
Analisi di un sistema Carrello-Pendolo Inverso e progettazione di un Controllore

Carmine Benedetto, Giacomo Tanganelli

01/03/2012

Indice

1	Sistema Fisico	4
1.1	Descrizione del sistema	4
1.2	Parametri del sistema	4
1.3	Sistema di equazioni	5
1.4	Funzione di Trasferimento e Spazio di Stato	6
2	Implementazione in MATLAB	8
2.1	Funzione di Trasferimento	8
2.2	Spazio di Stato	8
3	Controllo mediante Luogo delle Radici	10
4	Controllore PID	14
5	Discretizzazione del Controllore	17
6	Implementazione in Simulink	20
7	Conclusioni	24
8	Riferimenti	26

Elenco delle figure

1	Sistema Carrello - Pendolo Inverso	4
2	Diagramma a Corpo Libero del Sistema	5
3	Risposta Impulsiva FDT	9
4	Risposta dinamica del sistema	10
5	SisoTool	10
6	Sistema in ciclo aperto	11
7	Sistema con controllore	12
8	Risposta impulsiva	13
9	Scelta di K per soddisfare i requisiti	13
10	Risposta impulsiva	14
11	Risposta Impulsiva con Controllore PID (1)	15
12	Risposta Impulsiva con Controllore PID (2)	15
13	Risposta Impulsiva con Controllore PID (3)	16
14	PID ottimizzato	16
15	Diagrammi di Bode	17
16	Luogo delle Radici con Controllore Discretizzato	18
17	Risposta Impulsiva Campionata	18
18	Controllore Discretizzato (2)	20
19	Pendolo Inverso Simulink	20
20	Sistema non Controllato Simulink	21
21	Uscita Sistema non Controllato Simulink	21
22	Sistema con Controllore PID Simulink	22
23	Uscita Sistema con Controllore PID Simulink	22
24	Uscita Sistema con Controllore PID Simulink (2)	23
25	Valori del PID	23
26	Disturbo: impulso	23
27	Risposta con disturbo impulsivo	24
28	Disturbo: rumore gaussiano	24
29	Risposta con disturbo gaussiano	24

1 Sistema Fisico

1.1 Descrizione del sistema

Il sistema preso in analisi consta di un carrello in movimento su cui è fissato un pendolo. Il modello matematico e la sua linearizzazione sono stati ripresi da [1]

Dalla figura successiva è molto semplice comprendere il tipo di sistema in trattazione.

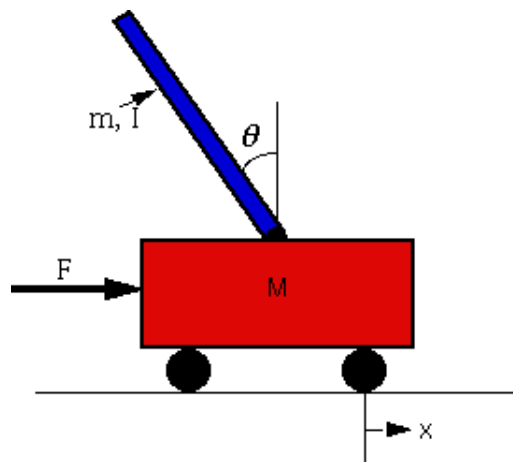


Figura 1: Sistema Carrello - Pendolo Inverso

1.2 Parametri del sistema

Per il sistema assumiamo che siano stabiliti i seguenti valori:

- M , massa del carrello pari a 0.5 kg;
- m , massa del pendolo pari a 0.2 kg;
- b attrito del carrello pari a 0.1 N/m/sec;
- l , lunghezza del pendolo dal centro di massa pari a 0.3 m;
- I , inerzia del pendolo pari a $0.006 \text{ kg} \cdot \text{m}^2$;
- F , forza applicata al carrello;
- x , posizione del carrello;
- θ , angolo verticale del pendolo.

1.3 Sistema di equazioni

Partendo da uno studio sul sistema fatto da terzi, sono state ricavate le equazioni caratteristiche del moto.

Il sistema è descrivibile nel seguente modo:

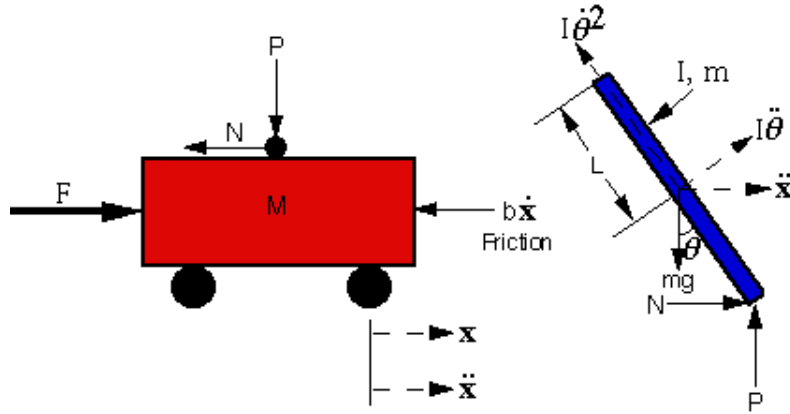


Figura 2: Diagramma a Corpo Libero del Sistema

Equazioni del moto:

Somma delle forze sul diagramma del corpo libero del carrello lungo la direzione orizzontale:

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

Somma delle forze sul diagramma del corpo libero del pendolo lungo la direzione orizzontale:

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

da cui si ricava:

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

inoltre sommando le forze sul pendolo in direzione verticale si ottiene:

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

sommando i momenti nel centro del pendolo otteniamo:

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

da cui combinando le ultime due otteniamo:

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

A questo punto linearizzando il sistema con $\theta = \pi + \phi$ (dove ϕ è un angolo molto piccolo) in modo che $\cos\theta = -1$ e $\sin\theta = -\phi$ e $\ddot{\theta} = 0$ otteniamo:

$$\begin{aligned}(I + ml^2)\ddot{\phi} - mgl\phi &= ml\ddot{x} \\ (M + m)\ddot{x} + b\dot{x} - ml\ddot{\phi} &= u\end{aligned}$$

1.4 Funzione di Trasferimento e Spazio di Stato

Basandoci sempre sullo studio del sistema considerato in precedenza, siamo in grado di ricavare la rappresentazione secondo la Funzione di Trasferimento e secondo lo Spazio di Stato.

