

UNIVERSITÀ DEGLI STUDI DI PADOVA  
FACOLTÀ DI INGEGNERIA  
Corso di Laurea in Ingegneria dell'Informazione

**Studio in ambiente Matlab/Simulink  
del sistema di controllo  
del beccheggio di un aereo**

Laureando: Giacomo Baruzzo

Relatore: prof. Alessandro Beghi

Anno accademico 2010 - 2011  
Padova, 27 settembre 2011

## Indice

<b>1</b>	<b>INTRODUZIONE</b>	<b>1</b>
<b>2</b>	<b>ANALISI DEL PROBLEMA</b>	<b>2</b>
2.1	DESCRIZIONE DEL PROBLEMA . . . . .	2
2.2	SPECIFICHE DI CONTROLLO . . . . .	3
2.3	DERIVAZIONE DEL MODELLO MATEMATICO . . . . .	4
<b>3</b>	<b>ANALISI DEL SISTEMA IN CATENA APERTA</b>	<b>6</b>
3.1	RISPOSTA AL GRADINO E STABILITA' . . . . .	6
3.2	RISPOSTA ALLA RAMPA . . . . .	7
3.3	RISPOSTA ALLA SINUSOIDE . . . . .	8
<b>4</b>	<b>CONTROLLORE PID</b>	<b>9</b>
4.1	INTRODUZIONE AI CONTROLLORI PID . . . . .	9
4.2	CONTROLLORE PROPORZIONALE - P . . . . .	10
4.3	CONTROLLORE PROPORZIONALE/DERIVATIVO - PD .	13
4.4	CONTROLLORE PROPORZIONALE/INTEGRALE/DERIVATIVO - PID . . . . .	18
4.5	CONTROLLORE PID - SOLUZIONE ALTERNATIVA . . .	33
<b>5</b>	<b>CONTROLLORE MEDIANTE LUOGO DELLE RADICI</b>	<b>35</b>
5.1	IL LUOGO DELLE RADICI E IL CALCOLO DEI PARA- METRI . . . . .	35
5.2	DISEGNO DEL LUOGO E AGGIUNTA DELLA RETE COR- RETTRICE . . . . .	35
<b>6</b>	<b>CONTROLLORE CON RETI CORRETTRICI (STUDIO IN FREQUENZA)</b>	<b>39</b>
6.1	INTRODUZIONE E CALCOLO DEI PARAMETRI . . . . .	39
6.2	SISTEMA IN CATENA APERTA E SISTEMA IN CATENA CHIUSA . . . . .	39
6.3	RETE CORRETTRICE . . . . .	41
6.4	RETE CORRETTRICE - SOLUZIONE ALTERNATIVA . .	46
<b>7</b>	<b>CONFRONTI E CONCLUSIONI</b>	<b>48</b>
7.1	CONFRONTO TRA I METODI . . . . .	48
7.2	CONFRONTO TRA I RISULTATI . . . . .	49
<b>A</b>	<b>FUNZIONI MATLAB</b>	<b>53</b>
<b>B</b>	<b>BIBLIOGRAFIA</b>	<b>54</b>
<b>C</b>	<b>FIGURE</b>	<b>55</b>



# 1 INTRODUZIONE

Questo elaborato rappresenta la relazione del lavoro da me svolto e descrive le fasi di analisi e progettazione del sistema di controllo del beccheggio di un aereo. La prima parte del lavoro consiste nella inquadratura preliminare del problema, lo studio del sistema e l'individuazione del modello matematico. In tale fase si introducono anche i vincoli in termini di prestazioni che il sistema dovrà rispettare. La seconda parte si occupa della progettazione vera e propria del controllore. Esso è stato sviluppato facendo ricorso a 3 metodologie: la sintesi tramite controllori PID, la sintesi tramite lo studio del luogo delle radici e la sintesi tramite reti correttrici nel dominio della frequenza. Per ognuna delle tre metodologie si espongono i vari passi della progettazione e i risultati ottenuti. Tutto il lavoro svolto è stato fatto facendo uso delle tecniche e della teoria nell'ambito della disciplina dei controlli automatici. Oltre a tali strumenti, ho fatto uso di software come Matlab, Simulink e SISOTOOL.

## 2 ANALISI DEL PROBLEMA

Le variazioni dell'assetto longitudinale di un velivolo vengono descritte da vari parametri. Tra i più importanti vi è l'angolo di beccheggio, che descrive l'angolo tra l'orizzonte e l'asse dell'aereo. Per modificare tale grandezza si agisce sugli equilibratori, apposite parti mobili poste sulla coda dei velivoli. Il sistema di controllo del beccheggio di un aereo si occupa di far variare l'angolo di beccheggio del velivolo in corrispondenza ad una variazione dell'inclinazione degli equilibratori. Analizziamo ora il sistema da controllare e le richieste che esso deve rispettare.

### 2.1 DESCRIZIONE DEL PROBLEMA

Il movimento di un aereo in volo è un fenomeno molto complesso. Per descriverlo è necessario un sistema di sei equazioni differenziali accoppiate non lineari. Manipolando tale sistema e facendo alcune assunzioni, è possibile disaccoppiare e linearizzare le equazioni originarie e ottenere due insiemi di equazioni: uno per i movimenti laterali e uno per quelli longitudinali. Proprio di questo ultimo insieme ho fatto uso nella mia analisi.

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

Legenda:

$\alpha$  = Angolo d'attacco (attack angle)

$\theta$  = Angolo di beccheggio (pitch angle)

$\gamma_e = \theta - \alpha$

$\delta_e$  = Angolo equilibratori

U = Velocità di volo all'equilibrio

q = Indice di beccheggio

$i_y$  = Momento di inerzia normalizzato

$\rho_e$  = 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 di beccheggio

$\eta = \mu\sigma C_M$

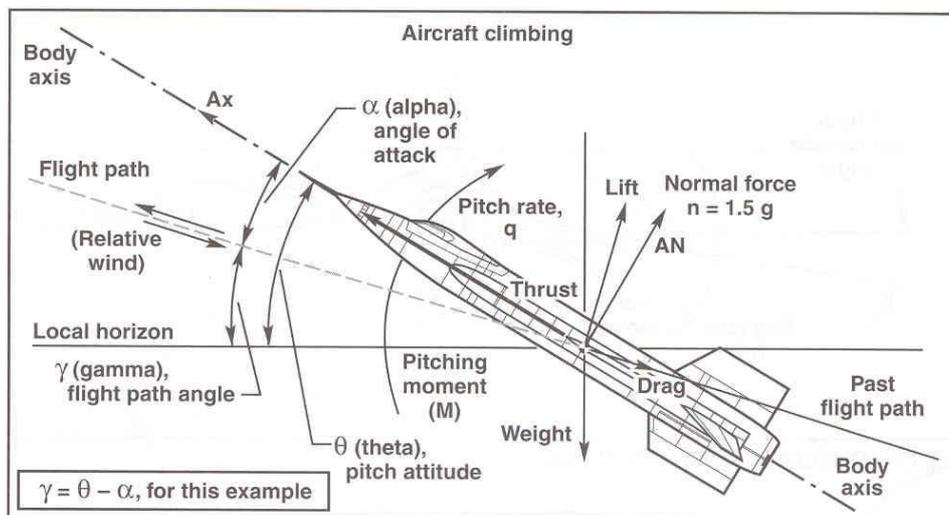


Figura 1: Parametri di Volo

Tali equazioni descrivono il movimento longitudinale dell'aereo nel caso esso si trovi in una situazione di velocità e altitudine costante. La spinta aerodinamica (thrust) e la resistenza dell'aria (drag) si cancellano e la portanza (lift) e la forza peso (weight) si equivalgono. Inoltre variazioni dell'angolo di beccheggio non causano variazioni nella velocità dell'aereo. Queste sono tutte semplificazioni al problema originale, ma permettono in prima battuta una modellizzazione più agevole.

## 2.2 SPECIFICHE DI CONTROLLO

Il sistema dovrà rispettare alcuni specifiche ben precise:

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

Un paio di esempi:

- Una variazione di 0.2 radianti ( $11^\circ$  circa) degli equilibratori deve corrispondere ad una variazione dell'angolo di beccheggio che raggiunga i 0.2 rad in meno di 2 secondi, senza eccedere oltre i 0.22 rad e stabilizzarsi in un intervallo del 2% dal valore obiettivo (quindi tra i 0.196 rad e i 0.204 rad) in un tempo inferiore ai 10 secondi

- Una variazione di 0.5 radianti (28,6° circa) degli equilibratori deve corrispondere ad una variazione dell'angolo di beccheggio che raggiunga i 0.5 rad in meno di 2 secondi, senza eccedere oltre i 0.55 rad e stabilizzarsi in un intervallo del 2% dal valore obiettivo (quindi tra i 0.49 rad e i 0.51 rad) in un tempo inferiore ai 10 secondi

### 2.3 DERIVAZIONE DEL MODELLO MATEMATICO

Nel sistema di equazioni riportato sopra bisogna dare dei valori ai parametri, per poter cominciare la progettazione di uno specifico controllore.

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

I valori sono presi dai dati di un velivolo commerciale della Boeing. Dal sistema appena scritto, devo ricavare la funzione di trasferimento. Per fare ciò, per prima cosa applico la trasformata di Laplace:

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

Con qualche passaggio algebrico ottengo la funzione di trasferimento:

$$\text{dalla prima equazione esplicito } \alpha(s) : \alpha(s) = \frac{56.7q(s) + 0.232\delta_e(s)}{s + 0.313}$$

lo sostituisco nella seconda equazione ed esplicito  $q(s)$  :

$$\begin{aligned}q(s)(s + 0.426) &= -0.0139 \frac{56.7q(s) + 0.232\delta_e(s)}{s + 0.313} + 0.0203\delta_e(s) \\ q(s) \left( s + 0.426 + 0.0139 \frac{56.7}{s + 0.313} \right) &= \delta_e(s) \left( -0.0139 \frac{0.232}{s + 0.313} + 0.0203 \right) \\ q(s) &= \delta_e(s) \frac{-0.0139 \frac{0.232}{s + 0.313} + 0.0203}{s + 0.426 + 0.0139 \frac{56.7}{s + 0.313}}\end{aligned}$$

ora sostituisco  $q(s)$  nell'ultima equazione :

$$\theta(s) = \frac{57.6}{s} \delta_e(s) \frac{-0.0139 \frac{0.232}{s + 0.313} + 0.0203}{s + 0.426 + 0.0139 \frac{56.7}{s + 0.313}}$$

isolo  $\frac{\theta(s)}{\delta_e(s)}$  e ottengo la funzione di trasferimento(FDT):

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

Dal sistema iniziale posso anche ricavare il modello di stato:

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

Da questo estrapolo l'equazione di interesse per il problema:

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

Ora che ho ricavato il modello matematico nelle sue diverse forme, posso cominciare a studiare il sistema descritto dall'equazione di stato.

### 3 ANALISI DEL SISTEMA IN CATENA APERTA

Eseguo lo studio del sistema in catena aperta, controllando la stabilità, la risposta al gradino, la risposta alla rampa e la risposta ai segnali sinusoidali.

#### 3.1 RISPOSTA AL GRADINO E STABILITÀ

Uso Matlab per definire la FDT e simulare la risposta al gradino unitario.

```
de=0.2; %ampiezza del gradino in input
num=[1.151 0.1774]; %numeratore della FDT
den=[1 0.739 0.921 0]; %denominatore della FDT
pitch=tf(num,den); %calcolo della FDT
sys=de*pitch; %imposto l'ampiezza del gradino
t=0:0.1:20;
step(sys,t) %applicazione del gradino al sistema
```

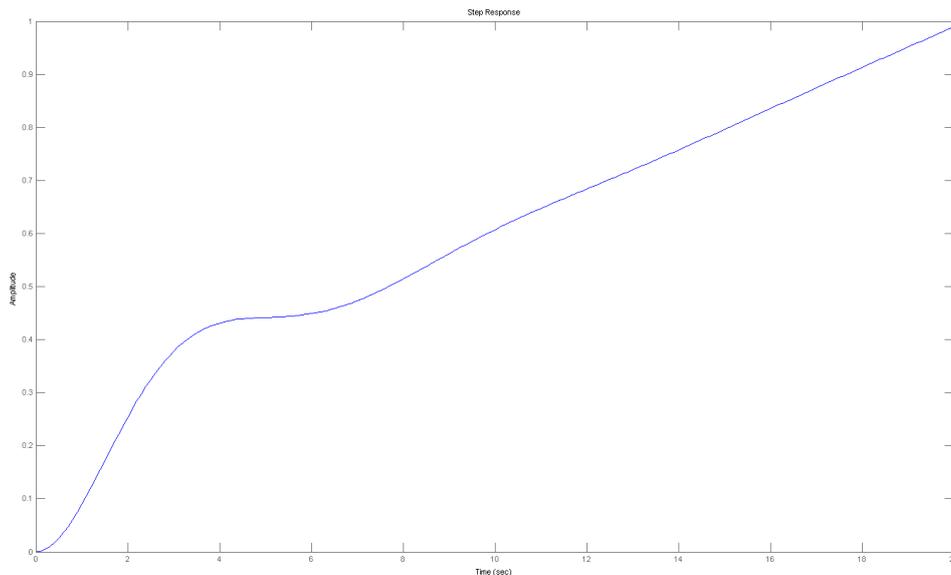


Figura 2: Risposta al gradino del sistema in catena aperta

Dal grafico è evidente l'instabilità del sistema: il segnale in output infatti diverge anziché assestarsi ad un valore costante. Per confermare l'instabilità applico la regola di Cartesio al denominatore della FDT: al polinomio manca il coefficiente di grado zero e quindi esso non è di Hurwitz. Questo a riprova della non stabilità del sistema in catena aperta.

### 3.2 RISPOSTA ALLA RAMPA

Per completezza, riporto la risposta del sistema con ingresso la rampa. Questa volta uso gli strumenti messi a disposizione da simulink. Per prima cosa posiziono il blocco corrispondente alla rampa. Poi inserisco il blocco relativo al sistema e infine posiziono dei blocchi per osservare la risposta del sistema. A questo punto lancio la simulazione e osservo l'output.

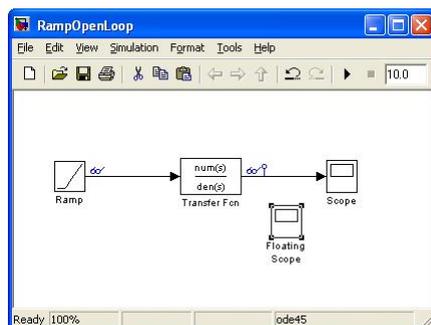


Figura 3: Schema a blocchi in Simulink del sistema in catena aperta, con ingresso la rampa

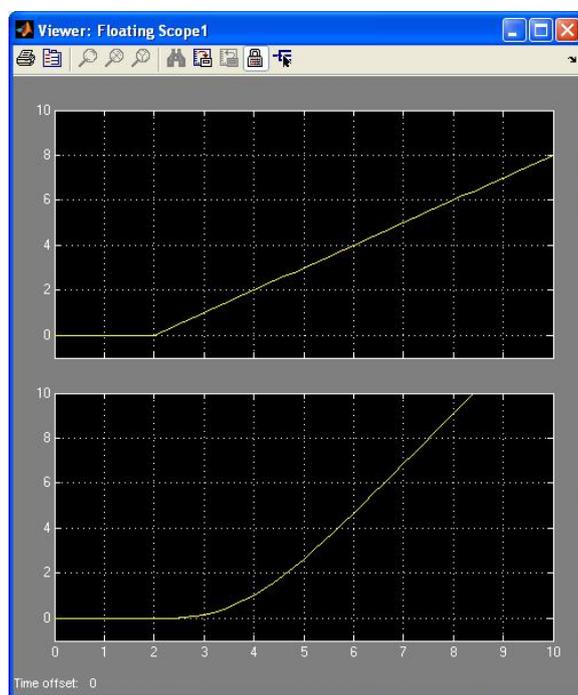


Figura 4: Input(in alto) e output(in basso) del sistema, con ingresso la rampa

### 3.3 RISPOSTA ALLA SINUSOIDE

Come ultima risposta ai segnali canonici, ecco il segnale sinusoidale. Analogamente a prima, ho fatto uso della simulazione simulink, variando il solo blocco di input.

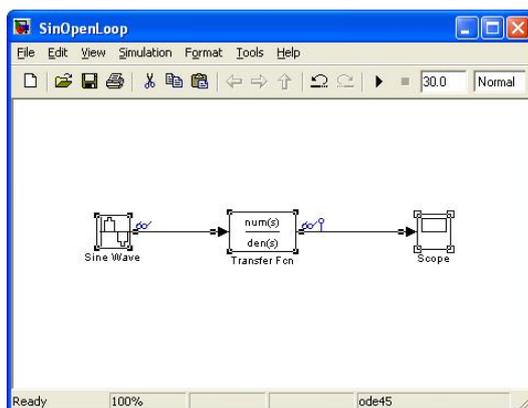


Figura 5: Schema a blocchi in Simulink del sistema in catena aperta, con input sinusoidale

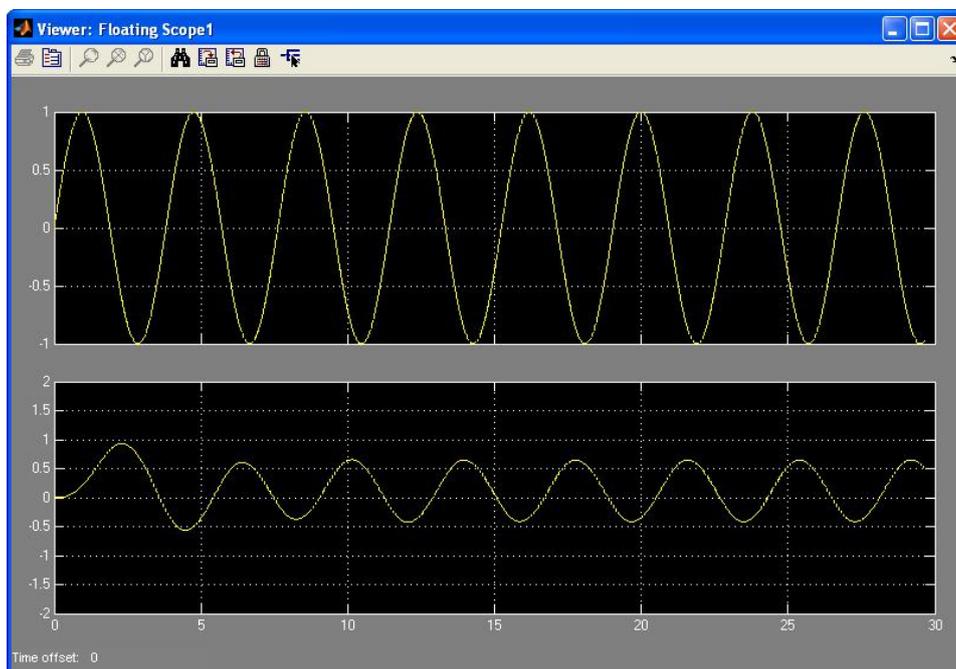


Figura 6: Input(in alto) e output(in basso) del sistema, con ingresso sinusoidale

## 4 CONTROLLORE PID

Espongo ora il primo dei metodi di sintesi usati, quello dei controllori PID.

