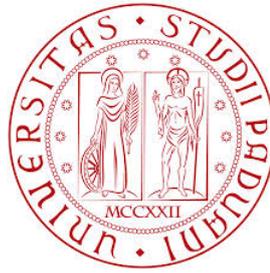


CONTROLLO DEL BECCHEGGIO
DI UN AEREO IN AMBIENTE
MATLAB / SIMULINK

RELATORE: Prof. Alessandro Beghi

Laureando: Giulio Storer

A.A. 2012-2013



UNIVERSITÀ DEGLI STUDI DI PADOVA
CORSO DI LAUREA IN INGEGNERIA ELETTRONICA

CONTROLLO DEL BECCHEGGIO
DI UN AEREO IN AMBIENTE
MATLAB / SIMULINK

RELATORE: Prof. Alessandro Beghi

LAUREANDO: *Giulio Storer*

Padova, A.A. 2012/2013

Indice

1	INTRODUZIONE	1
2	ANALISI DEL PROGETTO	2
2.1	Introduzione	2
2.2	Specifiche di progetto	4
2.3	Derivazione del modello matematico	5
3	ANALISI DEL SISTEMA IN CATENA APERTA	6
3.1	Risposta al gradino	6
3.2	Risposta alla sinusoide	7
4	CONTROLLORE PID	10
4.1	Introduzione	10
4.2	Controllore P	11
4.3	Controllore PI	15
4.4	Controllore PD	15
4.5	PID	20
5	LUOGO DELLE RADICI	25
5.1	Introduzione	25
5.2	Luogo delle radici e progetto rete corretttrice	26
6	STUDIO IN FREQUENZA	32
6.1	Introduzione	32
6.2	Analisi del sistema	32
6.3	Compensatore	33
6.4	Analisi con il metodo di Nyquist	39

7	CONCLUSIONI	41
7.1	Confronti	41
7.2	Risultati	42

Capitolo 1

INTRODUZIONE

Il controllo automatico del beccheggio di un aereo consiste in un sistema a retroazione in grado di far variare l'angolo degli equilibratori e di conseguenza di poter gestire il beccheggio di un aereo. Il controllore fa sì che l'aereo ottenga l'oscillazione desiderata rispettando i vincoli progettuali di velocità di raggiungimento dell'inclinazione desiderata, oscillazione, assestamento e di non superamento dell'angolo massimo.

Lo scopo dell'elaborato è dunque, di progettare il sistema di controllo del beccheggio di un aereo usando differenti tipi di approccio per la risoluzione del problema: come primo approccio vi è un'analisi del problema con la relativa derivazione del modello matematico. Successivamente viene illustrato il vero e proprio progetto del controllore, sviluppato con diversi approcci: sintesi tramite i controllori P, PI, PD, PID, sintesi mediante il luogo delle radici ed infine sintesi tramite il progetto di reti correttrici nel dominio della frequenza. Per lo svolgimento del progetto è stato utilizzato il software Matlab, con i relativi strumenti di progettazione e simulazione, Simulink e Sisotool.

Capitolo 2

ANALISI DEL PROGETTO

2.1 Introduzione

Nella nostra trattazione, viene eseguita una semplificazione alle equazioni che governano il moto dell'aereo, giungendo ad una forma di equazioni lineari longitudinali e laterali. In questo elaborato viene progettato un pilota automatico che controlla il beccheggio di un aeromobile attraverso la modifica dell'angolo di parti mobili posti alla coda dell'aereo, ovvero gli equilibratori. Viene di seguito analizzato il sistema con i vincoli delle specifiche che deve rispettare.

Viene assunto che il sistema abbia velocità di crociera ed un'altitudine costanti. In questo caso le forze che agiscono sull'aereo lungo l'asse x, ovvero spinta e resistenza, sono bilanciate, come pure quelle lungo l'asse y, il peso e la portanza. Viene inoltre semplificato il problema assumendo che la velocità del velivolo non cambi durante la variazione del beccheggio. Possiamo dunque ora definire le equazioni del moto dell'aereo:

$$\begin{aligned}\dot{\alpha} &= \mu\Omega\sigma[-(C_L + C_D)\alpha + (\frac{1}{\mu} - C_L)q - C_W\sin(\gamma_e)\theta + C_L] \\ \dot{q} &= \frac{\mu\Omega}{2iy}\{[C_M - \eta(C_L + C_D)]\alpha + [C_M + \sigma C_M(1 - \mu C_L)]q + \eta C_W\sin(\gamma_e)\delta_e\} \\ \dot{\theta} &= \Omega q\end{aligned}$$

Legenda:

α = Angolo d'attacco (attack angle)

θ = Angolo di beccheggio (pitch angle)

$\gamma_e = \theta - \alpha$

δ_e = Angolo degli equilibratori
 U = Velocità di volo all'equilibrio
 q = Indice di beccheggio
 i_y = Momento d'inerzia normalizzato
 ρ_e = Densità dell'aria
 S = superficie alare
 $\Omega = \frac{2U}{\bar{c}}$
 \bar{c} = Corda alare media
 m = Massa del velivolo
 $\sigma = \frac{1}{1+\mu C_L}$; $\mu = \frac{\rho_e S \bar{c}}{4m}$
 C_L = Coeff. di portanza
 C_D = Coeff. di resistenza dell'aria
 C_W = Coeff. di peso
 C_M = Coeff. del momento del beccheggio
 $\eta = \mu \sigma C_M$

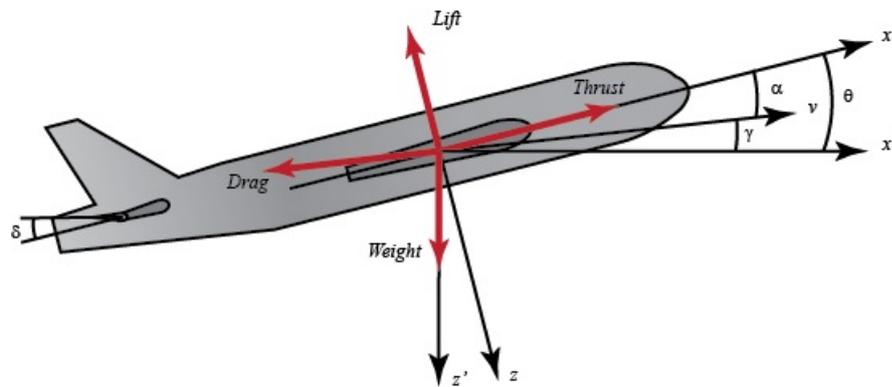


Figura 2.1: Assi e forze di base che agiscono sull'aereo

2.2 Specifiche di progetto

Il sistema deve rispettare i seguenti vincoli progettuali:

1. Sovraelongazione (Overshoot): minore del 10%
2. Tempo di salita (Rise time): minore di 2 secondi
3. Tempo di assestamento (Settling time): minore di 10 secondi
4. Errore a regime (Steady state error): minore del 2%

Esempio:

Ad una variazione degli equilibratori di 0.2 radianti (circa 11°), deve corrispondere una variazione dell'angolo di beccheggio che raggiunga i 0.2 radianti in meno di 2 secondi, senza superare il valore di 0.22 rad e stabilizzarsi tra i 0.196 e 0.204 rad in un tempo inferiore ai 10 secondi.

2.3 Derivazione del modello matematico

Vengono ora assegnati dei valori ai parametri delle equazioni, in modo da poter definire il progetto del controllore. Sono stati forniti i valori di un aereo commerciale della Boeing:

$$\begin{aligned}\dot{\alpha} &= -0.313\alpha + 56.7q + 0.232\delta_e \\ \dot{q} &= -0.0139\alpha - 0.426q + 0.0203\delta_e \\ \dot{\theta} &= 56.7q\end{aligned}$$

Posso ora ricavare la funzione di trasferimento $G(s)$ del sistema, applicando la trasformata di Laplace alle equazioni:

$$\begin{aligned}s\alpha(s) &= -0.313\alpha(s) + 56.7q(s) + 0.232\delta_e(s) \\ sq(s) &= -0.0139\alpha(s) - 0.426q(s) + 0.0203\delta_e(s) \\ s\theta(s) &= 56.7q(s)\end{aligned}$$

Ricavo la funzione di trasferimento $G(s)$:

$$\alpha(s) = \frac{56.7q(s) + 0.232\delta_e(s)}{s + 0.313} \quad (2.1)$$

$$q(s) = \frac{0.0203s - 0.00313}{s^2 + 0.739s + 0.921}\delta_e(s) \quad (2.2)$$

$$\theta(s) = \frac{1.151s + 0.1774}{s^3 + 0.739s^2 + 0.921s}\delta_e(s) \quad (2.3)$$

La funzione di trasferimento risulta dunque:

$$G(s) = \frac{\theta(s)}{\delta_e(s)} = \frac{1.151s + 0.1774}{s^3 + 0.739s^2 + 0.921s} \quad (2.4)$$

Posso ora ricavare il modello di stato:

$$\begin{bmatrix} \dot{\alpha} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} -0.313 & 56.7 & 0 \\ -0.0139 & -0.426 & 0 \\ 0 & 56.7 & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ q \\ \theta \end{bmatrix} + \begin{bmatrix} 0.232 \\ 0.0203 \\ 0 \end{bmatrix} \begin{bmatrix} \delta_e \end{bmatrix}$$

Posso ora descrivere l'uscita del sistema:

$$\begin{bmatrix} y \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ q \\ \theta \end{bmatrix} + \begin{bmatrix} 0 \end{bmatrix} \begin{bmatrix} \delta_e \end{bmatrix}$$

Capitolo 3

ANALISI DEL SISTEMA IN CATENA APERTA

Di seguito viene illustrata la risposta del sistema in catena aperta, ponendo come ingresso il gradino unitario, successivamente un segnale a rampa ed infine un ingresso sinusoidale. Viene dunque verificata la stabilità del sistema, ovvero la capacità di mantenere le sue grandezze in uscita entro valori limitati a fronte della sollecitazione al suo ingresso.

3.1 Risposta al gradino



Figura 3.1: Simulink. Schema sistema in catena aperta

Viene ora illustrata attraverso Matlab, la risposta del sistema al gradino unitario e ne viene verificata la stabilità, imponendo come valore di ampiezza dell'ingresso 0.2:

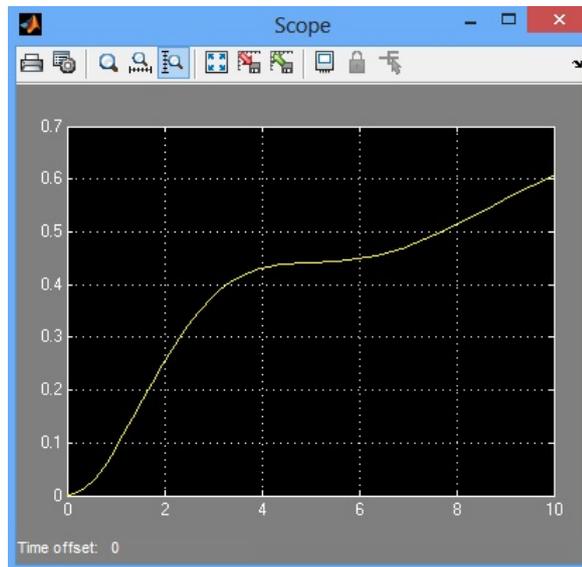


Figura 3.2: Simulink. Risposta al gradino in catena aperta

```
st=0.2;                %ampiezza gradino
num=[1.151 0.1774];   %numeratore pitch
den=[1 0.739 0.921 0]; %denominatore pitch
pitch=tf(num,den);    %FDT pitch normalizzata
openloop=st*pitch;    %FDT pitch con ampiezza gradino
t=0:0.1:15;          %timeframe
step(openloop,t);    %grafico risposta al gradino in catena aperta
```

Possiamo subito notare dal grafico l'instabilità del sistema in catena aperta, in quanto il segnale in uscita tende a divergere al crescere con il tempo.