Per ottenere la funzione di trasferimento del sistema di equazioni linearizzato in maniera analitica, bisogna prima considerare la trasformata di Laplace del sistema di equazioni:

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

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

Cominciamo risolvendo la prima equazione per $X(s)$

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

e sostituiamolo nella seconda equazione, ottenendo:

$$(M + m) \left[\frac{(I + ml^2)}{ml} + \frac{g}{s} \right] \Phi(s)s^2 + b \left[\frac{(I + ml^2)}{ml} + \frac{g}{s} \right] \Phi(s) - ml\Phi(s)s^2 = U(s)$$

Riordiniamo la funzione di trasferimento:

$$\frac{\Phi(s)}{U(s)} = \frac{\frac{ml}{q}s^2}{s^4 + \frac{b(I+ml^2)}{q}s^3 - \frac{(M+m)mgl}{q}s^2 - \frac{bmgl}{q}s}$$

dove

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

Effettuando la cancellazione del polo e dello zero nell'origine, la funzione di trasferimento diventa:

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

Considerando le formule ottenute, la rappresentazione in spazio di stato risulta:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\Phi} \\ \ddot{\Phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-(I+ml^2)b}{I(M+m)+Mml^2} & \frac{m^2gl^2}{I(M+m)+Mml^2} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{-mlb}{I(M+m)+Mml^2} & \frac{mgl(M+m)}{I(M+m)+Mml^2} & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\Phi} \\ \ddot{\Phi} \end{bmatrix}$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\Phi} \\ \ddot{\Phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u$$

2 Implementazione in MATLAB

2.1 Funzione di Trasferimento

Possiamo implementare la Funzione di Trasferimento in MATLAB usando i seguenti comandi:

```
M = 0.5;
m = 0.2;
b = 0.1;
i = 0.006;
g = 9.8;
l = 0.3;
q = (M+m)*(i+m*l^2)-(m*l)^2;
num = [m*l/q 0];
den = [1 b*(i+m*l^2)/q -(M+m)*m*g*l/q -b*m*g*l/q];
pend=tf(num,den)
```

Quello che otteniamo è un output del tipo:

$$\frac{4.545s}{s^3 + 0.1818s^2 - 31.18s - 4.455}$$

Se vogliamo verificare la Risposta Impulsiva del Sistema, inseriamo i comandi:

```
t=0:0.01:5;
impulse(pend,t)
axis([0 1 0 60])
```

Quello che otteniamo è un grafico di questo come quello di Figura 3.

2.2 Spazio di Stato

Per ottenere la rappresentazione del sistema in forma di Spazio di Stato possiamo utilizzare i seguenti comandi:

```
% Modellizzazione pendolo inverso - Spazio di Stato %
M = 0.5;      % massa del carrello
m = 0.2;      % massa del pendolo
b = 0.1;      % attrito del carrello
i = 0.006;    % inerzia del pendolo
g = 9.8;      % accelerazione g
```

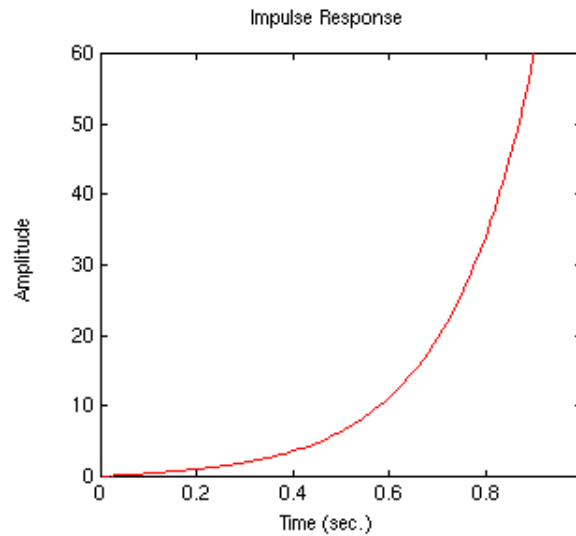



Figura 3: Risposta Impulsiva FDT

```

l = 0.3;      % lunghezza dal centro di massa del pendolo
p = i*(M+m)+M*m*l^2; % denominatore per le matrici A e B

A = [0      1      0      0;
      0 -(i+m*l^2)*b/p (m^2*g*l^2)/p 0;
      0      0      0      1;
      0 -(m*l*b)/p      m*g*l*(M+m)/p 0]
B = [      0;      (i+m*l^2)/p;      0;      m*l/p]
C = [1 0 0 0;      0 0 1 0]
D = [0;      0]

```

```

% Pendolo inverso in Spazio di Stato %
pend=ss(A,B,C,D);

```

Per rappresentare la risposta dinamica del sistema considerando un input a gradino unitario, possiamo utilizzare i seguenti comandi:

```

% Risposta a Gradino a Ciclo Aperto %
T=0:0.05:10;
U=1*ones(size(T));
[Y,T,X]=lsim(pend,U,T); % risposta dinamica del sistema
plot(T,Y)
axis([0 2 0 100])

```

Quello che otteniamo è:

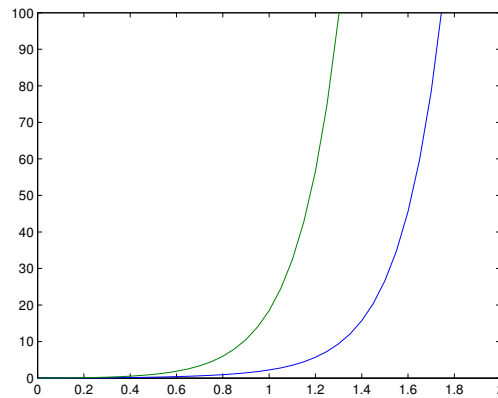


Figura 4: Risposta dinamica del sistema

3 Controllo mediante Luogo delle Radici

Per effettuare la progettazione mediante il luogo delle radici MATLAB mette a disposizione uno strumento molto utile e versatile denominato *sisotool*. Riprendiamo la precedente funzione di trasferimento relativa al pendolo inverso (pend) ed invochiamo il comando:

```
>> sisotool
```

Quello che si ottiene è:

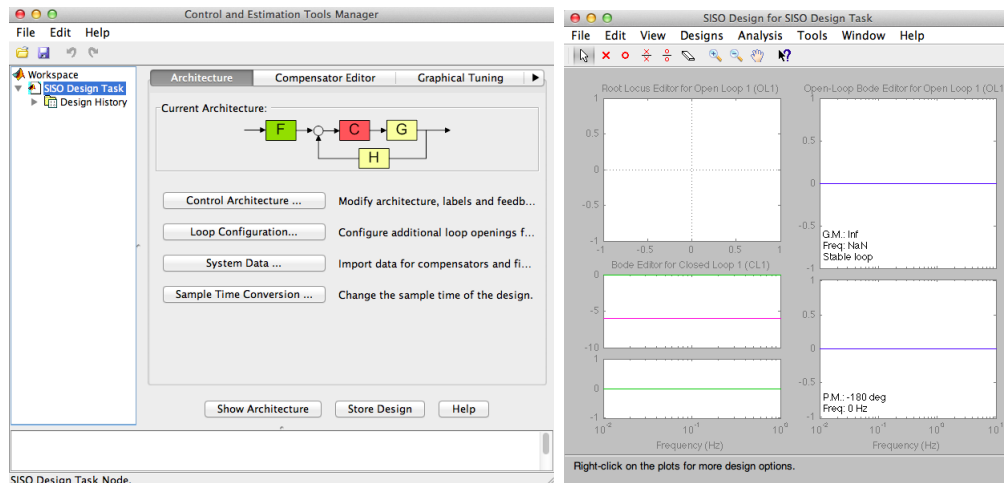


Figura 5: SisoTool

Cliccando sul pulsante *System Data* è possibile impostare i dati del problema ed in particolare la funzione *G* di trasferimento in ciclo aperto. Una

volta impostata G, i grafici cambiano mostrando il luogo delle radici ed il diagramma di Bode per il ciclo aperto e per il ciclo chiuso:

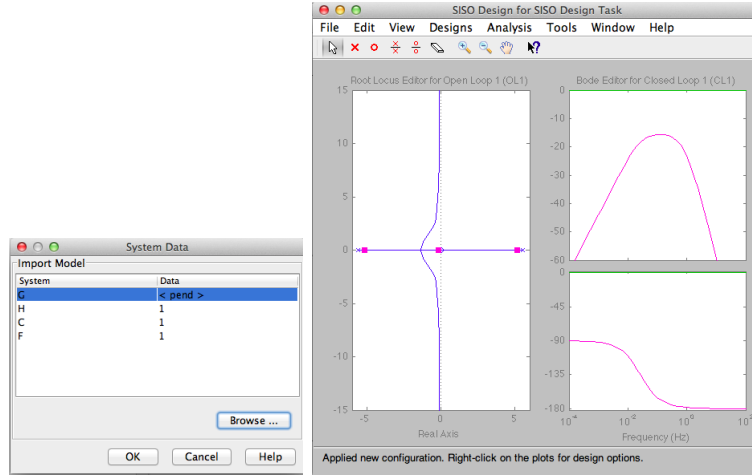


Figura 6: Sistema in ciclo aperto

Come si nota dal luogo delle radici, il sistema è instabile poiché presenta un polo nel semipiano positivo. Per stabilizzare il sistema possiamo procedere in diversi modi. In [1] si propone di risolvere realizzando un controllore con un polo nell'origine (in modo da provocare una cancellazione dei poli e degli zeri nell'origine), un polo nel semipiano negativo quanto più lontano possibile (per portare il sistema in uno stato stabile) e due zeri quanto più piccoli possibile (inseriti per ottenere un numero di asintoti opportuno). Per ottenere questo risultato si accede al pannello *Compensator Editor* di sisotool ottenendo un sistema complessivo come illustrato in Figura 7.

In particolare possiamo notare che adesso il sistema è stabile per adeguati valori del guadagno. L'esempio crea un controllore del tipo:

$$C(s) = 50.904 \frac{(1 + 0.33s)(1 + 0.25s)}{s(1 + 0.02s)} = 50.904 \frac{(s + 4)(s + 3)}{s(s + 50)}$$

