



UNIVERSITA' DEGLI STUDI DI PADOVA  
Facoltà di Ingegneria  
Corso di Laurea in Ingegneria dell'Informazione

**Progettazione in ambiente Matlab/Simulink di un sistema di  
controllo di un pendolo inverso**

**Laureando:** Mattia Sessolo  
**Relatore:** Prof. Alessandro Beghi

21 Febbraio 2012

## Abstract

Lo scopo di questa tesina è quello di utilizzare le conoscenze acquisite durante il percorso della laurea triennale, in particolare durante il corso di *controlli automatici* per realizzare alcuni tipi di compensatori, in ambiente Matlab, in grado di controllare un pendolo inverso. Nella prima sezione viene descritta l'impostazione del problema ovvero il viene descritto il sistema, determinato il modello matematico che lo descrive e in seguito vengono date le specifiche che dovranno essere raggiunte dal sistema compensato. Nella seconda fase vengono osservate le risposte indiciali e in frequenza del sistema aperto; In seguito, nella terza sezione vengono determinati tre tipi di controllori con i metodi della risposta in frequenza, luogo delle radici e infine viene determinato un controllore di tipo PID; infine viene fatto un confronto tra le risposte indiciali dei sistemi per valutare le prestazioni dei compensatori creati, e una verifica dei risultati con simulink

## Contents

<b>1</b>	<b>Impostazione del problema</b>	<b>4</b>
1.1	Descrizione del sistema . . . . .	4
1.2	Derivazione del modello matematico . . . . .	5
1.3	Impostazioni delle specifiche di controllo . . . . .	6
<b>2</b>	<b>Analisi del sistema in catena aperta</b>	<b>7</b>
2.1	Analisi delle varie risposte indiciali . . . . .	8
2.2	Analisi della risposta in frequenza . . . . .	9
<b>3</b>	<b>Realizzazione e analisi dei controllori</b>	<b>9</b>
3.1	Controllore PID . . . . .	10
3.2	Progetto di un controllore con il metodo della risposta in frequenza . . . . .	20
3.3	Progetto di un controllore con il metodo del luogo delle radici	32
<b>4</b>	<b>Confronto e verifica dei risultati</b>	<b>40</b>
4.1	Confronto delle risposte impulsive e al gradino . . . . .	40
4.2	Verifica risultati con Simulink . . . . .	45

# 1 Impostazione del problema

Il problema consiste nell'analisi di un pendolo inverso e nella costruzione di alcuni tipi di compensatori in modo tale da rispettare le specifiche che verranno in seguito introdotte.

## 1.1 Descrizione del sistema

Il sistema è composto da un carrello libero di muoversi in una direzione (orizzontale) sulla cui superficie superiore è stata vincolata un'asta, la quale è libera di ruotare sempre rispetto alla direzione di movimento del carrello. Per la

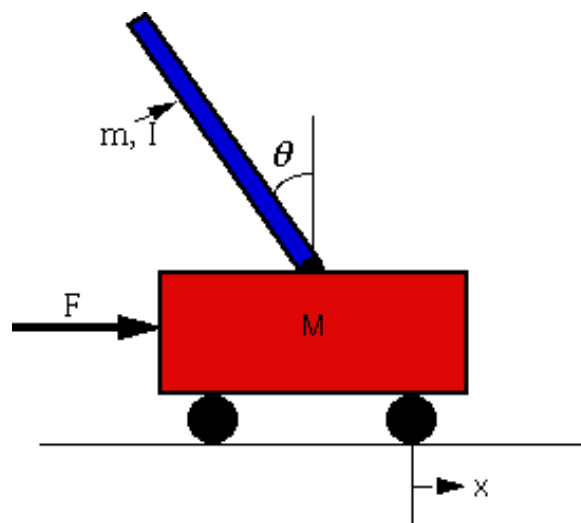


Figure 1: Schema di principio del sistema

derivazione del modello si tiene tra l'altro conto di attrito viscoso dovuto al movimento del carrello, massa del carrello e del pendolo e momento di inerzia del pendolo ma anche dell'asta a cui è vincolato. In primo luogo spiego il significato e gli eventuali valori dei simboli utilizzati:

- $M$  massa del carrello 0.5 [Kg]
- $m$  massa del pendolo 0.2 [Kg]
- $b$  coefficiente attrito viscoso del carrello 0.1
- $l$  lunghezza della distanza dal centro di massa del pendolo 0.3 [m]
- $I$  momento d'inerzia del pendolo 0.006 [Kg $m^2$ ]
- $F$  forza applicata al carrello [N]
- $x$  posizione del carrello [m]

- $\theta$  angolo tra il pendolo e la verticale al carrello

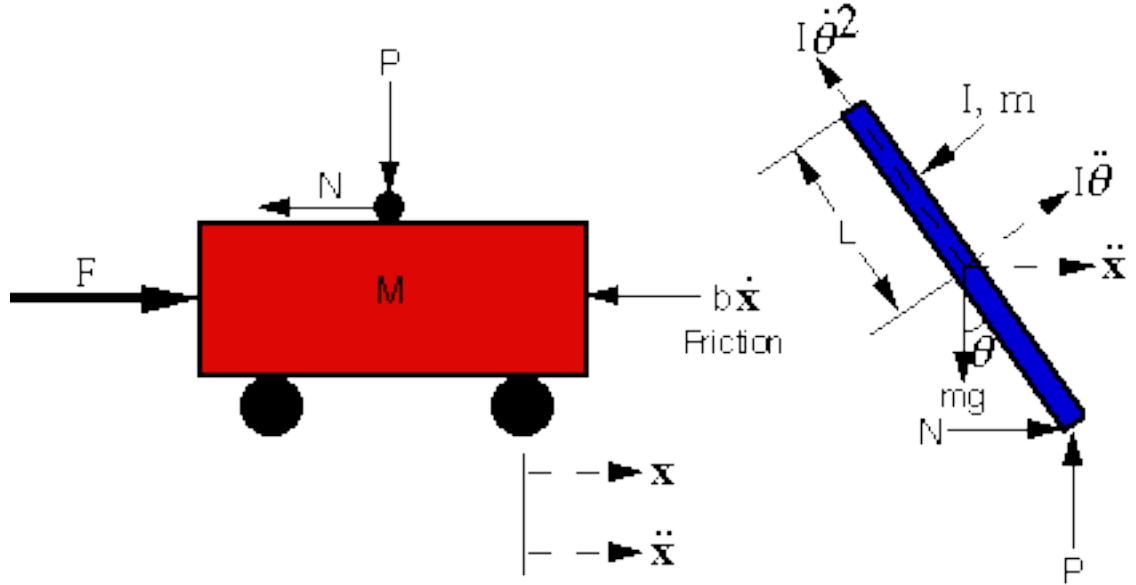


Figure 2: significato dei simboli utilizzati

## 1.2 Derivazione del modello matematico

Scrivendo la seconda legge di Newton applicata tenendo come origine del sistema di riferimento il vincolo che tiene solidale l'asta con il carrello, otteniamo le seguenti 4 equazioni:

$$-N(t) = m \frac{d^2}{dt^2}(x(t) - L \sin \theta(t)) \quad (1)$$

$$-P(t) - mg = m \frac{d^2}{dt^2}(L \cos \theta(t)) \quad (2)$$

$$-LN(t) \cos \theta(t) - LP(t) \sin \theta(t) = I \frac{d^2}{dt^2} \theta(t) \quad (3)$$

$$f(t) - b \frac{d^2}{dt^2} x(t) - N = M \frac{d^2}{dt^2} x(t) \quad (4)$$

Una volta trovate queste equazioni è necessario linearizzarle attorno ai suoi punti di equilibrio: per far questo basta porre a zero le derivate in quanto noi siamo interessati all'andamento a regime; Ponendo a zero le derivate, si ottiene il seguente punto di equilibrio:

$$\begin{cases} N(t) = 0 \\ P(t) = 0 \\ -Lmg \sin \theta(t) = 0 \Leftrightarrow \theta(t) = 0 + 2k\pi \vee \theta(t) = \pi + 2k\pi \\ f(t) = 0 \end{cases}$$

Ci sono infiniti punti di equilibrio, che in pratica però si riconducono a due punti,  $0 \vee \pi$ : il primo è stabile mentre il secondo è instabile. Scegliamo il primo che corrisponde al punto intorno al quale noi vorremmo stabilizzare il sistema. Inserendo (1) e (2) in (3) si ottiene:

$$Lm\ddot{x}(t) \cos \theta(t) + l^2 m \sin \theta(t) \dot{\theta}^2(t) \cos \theta(t) + L^2 m \cos^2 \theta(t) \ddot{\theta}(t) + Lmg \sin \theta(t) + l^2 m \sin^2 \theta(t) \ddot{\theta}(t) - l^2 m \cos \theta(t) \sin \theta(t) \dot{\theta}^2(t) = I\ddot{\theta}(t)$$

Linearizzando attorno a  $\theta = 0$  si ottiene:

$$Lm\ddot{x}(t) + lmg + l^2 m \ddot{\theta}(t) - I\ddot{\theta}(t) = 0$$

che raccogliendo diventa:

$$(I - Lm^2)\ddot{\theta}(t) - Lmg\theta(t) = lm\ddot{x}(t) \quad (5)$$

Allo stesso modo, inserendo (1) in (4) si ottiene:

$$f(t) - b\dot{x}(t) - m\ddot{x}(t) + mL \sin \theta(t) \dot{\theta}^2(t) + mL \cos \theta(t) \ddot{\theta}(t) = M\ddot{x}(t)$$

la quale linearizzando sempre attorno a  $\theta = 0$  e raccogliendo diventa:

$$f(t) - b\dot{x}(t) + mL\ddot{\theta}(t) = (m + M)\ddot{x}(t) \quad (6)$$

Per poter scrivere la funzione di trasferimento, prima bisogna che le due equazioni (5) e (6) siano portate dal dominio del tempo al dominio di Laplace:

$$(I - L^2 m)s^2 \Theta(s) - lmg\Theta(s) = lms^2 X(s) \quad (7)$$

$$F(s) - bsX(s) + mLs^2 \Theta(s) = (M + m)s^2 X(s) \quad (8)$$

A questo punto, non resta altro che esplicitare da (7)  $X(S)$  in funzione di  $\Theta(s)$  e sostituire  $X(S)$  nella (8); a questo punto portando si ottiene la funzione di trasferimento da  $F$  a  $\theta$ :

$$G(s) = \frac{Lms}{((M + m)s + b)(I + l^2 m)s^2 - lgm} - m^2 l^2 s^3 \quad (9)$$

La quale, dopo aver sviluppato il denominatore diventa:

$$G(s) = \frac{Lms}{((m + M)(I + l^2 m) - m^2 l^2)s^3 + b(I + l^2 m)s^2 - lgm(m + M)s - lgmb} \quad (10)$$

### 1.3 Impostazioni delle specifiche di controllo

Il controllore che dovrà essere progettato deve soddisfare le seguenti specifiche:

- Si vuole reiezione completa al disturbo cioè errore nullo ad un disturbo di tipo impulsivo o a gradino

- Setting Time minore di 5s, cioè si vuole che il pendolo sia nella condizione  $|\theta(t)| \leq 0.01rad$  entro 5 secondi.
- Unserhoot/Overshoot minore di 0.05 rad.

## 2 Analisi del sistema in catena aperta

Per poter analizzare il sistema in catena aperta con matlab, come prima cosa bisogna definire in matlab la nostra funzione di trasferimento da analizzare la quale viene inserita con il seguente codice:

```
% ++++++
% Definizione dei parametri del sistema
% ++++++
M = .5;
m = 0.2;
b = 0.1;
I = 0.006;
g = 9.8;
l = 0.3;

% ++++++
% Definizione della funzione di trasferimento
% ++++++
num=[1*m 0];
den=[(m+M)*(I+l^2*m)-m^2*l^2) b*(I+l^2*m) -l*g*m*(m+M) -l*g*m*b];
pend = tf(num,den); %corrisponde a G(s)
```

Da come si può vedere, la prima porzione di codice serve ad assegnare dei valori ai vari parametri utilizzati nel corso del programma come ad esempio la massa del carrello, il momento d'inerzia dell'asta etc. Per poter definire una funzione di trasferimento, in matlab si utilizza il comando tf; questo comando accetta come parametri due vettori che rappresentano rispettivamente il numeratore e il denominatore della funzione di trasferimento da rappresentare e restituisce un oggetto funzione di trasferimento. La funzione di trasferimento, una volta sostituiti al suo interno i valori assegnati diventa:

$$G(s) = \frac{0.06s}{0.0132s^3 + 0.0024s^2 - 0.4116s - 0.0588} \quad (11)$$

Analizzando il polinomio a denominatore si vede che esso non ha tutti i coefficienti dello stesso segno e quindi per la regola di Cartesio non è di Hurwitz: allora il sistema avrà sicuramente poli a parte reale positiva. Con Matlab si può facilmente verificare che il sistema ha poli a parte reale positiva: basta utilizzare il comando pole(s) che restituisce gli poli della f.d.t. s:

`pole(pend)`

I poli quindi risultano:  $p_1 = 5.5651$ ,  $p_2 = -5.6041$ ,  $p_3 = -0.1428$  e infatti, il sistema ha un polo instabile, e allora anche il sistema è instabile.

## 2.1 Analisi delle varie risposte indiciali

Come prima cosa, bisogna determinare le risposte indiciali ovvero le risposte del sistema quando viene sollecitato con un impulso o con un gradino; per far questo introduciamo due nuovi comandi matlab: **impulse** e **step**, entrambi ricevono come input una funzione di trasferimento e come output restituiscono il grafico della risposta del sistema rispettivamente all'impulso e al gradino. Di seguito viene mostrato il codice per generare le due risposte indiciali.

```
impulse(pend);  
step(pend);
```

e di seguito vengono mostrati i risultati del calcolo: Da come si può vedere

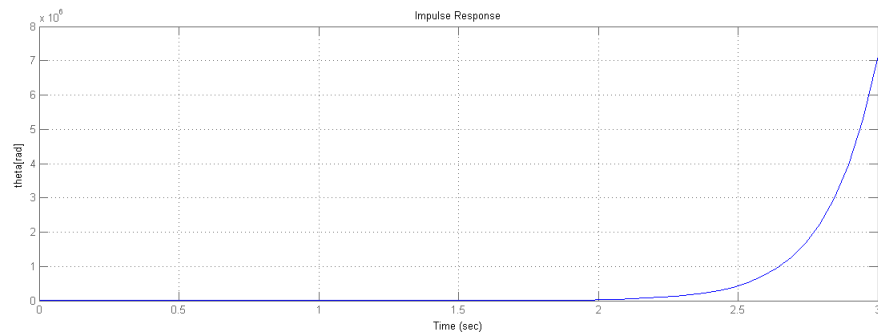


Figure 3: Risposta impulsiva del sistema ad anello aperto

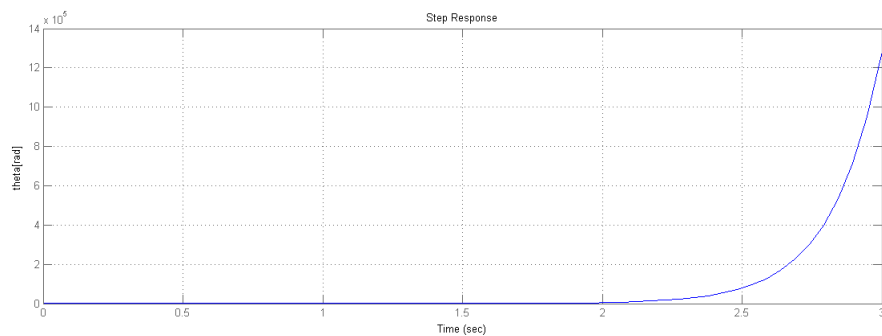


Figure 4: Risposta al gradino del sistema ad anello aperto

dai grafici delle risposte indiciali, il sistema è chiaramente instabile inquanto



l'uscita del sistema, con un ingresso limitato, non tende sicuramente a 0 rad ma, diverge; in sostanza il nostro pendolo non sta fermo sulla verticale ma cade.

