

UNIVERSITÀ DEGLI STUDI DI PADOVA
FACOLTA DI INGEGNERIA

Corso di Laurea in Ingegneria dell'Informazione

**STUDIO IN AMBIENTE MATLAB/SIMULINK
DEL SISTEMA DI CONTROLLO DI UNA
SOSPENSIONE DI UN AUTOBUS**

Relatore:
Prof. Alessandro Beghi

Candidato:
Gioele Sandri

Anno Accademico 2012/2013

Indice :

1.Introduzione:	3
1.1. Obiettivo	3
1.2. Generalità	3
2.Il modello della sospensione	6
2.1. Descrizione	6
2.2. Specifiche di controllo	7
2.3. Equazioni del moto	7
2.4. Funzione di trasferimento	7
3.Controllo in catena aperta	10
3.1. Inizializzazione MatLab	10
3.2. Risposta al gradino.	11
4.Controllo in catena chiusa	14
4.1. PID	14
4.1.1. Implementazione MatLab	16
4.1.2. Implementazione Simulink	19
4.2. Luogo radici	22
4.3. Analisi in Frequenza	27
5.Conclusioni	36

1) Introduzione

1.1) Obiettivo

L'obiettivo di questa tesi è utilizzare le conoscenze apprese durante il corso di controlli automatici per studiare il comportamento di un sistema fisico semplificato: la sospensione di un autobus. All'inizio verranno descritte le equazioni che descrivono il sistema, mentre in seguito si analizzerà la risposta al gradino unitario in catena chiusa del sistema che ci darà importanti informazioni sul sistema. Verranno in seguito descritti 3 metodi di controllo: PID, sintesi in frequenza con i diagrammi di Bode e luogo delle radici. In ognuno dei tre casi si cercherà di arrivare sempre allo stesso risultato: soddisfare le specifiche richieste in partenza. Per dare tutto ciò si è usato il software Matlab (versione R2012a), in particolare Simulink; verranno infatti inseriti notevoli listati di codice spiegando passo dopo passo cosa si sta facendo.

1.2) Generalità

Gli autoveicoli sono dotati di un sistema di sospensioni che, oltre a sorreggere il telaio 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 di costante K) e uno smorzatore viscoso (di costante b) 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), anche in funzione dell'impronta più o meno sportiva che si vuole conferire alla guida del mezzo.

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. Quando inoltre la macchina accelera o decelera, o percorre una curva, si generano sul veicolo delle forze che inducono

movimenti di “pitch” e “roll” (v. Figura 1) al quale le sospensioni offrono poca resistenza.

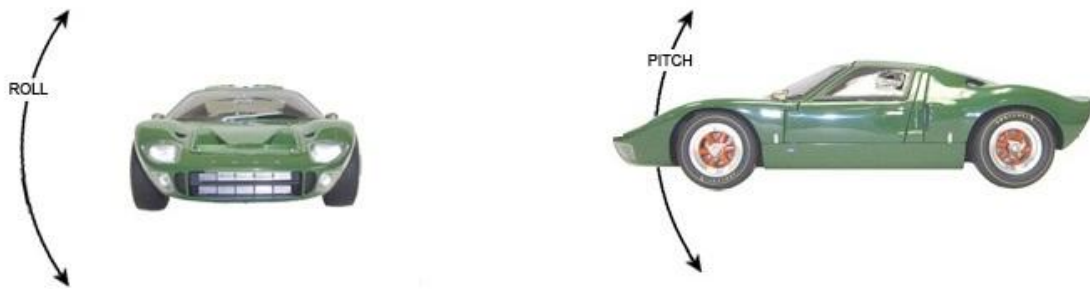


Figura 1: esempio di roll e pitch in un'automobile

Viceversa una taratura troppo rigida garantisce migliore aderenza ma provoca un aumento delle sollecitazioni verticali sulla cassa del veicolo (basso comfort di marcia).

Il problema del miglioramento del comfort di marcia viene come detto valutato in termini della minimizzazione delle accelerazioni verticali della massa. Il problema dell'handling (talvolta indicato come “controllo dell'assetto”) viene espresso sia con riferimento alla minimizzazione delle fluttuazioni della forza di contatto tra pneumatico e strada che con l'obiettivo di mantenere costante la quota rispetto al suolo delle masse sospese, a fronte delle forze, di varia natura (aerodinamica, trasferimenti di carico,...) che agiscono sulla carrozzeria in frenata, in accelerazione e in curva.

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 macchine sportive hanno sospensioni molto rigide, prestanti in termini di assetto e tenuta di strada, ma inutilizzabili nella vita quotidiana. E' chiaro come nelle competizioni sportive l'aspetto delle prestazioni in termini di assetto e stabilità venga chiaramente privilegiato rispetto al comfort di marcia del pilota, essendo l'obiettivo principale vincere la competizione in cui si gareggia.

2) Il modello della sospensione

2.1) Descrizione

Il problema consiste nel controllo del sistema di sospensioni di un autobus. Questo problema, che in generale coinvolgerebbe il sistema nella sua totalità, può essere ridotto allo studio di un'unica sospensione, in modo da renderlo più semplice, semplificazione che da ora in avanti useremo sempre.

Da un punto di vista fisico, il sistema può essere visto come composto da 2 masse, con una molla e un pistone ideali che separano le 2 masse e il terreno tra di loro.

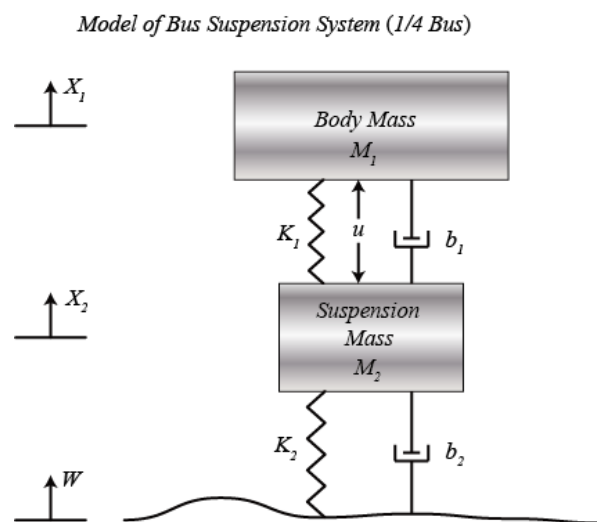


Figura 2: Modello fisico della sospensione di un bus

Dove:

- Body Mass (M_1) = 2500 kg;
- Suspension Mass (M_2) = 320 kg;
- Costante elastica del sistema di sospensione (k_1) = 80,000 N/m;
- Costante elastica di ruota e pneumatico (k_2) = 500,000 N/m;
- Costante di attrito viscoso del sistema di sospensione (b_1) = 350 Ns/m;
- Costante di attrito viscoso di ruota e pneumatico (b_2) = 15,020 Ns/m;
- Forza di controllo (u) = forza applicata dal controllore che dobbiamo progettare.

2.2) Specifiche di Controllo

Un buon sistema di sospensioni richiede che, di fronte ad una qualsiasi imperfezione stradale, il sistema reagisca prontamente riducendo le oscillazioni troppo brusche e che queste vengano dissipate velocemente, assicurando così ai passeggeri a bordo una permanenza durante il viaggio confortevole.

Essendo la distanza X_1-W difficile da misurare, e la deformazione della ruota trascurabile (ossia la distanza X_2-W), si preferisce concentrarsi sulla distanza X_1-X_2 . Il disturbo stradale W sarà simulato con un segnale a gradino (fisicamente rappresenta l'uscita della ruota da una buca) e gli obiettivi sono quelli di realizzare un controllo in retroazione tale per cui il nostro output X_1-X_2 abbia una sovraelongazione inferiore al 5% e un tempo di assestamento minore di 5 secondi.

2.3) Equazioni del Moto

Osservando la Figura 2 e applicando la legge di Newton, si ricavano le seguenti due equazioni che descrivono il sistema:

$$\begin{aligned}M_1\ddot{X}_1 &= -b_1(\dot{X}_1 - \dot{X}_2) - k_1(X_1 - X_2) + U \\M_2\ddot{X}_2 &= b_1(\dot{X}_1 - \dot{X}_2) + k_1(X_1 - X_2) + b_2(\dot{W} - \dot{X}_2) + k_2(W - X_2) - U\end{aligned}$$

2.4) Funzioni di Trasferimento

Assumendo le condizioni iniziali nulle, e applicando la trasformata di Laplace, otteniamo le seguenti equazioni:

$$\begin{aligned}(M_1s^2 + b_1s + k_1)X_1(s) - (b_1s + k_1)X_2(s) &= U(s) \\-(b_1s + k_1)X_1(s) + (M_2s^2 + (b_1 + b_2)s + (k_1 + k_2))X_2(s) &= (b_2s + k_2)W(s) - U(s)\end{aligned}$$

Scritte in forma matriciale diventano:

$$\begin{bmatrix} M_1s^2 + b_1s + k_1 & -(b_1s + k_1) \\ -(b_1s + k_1) & M_2s^2 + (b_1 + b_2)s + (k_1 + k_2) \end{bmatrix} \begin{bmatrix} X_1(s) \\ X_2(s) \end{bmatrix} = \begin{bmatrix} U(s) \\ (b_2s + k_2)W(s) - U(s) \end{bmatrix}$$

$$A = \begin{bmatrix} M_1s^2 + b_1s + k_1 & -(b_1s + k_1) \\ -(b_1s + k_1) & M_2s^2 + (b_1 + b_2)s + (k_1 + k_2) \end{bmatrix}$$

$$\Delta = \det \begin{bmatrix} M_1s^2 + b_1s + k_1 & -(b_1s + k_1) \\ -(b_1s + k_1) & M_2s^2 + (b_1 + b_2)s + (k_1 + k_2) \end{bmatrix}$$

Cioè:

$$\Delta = (M_1s^2 + b_1s + k_1)(M_2s^2 + (b_1 + b_2)s + (k_1 + k_2)) - (b_1s + k_1)(b_1s + k_1)$$

Invertendo A otteniamo la soluzione nel dominio di Laplace:

$$\begin{bmatrix} X_1(s) \\ X_2(s) \end{bmatrix} = \frac{1}{\Delta} \begin{bmatrix} M_2s^2 + (b_1 + b_2)s + (k_1 + k_2) & (b_1s + k_1) \\ (b_1s + k_1) & M_1s^2 + b_1s + k_1 \end{bmatrix} \cdot \begin{bmatrix} U(s) \\ (b_2s + k_2)W(s) - U(s) \end{bmatrix}$$

$$\begin{bmatrix} X_1(s) \\ X_2(s) \end{bmatrix} = \frac{1}{\Delta} \begin{bmatrix} M_2s^2 + b_2s + (k_1 + k_2) & b_1b_2s^2 + (b_1k_2 + b_2k_1)s + k_1k_2 \\ -M_1s^2 & M_1b_2s^3 + (M_1k_2 + b_1b_2)s^2 + (b_1k_2 + b_2k_1)s + k_1k_2 \end{bmatrix} \begin{bmatrix} U(s) \\ W(s) \end{bmatrix}$$

Da quest'ultima rappresentazione si possono infine ricavare le funzioni di trasferimento $G_1(s)$ e $G_2(s)$ per l'uscita $X_1(s)$ – $X_2(s)$ relative agli ingressi $U(s)$ e $W(s)$.

