5399i
```

```
k =
```

```

9.8985e+007

poles =

1.0e+002 *

-0.6419 + 5.9272i
-0.6419 - 5.9272i
-0.0294 + 0.1310i
-0.0294 - 0.1310i
-0.0196 + 0.0155i
-0.0196 - 0.0155i

```

---

Ora il valore di  $K$  trovato verrà utilizzato come guadagno per il compensatore.

```

k = 9.8985e+007;
sys_cl=F*feedback(G1,k*C);
t=0:0.01:2;
step(0.1*sys_cl,t)

```

---

La funzione `feedback(sys1,sys2)` crea la funzione di trasferimento di un sistema con un anello di retroazione (`sys2`). Quindi `sys_cl` è il sistema controllato della figura 4.1. La funzione `step()` visualizza la risposta al gradino che in questo caso è di 0.1m.

Dalla figura 4.6 si vede che il sistema ha una sovraelongazione accettabile e un tempo di assestamento di circa 2 secondi quindi il sistema rispetta le specifiche di progetto.

## 4.2 Diagrammi di Bode

L'idea principale per la progettazione del controllore basato sulla frequenza è di utilizzare il diagramma di Bode del sistema in catena aperta per stimare la funzione di trasferimento in catena chiusa. Dopo aver definito il sistema di figura 4.1 in MATLAB si può disegnare il diagramma di Bode in catena aperta.



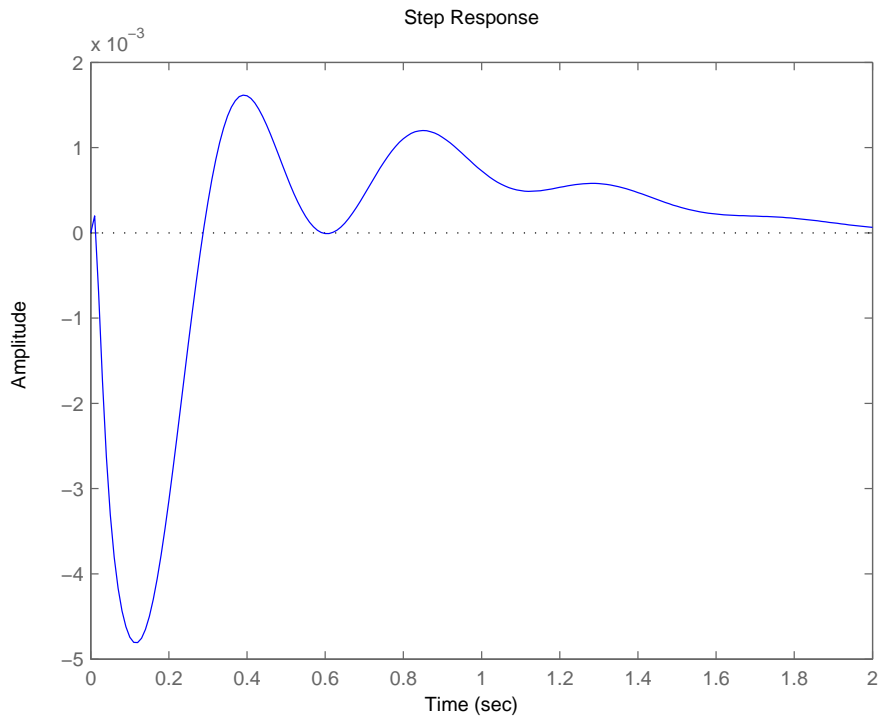


Figura 4.6: Risposta al gradino sistema controllato (luogo delle radici)

---