## 2.2 Analisi della risposta in frequenza

Per analizzare la risposta in frequenza del sistema bisogna ricorrere al diagramma di nyquist che viene calcolato con il seguente comando:

```
nyquist(pend)
```

Il risultato prodotto dall'elaborazione: Anche in questo caso si nota che il

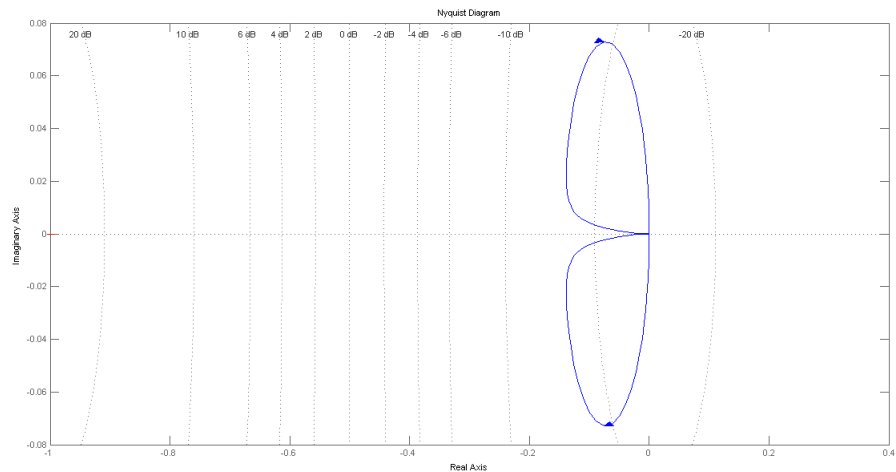


Figure 5: Diagramma di Nyquist del sistema

sistema non è stabile in quanto non compie nessun giro antiorario attorno a -1;

## 3 Realizzazione e analisi dei controllori

Il nostro obiettivo è quello di realizzare, con tecniche diverse, diversi tipi di controllori con l'obiettivo di raggiungere le specifiche impostate dal problema. Come primo passo si osserva lo schema a blocchi ad anello chiuso del nostro problema: In questo schema si indica con  $G(s)$  la f.d.t. del nostro sistema, con  $C(s)$  la f.d.t. del controllore (da determinare),  $r(t)$  il riferimento e con  $y(t)$  l'uscita del sistema che nel nostro caso è chiamata  $\theta(t)$ . Poiché il nostro obiettivo è quello di tenere il pendolo dritto sulla verticale del carrello (quindi avere  $\theta = 0$ ) il nostro riferimento sarà posto a 0. Ne consegue che il nostro sistema quindi avendo 0 come riferimento ha anche uscita pari

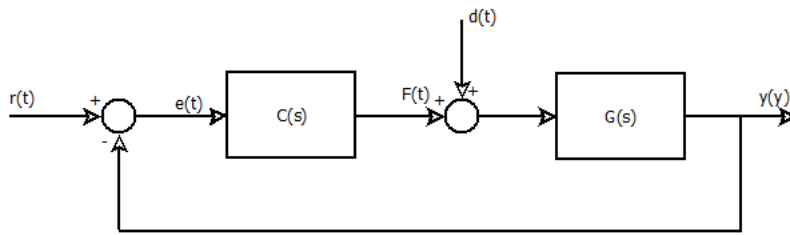


Figure 6: Schema a blocchi del sistema

a 0 (ovviamente se lo stato di partenza è uno stato di equilibrio e nel nostro caso lo è) e quindi se non agiscono forze esterne sul carrello il pendolo rimarrà sempre fermo infatti se  $r(t) \equiv 0$  allora anche  $R(s) \equiv 0$  in quanto per le proprietà delle funzioni nel dominio di Laplace  $\Theta(s) = R(s)G(s) = 0G(s) = 0$ . Ne consegue che il sistema dovrà essere stabilizzato solo rispetto alla sollecitazione di una forza esterna  $F(t)$  che nel nostro caso possiamo vederla come un disturbo. Lo schema a blocchi del sistema quindi si può riorganizzare in questo modo: La nuova f.d.t. del sistema da stabilizzare diventa

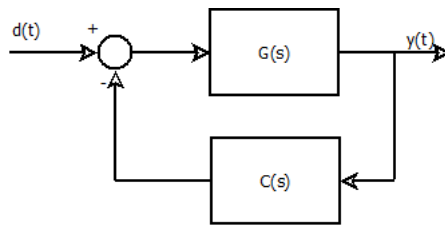


Figure 7: Schema a blocchi del sistema

$W(s) = \frac{G(s)}{1+G(s)C(s)}$  Per definirla in MATLAB, una volta che sono state definite le funzioni di trasferimento  $G(s)$  e  $C(s)$ , si usa il comando `feedback` che restituisce un oggetto che rappresenta la f.d.t.  $W(s)$ .

`W=feedback(G,H)`

A questo punto, una volta definito il problema si può proseguire alla realizzazione dei controllori.

### 3.1 Controllore PID

Il controllore di tipo PID è formato dalla somma di tre funzioni in modo tale da avere all'ingresso del sistema tre contributi: uno proporzionale all'errore, uno proporzionale alla derivata dell'errore e infine un contributo proporzionale all'integrale dell'errore: questo tipo di controllore è della forma:

$$C(s) = Kp + Kds + \frac{Ki}{s} = \frac{Kds^2 + Kps + Ki}{s} \quad (12)$$

dove con  $K_d$ ,  $K_p$  e  $K_i$  sono i rispettivamente le costanti derivativa,proporzionale e integrativa del controllore. Si è scelto di utilizzare direttamente un PID in quanto un controllore puramente proporzionale non riuscirebbe a stabilizzare il sistema e quindi devo passare ad un controllore dinamico. Per iniziare impongo le costanti di integrazione e derivazione uguali a 1 in quanto voglio trovare indicativamente l'ordine di grandezza della costante di proporzionalità; per far questo utilizzo il seguente codice MATLAB che implementa un ciclo for dove al suo interno viene creato un controllore per ogni valore di  $K_p$  contenuto nel vettore  $K$  e calcolata, sempre per ogni valore di  $K_p$  la risposta impulsiva e al gradino del sistema:

```
K=[1 5 10 20 40 60 80 100 120 140];
for i=1:1:10
    Kp=K(i);
    Kd=1;
    Ki=1;
    t=0:0.01:10;
    contPID=tf([Kd Kp Ki],[1 0]);
    pend_fback=feedback(pend,contPID)
    subplot(5,2,i);
    impulse(pend_fback,t);
    subplot(5,2,i);
    step(pend_fback,t);
    legend(strcat('Kp= ',num2str(Kp)));
end
```

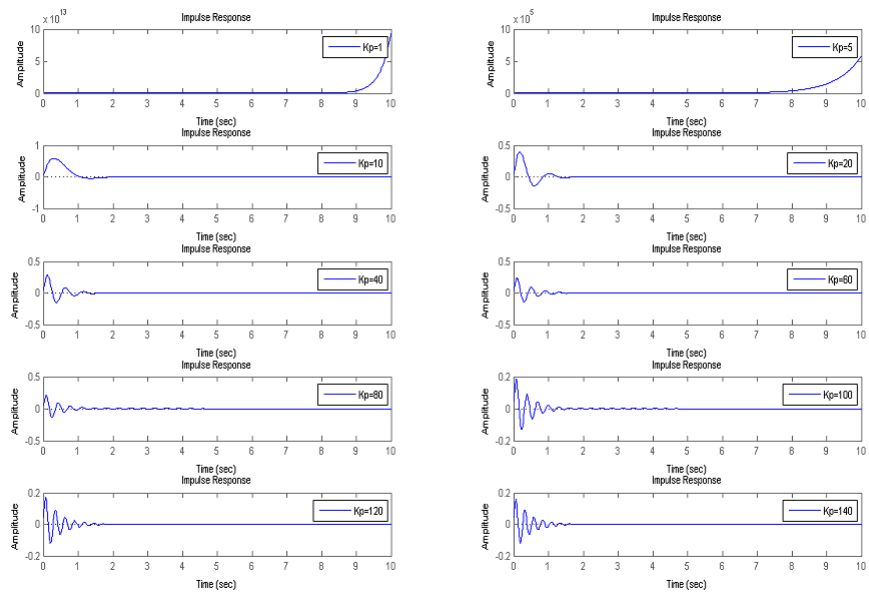


Figure 8: risposte impulsive al variare di  $K_p$

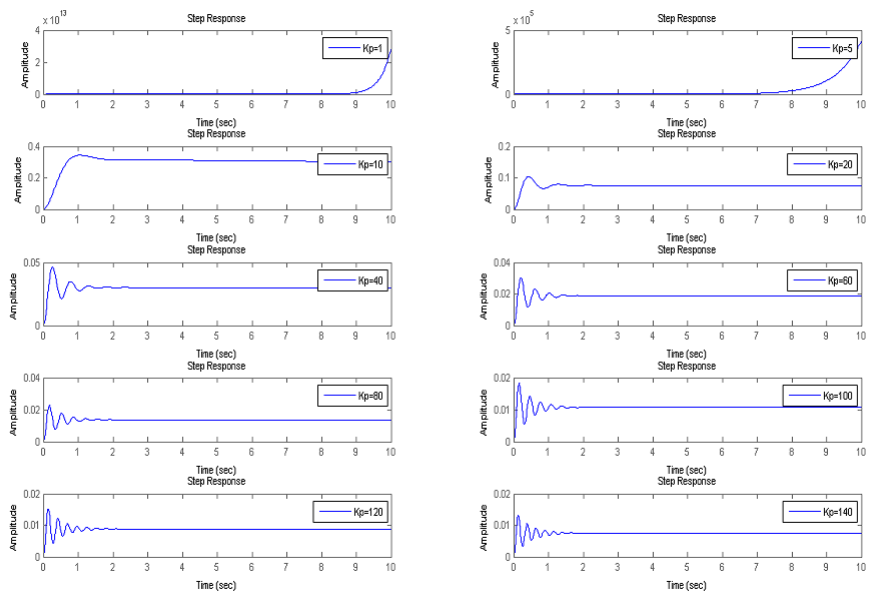


Figure 9: risposte al gradino al variare di  $K_p$

Osservando i grafici delle risposte si evince che la stabilità viene raggiunta già a partire da valori di  $K_p$  maggiori di 10; A questo punto scelgo  $K_p=140$  in quanto permette di avere oltre alla stabilità anche il raggiungimento, in entrambi i casi, della specifica riguardante il tempo di assestamento e nel caso specifico della risposta al gradino di avere anche l'altra specifica soddisfatta.

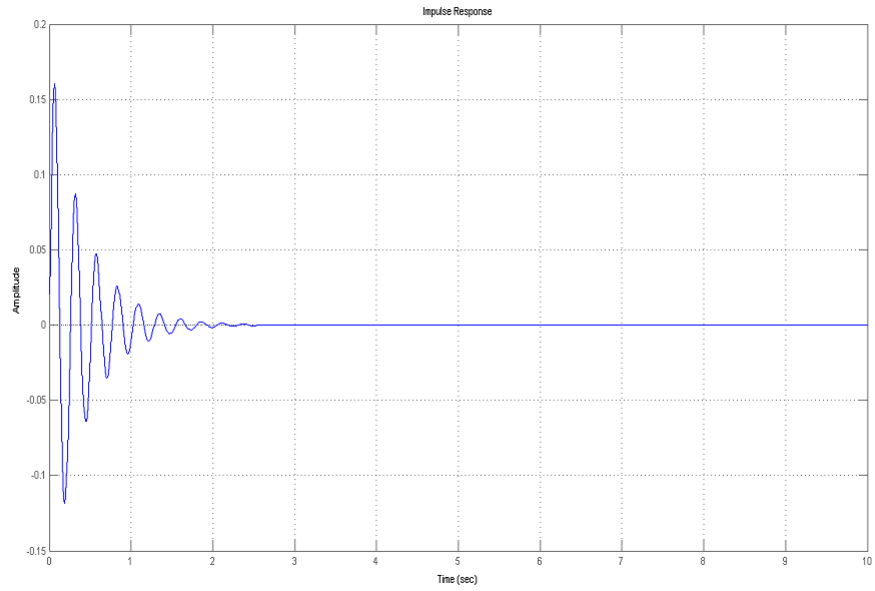


Figure 10: risposta impulsiva con  $K_p=140$

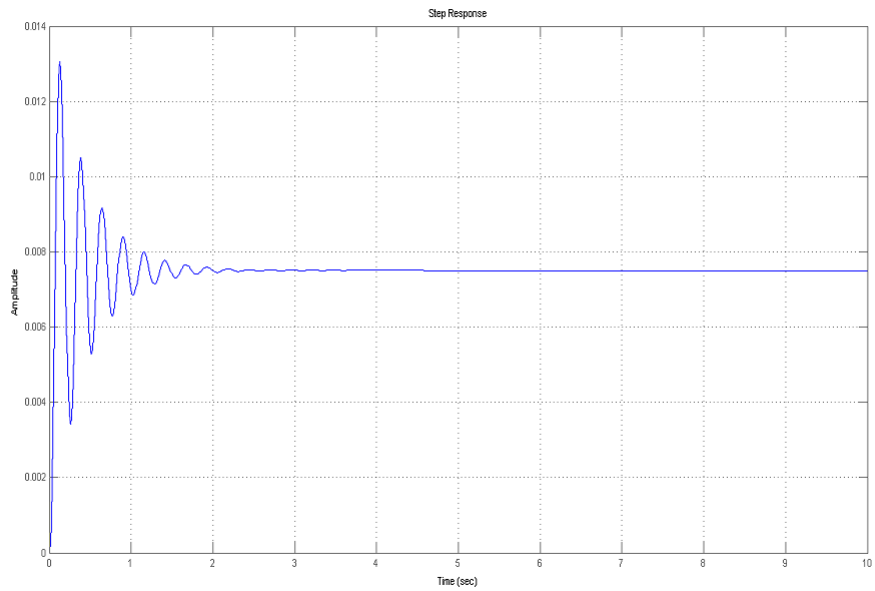


Figure 11: risposta al gradino con  $K_p=140$

A prima vista può sembrare che la risposta al gradino, che in ogni caso rispetta già le specifiche di progetto, non sia convergente a 0 ma abbia un er-

rore costante: questo non è vero in quanto se calcoliamo i poli del sistema in catena chiusa (con il comando `pole(pend_fback)`) otteniamo i seguenti risultati:  $p_{1,2} = -2.3636 \pm 24.4866i$ ;  $p_3 = -0.0002$  i poli hanno tutti parte reale negativa e quindi il sistema è stabile, solo che a causa di  $p_3$  ha una dinamica molto lenta. La risposta impulsiva invece non rispetta la specifica che l'overshoot del sistema sia minore di 0.05rad e quindi per cercare di migliorare la dinamica del sistema proviamo a far variare la costante che dipende dalla derivata dell'errore  $K_d$ : Come per  $K_p$  voglio osservare la variazione delle risposte al variare di  $K_d$ : In questo codice MATLAB, il vettore  $K$  è stato riempito con numeri piccoli rispetto al valore di  $K$  nell'iterazione per determinare  $K_p$  in quanto elevati valori di  $K_d$  possono portare all'instabilità

```
K=[2 5 8 10 12 15 18 20 25 30];
for i=1:1:10
    Kp=140;
    Kd=K(i);
    Ki=1;
    t=0:0.01:10;
    contPID=tf([Kd Kp Ki],[1 0]);
    pend_fback=feedback(pend,contPID)
    subplot(5,2,i);
    impulse(pend_fback,t);
    legend(strcat('Kd= ',num2str(Kd)));
    subplot(5,2,i);
    step(pend_fback,t);
    legend(strcat('Kd= ',num2str(Kd)));
```