Annullando $W(s)$ otteniamo:

$$G_1(s) = \frac{X_1(s) - X_2(s)}{U(s)} = \frac{(M_1 + M_2)s^2 + b_2s + k_2}{\Delta}$$

Ponendo invece $U(s)=0$ si ricava:

$$G_2(s) = \frac{X_1(s) - X_2(s)}{W(s)} = \frac{-M_1b_2s^3 - M_1k_2s^2}{\Delta}$$

3) Controllo in Catena Aperta

3.1) Inizializzazione Matlab

Andiamo ora a scrivere codice in Matlab, innanzitutto, dopo aver aperto il programma, inseriamo i dati delle masse e le costanti elastiche:

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

Creiamo ora le matrici che rappresentano i coefficienti della variabile s nei numeratori e denominatori delle funzioni di trasferimento $G_1(s)$ e $G_2(s)$

```
num1=[(m1 + m2) b2 k2];  
den1=  
[(m1 * m2) (m1 * (b1 + b2)) + (m2 * b1) (m1 * (k1 + k2)) + (m2 * k1) + (b1 * b2) (b1 * k2) + (b2 * k1) k1 * k2];  
  
num2=[-(m1 * b2) -(m1 * k2) 0 0];  
den2=den1;
```

Ora usando la funzione *tf* creiamo le due funzioni di trasferimento:

```
G1=tf(num1, den1);
```

```
Transfer function:  
                2820 s^2 + 15020 s + 500000  
-----  
800000 s^4 + 3.854e007 s^3 + 1.481e009 s^2 + 1.377e009 s + 4e010
```

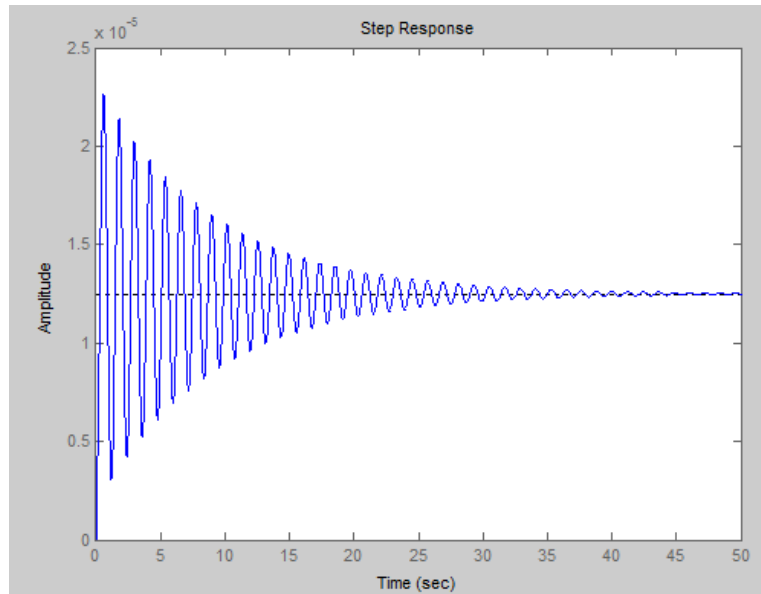
```
G2=tf(num2, den2);
```

```
Transfer function:  
          -3.755e007 s^3 - 1.25e009 s^2  
-----  
800000 s^4 + 3.854e007 s^3 + 1.481e009 s^2 + 1.377e009 s + 4e010
```

3.2) Risposta al Gradino

A questo punto usiamo la funzione *step* che ci permette di calcolare la risposta al gradino delle due trasformate:

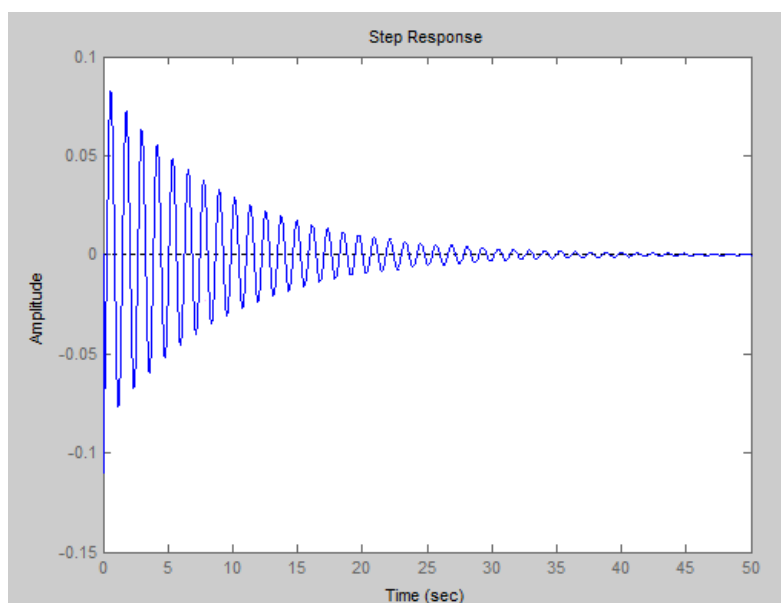
```
step(G1)
```



Come si vede, il sistema ha oscillazioni molto ridotte, ma il tempo di assestamento è molto lungo.

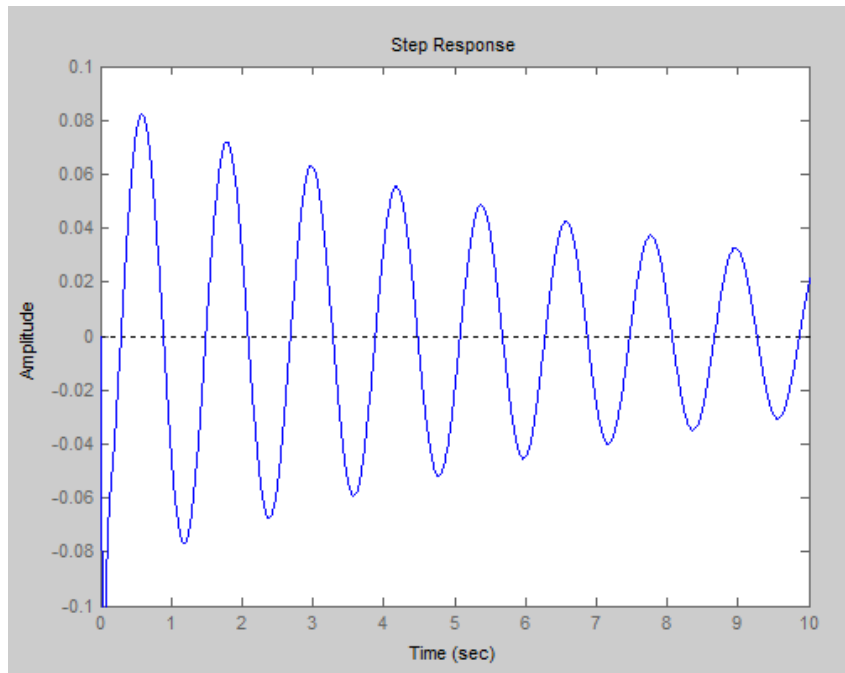
Ben più significativa è la risposta al gradino della seconda trasformata, a cui andiamo ad applicare un disturbo di 0.1m

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



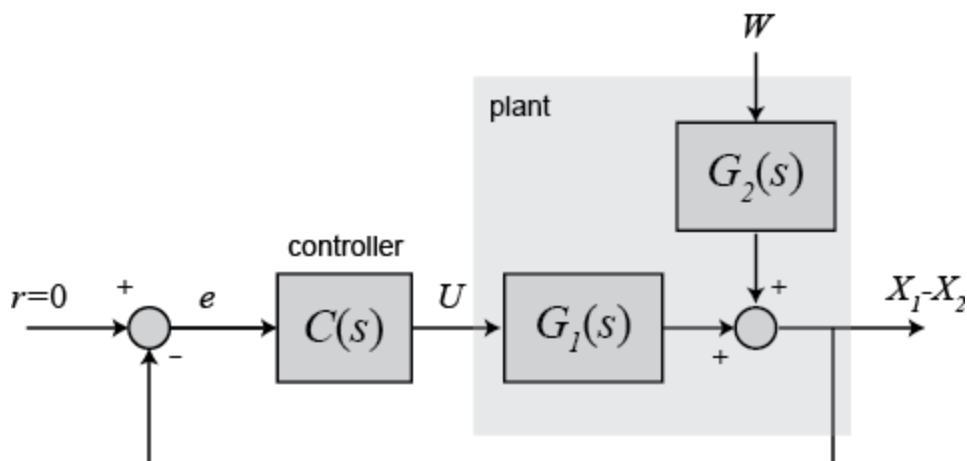
Utilizzando la funzione *axis* andiamo a zoomare su quello che accade nei primi 10 secondi:

```
axis([0 10 -0.1 0.1])
```



Osservando il grafico si nota come le oscillazioni siano molto ampie, toccando un massimo di 13-14 cm, ben al di sopra delle specifiche richieste; inoltre il tempo di assestamento è sempre troppo lungo.

Per risolvere questi problemi, si deve inserire un controllore in retroazione, facendo così diventare il nostro sistema da catena aperta a catena chiusa:



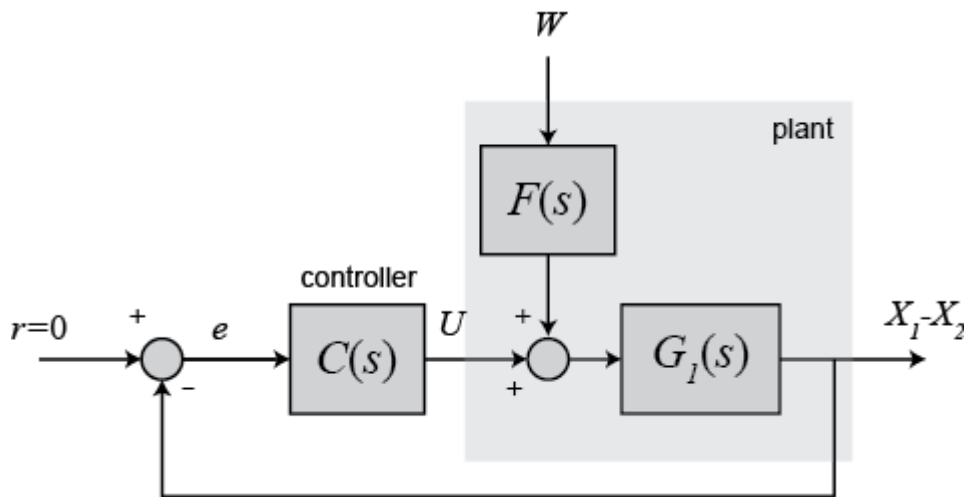
4) Controllo in Catena chiusa

Verranno di seguito discussi 3 metodi per il controllo in catena chiusa:

- Controllore PID
- Luogo delle Radici
- Sintesi in Frequenza (diagrammi di Bode)

dove andremo a definire di volta in volta il controllore $C(s)$, che interviene sull'errore, cioè la differenza tra il riferimento atteso r (ingresso del sistema) e l'uscita effettiva y (output del sistema).