```
w = logspace(-1,2);
bode(G1,w)
```

---

La funzione `logspace()` genera il piano su scala logaritmica dove la funzione `bode()` disegna il diagramma in figura 4.7

Dal grafico si vede che il diagramma non ha pulsazione di attraversamento necessaria per calcolare il margine di fase. Si può traslare in alto il diagramma del modulo aggiungendo un guadagno  $K$  che poi farà parte del controllore. L'aggiunta di  $K$  è necessaria anche per migliorare l'errore a regime; questo non è una specifica di progetto ma con  $K = 1$  si avrebbero delle oscillazioni a regime inaccettabili. Dalla figura 4.7 si vede che per rendere il guadagno di Bode di  $G_1(s)$  circa unitario è necessario aumentare il guadagno di  $100dB$  e visto che  $1dB = 20 \log K$  sarà  $K = 100000$ :

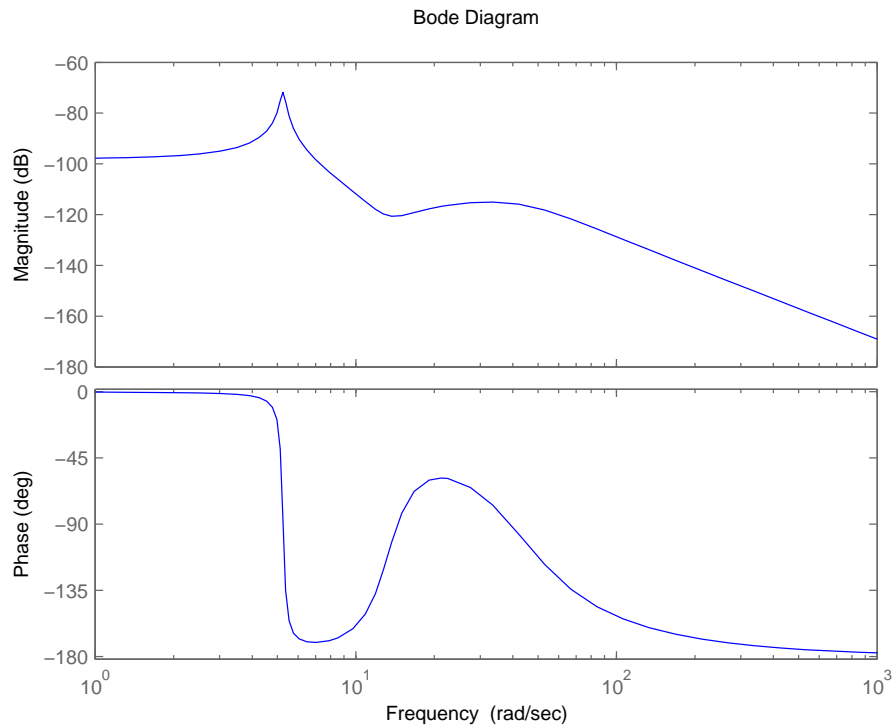


Figura 4.7: Diagramma di Bode di  $G_1(s)$

---

