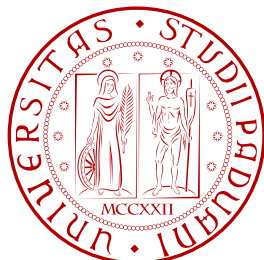


UNIVERSITÀ DEGLI STUDI DI PADOVA
DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE



CORSO DI LAUREA TRIENNALE IN INGEGNERIA ELETTRONICA

Progetto di un sistema di controllo di un pendolo inverso in ambiente MATLAB

Relatore: ***Prof. Alessandro Beghi***

Laureando: ***Alessandro Pin***
Matricola: ***541108***

Anno Accademico: 2011 - 2012

Indice

Indice	2
1 Introduzione	3
2 Modello Matematico	4
2.1 Sistema	4
2.2 Analisi del sistema ed equazioni	5
2.2.1 Funzione di Trasferimento	6
2.2.2 Schema a blocchi in retroazione	7
2.3 Rappresentazione in MATLAB del modello ad anello aperto .	8
2.4 Rappresentazione tramite Simulink del modello ad anello aperto	9
3 Studio del sistema utilizzando un controllore PID	11
3.1 Aggiunta del regolatore PID tramite MATLAB	12
3.2 Controllore PID in Simulink	16
4 Luogo delle radici	18
4.1 Studio del luogo delle radici in ambiente MATLAB	18
4.1.1 Aggiunta di una rete corretrice	20
4.2 Implementazione tramite SIMULINK	24
5 Studio in frequenza	26
5.1 Analisi mediante Bode e Nyquist	26
5.2 Aggiunta di un compensatore	28
5.3 Implementazione tramite SIMULINK	38
6 Conclusioni	40

Capitolo 1

Introduzione

Questo progetto consiste nel simulare in ambiente MATLAB un controllo di un pendolo inverso posizionato sopra un carrello mobile. Si partirà dalla descrizione del sistema per poi ricavarne il modello matematico, verrà analizzata la risposta dinamica a catena aperta. Si cercherà di progettare un controllore inserendolo nel sistema retroazionato in modo da rendere il sistema stabile. Il progetto del controllore verrà realizzato in tre modi differenti: utilizzando un controllore PID, studiando il luogo delle radici ed infine tramite lo studio in frequenza. Si effettueranno inoltre delle simulazioni tramite SIMULINK per vedere l'effettivo funzionamento dei controllori.

Capitolo 2

Modello Matematico

2.1 Sistema

Il sistema in esame è composto da un pendolo inverso posto al di sopra di un carrello mobile quest'ultimo viene mosso da una forza impulsiva F .

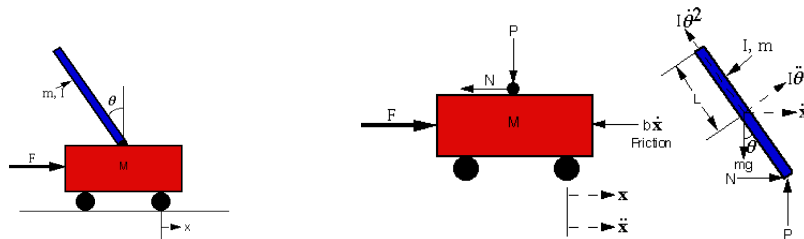


Figura 2.1: Shema di un pendolo inverso

Si suppone che il carrello ed il pendolo abbiano queste caratteristiche:

- $M = 0,5Kg$ Massa carrello
- $m = 0,2Kg$ Massa pendolo
- $b = 0,1N/m/s$ Attrito del carrello
- $l = 0,3m$ Lunghezza dal centro di massa del pendolo
- $I = 0,006Kg \cdot m^2$ Inerzia del pendolo
- F Forza applicata al carrello
- x Posizione del carrello
- θ Angolo del pendolo dalla verticale

Le tecniche di controllo che utilizzeremo si concentreranno solo sul controllo della posizione del pendolo, questo perché lo studio in frequenza, il luogo delle radici ed il controllore PID (*Proporzionale-Integrale-Derivativo*) sono tecniche che possono essere utilizzate solo per sistemi SISO (*SingleInput SingleOutput*). Se si volesse invece effettuare un controllo sia sulla posizione del pendolo che sulla posizione del carrello il sistema sarebbe multiuscita e sarebbe necessario compiere un'analisi con il modello degli spazi di stato, questo però non rientra nello scopo di questa simulazione.

Supponiamo di avviare il sistema in equilibrio (pendolo nella posizione verticale) e successivamente applichiamo una forza impulsiva al carrello di 1N. Le tecniche di controllo dovranno far sì che il pendolo ritorni nella posizione verticale entro 5 secondi e che l'angolo che esso forma con la verticale non superi mai i 0,05 radianti.

I requisiti di progettazione dunque saranno:

- Tempo di assestamento inferiore a 5 secondi
- Angolo del pendolo mai superiore a 0,05 rad dalla verticale

2.2 Analisi del sistema ed equazioni

Dividendo il sistema in due sottosistemi riusciamo ad ottenere le seguenti equazioni del moto:

La somma delle forze lungo l'asse orizzontale del carrello darà:

$$M\ddot{x} + b\dot{x} + N = F$$

Per il pendolo la somma delle forze lungo l'asse orizzontale dà:

$$N = m\ddot{x} + ml\ddot{\theta}\cos\theta - ml\dot{\theta}^2\sin\theta$$

Da queste due equazioni si ricava la prima equazione del moto di questo sistema:

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

La seconda equazione del moto si ricava sommando le forze perpendicolari al pendolo, mentre le forze perpendicolari al carrello non danno alcun contributo rilevante, consentendoci così di ottenere la seguente equazione:

$$P\sin\theta + N\cos\theta - mg\sin\theta = ml\ddot{\theta} + m\ddot{x}\cos\theta$$

Sommando i momenti intorno al baricentro del pendolo otteniamo:

$$-Pl\sin\theta - Nl\cos\theta = I\ddot{\theta}$$

Combinando le due equazioni otteniamo:

$$(I + ml^2)\ddot{\theta} + mgl \sin \theta = -ml\ddot{x} \cos \theta$$

Dato che MATLAB consente di lavorare esclusivamente con funzioni lineari le due equazioni devono essere linearizzate su $\theta = \pi$. Quindi assumiamo $\theta = \pi + \phi$ dove ϕ rappresenta un piccolo angolo rispetto alla direzione verticale quindi si avrà: $\cos \theta = -1$ $\sin \theta = -\phi$ $\dot{\theta}^2 = 0$

Risulterebbero le seguenti equazioni:

$$(I + ml^2)\ddot{\phi} - mgl\phi = ml\ddot{x}$$

$$(M + m)\ddot{x} + b\dot{x} - ml\ddot{\phi} = U$$

2.2.1 Funzione di Trasferimento

Le funzioni di trasferimento si ricavano partendo dalla trasformata di Laplace delle due equazioni del moto del sistema e rapportando l'uscita del sistema con il suo ingresso:

$$(I + ml^2)\Phi(s)s^2 - mgl\Phi(s) = mlX(s)s^2$$

$$(M + m)X(s)s^2 + bX(s)s + ml\Phi(s)s^2 = U(s)$$

Possiamo ora trovare il valore di $X(s)$ in funzione di $\Phi(s)$ dalla prima equazione, dato che Φ è l'uscita da controllare, risulta:

$$X(s) = \left[\frac{I + ml^2}{ml} + \frac{g}{s^2} \right] \Phi(s)$$

Ora sostituendo il valore di $X(s)$ nella seconda equazione possiamo ricavare la funzione di trasferimento:

$$G(s) = \frac{\Phi(s)}{U(s)} = \frac{\frac{ml}{q}s}{s^3 + \frac{b(I+ml^2)}{q}s^2 + \frac{(M+m)mgl}{q}s + \frac{bmgl}{q}}$$

dove:

$$q = [(M + m)(I + ml^2) - (ml)^2]$$

2.2.2 Schema a blocchi in retroazione

Controllare la posizione del pendolo, in modo da riportarlo sulla verticale dopo un'impulso, implica che il segnale di riferimento deve essere zero. La forza impulsiva applicata al carrello può essere considerata come disturbo per cui lo schema a blocchi in retroazione, con controllore sarà simile al seguente

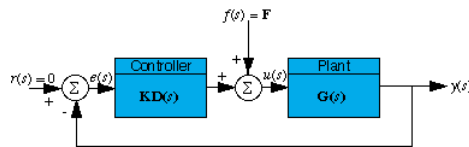


Figura 2.2: Schema a blocchi ad anello chiuso con controllore

che può essere riorganizzato nel seguente modo:

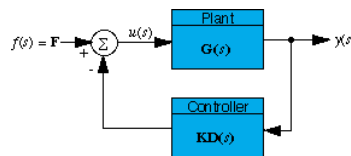


Figura 2.3: Schema equivalente

Da questo schema a blocchi risulta che il sistema sarà:

$$Y(s) = \frac{G(s)}{1 + G(s) \cdot KD(s)} \cdot F(s)$$

Dove $G(s)$ rappresenta la funzione di trasferimento a catena aperta K il guadagno del controllore $D(s)$ la funzione di trasferimento del controllore.

