

UNIVERSITÀ DEGLI STUDI DI PADOVA

Dipartimento di ingegneria dell'Informazione

Corso di laurea in Ingegneria dell'Informazione

Titolo tesi

*Progettazione e simulazione di un sistema di
“Cruise control” in ambiente Matlab/Simulink*

Relatore

Professor Alessandro Beghi

Laureando

Michele Lionello

Anno Accademico 2012/13

Indice

Introduzione	5
Analisi del sistema.....	6
1.1 - Descrizione fisica.....	6
1.2 - Equazioni del modello	6
1.3 - Parametri del modello	7
1.4 – Modello di stato.....	8
1.5 – Funzione di trasferimento	9
1.6 - Specifiche di progetto	10
Analisi in catena aperta.....	11
2.1 - Risposta al gradino.....	11
2.2 - Rappresentazione di zeri e poli.....	12
2.3 - Diagrammi di Bode.....	13
Progetto del controllore.....	14
3.1 - Introduzione ai controllori	14
3.2 – Controllori PID.....	15
3.2.1 - Controllore proporzionale P	16
3.2.2 – Controllore proporzionale-integrativo PI	18
3.2.3 – Controllore proporzionale-integrativo-derivativo PID	21
3.2.4 – Osservazioni conclusive sui controllori PID	23
3.3 – Analisi tramite luogo delle radici	23
3.3.1 - Introduzione della rete ritardatrice (phase LAG)	28
3.3.2 - Osservazioni conclusive sul luogo delle radici	31
3.4 - Analisi in frequenza	32
3.4.1 - Introduzione all’analisi in frequenza.....	32
3.4.2 - Considerazioni in catena aperta e catena chiusa	33
3.4.3 - Introduzione della rete ritardatrice	37
3.5 – Osservazioni conclusive sull’analisi in frequenza	41

Simulazione mediante simulink	42
4.1 - Introduzione all'ambiente Simulink	42
4.2 – Simulazione del sistema in catena aperta.....	42
4.3 - Creazione del sottosistema in catena aperta	44
4.4 – Simulazione del sistema in catena chiusa	45
4.4.1 - Simulazione con controllore PI.....	46
4.4.2 - Simulazione con rete ritardatrice	47
4.5 – Osservazioni conclusive sulla simulazione del sistema.....	49
Conclusioni	50
Bibliografia	51

Introduzione

Per *Cruise control* si intende il sistema elettronico presente in molte autovetture moderne che permette la regolazione automatica della velocità veicolo. Tale sistema permette al guidatore di impostare la velocità desiderata che, compatibilmente con le condizioni di assetto della vettura e con le condizioni esterne, verrà raggiunta e mantenuta durante la corsa.

Oggi giorno in commercio è possibile individuarne due diverse tipologie:

- cruise control;
- adaptive cruise control.

Il primo permette di raggiungere e mantenere la velocità impostata dall'utente, il quale la può variare solamente agendo su due pulsanti che, rispettivamente, consentono di aumentarla o diminuirla. In caso il guidatore desideri sorpassare un'altra vettura, gli è possibile agire sull'acceleratore modificando temporaneamente la velocità, che verrà in seguito riportata dal sistema al valore precedentemente impostato. Il secondo, diversamente dalla tipologia appena descritta, è in grado di diminuire automaticamente la velocità impostata nel caso in cui rilevi la presenza di ostacoli posti di fronte al mezzo entro una certa distanza minima. Tale distanza è impostata dal guidatore e viene mantenuta tramite l'ausilio di un radar situato anteriormente. In entrambe le tipologie il sistema viene disattivato in caso di pressione del freno, del freno a mano o della frizione o in presenza di guasti ai circuiti elettrici.

In questo elaborato considereremo un sistema di Cruise control della prima tipologia descritta e ci occuperemo inizialmente di modellizzare matematicamente tale sistema, per poi analizzarne la risposta al gradino in catena aperta e valutarne il comportamento dinamico. In seguito vedremo come progettare appositi controllori che migliorino l'andamento della risposta dinamica del sistema in catena chiusa.

Capitolo 1

Analisi del sistema

1.1 - Descrizione fisica

Come visto nell'introduzione, il cruise control è un perfetto esempio di sistema di controllo a retroazione ed il suo compito è mantenere la velocità del veicolo costante indipendentemente dalla presenza di disturbi esterni, come il vento o la pendenza della strada. Tale funzione è realizzata dal sistema misurando la velocità attuale del veicolo e comparandola con quella desiderata, detta di riferimento, regolando così l'acceleratore in accordo con una legge di controllo.



Figura 1.1: Modello semplificato del sistema

Consideriamo il modello semplificato della dinamica del veicolo illustrato in Figura 1.1. Il veicolo, di massa m , è mantenuto in movimento da una forza di controllo u , trasferita sulla strada tramite i pneumatici. Nel modello semplificato qui considerato assumiamo di poter controllare direttamente tale forza, trascurando le dinamiche delle componenti che contribuiscono a generarla. La forza resistiva bv , che ingloba la forza di attrito viscoso dell'aria e la forza di attrito tra pneumatici ed asfalto, agisce invece in direzione opposta a quella in cui si muove il veicolo e assumiamo che vari linearmente con la velocità del mezzo.