```
K=100000;
margin(K*G1)
```

---

La funzione `margin()` visualizza il diagramma di Bode della fdt in ingresso mettendo in evidenza il margine di fase e la frequenza di attraversamento.

Dalla figura 4.8 si vede che la pulsazione di attraversamento è circa  $\omega_a = 7 \text{ rad/s}$ . Il margine di fase dovrà essere  $m_\varphi = 1 - 0.8S = 0.96 \text{ rad} \simeq 55^\circ$  ( $S = 0.05 = \text{sovravelongazione}$ ). In  $\omega_a$  è necessario aggiungere  $140^\circ$  di fase per minimizzare la sovravelongazione. Il modo più semplice per ottenere ciò è una rete anticipatrice doppia perchè per ogni rete l'anticipo massimo può essere massimo  $90^\circ$ . Il compensatore avrà guadagno  $K$  con una funzione di trasferimento del tipo:

$$C(s) = K \frac{(1 + sT)^2}{(1 + saT)^2} \quad (4.4)$$

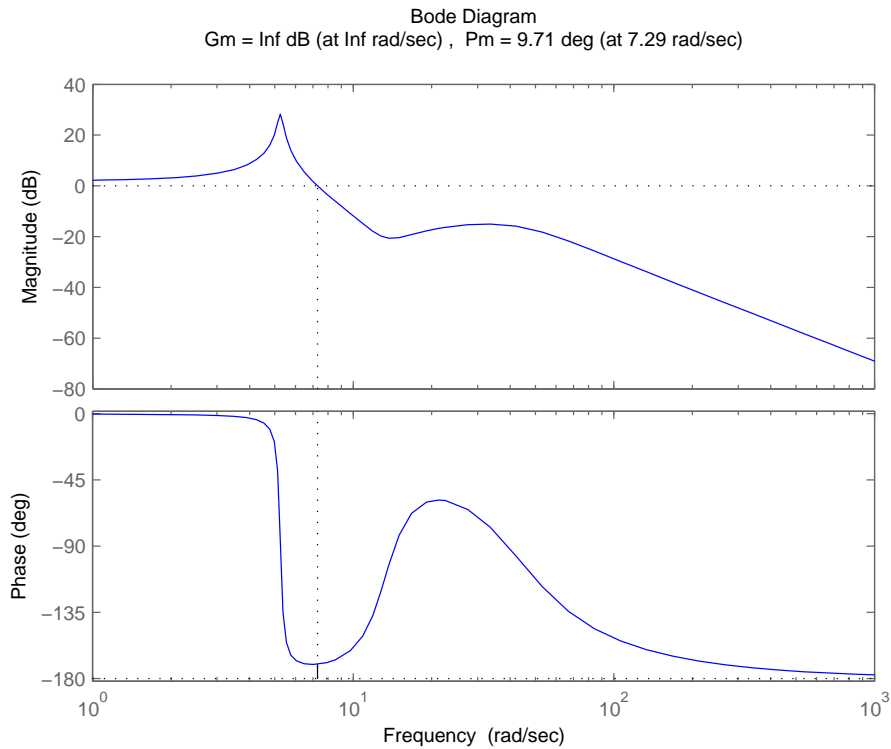


Figura 4.8: Diagramma di Bode di  $KG_1(s)$  con  $K = 100000$

Il parametro  $a$  determina lo spazio tra lo zero e il polo e dalla teoria si ha:

$$a = \frac{1 - \sin 70^\circ}{1 + \sin 70^\circ} = 0.031091 \quad (4.5)$$

Mentre per determinare il punto dello zero e del polo:

$$T = \frac{1}{W\sqrt{a}} = \frac{1}{7\sqrt{0.031091}} = 0.81019 \quad (4.6)$$

$$aT = \frac{\sqrt{a}}{W} = \frac{\sqrt{0.031091}}{7} = 0.02519 \quad (4.7)$$

A questo punto si possono inserire i parametri in MATLAB e visualizzare il risultato:

---

```
a = (1-sin(70/180*pi))/(1+sin(70/180*pi));
w=7;
T=1/(w*sqrt(a));
```

```

aT=sqrt(a)/w;
numc = conv([T 1], [T 1]);
denc = conv([aT 1], [aT 1]);
C = tf(numc,denc);
margin(K*C*G1)

```

---

L'effetto della rete anticipatrice si vede in figura 4.9 dove viene aumentato il margine di fase e la pulsazione di attraversamento.

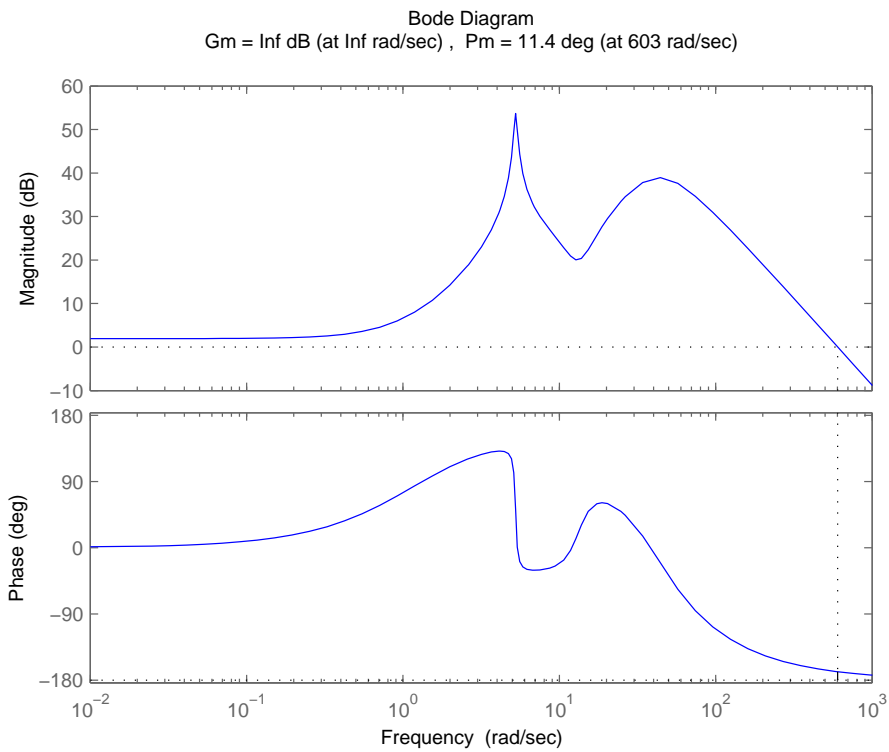


Figura 4.9: Diagramma di Bode di  $KC(s)G(s)$

Ora è possibile definire in MATLAB il sistema completo di figura 4.1 controllato in catena chiusa con il codice:

---