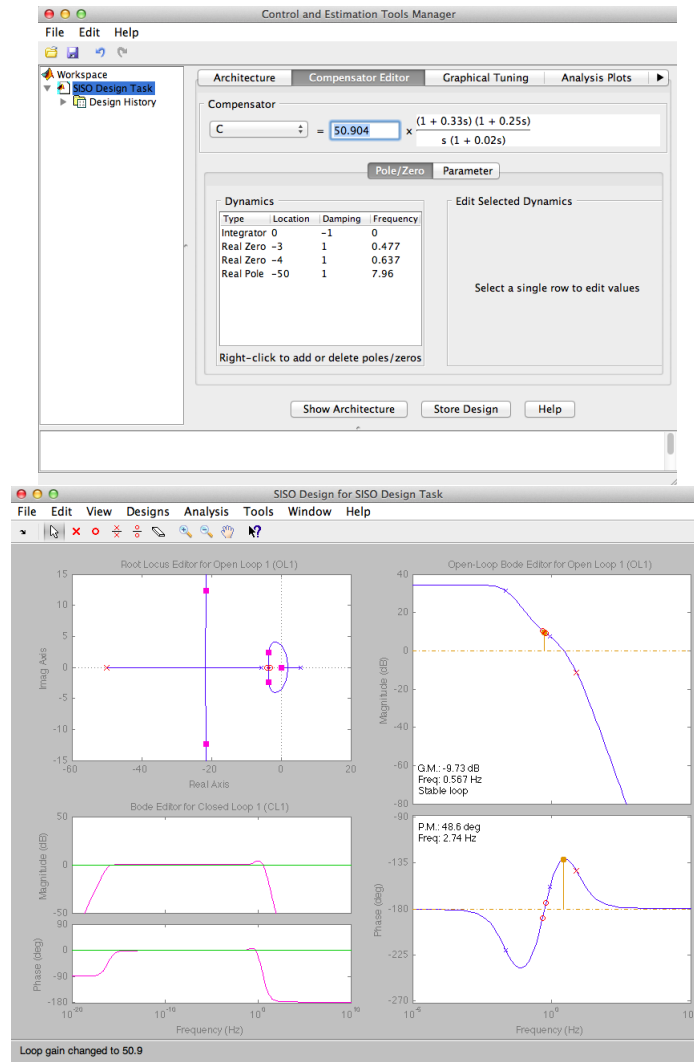


Figura 7: Sistema con controllore

Analizzando la risposta in ciclo chiuso all'impulso otteniamo il risultato di Figura 8.

Ovviamente questo risultato non è certo l'unico possibile, in particolare in [2] si propone un controllore leggermente diverso, ovvero:

$$C(s) = K \frac{(1 + 0.1s)^2}{s} = K \frac{(s + 10)^2}{s}$$

dove il guadagno K deve essere scelto in modo da far diventare il sistema stabile in accordo con determinate scelte di progetto. In particolare ipotizziamo di volere:

3 CONTROLLO MEDIANTE LUOGO DELLE RADICI

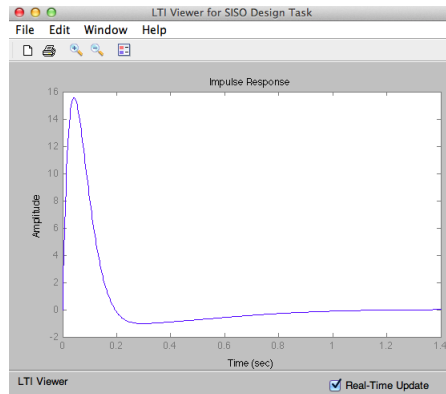


Figura 8: Risposta impulsiva

- Tempo di risposta a regime : 1 sec
- Overshoot max 20%

In tal caso avvalendoci di sisotool possiamo imporre tale vincoli e scegliere K opportunamente.

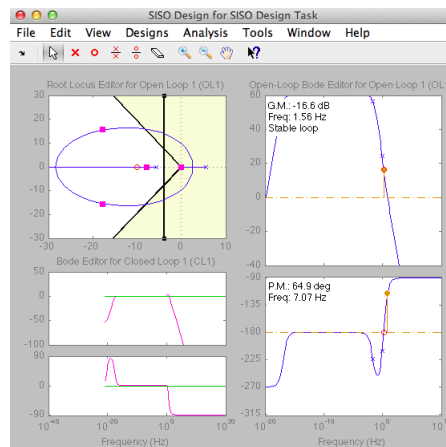


Figura 9: Scelta di K per soddisfare i requisiti

In tal caso possiamo scegliere il guadagno K in accordo con i requisiti ipotizzati.

$$C(s) = 945 \frac{(1 + 01s)^2}{s} = 945 \frac{(s + 10)^2}{s}$$

La cui risposta impulsiva è:

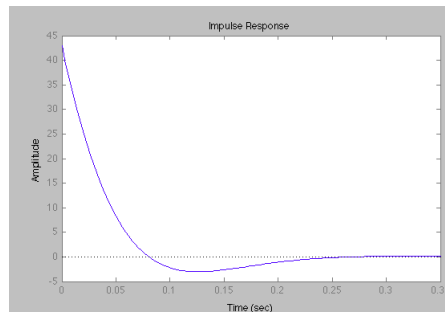


Figura 10: Risposta impulsiva

4 Controllore PID

Un altro modo per rendere il sistema stabile è quello di inserire un opportuno controllore PID. Possiamo procedere inserendo un controllore con tutte le componenti (proporzionale, integrativa e derivativa) poste a 1.

```
% Controllore PID %  
Kd = 1; % componente derivativa  
Kp = 1; % componente proporzionale  
Ki = 1; % componente integrativa  
  
contr=tf([Kd Kp Ki],[1 0]); % funzione di trasferimento del controllore  
  
% Funzione di Traferimento a Ciclo Chiuso con Controllore PID %  
sys_cl=feedback(pend,contr);  
  
% Risposta Impulsiva a Ciclo Chiuso %  
t=0:0.01:5; % intervallo  
impulse(sys_cl,t) % risposta impulsiva  
axis([0 1.5 0 40]) % settaggio degli assi
```

Come era facile intuire, il sistema risulta instabile e il grafico della Risposta Impulsiva ce ne dà conferma. A questo punto possiamo pensare di incrementare la componente proporzionale del controllore.

In questo modo il sistema si stabilizzerà dopo circa 2 secondi. Si nota però che l'overshoot (sovraelongazione) risulta troppo elevato, quindi si potrebbe agire sulla componente derivativa del controllore PID per alleviare il problema.