2.3 Rappresentazione in MATLAB del modello ad anello aperto

Per rappresentare la FDT(funzione di trasferimento) e la risposta impulsiva del sistema usiamo il seguente codice MATLAB:

```
1 M = 0.5; % massa Carrello [Kg]
2 m = 0.2; % massa pendolo [Kg]
3 b = 0.1; % attrito del carrello [N/m/s]
4 l = 0.3; % lunghezza dal' estremità al centro di massa del pendolo [m]
5 i = 0.006; %inerzia del pendolo [Kg*m^2]
6 g = 9.8; % accelerazione gravitazionale [m/s^2]
7
8 q = (M+m)*(i+m*l^2)-(m*l)^2; % valore di q
9
10 num = [m*l/q 0]; %numeratore della FDT
11 den = [1 b*(i+m*l^2)/q -(M+m)*m*g*l/q -b*m*g*l/q]; %denominatore della FDT
12
13 pend = tf (num, den) % ft da come output la FDT
14
15 t = [0:0.001:1];
16 impulse (pend, t) % stampa il grafico della risposta impulsiva
17 axis ([0 1 0 60])
```

L'output della funzione tf (num, den) sarà:

```
1 Transfer function:
2           4.545 s
3 -----
4 s^3 + 0.1818 s^2 - 31.18 s - 4.455
```

La funzione impulse (FDT, t) stampa il seguente grafico:

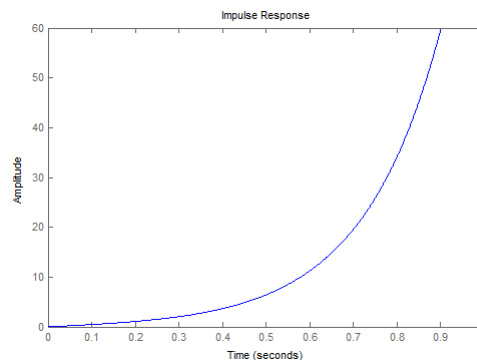


Figura 2.4: Risposta impulsiva

Dal grafico si nota che la risposta non è stabile ad anello aperto, lo stesso

risultato si può ottenere anche utilizzando il comando MATLAB $p=pole(pend)$ che restituisce il valore dei poli:

```

1 >> p = pole(pend)
2
3 p =
4
5     5.5651
6    -5.6041
7    -0.1428

```

e si vede subito che vi è un polo a parte reale positiva che rende il sistema a catena aperta instabile.

2.4 Rappresentazione tramite Simulink del modello ad anello aperto

SIMULINK può lavorare direttamente con equazioni non lineari quindi dalle equazioni ricavate precedentemente non linearizzate si otterrà il seguente schema in SIMULINK che rappresenta il modello matematico:

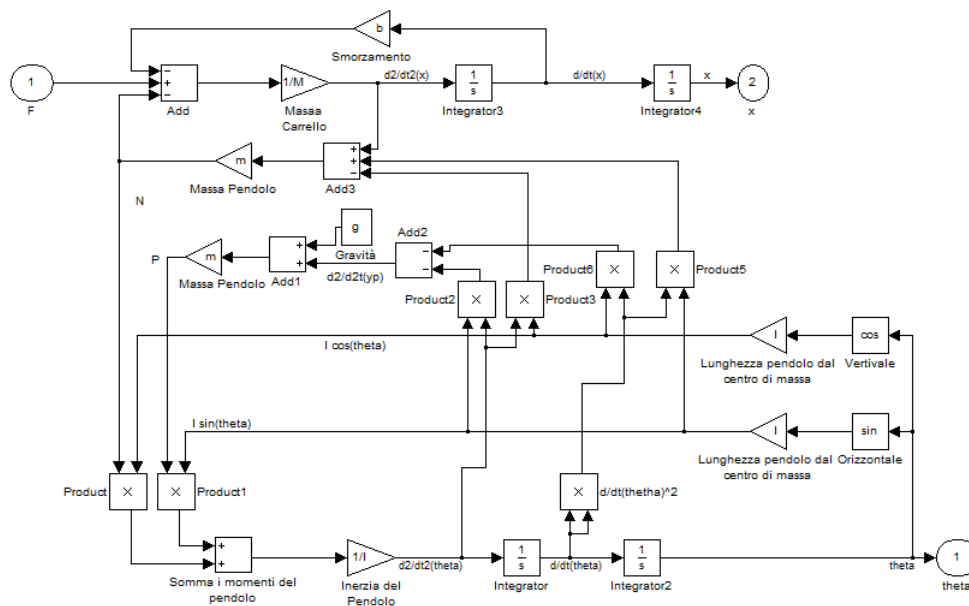


Figura 2.5: Modello matematico in simulink

inserendo il modello ricavato in precedenza in un blocco subsystem, aggiungendo all'ingresso un blocco generatore di impulsi e alle uscite 2 finestre scope come da schema:

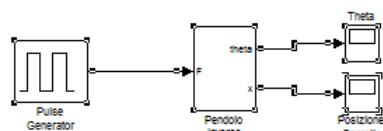


Figura 2.6: Sistema ad anello aperto

possiamo far partire la simulazione che darà le seguenti risposte ad anello aperto per il carrello e per l'angolo del pendolo

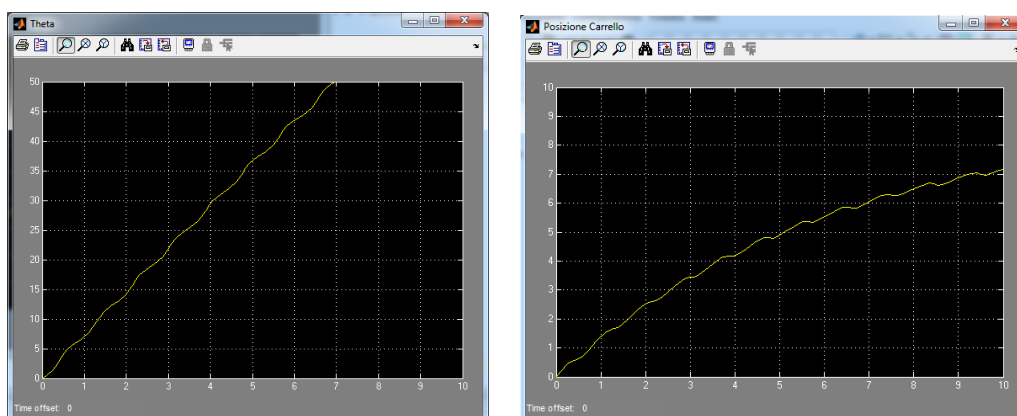


Figura 2.7: risposta impulsiva dell'angolo e del carrello

Capitolo 3

Controllore PID

Il regolatore PID (Proporzionale Integrale Derivativo) è un controllore che combina in modo pesato tre differenti azioni sull'errore all'uscita del sistema: proporzionale, integrale, derivativa; in generale un regolatore di questo tipo assume nel dominio del tempo la seguente espressione:

$$u(t) = Kp \cdot e(t) + Ki \cdot \int_0^t e(\tau) d\tau + Kd \cdot \frac{\partial e(t)}{\partial t}$$

dove $e(t)$ rappresenta l'errore. questa si tramuta nella seguente funzione di trasferimento:

$$K(s) = Kp + sKd + \frac{Ki}{s} = \frac{Kds^2 + Kps + Ki}{s}$$

- L'azione proporzionale Kp riduce l'errore a regime, diminuisce il tempo di salita aumenta però gli overshoot (sovraelongazioni)
- L'azione integrale Ki porta a zero l'errore a regime, ma aumenta gli overshoot ed il comportamento oscillatorio del sistema aumentando così il tempo di assestamento
- L'azione derivativa infine riduce gli overshoot e le oscillazioni del sistema diminuendo così il tempo di assestamento