```

sys_cl = F*feedback(G1,K*C);

```

---

A questo punto si visualizza la risposta al gradino del sistema.

---

```
t=0:0.01:5;  
step(0.1*sys_cl,t)  
axis([0 5 -.01 .01])
```

---

Per agevolare la visualizzazione viene modificata la scala del grafico con l'istruzione `axis([0 5 -.01 .01])`.

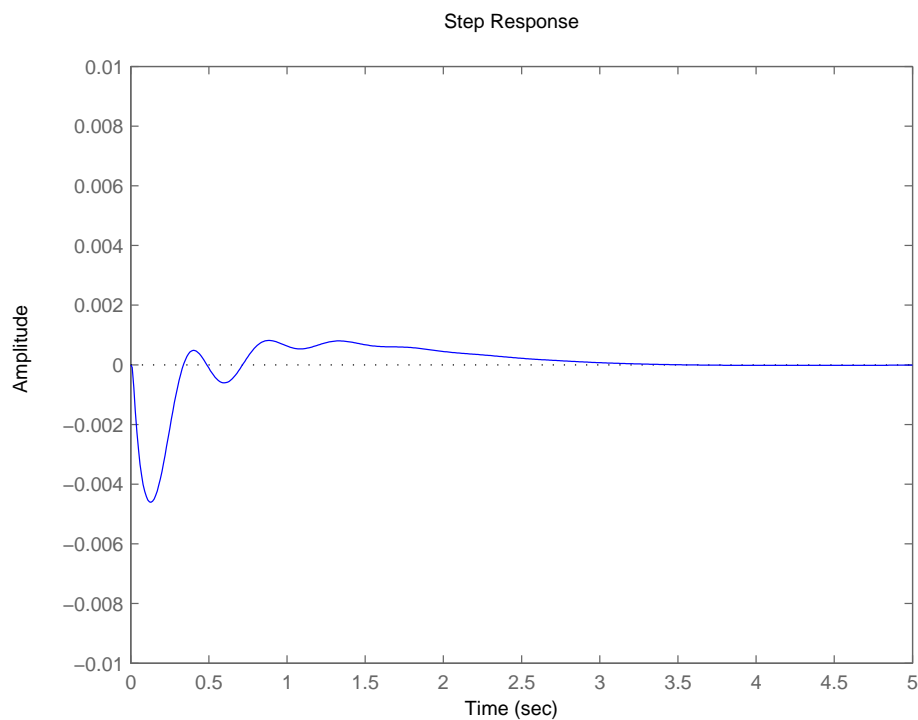


Figura 4.10: Risposta al gradino ( $0.1m$ ) del sistema controllato

Il risultato in figura 4.10 mostra che le specifiche di progettazione vengono soddisfatte infatti si ha una sovralongazione di circa l'1% e un tempo di assestamento di circa 3.5 secondi. Si può avere un ulteriore miglioramento nella risposta al gradino del sistema aumentando il guadagno del controllore: così facendo la pulsazione di attraversamento dovrebbe aumentare rendendo

il sistema più pronto. Con il codice:

---