3.2 Risposta alla sinusoide

3. ANALISI DEL SISTEMA IN CATENA APERTA

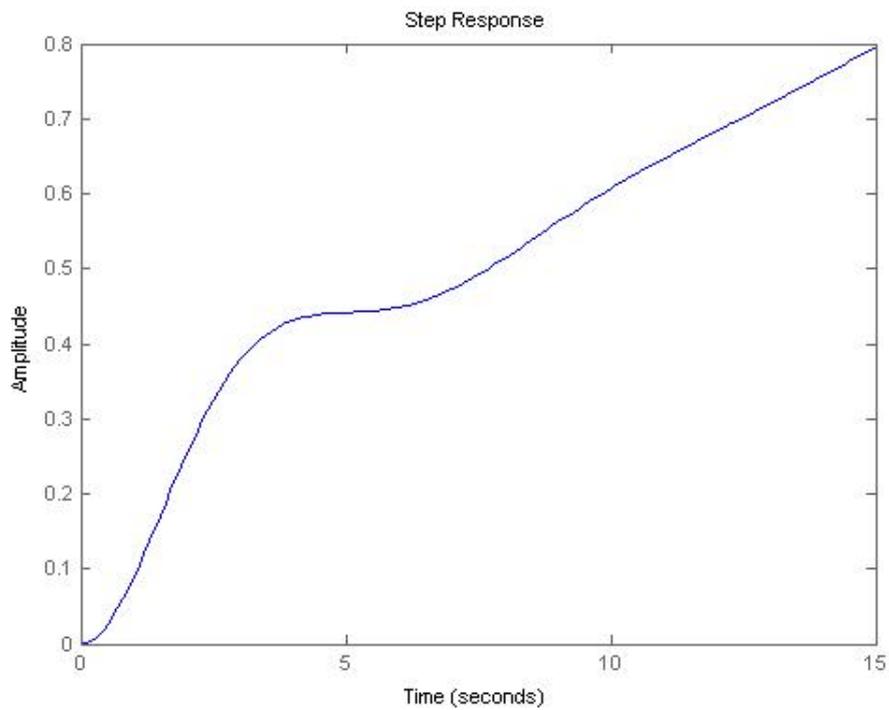


Figura 3.3: Risposta al gradino del sistema in catena aperta



Figura 3.4: Simulink. Schema sistema in catena aperta con ingresso a rampa

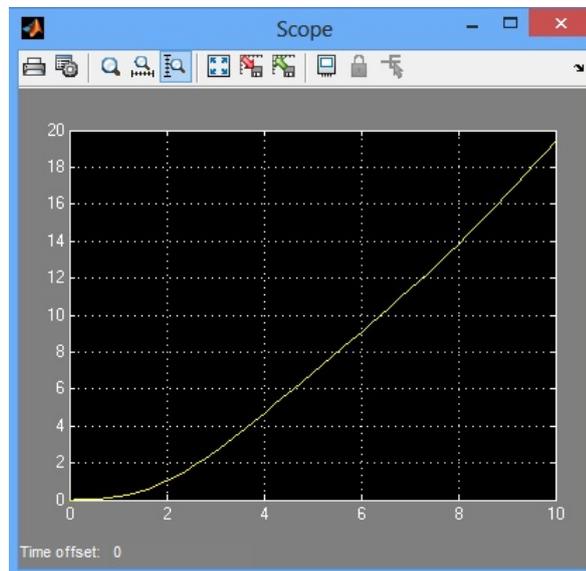


Figura 3.5: Simulink. Risposta alla rampa in catena aperta



Figura 3.6: Simulink. Schema sistema in catena aperta con ingresso sinusoidale

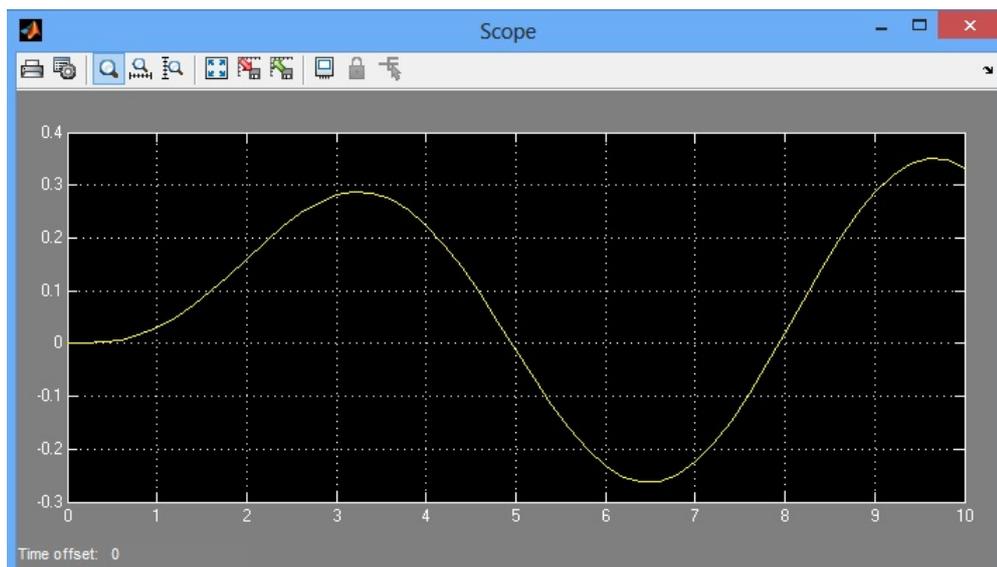


Figura 3.7: Simulink. Risposta all'ingresso sinusoidale

Capitolo 4

CONTROLLORE PID

4.1 Introduzione

Il primo metodo utilizzato per risolvere il progetto è l'introduzione del controllore PID, il quale è descritto dalla seguente funzione di trasferimento:

$$C(s) = K_P + \frac{K_I}{s} + K_D s = \frac{K_D s^2 + K_P s + K_I}{s} \quad (4.1)$$

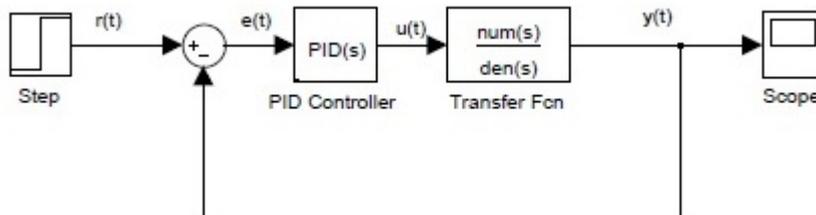


Figura 4.1: Simulink. Schema controllore PID

$r(t)$: segnale d'ingresso

$e(t)$: errore all'ingresso, differenza tra ingresso e uscita del sistema in catena chiusa

$u(t)$: segnale all'uscita del controllore

Sistema descritto in funzione di $u(t)$ nel dominio del tempo:

$$u(t) = K_P e(t) + K_I \int e(t) dt + K_D \frac{de(t)}{dt} \quad (4.2)$$

K_P : termine proporzionale

K_I : termine integrale

K_D : termine derivativo

Il controllore è possibile realizzarlo anche utilizzando solo alcuni termini, in base alla richiesta delle specifiche da rispettare. Nello svolgimento di questo progetto verranno prese in esame 4 configurazioni e verrà analizzato il loro comportamento per verificare se effettivamente sono utilizzabili per rispettare i vincoli progettuali.

I tipi di controllore utilizzati sono i seguenti:

1. Controllore di tipo P: solo componente proporzionale
2. Controllore di tipo PI: componente proporzionale e integrale
3. Controllore di tipo PD: componente proporzionale e derivativa
4. Controllore di tipo PID: tutte le componenti

Vengono ora provate le varie configurazioni, determinando graficamente quale sia la migliore ed affinando la ricerca nel campo di valori più vicino al miglior risultato.

4.2 Controllore P

Con il seguente controllore ho $C(s) = K_P$, e la funzione di trasferimento $W(s)$ è così definita, con $W_P(s)$ relativa al proporzionale:

$$W(s) = \frac{C(s)G(s)}{1 + C(s)G(s)} \quad (4.3)$$

$$W_P(s) = \frac{K_P(1.151s + 0.1774)}{K_P(1.151s + 0.1774) + s^3 + 0.739s^2 + 0.921s} = \quad (4.4)$$

$$W_P(s) = \frac{K_P(1.151s + 0.1774)}{s^3 + 0.739s^2 + (1.151K_P + 0.921)s + 0.1774K_P} \quad (4.5)$$

4. CONTROLLORE PID

Viene definito ora il codice per mettere a confronto la risposta del sistema in relazione a differenti valori di K_P

```
st=0.2;           %ampiezza gradino
num=[1.151 0.1774]; %numeratore pitch
den=[1 0.739 0.921 0]; %denominatore pitch
G=tf(num,den);   %FDT blocco G
t=0:0.1:15;      %timeframe
Kp=[0.1, 0.5, 1, 2, 5, 10, 50, 100 ];
for i=1:length(Kp)
subplot(length(Kp)/2,2,i)
Kpi=Kp(i);
W=feedback(Kpi*G,1);
step(st*W,t)
legend(['Kp= ', num2st(Kp(i))]);
end
```

Graficamente posso individuare i migliori valori di K_P , i quali sono compresi tra 1 e 5. Viene dunque approfondito lo studio in questo intervallo, in modo da ottenere il valore migliore possibile per K_P . É immediato lo scarto dei valori precedenti a tale range, in quanto il tempo di salita é decisamente al di fuori delle specifiche. Per valori superiori all'intervallo di riferimento, é possibile notare immediatamente come l'ondulazione risulti troppo elevata, oltre ad avere un valore troppo elevato di sovralongazione.

I grafici precedenti sono stati ottenuti sostituendo al codice Matlab precedente il nuovo intervallo di valori di $K_P = [1, 2, 3, 4, 5, 6]$;

Il risultato migliore é sicuramente quello ottenuto per $K_P = 2$. Il sistema é reso stabile dalla retroazione, ma dei vincoli progettuali viene rispettato solo il tempo di salita, mentre per le altre specifiche ci si avvicina ai valori desiderati senza però raggiungerli.