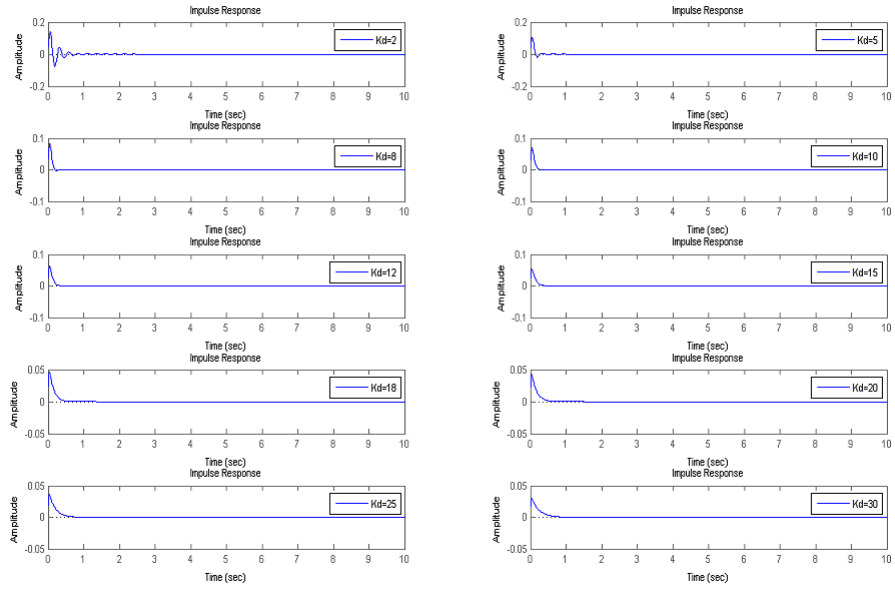


Figure 12: risposte impulsive al variare di  $K_d$

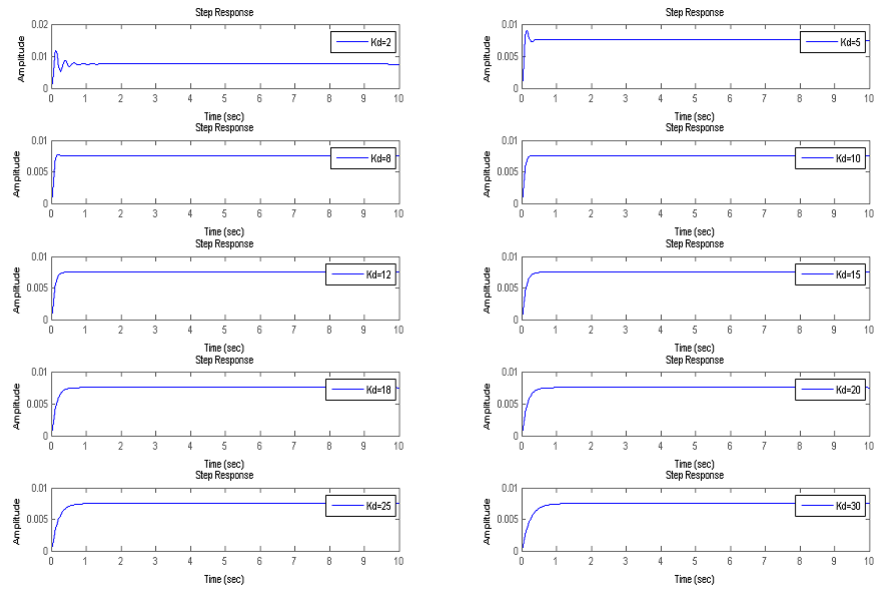


Figure 13: risposte al gradino al variare di  $K_d$

Sempre osservando i grafici si vede che sono sparite le oscillazioni da entrambe le risposte; Prendo come valore di  $K_d$  18 in quanto la risposta impul-



siva a partire da questo valore non presenta più oscillazioni, come si evince dal seguente grafico.

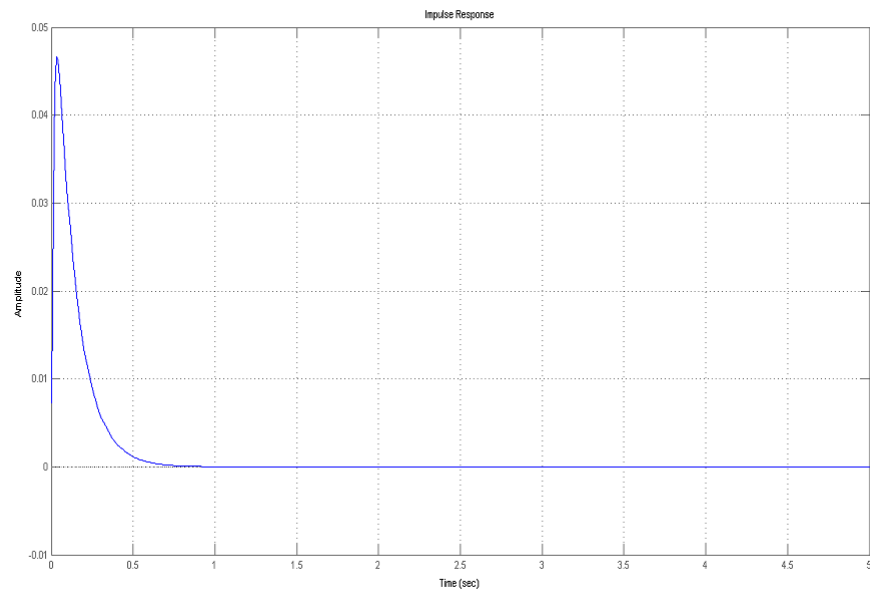


Figure 14: risposta impulsiva con  $K_p=140$  e  $K_d=18$

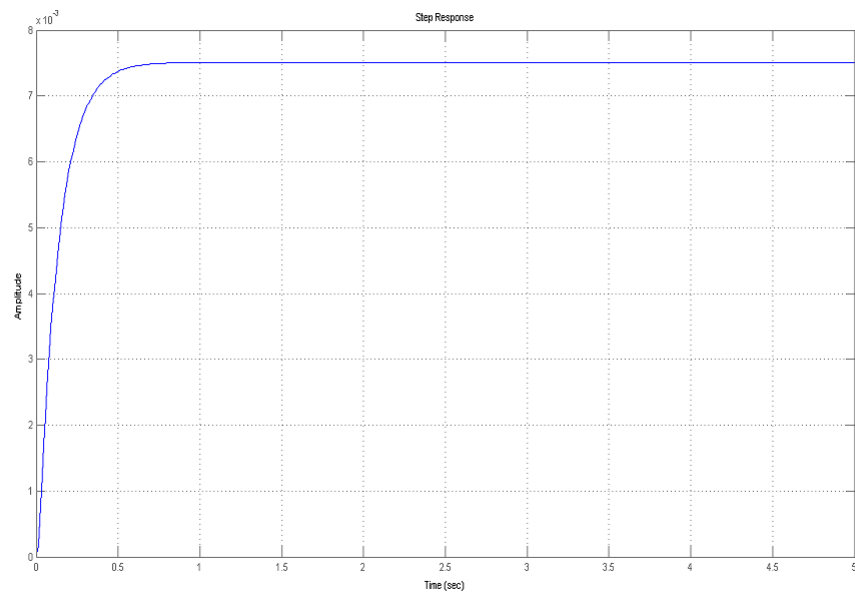


Figure 15: risposta al gradino con  $K_p=140$  e  $K_d=18$

Infine faccio variare la costante  $K_i$  questo perchè da come si può vedere dal grafico la risposta al gradino è ancora troppo "lenta" a scendere:

```
K=[2 5 10 12 15 18 20 25 30 40];
for i=1:1:10
    Kp=140;
    Kd=18;
    Ki=K(i);
    t=0:0.01:10;
    contPID=tf([Kd Kp Ki],[1 0]);
    pend_fback=feedback(pend,contPID)
    subplot(5,2,i);
    impulse(pend_fback,t);
    legend(strcat('Ki= ',num2str(Ki)));
    subplot(5,2,i);
    step(pend_fback,t);
    legend(strcat('Ki= ',num2str(Ki)));
```

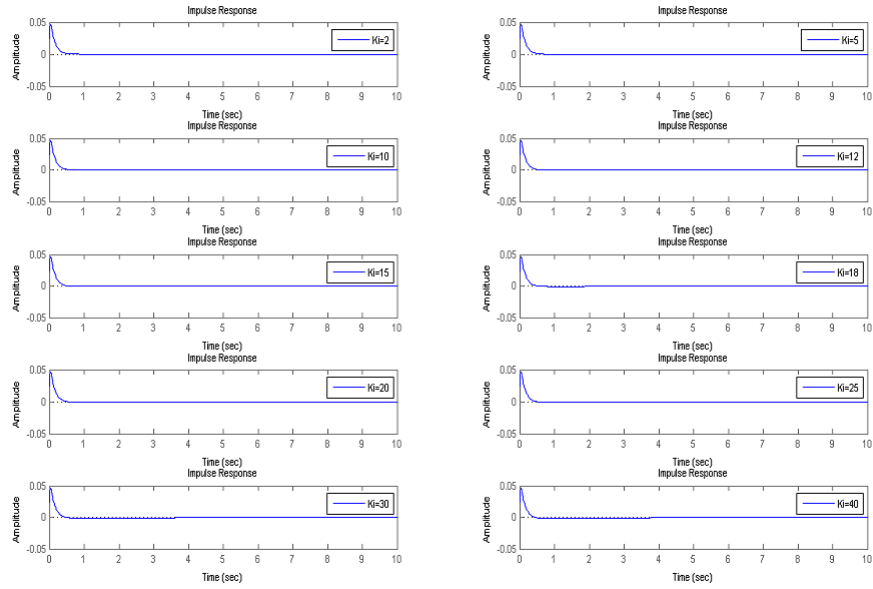


Figure 16: risposte impulsive al variare di  $K_i$

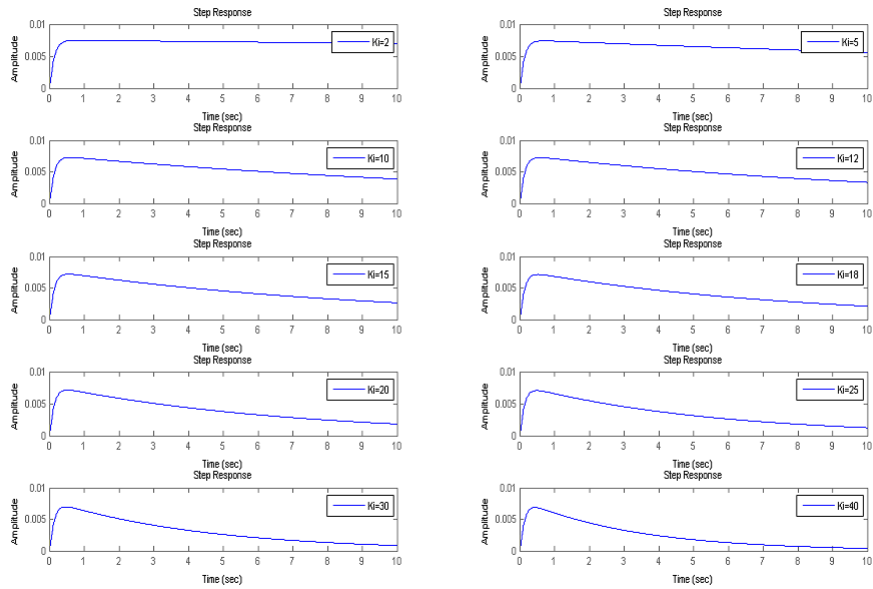


Figure 17: risposte al gradino al variare di  $K_i$

Si nota immediatamente che all'aumentare di  $K_i$  migliora molto sensibilmente la risposta al gradino quindi scelgo  $K_i=40$ . A questo punto visto che

ho definito i tre parametri posso definire il mio controllore:

$$C_{PID}(s) = \frac{18s^2 + 140s + 40}{s} \quad (13)$$

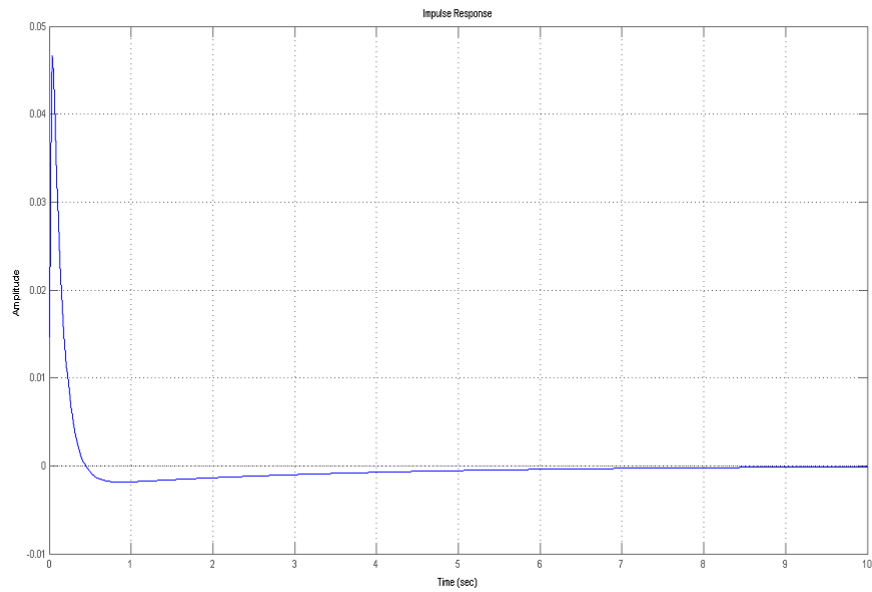


Figure 18: risposta impulsiva con  $C_{PID}(s)$

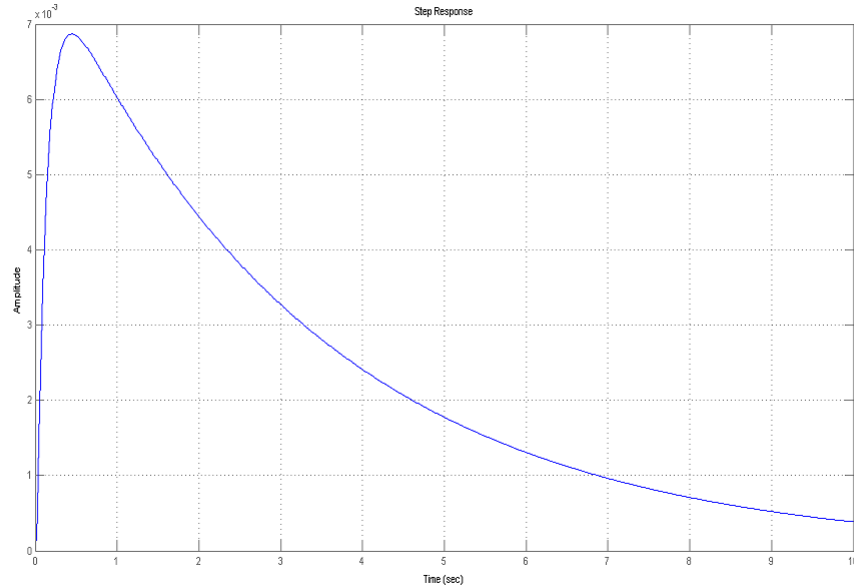


Figure 19: risposte al gradino con  $C_{PID}(s)$

### 3.2 Progetto di un controllore con il metodo della risposta in frequenza

Da come si evince dall'analisi del sistema in catena aperta, la f.d.t. del sistema ha un polo nel semipiano destro del piano complesso (cioè è instabile) di conseguenza non posso utilizzare il metodo di Bode per trovare un controllore che lo stabilizzi; Non potendo utilizzare quindi il metodo di Bode cerco di stabilizzare il sistema utilizzando il *Criterio di Nyquist* cioè cercando di trovare un controllore tale da soddisfare la condizione:

$$N = P - Z \quad (14)$$

dove:

- $N$  è il numero di giri in senso antiorario che il diagramma di Nyquist fa attorno al punto -1
- $Z$  è il numero di zeri instabili (che corrispondono ai poli instabili del sistema ad anello chiuso)
- $P$  è il numero di poli instabili

Nel nostro caso abbiamo che  $Z = 0$  e  $P = 1$ , ne consegue che  $N = -1$  cioè il diagramma di Nyquist dovrà compiere 1 giro in senso orario attorno al punto

-1 per avere un sistema ad anello chiuso stabile. Utilizzando MATLAB disegno i diagrammi di Nyquist e Bode:

```
subplot(1,2,1);
nyquist(pend);
subplot(1,2,2);
bode(pend);
grid on;
```