Per tutti i metodi verranno usate le funzioni $G_1(s)$ e $G_2(s)$ calcolate in precedenza, e il seguente schema a blocchi:

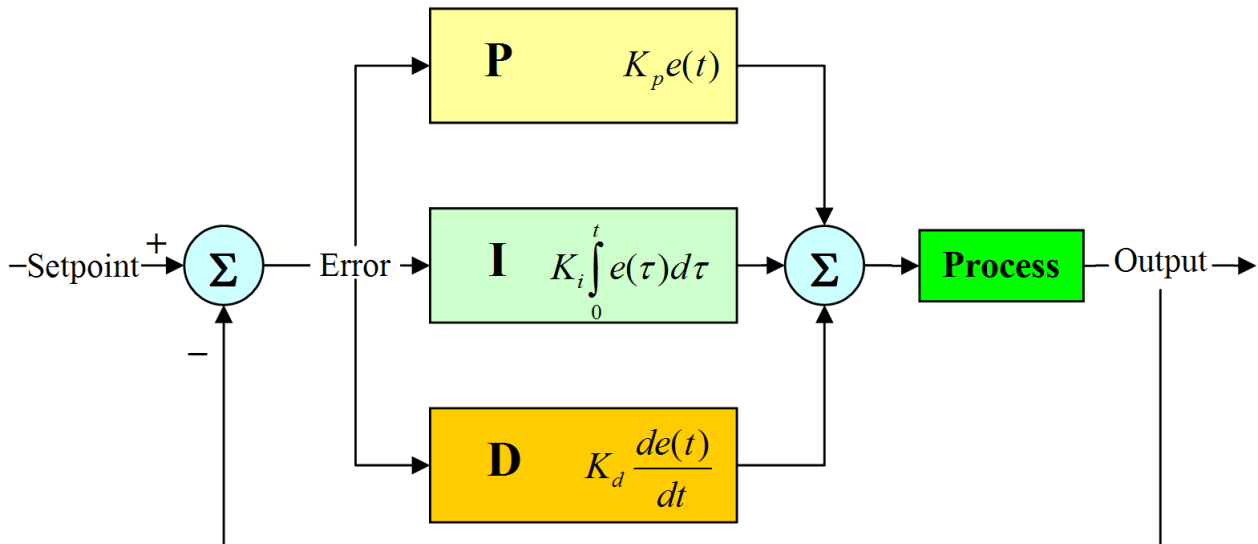


dove $F(s)G_1(s)=G_2(s)$.

4.1) PID

Il controllore PID è un metodo di sintesi euristico molto semplice da realizzare, attraverso la somma di 3 semplici blocchi:

- Il blocco **P**, di parametro K_P , proporzionale all'errore
- Il blocco **I**, di parametro K_I , proporzionale all'integrale dell'errore
- Il blocco **D**, di parametro K_D , proporzionale alla derivata dell'errore



$$u(t) = K_P e(t) + K_I \int e(t) dt + K_D \frac{de(t)}{dt}$$

Ogni blocco ha le sue caratteristiche:

- il blocco proporzionale corregge l'errore appena questo si scosta da zero riducendo il tempo di salita, inoltre riduce ma non elimina l'errore a regime;
- il blocco integrativo migliora l'errore a regime, eliminandolo se l'ingresso è una costante o un gradino, ma può rallentare la risposta al transitorio;
- il blocco derivativo migliora il transitorio del sistema retroazionato e aumenta la stabilità del sistema riducendo l'overshoot, ma in presenza di input impulsivi può destabilizzare il sistema.

Di seguito una tabella che riassume quanto detto:

Controllore	Tempo di Salita	Sovraelongazione	Transitorio	Errore a Regime
K_P	Diminuisce	Aumenta	Piccole Variazioni	Diminuisce
K_I	Diminuisce	Aumenta	Aumenta	Eliminato
K_D	Piccole Variazioni	Diminuisce	Diminuisce	Non Varia

Bisogna quindi trovare il giusto equilibrio tra questi 3 blocchi.

In frequenza il nostro controllore diventa:

$$C(s) = K_P + \frac{K_I}{s} + K_D s = \frac{K_D s^2 + K_P s + K_I}{s}$$

4.1.1) Implementazione MatLab

Ora andiamo su MatLab e scriviamo il seguente codice per creare F(s):

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

num1=[(m1+m2) b2 k2];
den1=[(m1*m2) (m1*(b1+b2))+(m2*b1) (m1*(k1+k2))+(m2*k1)+(b1*b2) (b1*k2)+(b2*k1) k1*k2];
G1=tf(num1,den1);

num2=[-(m1*b2) -(m1*k2) 0 0];
den2=[(m1*m2) (m1*(b1+b2))+(m2*b1) (m1*(k1+k2))+(m2*k1)+(b1*b2) (b1*k2)+(b2*k1) k1*k2];
G2=tf(num2,den2);

numf=num2;
denf=num1;
F=tf(numf,denf);
```

Ora implementiamo il controllore PID C(s), assegnando dei valore ai parametri:

```
Kd = 208025;
Kp = 832100;
Ki = 624075;
C = pid(Kp,Ki,Kd);
```

```
C =
      1
  Kp + Ki * --- + Kd * s
           s

with Kp = 8.32e+05, Ki = 6.24e+05, Kd = 2.08e+05

Continuous-time PID controller in parallel form.
```

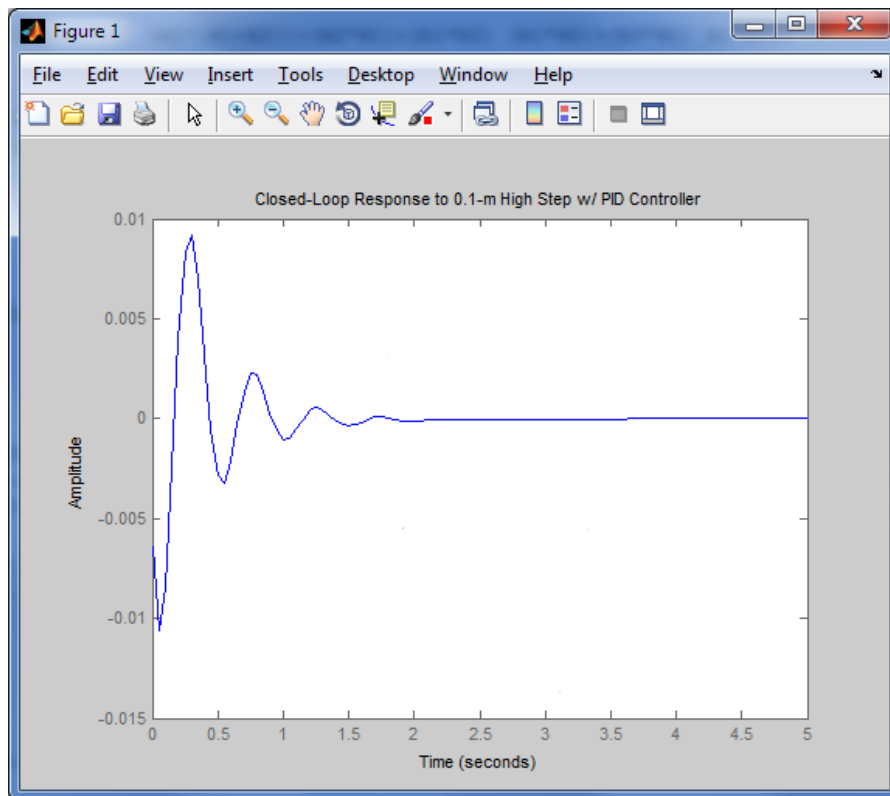
Adesso simuliamo la risposta del sistema al disturbo di un gradino, il comando `feedback(A,B)` restituisce il modello del sistema dove il blocco A è retroazionato dal

blocco B, come in questo caso, dove il blocco della sospensione è retroazionato dal blocco del controllore:

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

Ora che abbiamo creato la funzione di trasferimento dell'intero sistema, andiamo a visualizzarla risposta su un grafico, moltiplicando tale funzione per 0.1 (il nostro gradino di 10 cm) e osservando cosa accade nei primi 5 secondi (che è quello che ci interessa vedere).

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



Come si può osservare, mentre il tempo di assestamento è inferiore ai 5 secondi come richiesto dalle specifiche, non lo è la sovraelongazione, che raggiunge i 9 millimetri contro i 5 mm richiesti.

Ora che abbiamo la funzione di trasferimento del sistema, non resta che mettere a punto i valori delle costanti del controllore PID: proviamo con moltiplicare tutte le costanti per un fattore 2:

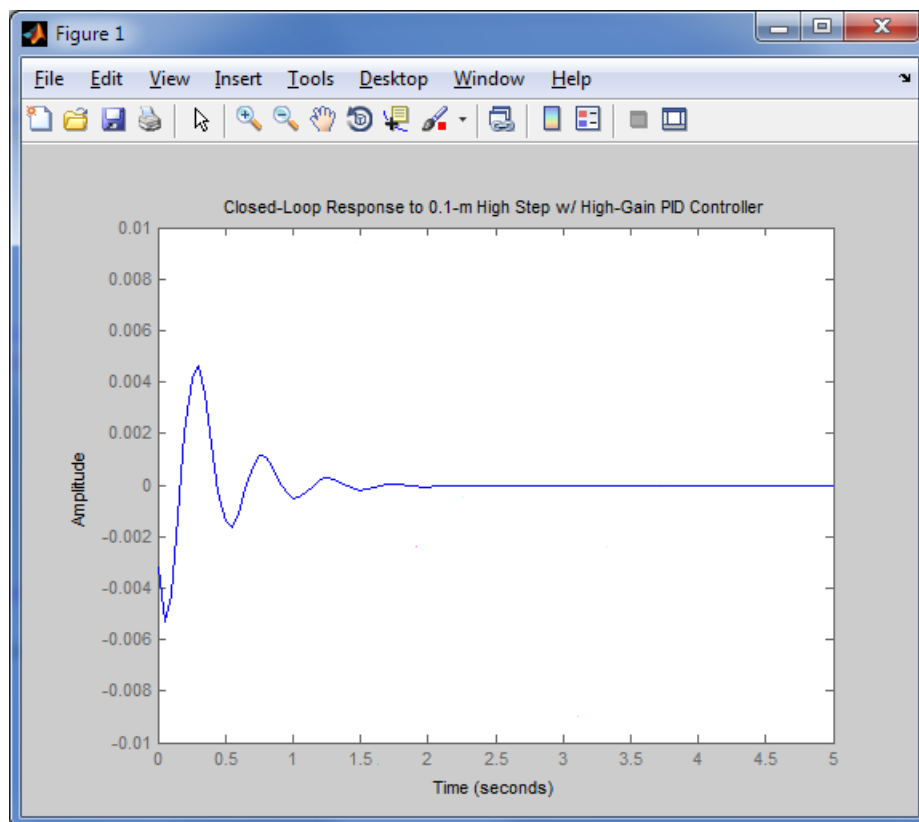
```

Kd=2*Kd;
Kp=2*Kp;
Ki=2*Ki;
C=pid(Kp,Ki,Kd);
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 PID Controller')

```

Usiamo la funzione axis per regolare il grafico come quello precedente per poterlo confrontare meglio