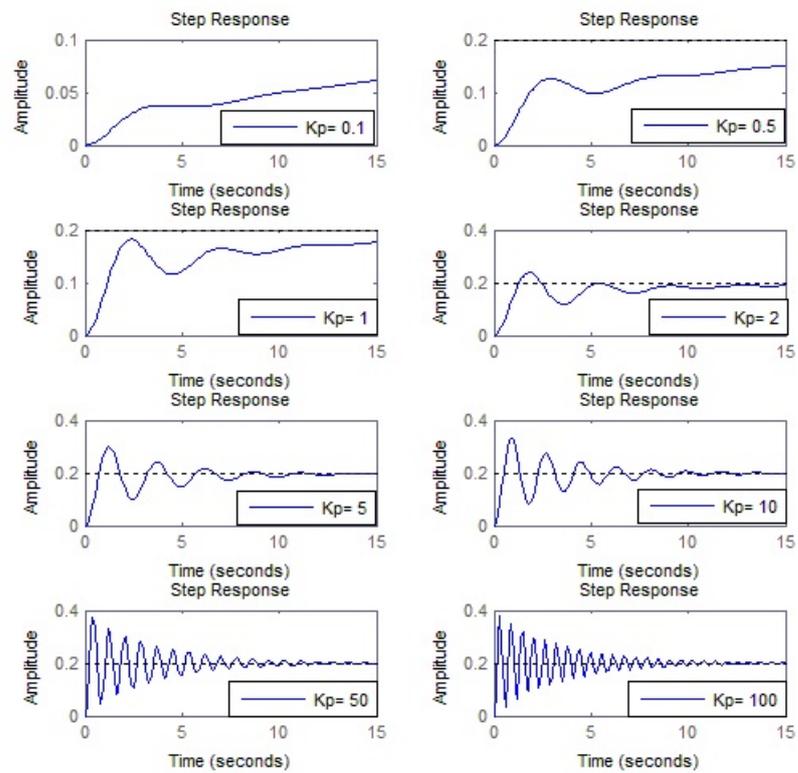


Figura 4.2: Risposta al gradino del sistema con valori di K_p compresi tra 0.1 e 100

4. CONTROLLORE PID

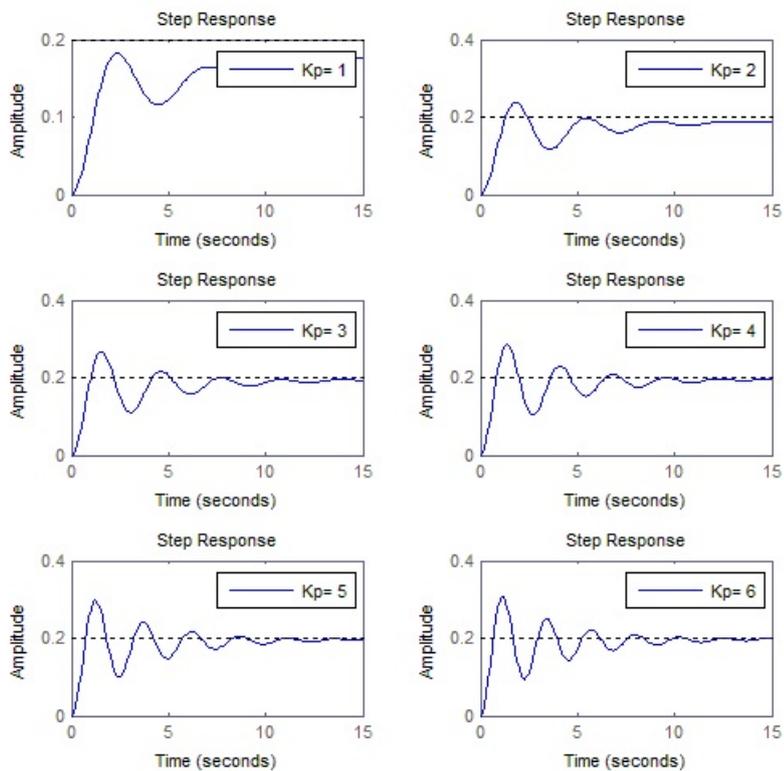


Figura 4.3: Risposta al gradino del sistema con valori di K_p compresi tra 1 e 6

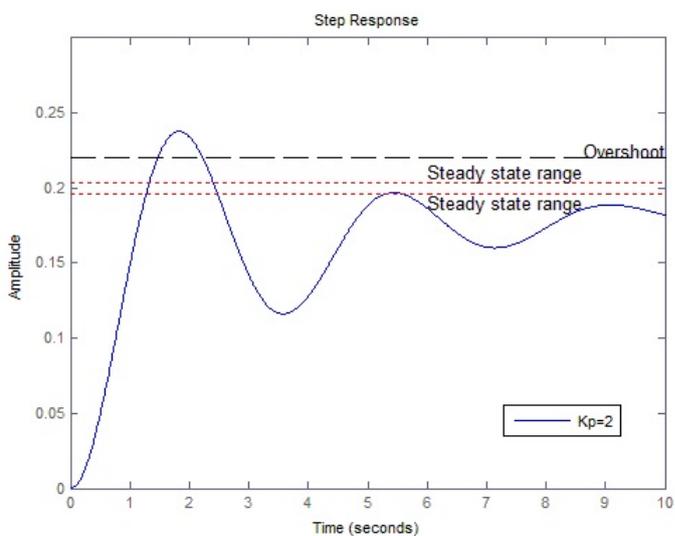


Figura 4.4: Risposta al gradino del sistema con $K_p = 2$

4.3 Controllore PI

Viene ora illustrato il comportamento del sistema in presenza del controllo PI, facendo variare i valori dei parametri K_P e K_I . $C(s) = K_P + K_I/s$. Il seguente grafico é stato ottenuto mediante Matlab, impostando $K_P = 1$ e $K_I = 0.1$. Nonostante questa sia la miglior configurazione trovata, il controllore PI non rispetta nessuna specifica. Infatti la funzione di trasferimento $G(s)$ contiene già un polo nell'origine, ed inserendo questo tipo di controllo andiamo ad aggiungerne un altro, facendo insorgere i problemi evidenziati nel grafico. Viene dunque scartata questa configurazione.

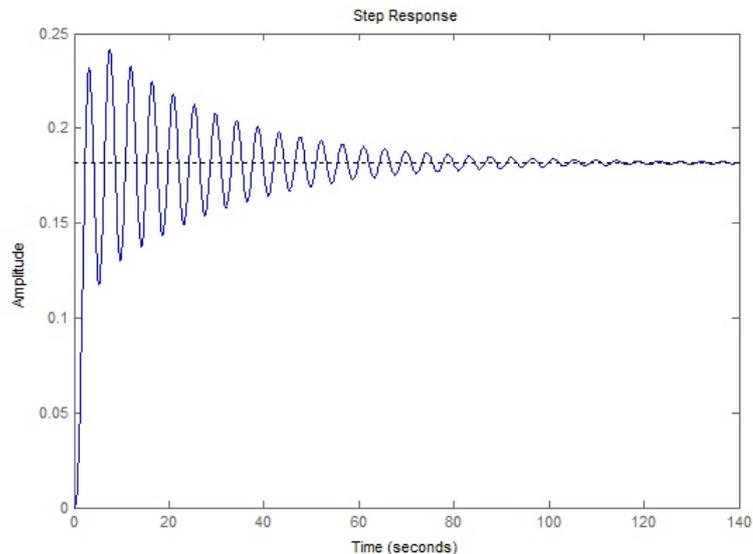


Figura 4.5: Risposta al gradino del sistema con $K_p = 1$, $K_i = 0.1$

4.4 Controllore PD

Viene di seguito presentato il controllore PD, mettendo in relazione la parte proporzionale e facendo variare il termine derivativo. Di seguito é descritto il codice Matlab:

```
st=0.2;
num =[1.151 0.1774]; % numeratore della FDT
den =[1 0.739 0.921 0]; % denominatore della FDT
```

4. CONTROLLORE PID

```
pitch=tf(num , den );
t=0:0.01:30;
KpVect=[ 0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 100 ];
KdVect=[ 0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 100 ];
for n=1:length(KpVect)
figure(n)
Kp=KpVect(n);
for m=1:length(KdVect)
Kd=KdVect(m);
contr=tf([Kd Kp],1);
sys_cl=feedback(contr*pitch,1);
step(st*sys_cl,t);
hold on
end
axis([0 20 0 0.35])
title(strcat('Kp= ',num2str(KpVect(n))));
legend('Kd=0.1','Kd=0.2','Kd=0.5','Kd=1','Kd=2','Kd=5',
'Kd=10','Kd=20','Kd=50','Kd=100');
hold on
end
hold off
```

Dai seguenti grafici possiamo subito notare che le prime figure, dove sono presenti le varie risposte per bassi valori di K_P , ci sia una difficoltà a raggiungere il valore di regime, infatti il tempo di salita è troppo elevato, indipendentemente dal valore di K_D . Inoltre possono essere subito scartati anche i valori per i quali K_D provoca delle forti ondulazioni, ovvero per bassi valori. Emerge poi che per alti valori di K_D si abbia sempre un buon andamento, con il rispetto di tutte le specifiche di progetto, ma per tali valori è messa a rischio la stabilità del sistema, poiché cambiando il riferimento da un valore ad un altro quasi istantaneamente, avendo il controllore un'azione derivativa, porterebbe l'uscita ad un valore troppo elevato. Questi valori devono dunque essere scartati a priori.

Viene ora riportato il grafico affinato per un intervallo più selettivo dei valori di K_P e di K_I , in modo da poter scegliere più semplicemente i valori corretti di tali parametri, in modo da rispettare le specifiche.

Dai seguenti grafici, un risultato soddisfacente, che merita di essere analizzato singolarmente in dettaglio, è quello per $K_D = 10$ e per $K_P = 10$.