### 4.1 INTRODUZIONE AI CONTROLLORI PID

Un controllore PID (Proporzionale Integrale Derivativo) è descritto da una funzione di trasferimento del tipo:

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

Esso lega il segnale  $u(t)$  con il segnale di ingresso  $e(t)$  attraverso la seguente relazione:

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

Il segnale  $e(t)$  rappresenta l'errore, mentre il segnale  $u(t)$  è l'input del sistema da controllare.

Il primo addendo è il termine proporzionale e ha l'effetto di ridurre il tempo di salita. Inoltre aiuta a ridurre l'errore a regime, pagando un peggioramento della sovra-elongazione. Il secondo addendo è il termine integrale e ha lo scopo di eliminare l'errore a regime nella risposta al gradino. Può causare però problemi con la stabilità, i transitori e le sovra-elongazioni. Infine l'ultimo addendo è il termine derivativo e ha la funzione di migliorare il transitorio.

La costruzione di un tale controllore consiste nel trovare dei valori per i parametri  $K_D$ ,  $K_P$  e  $K_I$ . Nella sintesi di questo tipo di controllori, non è sempre necessario ricorrere a tutti e tre i termini. Le specifiche di progetto si potrebbero raggiungere anche solo con uno o due dei tre termini a disposizione.

Le configurazioni solitamente usate sono:

- Controllore P: è presente la sola parte proporzionale
- Controllore PI: sono presenti la parte proporzionale e quella integrale
- Controllore PD: sono presenti la parte proporzionale e quella derivativa
- Controllore PID: sono presenti tutti i 3 termini

Nella ricerca del controllore, ho seguito l'ordine appena esposto. Ho tralasciato il solo controllore PI, in quanto il sistema ha già un polo nell'origine. Per ogni tipologia ho usato Matlab per generare varie configurazioni dei parametri e rappresentare graficamente i risultati. Osservando i grafici

ho poi estrapolato l'influenza che ogni parametro ha sulle specifiche e raffinato la ricerca su precisi intervalli di valori. In questo modo, con iterazioni successive, sono riuscito a trovare dei valori per le variabili in gioco.

## 4.2 CONTROLLORE PROPORZIONALE - P

Per prima cosa introduco il controllore proporzionale. Al momento quindi  $C(s) = K_P$ . Nel trovare  $K_P$ , uso un ciclo per simulare la risposta del sistema a diversi valori del parametro. Quest'ultimi sono contenuti in un vettore.

```
%cerchiamo l'ordine di grandezza di Kp
de=0.2;
t=0:0.01:30;
KpVect=[ 0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 100 ];
figure(1)
for n=1:length(KpVect)
    subplot(length(KpVect)/2,2,n)
    Kp=KpVect(n);
    sys_cl=feedback(Kp*pitch,1);
    step(de*sys_cl,t)
    legend(strcat('Kp= ',num2str(KpVect(n))));
    axis([0 30 0 0.4])
    hold on
end %end for
hold off
```

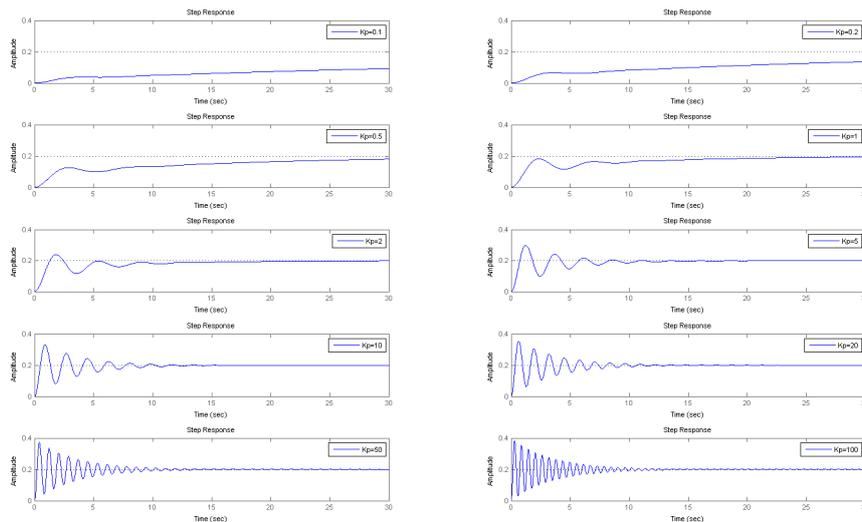


Figura 7: Risposta del sistema a diversi valori di  $K_P$

Come si evince dai grafici, i valori migliori di  $K_P$  sono circa tra 1 e 5. Analizzo in dettaglio tale intervallo:

```
%i valori migliori sono circa tra 1 e 5
%vediamoli meglio
KpVectRange =[ 1:0.2:5 ];
figure(2)
t=0:0.01:30;
for n=1:length(KpVectRange)
    subplot(7,3,n)
    Kp=KpVectRange(n);
    sys_cl=feedback(Kp*pitch,1);
    step(de*sys_cl,t)
    legend(strcat('Kp= ',num2str(KpVectRange(n))),
        'Location','SouthEast');
    axis([0 30 0 0.3])
    hold on
end %end for
hold off
```

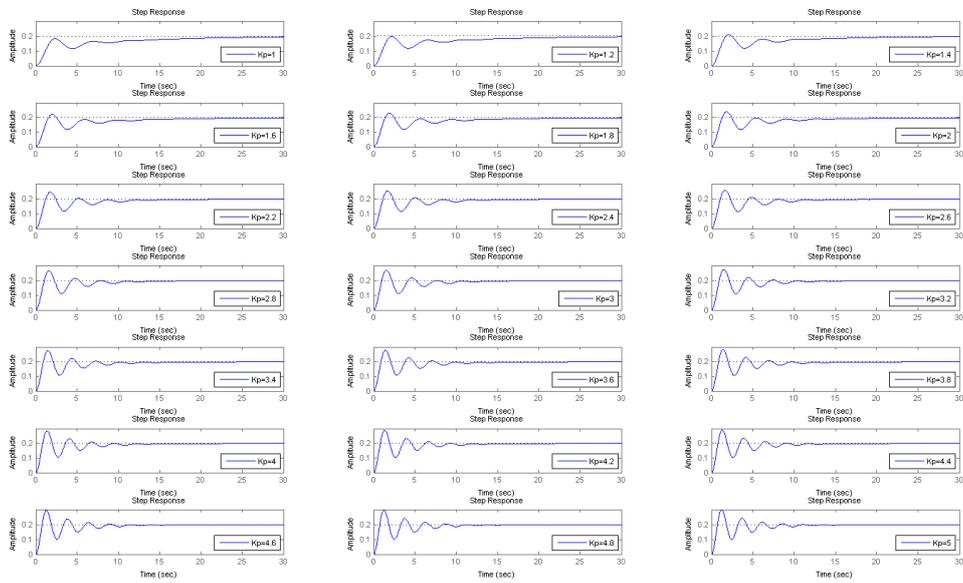


Figura 8: Risposta del sistema con valori di  $K_P$  scelti all'interno dell'intervallo individuato in precedenza

Tra questi, un valore di  $K_P = 2$  sembra essere tra i migliori.

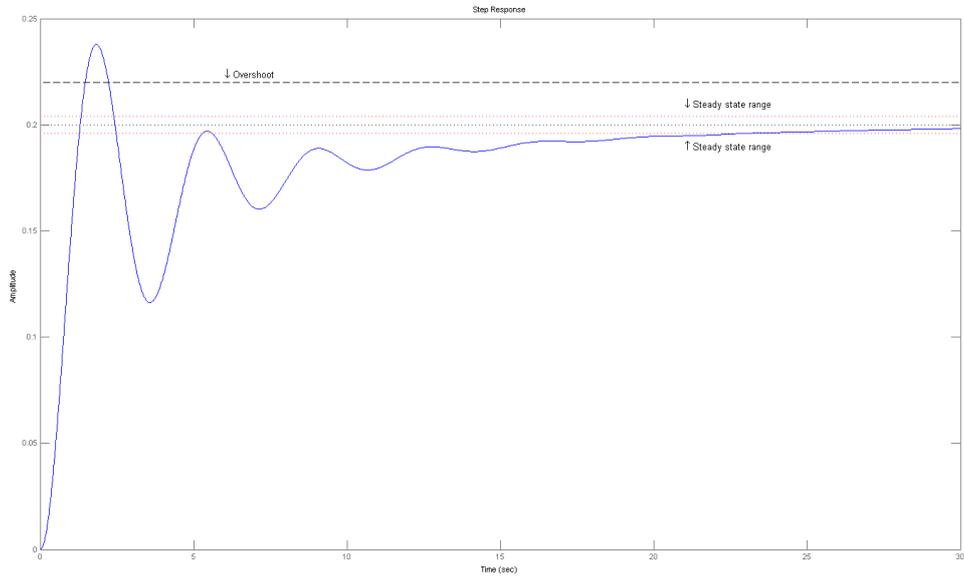


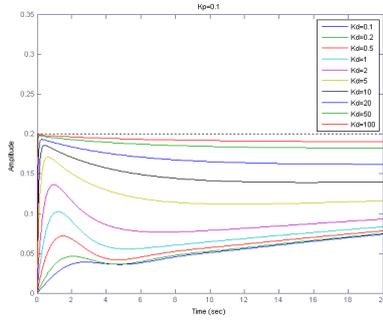
Figura 9: Risposta del sistema con  $K_P = 2$

Come si evince dal grafico, il solo controllore proporzionale non basta a soddisfare le richieste del progetto. Il tempo di salita è l'unico vincolo progettuale rispettato. L'introduzione della retroazione ha però reso stabile il sistema.

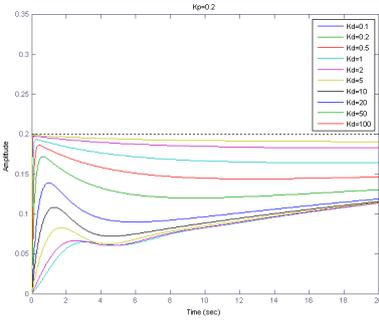
### 4.3 CONTROLLORE PROPORZIONALE/DERIVATIVO - PD

Aggiungo ora la parte derivativa. Il controllore assume quindi la forma  $C(s) = K_P + K_D s$ . Devo trovare dei valori per  $K_P$  e  $K_D$ . Analogamente a prima, uso Matlab per fare ciò.

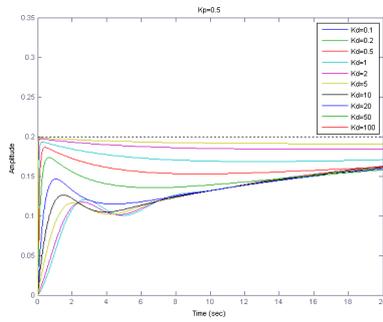
```
%introduciamo ora il controllo derivativo, Kd
de=0.2;
t=0:0.01:30;
KpVect=[ 0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 100 ];
KdVect=[ 0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 100 ];
%cerchiamo l'ordine di grandezza di Kp e Kd
for n=1:length(KpVect)
    figure(n)
    Kp=KpVect(n);
    for m=1:length(KdVect)
        Kd=KdVect(m);
        contr=tf([Kd Kp],1);
        sys_cl=feedback(contr*pitch,1);
        step(de*sys_cl,t);
        hold on
    end %end for
    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 %end for
hold off
```



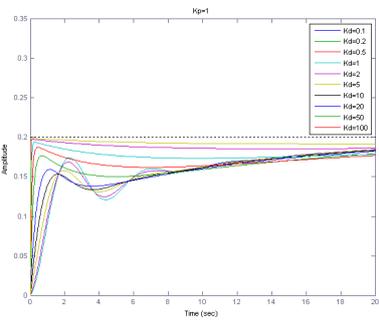
(a)  $K_P = 0.1$



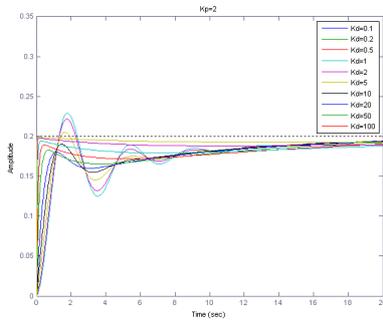
(b)  $K_P = 0.2$



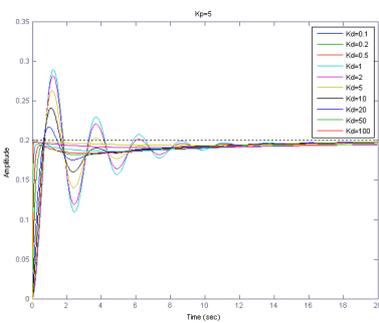
(c)  $K_P = 0.5$



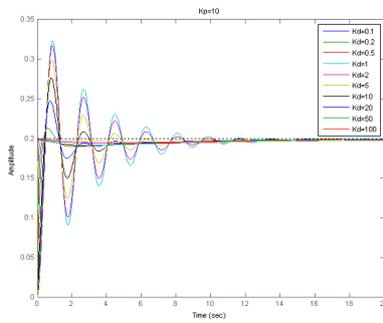
(d)  $K_P = 1$



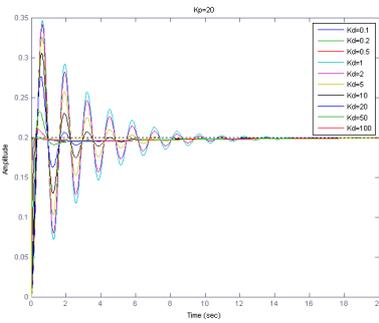
(e)  $K_P = 2$



(f)  $K_P = 5$



(g)  $K_P = 10$



(h)  $K_P = 20$

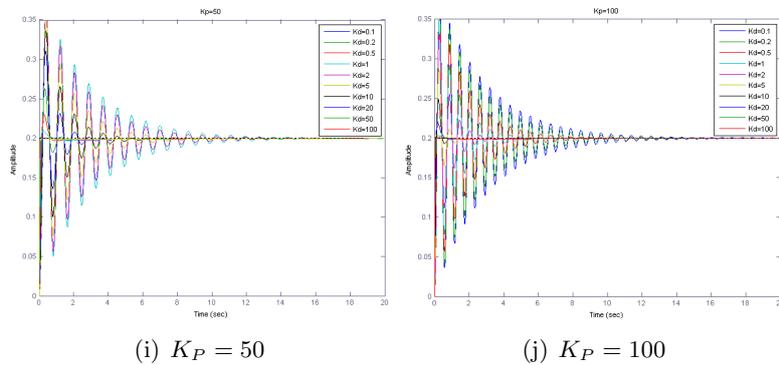


Figura 10: Risposte del sistema al variare di  $K_P$  e  $K_D$ . Ogni figura è relativa al valore di  $K_P$  riportato nella didascalia.

Per valori di  $K_P$  minori di 5, dai grafici si vede una difficoltà a raggiungere il valore di regime. Tollo perciò tali valori e mi concentro su quelli maggiori/uguali di 5. In questo nuovo insieme noto che i valori bassi di  $K_D$  causano oscillazioni molto pronunciate. Uso quindi solo valori di  $K_D \geq 2$ . Ecco i grafici con questi valori di  $K_P$  e  $K_D$ .

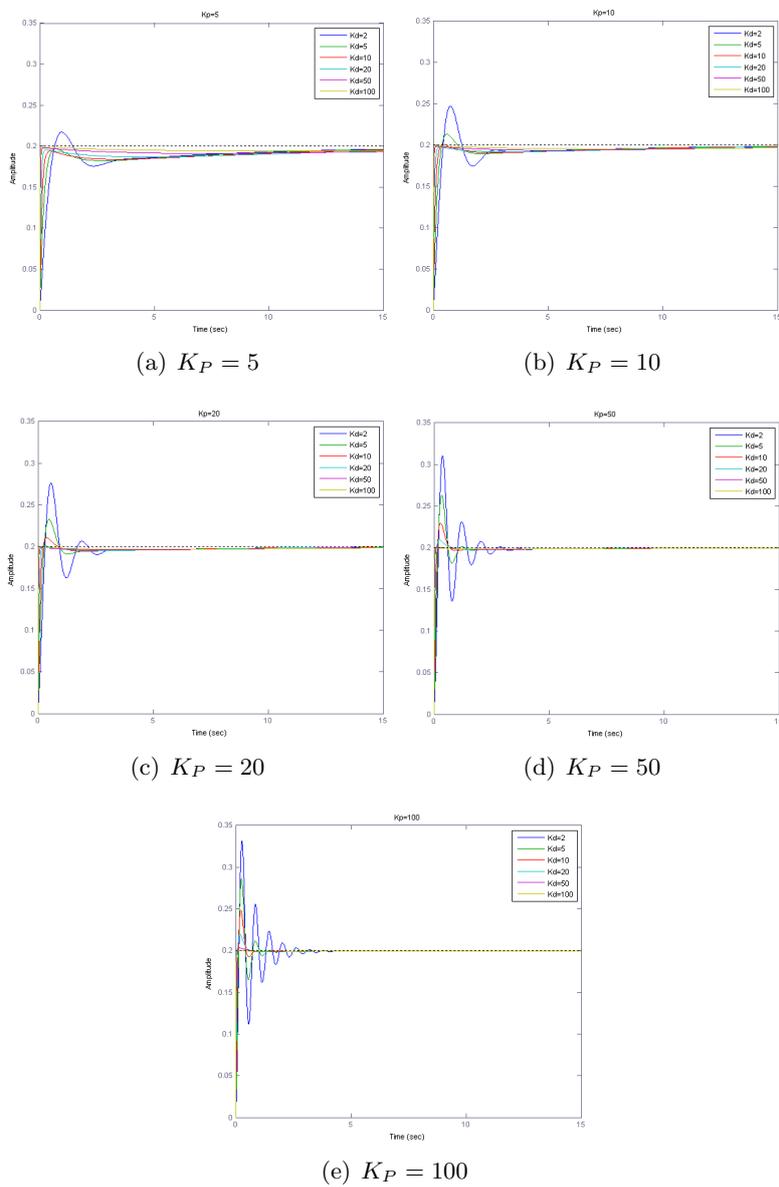


Figura 11: Risposta del sistema per diversi valori di  $K_P$  e  $K_D$ , scelti negli intervalli sopra specificati.