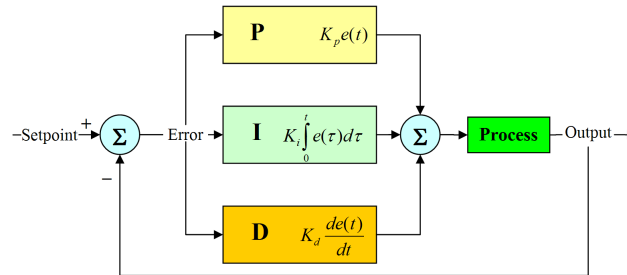


Figura 3.1: schema a blocchi di un generico controllore PID

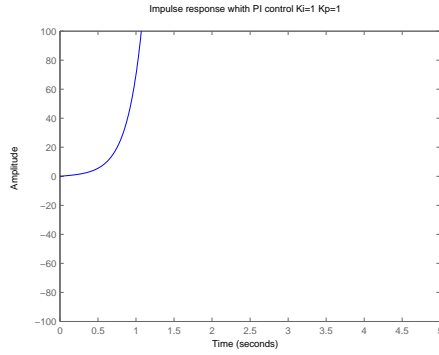
3.1 Aggiunta del regolatore PID tramite MATLAB

Dalla descrizione data in precedenza utilizzando un semplice controllore PI non si riescono ad ottenere delle risposte adeguate poichè non è possibile eliminare la componente oscillatoria solo con l'azione proporzionale e integrale. Dal seguente codice si può vedere la risposta utilizzando un regolatore PI

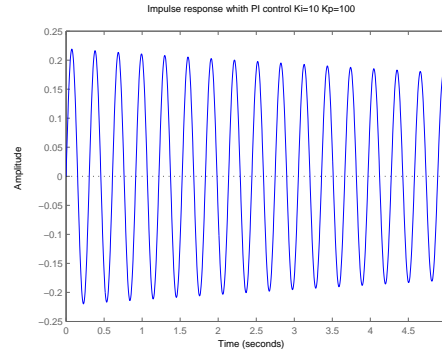
```

1 M = 0.5; % massa Carrello [Kg]
2 m = 0.2; % massa pendolo [Kg]
3 b = 0.1; % attrito del carrello [N/m/s]
4 l = 0.3; % lunghezza dal' estremità al centro di massa del pendolo [m]
5 i = 0.006; %inerzia del pendolo [Kg*m^2]
6 g = 9.8; % accelerazione gravitazionale [m/s^2]
7
8 q = (M+m)*(i+m*l^2)-(m*l)^2; % valore di q
9
10 num = [m*l/q 0]; %numeratore della FDT
11 den = [1 b*(i+m*l^2)/q -(M+m)*m*g*l/q -b*m*g*l/q]; %denominatore della FDT
12
13 pend = tf (num, den); % ft da come output la FDT
14
15 Ki = 10;
16 Kp = 100;
17
18 contr = tf ([Kp Ki], [1 0]) %crea la FDT del controllore
19 sys_cl = feedback (pend, contr); % mette il controllore in retroazione
    con il sistema
20 t = 0:0.005:5;
21 impulse(sys_cl, t) %stampa il grafico della risposta impulsiva del
    sistema con il controllore in retroazione
22 axis([0 5 -0.25 0.25])
23 title('Impulse response with PI control Ki=10 Kp=100')
```

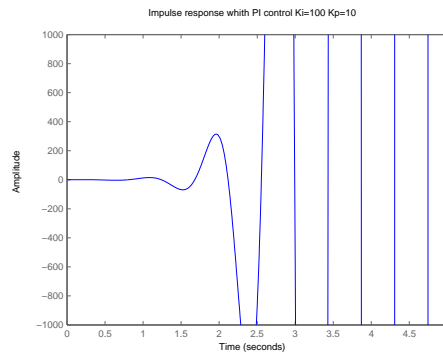
cambiando opportunamente i valori di K_i e K_p si ottengono le seguenti risposte impulsive:



(a) $K_i=1$ $K_p=1$



(b) $K_i=50$ $K_p=100$



(c) $K_i=100$ $K_p=10$

Figura 3.2: Varie risposte impulsive attenuate con un controllore PI

Come si evince dai grafici precedentemente esposti la risposta di un controllore PI non è adeguata.

Mediante l'utilizzo di un controllore PID utilizzando il seguente codice e variando i valori di K_i , K_p e K_d in modo opportuno si riuscirà ad ottenere una risposta stabile che soddisfi le specifiche di progetto:

```

1 M = 0.5; % massa Carrello [Kg]
2 m = 0.2; % massa pendolo [Kg]
3 b = 0.1; % attrito del carrello [N/m/s]
4 l = 0.3; % lunghezza dal' estremità al centro di massa del pendolo [m]
5 i = 0.006; %inerzia del pendolo [Kg*m^2]
6 g = 9.8; % accelerazione gravitazionale [m/s^2]
7

```

```

8 q = (M+m)*(i+m*l^2)-(m*l)^2; % valore di q
9
10 num = [m*l/q 0]; %numeratore della FDT
11 den = [1 b*(i+m*l^2)/q -(M+m)*m*g*l/q -b*m*g*l/q]; %denominatore della FDT
12
13 pend = tf (num, den); % ft da come output la FDT
14
15 Ki = 1;
16 Kp = 100;
17 Kd = 20;
18 contr = tf ([Kd Kp Ki], [1 0]);
19 sys_cl = feedback (pend, contr);
20 t = 0:0.005:5;
21 impulse(sys_cl, t)
22 axis([0 5 -2 2])

```

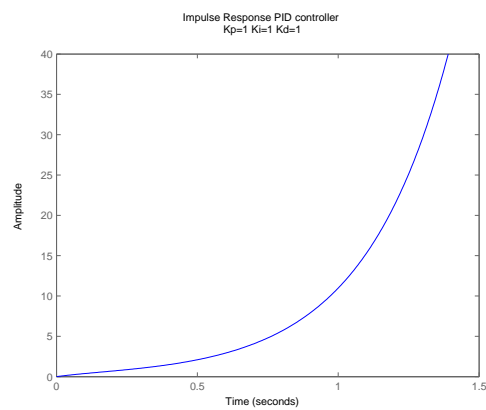
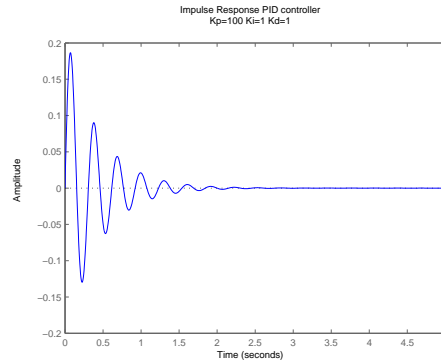


Figura 3.3: Risposta impulsiva con PID $K_p=1$ $K_i=1$ $K_d=1$

Questa risposta non è stabile quindi potremmo cambiare il valore di K_p in $K_p=100$ e studiarne l'effetto, il risultato sarà il seguente:



Risposta impulsiva con PID $K_p=100$ $K_i=1$ $K_d=1$

Poiché l'errore a regime si riduce a zero dopo circa tre secondi non vi è alcun bisogno di aumentare il guadagno integrale, il problema ora è dato dall'overshoot iniziale che supera i 0,05 radianti, per migliorare la risposta dobbiamo agire sulla parte derivatrice aumentando K_d , con $K_d=20$ otteniamo la seguente risposta:

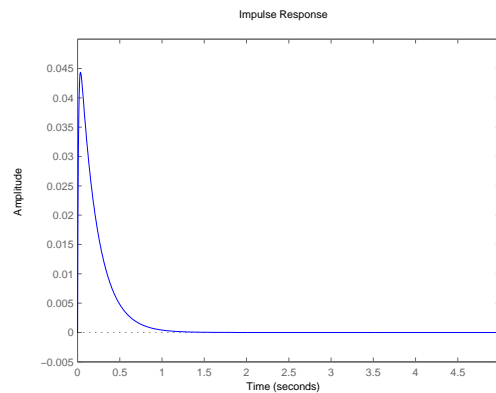


Figura 3.5: Risposta impulsiva con PID $K_p=100$ $K_i=1$ $K_d=20$

Da questo grafico si vede dunque che tramite un controllore PID otteniamo una risposta impulsiva stabile che rientra nelle specifiche di progetto. Si nota però che il modello utilizzato per il controllore PID non è realizzabile perché la FDT del PID è impropria. Per renderla propria basta aggiungere un polo negativo ad alta frequenza e come si vede la risposta impulsiva non cambia:

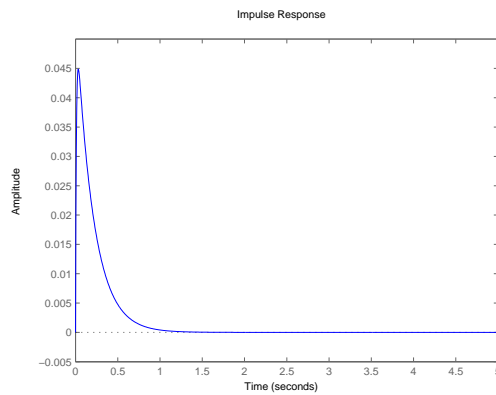


Figura 3.6: Risposta impulsiva del PID con polo in 1000

3.2 Controllore PID in Simulink

Aggiungendo un PID in retroazione al modello SIMULINK, precedentemente ottenuto, otterremo il seguente schema:

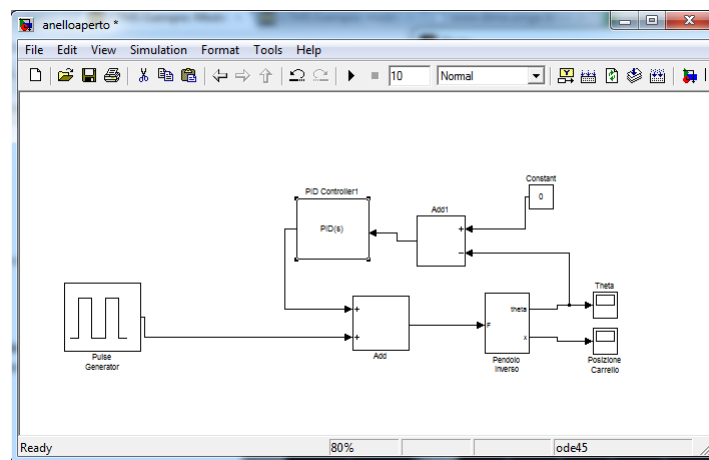


Figura 3.7: Schema simulink con un PID in retroazione

impostando i valori del PID con $K_p=100$ $K_i=1$ e $K_d=20$ e il valore del polo in alta frequenza verrà settato a 1000 avviando la simulazione otterremo:

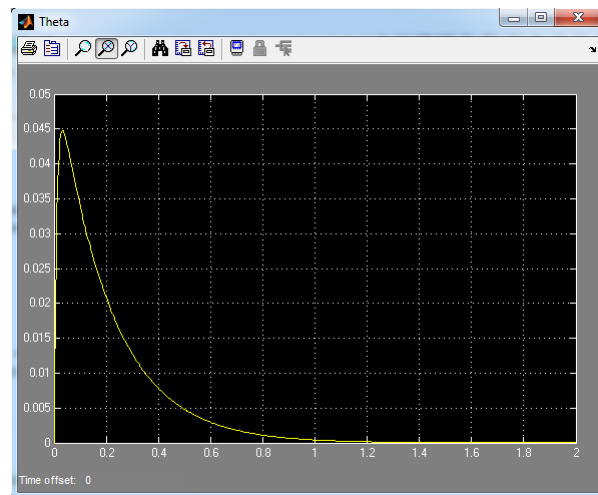


Figura 3.8: Controllore PID $K_p=100$ $K_i=1$ $K_d=20$

Si osserva dalla simulazione che le specifiche di progetto sono soddisfatte.

Capitolo 4

Luogo delle radici

Il luogo delle radici è uno strumento grafico per l'analisi e la sintesi di sistemi di controllo a retroazione. Data una funzione di trasferimento $G(s)$ esso mostra la posizione, nel piano complesso, di tutti i possibili poli che si ottengono ad anello chiuso al variare di guadagno proporzionale K . Si deduce che la funzione di trasferimento nel nostro caso con K nella catena di retroazione sarà:

$$W(s) = \frac{G(s)}{1 + KG(s)}$$

Essendo $G(s) = \frac{num(s)}{den(s)}$ i poli della funzione di trasferimento $W(s)$ varieranno al variare di K secondo la seguente espressione:

$$K \cdot num(s) + den(s) = 0$$

4.1 Studio del luogo delle radici in ambiente MATLAB

Analizziamo innanzitutto il luogo senza alcun compensatore, tenendo conto delle specifiche avremmo un tempo di assestamento $t_s \leq 5s$ quindi σ , che rappresenta la parte reale negativa del polo, dovrà essere:

$$\sigma \geq \frac{4,6}{t_s}$$

Per poter osservare il valore di σ nel grafico del luogo delle radici utilizzeremo la seguente funzione MATLAB:

```

1 function [] = sigrid(sig)
2
3 %SIGRID Generate s-plane grid lines for a root locus or pole-zero map.
4 %
5 % SIGRID generates a grid over an existing continuous s-plane root
6 % locus or pole-zero map. Lines of constant sigma are drawn in.
7 % To be used with SGRID if sigma, zeta, and Wn requirements are required
8 % simultaneously. Can also be used by itself.
9 %
10 % See also: RLOCUS, ZGRID, SGRID, and PZMAP.
11
12 error(nargchk(1,1,nargin));
13
14 hold on
15
16 %Plot sigma line
17     limits = axis;
18     mx=limits(1,4);
19     mn=limits(1,3);
20     stz=abs(mx)+abs(mn);
21     st=stz/50;
22     im=mn:st:mx;
23     lim=length(im);
24     for i=1:lim
25         re(i)=-sig;
26     end
27     re(:);
28     plot(re,im,'k--')
29
30 hold off
31
32 return

```

Utilizzando il codice per creare la FDT del sistema descritto nel capitolo due e successivamente utilizzando i seguenti comandi:

```

1 rlocus(pend)
2 sigrid(0.92)
3 axis([-6 6 -6 6])

```

verrà visualizzato il seguente grafico:

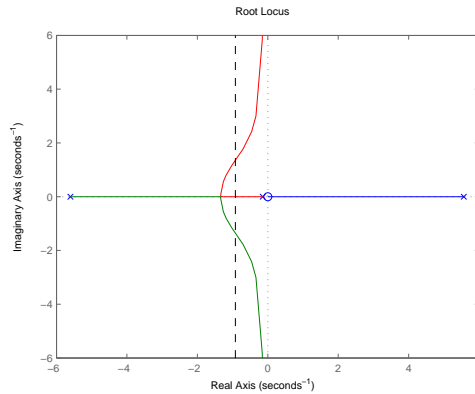


Figura 4.1: Luogo delle radici senza alcun compensatore

Il grafico sopra riportato rappresenta il luogo delle radici senza alcuna compensazione; la linea nera tratteggiata rappresenta il valore di σ . Come si può evincere da questo grafico il sistema è instabile per qualsiasi guadagno K perchè vi sono sempre poli nel semipiano reale positivo.

4.1.1 Aggiunta di una rete correttrice

Per risolvere questo problema è necessario, inizialmente, eliminare lo zero nell'origine, perchè tracciando il luogo delle radici per un sistema con uno zero nell'origine e un polo reale positivo non si otterrà stabilità, questo è dovuto al fatto che il polo positivo per $K \rightarrow \infty$ tenderà o allo zero nell'origine o a ∞ . Il controllore dovrà quindi contenere sicuramente un termine $\frac{1}{s}$. utilizzando il seguente codice:

```
1 contr=tf(1,[1 0]);  
2 rlocus(contr*pend)  
3 sigrid(0.92)  
4 axis([-10 10 -10 10])
```

si otterrà il seguente luogo:

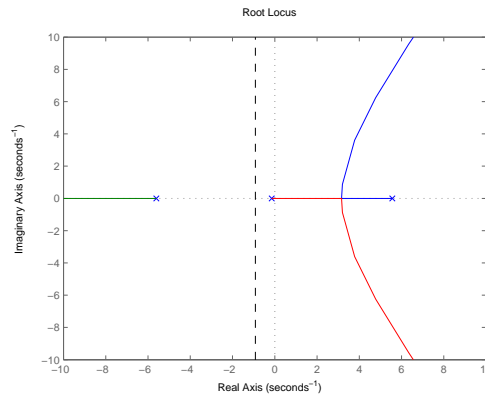


Figura 4.2: Luogo delle radici con integratore

Dal grafico si può dedurre che uno dei modi per rendere il sistema stabile è quello di aggiungere un polo distante, a sinistra degli altri poli e due zeri con valori relativamente bassi ma compresi però tra i due poli negativi adiacenti l'origine, in modo da riuscire, per K elevati, a portare il polo a parte reale positiva nel semipiano sinistro ottenendo quindi stabilità.