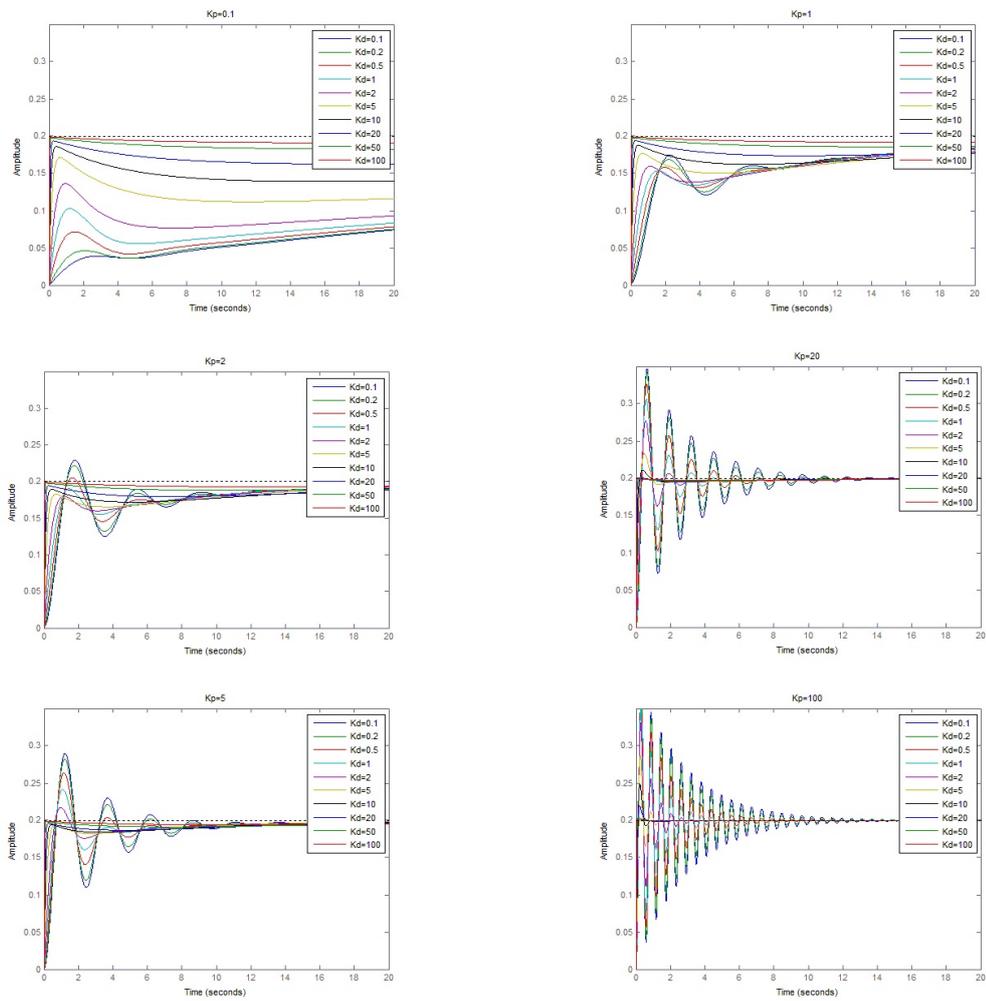


Figura 4.6: Risposta al gradino al variare di K_p e K_d

4. CONTROLLORE PID

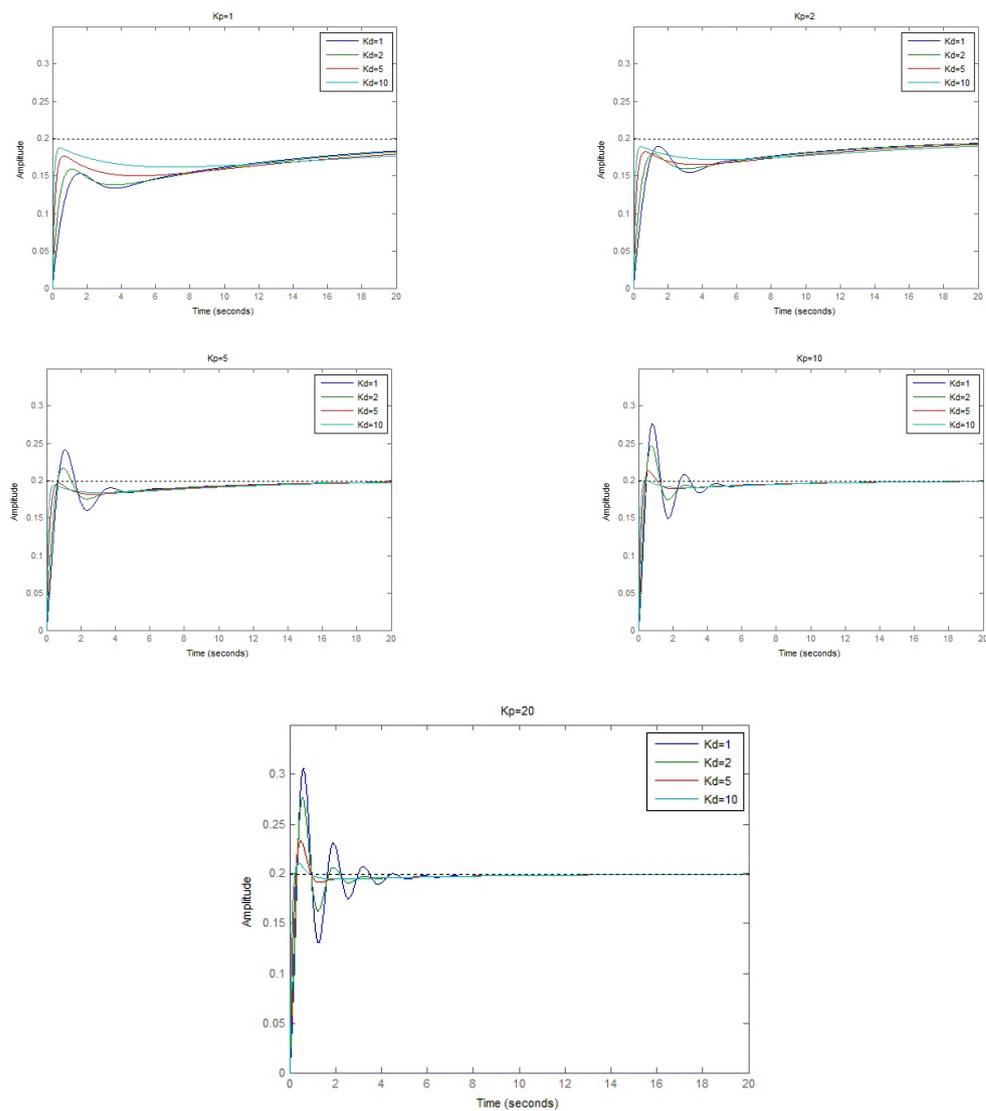


Figura 4.7: Risposta al gradino al variare di K_p e K_d per valori piú affinati

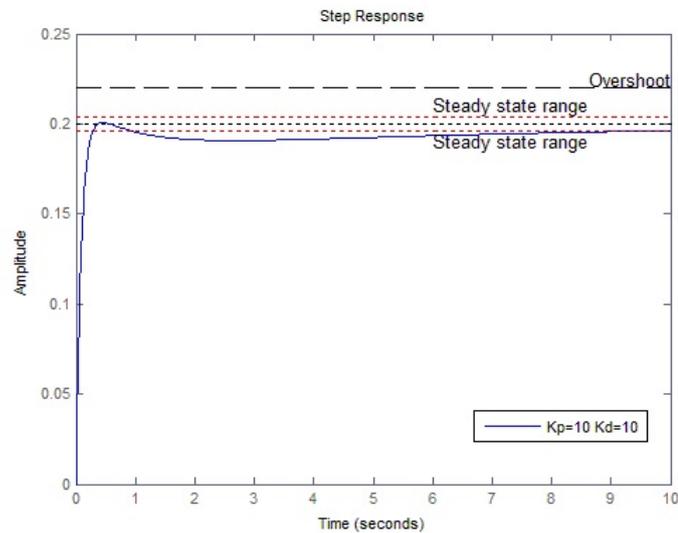


Figura 4.8: Risposta al gradino del sistema con $K_p = 10$ e $K_d = 10$

Dal grafico possiamo osservare se le specifiche vengono ora rispettate:

- La sovralongazione (Overshoot) é inferiore al 10% .
- Il tempo di salita (Rise time) é inferiore ai 2 secondi
- Il tempo di assestamento (Settling time) é minore di 10 secondi
- L'errore a regime (Steady state error) é minore del 2%

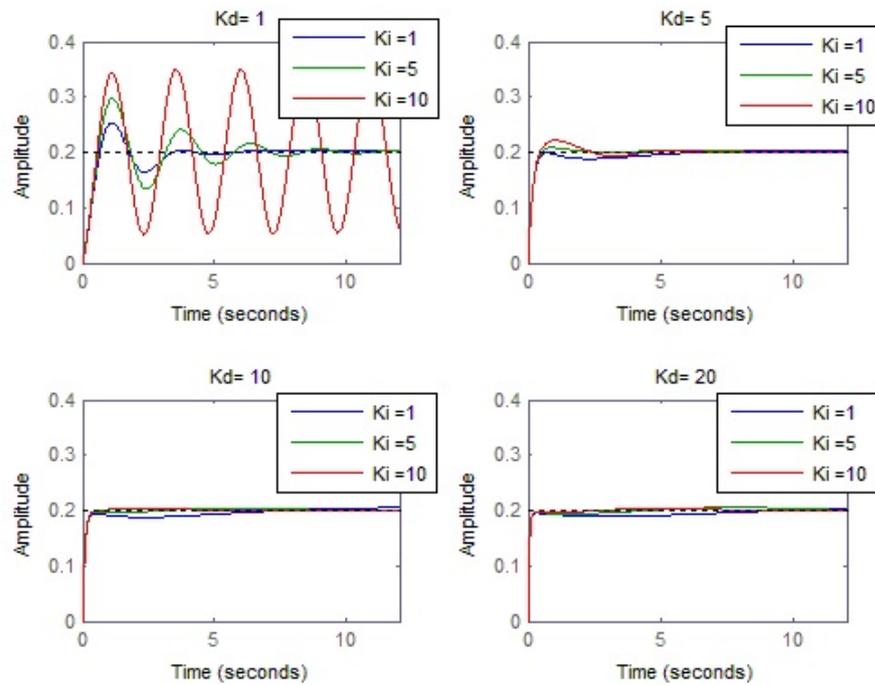
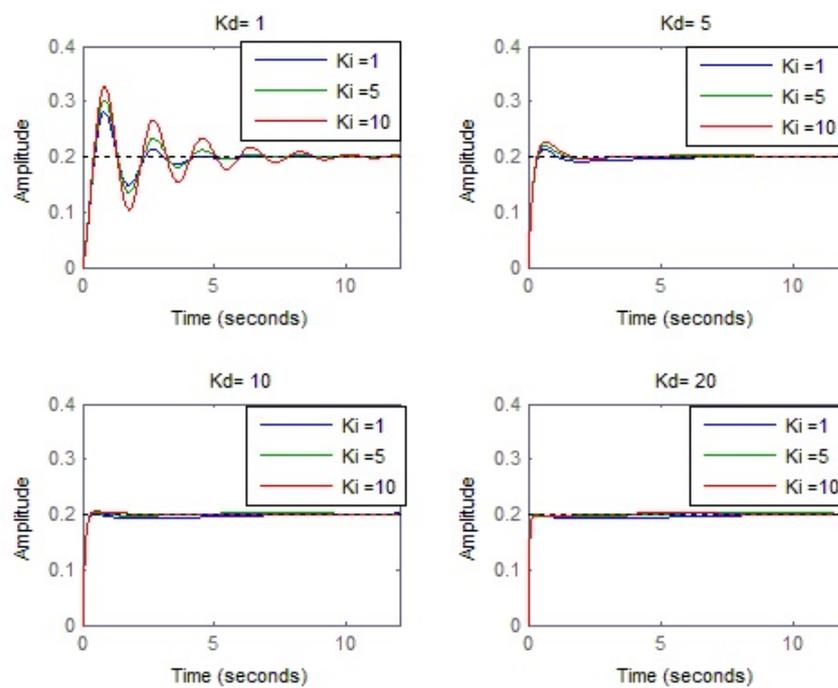
Posso dunque concludere che il semplice controllore PD, attraverso i parametri appena discussi rispetta totalmente le specifiche di progetto.

4. CONTROLLORE PID

```
RiseTime: 0.1769
SettlingTime: 8.9832
SettlingMin: 0.9007
SettlingMax: 1.0038
Overshoot: 0.3790
Undershoot: 0
Peak: 1.0038
PeakTime: 0.4496
```

4.5 PID

Come ultimo controllore analizziamo il comportamento del PID: Il blocco é così descritto: $C(s) = K_P + \frac{K_I}{s} + K_D s$. Posso sfruttare ora i risultati dei punti precedenti per determinare l'intervallo dove ricercare i valori migliori per ogni addendo del controllore. Come descritto precedentemente andrò ad escludere i valori troppo elevati di K_P , sia quelli troppo bassi. Per la parte integrale del controllore devo escludere i valori superiori al 20, poiché ciò provocherebbe forti oscillazioni al sistema, tendendo anche a far divergere l'uscita. Anche il parametro K_D va ricercato negli intervalli intermedi, in modo da ottenere tempi di salita molto brevi. Nei seguenti grafici viene presentato il comportamento dell'uscita per il valore fisso di $K_P = 5, 10, 20$, facendo variare gli altri parametri. Viene infine riportato il codice Matlab relativo alla costruzione dei grafici.