```

numc = 4*conv([T 1], [T 1]);
denc = conv([aT 1], [aT 1]);
C = tf(numc,denc);
sys_cl = F*feedback(G1,K*C);
t=0:0.01:5;
step(0.1*sys_cl,t)
axis([0 5 -.01 .01])

```

---

In questo modo il guadagno del controllore viene aumentato di un fattore 4 (figura 4.11).

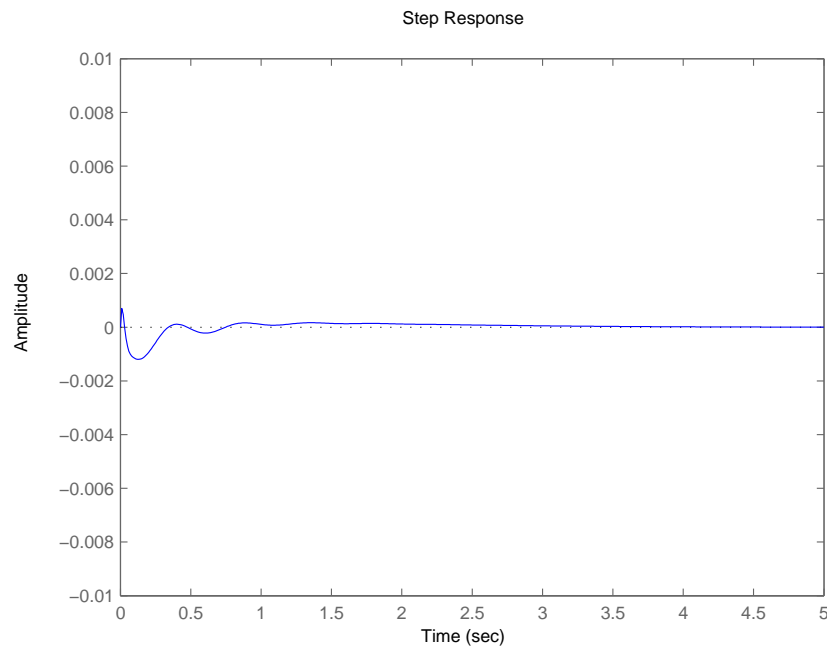


Figura 4.11: Risposta al gradino ( $0.1m$ ) del sistema controllato con guadagno maggiore

Il risultato dell'aumento di guadagno di  $C(s)$  mostra che la sovralongazione si è ridotta rispetto alla precedente configurazione. Si può dire, quindi, che questo sistema (più del precedente) rispetta le specifiche di progetto.

## 4.3 PID

Uno schema ampiamente utilizzato è quello PID, acronimo che indica l'utilizzo combinato di tre funzioni di controllo di tipo Proporzionale, Integrale e Differenziale. La funzione di trasferimento di un controllore PID è del tipo:

$$C(s) = K_p + \frac{K_i}{s} + K_d s = \frac{K_d s^2 + K_p s + K_i}{s} \quad (4.8)$$

In pratica l'uscita di un controllore PID è costituita dalla somma di tre termini; i tre parametri determinano il controllo del sistema. Il contributo dovuto a  $K_p$  è proporzionale al modulo dell'errore e diminuisce man mano che l'errore si avvicina a zero. L'azione integrale è proporzionale all'integrale nel tempo del segnale di errore  $e$  moltiplicato per la costante  $K_I$  e fa sì che il controllore abbia memoria dei valori passati del segnale d'errore riducendo l'errore a regime. L'azione derivativa ( $K_D$ ) cerca di compensare le rapide variazioni del segnale di errore  $e$  attenuandone gli effetti negativi. Siccome non vengono definite specifiche sull'errore a regime si può fare a meno dell'azione integrale realizzando, così, un controllore PD:

$$C(s) = K_p + K_d s \quad (4.9)$$

Questo metodo di sintesi è euristico: si cercherà di “indovinare” i parametri in modo da avere una risposta soddisfacente. Si può quindi valutare subito la risposta del sistema controllato assegnando dei valori a piacere ai parametri:  $K_d = 100000$ ,  $K_p = 100000$ .

---