```
axis([0 5 -.01 .01])
```



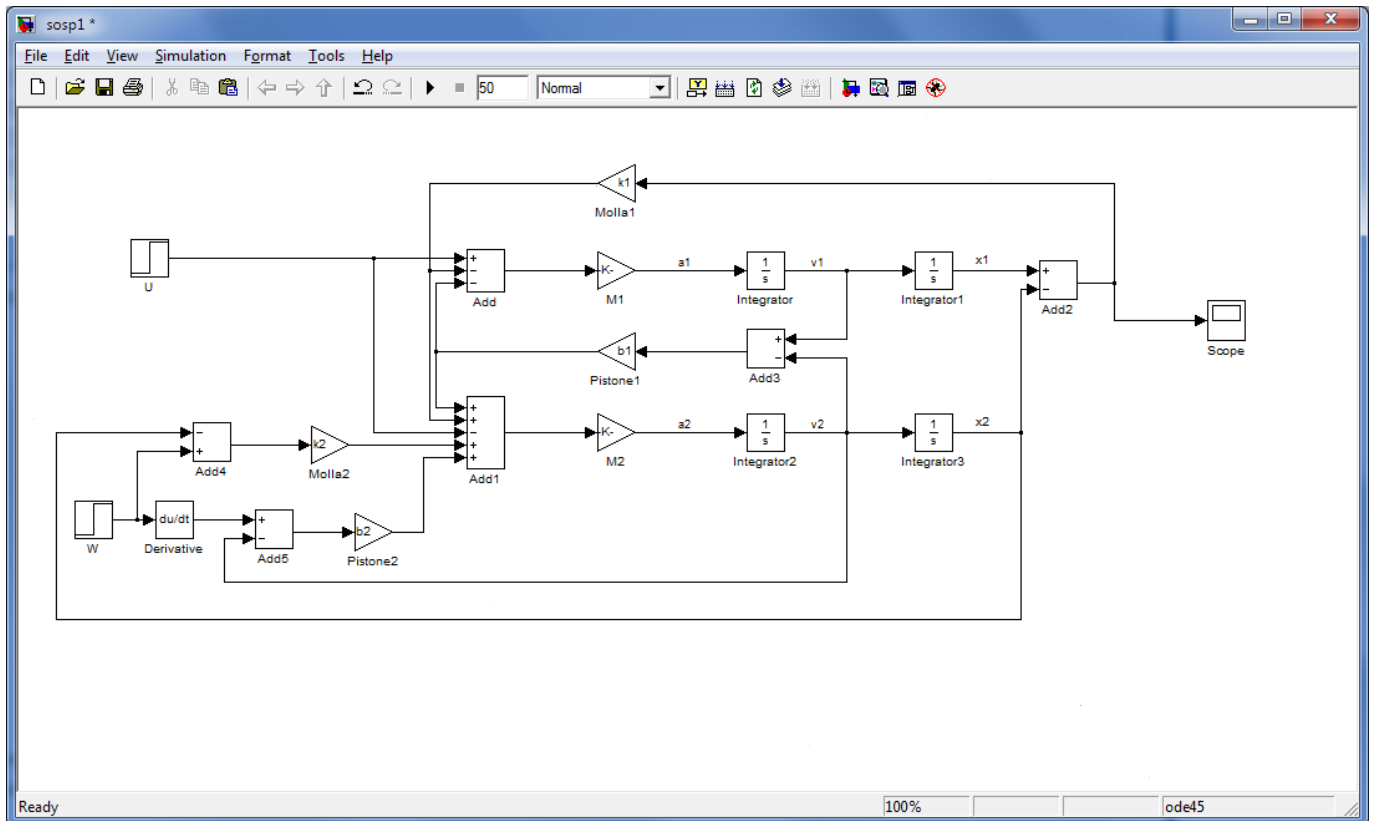
Adesso finalmente entrambe le specifiche sono soddisfatte: la sovraelongazione è di circa 5mm, cioè soddisfa appieno il 5% richiesto e il tempo di assestamento (2 secondi) è nettamente inferiore ai 5 richiesti.

Siamo quindi riusciti a progettare un controllore PID che risolve il problema iniziale.

4.1.2) Implementazione Simulink

Inserendo su MatLab il comando *simulink*, si aprirà una finestra da qui è possibile iniziare a creare i vari blocchi del nostro modello.

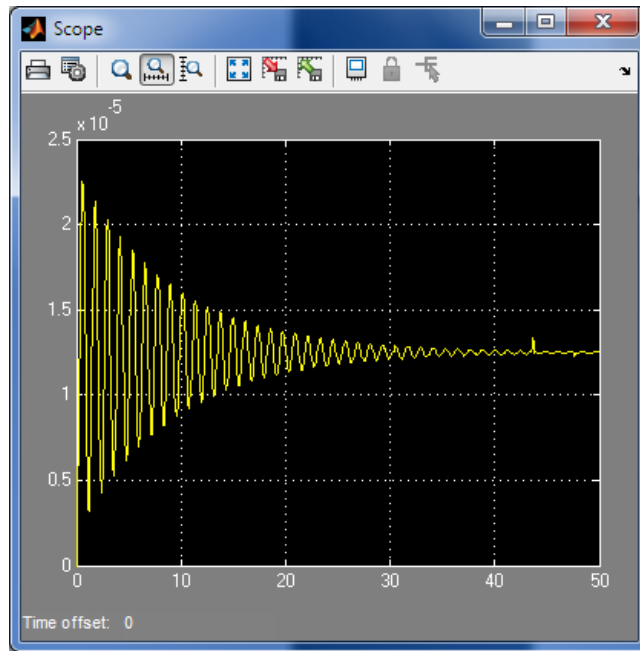
Ecco qui il modello finito:



Ora impostiamo il tempo di simulazione a 50 secondi e inseriamo i parametri su MatLab (nei blocchi c'è il riferimento a tali parametri):

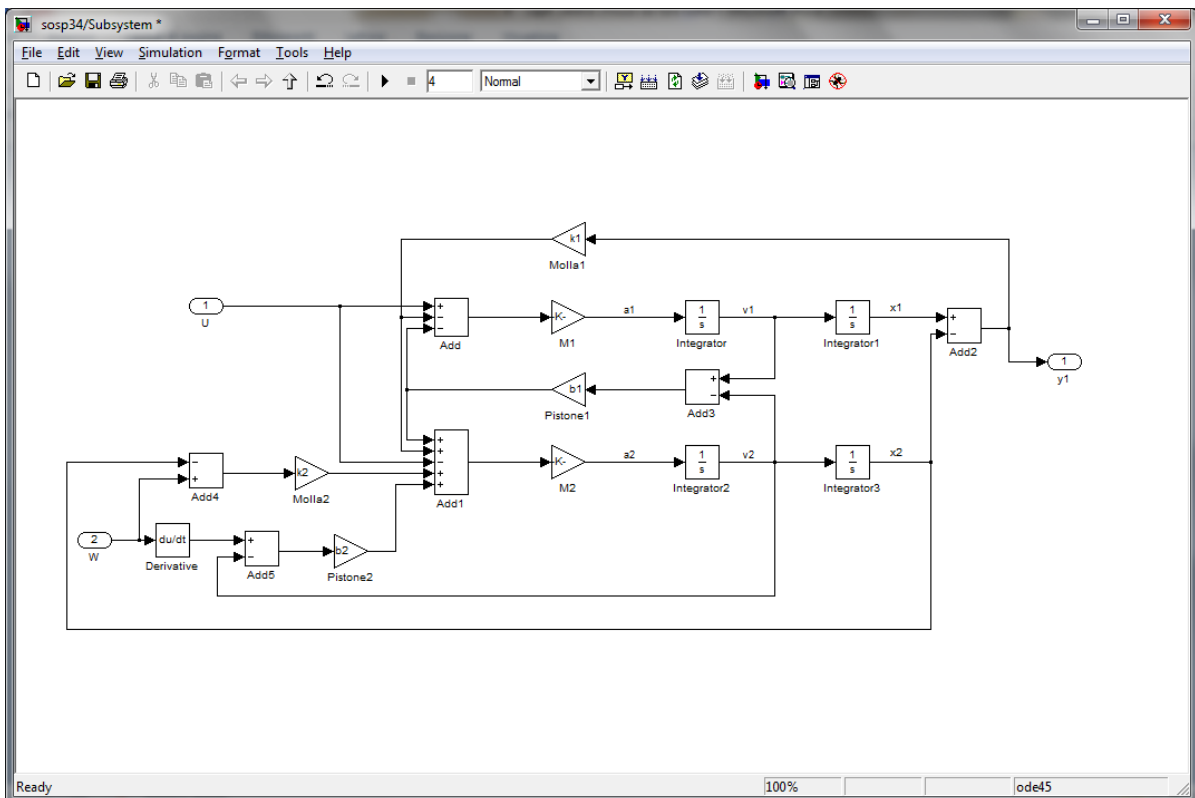
```
m1 = 2500;  
m2 = 320;  
k1 = 80000;  
k2 = 500000;  
b1 = 350;  
b2 = 15020;
```

Ora facciamo partire la simulazione, per vedere il risultato facciamo doppio click sul blocco *Scope*:

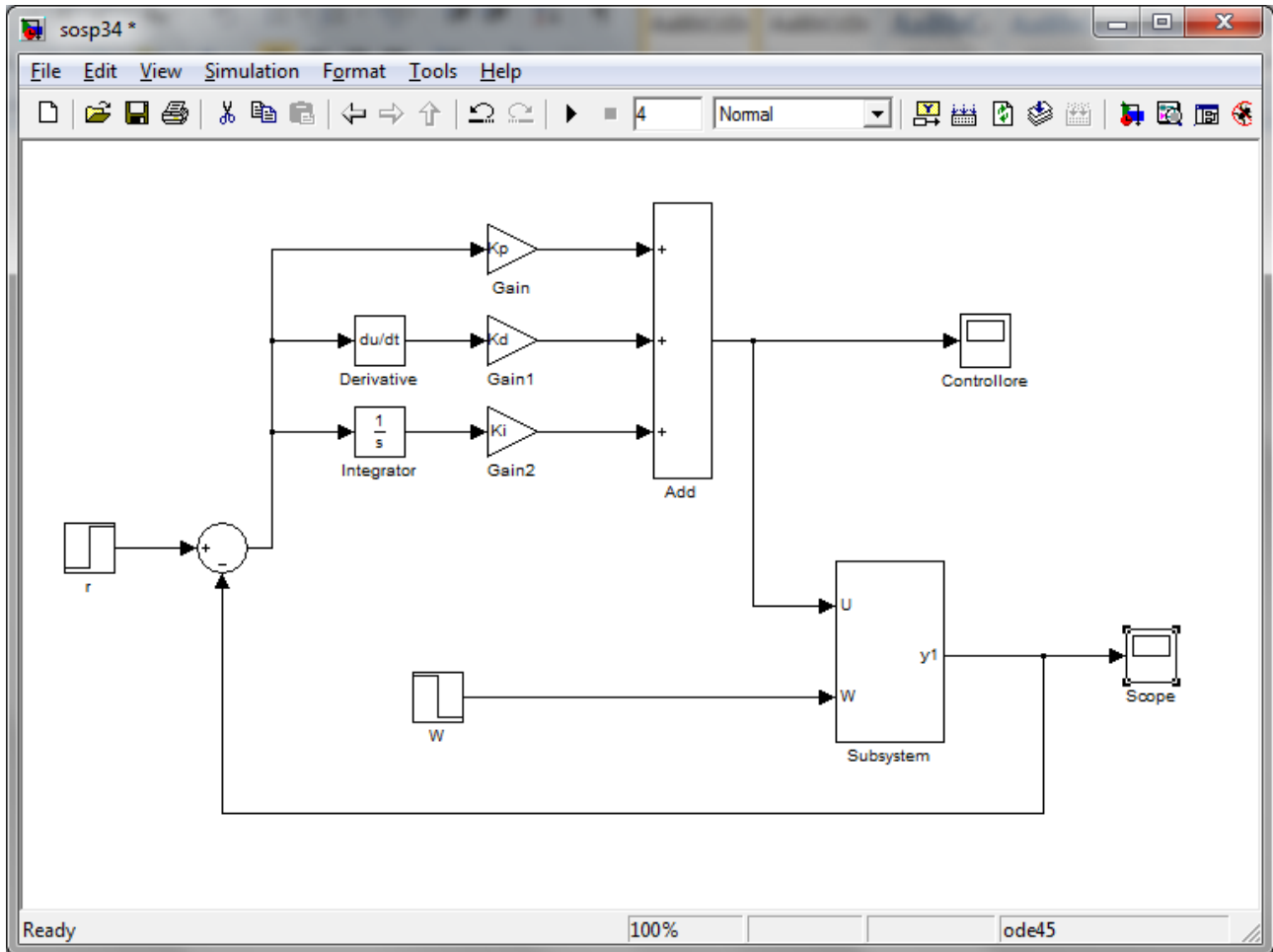


Questo è il modello in catena aperta senza controllore, e come visto in precedenza, il tempo di assestamento è molto lungo (50 secondi).

Andiamo quindi a implementare un controllore PID. Per prima cosa creiamo un nuovo modello su Simulink, e inseriamo il blocco *subsystem*, doppio click e ci apre una nuovo foglio dove andiamo a copiare il modello visto sopra, modificando i blocchi di ingresso e di uscita come in figura:



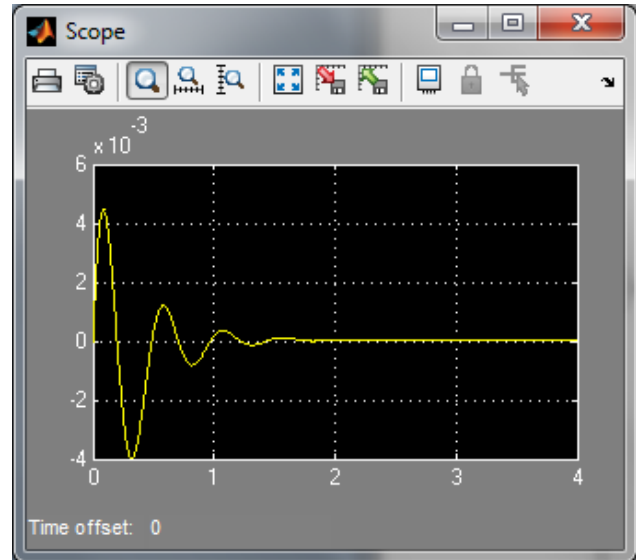
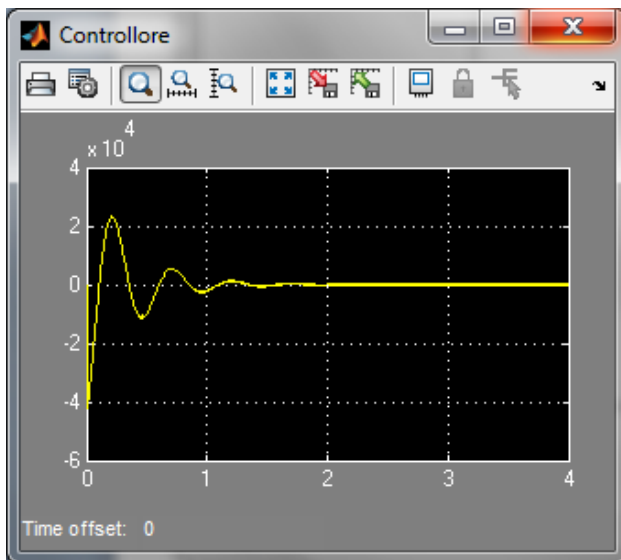
Ora intorno a quel blocco andiamo a creare il controllore, aggiungiamo integratore e derivatore, e i vari guadagni K_P , K_D e K_I .



A questo punto inseriamo ancora una volta i dati del modello, e i guadagni trovati precedentemente:

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