Figura 4.9: Risposta al gradino del sistema $K_p = 5$ e K_i , K_d variabiliFigura 4.10: Risposta al gradino del sistema $K_p = 10$ e K_i , K_d variabili

4. CONTROLLORE PID

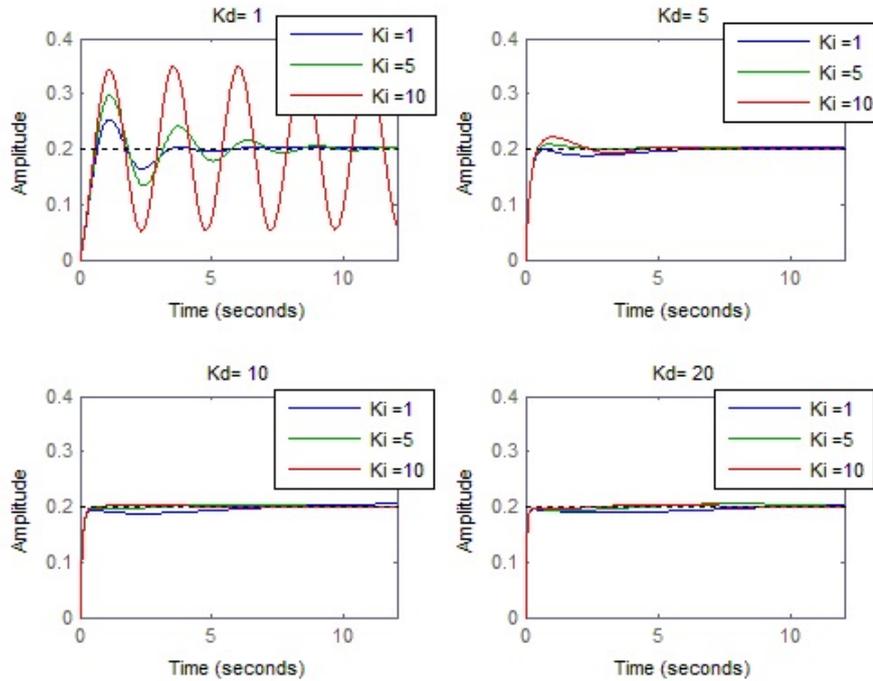


Figura 4.11: Risposta al gradino del sistema $K_p = 20$ e K_i , K_d variabili

```

num = [1.151 0.1774]; % numeratore della FDT
den = [1 0.739 0.921 0]; % denominatore della FDT
FDT = tf(num ,den); % calcolo della FDT
st = 0.2; % altezza del gradino
t = 0:0.01:12;
Kp = 5; % fattore proporzionale
Kd = [ 1, 5, 10, 20 ]; % vettore derivativo
Ki = [ 1, 5, 10 ]; % vettore integrale
for n=1:length(Kp)
figure(n)
title(['Kp= ', num2str(Kp(n))]);
Kp_n =Kp(n);
for i=1:length(Kd) % simulzione per ogni Kp nel vettore
subplot(length(Kd)/2,2,i)
Kd_i=Kd(i);
for l=1:length(Ki)
Ki_l=Ki(l);
Cs=tf([Kd_i Kp_n Ki_l ] , [1 0]);
sys_cl = feedback(Cs*FDT ,1);

```

```
step(st*sys_cl,t)
hold on
end
legend('Ki =1 ', 'Ki =5 ', 'Ki =10 ', 'location ', 'EastOutside ');
title(['Kd= ', num2str(Kd(i))]);
end
hold on
end
hold off
```

Vengono scelti ora i valori che piú sembrano soddisfare i requisiti: da un'analisi grafica i parametri scelti sono: $K_P = 10$, $K_I = 5$, $K_D = 10$:

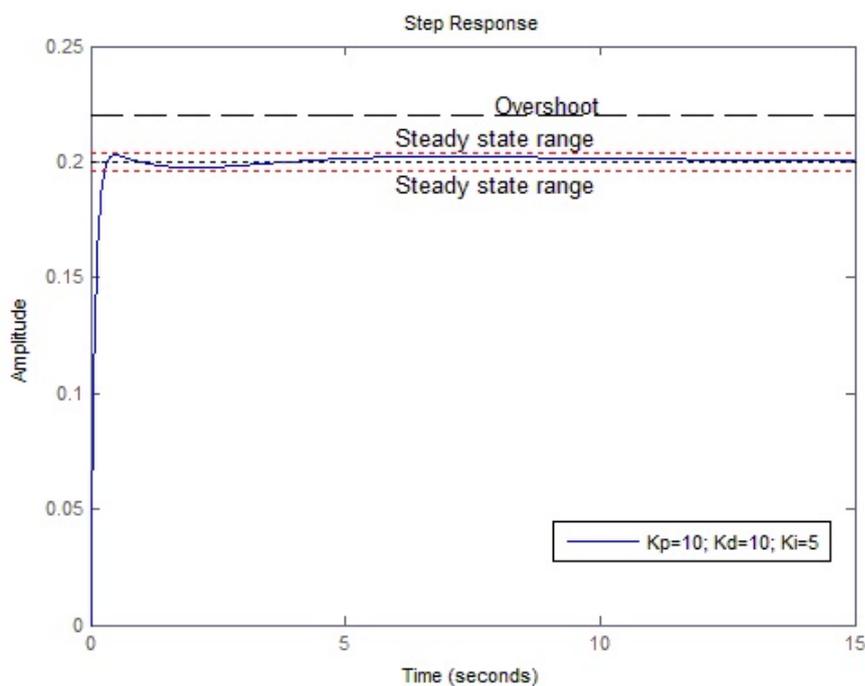


Figura 4.12: Risposta al gradino del sistema per $K_p = 10$, $K_i = 5$, $K_d = 10$

```
RiseTime: 0.1753
SettlingTime: 0.2733
SettlingMin: 0.9017
SettlingMax: 1.0145
Overshoot: 1.4495
Undershoot: 0
```

4. CONTROLLORE PID

Peak: 1.0145

PeakTime: 0.4902

Si verifica che l'andamento della risposta al gradino, soddisfa ampiamente le specifiche di progetto, avendo un tempo di salita ed un tempo di assestamento eguali. Anche la sovraelongazione é molto ridotta. Di seguito viene inoltre presentata la risposta ottenuta mediante Simulink, con la configurazione presentata inizialmente.

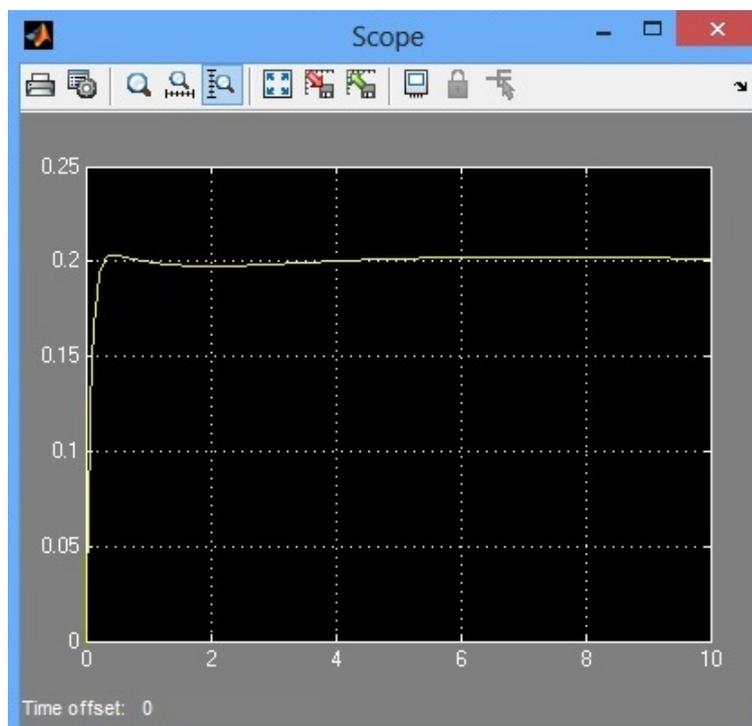


Figura 4.13: Simulink. Risposta al gradino del sistema per $K_p = 10$, $K_i = 5$, $K_d = 10$

Capitolo 5

LUOGO DELLE RADICI

5.1 Introduzione

Viene svolto ora il progetto utilizzando il metodo del controllore mediante il luogo delle radici. Questo procedimento permette di identificare con precisione la posizione dei poli della funzione di trasferimento in catena chiusa, in base alla conoscenza esatta dei polinomi della funzione di trasferimento in catena aperta.

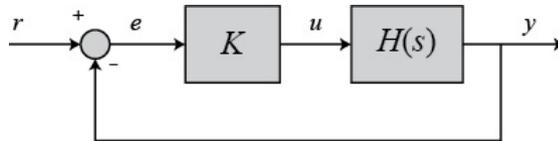


Figura 5.1: Schema a blocchi per l'analisi attraverso il luogo delle radici

La funzione di trasferimento del sistema viene descritta come segue:

$$F.d.T.(s) = \frac{Y(s)}{R(s)} = \frac{KH(s)}{1 + KH(s)} \quad (5.1)$$

con evidenziato il numeratore ed il denominatore di $H(s)$:

$$H(s) = \frac{a(s)}{b(s)} \quad (5.2)$$

la funzione di trasferimento diventa dunque:

$$\frac{Y(s)}{R(s)} = \frac{Ka(s)}{b(s) + Ka(s)} \quad (5.3)$$

Dunque i poli del sistema sono soluzione dell'equazione:

$$b(s) + Ka(s) = 0 \quad (5.4)$$

5. LUOGO DELLE RADICI

Di seguito sono riportate inoltre alcune considerazioni sul luogo delle radici: Considerando la funzione di trasferimento del sistema, la presenza di poli nel semipiano destro implica l'instabilità del sistema. La vicinanza di un polo all'asse immaginario, ha una grossa influenza nella risposta del sistema in catena chiusa. Infine i punti di attraversamento dell'asse immaginario del luogo sono definiti come punti critici, infatti il sistema varia le sue condizioni di stabilità. Vengono di seguito dunque calcolati i parametri per individuare nel luogo delle radici, le zone nelle quali vengono rispettate le specifiche. Questi parametri sono la pulsazione naturale ed il coefficiente di smorzamento.

$$\omega_n \geq \frac{1.8}{T_r} \quad (5.5)$$

$$\xi \geq \sqrt{\frac{(\ln \frac{M_p}{\pi})^2}{1 + (\ln \frac{M_p}{\pi})^2}} \quad (5.6)$$

Con ω_n = Pulsazione naturale, T_r = Tempo di salita, ξ = Coeff. di smorzamento, M_p = Valore massimo di sovranelongazione.

I valori sono stati calcolati precedentemente e vengono ora riportati:

$$\omega_n \geq 0.9; \xi \geq 0.936$$

5.2 Luogo delle radici e progetto rete correttiva

Viene di seguito riportato il codice Matlab per lo sviluppo del luogo delle radici, imposto con i parametri appena calcolati.

```
Ts=10;
Tr=2;
Oversh=0.1;
st=0.2;
Wn=0.9;
Mp=st+st*Oversh; %Max overshoot
z=0.936; %Damping ratio
num=[1.151 0.1774];
den=[1 0.739 0.921 0];
plant=tf(num,den);
rlocus (plant);
```

```
sgrid (z,Wn);
axis ([-1.5 0 -3 3])
```

Il grafico generato dalla funzione Matlab risulta il seguente:

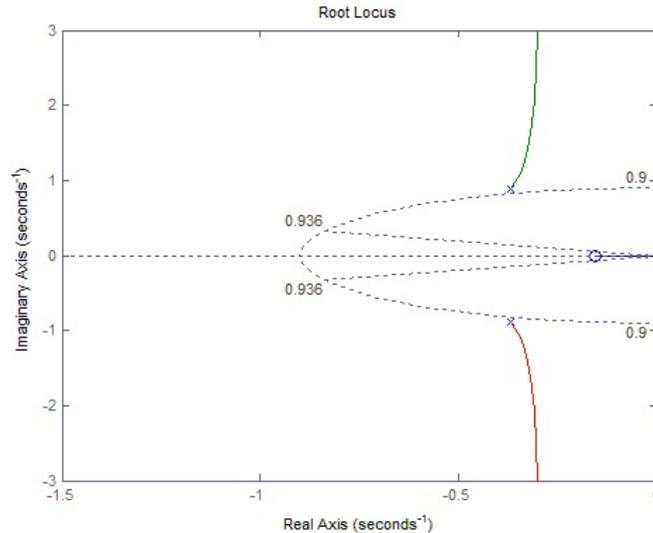


Figura 5.2: Luogo delle radici

Le zone del grafico che soddisfano i requisiti richiesti sono individuate dalle linee tratteggiate presenti nel grafico. Il semicerchio rappresenta l'insieme dei punti in cui la pulsazione naturale risulta uguale a 0.9. All'interno di esso la pulsazione naturale diminuisce, mentre aumenta nei punti che risiedono all'esterno. Le 2 linee rette che intersecano il grafico sono formate dai punti in cui il coefficiente di smorzamento risulta pari a 0.939, come precedentemente stabilito. Lo smorzamento risulta crescente all'interno dell'area delle 2 rette mentre diminuisce all'esterno. Un piccolo angolo formato da queste 2 rette, corrisponde ad una richiesta di piccola sovralongazione. Per rispettare i vincoli, imponiamo che la zona di nostro interesse sia quella compresa tra le 2 rette e quella al di fuori del semicerchio.