```
Kd = 100000;
Kp = 100000;
C=tf([Kd Kp],[1]);
sys_cl=F*feedback(F*G1,C);
```

---

Il sistema in retroazione viene creato con la funzione `feedback()` e `tf()` già note dai paragrafi precedenti. A questo punto si può visualizzare la risposta al gradino con il codice:

---

```
t=0:0.05:5;
step(0.1*sys_cl,t)
title('Closed-Loop Response to 0.1-m High Step w/
      PD Controller')
```

---

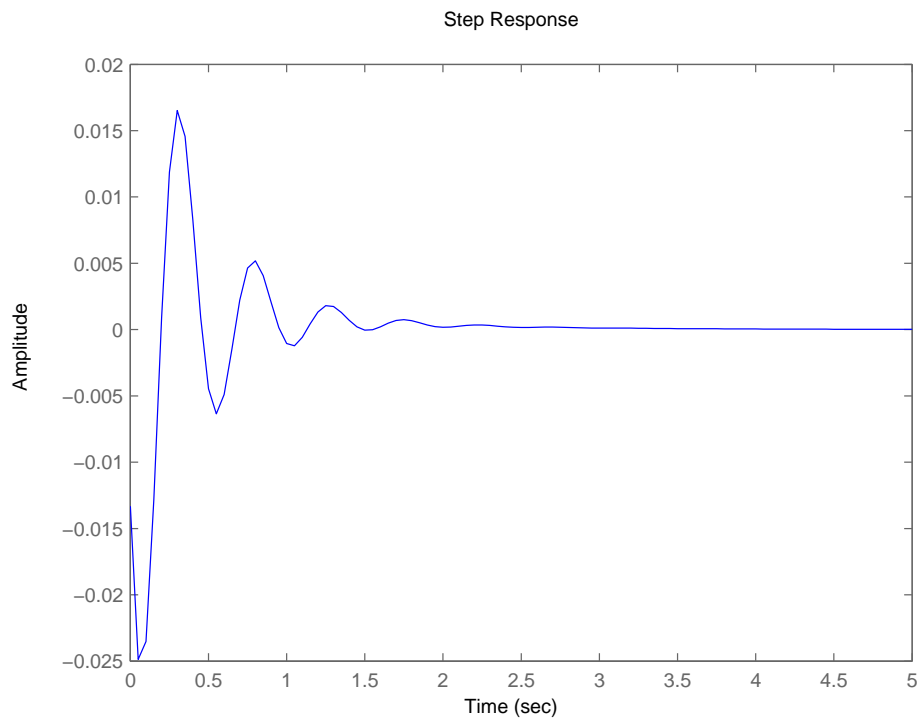


Figura 4.12: Risposta al gradino ( $0.1m$ ) del sistema controllato

Si valuta la risposta in catena chiusa al gradino di  $0.1\text{ m}$  in figura 4.12.

Dalla figura 4.12 si vede che, mentre il tempo di assestamento rispetta le specifiche, la sovravelazione è inaccettabile. Si andrà, quindi, ad agire sui parametri per di  $C(s)$  per aggiustare il grafico. Una soluzione soddisfacente si ha ponendo  $K_p = K_d = 800000$ .

---