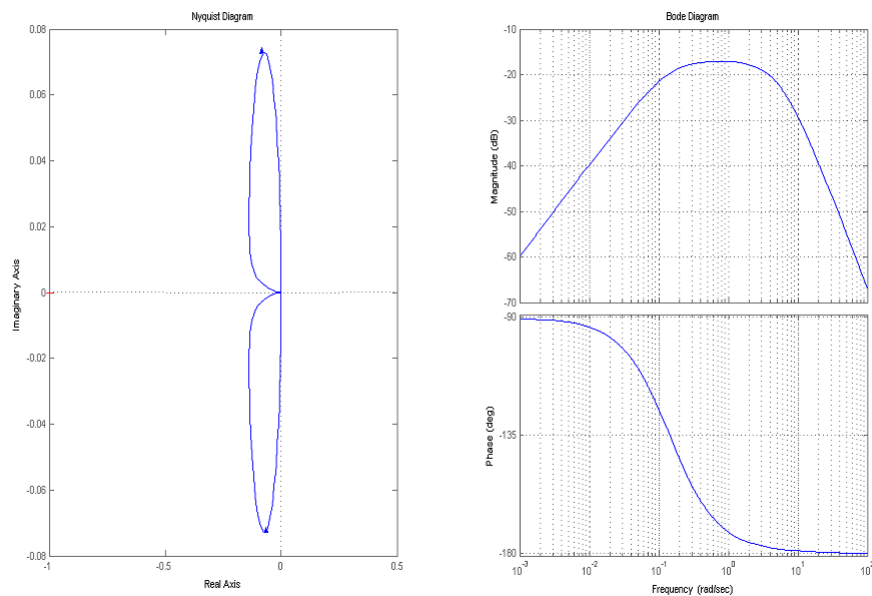


Figure 20: Diagrammi di Nyquist e Bode del sistema

Osservando il diagramma di Nyquist si notano immediatamente due fatti:  $\lim_{j\omega \rightarrow 0} |G(j\omega)| = 0$  e  $\lim_{j\omega \rightarrow +\infty} |G(j\omega)| = 0$  ovvero il diagramma inizia e finisce nell'origine, inoltre il diagramma non gira attorno a nessun punto dell'asse reale. Si può notare che la funzione di trasferimento  $G(s)$  ha uno zero nell'origine, senza il quale il diagramma non inizierebbe più dall'origine ma da un altro punto, mentre per  $j\omega \rightarrow \infty$  il diagramma tenderà ancora a 0. Definisco quindi un compensatore "grezzo"  $C_{freq}(s) = \frac{1}{s}$  in modo da togliere lo zero nell'origine a numeratore, e con MATLAB calcolo il nuovo diagramma di nyquist della f.d.t. (ad anello aperto)  $C_{freq}(s)G(s)$ :

```
cont_freq=tf([1],[1 0]);
pend_OL=pend*cont_freq; %fdt sistema open loop C*G
subplot(1,2,1);
nyquist(pend_OL);
```

```
subplot(1,2,2);
bode(pend_OL);
grid on;
```

di seguito i risultati:

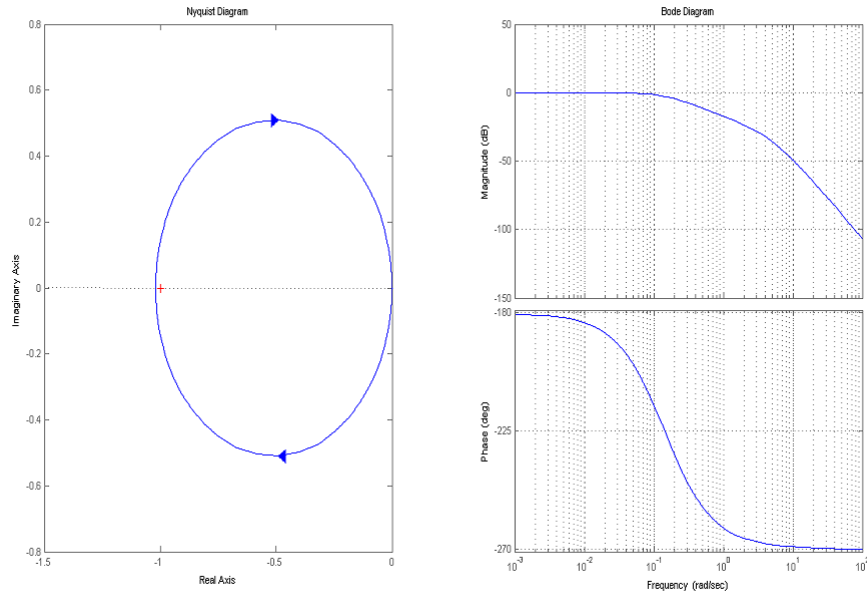


Figure 21: Diagrammi di Nyquist e Bode del sistema più compensatore

Si nota immediatamente che il diagramma di Nyquist è diventato di forma ellittica e, diversamente da prima ruota attorno al punto -1 (rappresentato da una crocetta sull'asse reale), però ci ruota in senso orario e quindi, dal criterio di Nyquist  $Z=P-Z=1+1=2$  cioè, il sistema ad anello chiuso ha ben due poli instabili. Con MATLAB è facilmente verificabile che il sistema, con compensatore, ad anello chiuso ha effettivamente due poli positivi:

```
s=feedback(pend,cont_freq);
pole(s);
```

Dal calcolo giustamente risultano due poli a parte reale positiva rispettivamente in 5.4924 e 0.0029 (il polo negativo è in -5.6772). A questo punto, avendo aggiunto un polo nell'origine ho ottenuto un diagramma di Nyquist che è più facile da manipolare rispetto al diagramma di partenza. Devo aggiungere fase cioè devo far in modo da creare un cerchio in senso antiorario; di conseguenza modifico il controllore in questo modo aggiungendogli uno zero:

$$C_{freq}(s) = \frac{s+1}{s} \quad (15)$$

Allo stesso modo di prima controllo con MATLAB come è si è modificato il diagramma:

```
cont_freq=tf([1 1],[1 0]);  
pend_OL=pend*cont_freq; %fdt sistema open loop C*G  
subplot(1,2,1);  
nyquist(pend_OL);  
subplot(1,2,2);  
bode(pend_OL);  
grid on;
```

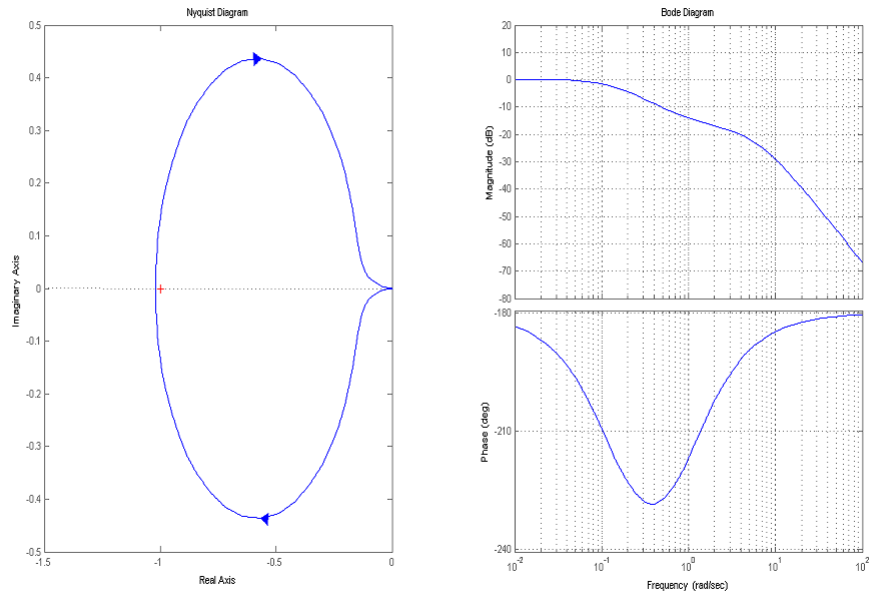


Figure 22: Diagrammi di Nyquist e Bode del sistema più compensatore



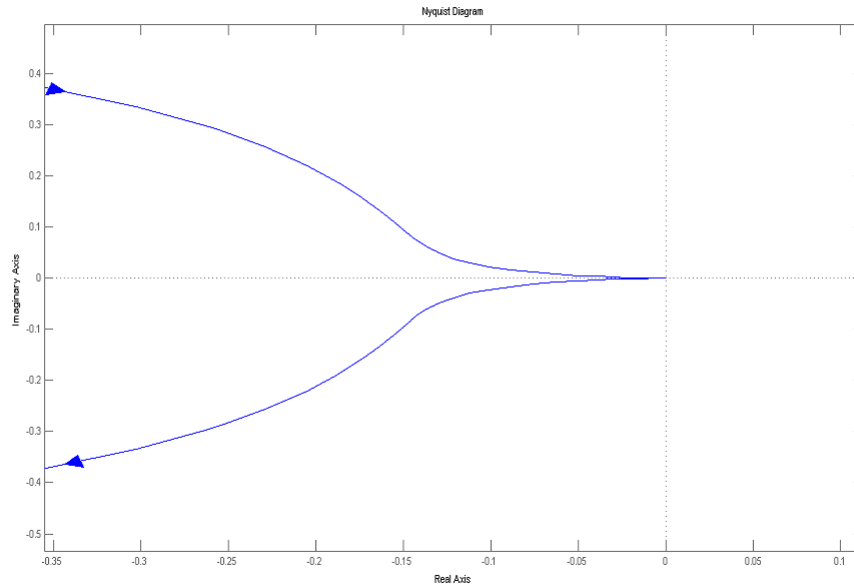


Figure 23: Dettaglio del diagramma di Nyquist

Guardando il grafico si vede subito che siamo ancora allo stesso punto di prima (cioè con 2 poli a parte reale positiva) ma se osserviamo in dettaglio il grafico vicino all'origine si nota che i due rami del diagramma sono più "schacciati" verso l'asse reale; A questo punto proviamo ad aggiungere ancora fase aggiungendo un'altro zero nel compensatore:

$$C_{freq}(s) = \frac{(s+1)(s+1)}{s} \quad (16)$$

Controlliamo con MATLAB il risultato:

```
cont_freq=tf(conv([1 1],[1 1]),[1 0]);
pend_OL=pend*cont_freq; %fdt sistema open loop C*G
subplot(1,2,1);
nyquist(pend_OL);
subplot(1,2,2);
bode(pend_OL);
grid on;
```

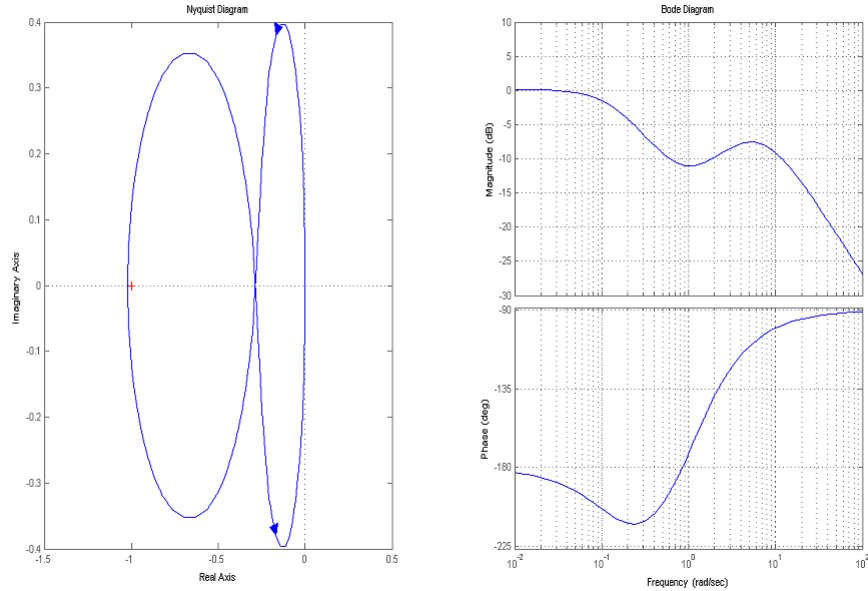


Figure 24: Diagrammi di Nyquist e Bode del sistema più compensatore

Anche in questo caso ci sono 2 poli instabili ma a differenza di prima, se trasliamo il diagramma a sinistra, il diagramma percorrerà un giro in senso antiorario intorno al punto -1 e quindi non ci saranno più poli instabili nel sistema in catena chiusa. Per poter traslare il diagramma devo moltiplicare il compensatore per una costante  $K$  reale:

$$C_{freq}(s) = K \frac{(s+1)(s+1)}{s} \quad (17)$$

$K$  deve soddisfare la seguente disequazione:

$$-0.285 \leq \frac{1}{K} \leq 0 \quad (18)$$

che risulta dà la condizione

$$K > \frac{1}{0.285} = 3.5 \quad (19)$$

Il valore -0.285 è stato determinato dal diagramma di nyquist e corrisponde (circa) al punto nell'asse reale dove si incontrano i due rami del diagramma:

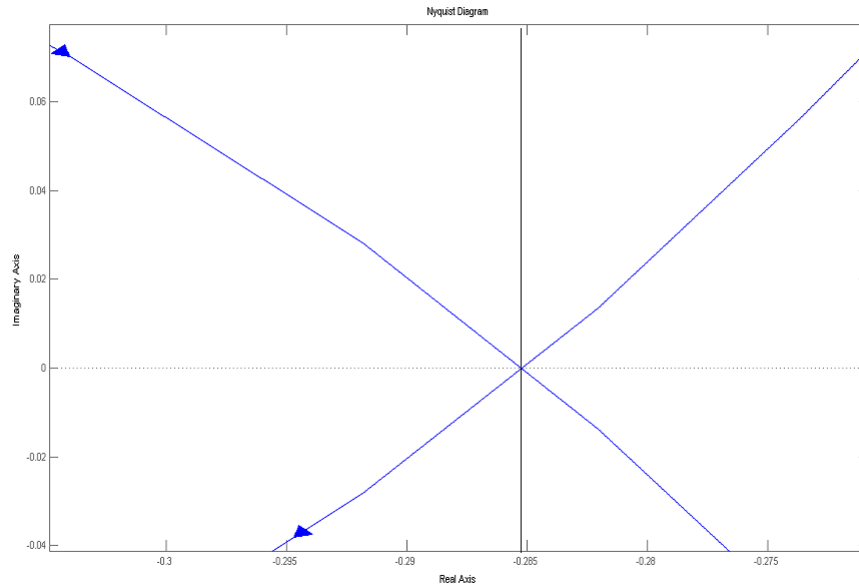


Figure 25: Dettaglio del diagramma di Nyquist

Pongo  $K=8$  (un numero a caso) e calcolo il nuovo diagramma di Nyquist con

$$C_{freq}(s) = 4 \frac{(s+1)(s+1)}{s} \quad (20)$$

```
cont_freq=8*tf(conv([1 1],[1 1]),[1 0]);
pend_OL=pend*cont_freq; %fdt sistema open loop C*G
subplot(1,2,1);
nyquist(pend_OL);
subplot(1,2,2);
bode(pend_OL);
grid on;
```

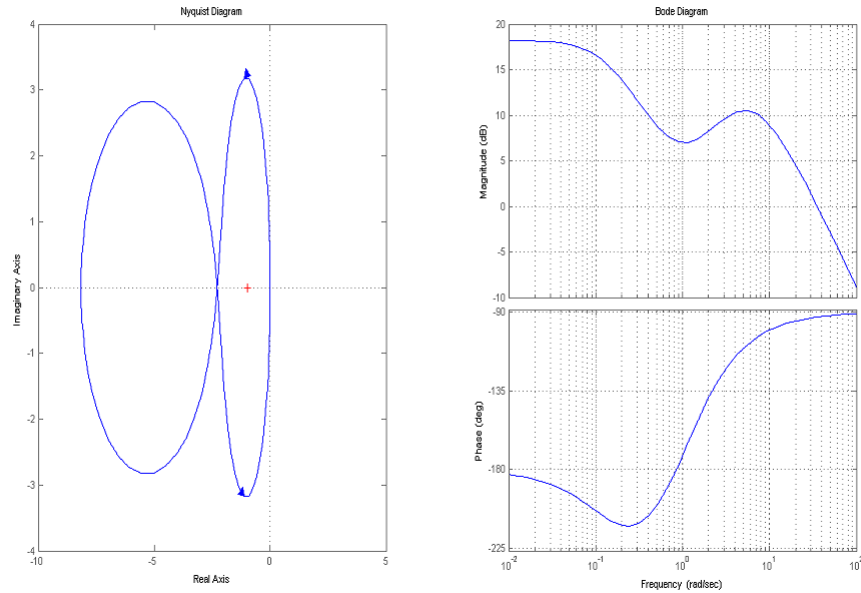


Figure 26: Diagrammi di Nyquist e Bode del sistema più compensatore

La curva ora percorre un giro in senso antiorario attorno al punto -1, ne consegue che, per il criterio di nyquist il sistema in catena chiusa è stabile infatti  $Z=P-Z=1-1=0$ : non ci sono poli instabili in catena chiusa. Si verifica la stabilità del sistema anche con il calcolo dei poli del sistema in catena chiusa:

```
cont_freq=8*tf(conv([1 1],[1 1]),[1 0]);
pend_fback=feedback(pend,cont_freq);
pole(pend_fback)
```

che da come risultato  $p_1 = -35.4$ ,  $p_{2,3} = -0.5741 \pm j0.7562$  che sono evidentemente poli stabili. Non ci resta altro da fare che provare a calcolare la f.d.t. del sistema in catena chiusa e vedere l'evoluzione delle risposte indicali:

```
cont_freq=8*tf(conv([1 1],[1 1]),[1 0]);
pend_fback=feedback(pend,cont_freq);
impulse(pend);
figure();
step(pend);
```