Noto dal grafico che nessun punto di interesse è compreso entro queste zone. Riporto di seguito il grafico della risposta al gradino del sistema, la quale infatti non rispetta i vincoli di progetto.

Dunque devo utilizzare una rete anticipatrice, in modo da spostare il grafico verso sinistra e poter far rientrare i vincoli nella zona desiderata.

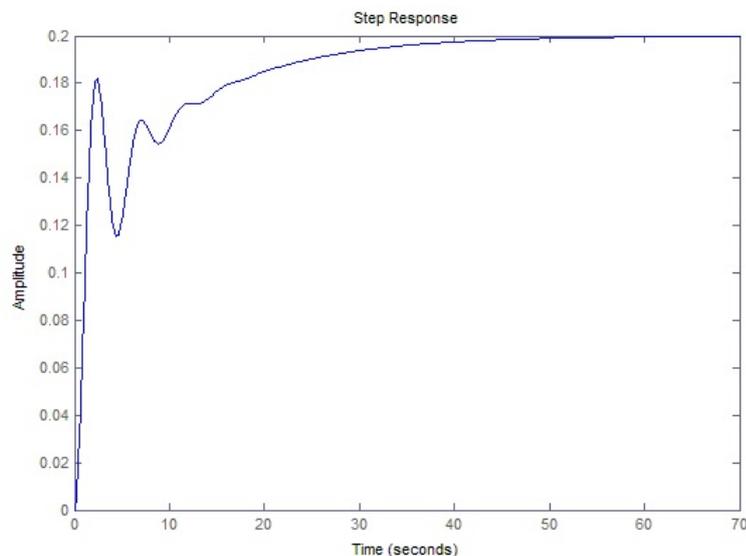


Figura 5.3: Risposta al gradino

Costruisco ora la rete anticipatrice attraverso SISOTOOL, la quale viene definita nel seguente modo:

$$H(s) = K_C \frac{(s - z_0)}{(s - p_0)} \quad (5.7)$$

Devo dunque trovare i valori del polo e dello zero, per fare in modo di rientrare nell'area desiderata. Con SISOTOOL é sufficiente posizionare iterativamente il polo e lo zero e visualizzare la risposta del sistema al gradino per la configurazione appena effettuata e visualizzare cosí il nuovo luogo delle radici ottenuto.

Nel seguente grafico viene visualizzato il luogo della radici, con i vincoli di progetto da rispettare:

- impostazione settling time < 10 secondi: nel disegno viene impostato il vincolo della retta verticale
- rise time < 2 secondi, corrisponde ad una ω_n pari a 0.9 rad/sec, nel disegno corrisponde al semicerchio
- overshoot < 10% implica le 2 rette che partono dall'origine

Vengono ora definiti i valori del polo e dello zero del compensatore della rete anticipatrice, ed il parametro del guadagno K . É possibile muovere direttamente sul luogo delle radici i valori di z_0 e p_0 , vedendo in tempo reale il grafico della risposta al gradino del sistema. Provando varie configurazioni iterativamente, un'ottima soluzione é derivata dalla scelta di $z_0 = -0.9$, $p_0 = -30$, $K_C = 200$.

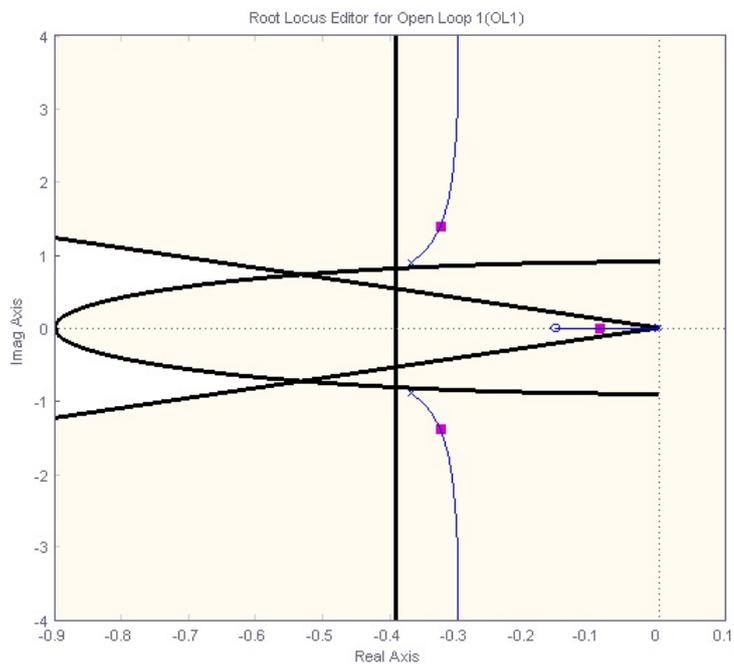


Figura 5.4: Luogo delle radici con SISO

Di seguito viene inoltre presentata la risposta che si otterrebbe per un guadagno K_C ancora maggiore. Notiamo come la risposta risulti peggiorata, specialmente nel parametro riguardante l'overshoot.

5. LUOGO DELLE RADICI

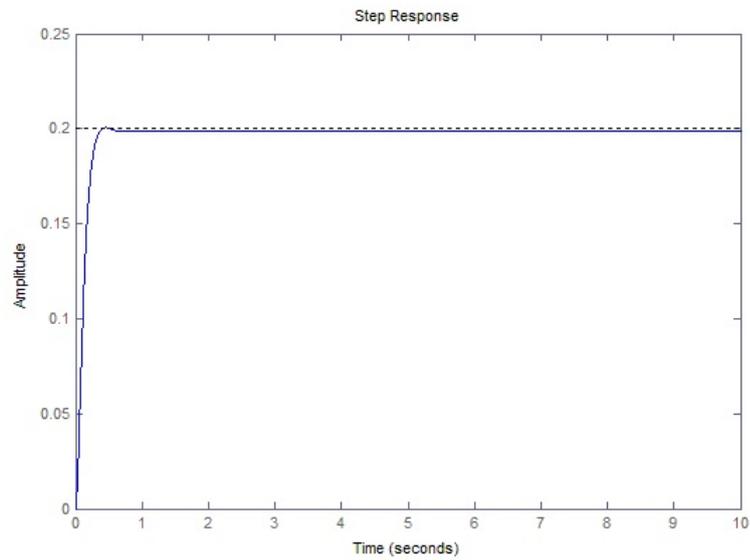


Figura 5.5: Risposta al gradino del sistema con rete anticipatrice con $K_c=200$

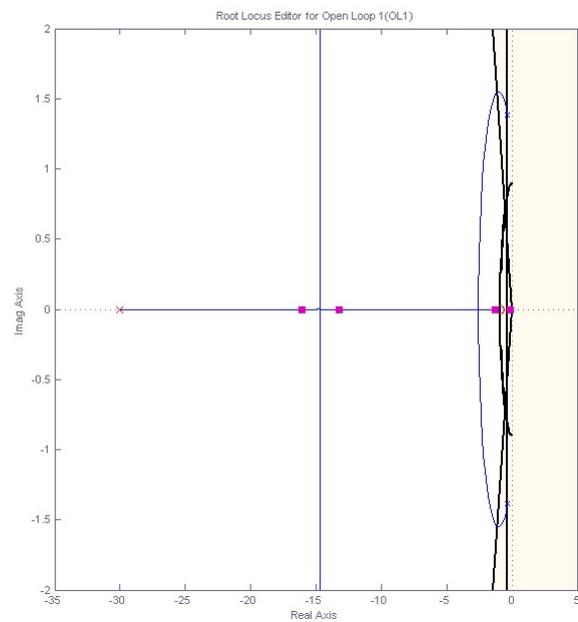


Figura 5.6: Luogo delle radici finale

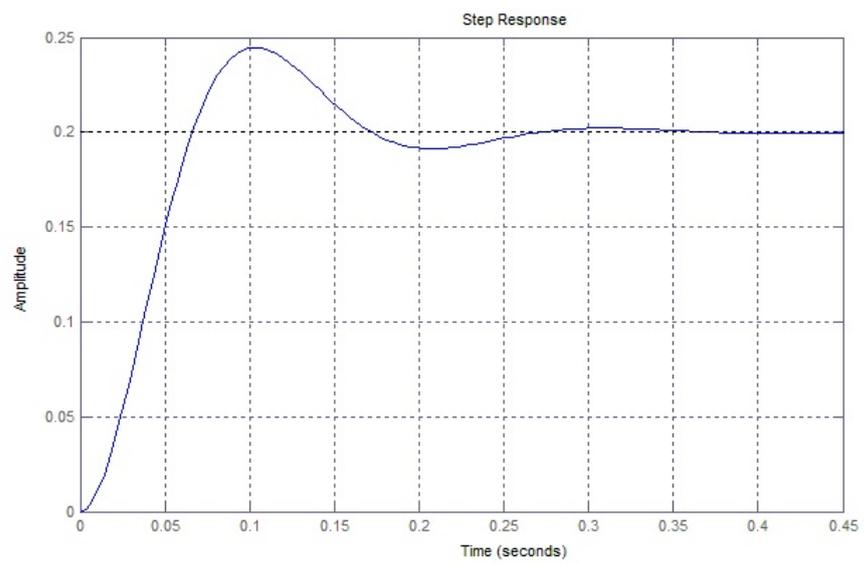


Figura 5.7: Risposta del sistema con $K_c=1000$

Capitolo 6

STUDIO IN FREQUENZA

6.1 Introduzione

Viene ora sviluppato il progetto attraverso lo studio nel dominio della frequenza, attraverso i diagrammi di Bode e Nyquist. I parametri fondamentali per lo studio in frequenza, sono la pulsazione naturale, già calcolata nel capitolo precedente, ed il margine di fase m_ϕ . Nel diagramma di Bode, equivale alla differenza, misurata nel punto di attraversamento, tra la curva di fase e la retta orizzontale corrispondente al valore di -180° .

6.2 Analisi del sistema

Viene di seguito riportata la risposta al gradino del sistema in catena aperta, sviluppata con Matlab:

```
st = 0.2;
t=0:0.1:15;
num = [1.151 0.1774];
den = [1 0.739 0.921 0];
pitch = tf(num,den);
figure(Bodeopen);
subplot(1,2,1);
step(st*pitch,t);
subplot(1,2,2);
bode(pitch);
```

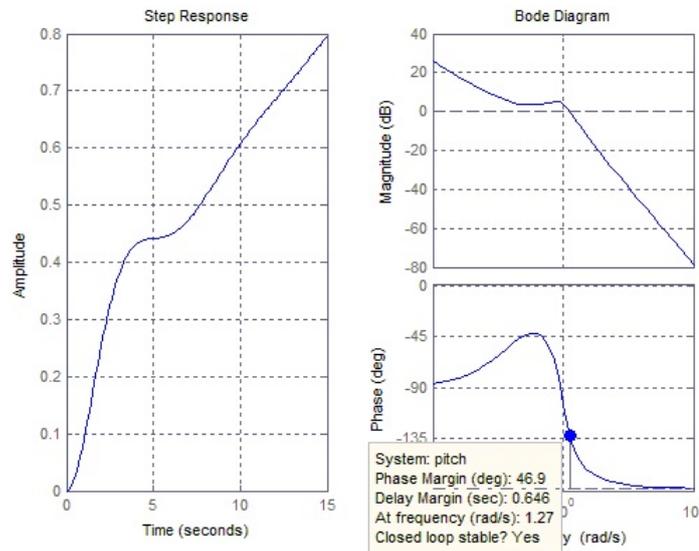


Figura 6.1: Risposta in catena aperta e diagramma di Bode

Il sistema é ovviamente instabile, come precedentemente constatato. Ripeto quindi l'operazione per il sistema in catena chiusa.