1.2 - Equazioni del modello

Dati questi assunti il modello semplificato considerato è rappresentabile da un sistema del primo ordine ed ora cercheremo di ricavare le equazioni che ne rappresentano la modellizzazione matematica. Sommando le forze nella direzione delle ascisse e applicando la seconda legge di Newton si ottiene la seguente equazione:

$$m\dot{v} + bv = u \quad (1.1)$$

Inoltre, essendo interessati al controllo della velocità del veicolo, l'equazione di uscita del sistema risulta la seguente:

$$y = v \quad (1.2)$$

Le due equazioni ottenute rappresentano un modello matematico in grado di descrivere un fenomeno dinamico e continuo. Inoltre lo descrivono in modo deterministico, dove l'andamento temporale della nostra variabile d'interesse è univocamente determinato dal valore che essa assume in un opportuno istante iniziale. La conoscenza delle condizioni iniziali dell'unica variabile di stato e del valore dell'ingresso u applicato al sistema permette la loro risoluzione.

1.3 - Parametri del modello

Prima di proseguire con l'analisi del nostro modello, dobbiamo assegnare dei valori ai parametri incogniti che compaiono nelle equazioni (1.1) e (1.2) che lo descrivono. In questo elaborato considereremo i seguenti valori:

Parametro	Simbolo	Valore
Massa del veicolo	m	1000 kg
Coefficiente di	b	50 N·s/m
Forza di controllo	u	500 N

Possiamo definire tali variabili in Matlab digitando nella Command Window i comandi:

```
m = 1000;    % Massa del veicolo
b = 50;      % Coefficiente di attrito
u = 500;     % Forza impressa
```

1.4 – Modello di stato

A partire dalle equazioni (1.1) e (1.2) ed essendo il sistema lineare possiamo ottenere facilmente il modello di stato:

$$\begin{cases} \dot{x} = [v] = \left[-\frac{b}{m}\right][v] + \left[\frac{1}{m}\right][u] \\ y = [1][v] \end{cases} \quad (1.3)$$

Esso è composto da una equazione differenziale del primo ordine e da una equazione di uscita ed è un caso particolare della rappresentazione generale di modello di stato lineare:

$$\begin{cases} \dot{x}(t) = F \cdot x(t) + G \cdot u(t) \\ y(t) = H \cdot x(t) + J \cdot u(t) \end{cases} \quad (1.4)$$

dove:

- F è una matrice quadrata di dimensioni nxn, con n numero di variabili di stato
- G è una matrice di dimensioni nxm, con m numero di ingressi
- H è una matrice di dimensioni pxn, con p numero di uscite
- J è una matrice di dimensioni pxm e nel nostro caso è nulla

Vediamo come ottenere in Matlab tale modello in spazio di stato. Per prima cosa definiamo le variabili necessarie:

```
F = -b/m;    % Coefficienti
G = 1/m;     % dello
H = 1;       % spazio di
J = 0;       % stato
```

A questo punto è sufficiente utilizzare l'istruzione Matlab:

```
space_state = ss(F, G, H, J); % Definizione del modello in spazio di stato
```


1.5 – Funzione di trasferimento

Consideriamo la soluzione dell'equazione differenziale che compare nel modello di stato (1.4). Si ottiene, dopo alcuni passaggi matematici:

$$x(t) = e^{F \cdot t} \cdot x(0) + \int_0^t e^{F(t-\tau)} * G \cdot u(\tau) d\tau$$

dove il primo termine rappresenta l'evoluzione libera del sistema, mentre il secondo termine ne rappresenta l'evoluzione forzata.

Sostituiamo ora tale espressione nell'equazione di uscita che compare nel sistema (1.4), ottenendo:

$$y(t) = H \cdot e^{F \cdot t} \cdot x(0) + \int_0^t [H \cdot e^{F(t-\tau)} \cdot G + J \cdot \delta(t - \tau)] * u(\tau) d\tau$$

L'integrale che compare nell'espressione ottenuta rappresenta la convoluzione tra la risposta impulsiva

$$w(t) = H \cdot e^{F \cdot t} \cdot G + J \cdot \delta(t) \quad (1.5)$$

e l'ingresso $u(t)$.