```

Kd=800000;
Kp=800000;
C=tf([Kd Kp],[1]);
sys_cl=F*feedback(F*G1,C);
step(0.1*sys_cl,t)
title('Closed-Loop Response to 0.1-m High Step w/
High-Gain PD Controller')

```

---



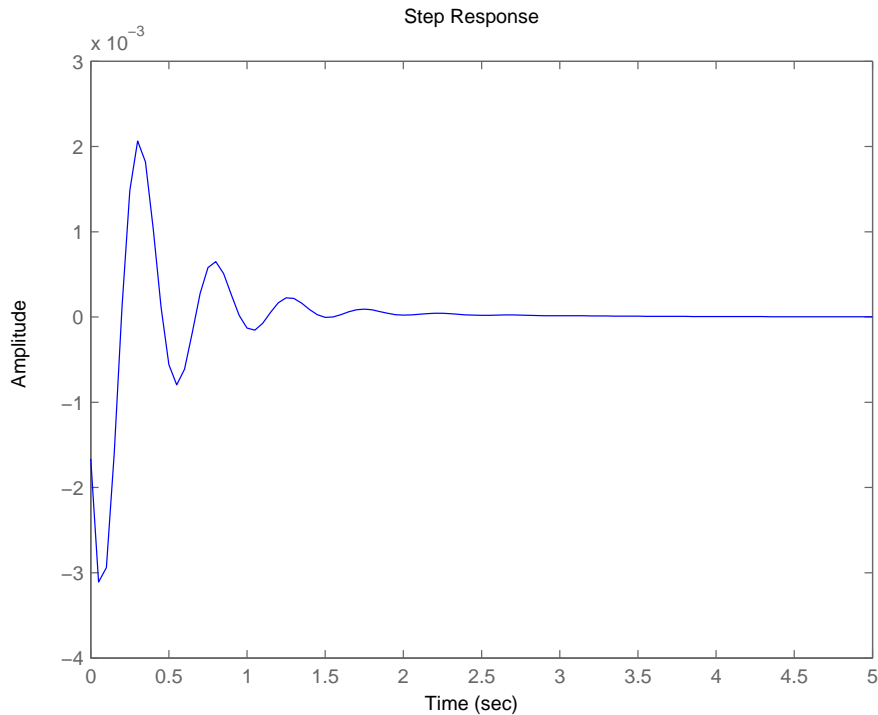


Figura 4.13: Risposta al gradino ( $0.1m$ ) del sistema controllato con  $K_p = K_d = 800000$

Dalla figura 4.13 si vede che il sistema ha un tempo di assestamento di circa  $2.5s$  e una sovralongazione di circa  $2mm$  ampiamente sotto il limite quindi il sistema rispetta le specifiche di progetto.

## 4.4 Implementazione Simulink

Per studiare la risposta in catena chiusa del sistema si deve aggiungere il controllore allo schema Simulink di figura 3.4. Per fare questo si racchiude lo schema di figura 3.4 in un apposito blocco chiamato *subsystem*. Gli ingressi di questo blocco saranno i segnali  $U$  e  $W$  mentre l'uscita coinciderà con quella del sistema. Successivamente si crea lo schema del controllore che in questo caso sarà PID (figura 4.15).

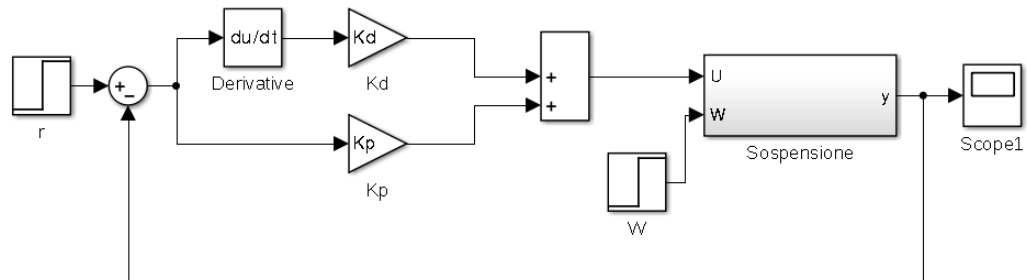


Figura 4.14: Simulink: schema controllo in catena chiusa

Prima di far partire la simulazione è necessario inizializzare i parametri del sistema nel prompt MATLAB: i valori sono quelli trovati nella progettazione del controllore PD nel paragrafo precedente.

---