```
Kd = 208025*2;  
Kp = 832100*2;  
Ki = 624075*2;
```



Il primo grafico mostra la forza applicata dal controllore al sistema quando si incontra una buca di 10 cm, il secondo invece mostra le oscillazioni che ha il bus, e come ci aspettavamo avendo usato i dati che sapevamo essere ideali per il nostro problema, sovraelongazione e tempo di assestamento soddisfano i requisiti di partenza.

4.2) Luogo delle radici

Proviamo ora ad analizzare il problema utilizzando il metodo del luogo delle radici: si tratta di un metodo grafico dove si proiettano i poli e zeri della funzione di trasferimento sul piano dei numeri complessi e si va a studiarne la stabilità in base al variare di un parametro k .

Per iniziare, riprendiamo il codice MatLab per le funzioni di trasferimento:

```

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

num1=[(m1+m2) b2 k2];
den1=[(m1*m2) (m1*(b1+b2))+(m2*b1) (m1*(k1+k2))+(m2*k1)+(b1*b2) (b1*k2)+(b2*k1) k1*k2];
G1=tf(num1,den1);

```



```

num2=[-(m1*b2) -(m1*k2) 0 0];
den2=[(m1*m2) (m1*(b1+b2))+(m2*b1) (m1*(k1+k2))+(m2*k1)+(b1*b2) (b1*k2)+(b2*k1) k1*k2];
G2=tf(num2,den2);

numf=num2;
denf=num1;
F=tf(numf,denf);

```

Andiamo ora ad analizzare il polinomio al denominatore, che è lo stesso sia per G_1 che per G_2 usando il comando *roots*.

```

R = roots(denp)

R =

    -23.9758 +35.1869i
    -23.9758 -35.1869i
     -0.1098 + 5.2504i
     -0.1098 - 5.2504i

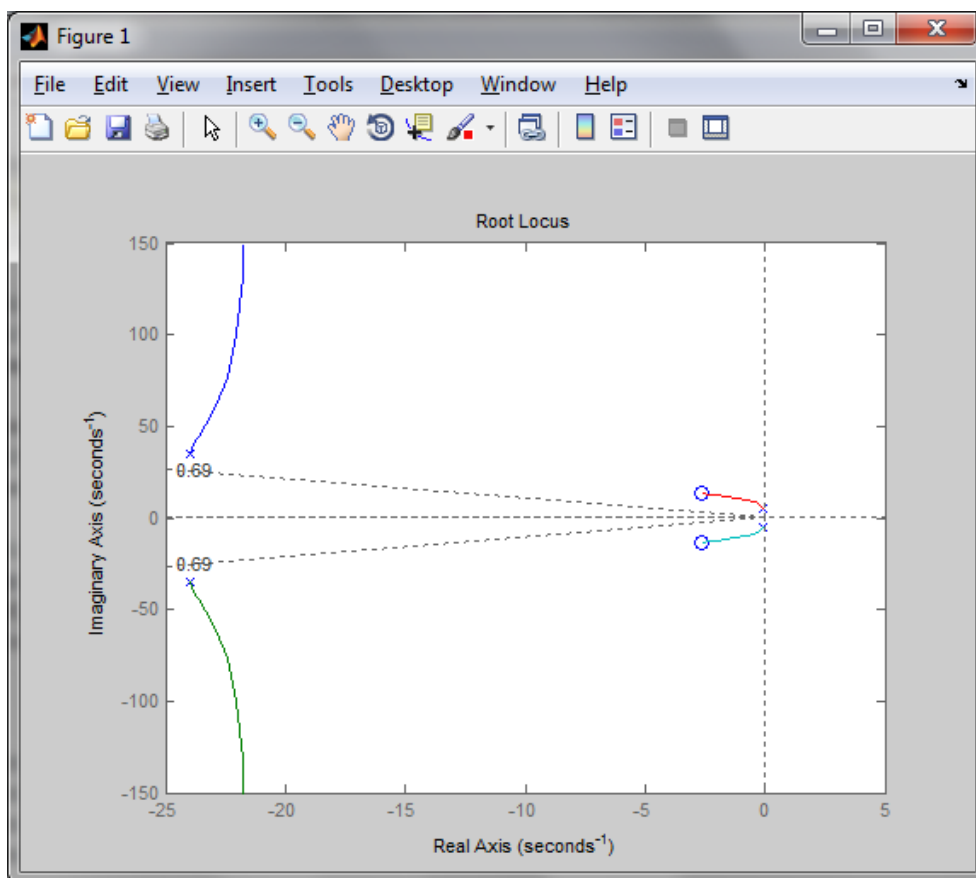
```

Si utilizza il comando *roots* per rappresentare su un grafico le radici appena trovate e il comando *sgrid* per tracciare le rette che delimitano il 5% che è il nostro obiettivo:

```

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

```



$$s^2 + 2\xi\omega_n + \omega_n^2$$

Come si può notare, i due poli dominanti sono quelli più vicini all'asse immaginario, cioè $-0.1098 \pm 5.2504i$, che sono radici complesse coniugate.

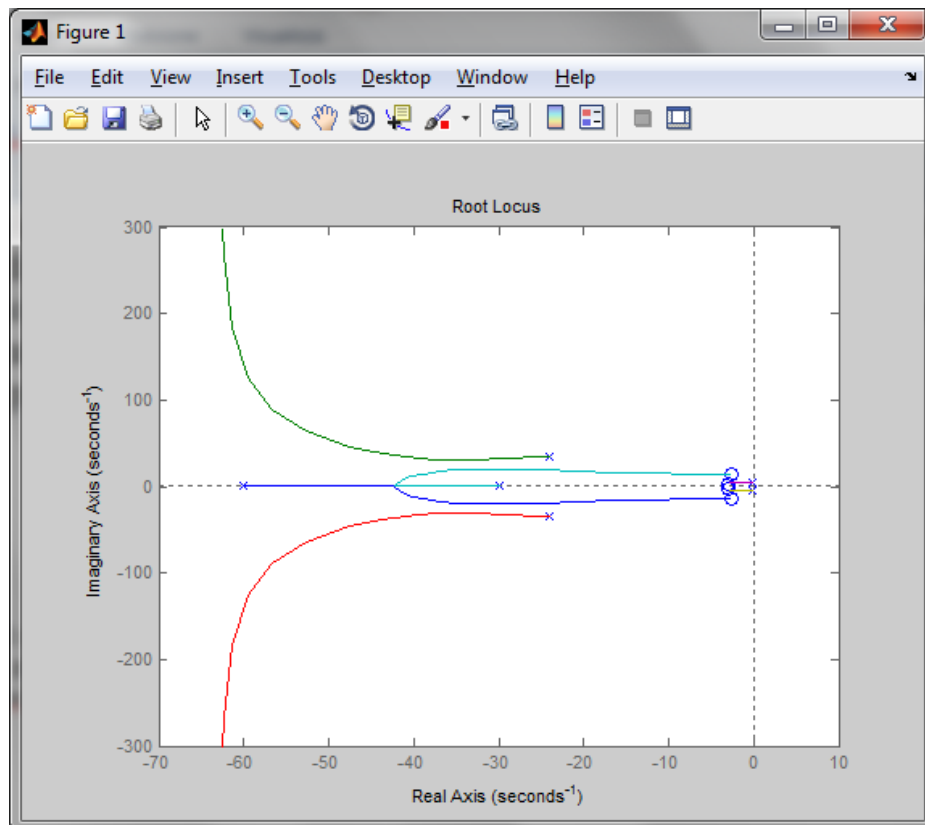
La vicinanza di due poli e zeri all'asse immaginario potrebbe causare problemi di stabilità, quindi dobbiamo cercare di spostarli più a sinistra possibile, attraverso l'aggiunta di un controllore. Dobbiamo mettere due zeri più vicini possibili all'asse immaginario per compensare i due poli dominanti, e due poli abbastanza lontani dall'asse immaginario per aumentare la prontezza del sistema.

Inseriamo quindi il seguente controllore:

$$C(s) = \frac{s^2 + 6s + 21.25}{s^2 + 90s + 1800}$$

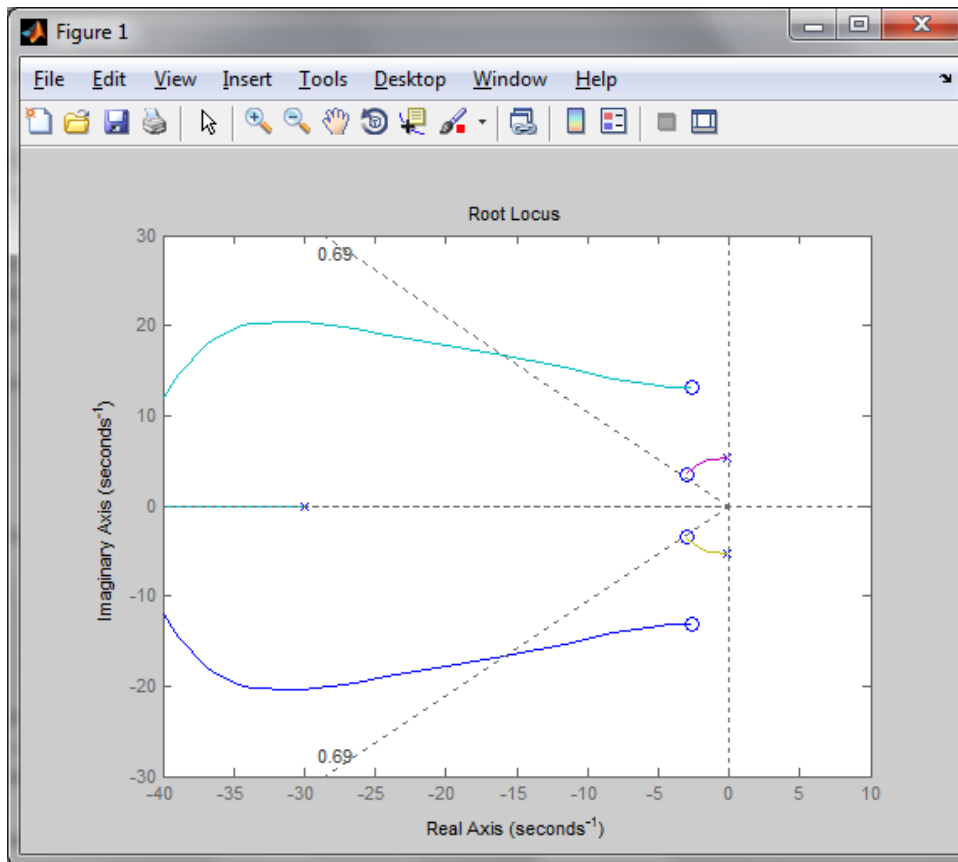
```
z1=3+3.5i;  
z2=3-3.5i;  
p1=30;  
p2=60;  
numc=conv([1 z1],[1 z2]);  
denc=conv([1 p1],[1 p2]);  
C=tf(numc,denc);
```