Tra questi, i sistemi che si avvicinano più alle specifiche sono quelli con  $K_D$  elevato. Valori di  $K_D$  elevati sono però pericolosi per la stabilità. Per tali motivi, i valori di  $K_D$  maggiori di 10 li ho esclusi, pur garantendo prestazioni elevate. Tra tutte quelle riportate negli ultimi grafici, ho scelto la configurazione con  $K_P = 20$  e  $K_D = 10$ , che sembra essere tra le migliori.

```
figure(21)
t=0:0.01:15;
de=0.2; %Ampiezza gradino
er=0.02; %Errore a regime
ov=0.1; %Overshoot(%)
Kp=20; Kd=10;
contr=tf([Kd Kp],1);
sys_cl=feedback(contr*pitch,1);
step(de*sys_cl,t)
legend('Kp=20 Kd=10');
hold on
plotInformation(de,t,6,10,er,ov);
hold off
```

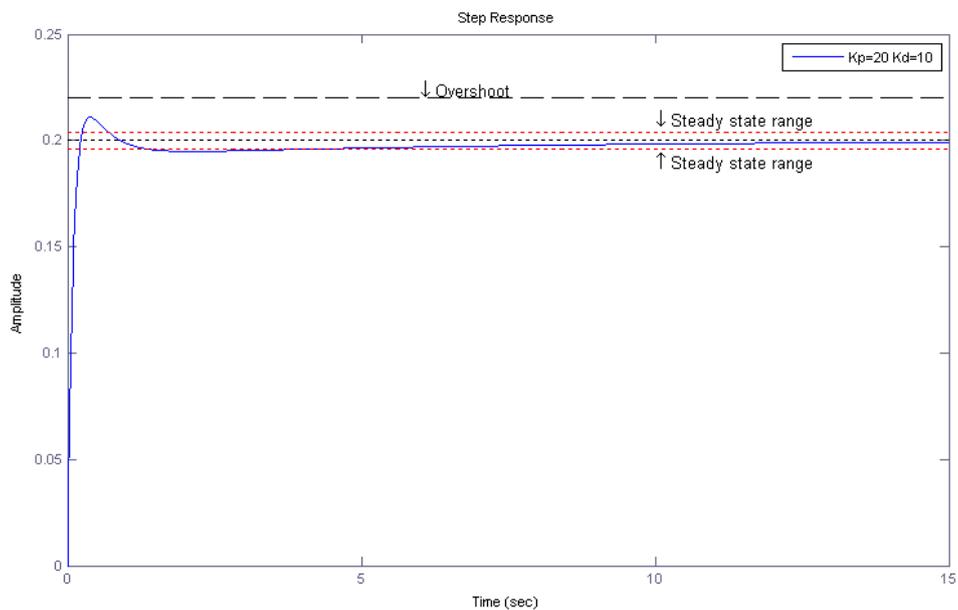


Figura 12: Risposta del sistema con  $K_P = 20$  e  $K_D = 10$

Come si evince dal grafico, il controllore così ottenuto soddisfa tutte le specifiche di progetto. Il tutto senza dover ricorrere alla azione integrale. Il solo controllore PD è quindi sufficiente per rispettare i vincoli di progetto.

#### 4.4 CONTROLLORE PROPORZIONALE/INTEGRALE/DERIVATIVO - PID

Pur avendo già raggiunto gli obiettivi con il controllore PD, per completezza studio anche gli effetti dell'aggiunta della parte integrale. Il controllore assume quindi la forma  $C(s) = K_P + K_D s + K_I/s$ . Devo trovare dei valori per  $K_P$ ,  $K_D$  e  $K_I$ . Analogamente a prima, uso Matlab per fare ciò.

```
%cerchiamo l'ordine di grandezza di Kp e Kd e Ki
t=0:0.1:20;
de=0.2;
for n=1:length(KpVect)

    figure(n)
    title(strcat('Kp= ',num2str(KpVect(n))));
    Kp=KpVect(n);
    num=[1.151 0.1774];
    den=[1 0.739 0.921 0];
    for m=1:length(KdVect)
        Kd=KdVect(m);
        subplot(length(KdVect)/2,2,m);
        pitch=tf(num,den);
        for l=1:length(KiVect)
            Ki=KiVect(l);
            contr=tf([Kd Kp Ki],[1 0]);
            sys_cl=feedback(contr*pitch,1);
            step(de*sys_cl,t)
            axis([0 20 0 0.35])
            hold on
        end %end for
        legend('Ki=0.1','Ki=0.2','Ki=0.5','Ki=1','Ki=2','Ki=5',
            'Ki=10','Ki=20','Ki=50','Ki=100','location','EastOutside');
        title(strcat('Kd= ',num2str(KdVect(m))));
    end %end for
    hold on
end %end for
hold off
```

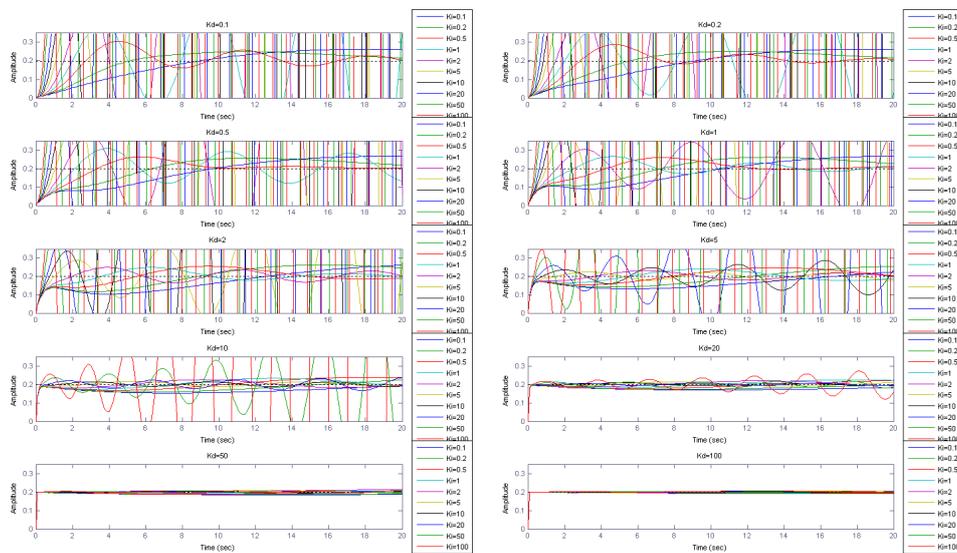


Figura 13: Risposta del sistema con  $K_P = 0.1$  e vari valori di  $K_D$  e  $K_I$

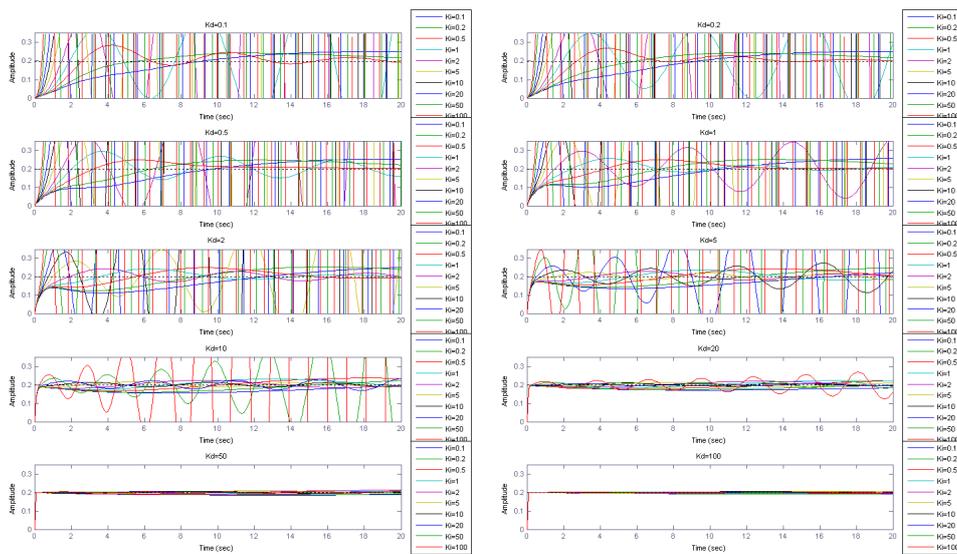


Figura 14: Risposta del sistema con  $K_P = 0.2$  e vari valori di  $K_D$  e  $K_I$

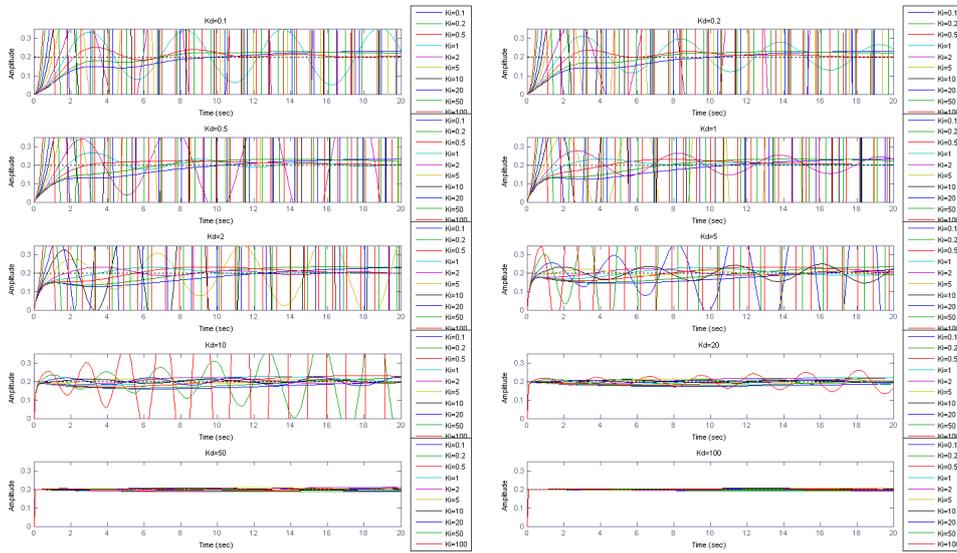


Figura 15: Risposta del sistema con  $K_P = 0.5$  e vari valori di  $K_D$  e  $K_I$

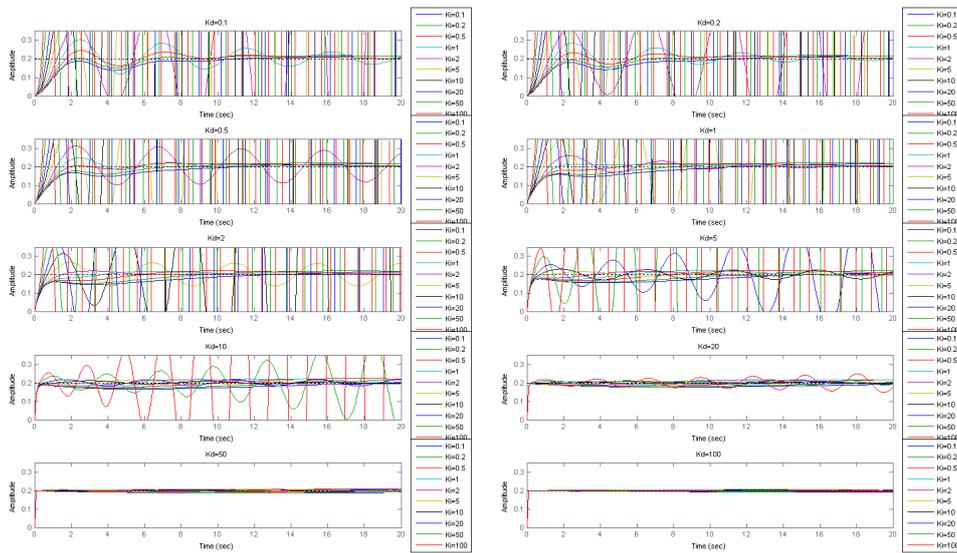


Figura 16: Risposta del sistema con  $K_P = 1$  e vari valori di  $K_D$  e  $K_I$

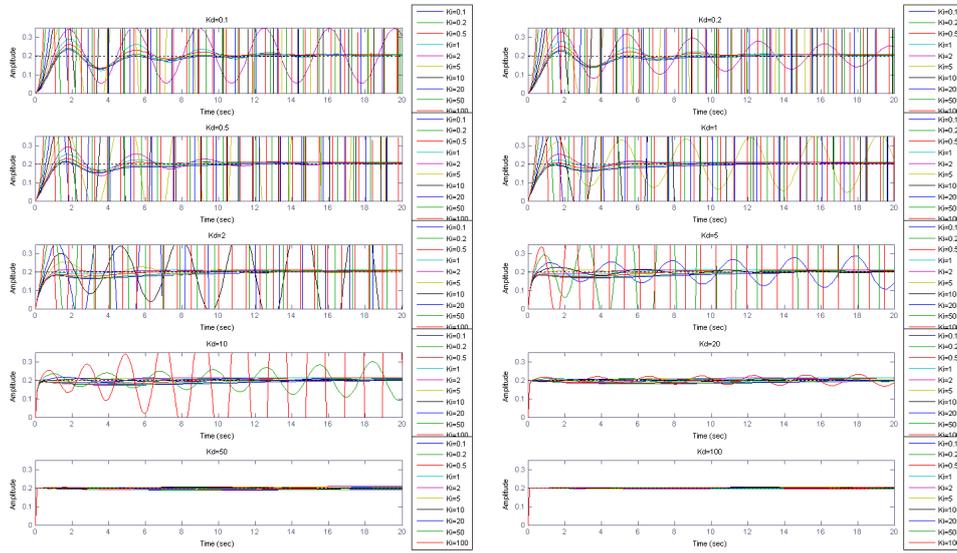


Figura 17: Risposta del sistema con  $K_P = 2$  e vari valori di  $K_D$  e  $K_I$

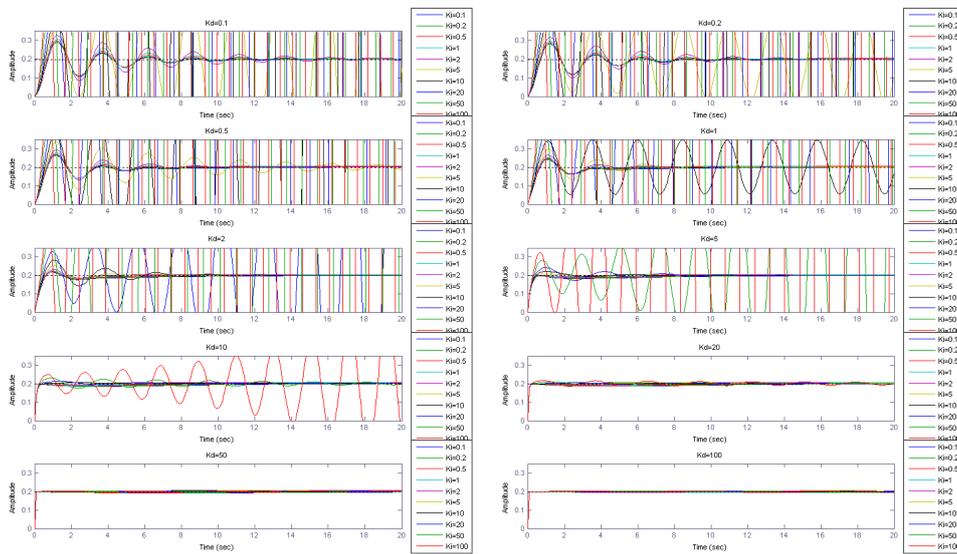


Figura 18: Risposta del sistema con  $K_P = 5$  e vari valori di  $K_D$  e  $K_I$

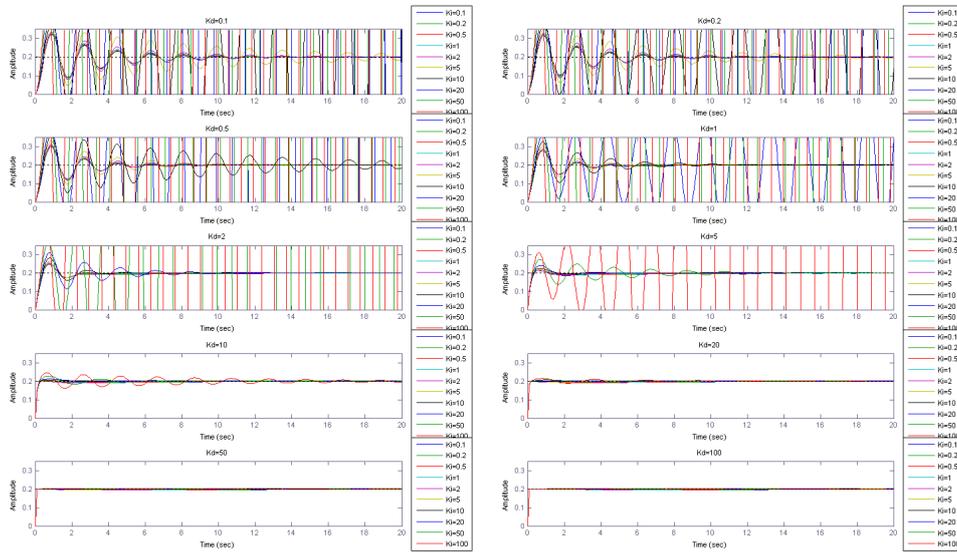


Figura 19: Risposta del sistema con  $K_P = 10$  e vari valori di  $K_D$  e  $K_I$

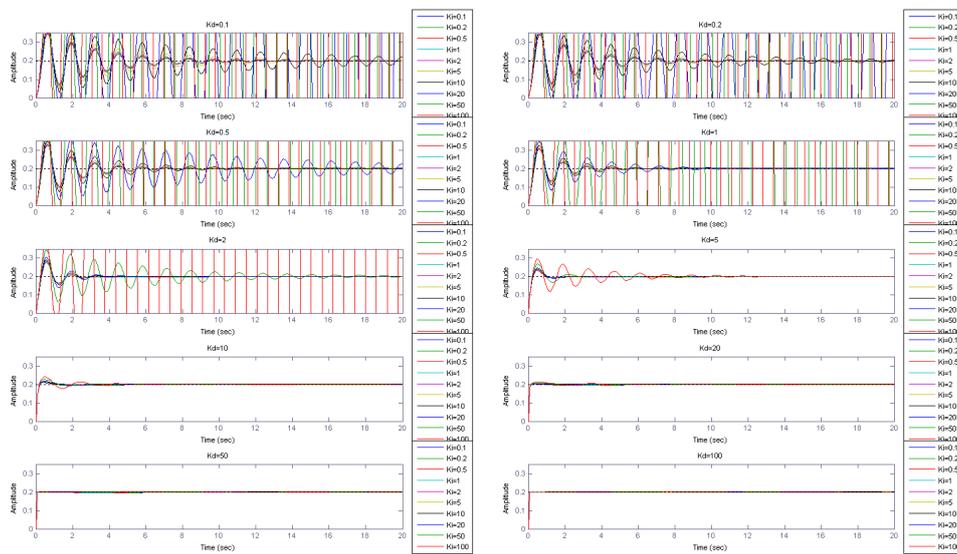


Figura 20: Risposta del sistema con  $K_P = 20$  e vari valori di  $K_D$  e  $K_I$