```
Kd = 800000;
Kp = 800000;
```

---

A questo punto si può far partire la simulazione e visualizzare il risultato attraverso il blocchetto *scope* che visualizza l'andamento del segnale di uscita.

Dalla figura 4.15 si vede che la risposta al gradino coincide con quella trovata utilizzando la funzione di trasferimento del sistema e che rispetta le specifiche di progetto.

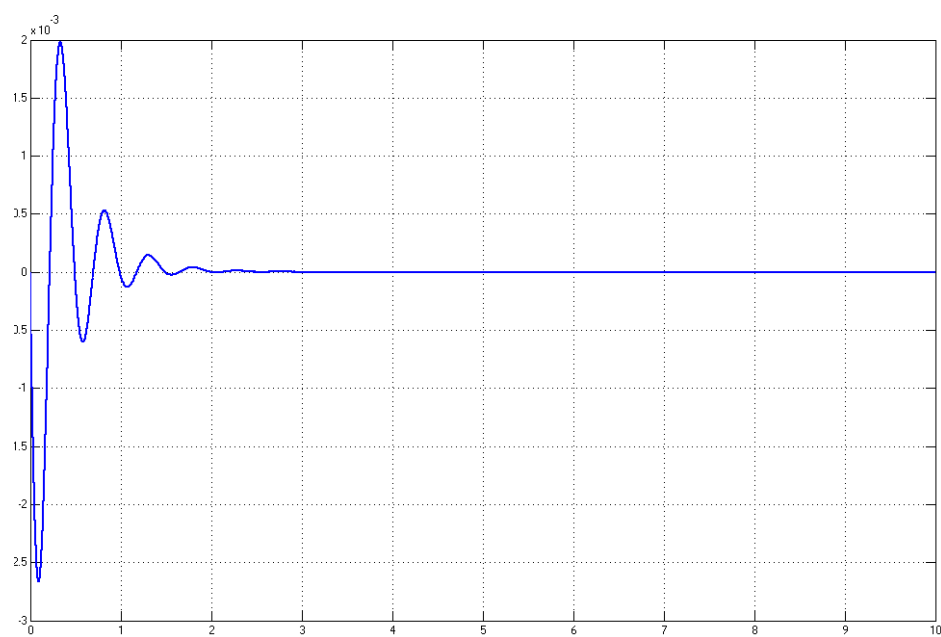


Figura 4.15: Simulink: risposta al gradino del sistema controllato in catena chiusa



# Capitolo 5

## Osservazioni conclusive

In questo elaborato sono stati progettati tre controllori con tre metodi diversi. Tutti rispettano le specifiche di progetto anche se l'unico test di verifica è stata la risposta al gradino. Questo è stato sufficiente per verificare la sovraelongazione e il tempo di assestamento ma per un'analisi più approfondita sarebbe stato utile sottoporre il sistema a test più realistici come ad esempio un disturbo  $W$  casuale. Questo avrebbe dato l'idea di come si comporta il sistema in condizioni reali anche se non è stato l'oggetto di questo elaborato. E' sicuramente stato utile imparare ad utilizzare il tool Simulink che mette a disposizione un'infinità di strumenti di ausilio alla simulazione e progettazione di modelli. Si è visto come la simulazione fatta in MATLAB segua un procedimento molto diverso da quella fatta in Simulink dove è possibile "aggirare" l'utilizzo di equazioni di stato e di funzioni di trasferimento con dei comodi blocchetti. In questo elaborato le funzionalità offerte da Simulink sono state viste solo in parte: il controllore utilizzato nel modello a blocchi (Simulink) è quello (PID) trovato con MATLAB nel paragrafo precedente anche se Simulink ha degli strumenti specifici per la progettazione del controllore che non sono stati trattati in questo elaborato.

In definitiva posso dire che questo lavoro è servito per approfondire argomenti visti solo in teoria nel corso di controlli automatici e potrà servirmi in futuro per approfondire gli studi fatti finora o in ambito lavorativo.



# Bibliografia

- [1] Control Tutorials for Matlab and Simulink.  
(<http://www.engin.umich.edu/class/ctms/>)
- [2] Dispensa di controlli Automatici, Edizioni Libreria Progetto, Padova, 2011