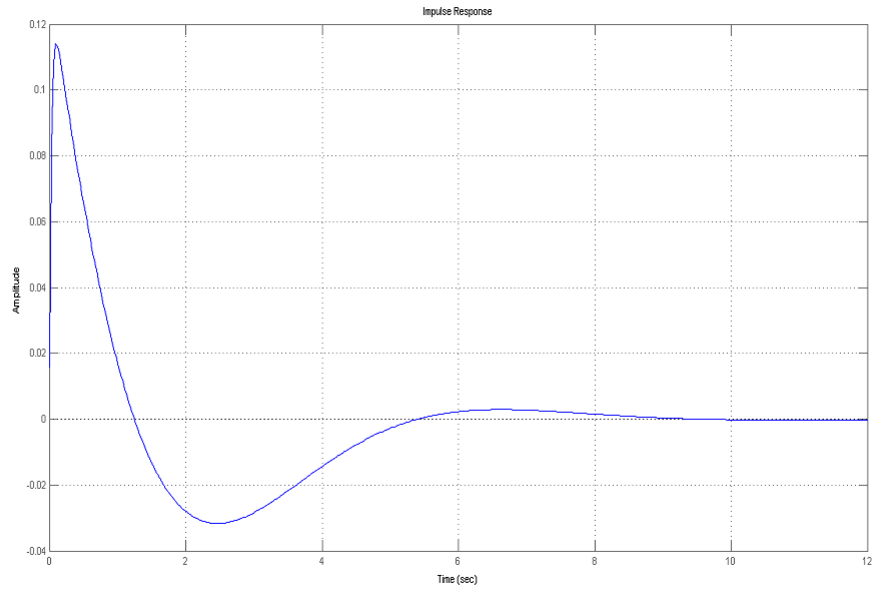


Figure 27: Risposta impulsiva del sistema in catena chiusa

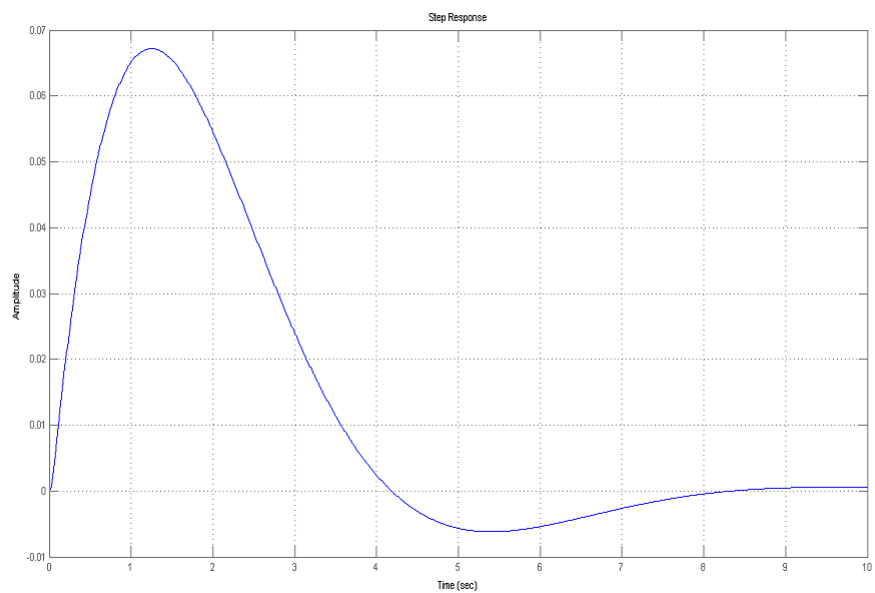


Figure 28: Risposta al gradino del sistema in catena chiusa

Dai grafici delle risposte si vede subito che esse non rispettano nessuna delle due specifiche in quanto non è verificata la specifica sull'overshoot e il setting time è ben dopo i 5s richiesti dalle specifiche ; occorre quindi modificare la posizione dei due zeri aggiunti al controllore. Per fare queste modifiche, ho scelto di utilizzare Sisotool un tool di matlab che consente di eseguire una manipolazione grafica dei zeri e poli del compensatore e vedere in tempo reale gli effetti che queste modifiche comportano sull'evoluzione del sistema.

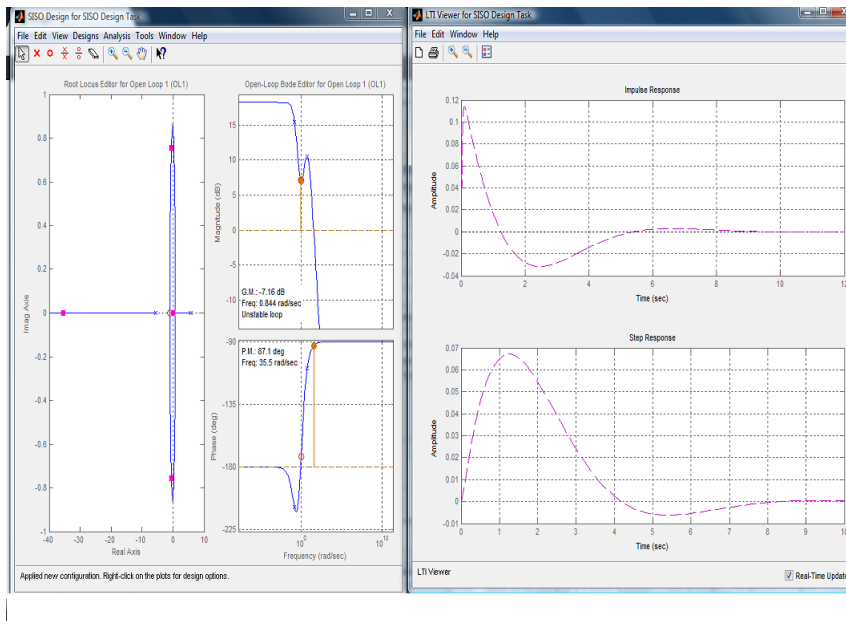


Figure 29: Tipica schermata di SISOTOOL: a sinistra la finestra per la modifica grafica e a destra la risposta impulsiva e al gradino in tempo reale

Una volta scelta dalla schermata iniziale di sisotool l'architettura del sistema e caricato al suo interno le funzioni di trasferimento ( $G(s)$ ) e il controllore trovato con il metodo della risposta in frequenza) si può scegliere "show design plot" nel menù Graphical Tuning, comparirà la finestra per la modifica grafica che conterrà un diagramma di Bode e un luogo delle radici. Per vedere in tempo reale l'andamento delle risposte indiciali si entra nella scheda "Analysis Plot" sempre dal menù principale e una volta selezionati i grafici che interessano (impulse response e step response) si sceglie il tipo di risposta desiderata che nel nostro caso è "Closed Loop-From du to y" cioè voglio visualizzare la risposta impulsiva e al gradino dell'anello chiuso che va dal disturbo  $du(F)$  nel nostro caso) all'uscita  $y(\theta)$ . L'utilità di questo tool di matlab è quella di permettere l'inserimento, rimozione e spostamento di zeri e poli in maniera immediata semplicemente cliccando sul grafico e vedere sull'altra finestra il risultato; trascinando il diagramma di Bode verso l'alto o verso il basso è possibile incrementare o diminuire la costante  $K$ . Spostando

i zeri e aumentando un po la costante K ottenendo il seguente risultato:

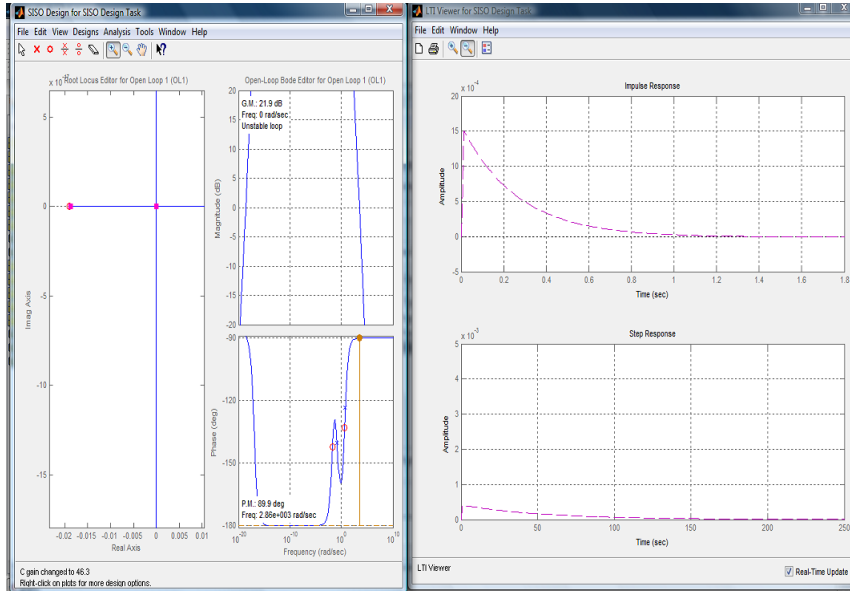


Figure 30: Risposte in seguito allo spostamento degli zeri del compensatore

guardando nella scheda "Compensator Editor" è possibile vedere i parametri del nuovo compensatore che è:

$$C_{freq}(s) = 638.4136 \frac{(s + 3.846)(s + 0.01887)}{s} = \frac{638.4s^2 + 2467s + 46.33}{s} \quad (21)$$

A questo punto andiamo ad analizzare in dettaglio il sistema con compensatore:

```
cont_freq=46.329*tf(conv([53 1],[0.26 1]),[1 0]);
pend_fback1=feedback(pend,cont_freq);
impulse(pend_fback1);
figure();
step(pend_fback1);
```

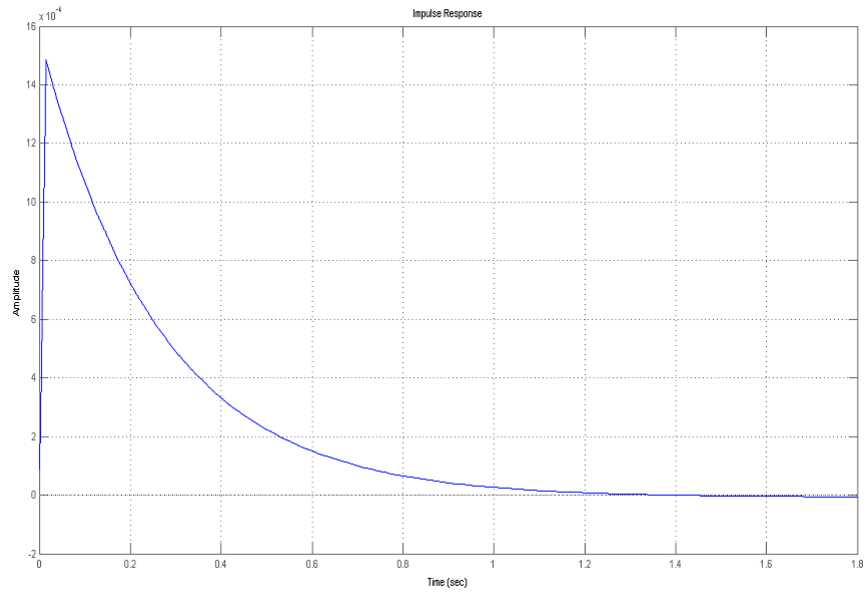


Figure 31: Risposte impulsiva del sistema con compensatore  $C_{freq}(s)$

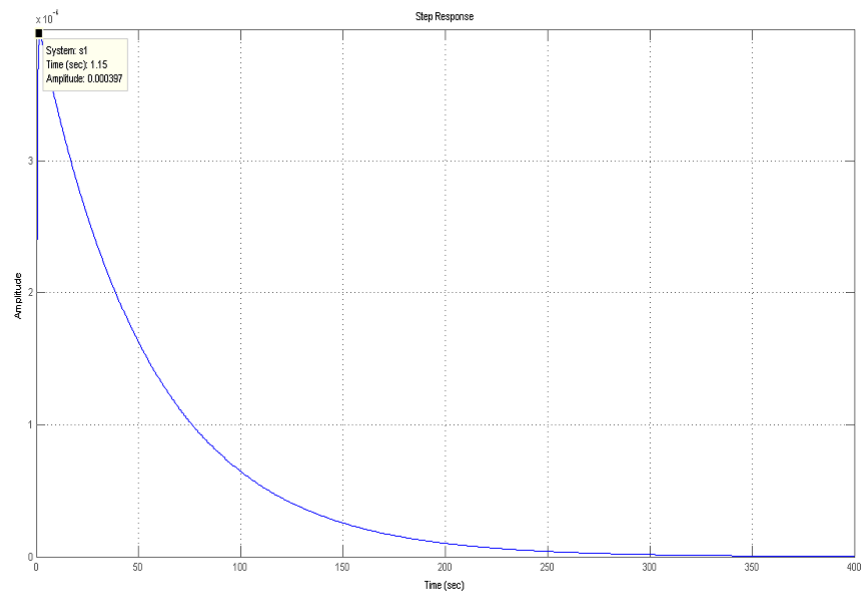


Figure 32: Risposte al gradino del sistema con compensatore  $C_{freq}(s)$



Da come si evince dai grafici il sistema rispetta ampiamente le specifiche di progetto. Di seguito vengono rappresentati con MATLAB i diagrammi di Bode per ogni modifica che è stata fatta al compensatore:

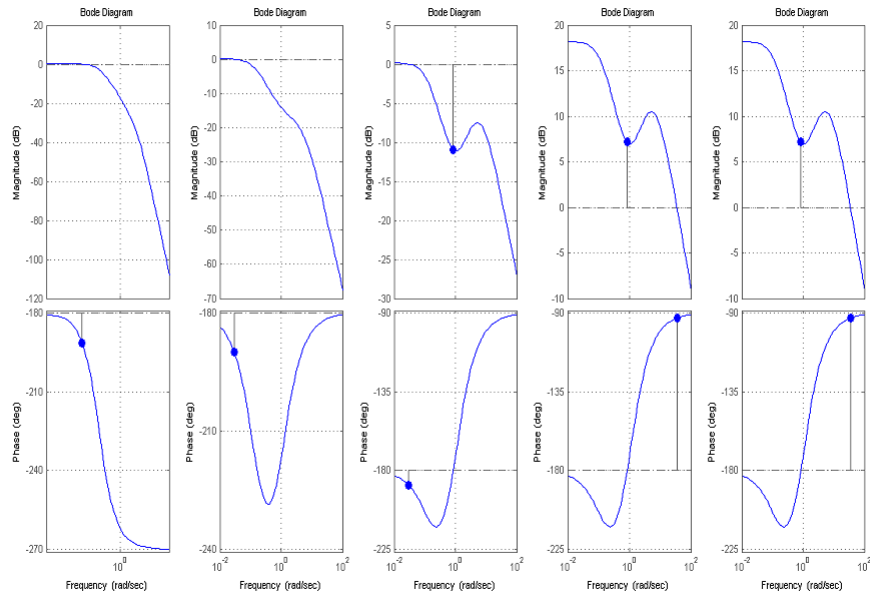


Figure 33: Diagrammi di Bode al variare del compensatore  $C_{freq}(s)$

### 3.3 Progetto di un controllore con il metodo del luogo delle radici

Per poter progettare un controllore con il metodo del luogo delle radici è necessario, come prima cosa disegnare il luogo delle radici.

```
rootlocus(pend); %disegna il luogo delle radici di pend(G)
```