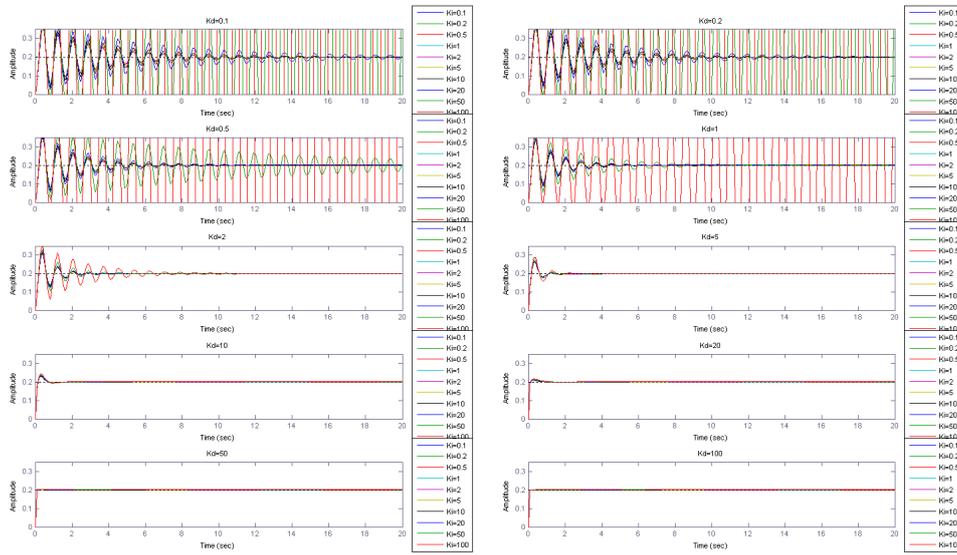


Figura 21: Risposta del sistema con  $K_P = 50$  e vari valori di  $K_D$  e  $K_I$

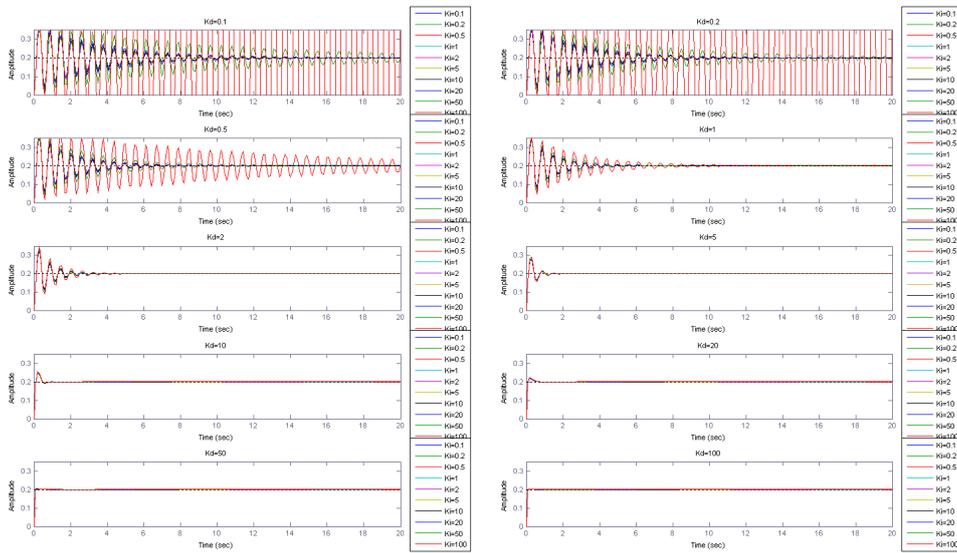


Figura 22: Risposta del sistema con  $K_P = 100$  e vari valori di  $K_D$  e  $K_I$

Dai grafici osservo che valori alti di  $K_I$  sono spesso fonte di oscillazioni divergenti e/o overshoot pronunciati. Tralascio quindi i valori 50 e 100. Sempre osservando i grafici, vedo che i valori di  $K_D$  fino a 1 danno tempi di salita lunghi o overshoot. Tolgo quindi tali valori dalla prossima analisi, insieme ai valori 50 e 100. Cerco negli insiemi di valori così ristretti le configurazioni migliori.

```

for n=1:length(KpVect)
    figure(n+20)
    title(strcat('Kp= ',num2str(KpVect(n))));
    Kp=KpVect(n);
    num=[1.151 0.1774];
    den=[1 0.739 0.921 0];
    for m=5:length(KdVect)-2
        Kd=KdVect(m);
        subplot(2,2,m-4);
        pitch=tf(num,den);
        for l=1:length(KiVect)-2
            Ki=KiVect(l);
            contr=tf([Kd Kp Ki],[1 0]);
            sys_cl=feedback(contr*pitch,1);
            step(de*sys_cl,t)
            axis([0 20 0 0.35])
            hold on
        end %end for
        legend('Ki=0.1','Ki=0.2','Ki=0.5','Ki=1','Ki=2',
            'Ki=5','Ki=10','Ki=20','location','EastOutside');
        title(strcat('Kd= ',num2str(KdVect(m))));
    end %end for
    hold on
end %end for
hold off

```

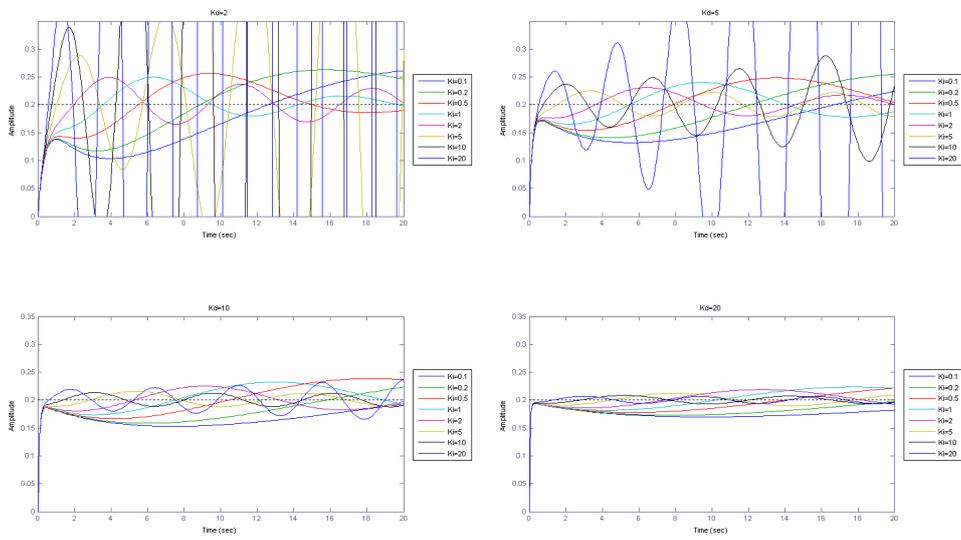


Figura 23: Risposta del sistema con  $K_P = 0.1$  e vari valori di  $K_D$  e  $K_I$ , scelti negli intervalli sopra specificati.

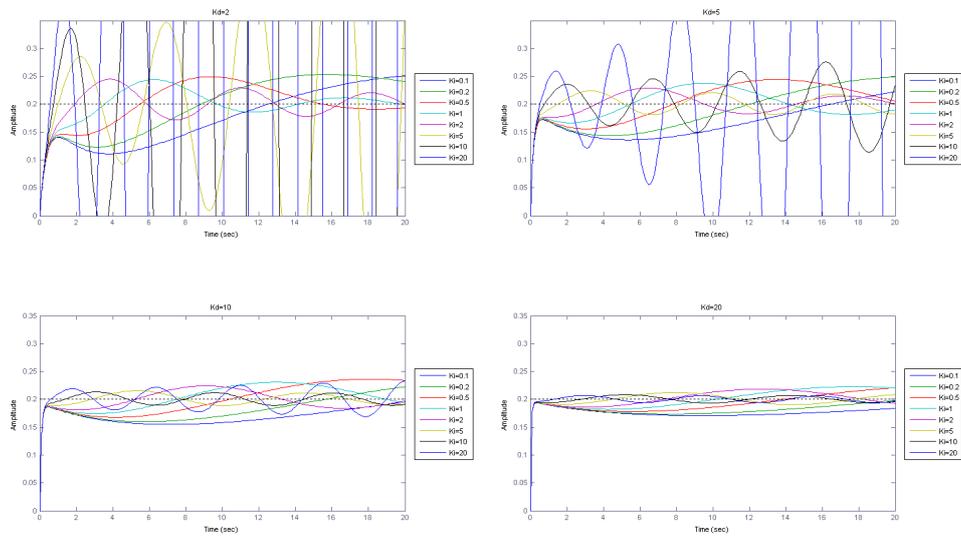


Figura 24: Risposta del sistema con  $K_P = 0.2$  e vari valori di  $K_D$  e  $K_I$ , scelti negli intervalli sopra specificati.

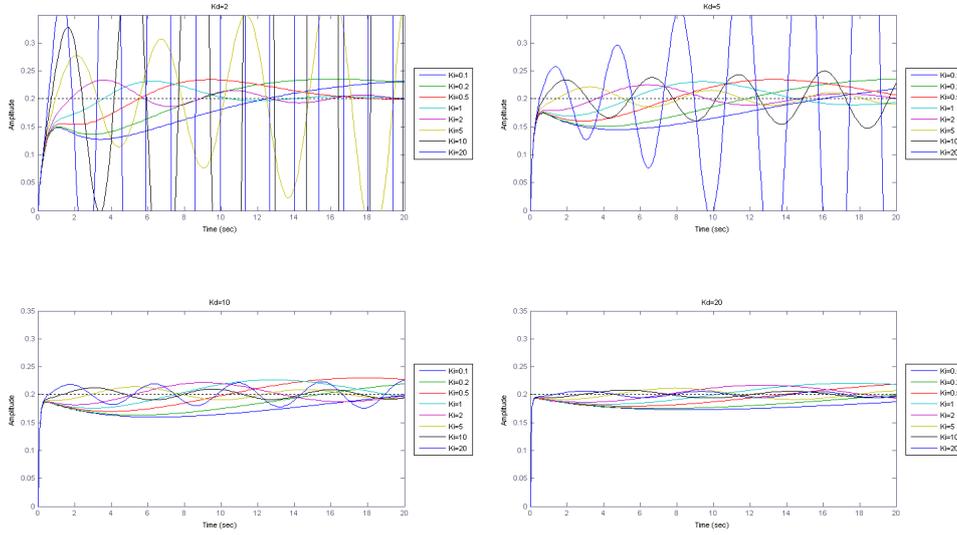


Figura 25: Risposta del sistema con  $K_P = 0.5$  e vari valori di  $K_D$  e  $K_I$ , scelti negli intervalli sopra specificati.

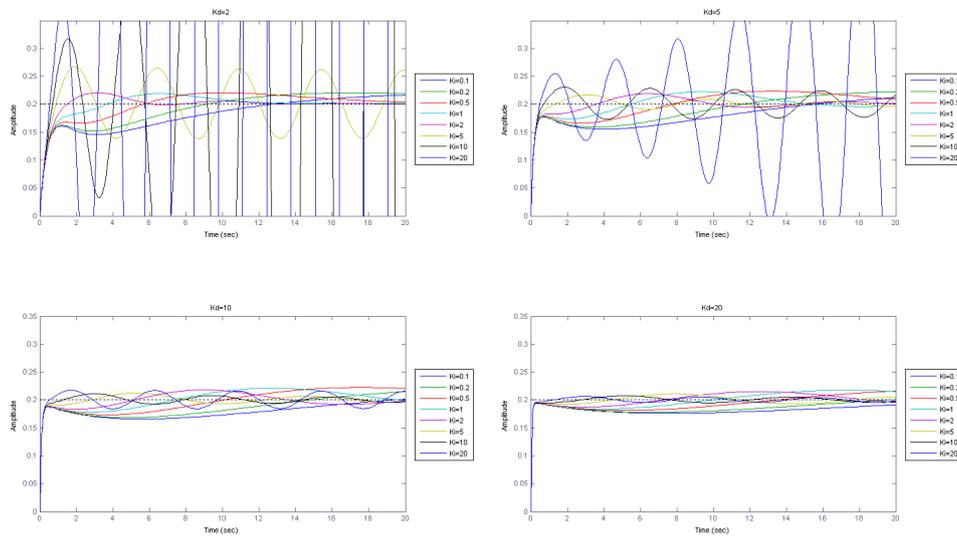


Figura 26: Risposta del sistema con  $K_P = 1$  e vari valori di  $K_D$  e  $K_I$ , scelti negli intervalli sopra specificati.

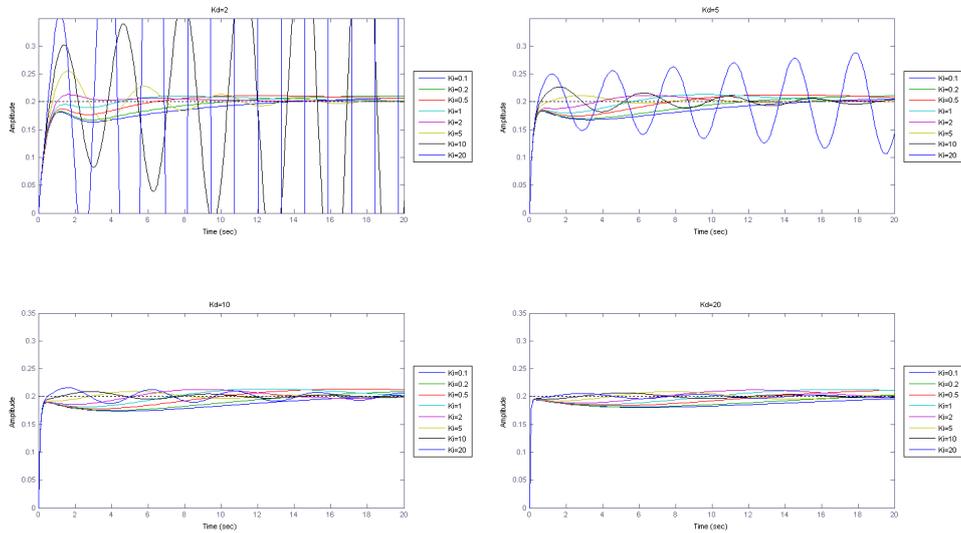


Figura 27: Risposta del sistema con  $K_P = 2$  e vari valori di  $K_D$  e  $K_I$ , scelti negli intervalli sopra specificati.

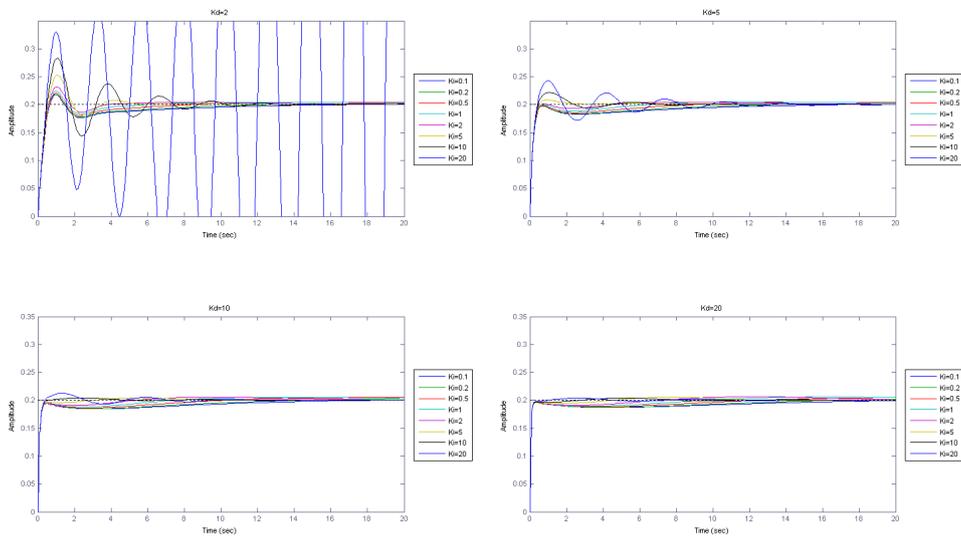


Figura 28: Risposta del sistema con  $K_P = 5$  e vari valori di  $K_D$  e  $K_I$ , scelti negli intervalli sopra specificati.

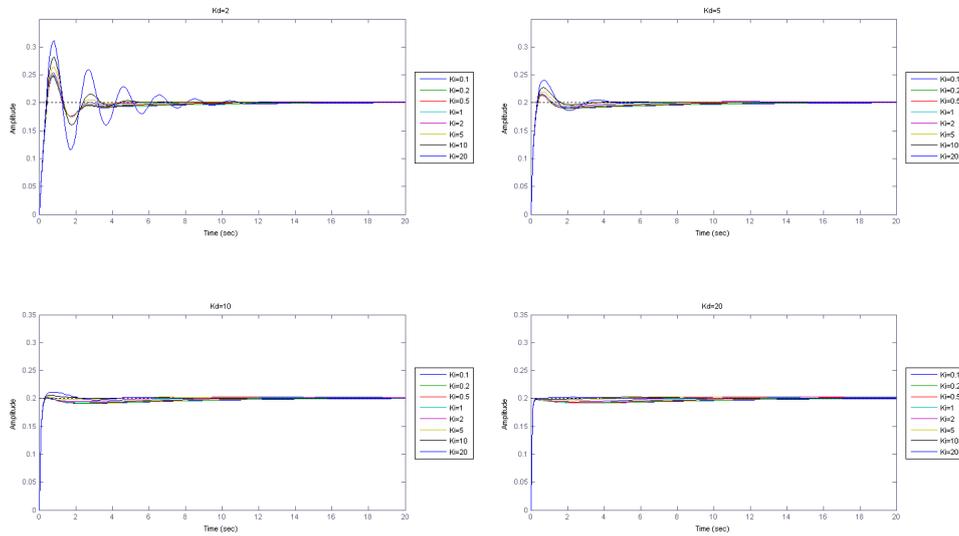


Figura 29: Risposta del sistema con  $K_P = 10$  e vari valori di  $K_D$  e  $K_I$ , scelti negli intervalli sopra specificati.

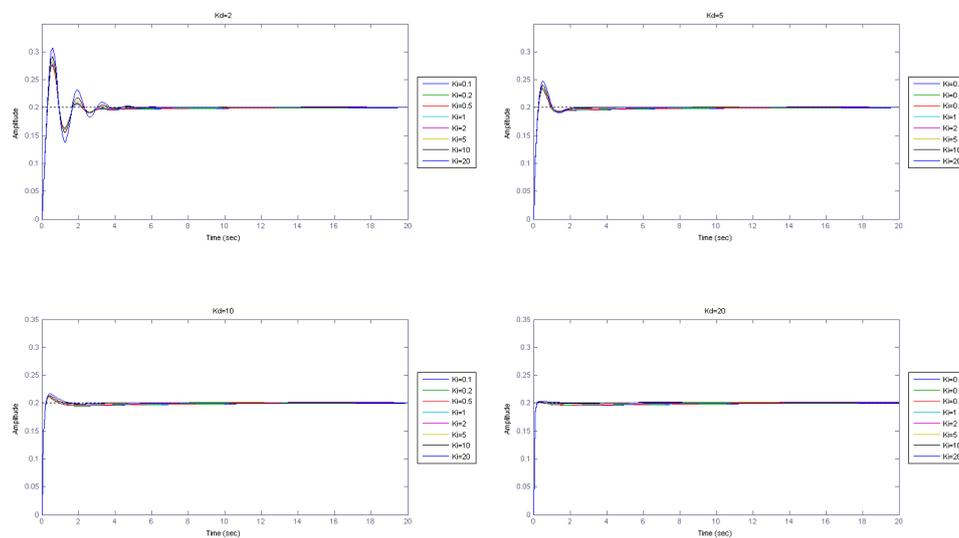


Figura 30: Risposta del sistema con  $K_P = 20$  e vari valori di  $K_D$  e  $K_I$ , scelti negli intervalli sopra specificati.