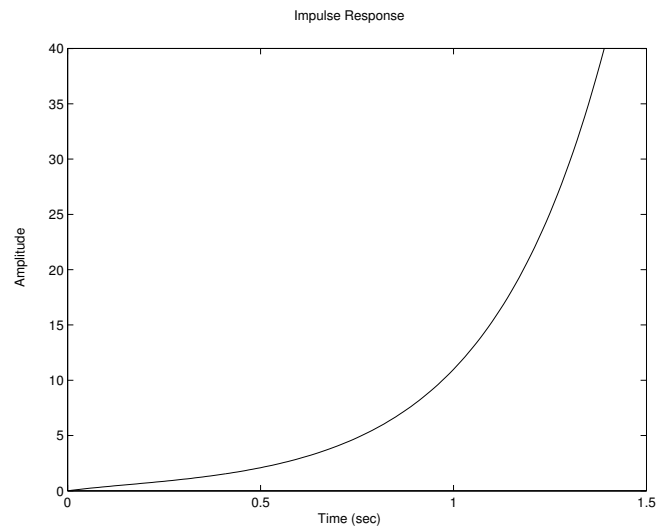


Figura 11: Risposta Impulsiva con Controllore PID (1)

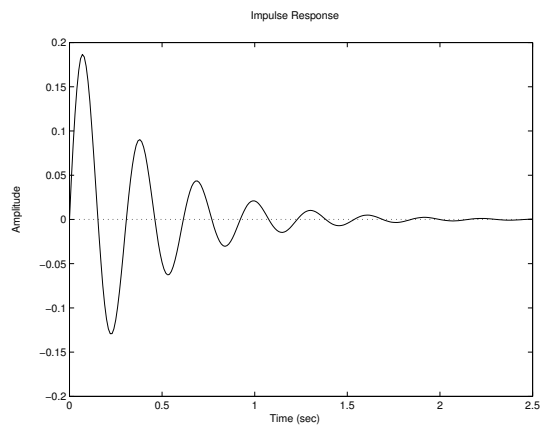


Figura 12: Risposta Impulsiva con Controllore PID (2)

$K_d = 20;$
 $K_p = 100;$
 $K_i = 1;$

In questo modo riusciamo ad ottenere un risultato accettabile come si evince dalla Figura 13.

Un PID più preciso può essere ottenuto sfruttando ancora una volta *siso-tool* che permette di ottimizzare automaticamente i parametri (K,I,D) massimizzando una certa funzione obiettivo.

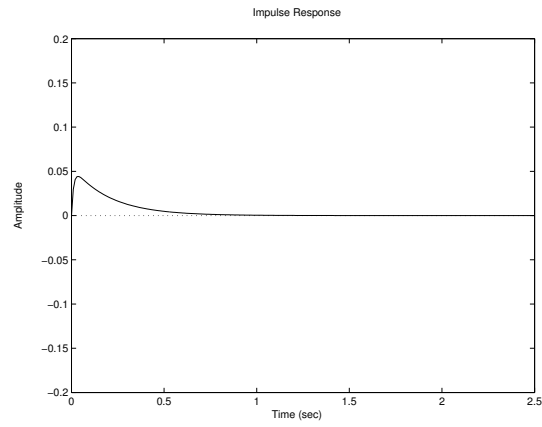


Figura 13: Risposta Impulsiva con Controllore PID (3)

Ipotizzando di voler ottimizzare il tempo di risposta quello che si ottiene utilizzando un controllore PID è:

$$C(s) = 33.196 \frac{(1 + 0.07s)(1 + 2.2s)}{s}$$

che equivale a:

$$K_p = 75.354$$

$$K_i = 33.196$$

$$K_d = 5.1122$$

la cui risposta è:

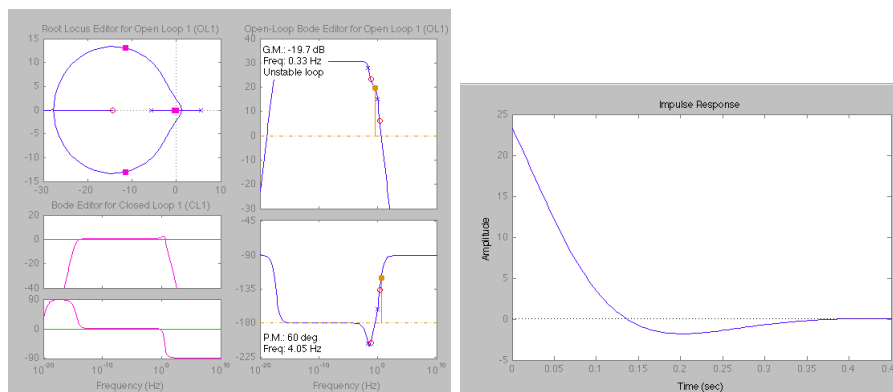


Figura 14: PID ottimizzato

5 Discretizzazione del Controllore

Per la discretizzazione siamo partiti da un controllore realizzato attraverso i seguenti comandi MATLAB:

```
num = conv([1 1.1],[1 5])  
den = [1 0]  
k = 10  
contr = k*tf(num,den);
```

In questo modo generiamo un controllore con una funzione di trasferimento:

$$C(s) = \frac{10s^2 + 61s + 55}{s}$$

Mediante i diagrammi di Bode abbiamo analizzato la banda a -3dB (0.0235 Hz).