```
rlocus(C*G1)
```



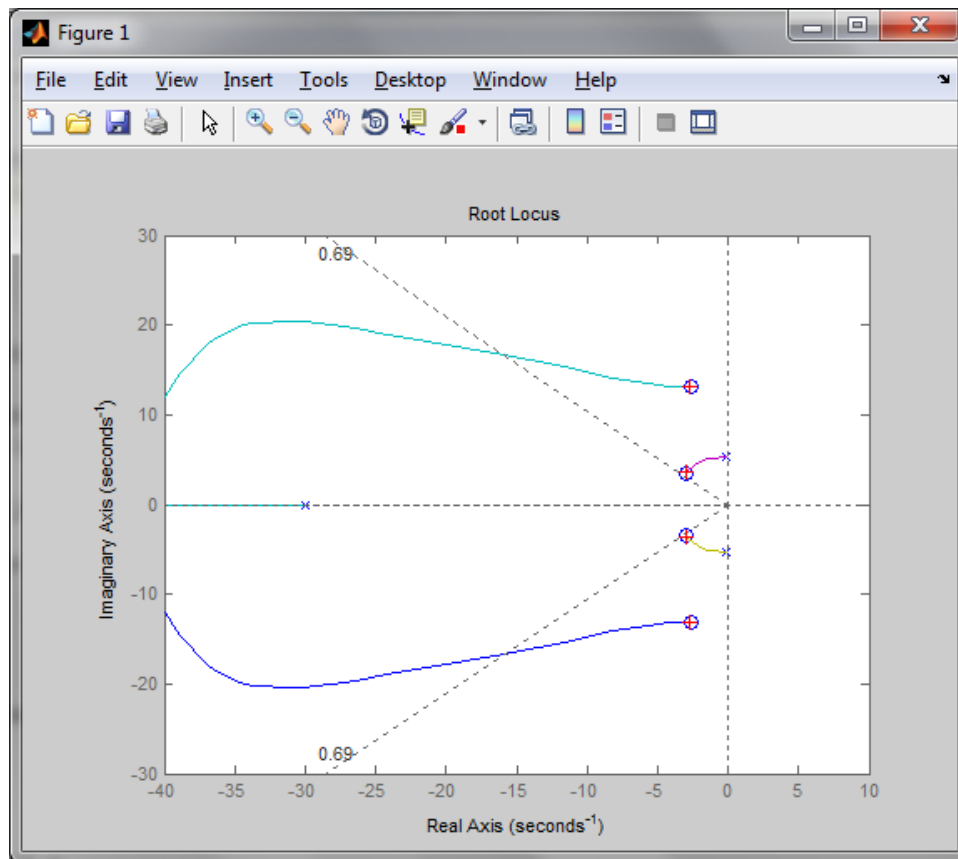
Andiamo ora a zoomare per vedere nel dettaglio:

```
axis([-40 10 -30 30])  
z=-log(0.05)/sqrt(pi^2+(log(0.05)^2))  
sgrid(z,0)
```



Ora che abbiamo spostato il luogo delle radici oltre la linea che delimita il 5% delle specifiche di smorzamento, possiamo scegliere il guadagno adeguato per soddisfare le altre specifiche: il tempo di assestamento al massimo di 5 secondi e una sovraelongazione più piccola possibile. Per la seconda, di solito è necessario selezionare un guadagno corrispondente ad un punto sul luogo delle radici vicino all'asse reale e lontano dall'asse immaginario; invece in questo caso abbiamo bisogno di una cancellazione zero/polo vicino all'asse immaginario quindi abbiamo bisogno di selezionare un guadagno corrispondente ad un punto del luogo delle radici in prossimità di uno zero. Per fare ciò usiamo il comando *rlocfind* che ci permetterà di scegliere sul luogo il punto desiderato.

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



Select a point in the graphics window

selected_point =

$-2.9739 + 3.4472i$

k =

$3.5339e+08$

poles =

$1.0e+03 *$

$-0.0634 + 1.1171i$

$-0.0634 - 1.1171i$

$-0.0027 + 0.0131i$

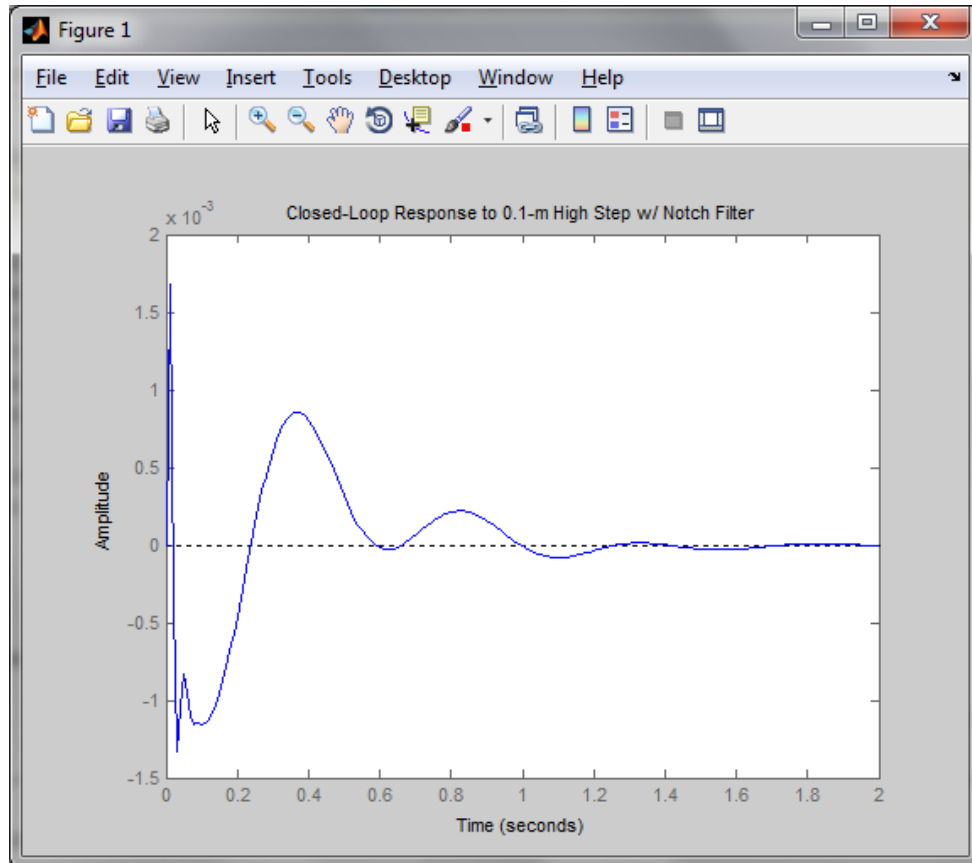
$-0.0027 - 0.0131i$

$-0.0030 + 0.0036i$

$-0.0030 - 0.0036i$

Prendiamo il guadagno trovato e andiamo a vedere come si comporta il nostro sistema con questo nuovo dato:

```
sys_cl=F*feedback(G1,k*C);  
  