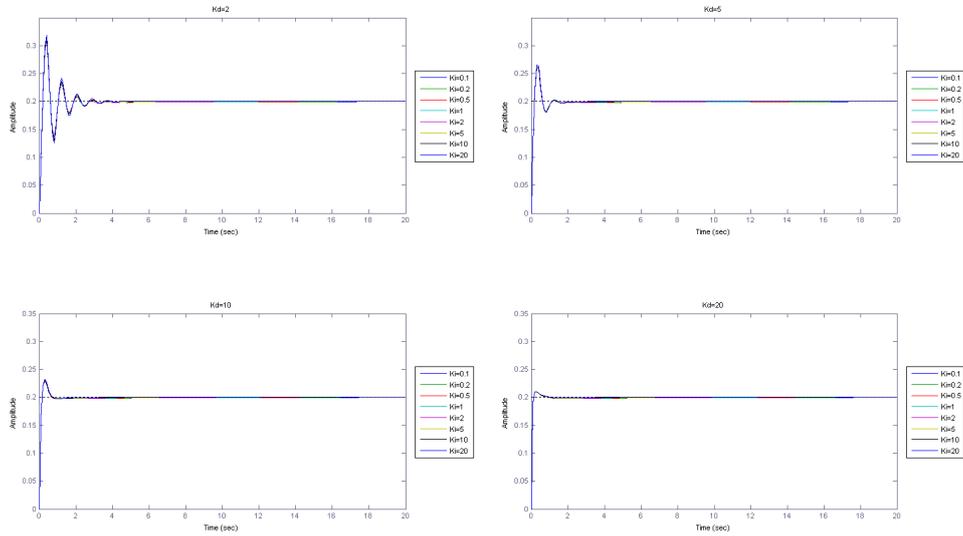


Figura 31: Risposta del sistema con  $K_P = 50$  e vari valori di  $K_D$  e  $K_I$ , scelti negli intervalli sopra specificati.

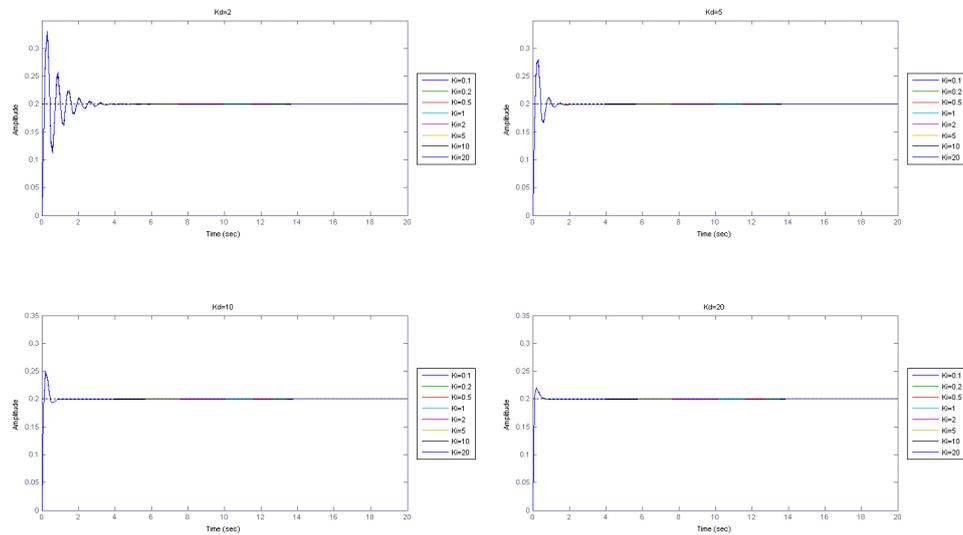


Figura 32: Risposta del sistema con  $K_P = 100$  e vari valori di  $K_D$  e  $K_I$ , scelti negli intervalli sopra specificati.

Dopo aver osservato i grafici, scelgo la configurazione con  $K_P = 10$ ,  $K_D = 10$ ,  $K_I = 5$ .

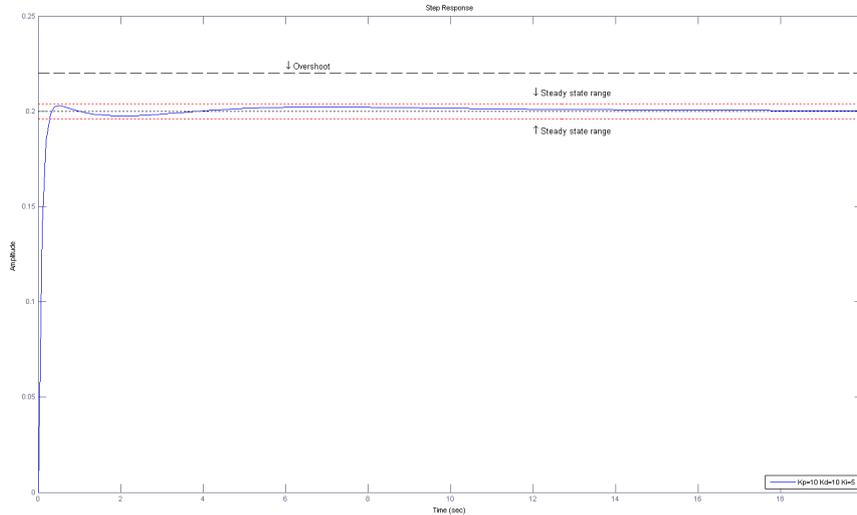


Figura 33: Risposta del sistema con  $K_P = 10$ ,  $K_D = 10$ ,  $K_I = 5$

Uso simulink per simulare il sistema con tali parametri ed imposto il controllore PID dall'apposito pannello.

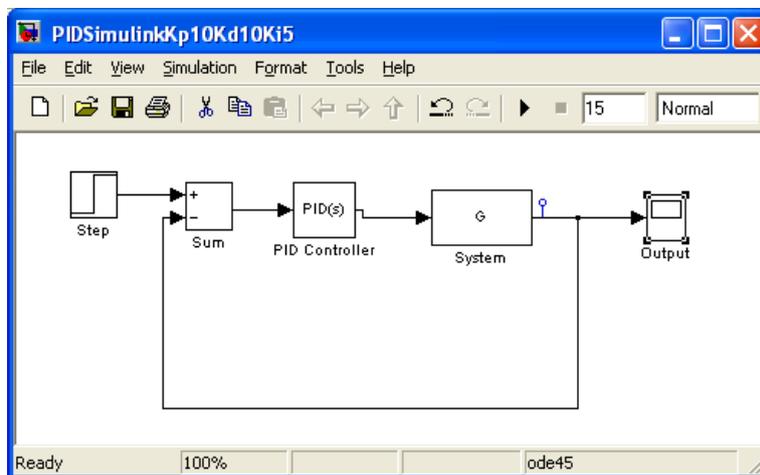


Figura 34: Schema simulink del sistema in catena chiusa con la presenza del controllore PID

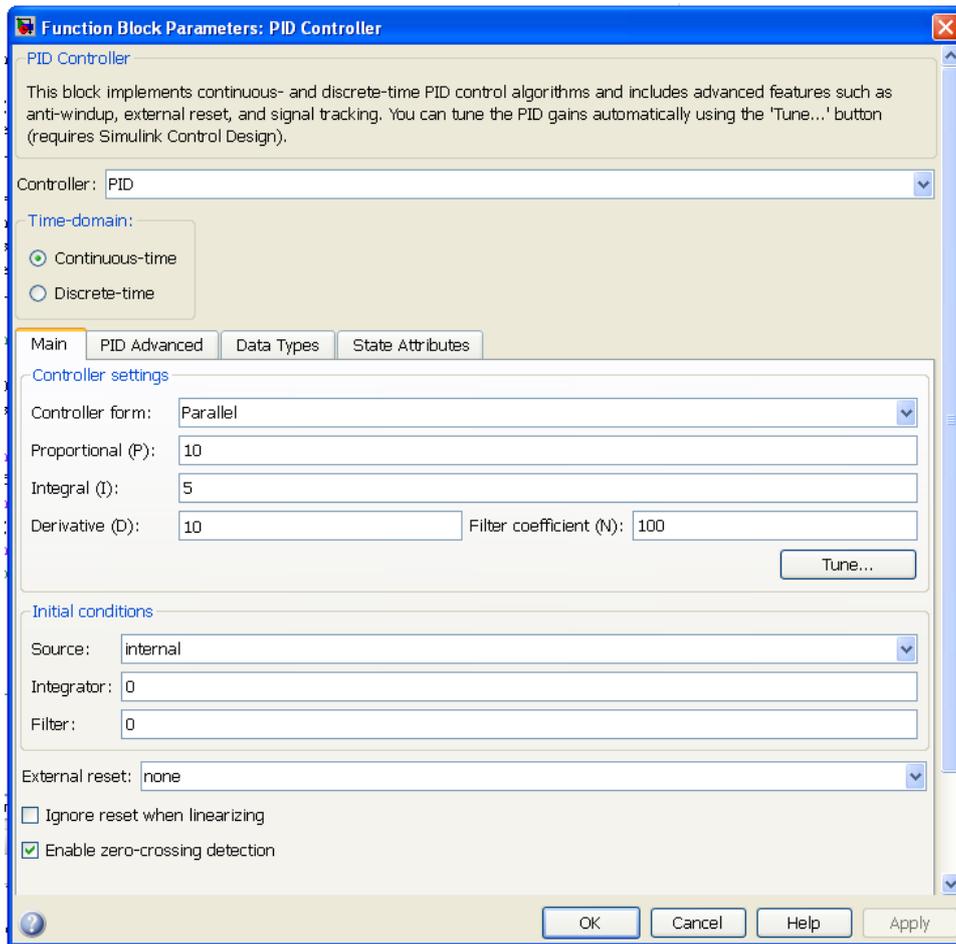


Figura 35: Pannello di configurazione del controllore PID

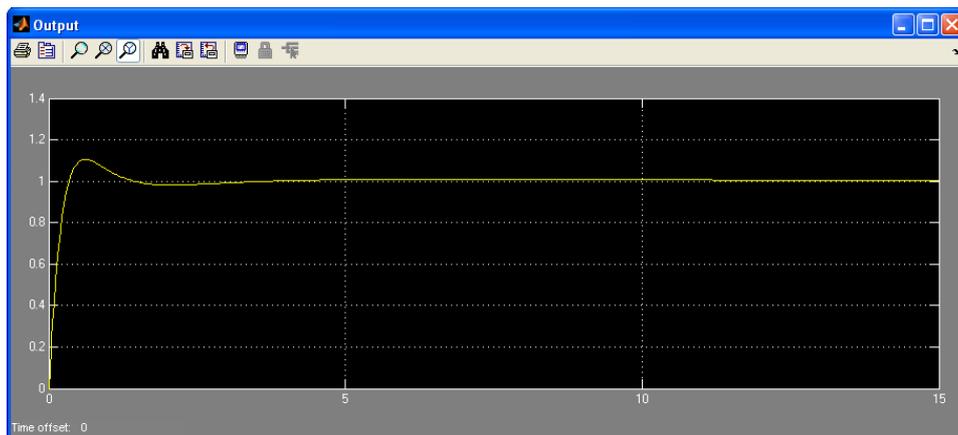


Figura 36: Esito della simulazione con  $K_P = 10$ ,  $K_D = 10$ ,  $K_I = 5$

Per ultima cosa faccio un paragone tra le configurazioni PID e PD. In confronto al segnale ottenuto con il controllore PD, il segnale ottenuto con il PID ha un picco meno pronunciato e raggiunge prima l'intervallo di tolleranza. Tutto ciò si paga però con una maggiore complessità del controllore.

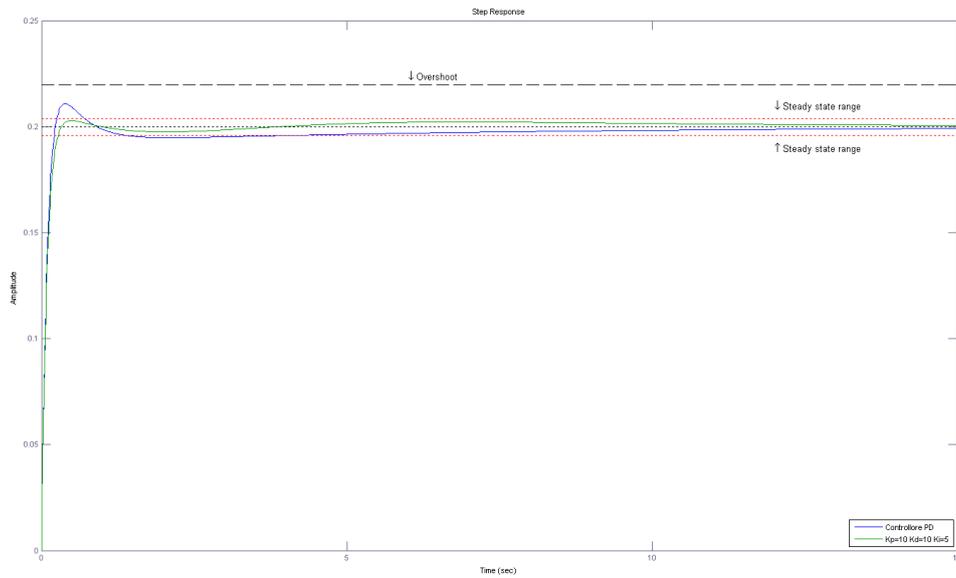


Figura 37: Confronto tra le risposte del sistema controllato da un PID(verde) e da un PD(blu)

## 4.5 CONTROLLORE PID - SOLUZIONE ALTERNATIVA

I valori trovati per  $K_P$ ,  $K_D$  e  $K_I$  in precedenza non sono gli unici possibili ovviamente. Cerco un'altra configurazione dei 3 parametri usando SISOTOOL. Dopo aver scelto lo schema di retroazione, inserisco la funzione del sistema e comincio a impostare il controllore PID.

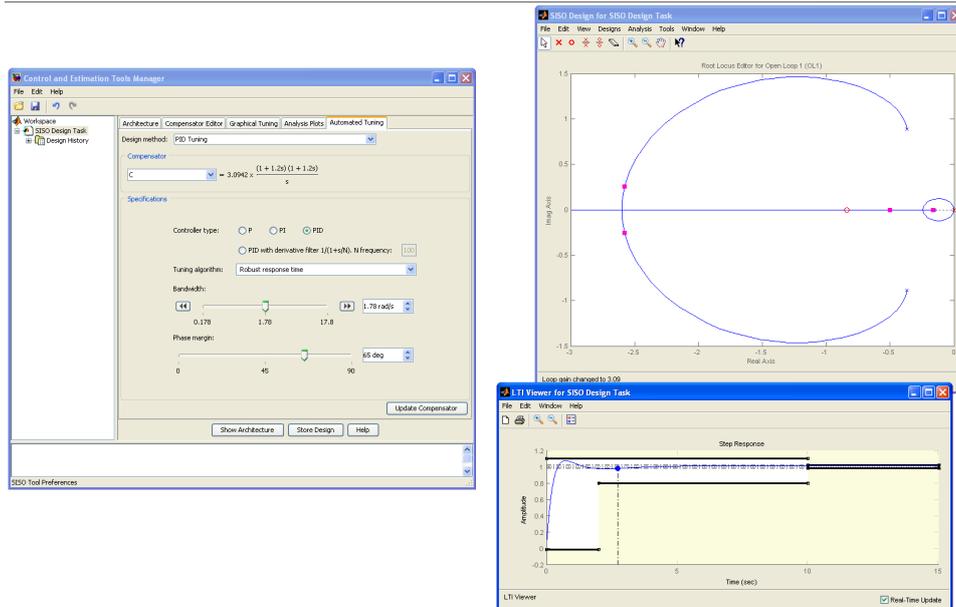


Figura 38: Schermata di SISOTOOL. A sinistra la schermata che permette la ricerca dei tre parametri. A destra il risultato sul sistema.

Aprò la finestra che mostra la risposta del sistema retro-azionato in tempo reale e imposto lì i vincoli di progetto, così da avere un riscontro grafico del lavoro. Uso il luogo delle radici per cercare un valore del guadagno che mi faccia rispettare i vincoli progettuali. Dopo vari tentativi, alla fine scelgo la seguente configurazione per  $C(s)$ :

$$\begin{aligned} C(s) &= 3.0942 \frac{(1 + 1.2s)(1 + 1.2s)}{s} = \\ &= \frac{4.455648s^2 + 742608s + 3.0942}{s} = \frac{K_D s^2 + K_P s + K_I}{s} \end{aligned}$$

E quindi:  $K_P = 7.42608$   $K_D = 4.455648$   $K_I = 3.0942$

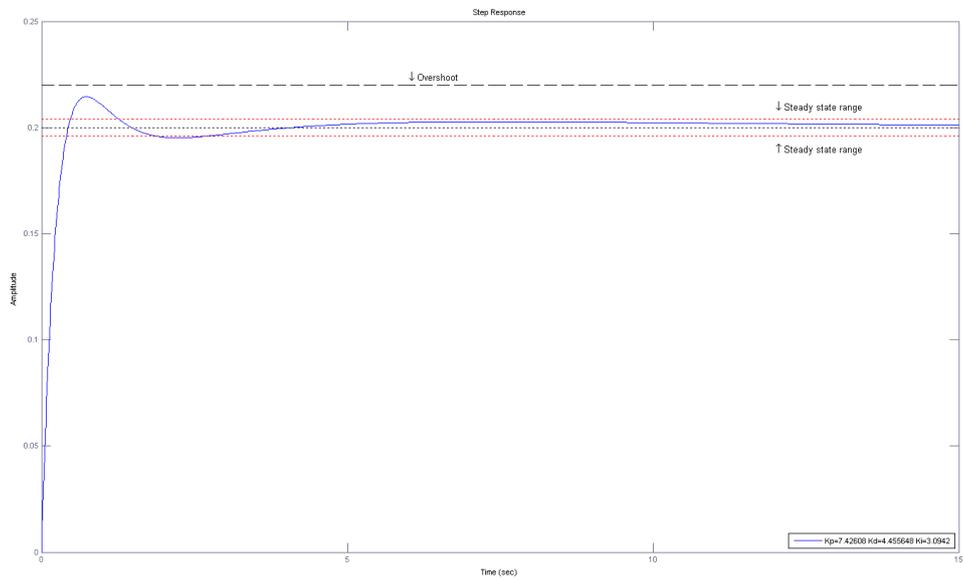


Figura 39: Risposta del sistema con il controllore PID ricavato tramite SISOTOOL

## 5 CONTROLLORE MEDIANTE LUOGO DELLE RADICI

Ecco la progettazione del controllore, eseguita mediante l'uso del luogo delle radici.