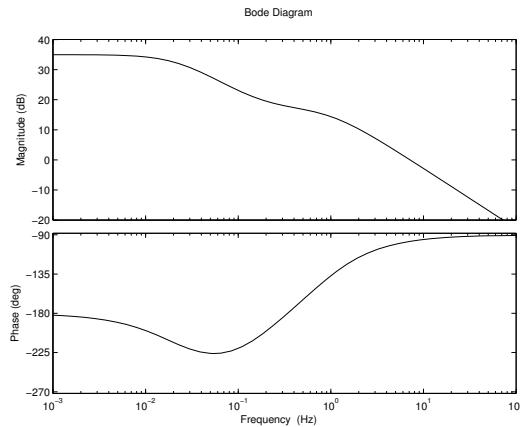


Figura 15: Diagrammi di Bode

In seguito abbiamo utilizzato la convenzione secondo cui

$$\frac{1}{T} > 60 * B_{-3dB}$$

ottenendo

$$T < 0.7sec$$

Le prove sono state perciò eseguite con un periodo di campionamento di 0.5 sec che, come si vede dal grafico del luogo delle radici, mantiene i poli

all'interno del cerchio unitario dopo la discretizzazione eseguita mediante Tustin.

$$s \longleftrightarrow \frac{2}{T} \frac{z - 1}{z + 1}$$

Transfer function:

$$\frac{0.008983 z^5 - 0.008983 z^4 - 0.01797 z^3 + 0.01797 z^2 + 0.008983 z - 0.008983}{z^5 - 1.605 z^4 - 0.1776 z^3 + 1.135 z^2 - 0.3165 z - 0.03589}$$

Sampling time: 0.5

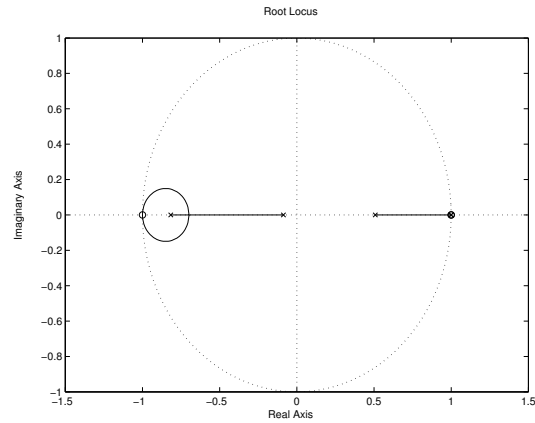


Figura 16: Luogo delle Radici con Controllore Discretizzato

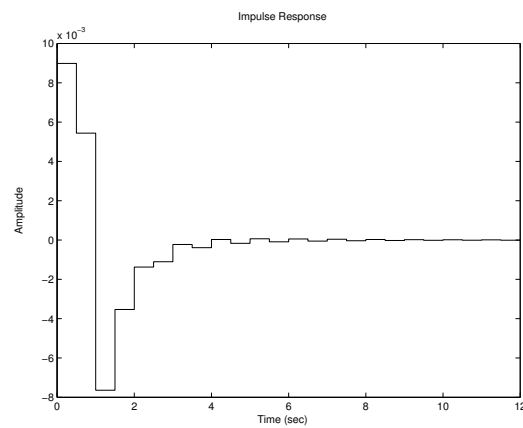


Figura 17: Risposta Impulsiva Campionata

Variando il tempo di campionamento si osserva che varia la risposta impulsiva del segnale campionato, inoltre scegliendo un tempo di campionamento troppo piccolo, i poli escono dal cerchio unitario portando il sistema (con il controllore discreto) a divergere.

Per ottenere un risultato più performante, abbiamo discretizzato il secondo controllore preso in analisi nel paragrafo precedente.

Realizziamo il controllore in MATLAB con i seguenti comandi:

```
k=945
cont=tf([1 20 100], [1 0])
sys_cl=feedback(pend,k*contr)
```

Analizzando la banda a -3dB del relativo diagramma di Bode (3.87 Hz), si ottiene con la stessa procedura utilizzata in precedenza un tempo di campionamento:

$$T < 4.3 msec$$

Scegliendo $T=3msec$, otteniamo un controllore discretizzato nel seguente modo:

```
%B-3 3.87 Hz -> 0.0043 sec
T=0.003
```

```
fz=c2d(sys_cl,T,'tustin')
figure(3)
rlocus(fz)
figure(4)
impz(fz)
```

Transfer function:

```
1.339e-06 z^4 - 2.678e-06 z^2 + 1.339e-06
-----
z^4 - 2.21 z^3 + 0.7349 z^2 + 1.163 z - 0.6876
```

Sampling time: 0.003

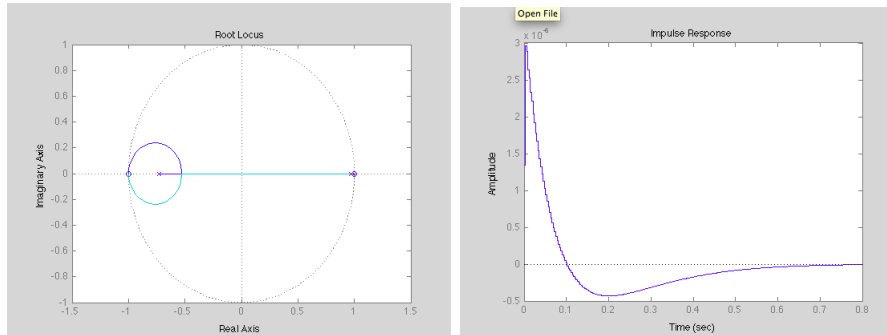


Figura 18: Controllore Discretizzato (2)

6 Implementazione in Simulink

Per testare il corretto funzionamento del Sistema realizzato, sono stati implementati diversi moduli Simulink per poter effettuare delle simulazioni. Per prima cosa, con riferimento allo studio [1], abbiamo creato l'elemento relativo al Pendolo Inverso (NON LINEARE) comprensivo di tutti i parametri necessari per lo studio del sistema.

Il modello fornito risultava «deprecated», quindi abbiamo aggiornato i moduli al fine di ottenere la compatibilità con le recenti versioni di MATLAB.