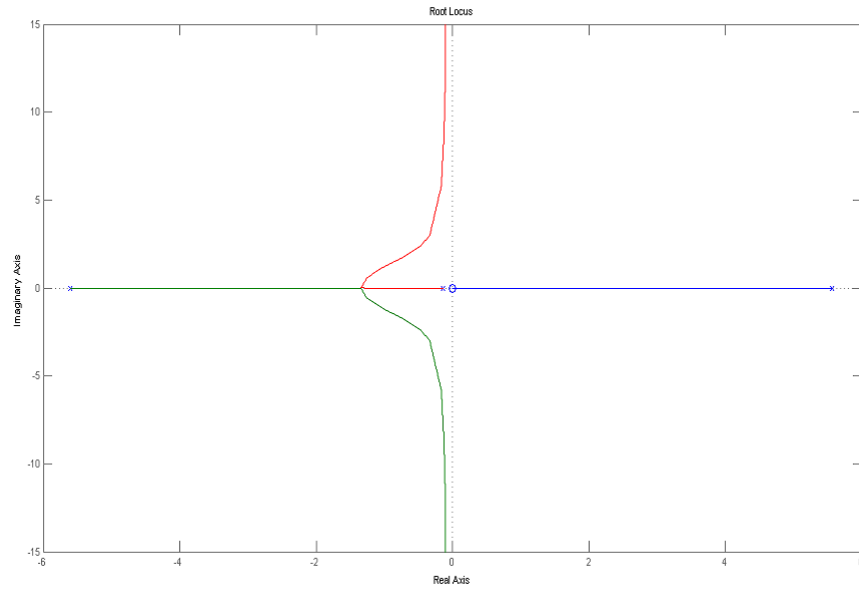


Figure 34: Luogo delle radici di  $G(s)$

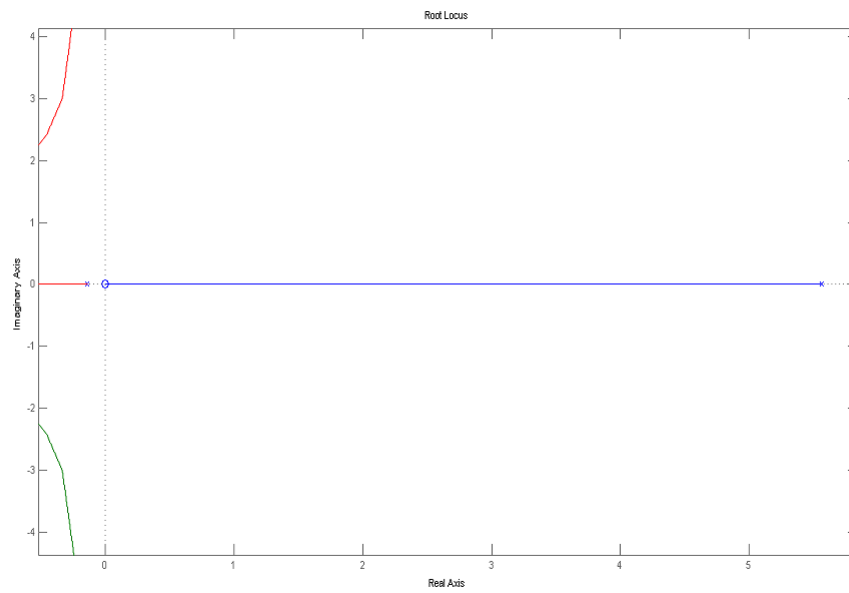


Figure 35: Dettaglio del semipiano destro del luogo delle radici di  $G(s)$

Da come si può vedere dal grafico, in particolare in Fig, il luogo delle radici è formato da un ramo che per ogni scelta della costante  $K > 0$  (guadagno) rimarrà sempre nel semipiano destro e quindi quello sarà sempre un polo instabile. Di conseguenza non esiste nessun  $K$  tale che il sistema diventi stabile. Per stabilizzare questo sistema scelgo di utilizzare una rete a sella (Lead-Lag controller) che è una rete del tipo:

$$C_{rloc}(s) = K_c \frac{(s + z_1)(s + z_2)}{(s + p_1)(s + p_2)} = K_c \frac{(s + z_1)}{(s + p_1)} \frac{(s + z_2)}{(s + p_2)} = K_c C_{ant}(s) C_{rit}(s) \quad (22)$$

dove  $|z_1| < |p_1|$  e  $|z_2| > |p_2|$

cioè è il prodotto di una rete anticipatrice con una rete ritardatrice. Una rete anticipatrice tende a spostare il luogo verso il semipiano sinistro e quindi di conseguenza aumenta la stabilità e la risposta del sistema; mentre una rete ritardatrice tende a spostare il luogo verso destra, il che non è molto desiderabile, e quindi il suo polo e il suo zero devono essere messi molto vicini tra di loro, di solito vicino all'origine. Però una rete ritardatrice ha anche il vantaggio di migliorare l'errore (steady-state response) questo perché per frequenze molto alte,  $C_{rit} \rightarrow 1$  mentre per frequenze prossime allo zero tende a  $\frac{z_0}{p_0}$  che è  $> 1$ . Procediamo quindi in questo modo: prima posizioniamo i poli e gli zeri delle reti correttive e poi, se non sono rispettate le specifiche procediamo a ottimizzare il compensatore con Sisotool. Posiziono i poli e zeri nei seguenti punti:  $z_1 = -4$ ;  $p_1 = -100$ ;  $z_2 = -3$ ;  $p_2 = 0$  e provo a vedere come cambia il luogo delle radici:

$$C_{rloc}(s) = K_c \frac{(s + 4)(s + 3)}{(s + 100)(s)} \quad (23)$$

```
cont_rloc=tf(conv([1 3],[1 4]),[1 100 0]);
rlocus(cont_rloc*pend);
```

risulta:

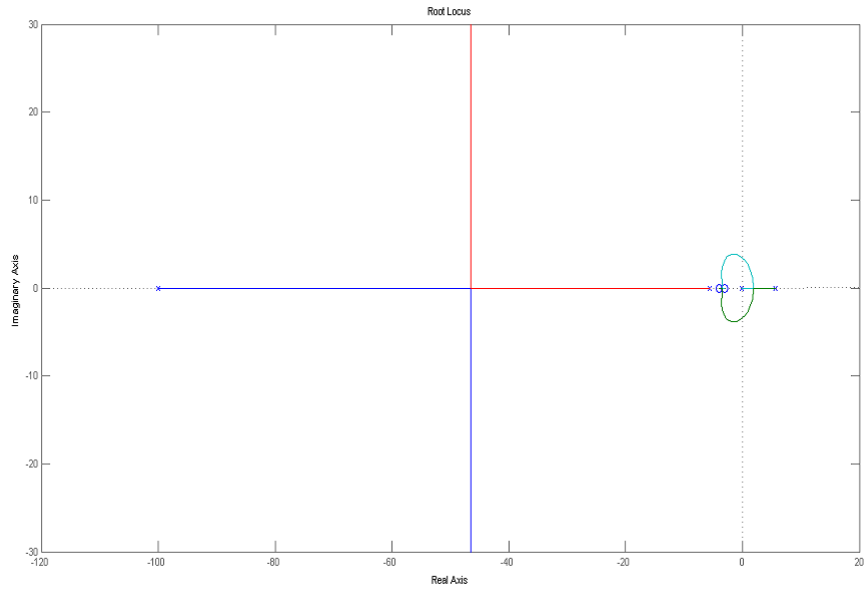


Figure 36: Luogo delle radici di  $C_{rloc}(s)G(s)$

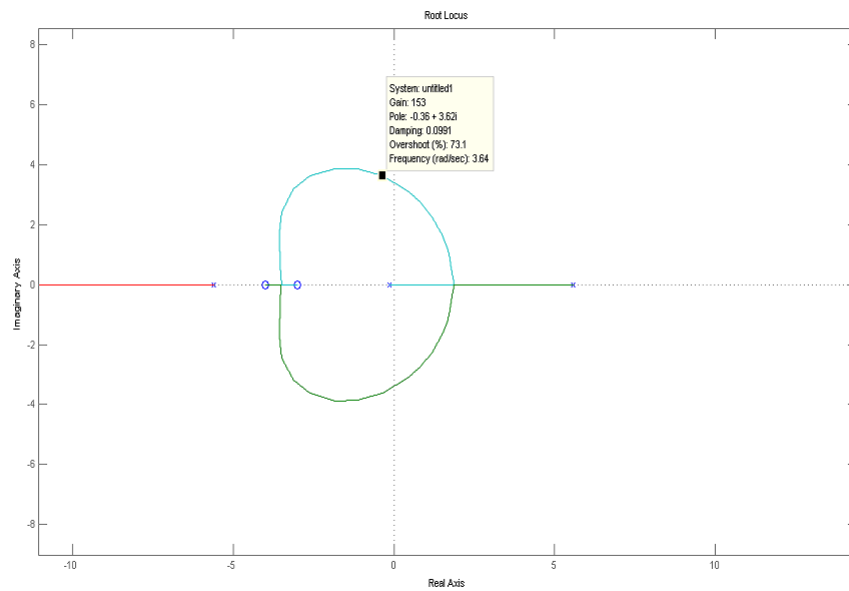


Figure 37: Dettaglio in un intorno dell'origine di  $C_{rloc}(s)G(s)$

Ora il luogo delle radici contiene sicuramente delle radici stabili per un qualche  $K$ ; Guardando in dettaglio il luogo delle radici si osserva che per un  $K > 153$  tutti i poli del sistema sono poli stabili; non resta altro che vedere se il controllore soddisfa i requisiti:

```
s_rloc=feedback(pend,200*cont_rloc);  
impulse(s_rloc);  
figure;  
step(s_loc);
```

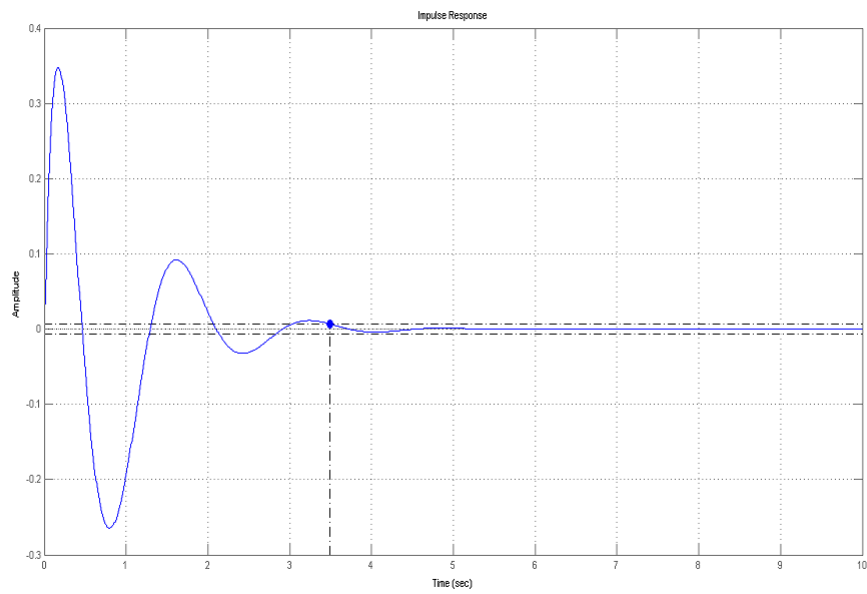


Figure 38: Risposta impulsiva del sistema con  $K=200$

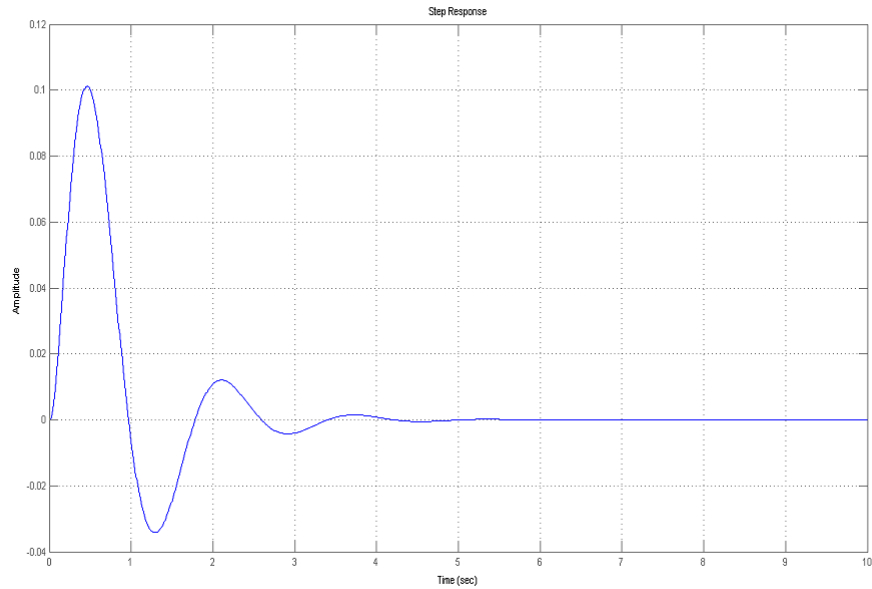


Figure 39: Risposta al gradino del sistema con  $K=200$

Si nota subito che il controllore fa sì che il sistema rispetti la specifica riguardante il tempo di assestamento, ma però non rispetta per niente la specifica sulla sovralongazione. Provo a vedere quindi cosa succede all'aumentare di  $K$ .

```
K=[300 500 1000 1500 2000 2500 3000 4000]
for i=1:1:8
    s_rloc=feedback(pend,K(i)*cont_rloc);
    subplot(4,2,i);
    impulse(s_rloc,0:0.01:10);
    legend(strcat('K= ',num2str(K(i))));
end;
figure;
for i=1:1:8
    s_rloc=feedback(pend,K(i)*cont_rloc);
    subplot(4,2,i);
    step(s_rloc,0:0.01:10);
    legend(strcat('K= ',num2str(K(i))));
end;
```