### 5.1 IL LUOGO DELLE RADICI E IL CALCOLO DEI PARAMETRI

I vincoli progettuali fanno sì che solo un sotto insieme dei poli rappresentati sul luogo delle radici vada bene. Per determinare tali valori ho prima calcolato 2 parametri necessari per la loro individuazione: la pulsazione naturale (natural frequency) e il coefficiente di smorzamento (damping ratio).

$$\omega_n \geq \frac{1.8}{T_r}$$
$$\xi \geq \sqrt{\frac{\left(\ln \frac{M_p}{\pi}\right)^2}{1 + \left(\ln \frac{M_p}{\pi}\right)^2}}$$

Legenda:

$\omega_n$  = Pulsazione naturale

$T_r$  = Tempo di salita

$\xi$  = Coeff. di smorzamento

$M_p$  = Valore massimo sovraelongazione

Dai valori presenti nelle specifiche risulta:

$$\omega_n \geq 0.9 \text{ e } \xi \geq 0.936$$

### 5.2 DISEGNO DEL LUOGO E AGGIUNTA DELLA RETE CORRETRICE

Disegno il luogo delle radici con Matlab:

```
Ts=10;
```

```
Tr=2;
```

```
Oversh=0.1;
```

```
de=0.2;
```

```
Mp=de+de*Oversh; %Maximum overshoot
```

```
Wn=WnCalc(Tr); %Natural frequency
```

```

z=zeta(Mp); %Damping ratio

num=[1.151 0.1774];
den=[1 0.739 0.921 0];
plant=tf(num,den);

rlocus (plant); %Luogo delle radici
sgrid (z,Wn); %Indicazione grafica dei vincoli
axis ([-1.5 0 -3 3])

```

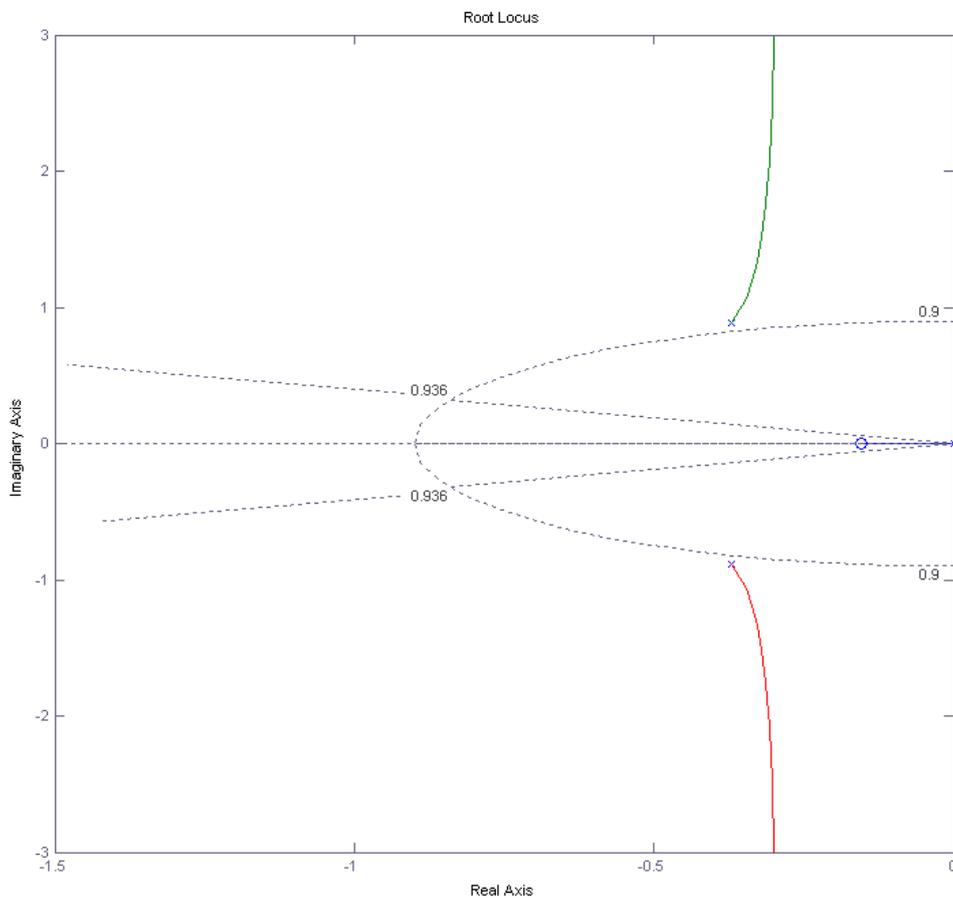


Figura 40: Luogo delle radici e rappresentazione grafica dei vincoli

La zona che viene indicata dai parametri è quella compresa tra le 2 rette e al di fuori del cerchio. Noto che nessuna porzione del luogo delle radici occupa tale area. E' necessario quindi spostare il grafico verso sinistra. Per fare ciò progetto una rete anticipatrice con SISOTOOL.

La rete anticipatrice è fatta nel seguente modo:

$$H(s) = K_C \frac{(s - z_0)}{(s - p_0)} \text{ con } |p_0| > |z_0|$$

Devo trovare dei valori per  $z_0$  e  $p_0$  affinché il luogo delle radici entri nell'area di interesse. Dopo aver inserito la FDT del sistema, attraverso la finestra grafica posiziono uno zero e un polo sul luogo delle radici e comincio a provare varie posizioni per essi. Nel muoverli, mi regolo guardando la risposta al gradino e il nuovo luogo delle radici, entrambi aggiornati in tempo reale. Nella finestra contenente il grafico della risposta del sistema, imposto da programma i vincoli sul tempo di assestamento e sull'overshoot, così da averli sempre sotto controllo.

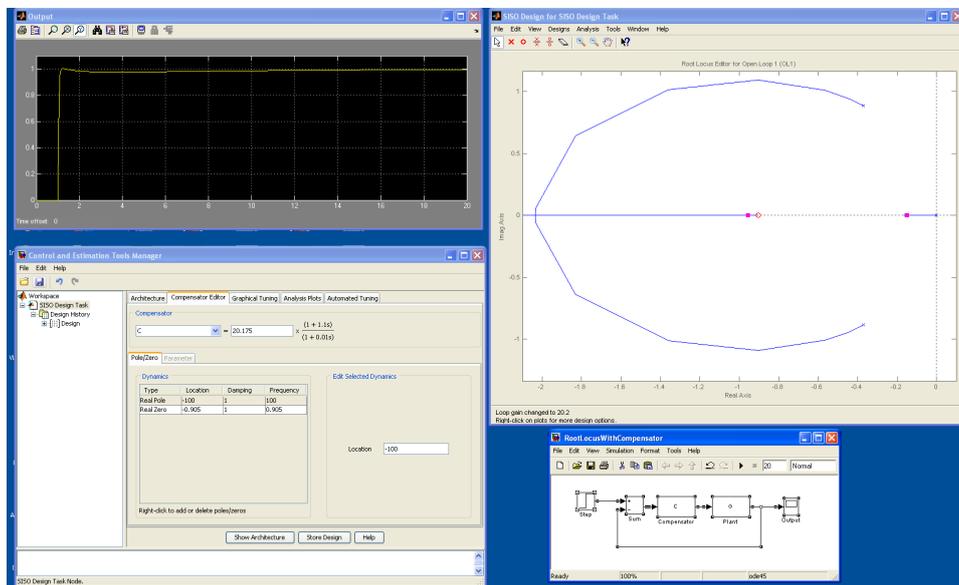


Figura 41: Schermate di simulink e SISOTOOL

Dopo varie prove trovo un valore di  $p_0 = -100$ ,  $z_0 = -0.905$  e  $K_C = 2229.282$ . Con tali valori, il sistema in catena chiusa rispetta i vincoli progettuali.

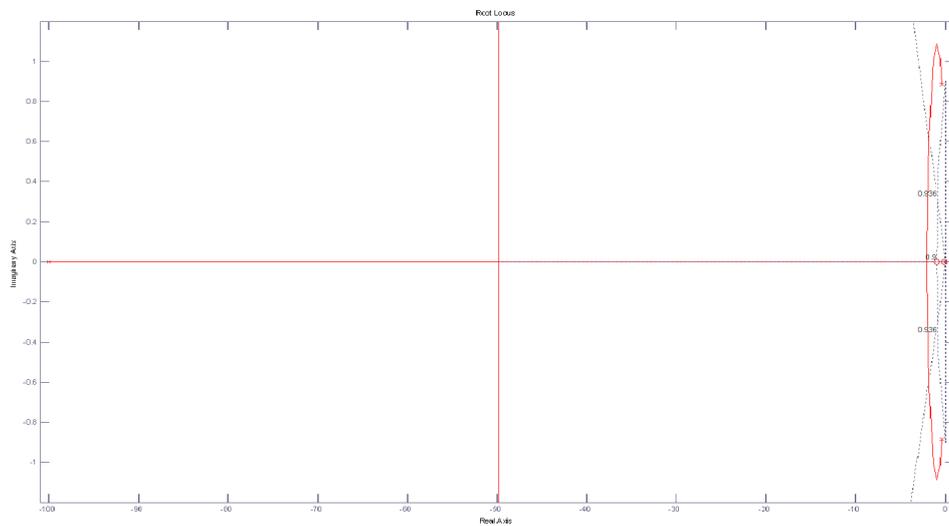


Figura 42: Nuovo luogo delle radici ottenuto con l'aggiunta della rete anticipatrice

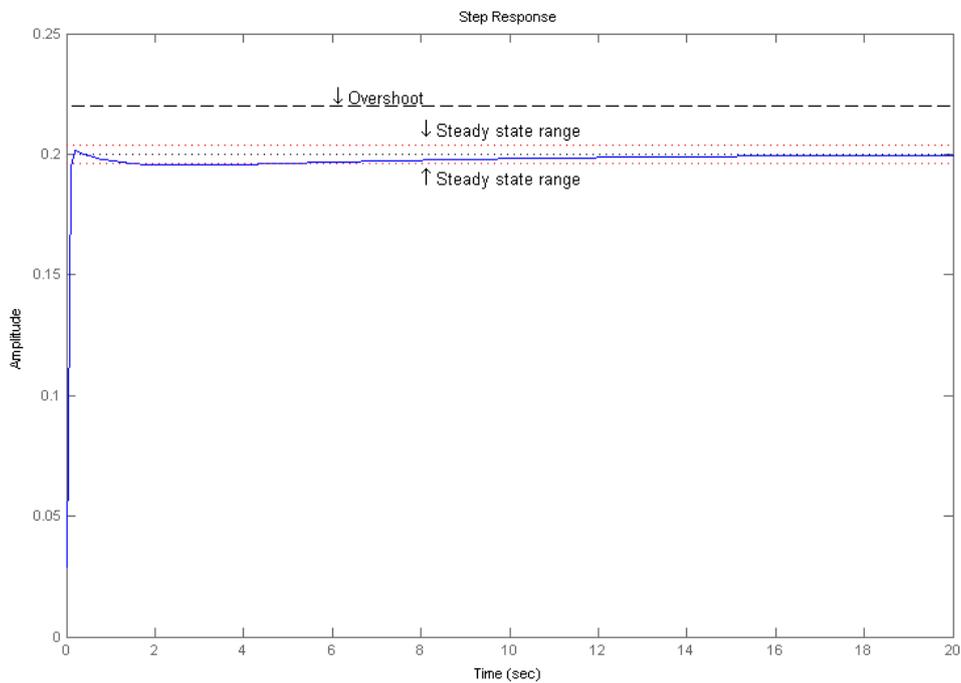


Figura 43: Risposta del sistema con i parametri trovati

## 6 CONTROLLORE CON RETI CORRETTTRICI (STUDIO IN FREQUENZA)

Eseguo ora lo studio del controllore nel dominio della frequenza, facendo uso di reti correttrici.

### 6.1 INTRODUZIONE E CALCOLO DEI PARAMETRI

Lo studio in frequenza fa uso dei diagrammi di Bode e di alcune grandezze ricavabili da essi. Per prima cosa calcolo i due parametri fondamentali: la pulsazione naturale e il margine di fase.

$$\omega_n = \frac{1.8}{T_r}$$

$$m_\phi = 100\xi$$

Legenda:

$\omega_n$  = Pulsazione naturale

$T_r$  = Tempo di salita

$\xi$  = Coeff. di smorzamento

$m_\phi$  = Margine di fase (in gradi)

Dai dati in mio possesso risulta :

$$\omega_n = 0.9 \text{ e } m_\phi = 52^\circ$$

Il sistema per essere stabile deve avere i valori dei 2 parametri maggiori di quelli appena calcolati.

### 6.2 SISTEMA IN CATENA APERTA E SISTEMA IN CATENA CHIUSA

Vediamo il sistema in catena aperta. Uso Matlab del disegnare la risposta al gradino e i diagrammi di Bode.

```
de = 0.2;  
t=0:0.1:30;  
num = [1.151 0.1774];  
den = [1 0.739 0.921 0];  
pitch = tf(num,den);  
figure(1);  
subplot(1,2,1);  
step(de*pitch,t);
```

```
subplot(1,2,2);
bode(pitch); %Disegno il diagramma di bode
```

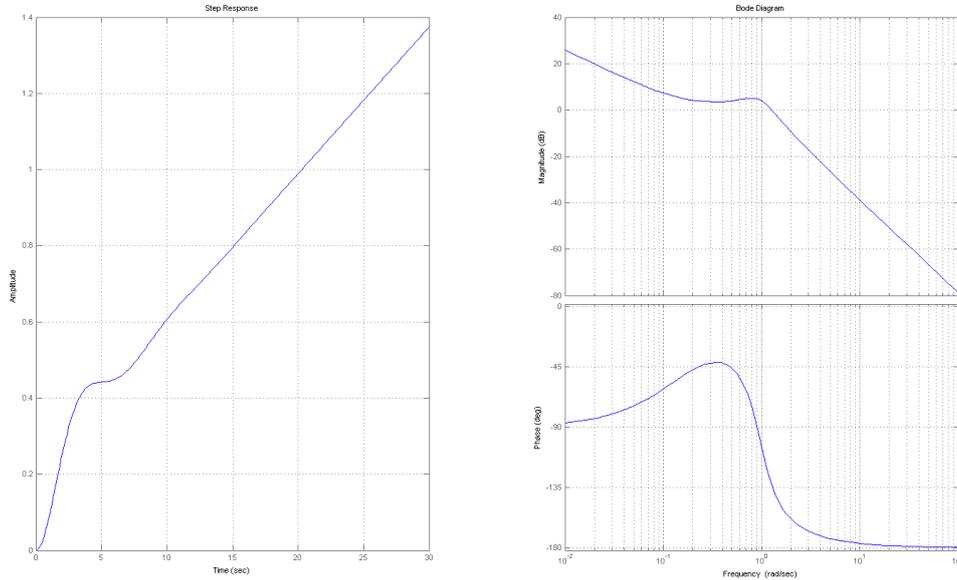


Figura 44: Risposta al gradino (a sinistra) e diagramma di Bode (a destra) del sistema in catena aperta

Come si evince dal grafico della risposta al gradino, il sistema è instabile e quindi non rispetta i vincoli.

Provo la configurazione retro-azionata. Anche in questo caso traccio il diagramma di Bode.

```
de = 0.2;
t=0:0.1:30;
num = [1.151 0.1774];
den = [1 0.739 0.921 0];
pitch = tf(num,den);
sys_cl = feedback(pitch,1);
figure(2);
subplot(1,2,1);
step(de*sys_cl,t);
subplot(1,2,2);
margin(sys_cl);
```

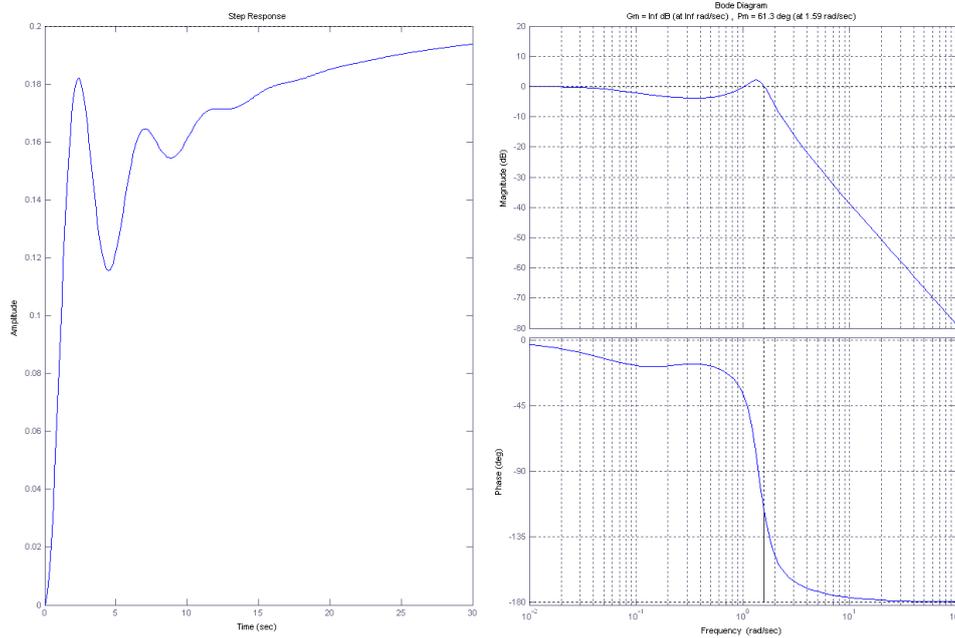


Figura 45: Risposta al gradino (a sinistra) e diagramma di Bode (a destra) del sistema in catena chiusa

La configurazione ad anello chiuso rispetta entrambi i vincoli di stabilità: la pulsazione di attraversamento è 1.59 e il margine di fase è 61.3°.

### 6.3 RETE CORRETRICE

Comincio allora la progettazione della rete correttiva. Per prima cosa ricavo i parametri di interesse da  $G(s)$ :

$$G(s) = \frac{1.151s + 0.1774}{s(s^2 + 0.739s + 0.921)} = \frac{0.1774 \left(1 + \frac{1.151}{0.1774}s\right)}{0.921s \left(1 + \frac{0.739}{0.921}s + \frac{s^2}{0.921}\right)} = \frac{K_G \bar{G}(s)}{s^{h_G}}$$

con:

$$K_G = 0.1774/0.921 \quad h_G = 1$$

$$\bar{G}(s) = \frac{\left(1 + \frac{1.151}{0.1774}s\right)}{\left(1 + \frac{0.739}{0.921}s + \frac{s^2}{0.921}\right)}$$

Nota che:

$$W(s) = C(s)G(s) = \frac{K_C K_G}{s^{h_C + h_G}} \bar{C}(s) \bar{G}(s) = \frac{K_W}{s^{h_W}} \widehat{W}(s)$$

con:

$$C(s) = \frac{K_C}{s^{h_C}} \bar{C}(s), \quad \bar{C}(0) = 1 \quad \widehat{W}(s) = \frac{K_C}{s^{h_C}} G(s)$$

$$K_W = K_C + K_G = \frac{1}{\epsilon} \quad h_W = h_C + h_G = 1 \text{ nel caso in esame}$$

Ottengo:

$$K_C = \frac{K_W}{K_G} = \frac{1/\epsilon}{K_G} = \frac{1/0.02}{0.921} = \frac{230250}{887} \cong 259.58 \quad h_C = 0$$

Traccio il diagramma di Bode di  $\widehat{W}(s) = \frac{K_C}{s^{h_C}} G(s)$ :

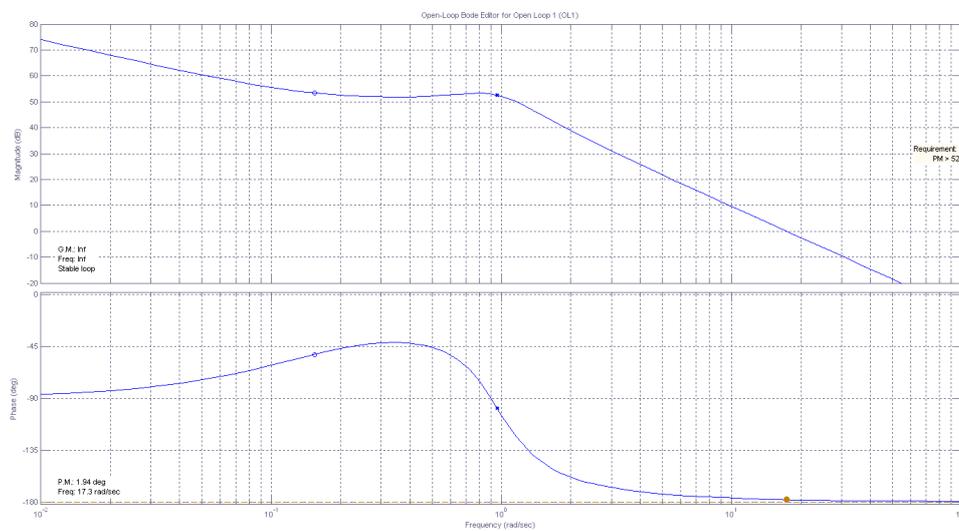


Figura 46: Diagramma di Bode della funzione  $\frac{K_C}{s^{h_C}} G(s)$

La pulsazione di attraversamento è maggiore della pulsazione naturale originale e il margine di fase è insufficiente. Ho quindi bisogno di una rete al sella. Uso gli strumenti di SISOTOOL per eseguire una progettazione grafica della disposizione dei poli e degli zeri. Il tutto rispettando sempre i vincoli di stabilità.