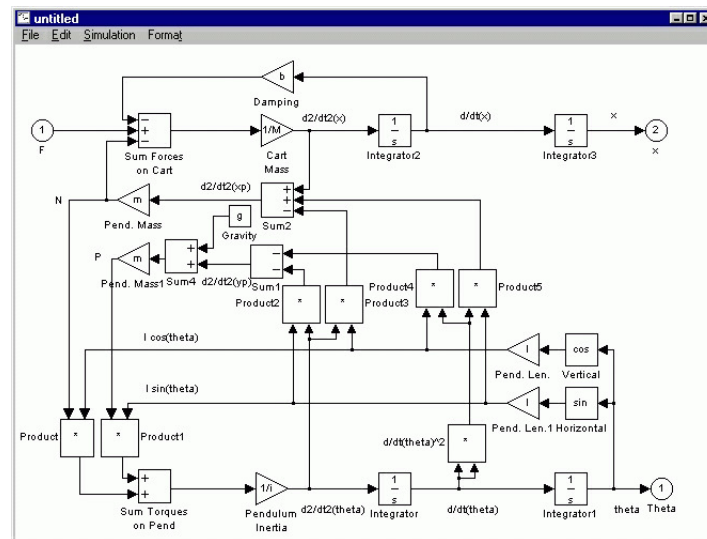


Figura 19: Pendolo Inverso Simulink

Successivamente abbiamo studiato il sistema con un Generatore di Impulsi in ingresso.

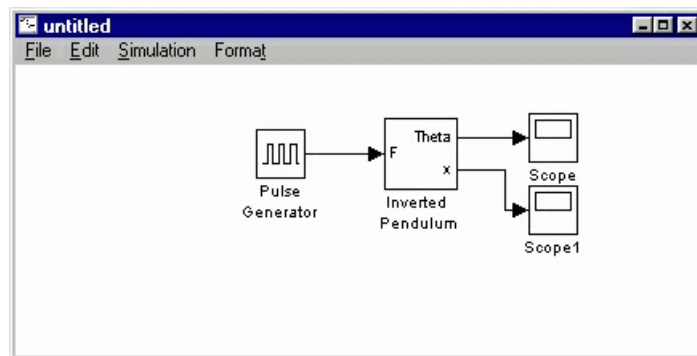


Figura 20: Sistema non Controllato Simulink

Come previsto, otteniamo un risultato instabile.

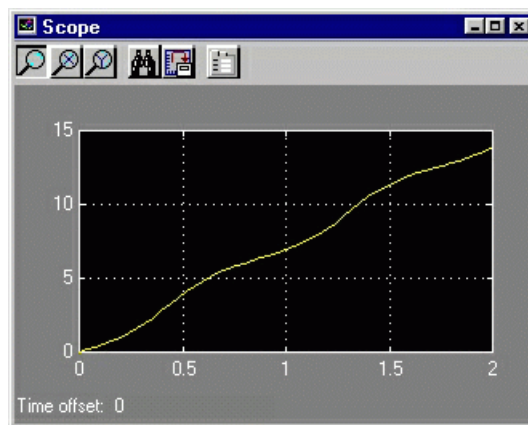


Figura 21: Uscita Sistema non Controllato Simulink

Ricalchiamo lo studio già effettuato in MATLAB e inseriamo nel Sistema utilizzato fino ad ora un controllore PID con gli stessi parametri utilizzati in precedenza, ovvero:

- componente proporzionale pari a 100:
- componente integrativa pari a 20:
- componente derivativa pari a 1.

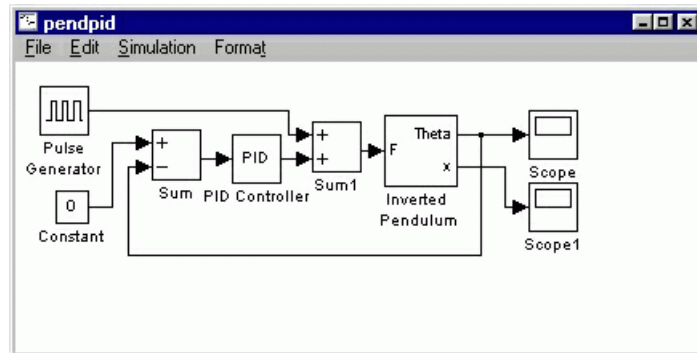


Figura 22: Sistema con Controllore PID Simulink

Come in precedenza, siamo riusciti a stabilizzare l'uscita del sistema.

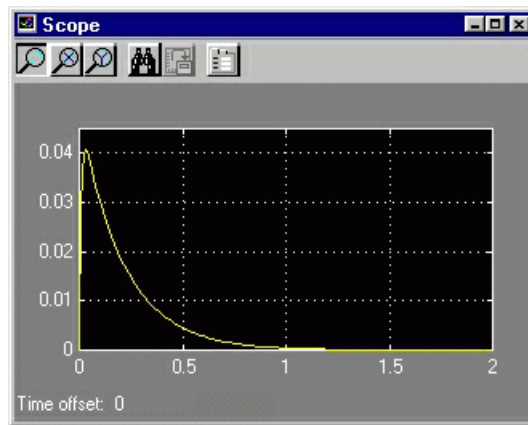


Figura 23: Uscita Sistema con Controllore PID Simulink

Utilizzando il PID ottimizzato (creato con l'aiuto di sisotool), otteniamo una risposta impulsiva in cui viene evidenziato che il Sistema si stabilizza più lentamente.

Tuttavia l'ottimizzazione di questo PID rende il sistema più robusto, infatti, introducendo un disturbo, il Sistema si comporta in maniera efficiente.