Applichiamo ora la trasformata di Laplace alla seconda equazione del sistema (1.4):

$$Y(s) = HX(s) + JU(s)$$

e ricordando che

$$X(s) = (sI - F)^{-1}x(0) + (sI - F)^{-1}G U(s)$$

si ottiene:

$$Y(s) = H(sI - F)^{-1}x(0) + [H(sI - F)^{-1}G + J]U(s)$$

Definiamo a questo punto la funzione di trasferimento del sistema:

$$W(s) := H(sI - F)^{-1}G + J = \mathcal{L}[H \cdot e^{F \cdot t} \cdot G + J \cdot \delta(t)] = \mathcal{L}[w(t)] \quad (1.6)$$

Sostituendo nell'equazione (1.6) i valori di F, G, H e J del nostro esempio, ricavati nel paragrafo precedente, si ottiene la funzione di trasferimento:

$$W(s) = 1 \cdot \left(s - \left(-\frac{b}{m} \right) \right)^{-1} \cdot \frac{1}{m} + 0 = \frac{1}{m \left(s + \frac{b}{m} \right)} = \frac{1}{ms + b} \quad (1.7)$$

Un metodo più semplice per ottenere la funzione di trasferimento del nostro sistema è applicare la trasformata di Laplace alla prima equazione del sistema (1.3), assumendo condizioni iniziali nulle. Dopo alcuni passaggi algebrici si ottiene:

$$U(s) = msV(s) + bV(s)$$

da cui:

$$V(s) = \frac{U(s)}{ms + b}$$

La funzione di trasferimento che si ottiene è proprio la (1.7), infatti:

$$W(s) = \frac{V(s)}{U(s)} = \frac{1}{ms+b}$$

In Matlab la possiamo definire tramite i comandi:

```
s = tf('s');           % Definizione della variabile di Laplace
P_Cruise = 1/(m*s + b); % Definizione della funzione di trasferimento
```

1.6 - Specifiche di progetto

Definiamo ora le specifiche progettuali che vogliamo essere rispettate dal sistema.

Desideriamo che il nostro mezzo sia in grado di raggiungere, con una forza fornita dal motore di 500 N, la velocità di 10 m/s con un tempo di salita inferiore a 5s; per tali valori possiamo tollerare, per la velocità del veicolo, una sovraelongazione massima del 10% e un errore a regime massimo del 2%.

Le specifiche di progetto per questo problema sono quindi le seguenti:

- Tempo di salita < 5s
- Errore a regime < 2%
- Sovraelongazione < 10%

Capitolo 2

Analisi in catena aperta

2.1 - Risposta al gradino

Per valutare il comportamento del nostro sistema in catena aperta, visualizziamo la risposta forzata in corrispondenza del gradino di ampiezza 500 N. Ricordiamo che in catena aperta, se l'uscita non è quella desiderata, il sistema non è in grado di modificare l'ingresso del sistema per ridurre l'errore e quindi la correttezza dell'uscita è garantita solamente dalla precisa costruzione del modello matematico. Per ottenere la risposta al gradino ricorriamo ai comandi Matlab:

```
m = 100;
```

```
b = 50;
```

```
u = 500;
```

```
s = tf('s');
```

```
P_Cruise = 1/(m*s + b);
```

```
step(u*P_Cruise); % visualizza la risposta al gradino in corrispondenza dell'ingresso u
```