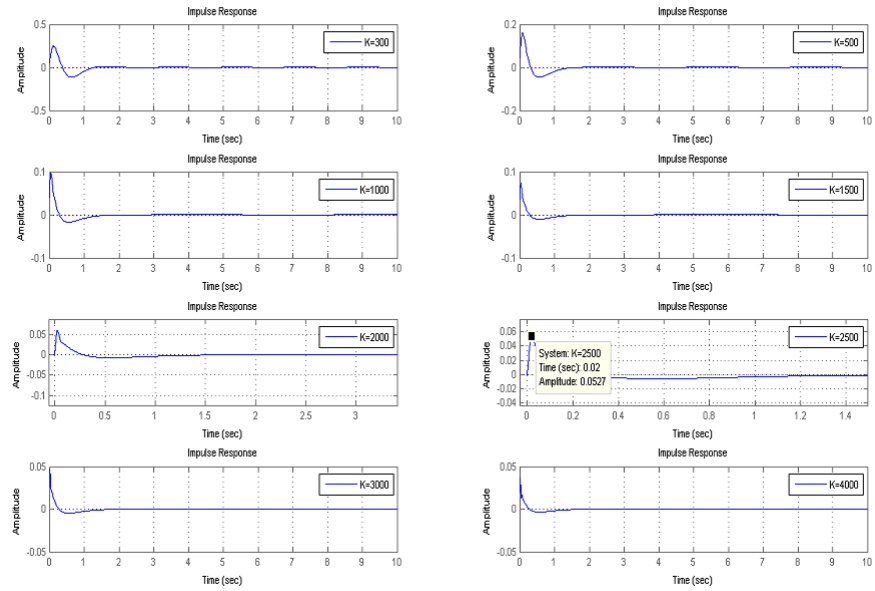


Figure 40: Risposte impulsive al variare di K

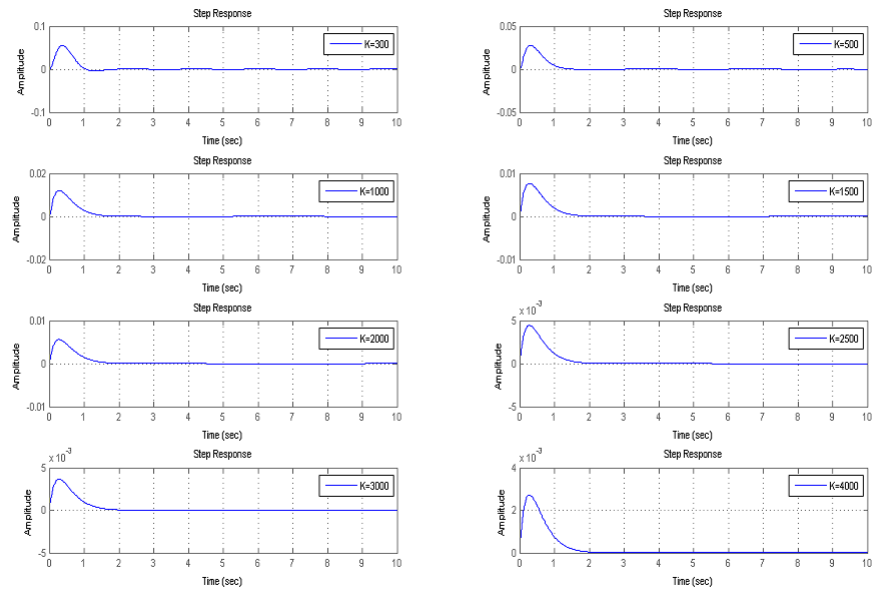


Figure 41: Risposte al gradino al variare di K

Dai grafici si vede che la risposta al gradino rispetta le specifiche per già per  $K=500$  mentre per quanto riguarda la risposta impulsiva  $K$  deve crescere fino a 3000. Quindi scegliendo  $K=3000$  il controllore diventa:

$$C_{rloc}(s) = 3000 \frac{(s+4)(s+3)}{s(s+100)} \quad (24)$$

In seguito calcolo le risposte indiciali per  $K=3000$ :

```
s_rloc=feedback(pend,3000*cont_rloc);
impulse(s_rloc);
figure;
step(s_loc);
```

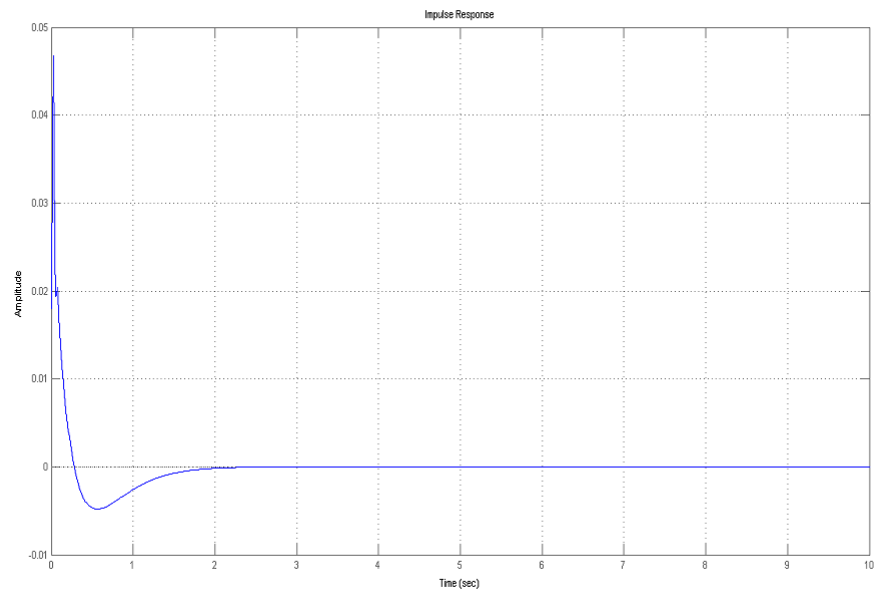


Figure 42: Risposta impulsiva con  $K=3000$



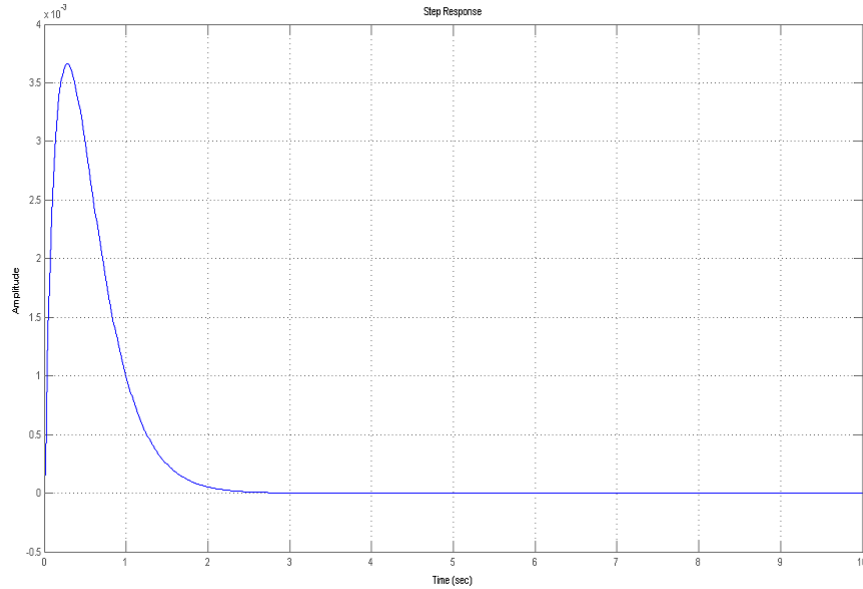


Figure 43: Risposta al gradino con  $K=3000$

## 4 Confronto e verifica dei risultati

### 4.1 Confronto delle risposte impulsive e al gradino

Con i metodi sopra descritti abbiamo trovato 3 tipi di controllori distinti:  
 Controllore PID:

$$C_{PID}(s) = \frac{18s^2 + 140s + 40}{s} \quad (25)$$

Controllore trovato con il metodo della risposta in frequenza:

$$C_{freq}(s) = 638.4136 \frac{(s + 3.846)(s + 0.01887)}{s} \quad (26)$$

Controllore trovato con il metodo del luogo delle radici:

$$C_{rloc}(s) = 3000 \frac{(s + 4)(s + 3)}{(s + 100)(s)} \quad (27)$$

Dal punto di vista della complessità (intesa come complessità della funzione di trasferimento) si nota che il controllore PID e quello trovato con il metodo della risposta in frequenza hanno entrambi un polinomio di 2 grado al numeratore e un polo nell'origine a denominatore mentre il controllore trovato con il metodo del luogo delle radici è il più complesso, infatti ha un polinomio di secondo ordine sia al numeratore che al denominatore. A questo

punto vogliamo fare un confronto delle prestazioni dei tre compensatori; per far questo andremo a disegnare la risposta impulsiva e al gradino dei tre compensatori sovrapposta:

```
impulse(s_pid,':',s_nyq,'--',s_rloc,'-')
grid on
legend('PID','nyquist','root locus');
figure;
step(s_pid,':',s_nyq,'--',s_rloc,'-')
grid on
legend('PID','nyquist','root locus');
```

Come prima cosa valutiamo gli andamenti della risposta impulsiva del sistema al variare del tipo di compensatore utilizzato:

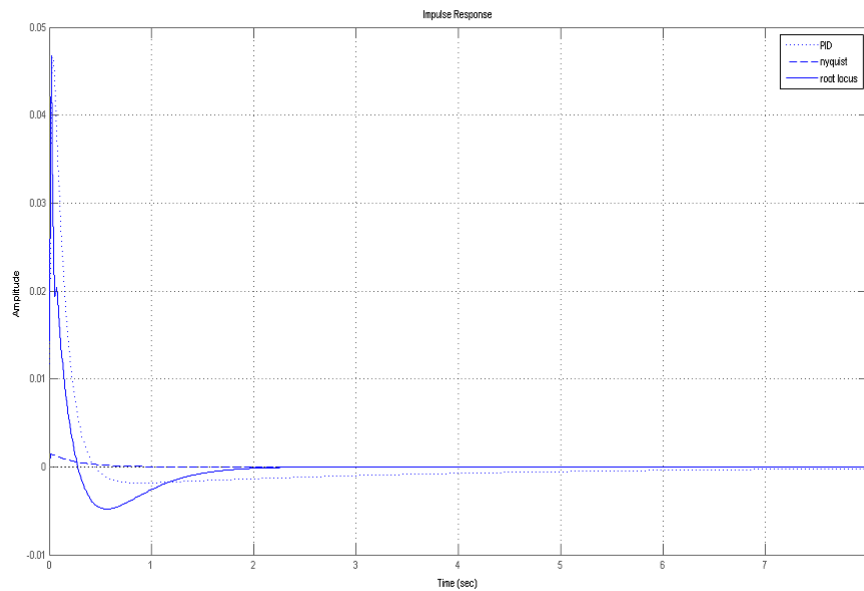


Figure 44: Confronto delle risposte impulsive

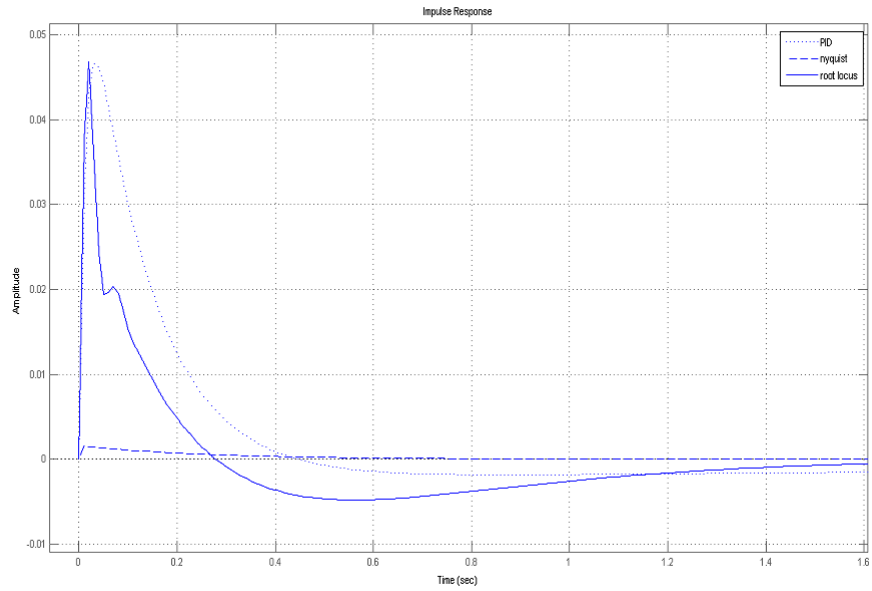


Figure 45: Overshoot delle risposte impulsive

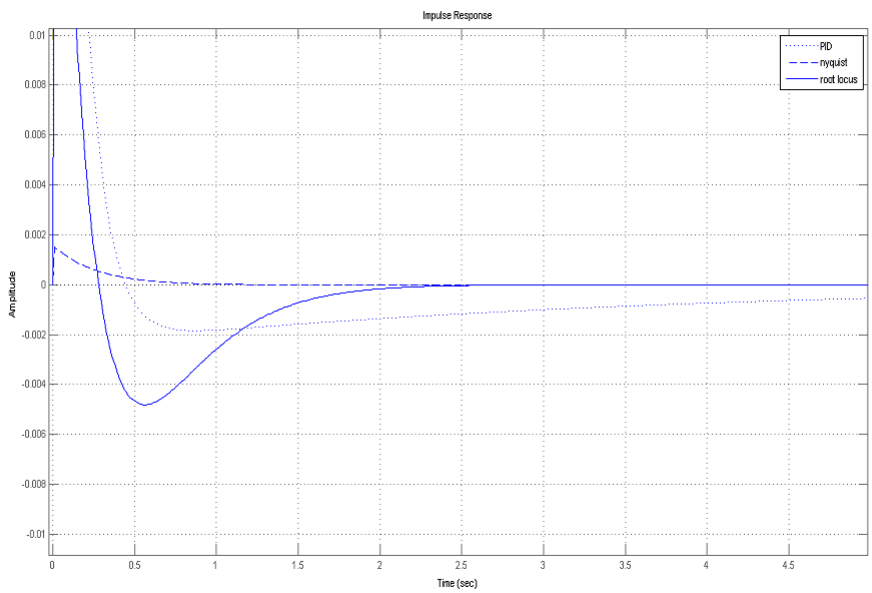


Figure 46: Tempi di assestamento delle risposte impulsive

Andiamo ad analizzare come i sistemi con i diversi controllori rispettano le specifiche: Per quanto riguarda l'overshoot, si vede che il controllore trovato con il metodo della risposta in frequenza è nettamente migliore degli altri, infatti ha come picco circa 0.002 rad mentre gli altri due si equivalgono in circa 0.045rad; allo stesso modo per il tempo di assestamento: il controllore dovuto al metodo dell'analisi in frequenza ha un tempo di assestamento pari a 0s in quanto è in modulo sempre minore di 0.01rad(per la precisione è sempre minore di 0.002 rad) mentre gli altri due hanno circa lo stesso tempo di assestamento di circa 0.25s.c'è da notare che il sistema che ha come controllore quello trovato con il luogo delle radici ha un undershoot che è maggiore del controllore PID.

Per quanto riguarda le risposte al gradino invece abbiamo i seguenti grafici:

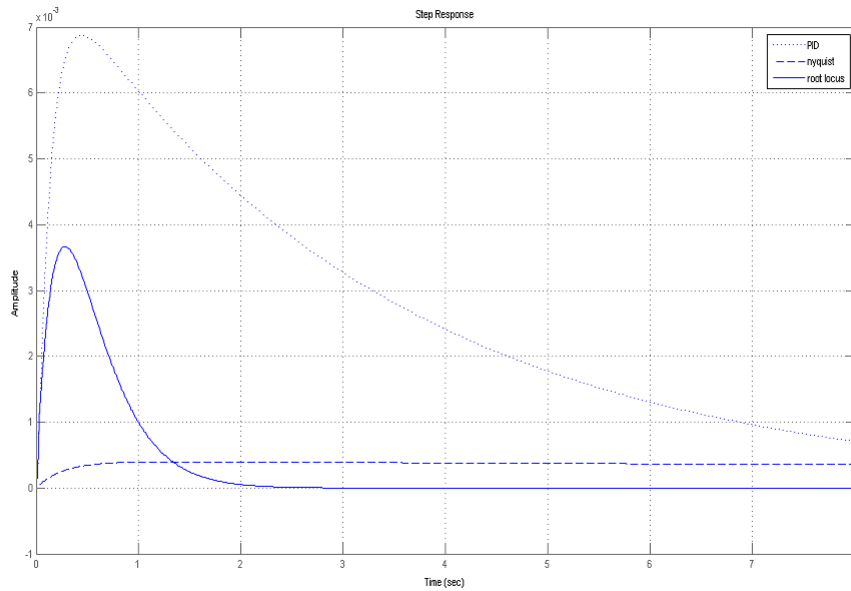


Figure 47: Confronto delle risposte al gradino

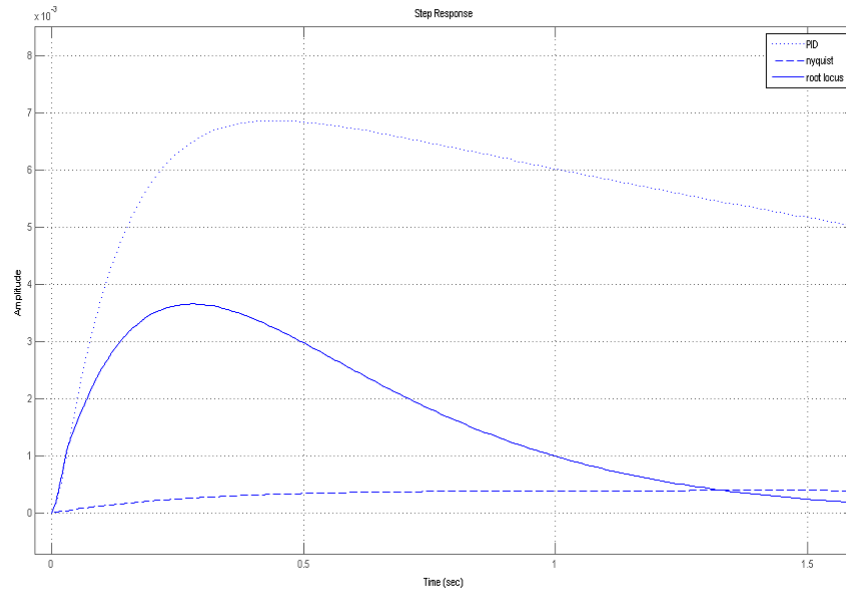


Figure 48: Overshoot delle risposte al gradino

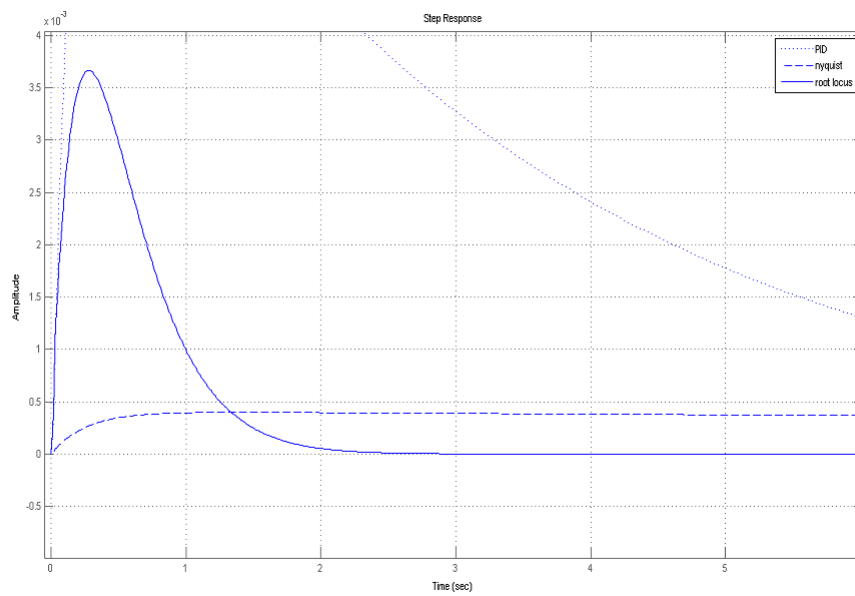


Figure 49: Tempi di assestamento delle risposte al gradino

Procedendo allo stesso modo di prima si vede tutti e tre i compensatori fanno sì che le specifiche siano ampiamente soddisfatte. Si vede che rispetto alla specifica sull'overshoot il controllore sviluppato con il metodo della risposta in frequenza ha un overshoot molto minore rispetto agli altri due. Tutti i controllori rispettano già dall'istante iniziale la specifica sul setting time ma comunque si può vedere che il controllore determinato con il luogo delle radici, nonostante abbia un overshoot maggiore, ha una dinamica molto più veloce degli altri (cioè va a 0 più rapidamente). Il controllore con il metodo della risposta in frequenza invece sembra essere afflitto da un errore costante, ma invece ha solo una dinamica molto lenta, ma in ogni caso rispetta ampiamente le specifiche. Infine qualche considerazione sul segnale di errore: Il segnale di errore è uguale all'opposto del segnale di uscita in quanto  $e(t) = r(t) - \theta(t) = 0 - \theta(t)$  quindi raggiunge come valore massimo i valori massimi del segnale  $\theta(t)$ . Dal punto di vista energetico invece possiamo andare a vedere qual'è l'energia del segnale che dovrà entrare nel controllore:

```
x1=impulse(s1,0:0.01:10);
x2=impulse(s2,0:0.01:10);
x3=impulse(s3,0:0.01:10);
e1=(sum(x1.^2))^2 %E con controllore PID
e2=(sum(x2.^2))^2 %E con controllore trovato con il metodo della risp in frequenza
e3=(sum(x3.^2))^2 %E con controllore trovato col metodo %E con controllore PID
```

che risulta:  $E_{PID} = 4.3583e-004$ ,  $E_{nyq} = 9.3974e-010$ ,  $E_{rloc} = 9.5166e-005$  quindi risulta che il controllore di tipo PID è quello che ha il segnale di errore con più energia mentre quello trovato con il metodo della risposta in frequenza è quello che fa in modo di avere il segnale di errore con minor energia.

## 4.2 Verifica risultati con Simulink

In questa sezione andremo ad utilizzare uno dei più importanti tool di matlab: Simulink; Simulink è un software che permette la modellizzazione, simulazione e l'analisi di sistemi. Il tutto viene gestito da interfaccia grafica a blocchi, in questo modo per descrivere un sistema basta inserire i blocchi di cui è formato, blocchi che sono contenuti in librerie del programma. Lo schema a blocchi del pendolo inverso è il seguente:

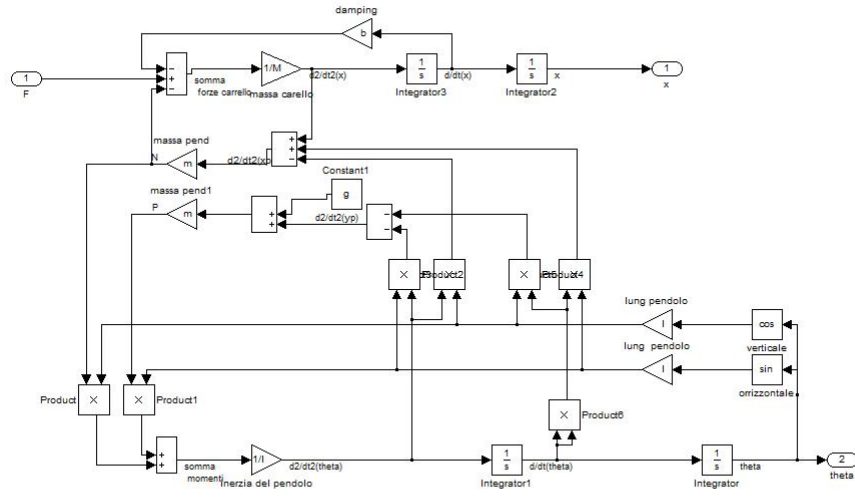


Figure 50: Schema a blocchi sistema pendolo-carrello

Questo modello simulink è stato implementato a partire dalle equazioni del moto descritte nella prima sezione; ha un ingresso (la forza  $F$ ) e la posizione  $x$  del carrello e l'angolo del pendolo con la verticale ( $\theta$ ) come uscite. Una volta creato il modello in simulink non resta altro che racchiuderlo in un blocco di tipo subsystem e collegare tale blocco con un controllore e poi chiudere l'anello e inserire gli ingressi (una costante 0 per il riferimento e un ingresso per la forza), come mostrato nella figura di seguito.

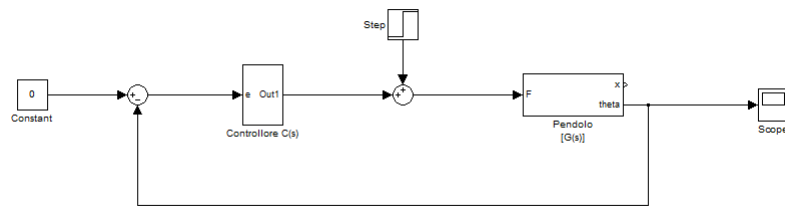


Figure 51: Schema a blocchi sistema pendolo-carrello

A questo punto, per simulare un controllore di quelli creati, ad esempio quello PID, basta sostituire a  $C(s)$  del diagramma precedente, un diagramma di un controllore PID che è formato, da come si può vedere in figura, da un derivatore e un integratore e un guadagno in parallelo; per quanto riguarda l'ingresso, bisogna dire che, per quanto riguarda la generazione

dell'impulso come ingresso è stato necessario impostare il blocco impulse generator con ampiezza 1000 e durata 0.01 in modo che tempo X durata sia circa uguale a 1. Per selezionare l'ingresso basta collegare l'ingresso che si desidera (impulse o step).

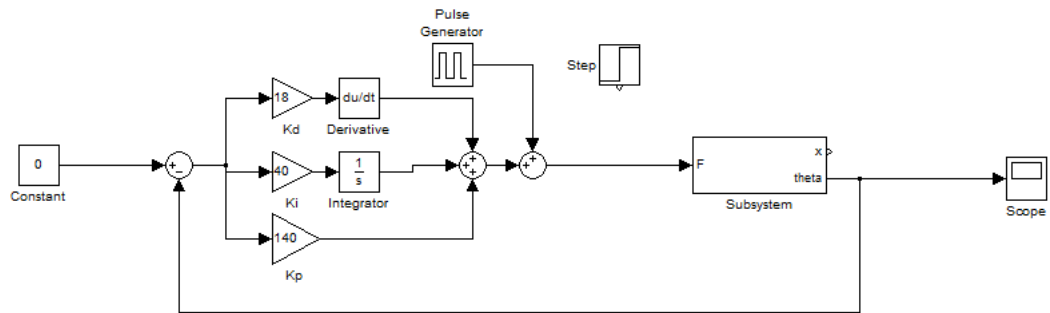


Figure 52: Schema a blocchi sistema con controllore  $C_{PID}(s)$

Di seguito, premendo il pulsante "play" si può simulare la risposta del sistema per ogni tipo di ingresso; di seguito sono rappresentate quelle relative alla risposta impulsiva e al gradino:

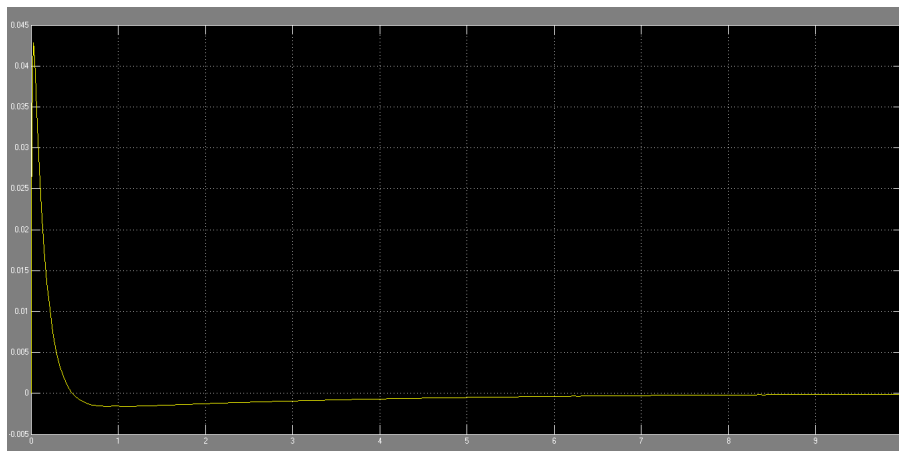


Figure 53: risposta impulsiva con controllore  $C_{PID}(s)$



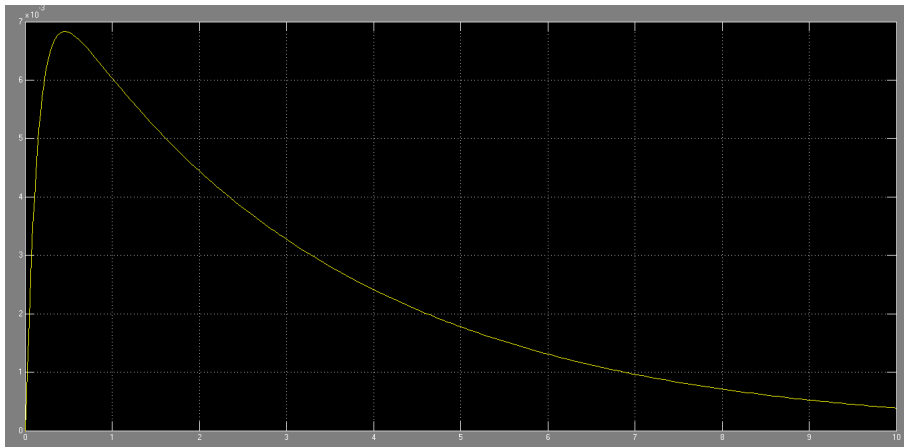


Figure 54: risposta al gradino con controllore  $C_{PID}(s)$

Come si può vedere, gli andamenti del sistema rispecchiano gli andamenti trovati in precedenza. Analogamente a come è stato fatto sopra, si possono simulare anche gli altri tipi di controllore trovati trovando lo stesso risultato.

## List of Figures

1	Schema di principio del sistema . . . . .	4
2	significato dei simboli utilizzati . . . . .	5
3	Risposta impulsiva del sistema ad anello aperto . . . . .	8
4	Risposta al gradino del sistema ad anello aperto . . . . .	8
5	Diagramma di Nyquist del sistema . . . . .	9
6	Schema a blocchi del sistema . . . . .	10
7	Schema a blocchi del sistema . . . . .	10
8	risposte impulsive al variare di $K_p$ . . . . .	11
9	risposte al gradino al variare di $K_p$ . . . . .	12
10	risposta impulsiva con $K_p=140$ . . . . .	13
11	risposta al gradino con $K_p=140$ . . . . .	13
12	risposte impulsive al variare di $K_d$ . . . . .	15
13	risposte al gradino al variare di $K_d$ . . . . .	15
14	risposta impulsiva con $K_p=140$ e $K_d=18$ . . . . .	16
15	risposta al gradino con $K_p=140$ e $K_d=18$ . . . . .	16
16	risposte impulsive al variare di $K_i$ . . . . .	18
17	risposte al gradino al variare di $K_i$ . . . . .	18
18	risposta impulsiva con $C_{PID}(s)$ . . . . .	19
19	risposte al gradino con $C_{PID}(s)$ . . . . .	20
20	Diagrammi di Nyquist e Bode del sistema . . . . .	21
21	Diagrammi di Nyquist e Bode del sistema più compensatore . . . . .	22
22	Diagrammi di Nyquist e Bode del sistema più compensatore . . . . .	23
23	Dettaglio del diagramma di Nyquist . . . . .	24
24	Diagrammi di Nyquist e Bode del sistema più compensatore . . . . .	25
25	Dettaglio del diagramma di Nyquist . . . . .	26
26	Diagrammi di Nyquist e Bode del sistema più compensatore . . . . .	27
27	Risposta impulsiva del sistema in catena chiusa . . . . .	28
28	Risposta al gradino del sistema in catena chiusa . . . . .	28
29	Tipica schermata di SISOTOOL: a sinistra la finestra per la modifica grafica e a destra la risposta impulsiva e al gradino in tempo reale . . . . .	29
30	Risposte in seguito allo spostamento degli zeri del compensatore . . . . .	30
31	Risposte impulsiva del sistema con compensatore $C_{freq}(s)$ . . . . .	31
32	Risposte al gradino del sistema con compensatore $C_{freq}(s)$ . . . . .	31
33	Diagrammi di Bode al variare del compensatore $C_{freq}(s)$ . . . . .	32
34	Luogo delle radici di $G(s)$ . . . . .	33
35	Dettaglio del semipiano destro del luogo delle radici di $G(s)$ . . . . .	33
36	Luogo delle radici di $C_{rloc}(s)G(s)$ . . . . .	35
37	Dettaglio in un intorno dell'origine di $C_{rloc}(s)G(s)$ . . . . .	35
38	Risposta impulsiva del sistema con $K=200$ . . . . .	36
39	Risposta al gradino del sistema con $K=200$ . . . . .	37
40	Risposte impulsive al variare di $K$ . . . . .	38

41	Risposte al gradino al variare di $K$ . . . . .	38
42	Risposta impulsiva con $K=3000$ . . . . .	39
43	Risposta al gradino con $K=3000$ . . . . .	40
44	Confronto delle risposte impulsive . . . . .	41
45	Overshoot delle risposte impulsive . . . . .	42
46	Tempi di assestamento delle risposte impulsive . . . . .	42
47	Confronto delle risposte al gradino . . . . .	43
48	Overshoot delle risposte al gradino . . . . .	44
49	Tempi di assestamento delle risposte al gradino . . . . .	44
50	Schema a blocchi sistema pendolo-carrello . . . . .	45
51	Schema a blocchi sistema pendolo-carrello . . . . .	46
52	Schema a blocchi sistema con controllore $C_{PID}(s)$ . . . . .	46
53	risposta impulsiva con controllore $C_{PID}(s)$ . . . . .	47
54	risposta al gradino con controllore $C_{PID}(s)$ . . . . .	47