Per fare ciò utilizziamo il seguente codice:

```

1 M = 0.5; % massa Carrello [Kg]
2 m = 0.2; % massa pendolo [Kg]
3 b = 0.1; % attrito del carrello [N/m/s]
4 l = 0.3; % lunghezza dal' estremità al centro di massa del pendolo [m]
5 i = 0.006; %inerzia del pendolo [Kg*m^2]
6 g = 9.8; % accelerazione gravitazionale [m/s^2]
7
8 q = (M+m)*(i+m*l^2)-(m*l)^2; % valore di q
9
10 num = [m*l/q 0]; %numeratore della FDT
11 den = [1 b*(i+m*l^2)/q -(M+m)*m*g*l/q -b*m*g*l/q]; %denominatore della FDT
12
13 pend = tf (num, den); % ft da come output la FDT
14 %creazione del controllore
15 z1 = 3; %posizione del primo zero
16 p1 = 0; %polo che cancella lo zero nell'origine
17 z2 = 4; %posizione del secondo zero
18 p2 = 60; %polo distante dai 3 poli del sistema
19 numlag = [1 z1]; %(s+z1)
20 denlag = [1 p1]; %(s+p1)
21 numlead = [1 z2]; %(s+z2)
22 denlead = [1 p2]; %(s+p2)
23 numc = conv(numlead, numlag); %numeratore del controllore
24 denc = conv(denlead, denlag); %denominatore del controllore

```

```

25 contr=tf(numc,denc); %funzione di trasferimento del controllore
26
27
28 %creazione del grafico del luogo con il controllore
29 rlocus(contr*pend)
30 sigrid(0.92)
31 axis([-60 60 -50 50])
32 pause
33 figure
34 rlocus(contr*pend)
35 sigrid(0.92)
36 axis([-5 0 -10 10])
37 %relocfind() permette di scegliere un punto del grafico e restituisce
38 %il valore di K e i corrispondenti poli per quel K
39 [k,poles]=rlocfind(contr*pend)
40 %retroazionato il sistema con il controllore ed il K scelto guardiamo se
    la
41 %risposta impulsiva è adeguata
42 sys_cl=feedback(pend,contr*k);
43
44 %risposta impulsiva con il K scelto
45 figure
46 impulse(sys_cl)
47 axis([0 2 -0.1 0.1])

```

che visualizzerà inizialmente il seguente grafico:

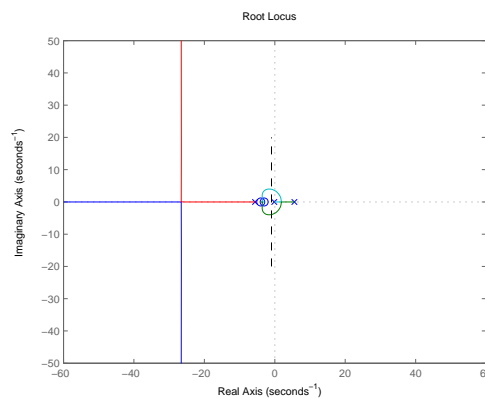


Figura 4.3: Luogo delle radici con compensatore

dopo questo grafico ne verrà visualizzato uno identico ingrandito in modo da poter vedere meglio come si comporta il luogo nei pressi dell'origine e successivamente verrà data la possibilità di scegliere un polo che dovrà essere scelto con parte reale a sinistra di σ per poter avere una risposta impulsiva

con $t_s \leq 5s$ scegliendo però un polo in prossimità di σ non sarà soddisfatta la condizione riguardante l'overshoot.

Una possibile soluzione è data scegliendo dei poli come da seguente figura:

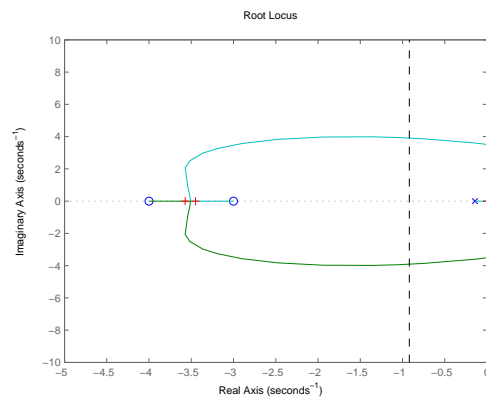


Figura 4.4: Comportamento del luogo nei pressi dell'origine

otterremo il seguente valore di K ed i seguenti poli:

```

1 k =
2
3 3.2290e+003
4
5
6 poles =
7
8 1.0e+002 *
9
10 0
11 -0.2658 + 1.1647i
12 -0.2658 - 1.1647i
13 -0.0357
14 -0.0345

```


e la seguente risposta impulsiva:

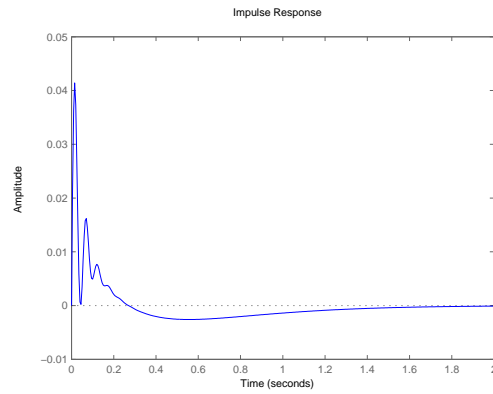


Figura 4.5: Risposta impulsiva con compensatore $K \frac{(s+3)(s+4)}{s(s+60)}$

la quale soddisfa appieno le specifiche di controllo.

4.2 Implementazione tramite SIMULINK

Procedendo con le stesse modalità utilizzate per il controllore PID, al posto del blocco PID in retroazione mettiamo un blocco con guadagno pari al K che abbiamo appena trovato seguito da un blocco funzione di trasferimento con all'interno $K \frac{(s+3)(s+4)}{s(s+60)}$, lo schema a blocchi sarà il seguente:

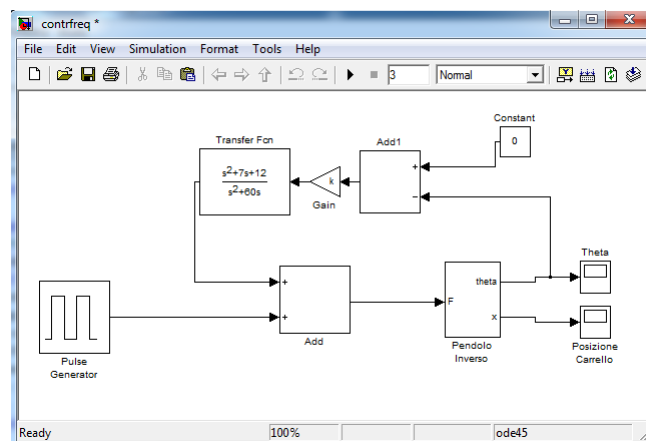


Figura 4.6: Schema a blocchi con compensatore $K \frac{(s+3)(s+4)}{s(s+60)}$

e questa la risposta della simulazione:

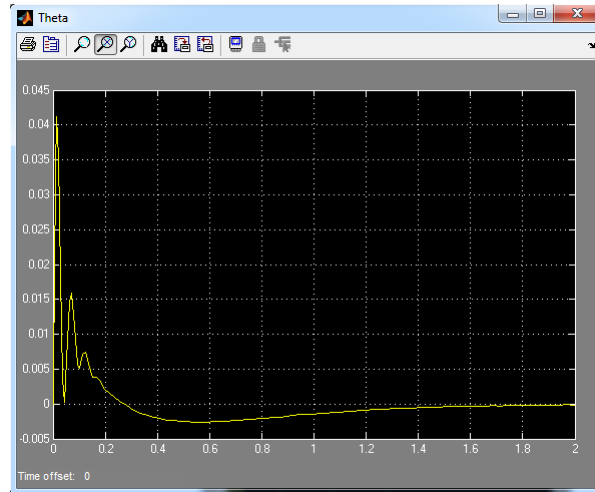


Figura 4.7: Risposta impulsiva della simulazione con controllore: $K \frac{(s+3)(s+4)}{s(s+60)}$

Che soddisfa appieno le specifiche di progetto.

Capitolo 5

Studio in frequenza

Lo studio in frequenza di un generico sistema dinamico lineare, con funzione di trasferimento $G(s)$, viene valutato imponendovi un ingresso $u(t)$ sinusoidale del tipo:

$$u(t) = A \cos(\omega t + \varphi)$$

osservando come all'uscita varia l'ampiezza A , in modulo, e lo sfasamento φ , variando la frequenza ω ; queste variazioni possono essere rappresentate mediante grafici cartesiani, diagrammi di Bode; o attraverso grafici polari, diagrammi di Nyquist. Quest'ultimo è particolarmente utile nell'analisi della stabilità di un sistema a retroazione unitaria grazie al criterio di Nyquist secondo il quale:

$$P_{W(s)}^+ = P_{G(s)}^+ - N_g$$

dove $P_{W(s)}^+$ sono i poli a parte reale positiva di $W(s)$ ($W(s)$ rappresenta il sistema in retroazione unitaria) $P_{G(s)}^+$ sono i poli a parte reale positiva del sistema ad anello aperto e N_g sono il numero di giri che la curva del diagramma di Nyquist effettua attorno al punto -1 in senso antiorario.

5.1 Analisi mediante bode e Nyquist

Come si è già visto nei precedenti capitoli il sistema è instabile a catena aperta a causa del polo reale positivo. Questo significa secondo il criterio di Nyquist che per essere stabile in retroazione unitaria il diagramma di Nyquist dovrà effettuare un giro in senso antiorario intorno al punto -1.