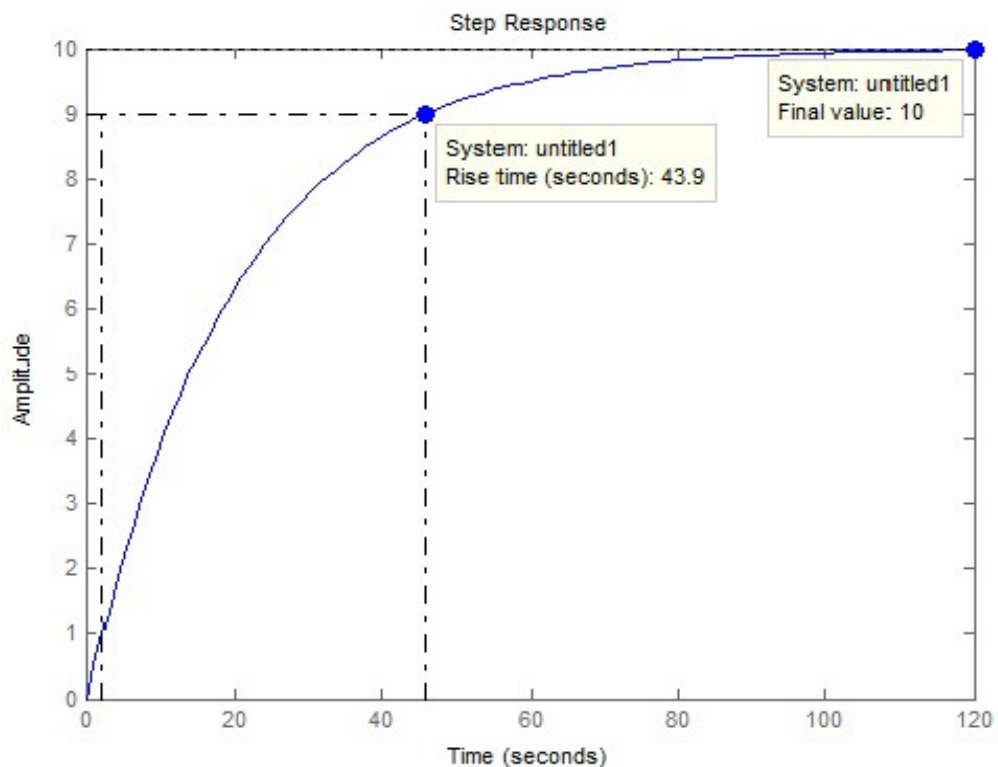


Figura 2.1: Risposta al gradino in catena aperta

Si osserva che il sistema in catena aperta raggiunge la velocità a regime desiderata di 10 m/s e non presenta alcuna sovraelongazione ed/od oscillazione. Tuttavia, il tempo di salita è eccessivamente lungo, quindi abbiamo bisogno di progettare un controllore in retroazione che velocizzi significativamente la risposta al gradino, senza però peggiorare le altre caratteristiche dinamiche del sistema.

2.2 - Rappresentazione di zeri e poli

Se consideriamo la funzione di trasferimento (1.7) del nostro sistema, è immediato verificare che l'unico suo polo è situato in $s = -\frac{b}{m}$.

Possiamo rappresentare graficamente tale polo, con i valori dei parametri considerati in questo elaborato, utilizzando i comandi Matlab:

```
pzmap(P_Cruise);    % visualizza sul piano complesso zeri e poli  
axis([-1 1 -1 1]);  % definisce l'intervallo di valori di asse reale e immaginario  
ottenendo:
```

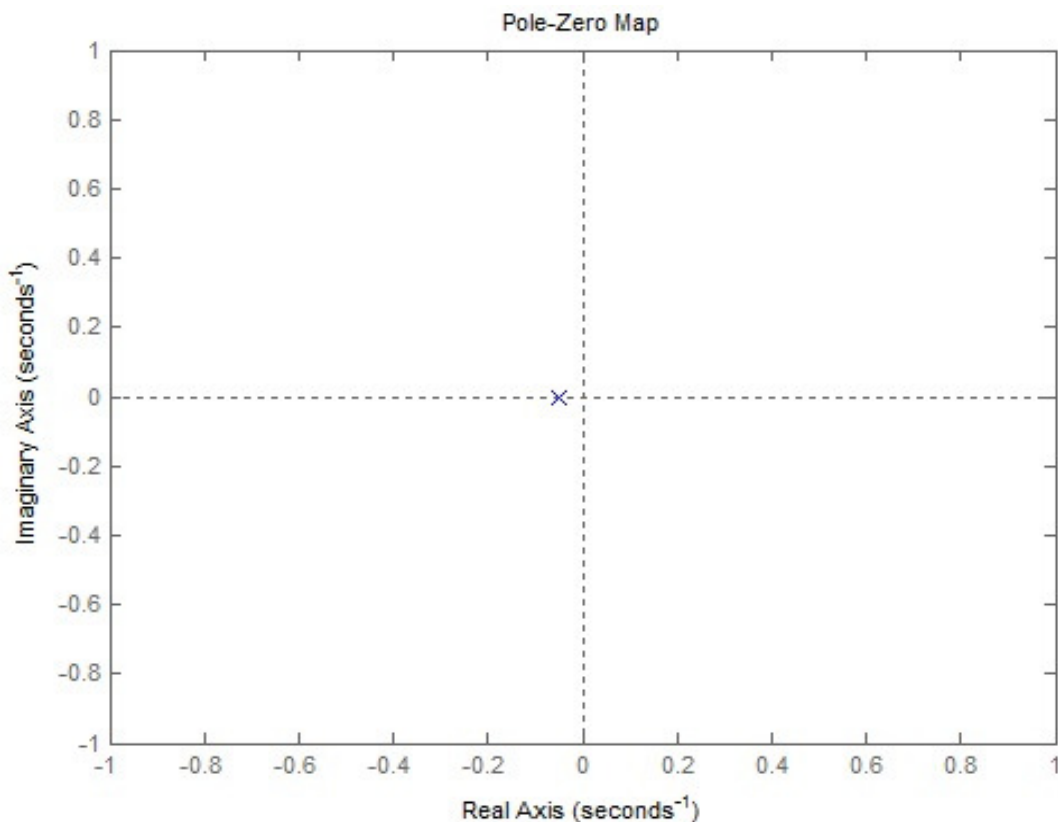


Figura 2.2: Rappresentazione di zeri e poli nel piano complesso

Si osserva che l'unico polo del sistema in catena aperta è reale e negativo, quindi siamo in presenza di BIBO-stabilità e di un carattere non oscillatorio. Inoltre, dato che la rapidità della risposta è determinata dal modulo di tale polo, ovvero $\frac{b}{m}$, si evince dalla Figura 2.2 che il nostro sistema risulta particolarmente lento. Non potendo modificare i parametri del sistema per variare la posizione del polo, dobbiamo progettare un opportuno controllore che alteri la posizione di zeri e poli del sistema in modo da ottenere le performance richieste.