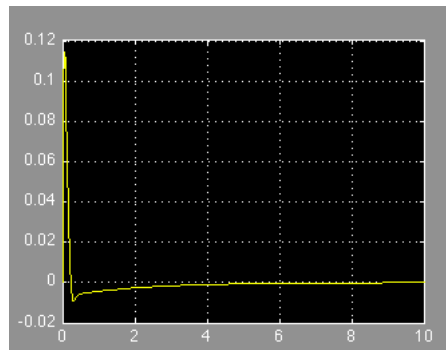


Figura 24: Uscita Sistema con Controllore PID Simulink (2)

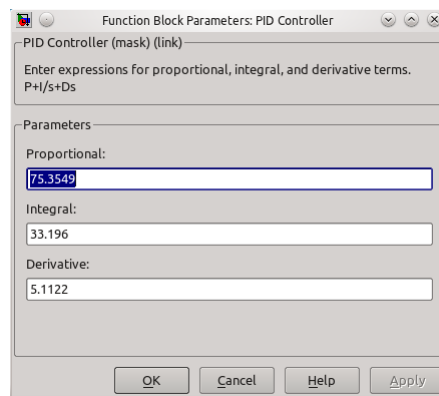


Figura 25: Valori del PID

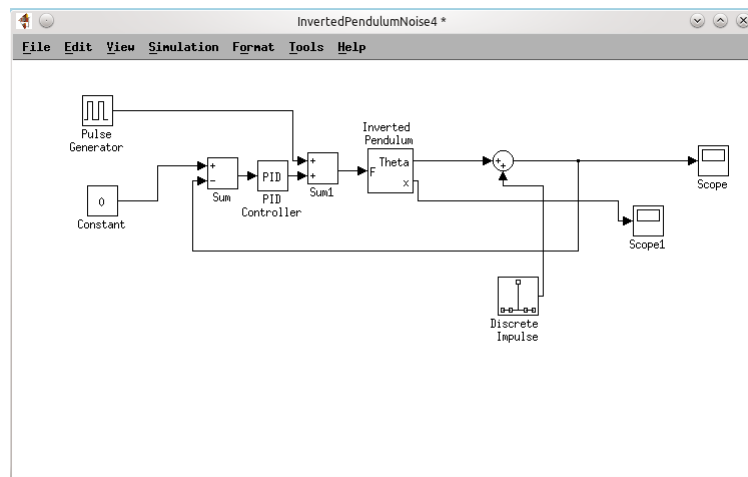


Figura 26: Disturbo: impulso

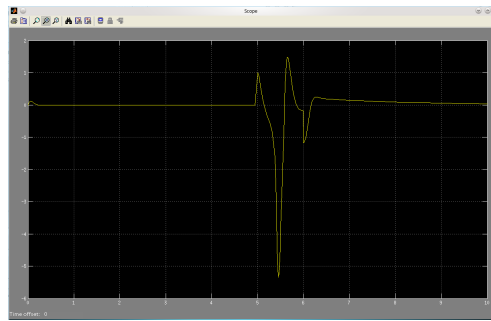


Figura 27: Risposta con disturbo impulsivo

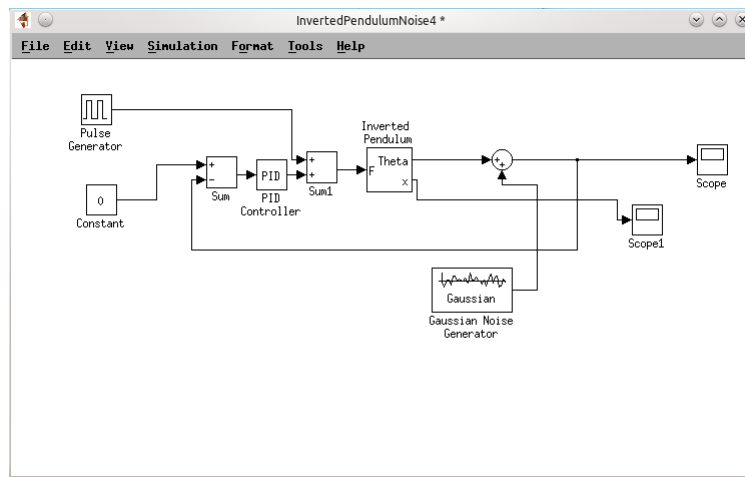


Figura 28: Disturbo: rumore gaussiano

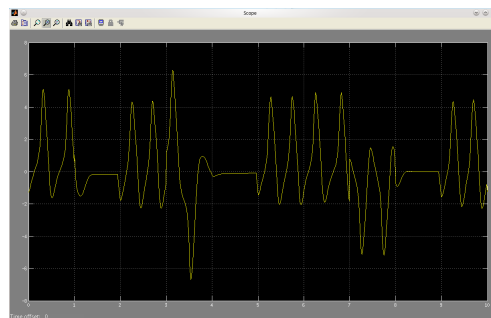


Figura 29: Risposta con disturbo gaussiano

7 Conclusioni

Il problema del Pendolo Inverso è un problema noto e il modello matematico associato è ben definito.

Dal punto di vista del controllo si possono realizzare diverse implementazioni che risolvono lo stesso problema in modo diverso. In particolare abbiamo notato che, nella progettazione, possiamo procedere in modi differenti a seconda delle specifiche da rispettare. Ad esempio, nel paragrafo relativo al Luogo delle Radici, abbiamo sviluppato due controllori, il secondo dei quali, rispettava dei particolari requisiti sul tempo di assestamento e overshoot.

Grazie agli strumenti forniti da MATLAB, la progettazione risulta agevole sia nella prima fase di definizione di particolari controllori (sisotool), sia nello studio dinamico del sistema (simulink).

Proprio grazie a simulink siamo riusciti a testare la bontà del modello considerando l'azione di particolari disturbi.

8 Riferimenti

- [1] <http://www.engin.umich.edu/group/ctm/examples/pend/invpen.html>
- [2] <http://www.engr.usask.ca/classes/EE/480/Inverted%20Pendulum.pdf>
- [3] http://ophelia.princeton.edu/~page/phy210/2007_projects/InvertedPendulum.pdf
- [4] http://www.ece.ucsb.edu/~roy/student_projects/Johnny_Lam_report_238.pdf
- [5] <http://arri.uta.edu/acs/ee4314/lectures/Lecture%207b.pdf>
- [6] http://web.mit.edu/2.737/www/extra_files/andrew.pdf