Utilizzando il comando `bode(pend)` otteniamo il diagramma di Bode del sistema:

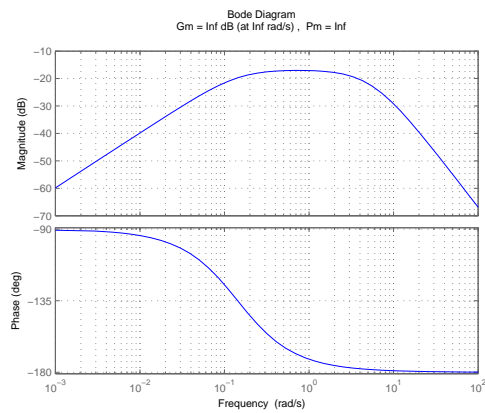


Figura 5.1: Diagramma di Bode del sistema

Utilizzando invece il comando `nyquist(pend)` otteniamo il diagramma di Nyquist:

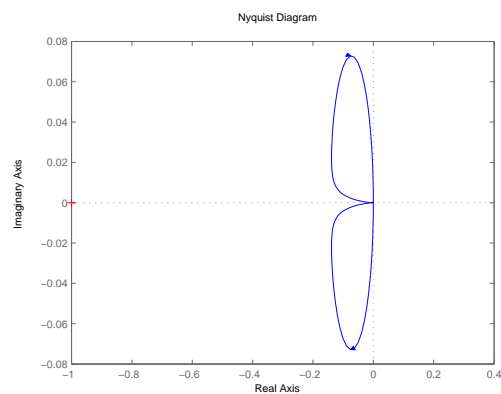


Figura 5.2: Diagramma di Nyquist del sistema

dal quale notiamo come il sistema in retroazione unitaria risulti instabile, in quanto il grafico non compie un giro in senso anti-oraio attorno a -1. A conferma di quanto appena detto osserviamo la risposta impulsiva a catena chiusa utilizzando il comando `impulse(feedback(pend,1))` otterremo:

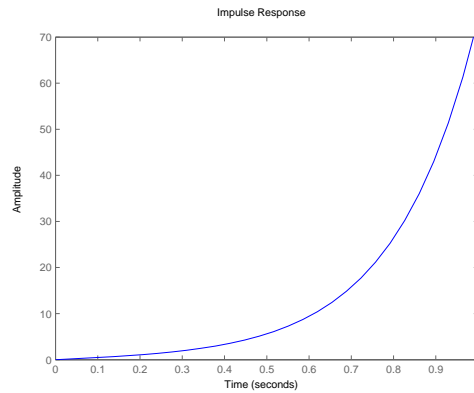


Figura 5.3: Risposta impulsiva a retroazione unitaria

Anche da questo grafico possiamo notare come la retroazione non riesca a stabilizzare il sistema.

5.2 Aggiunta di un compensatore

Aggiungendo un compensatore nella catena di retroazione analizzeremo la funzione di trasferimento $H(s) = G(s) \cdot G_c(s)$ con $G(s)$ la FDT del sistema e $G_c(s)$ la FDT del controllore, dovremmo quindi fare in modo che il diagramma di Nyquist di $H(s)$ faccia un giro in senso anti-orario attorno a -1. Per aiutarci nella progettazione del controllore utilizzeremo il seguente codice che riceve come input il numeratore, il denominatore ed il guadagno della funzione di trasferimento del controllore.

```

1 function[] = frequenza()
2
3 %definisco la funzione di trasferimento
4 M = 0.5; % massa Carrello [Kg]
5 m = 0.2; % massa pendolo [Kg]
6 b = 0.1; % attrito del carrello [N/m/s]
7 l = 0.3; % lunghezza dal' estremità al centro di massa del pendolo [m]
8 i = 0.006; %inerzia del pendolo [Kg*m^2]
9 g = 9.8; % accelerazione gravitazionale [m/s^2]
10
11 q = (M+m)*(i+m*l^2)-(m*l)^2; % valore di q
12
13 num = [m*l/q 0]; %numeratore della FDT
14 den = [1 b*(i+m*l^2)/q -(M+m)*m*g*l/q -b*m*g*l/q]; %denominatore della FDT
15
16 pend = tf (num, den); % ft da come output la FDT
17 figure(1)

```

```

18
19 %riceve da tastiera i valori del controllore
20 nc = input('inserire il numeratore del controllore: ');
21 dc = input('inserire il denominatore del controllore: ');
22 k = input('inserire il guadagno del controllore: ');
23
24 %visualizza il diagramma di bode del sistema compensato
25 contr = k*tf(nc,dc)
26 siscon = pend*contr;
27 bode(siscon);
28
29 %visualizza in un'altra finestra il diagramma di nyquist
30 figure(2)
31 nyquist(siscon)
32
33 %visualizza in una terza finestra la risposta impulsiva in retroazione
34 figure(3)
35 sys_cl = feedback(pend,contr);
36 impulse(sys_cl)
37 axis([0 5 0 100])

```

La nostra prima azione sarà quella di aggiungere un'azione integrativa per eliminare lo zero nell'origine dando come input:

```

1  inserire il numeratore del controllore: 1
2  inserire il denominatore del controllore: [1 0]
3  inserire il guadagno del controllore: 1

```

otterremo i seguenti tre grafici:

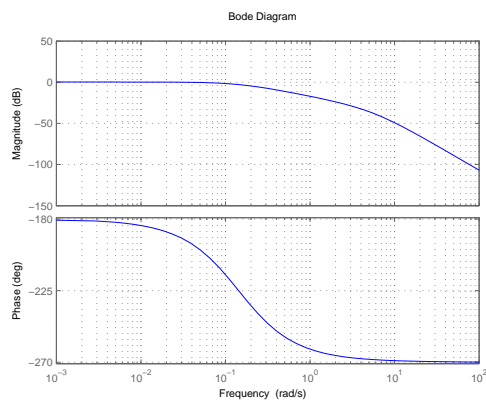


Figura 5.4: Bode con integratore

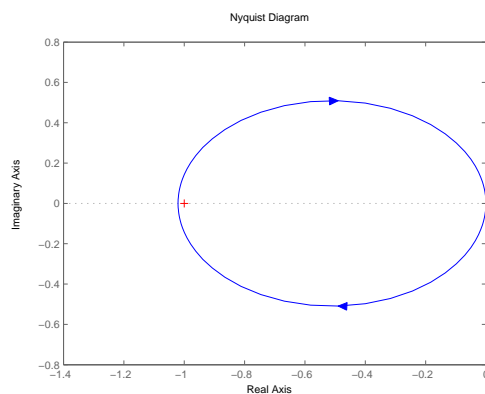


Figura 5.5: Nyquist con integratore

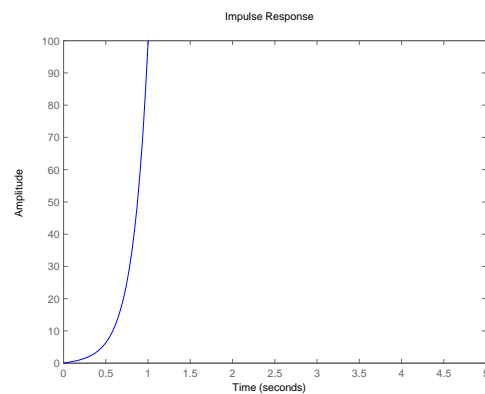


Figura 5.6: Risposta impulsiva con integratore in retroazione

Dal diagramma di Nyquist si nota che la risposta non sarà stabile, poiché si circonda il punto -1 ma lo si fa in senso orario pertanto secondo il criterio di Nyquist abbiamo due poli a parte reale positiva. Per far ruotare la curva del diagramma in senso antiorario è necessario aggiungere fase al diagramma di Bode, aggiungendo uno zero al controllore in -1 avremmo:

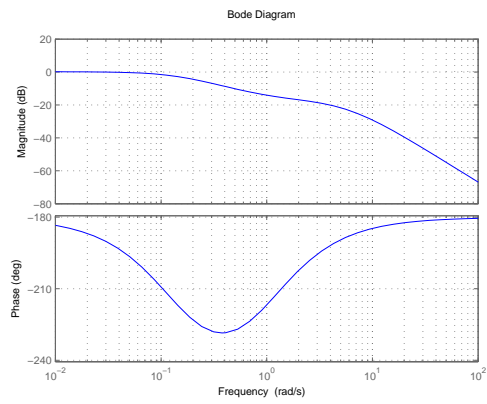


Figura 5.7: Bode con controllore $\frac{s+1}{s}$

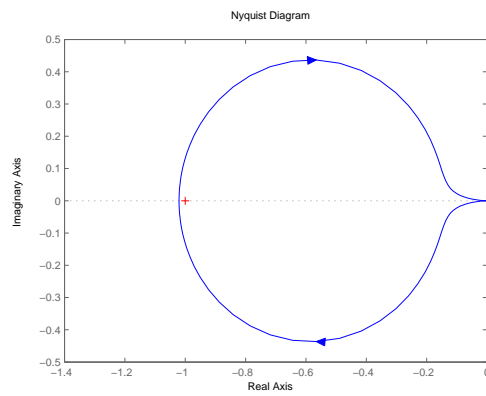


Figura 5.8: Nyquist con controllore $\frac{s+1}{s}$

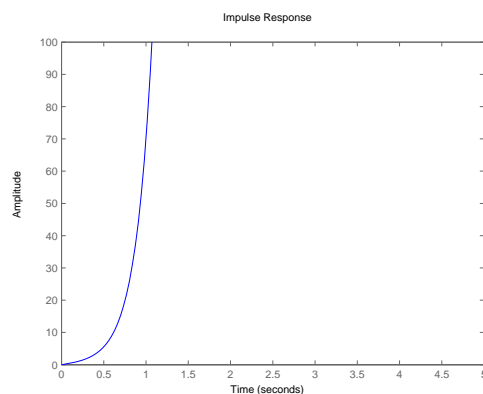


Figura 5.9: Risposta impulsiva con controllore $\frac{s+1}{s}$

dai presenti grafici possiamo notare che la fase aggiunta non è sufficiente, di conseguenza dobbiamo aggiungere un'altro zero, sempre in -1; in tal modo abbiamo uno zero doppio in -1, ed i seguenti grafici:

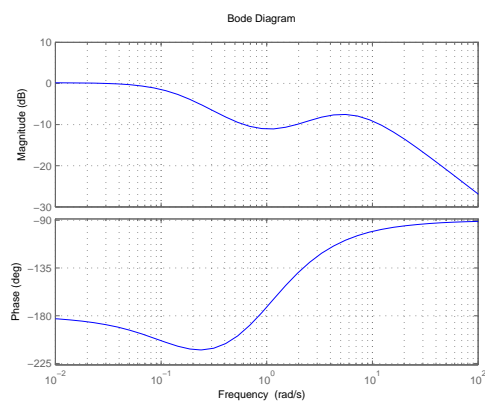


Figura 5.10: Bode con controllore $\frac{(s+1)^2}{s}$

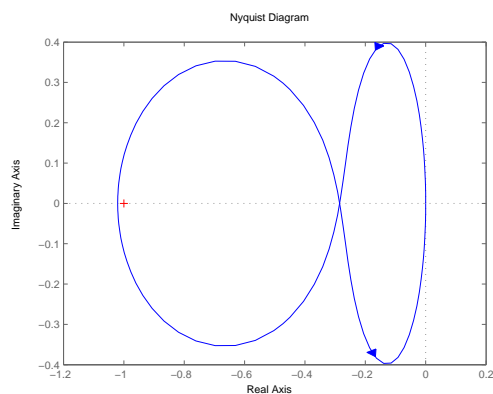


Figura 5.11: Nyquist con controllore $\frac{(s+1)^2}{s}$

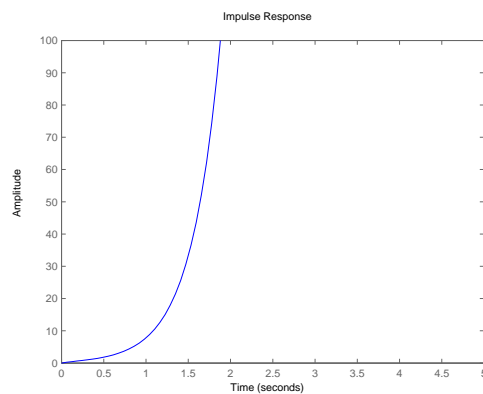


Figura 5.12: Risposta impulsiva con controllore $\frac{(s+1)^2}{s}$

si noti ora che il grafico è composto da due circonferenze una in senso orario che circonda -1 e una in senso antiorario.

Pertanto aumentando il guadagno dunque si ottiene un'espansione del grafico verso sinistra, in tal modo la circonferenza con senso anti-orario accerchierà il punto -1.

Aggiungendo un guadagno $K=10$ si ottiene:

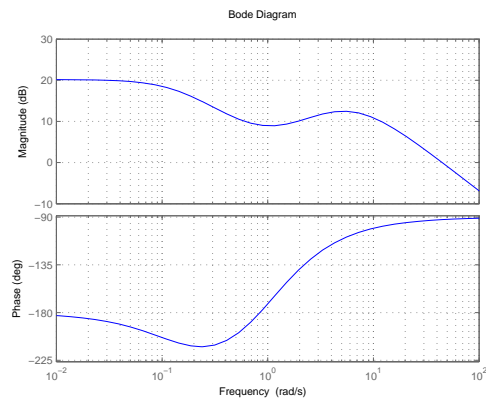


Figura 5.13: Bode con controllore $10 \frac{(s+1)^2}{s}$

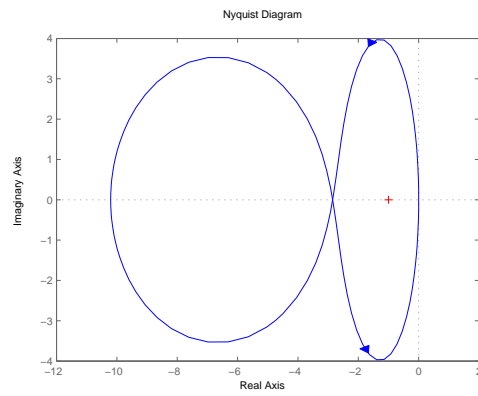


Figura 5.14: Nyquist con controllore $10 \frac{(s+1)^2}{s}$

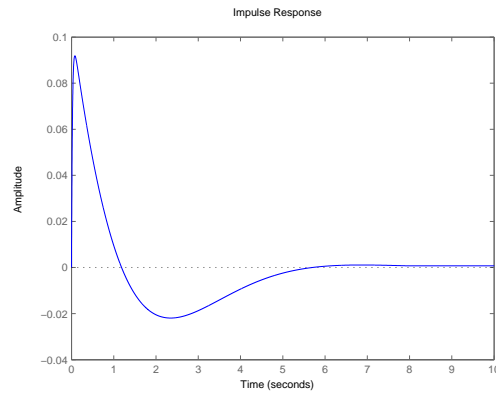


Figura 5.15: Risposta impulsiva con controllore $10 \frac{(s+1)^2}{s}$

Come si vede il sistema ora è stabile ma le specifiche di progetto non sono rispettate infatti l'overshoot è superiore a 0,05 rad ed anche il tempo di assestamento supera i 5 secondi.

Per risolvere questo problema dobbiamo agire sugli zeri e sul guadagno del controllore. Ad esempio mettendo uno zero in -5 uno in -2 ed un guadagno di 20 otterremo una risposta stabile che soddisfa le specifiche di progetto

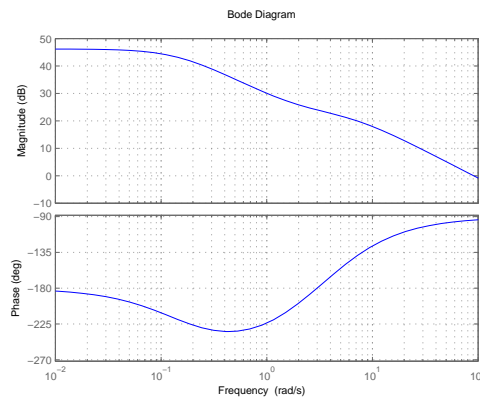


Figura 5.16: Bode con controllore $20 \frac{(s+2)(s+5)}{s}$

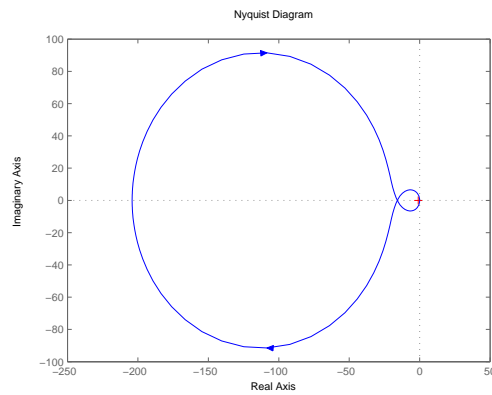


Figura 5.17: Nyquist con controllore $20 \frac{(s+2)(s+5)}{s}$

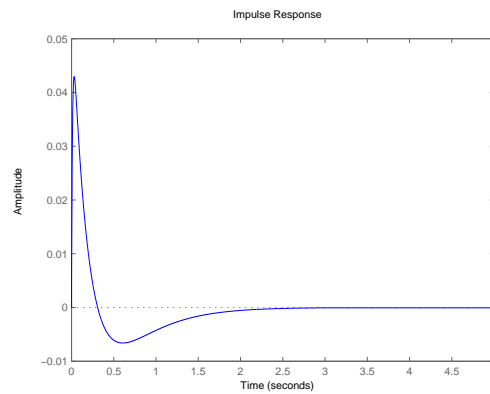


Figura 5.18: Risposta impulsiva con controllore $20 \frac{(s+2)(s+5)}{s}$

Notiamo però che la FDT del controllore è impropria quindi il controllore non è realizzabile, l'unica cosa da fare ora consiste nell'aggiungere un polo in alta frequenza in modo da rendere la funzione propria, questo come vedremo lascerà la risposta inalterata.