2.3 - Diagrammi di Bode

Consideriamo ora la risposta in frequenza del sistema in catena aperta. Ne tracciamo i diagrammi di Bode tramite il comando di Matlab:

```
bode(P_Cruise);    % visualizza i diagrammi di Bode
```

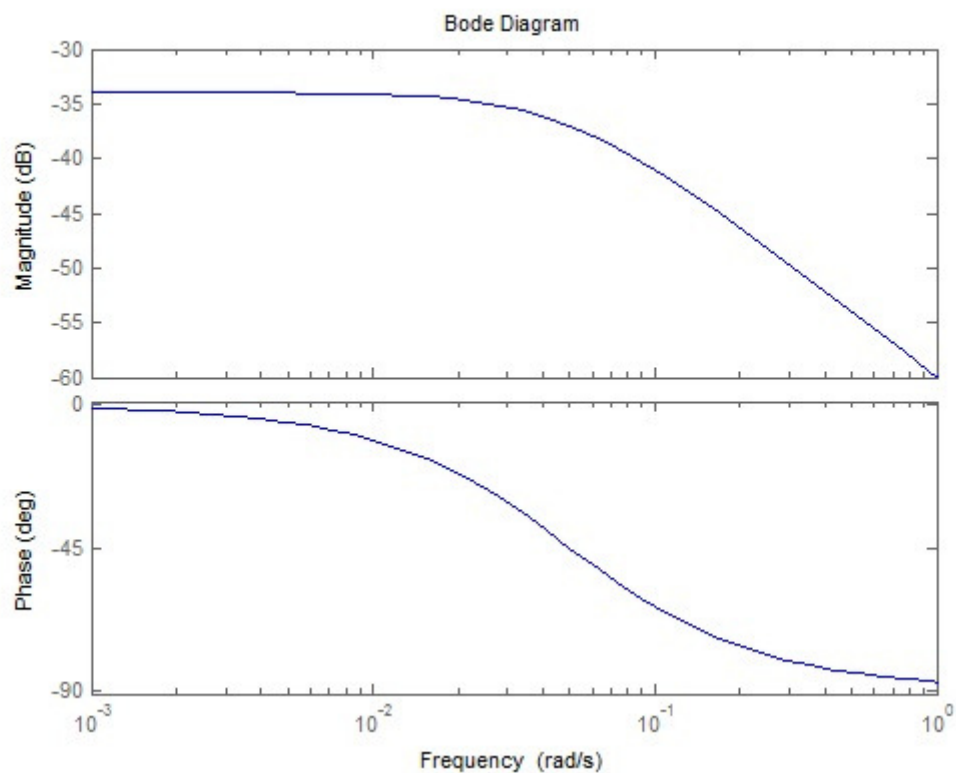


Figura 2.3: Diagramma di Bode della risposta in frequenza

I diagrammi di Bode ottenuti esibiscono il comportamento tipico di un sistema del primo ordine, presentando modulo di -3dB e fase di -45° in corrispondenza della pulsazione $\omega = \frac{b}{m} = 0.05$ rad/s e una pendenza del diagramma del modulo di -20 dB/dec in alta frequenza, dovuta al contributo del polo.

Capitolo 3

Progetto del controllore

3.1 - Introduzione ai controllori

Consideriamo ora il nostro sistema con retroazione unitaria negativa, il cui schema è riportato nella figura seguente:

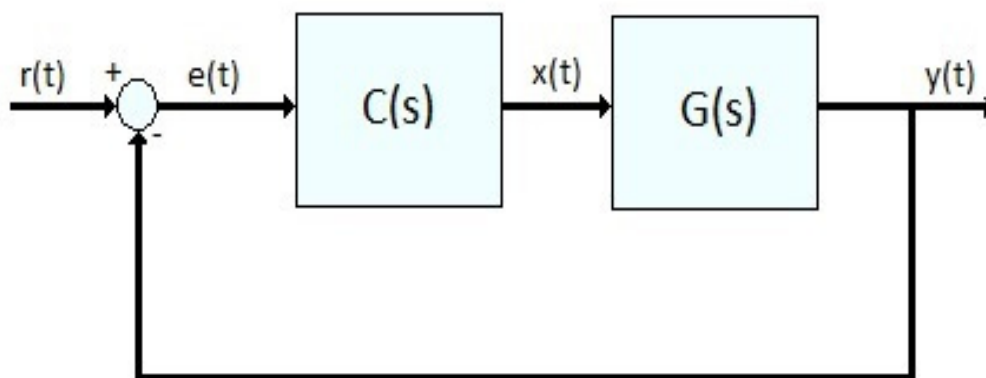


Figura 3.1: Schema di controllo in retroazione unitaria negativa ad un grado di libertà

dove:

- $r(t)$ rappresenta l'ingresso di riferimento
- $y(t)$ rappresenta il segnale di uscita
- $e(t)$ rappresenta l'errore, definito come la differenza $r(t) - y(t)$ tra l'ingresso di riferimento e il segnale di uscita, istante per istante

Al nostro processo noto $G(s)$, che rappresenta la funzione di trasferimento in catena aperta, è stato aggiunto un ulteriore blocco $C(s)$, detto compensatore, che può essere modellizzato a piacimento dal progettista per modificare le caratteristiche del sistema con lo scopo di migliorarne le prestazioni. Si dice che l'aggiunta del compensatore ha reso il sistema ad un grado di libertà.