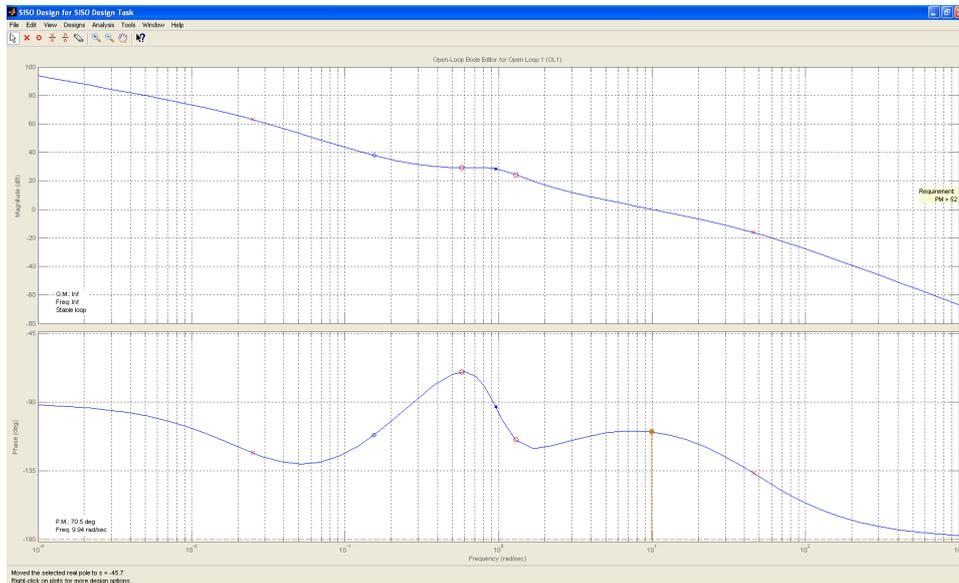


Figura 47: Finestra per la progettazione grafica in SISOTOOL. Sul diagramma di Bode si vedono l'aggiunta di 2 poli e 2 zeri, indicati con 2 cerchietti e 2 crocette

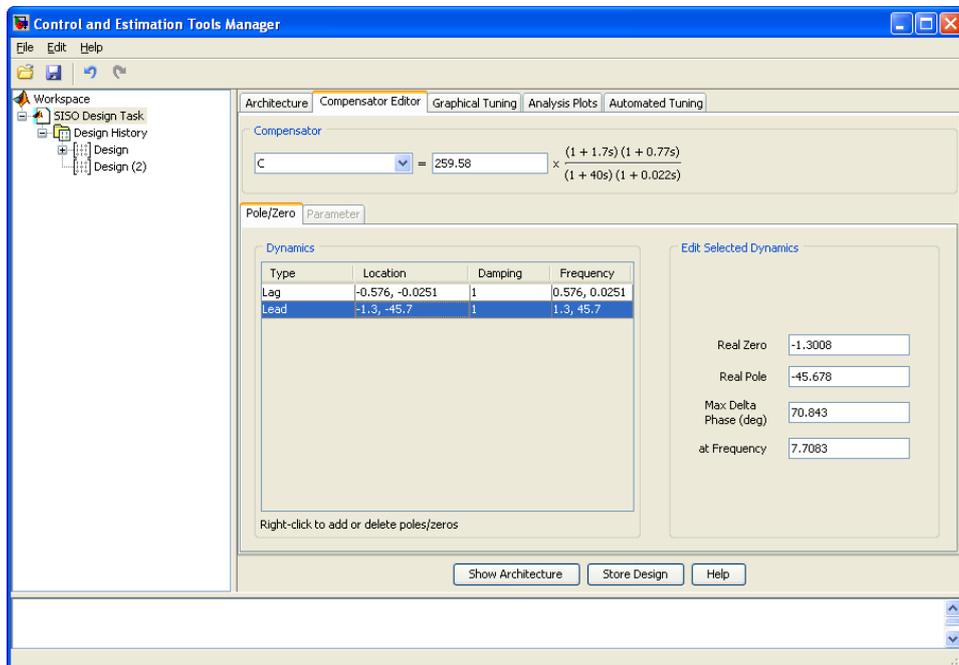


Figura 48: Finestra delle opzioni della rete correttiva. Da qui ricavo i valori degli zeri e dei poli inseriti.

La rete così ottenuta è:

$$C(s) = K_C \frac{(1 + 1.7s)(1 + 0.77s)}{(1 + 40s)(1 + 0.022s)}$$

$$z_1 = -\frac{10}{17} \cong -0.6 \quad z_2 = -\frac{100}{77} \cong -1.3 \quad p_1 = -0.025 \quad p_2 = -\frac{500}{11} \cong -45.45$$

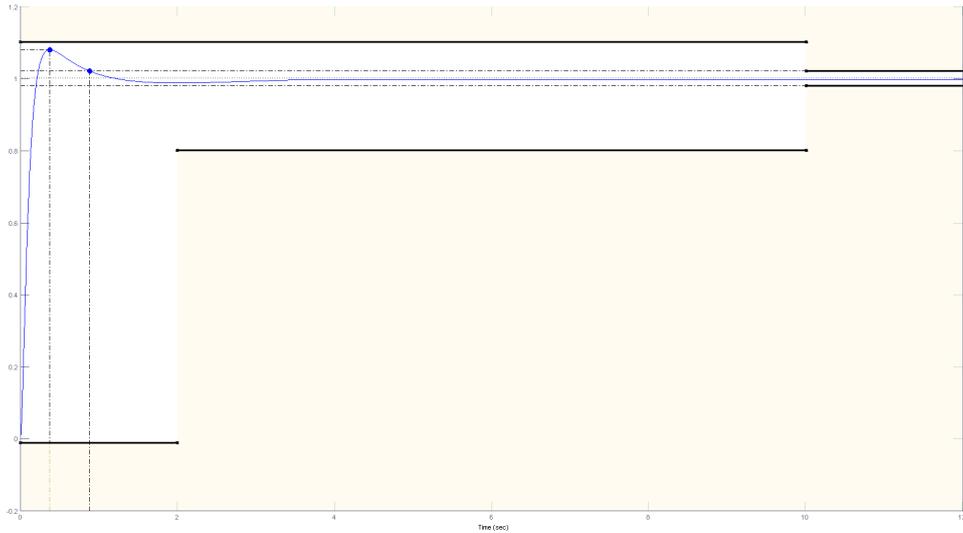


Figura 49: Risposta del sistema in catena chiusa con la presenza della rete a sella.

Come si evince dal grafico, il sistema così ottenuto rispetta tutti i vincoli progettuali.

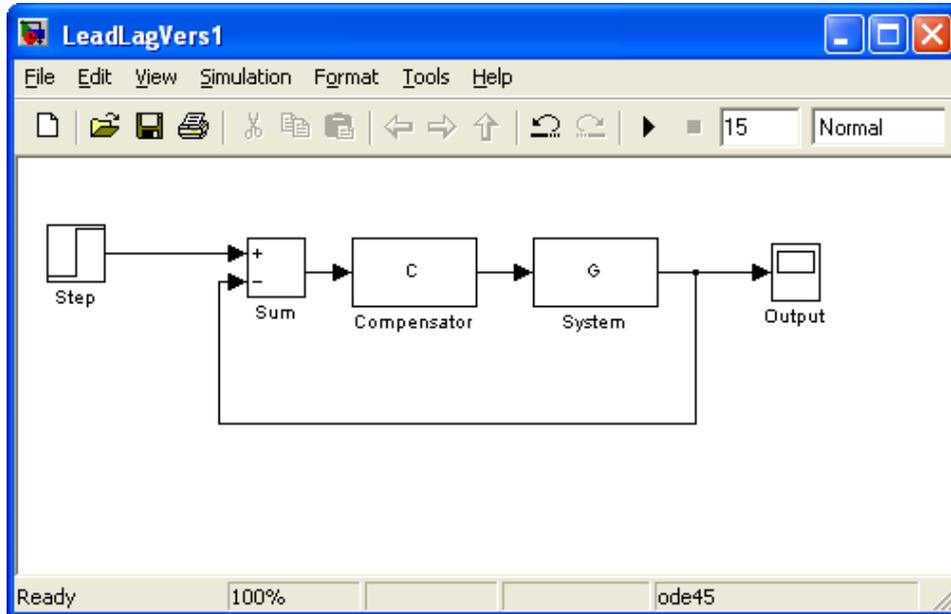


Figura 50: Schema a blocchi simulink contenente la rete correttiva.

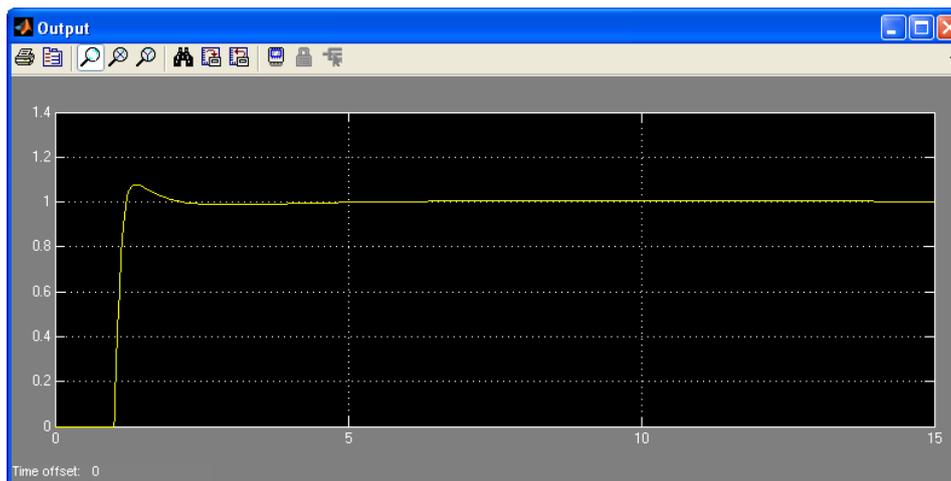


Figura 51: Risultato della simulazione con la rete a sella.

## 6.4 RETE CORRETRICE - SOLUZIONE ALTERNATIVA

Il controllore trovato non è l'unico ovviamente. Utilizzando ancora SISO-TOOL trovo un'altra rete valida.



Figura 52: Altra disposizione dei poli e degli zeri

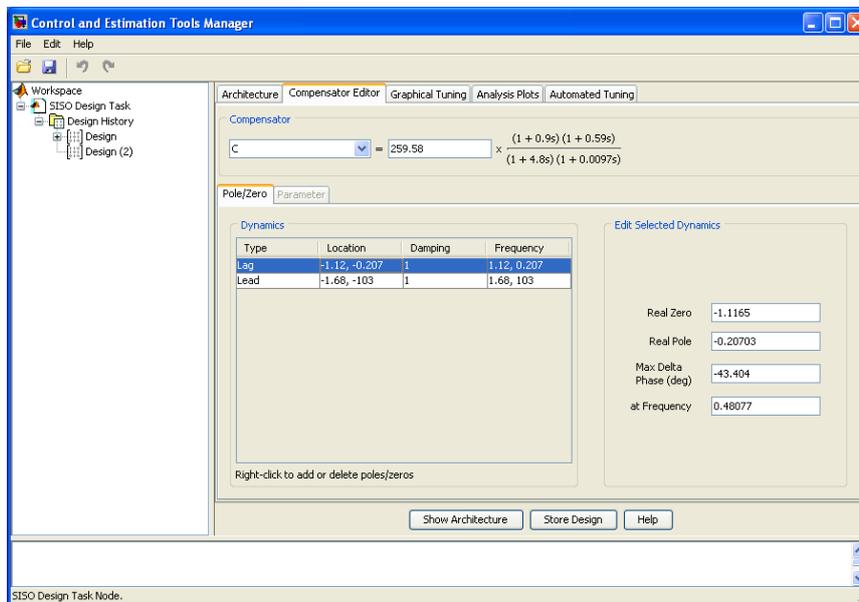


Figura 53: Nuova rete a sella ricavata

La rete a sella ora ottenuta è:

$$C(s) = K_C \frac{(1 + 0.9s)(1 + 0.59s)}{(1 + 4.8s)(1 + 0.0097s)}$$

$$z_1 = -\frac{10}{9} \cong -1.11 \quad z_2 = -\frac{100}{59} \cong -1.69 \quad p_1 = -\frac{5}{24} \cong -0.21 \quad p_2 = -\frac{10000}{97} \cong -103.09$$

Anche con questa configurazione le specifiche di controllo sono rispettate.

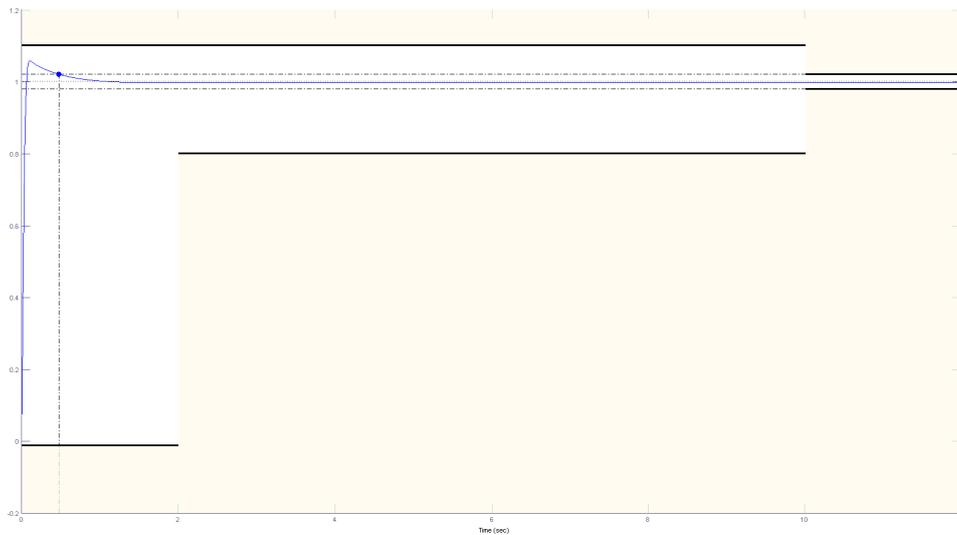


Figura 54: Risposta del sistema con la nuova rete correttiva

## 7 CONFRONTI E CONCLUSIONI

Espongo ora alcune considerazioni sul lavoro svolto, ponendo l'accento sulle diversità dei metodi e dei risultati ottenuti.

### 7.1 CONFRONTO TRA I METODI

Le tre strade adottate per ottenere il sistema di controllo mi hanno portato tutte ad ottenere un controllore che rispetta i vincoli di progetto. Come si evince dal lavoro però, le tre tecniche sono diverse per approccio e metodo.

L'uso di controllori PID nella pratica è molto frequente, soprattutto quando le informazioni sul sistema da controllare sono poche. Spesso si comincia con l'introduzione della parte proporzionale per ottenere stabilità e poi si aggiungono le altre parti (una o entrambe) per raggiungere i vincoli progettuali. Per ottenere i valori dei parametri, ho eseguito simulazioni successive ed affinato gli intervalli su cui eseguire i test. Questo modo di operare ha il vantaggio di essere semplice dal punto di vista concettuale. Di contro ha la necessità di dover eseguire iterazioni successive fino al raggiungimento degli obiettivi. Si può fare uso di strumenti software per la taratura dei parametri, come quelli messi a disposizione da SISOTOOL. Ho usato tale approccio per definire una versione alternativa del controllore PID. Nel mio lavoro, ho ottenuto un controllore nella forma PD che soddisfa le specifiche. Ho poi proseguito con l'analisi ed ottenuto anche una versione PID. Per un confronto tra questi vedi Figura 37 a pag. 32.

Il secondo metodo, quello del luogo delle radici, permette una progettazione che fa uso anche di grafici. Per prima cosa si calcolano dei parametri fondamentali (pulsazione naturale e coeff. di smorzamento). Successivamente si traccia il luogo delle radici e si usano i parametri appena trovati per delimitarne alcune aree. Infine si cerca un valore di  $K$  in tali aree che permetta di raggiungere i vincoli progettuali. Può accadere che il luogo delle radici non appartenga alle aree indicate dai parametri. In tal caso, si usa una rete correttiva per 'muovere' il grafico nella posizione di interesse. Nel mio lavoro ho usato Matlab per ricavare i parametri e disegnare il luogo delle radici. Ho poi fatto uso degli strumenti di SISOTOOL per ricavare le grandezze in gioco, il tutto tramite una interfaccia grafica. Questi strumenti hanno il vantaggio di mostrare in tempo reale gli effetti sul sistema. Inoltre permettono una 'progettazione grafica' del controllore.

L'ultimo metodo usato, le reti correttive, permette lo studio del sistema nel dominio della frequenza. Vengono introdotti nuovi parametri (come il margine di fase) e nuovi strumenti (come i diagrammi di Bode). Questo metodo prevede il calcolo preliminare di alcune grandezze e la successiva individuazione del tipo di rete necessaria. Capito la tipologia di rete da usare, si passa poi alla sua definizione. Si trova l'espressione della FDT (quindi tipo, guadagno, zeri e poli) basandosi su alcune regole della teoria

dei controlli. Anche questo approccio prevede l'uso di grafici come aiuto alla progettazione. Inoltre gli strumenti di SISOTOOL garantiscono un valido aiuto nella ricerca della configurazione migliore.