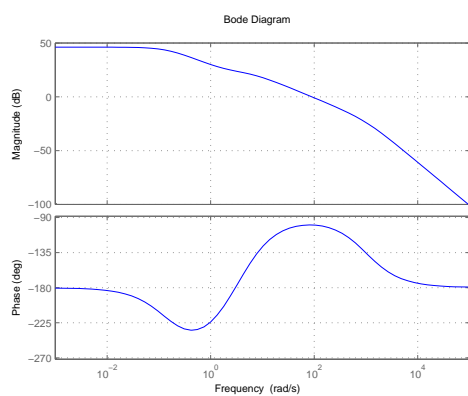


Figura 5.19: Bode con controllore $20 \frac{(s+2)(s+5)}{s(\frac{s}{1000}+1)}$

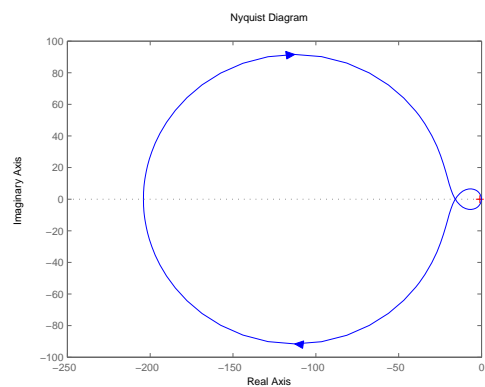


Figura 5.20: Nyquist con controllore $20 \frac{(s+2)(s+5)}{s(\frac{s}{1000}+1)}$

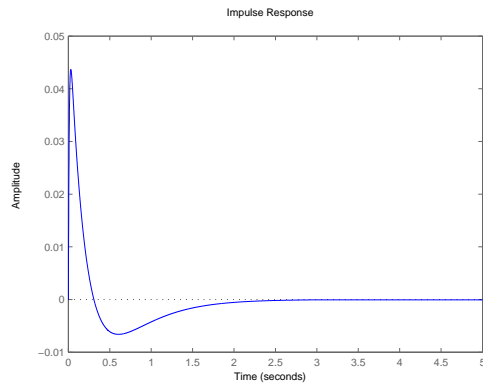


Figura 5.21: Risposta impulsiva con controllore $20 \frac{(s+2)(s+5)}{s(\frac{s}{1000}+1)}$

5.3 Implementazione tramite SIMULINK

Procedendo come per il luogo delle radici toglieremo il blocco a guadagno K ed varieremo il blocco funzione di trasferimento con la funzione: $20 \frac{(s+2)(s+5)}{s(\frac{s}{1000}+1)}$, che abbiamo appena trovato. Lo schema a blocchi sarà il seguente:

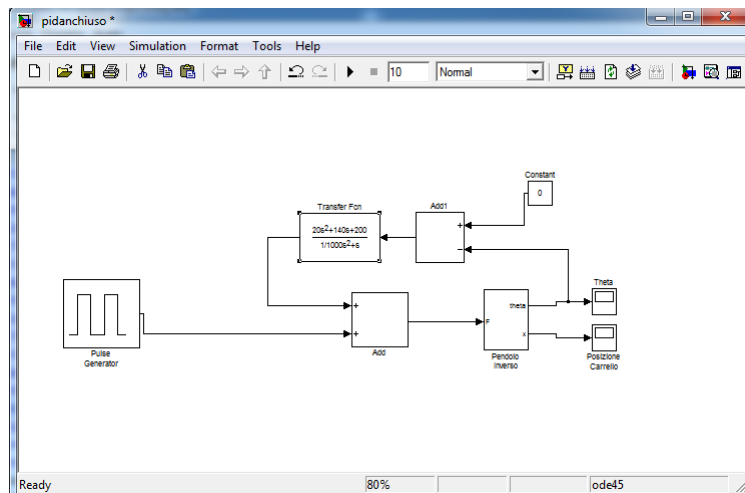


Figura 5.22: Schema a blocchi con controllore: $20 \frac{(s+2)(s+5)}{s(\frac{s}{1000}+1)}$

questa la risposta della simulazione:

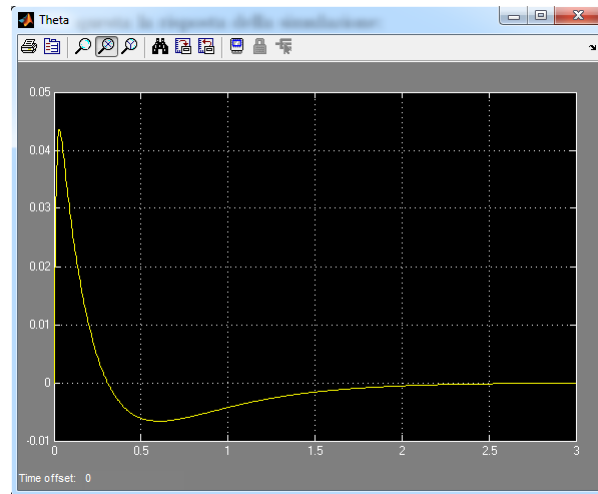


Figura 5.23: Risposta della simulazione con FDT del controllore: $20 \frac{(s+2)(s+5)}{s(\frac{s}{1000}+1)}$

Che soddisfa appieno le specifiche di controllo.

Capitolo 6

Conclusioni

Il fine di questa simulazione è quello di sintetizzare dei sistemi di controllo per un pendolo inverso utilizzando procedure diverse.

Il passo successivo è quello di attuare un confronto tra i tre controllori trovati con i tre diversi metodi; tale procedimento verrà portato avanti osservando la risposta a gradino del sistema controllore-pendolo retroazionato.

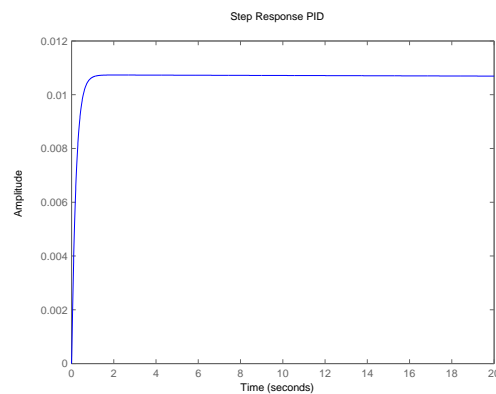


Figura 6.1: Risposta a gradino ottenuta con un controllore PID

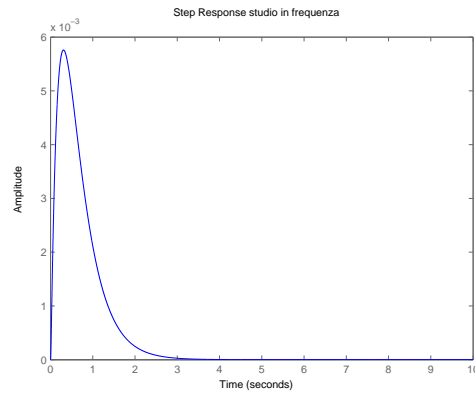


Figura 6.2: Risposta a gradino del controllore ottenuto con lo studio in frequenza

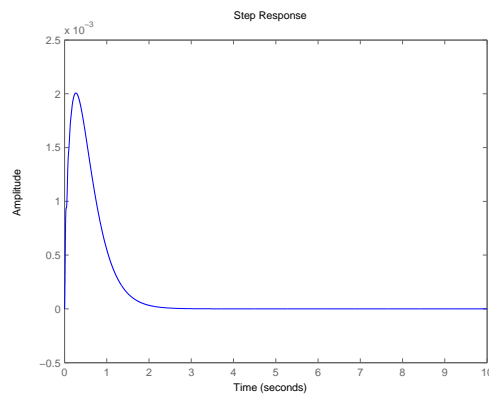


Figura 6.3: Risposta a gradino del controllore ottenuto tramite il luogo

Si può notare come la risposta del PID non sia adeguata perché la risposta a gradino mantiene un'errore costante a regime, anche se piccolo. A dare risultati migliori, invece, è il controllore trovato tramite il luogo delle radici, poiché presenta un'overshoot minore rispetto al controllore ottenuto con lo studio in frequenza, ed è anche più veloce nel portare a zero l'uscita.