Con l'aggiunta del compensatore la funzione di trasferimento del sistema in catena aperta diventa:

$$W(s) = C(s)G(s) \quad (3.1)$$

Vediamo ora di ricavare la rispettiva funzione di trasferimento del sistema ad anello chiuso, ricorrendo alla trasformata di Laplace:

$$E(s) = R(s) - Y(s)$$

e

$$Y(s) = C(s)G(s)E(s)$$

Unendo le due equazioni si ottiene:

$$Y(s) = C(s)G(s)(R(s) - Y(s))$$

da cui si deriva facilmente la funzione di trasferimento del sistema retroazionato:

$$H(s) := \frac{C(s)G(s)}{1+C(s)G(s)} \quad (3.2)$$

Osservando la struttura della funzione di trasferimento ottenuta si deduce che l'aggiunta del controllore $C(s)$ permette di modificare il guadagno di Bode e di introdurre ulteriori zeri e/o poli al processo.

3.2 – Controllori PID

I controllori denominati PID realizzano al loro ingresso la combinazione lineare di tre azioni: un'azione proporzionale (P), un'azione integrativa (I) ed un'azione derivativa (D).

La funzione di trasferimento di un controllore PID è dunque nella forma:

$$C(s) = k_p + \frac{k_i}{s} + k_d \cdot s = \frac{k_d \cdot s^2 + k_p \cdot s + k_i}{s} \quad (3.3)$$

con k_p , k_d e k_i parametri reali.

Definiamo tale controllore in Matlab tramite i comandi:

```
Kp = 1;
```

```
Ki = 1;
```

```
Kd = 1;
```

```
s = tf('s');
```

```
C = Kp + Ki/s + Kd*s
```

oppure, in alternativa, utilizziamo l'oggetto "pid" presente nelle librerie di Matlab:

```
C = pid(Kp, Ki, Kd)
```

I due controllori che si ottengono nei due differenti modi presentati sono del tutto equivalenti tra loro.

3.2.1 - Controllore proporzionale P

Il sistema ottenuto aggiungendo un controllore proporzionale può essere visto come in Figura 3.1, dove l'espressione del compensatore $C(s)$ può essere ottenuta ponendo $k_d = 0$ e $k_i = 0$ nella (3.3):

$$C(s) = k_p \quad (3.4)$$

Sostituendo tale espressione di $C(s)$ nell'equazione (3.2) otteniamo la funzione di trasferimento del sistema retroazionato:

$$H(s) = \frac{G(s) \cdot C(s)}{1 + G(s) \cdot C(s)} = \frac{\frac{k_p}{m \cdot s + b}}{1 + \frac{k_p}{m \cdot s + b}} = \frac{\frac{k_p}{m \cdot s + b}}{\frac{m \cdot s + b + k_p}{m \cdot s + b}} = \frac{k_p}{m \cdot s + b + k_p} \quad (3.5)$$

Proviamo a porre la costante $k_p = 100$ e vediamo come risulta la risposta al gradino corrispondente. Creiamo un nuovo file-m in Matlab contenente i comandi:

```
m = 1000;
b = 50;
r = 10;
s = tf('s');
P_Cruise = 1 / (m*s + b);
Kp = 100;
C = pid(Kp);
T = feedback(C*P_Cruise, 1); % istruzione per ottenere la f.d.t. in catena chiusa
t = 0:0.1:20;
step(r*T,t);
axis([0 20 0 10]);
```


Eseguendo tale file-m si ottiene la seguente risposta al gradino:

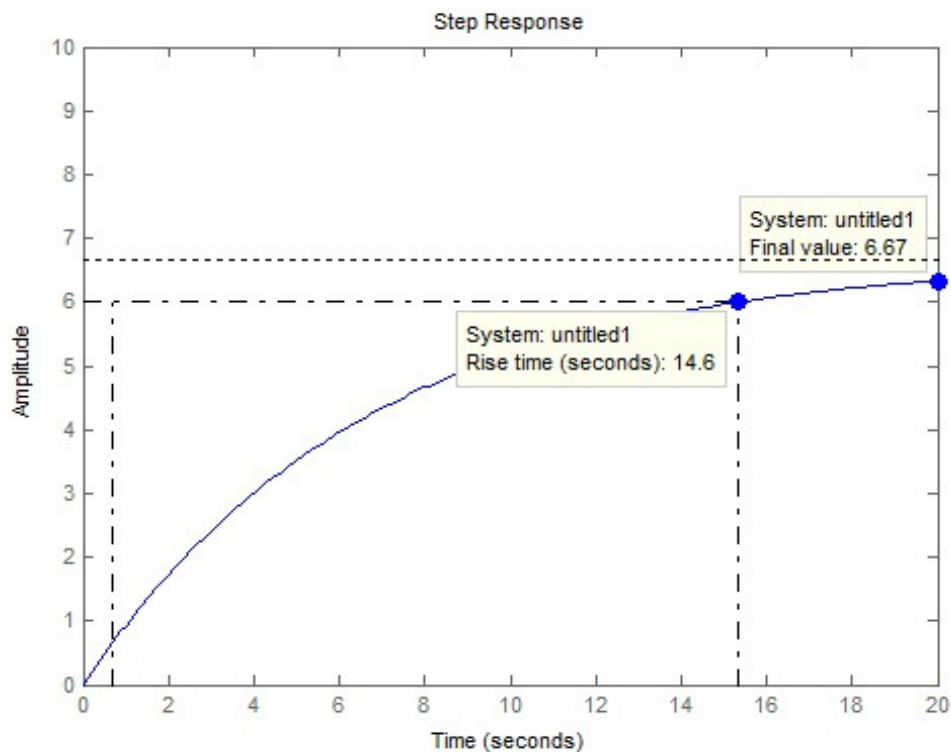


Figura 3.2: Risposta al gradino con $k_p = 100$

Si vede chiaramente che né il tempo di salita né l'errore a regime soddisfano le specifiche richieste. Il valore a regime è infatti nettamente inferiore a quello di riferimento, ovvero 10m/s, e viene raggiunto dopo un tempo di salita di oltre 20s, ben superiore ai 5s che ci eravamo prefissati di raggiungere.

Proviamo allora ad aumentare il guadagno proporzionale k_p per ridurre sia il tempo di salita che l'errore a regime. Modifichiamo il file-m precedentemente creato ponendo ad esempio $k_p = 5000$:

```
Kp = 5000;  
C = pid(Kp);  
T = feedback(C*P_Cruise, 1);  
step(r*T,t);  
axis([0 20 0 12]);
```

L'esecuzione del file-m così modificato porta alla risposta al gradino qui riportata:

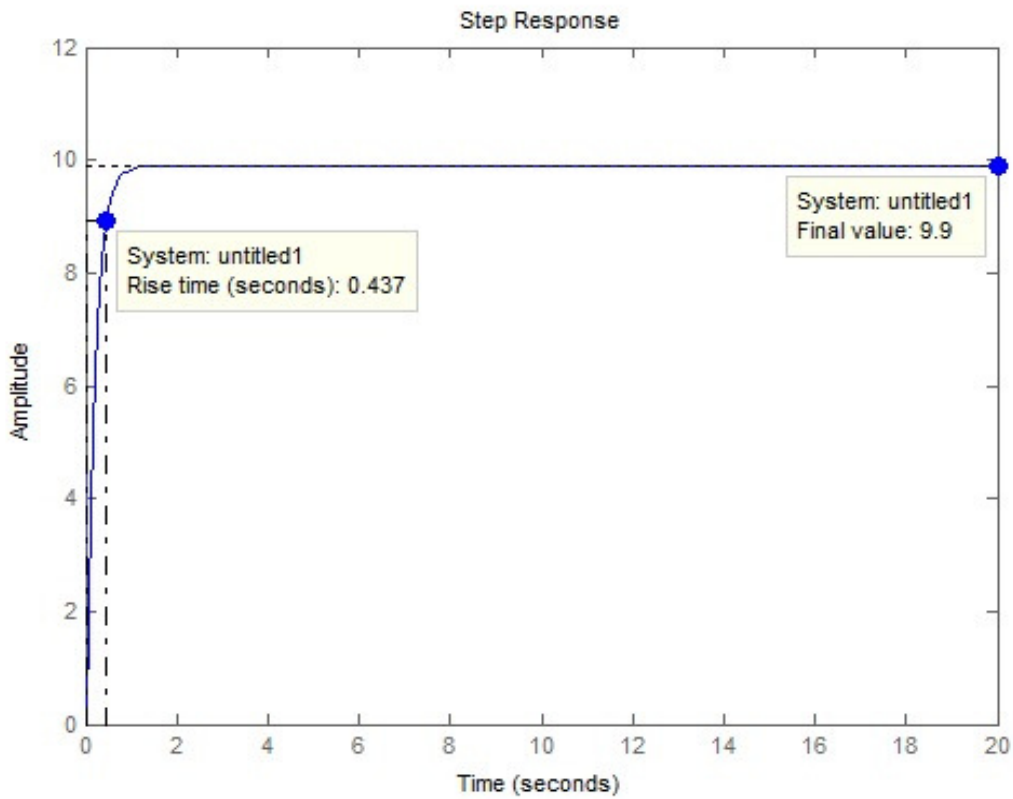


Figura 3.3: Risposta al gradino con $k_p = 5000$

Si osserva che ora l'errore a regime è quasi nullo e il tempo di salita si è ridotto notevolmente. Tale risposta non è però realistica, in quanto fisicamente è improponibile che un sistema di Cruise control riesca a variare la velocità del veicolo da 0 a 10 m/s in meno di 0.5 secondi, a causa della potenza limitata messa a disposizione dal motore e dagli organi di trasmissione.