## 7.2 CONFRONTO TRA I RISULTATI

Per prima cosa riporto le espressioni dei vari controllori ottenuti:

- Controllore PD

$$C(s) = K_P + K_D s = 20 + 10s = 20(1 + 0.5s)$$

- Controllore PID

$$\begin{aligned} C(s) &= K_P + K_D s + K_I/s = 10 + 10s + 5/s = \\ &= \frac{10s^2 + 10s + 5}{s} = \frac{5}{s} (1 + 2s + 2s^2) \end{aligned}$$

- Controllore luogo delle radici

$$C(s) = K_C \frac{s - z_0}{s - p_0} = 2229.282 \frac{s + 0.905}{s + 100} = 20.18 \frac{1 + 1.11s}{1 + 0.01s}$$

- Controllore reti correttive

$$C(s) = 259.58 \frac{(1 + 1.7s)(1 + 0.77s)}{(1 + 40s)(1 + 0.022s)}$$

Noto subito molte differenze, dal numero di zeri e poli al valore del tipo. La più semplice risulta essere la FDT del controllore PD. (Per precisione, bisogna ricordare però che l'azione derivativa ideale non è realizzabile e quella ottenuta è una sua approssimazione). Il guadagno più elevato e il maggior numero di poli e zeri si ottiene invece dal controllore mediante rete correttiva. Dal punto di vista della complessità nel ricavarle, le FDT nell'ambito dei PID sono quelle che hanno richiesto maggiori risorse in termini di tempo. Basando la loro ricerca su tentativi e affinamenti iterati infatti, hanno richiesto numerose prove prima di individuare la configurazione riportata. Al contrario, i controllori ottenuti tramite luogo delle radici e rete correttiva hanno richiesto minori risorse temporali. Questo grazie agli strumenti di progettazione grafica da me usati. D'altra parte la taratura dei parametri, seppur eseguita attraverso un'interfaccia grafica, ha richiesto maggiore attenzione e sensibilità. In alcuni casi infatti, anche piccole variazioni delle variabili in gioco causavano modifiche significative alla risposta del sistema.

Studio ora le differenze dal punto di vista delle prestazioni ottenute. Per fare ciò disegno le risposte del sistema con i vari controllori trovati.

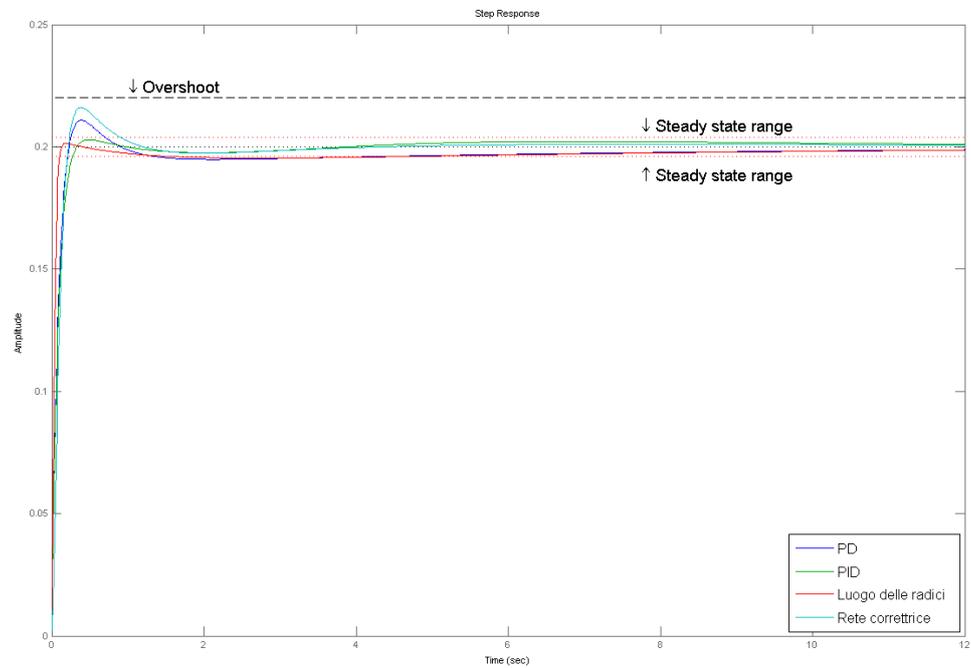


Figura 55: Risposta del sistema al gradino, usando i diversi controllori

Analizzo in dettaglio alcune aree. Per prima osservo l'area dove i segnali presentano dei picchi.

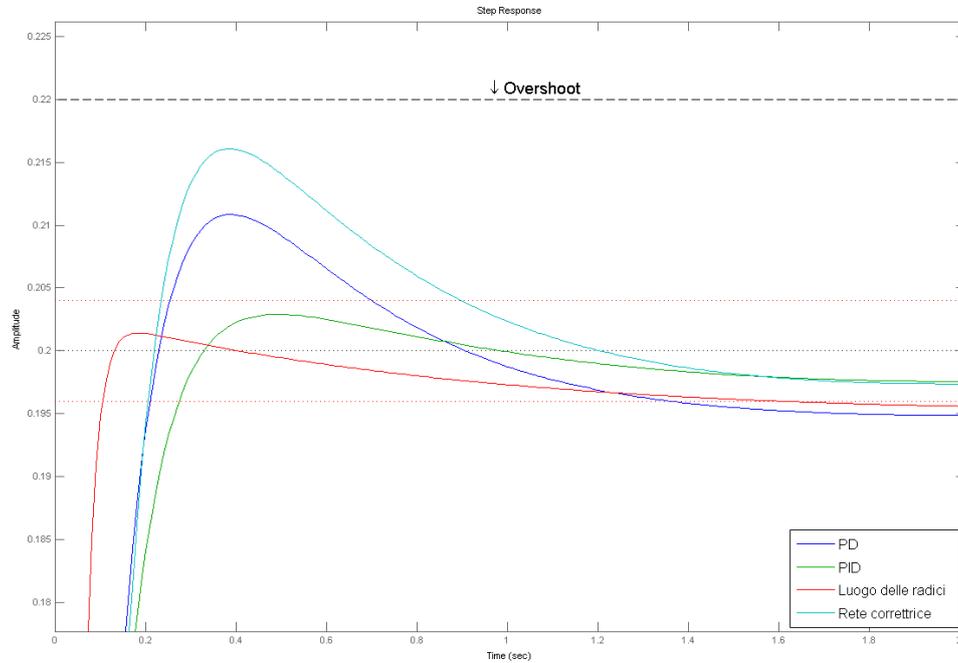


Figura 56: Zoom della zona di picco

Come già detto, tutti i controllori garantiscono la specifica sull'overshoot. Il controllore ricavato con il luogo delle radici presenta il picco meno pronunciato, mentre quello mediante rete correttiva ha il picco maggiore. I controllori nell'ambito dei PID si pongono tra i due. Quanto al tempo di salita, noto che tutti i quattro sistemi rispettano ampiamente il vincolo, che da specifiche è di 2 secondi. Il controllore ricavato con il luogo delle radici è il più veloce, mentre il più lento è il controllore PID.

Analizzo ora in dettaglio l'errore a regime e il tempo di assestamento.

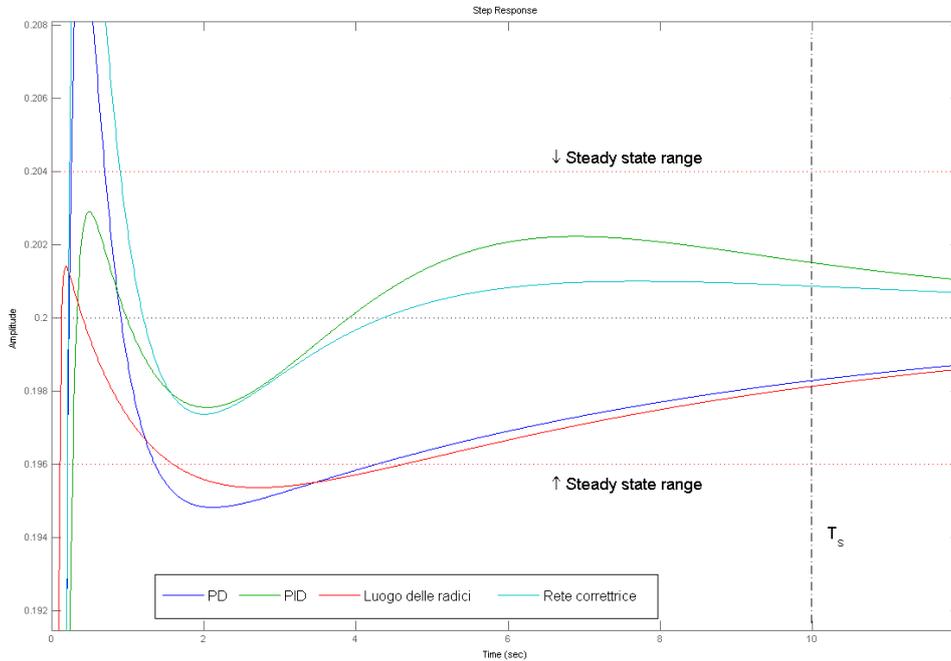


Figura 57: Zoom della zona di assestamento

Per tutti i controllori noto che l'errore a regime e il tempo di assestamento sono ampiamente soddisfatti. In particolare il controllore PID raggiunge l'intervallo stazionario in meno di mezzo secondo, mentre quello mediante rete correttiva in circa un secondo. Gli altri due in poco più di quattro secondi. I valori precisi dei parametri appena discussi sono riportati in tabella. Tali valori sono ricavati con SISOTOOL.

Controllore	$T_r (< 2sec)$	Overshoot ( $< 10\%$ )	$T_s (< 10sec)$
<b>PD</b>	0.15 sec	5.43 %	4.26 sec
<b>PID</b>	0.175 sec	1.45 %	0.273 sec
<b>Rete correttiva</b>	0.142 sec	8.05 %	0.898 sec
<b>Luogo delle radici</b>	0.0635 sec	0.705 %	4.61 sec

Tabella 1: Tabella riassuntiva delle prestazioni ottenute con i vari sistemi di controllo

Il controllore mediante luogo delle radici ha le migliori prestazioni in termini di tempo di salita ed overshoot. Risulta invece il peggiore quanto a tempo di assestamento. L'overshoot più pronunciato è quello del controllore mediante rete correttiva. Infine il controllore PID ha il valore di tempo di salita più elevato e il tempo di assestamento migliore.

## A FUNZIONI MATLAB

Questa appendice contiene le funzioni Matlab da me definite nel corso della progettazione.

- WnCalc.m:  
%Calcolo della pulsazione naturale,  
%dato il parametro Tr(Tempo di salita)  
function z=WnCalc(Tr)  
    z=(1.8)/Tr;
- zeta.m:  
%Calcolo del fattore di smorzamento,  
%dato il parametro Mp(Sovra-elongazione massima)  
function z=zeta(Mp)  
    z=sqrt( ((log(Mp/pi))^2) / ((1+(log(Mp/pi))^2)) );
- plotInformation.m:  
%Aggiunta al grafico della risposta al gradino  
%di informazioni utili quali  
%il limite di overshoot  
%e l'intervallo dell'errore a regime  
%Lista parametri:  
%de = Ampiezza del gradino  
%t = Vettore dei tempi, usato per disegnare le rette  
%n1 = Posizione del testo indicante  
%n2 = Posizione del testo indicante l'overshoot  
%er = Errore a regime  
%ov = Overshoot(%)  
function plotInformation(de,t,n1,n2,er,ov)  
    y1=(de+de\*er)\*ones(length(t));  
    y2=(de-de\*er)\*ones(length(t));  
    oversh=(de+de\*ov)\*ones(length(t));  
    plot(t,y1,':r',t,y2,':r');  
    text(n1,(de-de\*0.05),'\uparrow Steady state range',  
        'HorizontalAlignment','left');  
    text(n1,(de+de\*0.05),'\downarrow Steady state range',  
        'HorizontalAlignment','left');  
    hold on  
    plot(t,oversh,'--k');  
    text(n2,(de+de\*0.12),'\downarrow Overshoot',  
        'HorizontalAlignment','left');  
    hold on

## B BIBLIOGRAFIA

### Riferimenti bibliografici

- [1] SANDRO ZAMPIERI, *Dispensa di controlli automatici*, Padova, Edizioni Libreria Progetto Padova
- [2] <http://www.engin.umich.edu/class/ctms/index.htm>, Control Tutorial for Matlab and Simulink
- [3] [http://en.wikipedia.org/wiki/PID\\_controller](http://en.wikipedia.org/wiki/PID_controller), PID Controller
- [4] <http://www.capobianconicola.eu/Scilab/pid.pdf>, PID Controller
- [5] <http://www.aerospaceweb.org>, Aeronautica
- [6] [http://it.wikipedia.org/wiki/Piano\\_orizzontale\\_%28aeronautica%29](http://it.wikipedia.org/wiki/Piano_orizzontale_%28aeronautica%29), Aeronautica

## C FIGURE

### Elenco delle figure

1	Parametri di Volo . . . . .	3
2	Risposta al gradino del sistema in catena aperta . . . . .	6
3	Schema a blocchi in Simulink del sistema in catena aperta, con ingresso la rampa . . . . .	7
4	Input(in alto) e output(in basso) del sistema, con ingresso la rampa . . . . .	7
5	Schema a blocchi in Simulink del sistema in catena aperta, con input sinusoidale . . . . .	8
6	Input(in alto) e output(in basso) del sistema, con ingresso sinusoidale . . . . .	8
7	Risposta del sistema a diversi valori di $K_P$ . . . . .	10
8	Risposta del sistema con valori di $K_P$ scelti all'interno del- l'intervallo individuato in precedenza . . . . .	11
9	Risposta del sistema con $K_P = 2$ . . . . .	12
10	Risposte del sistema al variare di $K_P$ e $K_D$ . Ogni figura è relativa al valore di $K_P$ riportato nella didascalia. . . . .	15
11	Risposta del sistema per diversi valori di $K_P$ e $K_D$ , scelti negli intervalli sopra specificati. . . . .	16
12	Risposta del sistema con $K_P = 20$ e $K_D = 10$ . . . . .	17
13	Risposta del sistema con $K_P = 0.1$ e vari valori di $K_D$ e $K_I$ .	19
14	Risposta del sistema con $K_P = 0.2$ e vari valori di $K_D$ e $K_I$ .	19
15	Risposta del sistema con $K_P = 0.5$ e vari valori di $K_D$ e $K_I$ .	20
16	Risposta del sistema con $K_P = 1$ e vari valori di $K_D$ e $K_I$ . .	20
17	Risposta del sistema con $K_P = 2$ e vari valori di $K_D$ e $K_I$ . .	21
18	Risposta del sistema con $K_P = 5$ e vari valori di $K_D$ e $K_I$ . .	21
19	Risposta del sistema con $K_P = 10$ e vari valori di $K_D$ e $K_I$ .	22
20	Risposta del sistema con $K_P = 20$ e vari valori di $K_D$ e $K_I$ .	22
21	Risposta del sistema con $K_P = 50$ e vari valori di $K_D$ e $K_I$ .	23
22	Risposta del sistema con $K_P = 100$ e vari valori di $K_D$ e $K_I$ .	23
23	Risposta del sistema con $K_P = 0.1$ e vari valori di $K_D$ e $K_I$ , scelti negli intervalli sopra specificati. . . . .	25
24	Risposta del sistema con $K_P = 0.2$ e vari valori di $K_D$ e $K_I$ , scelti negli intervalli sopra specificati. . . . .	25
25	Risposta del sistema con $K_P = 0.5$ e vari valori di $K_D$ e $K_I$ , scelti negli intervalli sopra specificati. . . . .	26
26	Risposta del sistema con $K_P = 1$ e vari valori di $K_D$ e $K_I$ , scelti negli intervalli sopra specificati. . . . .	26
27	Risposta del sistema con $K_P = 2$ e vari valori di $K_D$ e $K_I$ , scelti negli intervalli sopra specificati. . . . .	27

28	Risposta del sistema con $K_P = 5$ e vari valori di $K_D$ e $K_I$ , scelti negli intervalli sopra specificati. . . . .	27
29	Risposta del sistema con $K_P = 10$ e vari valori di $K_D$ e $K_I$ , scelti negli intervalli sopra specificati. . . . .	28
30	Risposta del sistema con $K_P = 20$ e vari valori di $K_D$ e $K_I$ , scelti negli intervalli sopra specificati. . . . .	28
31	Risposta del sistema con $K_P = 50$ e vari valori di $K_D$ e $K_I$ , scelti negli intervalli sopra specificati. . . . .	29
32	Risposta del sistema con $K_P = 100$ e vari valori di $K_D$ e $K_I$ , scelti negli intervalli sopra specificati. . . . .	29
33	Risposta del sistema con $K_P = 10$ , $K_D = 10$ , $K_I = 5$ . . . . .	30
34	Schema simulink del sistema in catena chiusa con la presenza del controllore PID . . . . .	30
35	Pannello di configurazione del controllore PID . . . . .	31
36	Esito della simulazione con $K_P = 10$ , $K_D = 10$ , $K_I = 5$ . . . . .	31
37	Confronto tra le risposte del sistema controllato da un PID(verde) e da un PD(blu) . . . . .	32
38	Schermata di SISOTOOL. A sinistra la schermata che per- mette la ricerca dei tre parametri. A destra il risultato sul sistema. . . . .	33
39	Risposta del sistema con il controllore PID ricavato tramite SISOTOOL . . . . .	34
40	Luogo delle radici e rappresentazione grafica dei vincoli . . . . .	36
41	Schermate di simulink e SISOTOOL . . . . .	37
42	Nuovo luogo delle radici ottenuto con l'aggiunta della rete anticipatrice . . . . .	38
43	Risposta del sistema con i parametri trovati . . . . .	38
44	Risposta al gradino (a sinistra) e diagramma di Bode (a de- stra) del sistema in catena aperta . . . . .	40
45	Risposta al gradino (a sinistra) e diagramma di Bode (a de- stra) del sistema in catena chiusa . . . . .	41
46	Diagramma di Bode della funzione $\frac{K_C}{s^h C} G(s)$ . . . . .	42
47	Finestra per la progettazione grafica in SISOTOOL. Sul dia- gramma di Bode si vedono l'aggiunta di 2 poli e 2 zeri, indicati con 2 cerchietti e 2 crocette . . . . .	43
48	Finestra delle opzioni della rete corretttrice. Da qui ricavo i valori degli zeri e dei poli inseriti. . . . .	43
49	Risposta del sistema in catena chiusa con la presenza della rete a sella. . . . .	44
50	Schema a blocchi simulink contenente la rete corretttrice. . . . .	45
51	Risultato della simulazione con la rete a sella. . . . .	45
52	Altra disposizione dei poli e degli zeri . . . . .	46
53	Nuova rete a sella ricavata . . . . .	46
54	Risposta del sistema con la nuova rete corretttrice . . . . .	47

55	Risposta del sistema al gradino, usando i diversi controllori . . .	50
56	Zoom della zona di picco . . . . .	51
57	Zoom della zona di assestamento . . . . .	52

## D TABELLE

### Elenco delle tabelle

1	Tabella riassuntiva delle prestazioni ottenute con i vari sistemi di controllo . . . . .	52
---	--	----