La risposta retroazionata, come ci si attendeva, ha reso il sistema stabile, seppur non rispettando i vincoli progettuali. Il margine di fase ottenuto é incrementato e vale 61.3° , mentre l'attraversamento é presente a 1.59 rad/sec. Entrambi i parametri sono positivi, indicando la stabilitá del sistema in catena chiusa.

Viene dunque introdotta una rete corretttrice, la quale incrementará il guadagno alla frequenza di attraversamento, in modo tale da aumentare la velocità della risposta. Incrementará inoltre il margine di fase, il quale é inversamente legato all'overshoot del sistema in catena chiusa.

6.3 Compensatore

Viene dunque progettata la rete corretttrice anticipatrice, la quale produce un incremento di fase del sistema e dello smorzamento. Generalmente viene inoltre incrementato il guadagno a catena aperta alle alte frequenze provocando un incremento della frequenza di attraversamento e della velocità del sistema. Viene inoltre diminuito il tempo di assestamento. La forma della rete corretttrice é la seguente:

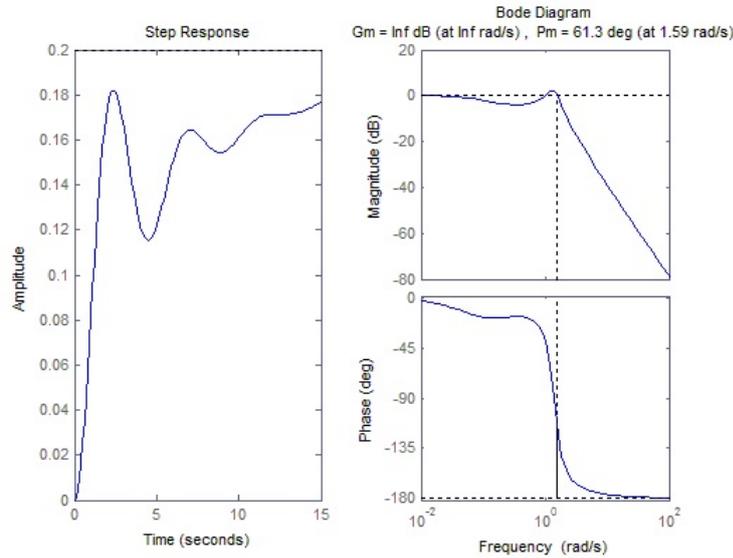


Figura 6.2: Risposta in catena chiusa e diagramma di Bode

$$C(s) = K_C \frac{Ts + 1}{\alpha Ts + 1} \quad (6.1)$$

I parametri da ricercare ora sono α , T e K .

K é impostato per soddisfare i requisiti dell'errore a regime, ma poiché é presente un integratore nella nostra funzione di sistema $G(s)$, l'errore sará 0 per ogni valore di K_C . Posso ricavare ora i parametri per progettare la rete correttiva partendo dalla funzione $G(s)$.

$$G(s) = \frac{1.151s + 0.1774}{s(s^2 + 0.739s + 0.921)} = \quad (6.2)$$

$$= \frac{0.1774(1 + \frac{1.151s}{0.1774})}{0.921s(1 + \frac{0.739s}{0.921} + \frac{s^2}{0.921})} = \frac{K_G}{s^h} \tilde{G}(s) \quad (6.3)$$

Posso dunque sostituire il blocco contenente $C(s)$ e quello della funzione $G(s)$, posti in successione, con un unico blocco, composto dal prodotto dei 2 blocchi, dato che ci troviamo nel dominio della frequenza.

Procedo alla determinazione dei parametri del controllore. Ho che il guadagno K_G é pari a 0.193 e il valore di h é 1. É presente inoltre un polo nell'origine nella funzione $G(s)$, dunque come menzionato precedentemente, l'uscita avrá sempre

un errore a regime pari a 0. Dunque il valore di K_C può essere impostato ad un valore base di 10. Viene ora tracciato il diagramma di Bode attraverso la seguente funzione Matlab:

```
K = 10;
margin(K*P_pitch), grid
figure;
sys_cl = feedback(K*P_pitch,1);
step(0.2*sys_cl), grid
title('Risposta con K = 10')
```

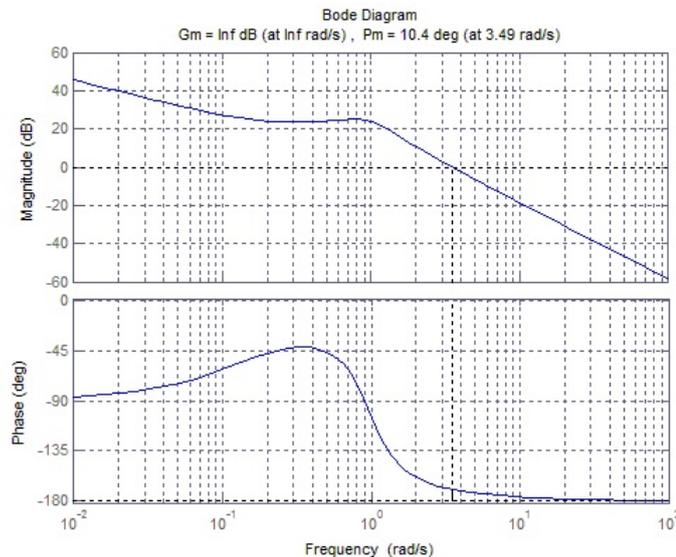
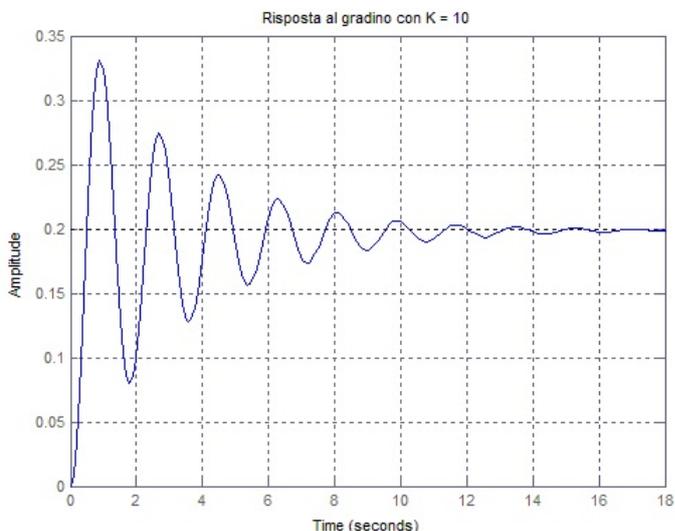


Figura 6.3: Risposta in catena chiusa e diagramma di Bode con $K_C=10$

Come possiamo vedere dal diagramma, il margine di fase è diminuito, dunque la risposta al gradino è peggiorata introducendo solo il parametro K_C al controllore.

Vengono calcolati ora i parametri zero-polo, della rete anticipatrice, in modo da aumentare il margine di fase, aumentando lo smorzamento e diminuendo dunque l'overshoot del sistema. Iniziamo col calcolare α , ovvero il rapporto tra il valore dello zero e del polo. La seguente equazione identifica il massimo margine di fase che è possibile aggiungere con una rete correttiva.

$$\sin(\phi_m) = \frac{1 - \alpha}{1 + \alpha} \quad (6.4)$$

Figura 6.4: Risposta in catena chiusa con $K_c=10$

Per il fattore di smorzamento possiamo utilizzare la seguente equazione:

$$\xi \simeq \frac{P.M.}{100^\circ} \quad (6.5)$$

Dalle specifiche, dobbiamo ottenere un overshoot minore del 10% dobbiamo imporre un ξ maggiore di 0.59, e un margine di fase maggiore di 59° . Dal diagramma di Bode osservo che il margine di fase attuale é di circa 10.4° , dunque per soddisfare la richiesta devo aggiungere 50° . Viene poi arrotondato per eccesso a 60 in modo da compensare la modifica della frequenza di attraversamento. Posso dunque scrivere l'equazione:

$$\alpha = \frac{1 - \sin(60^\circ)}{1 + \sin(60^\circ)} \simeq 0.09 \quad (6.6)$$

Per questo valore di α posso determinare l'incremento di ampiezza fornito dal compensatore.

$$20\log\left(\frac{1}{\sqrt{\alpha}}\right) \simeq 20\log\left(\frac{1}{\sqrt{0.09}}\right) \simeq 10dB \quad (6.7)$$

L'aggiunta del compensatore aumenta la ω di attraversamento da 3,49 rad/sec a 6.1 rad/sec. Posso ora calcolare T in funzione dell'ottimizzazione dell'incremento del margine di fase.

$$\omega_m = \frac{1}{T\sqrt{\alpha}}; T = \frac{1}{6.1\sqrt{0.09}} \simeq 0.52 \quad (6.8)$$

Ottenuti i valori richiesti inizialmente, posso dunque inserirli nella funzione di trasferimento del compensatore e disegnare il diagramma di Bode.

```
K = 10;
alpha = 0.09;
T = 0.52;
C_lead = K*(T*s + 1) / (alpha*T*s + 1);
margin(C_lead*P_pitch), grid
```

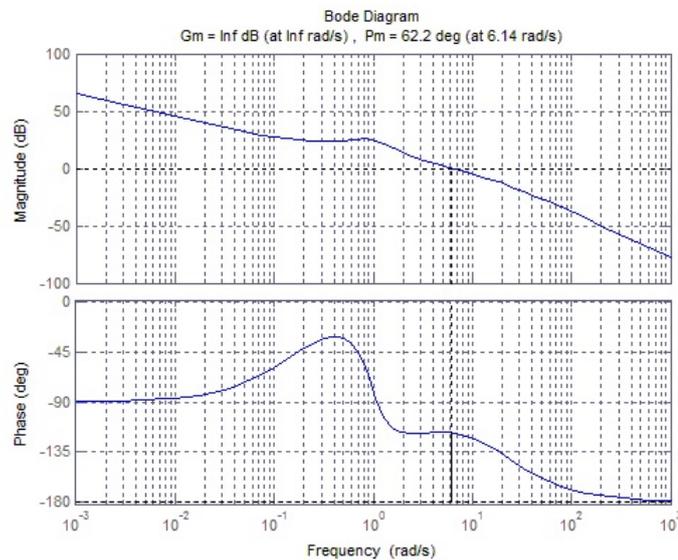


Figura 6.5: Diagramma di Bode in presenza del compensatore

Come evidenziato dal diagramma, il margine di fase vale 62.2° ed è inoltre incrementato il guadagno alla frequenza di crossover a 6.14 rad/sec. Di seguito è riportata infine la risposta al gradino del sistema.

Da questo diagramma posso osservare che tutte le specifiche di progetto sono rispettate, eccetto che per l'errore di sovralongazione, il quale supera di poco il valore richiesto di 10%. Provo dunque una nuova configurazione di parametri, decrementando il valore di α e aumentando il valore di T. Dopo varie prove, le configurazioni migliori sono le seguenti, espresse in figura 6.7.