t=0:0.01:2;  
step(0.1*sys_cl,t)  
title('Closed-Loop Response to 0.1-m High Step w/ Notch Filter')
```



Da questo grafico si vede che quando il bus incontra un disturbo di 0.1m la deviazione massima resta tra $\pm 3\text{mm}$ e dopo 2 secondi le oscillazioni si sono stabilizzate: abbiamo quindi trovato un guadagno che soddisfi entrambe le specifiche iniziali.

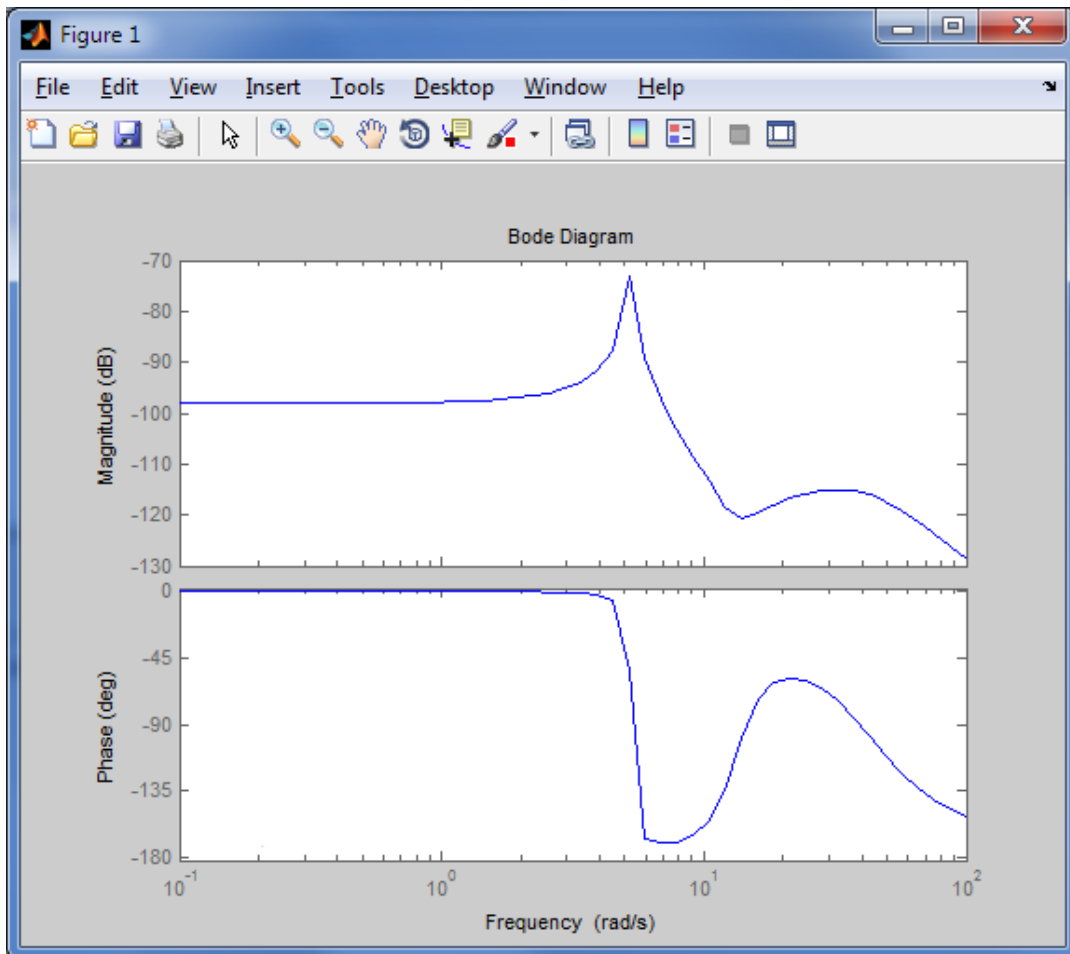
4.3) Frequenza(Bode)

L'ultimo metodo di sintesi proposto sfrutta i diagrammi di Bode per studiare e analizzare il sistema in frequenza.

La strategia su cui si basa è tradurre le richieste progettuali in vincoli nel dominio della frequenza, vincoli che il sistema in catena aperta, avente funzione di trasferimento $W(s)=C(s)G_I(s)$, deve soddisfare, ottenendo così il controllore $C(s)$ cercato.

Inseriamo innanzitutto su MatLab i soliti dati del modello fisico, e andiamo a disegnare il diagramma di Bode:

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



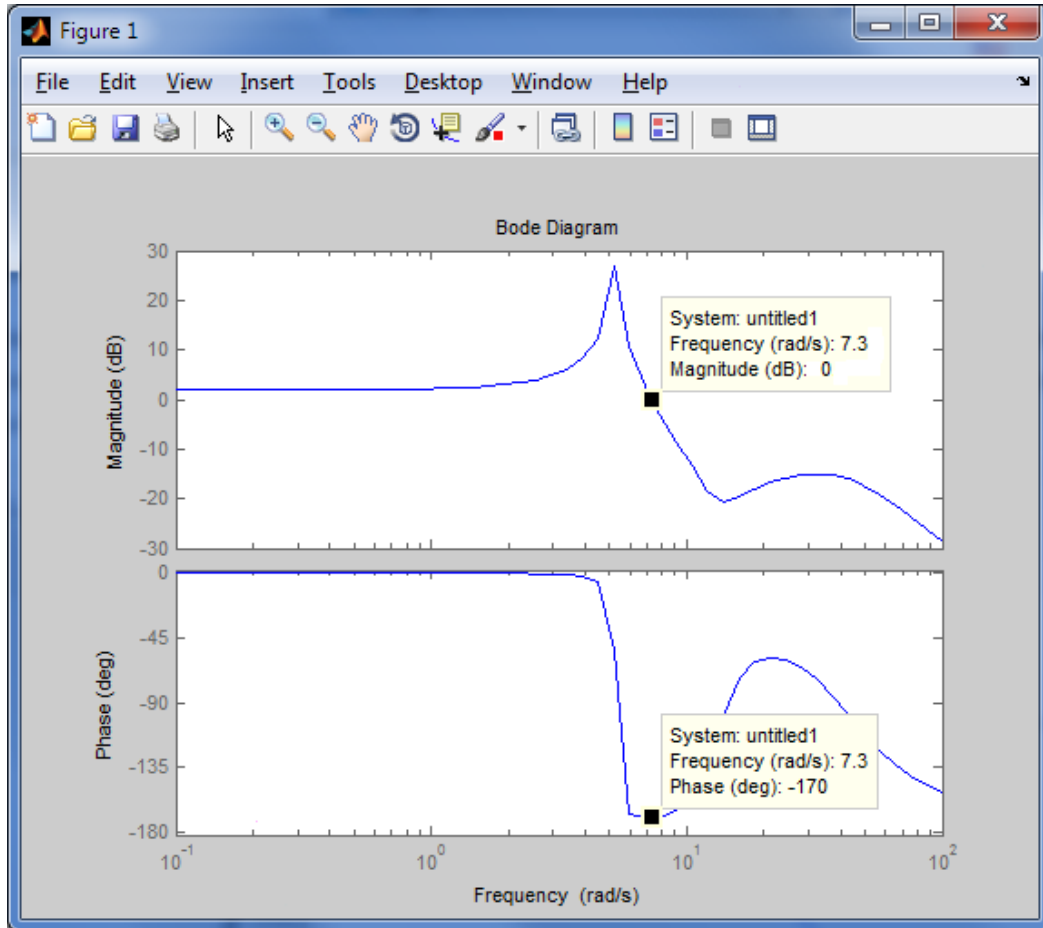
Preso il controllore $W(s)$:

$$W(s) = \frac{K_W}{s^{h_w}} \bar{W}(s)$$

Per prima cosa vogliamo che per basse frequenze il guadagno sia intorno ai 0dB, cioè $\bar{W}(0) = 1$, e questo si può ottenere facilmente guardando il modulo della FdT e

osservando che basta aggiungere 100dB per traslare il grafico del modulo verso l'alto, lasciando inalterata la fase. Andiamo quindi a scrivere:

```
K=100000;
bode(K*G1,w)
```



Ora, osservando il grafico si nota che la pulsazione di attraversamento ω_a è 7.3 rad/s alla quale corrisponde un margine di fase di -80° (il margine di fase si ottiene prendendo l'angolo osservato e aggiungendo 90°).

Dai requisiti progettuali vogliamo che la sovraelongazione massima sia $S=0.05$, che tradotta in specifiche di Bode diventa una condizione sul margine di fase $m_\phi=1-0.8S=1-0.04=0.96rad$ ($\sim 55^\circ$).

Essendo quindi che vogliamo un margine di fase di circa 55° , dobbiamo aggiungere circa 140° . Per fare ciò, useremo due reti anticipatrici identiche, che ci faranno guadagnare 70° l'una.

Il modello del controllore è il seguente:

$$\bar{C}(s) = \frac{1 + sT_a}{1 + saT_a}$$

Ora non ci resta che determinare a e T_a ; per determinare la prima dobbiamo usare la fase positiva che vogliamo aggiungere:

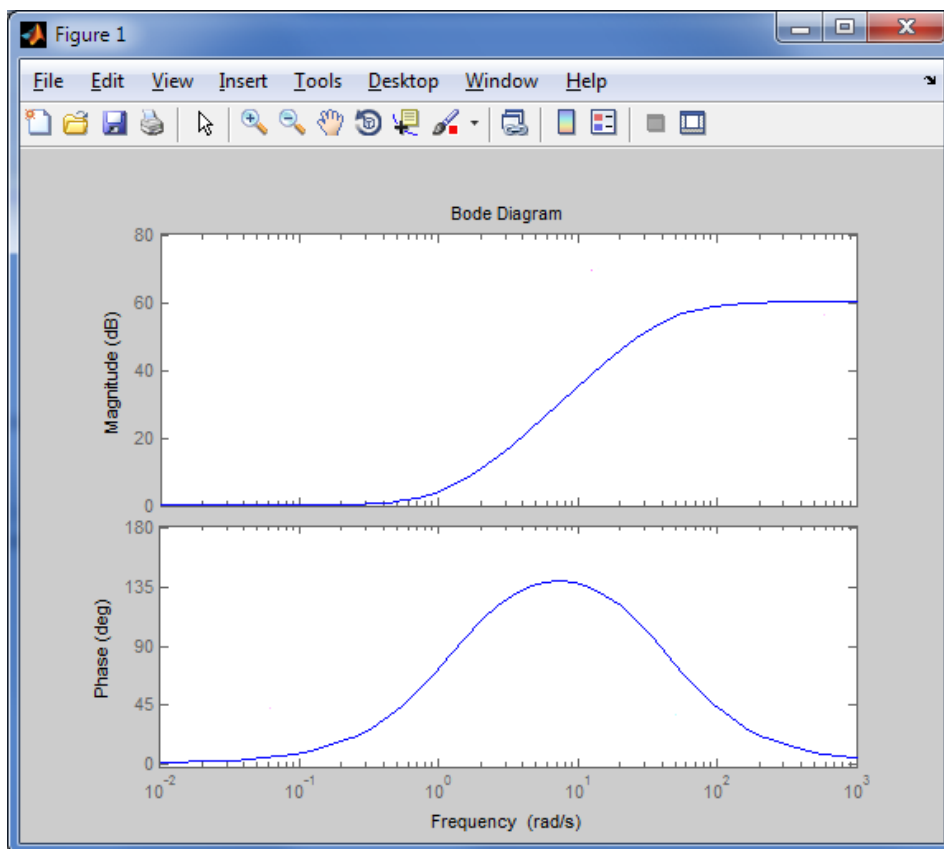
$$a = \frac{1 - \sin 70^\circ}{1 + \sin 70^\circ} = 0.031091$$

Per calcolare il secondo parametro, usiamo anche la pulsazione di attraversamento:

$$T_a = \frac{1}{\omega_a \sqrt{a}} = 0.77689$$

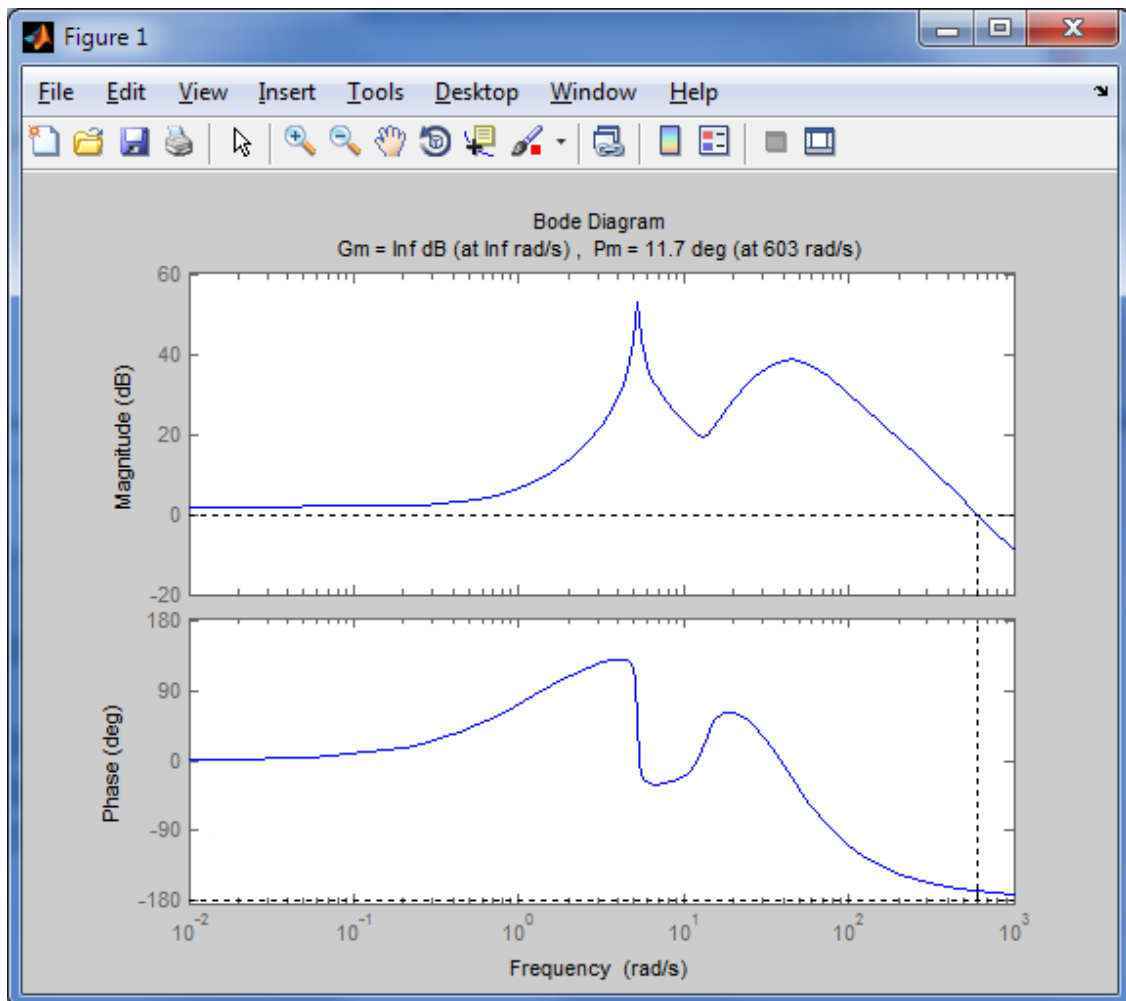
Andiamo quindi ora a creare il controllore con MatLab:

```
a = (1-sin(70/180*pi))/(1+sin(70/180*pi));
w=7.3;
T=1/(w*sqrt(a));
aT=a*T;
numc = conv([T 1], [T 1]);
denc = conv([aT 1], [aT 1]);
C = tf(numc,denc);
bode(C);
```



Infine inseriamo il controllore nel nostro sistema:

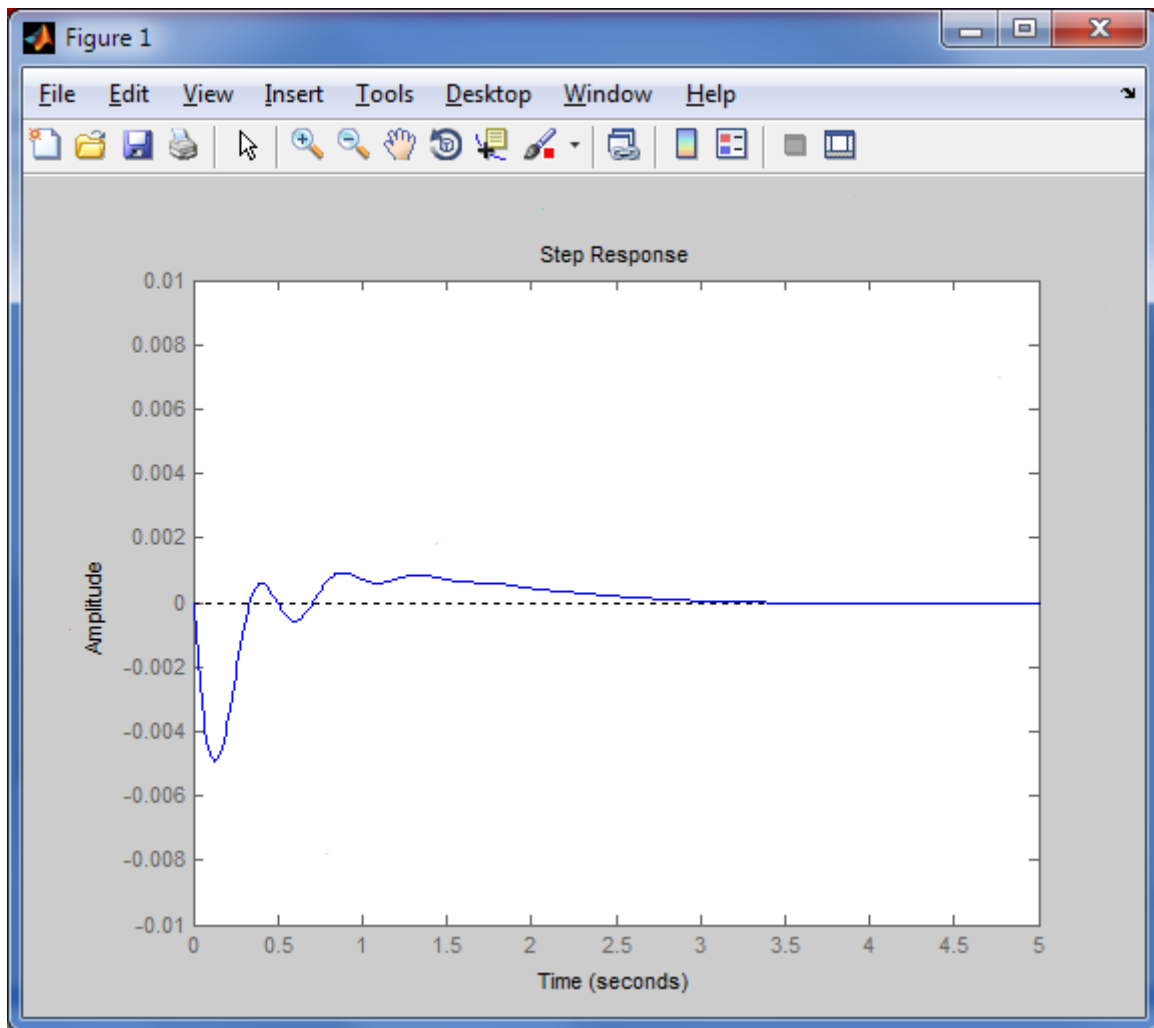
```
margin(K*C*G1)
```



Come si può osservare, i requisiti sembrano soddisfare le richieste progettuali, andiamo quindi ora a verificare la loro attendibilità, creando la funzione di trasferimento del sistema in catena chiusa e osservando la risposta al gradino unitario:

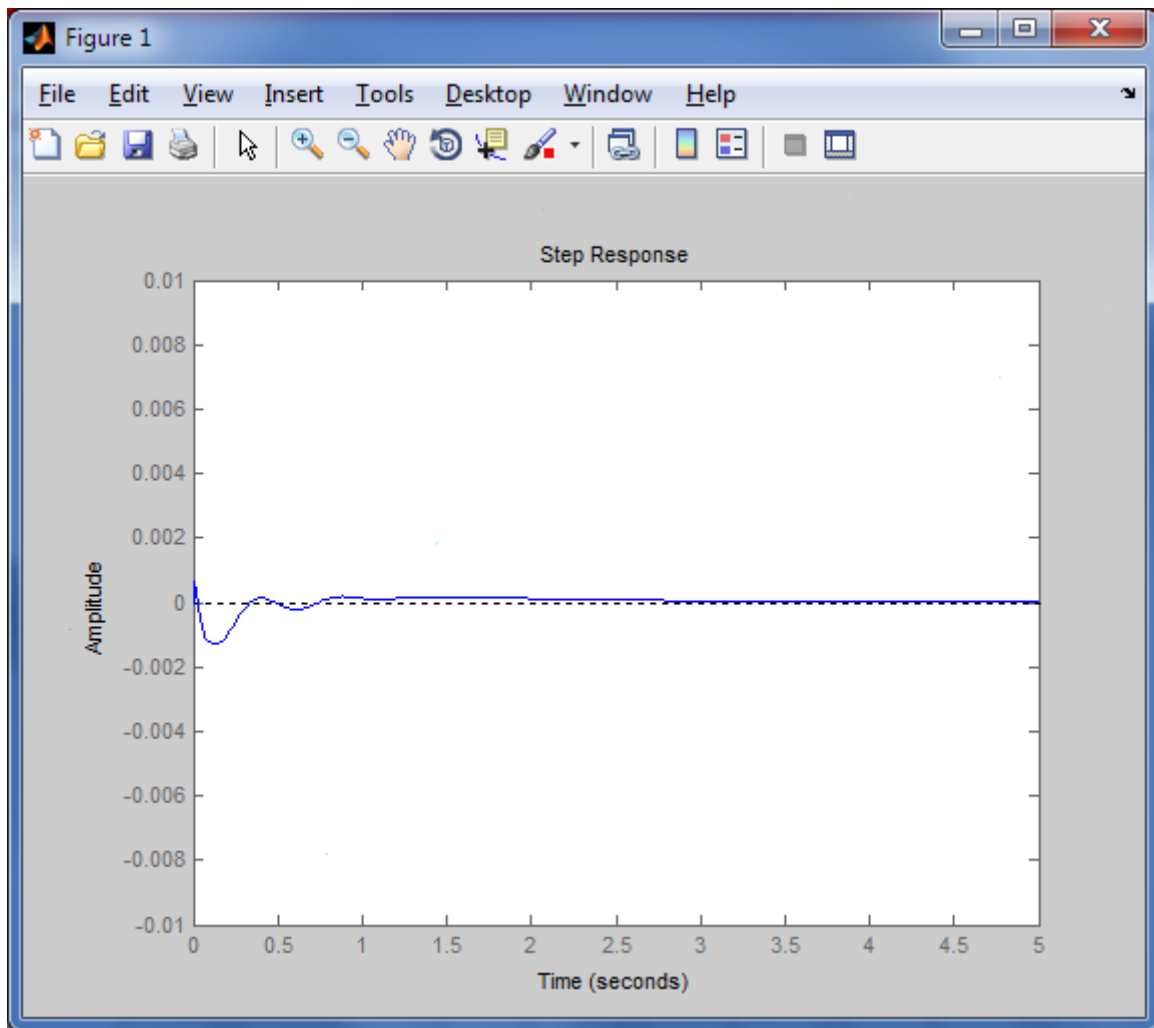
```
sys_cl = F*feedback(G1,K*C);
```

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



Possiamo ulteriormente aumentare la velocità di risposta aumentando il guadagno di un fattore 4:

```
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])
```



Abbiamo migliorato la sovralongazione percentuale del sistema e ridotto il tempo di assestamento, tutti i requisiti di progetto sono stati ampiamente soddisfatti.

6) Conclusioni

Discutiamo ora in questo capitolo il lavoro svolto.

Come ampiamente discusso in precedenza, lo scopo di questa tesi è stato di prendere in esame un problema reale, semplificarlo, tradurlo in termini matematici ed elaborare diverse strategie per risolverlo, cosa portata a termine con ogni tipologia di controllo proposta.

A fronte degli esercizi svolti abitualmente durante il corso di Controlli Automatici dove le specifiche erano sul modello in esame, qui sono state poste sul disturbo esterno. Questo tipo di richieste mi hanno aiutato a osservare il problema da un altro punto di vista, cambiano prospettiva.

Inoltre, questa tesi mi ha fatto familiarizzare e comprendere le enormi potenzialità di elaborazione calcoli che ha un programma come MatLab, che mi ha dato modo di mettere in pratica le conoscenze acquisite durante il corso, di approfondirle e ampliarle.

Bibliografia

- [1] Sandro Zampieri, *Dispensa di Controlli Automatici*, Edizioni Libreria Progetto, Padova, 2011.
- [2] <http://ctms.engin.umich.edu/CTMS/index.php?aux=Home>, Control Tutorials for MatLab and Simulink.