Per questa ultima configurazione possiamo notare che l'errore di sovralongazione è diminuito, e per correttezza esamino tutti i valori inerenti il grafico.

ans =

6. STUDIO IN FREQUENZA

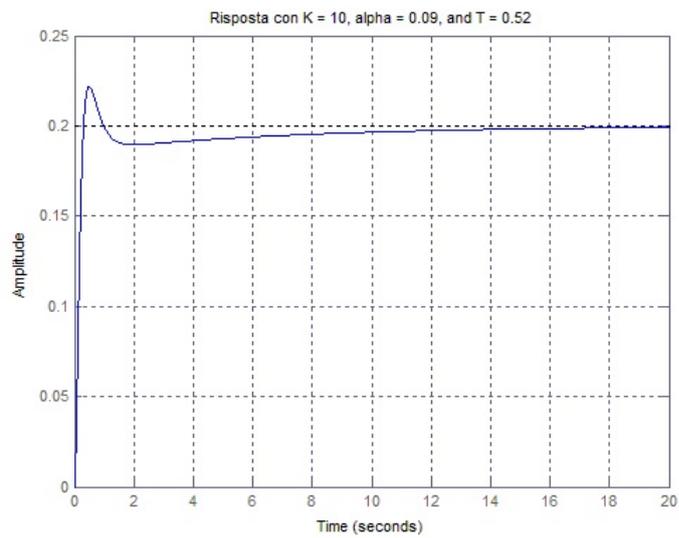


Figura 6.6: Risposta al gradino con $K=10$, $\alpha = 0.09$, $T=0.52$

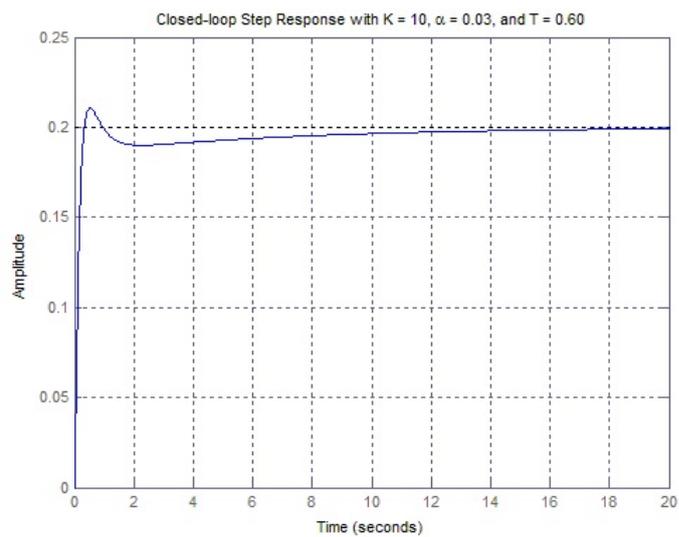


Figura 6.7: Risposta al gradino con $K=10$, $\alpha = 0.03$, $T=0.60$

```

RiseTime: 0.2202
SettlingTime: 9.0427
SettlingMin: 0.1828
SettlingMax: 0.2137
Overshoot: 6.8495
Undershoot: 0
Peak: 0.2137
PeakTime: 0.5344

```

Dunque la forma del compensatore sarà la seguente:

$$C(s) = 10 \frac{0.6s + 1}{0.03 \cdot 0.6s + 1} = 10 \frac{0.6s + 1}{0.018s + 1} \quad (6.9)$$

6.4 Analisi con il metodo di Nyquist

Viene ora descritto il diagramma di Nyquist. I parametri per il suo disegno sono calcolati direttamente da Bode. Grazie a questo metodo posso dedurre la BIBO stabilità del sistema a catena chiusa a partire dalla funzione di trasferimento del sistema. Inizialmente viene disegnato il grafico, tramite Matlab, senza l'ausilio del compensatore.

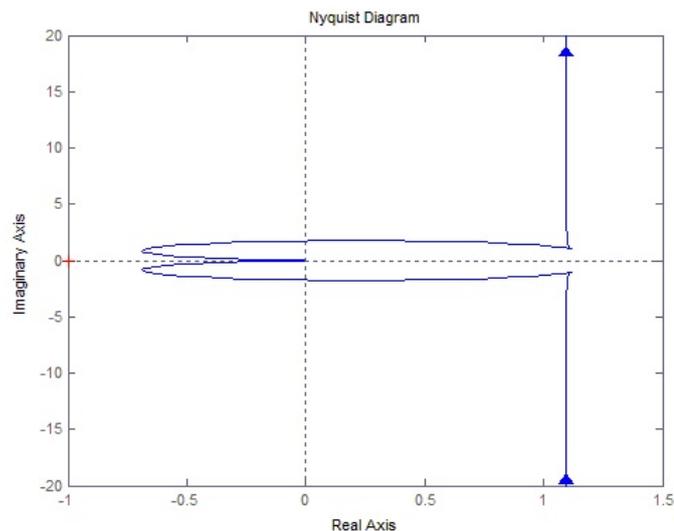


Figura 6.8: Diagramma di Nyquist

Per determinare la stabilità osservo che la curva, partendo da $\omega \implies -\infty$ e giungendo a $+\infty$, non fa alcun giro attorno al punto $-1+j0$. Avendo infatti tutti i

6. STUDIO IN FREQUENZA

poli della funzione a parte reale negativa, il numero N di giri coincide con il numero NP di poli a parte positiva. Posso dunque affermare che il sistema in catena chiusa é stabile, come infatti stabilito dalle trattazioni precedenti. Dal grafico posso poi notare la vicinanza del passaggio della curva vicino al punto critico $-1 + j0$, la quale implica vari problemi relativi alle varie specifiche da rispettare, quali la sovravelongazione e l'oscillazione. Viene successivamente presentato il diagramma di Nyquist con l'applicazione della rete correttiva vista per Bode. Il sistema rimane stabile, come ipotizzato, e possiamo subito notare come il passaggio vicino al punto critico si sia allontanato, garantendo cosí una migliore risposta alla sovravelongazione e alle oscillazioni.

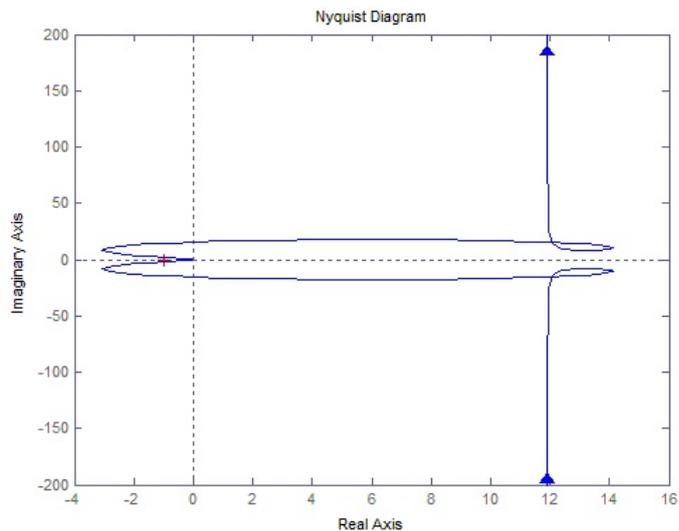


Figura 6.9: Diagramma di Nyquist con l'introduzione della rete correttiva

Capitolo 7

CONCLUSIONI

7.1 Confronti

Vengono di seguito messi a confronto i vari risultati per i vari tipi di controllori utilizzati, mettendo in evidenza le differenti caratteristiche di progetto e di risultato ottenuto. Tutti i metodi messi a confronto rispettano interamente le specifiche poste. Il controllore PID é stato sviluppato dapprima introducendo la parte proporzionale P la quale ha permesso di ottenere la stabilit  del sistema. Successivamente é stata introdotta la parte derivativa, ed attraverso varie scelte di parametri, iterativamente si é ristretto l'intervallo di ricerca, che portava ad un raggiungimento dei risultati desiderati. Infine é stata introdotta anche la parte integrativa, in modo da ottenere un PID completo. Quest'ultima versione ha migliorato i risultati ottenuti rispetto alla sola parte PD, come possiamo notare dall'andamento del nuovo grafico.

L'impiego del PID é comunemente molto utilizzato, per la sua semplicit  e per la sua buona risposta. Per una ulteriore verifica é stato utilizzato Simulink: é stata disegnata la rete impostando i vari parametri del sistema ed ottenuto in uscita la risposta corrispondente.

Successivamente abbiamo proceduto con il secondo metodo, il luogo delle radici. Sono stati derivati i parametri di interesse quali il coefficiente di smorzamento e la pulsazione naturale. É poi stato tracciato il luogo con Matlab e sono stati posti i vincoli di sovralongazione, tempo di salita e di assestamento, dando forma alle varie aree che definivano il rispetto di tali parametri. É stato poi fatto uso di una rete correttiva anticipatrice per fare in modo di spostare il grafico verso

sinistra e rientrare così nell'area di rispetto dei vincoli di progetto. È stato utilizzato inoltre un altro strumento grafico quale Sisotool. Con questa applicazione è possibile spostare graficamente i vari parametri ed ottenere in tempo reale la risposta in modo da poter identificare la soluzione migliore. L'ultimo metodo utilizzato è l'analisi in frequenza attraverso i diagrammi di Bode con l'introduzione delle reti correttrici. Il parametro principale per la stabilità del sistema è ora il margine di fase, identificato direttamente dalla relazione tra il diagramma di ampiezza e fase di Bode. Viene poi progettata la funzione di trasferimento della rete corretttrice, trovando il guadagno K , il polo e lo zero. La soluzione ottenuta inizialmente non è la migliore, in quanto i risultati derivano da delle equazioni approssimative imposte. Affinando tali valori però è possibile ottenere una buona risposta che rispetta interamente tutti i vincoli progettuali.

7.2 Risultati

Valori ottenuti mediante i vari metodi di analisi:

- Controllore PD:

$$C(s) = K_P + K_D s = 10 + 10s \quad (7.1)$$

- Controllore PID:

$$C(s) = K_P + K_I/s + K : D = 10 + 5/s + 10s \quad (7.2)$$

- Controllore mediante luogo delle radici:

$$C(s) = K_C \frac{s - z_0}{s - p_0} = 200 \frac{s + 0.9}{s + 30} \quad (7.3)$$

- Risposta in frequenza:

$$C(s) = 10 \frac{0.6s + 1}{0.018s + 1} \quad (7.4)$$

Dalla tabella possiamo mettere in relazione i vari parametri tra i controllori. Nel PID notiamo come il tempo di assestamento sia ridotto notevolmente rispetto al solo PD. Il luogo delle radici fornisce un'ottima risposta, anche se il tempo di salita risulta leggermente maggiore rispetto agli altri. In compenso l'overshoot è assente, ed il tempo di assestamento coincide con quello di salita. L'impiego della rete corretttrice con l'analisi in frequenza dá un prolungato tempo di assestamento,

Controllore	T. salita (<2 sec.)	T. assestamento (<10 sec.)	Sovraelongazione (<10%)
PD	0,177s	8,965s	0,38%
PID	0,175s	0,273s	1,45%
Luogo delle radici	0,334s	0,332s	0,00%
Rete correttiva	0,220s	9,043s	6,85%

Figura 7.1: Relazione tra i vari controllori

appena entro le specifiche, ed un valore di sovraelongazione maggiore rispetto a tutti gli altri controllori. Il tempo di salita resta comunque buono. Per quanto riguarda l'errore a regime, tutti i controllori rispettano la specifica.

Bibliografia

Riferimenti bibliografici:

- <http://ctms.engin.umich.edu//CTMS>, Control tutorial for Matlab and Simulink
- <http://en.wikipedia.org/wiki/PIDcontroller>, PID Controller
- Controllo a retroazione di sistemi dinamici, Franklin, Powell, Prentice-Hall, 2002
- <http://www.grc.nasa.gov/WWW/k-12/airplane/pitch.html>