Per risolvere questo problema e giungere ad una soluzione fisicamente realizzabile, riduciamo il guadagno proporzionale k_p in modo da ottenere un tempo di salita accettabile ed aggiungiamo un controllore integratore, il quale permette di incrementare il tipo del sistema in modo da inseguire con errore a regime nullo l'ingresso a gradino.

3.2.2 – Controllore proporzionale-integrativo PI

Aggiungiamo ora al controllore proporzionale un integratore. La funzione di trasferimento di un controllore PI si ottiene ponendo $k_d = 0$ nella generica (3.3):

$$C(s) = k_p + \frac{k_i}{s} \quad (3.6)$$

Analogoamente per quanto fatto con il controllore P, sostituiamo la funzione di trasferimento (3.6) nell'equazione generale (3.2) ottenendo la seguente funzione di trasferimento ad anello chiuso:

$$H(s) = \frac{G(s) \cdot C(s)}{1 + G(s) \cdot C(s)} = \frac{\frac{k_p \cdot s + k_i}{s \cdot (m \cdot s + b)}}{\frac{s \cdot (m \cdot s + b) + k_p \cdot s + k_i}{s \cdot (m \cdot s + b)}} = \frac{k_p \cdot s + k_i}{m \cdot s^2 + (b + k_p) \cdot s + k_i} \quad (3.7)$$

Proviamo a porre $k_p = 100$ e $k_i = 1$ e vediamo cosa accade alla risposta al gradino.

Modifichiamo il file-m precedente nel modo seguente:

`Kp = 600;`

`Ki = 1;`

`C = pid(Kp, Ki);`

`T = feedback(C*P_Cruise, 1);`

`step(r*T,t);`

L'esecuzione del file così modificato porta alla risposta al gradino:

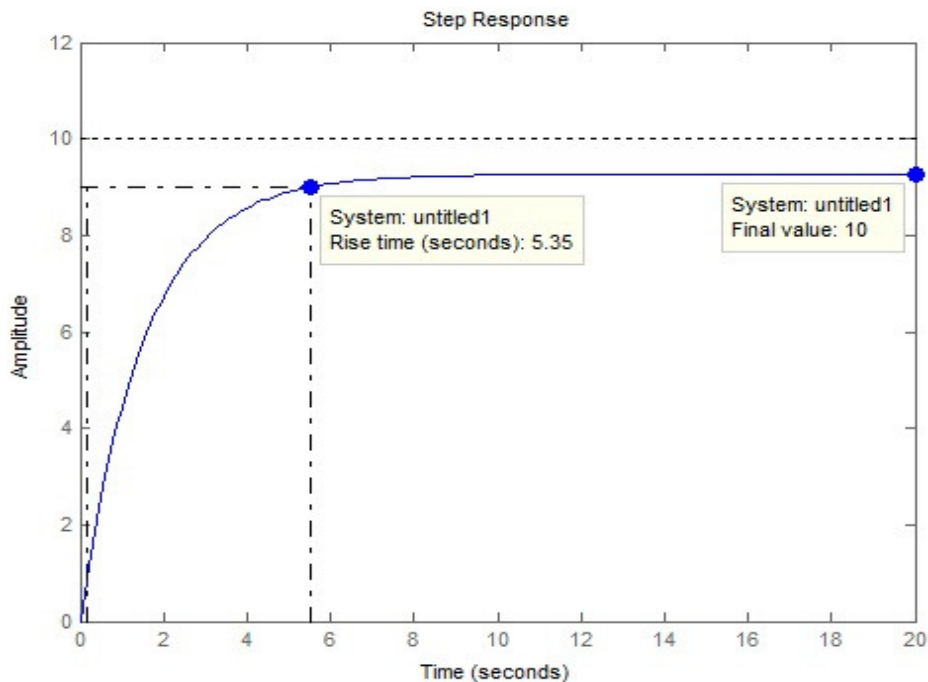


Figura 3.4: Risposta al gradino con $k_p = 100$ e $k_i = 1$

Dalla Figura 3.4 si osserva che la specifica sul tempo di salita non è rispettata, mentre grazie all'introduzione dell'integratore abbiamo ottenuto un errore a regime nullo. Il valore a regime viene però raggiunto dopo un tempo di assestamento molto lungo, infatti dal grafico si osserva come dopo un tempo di 20s non sia ancora stata raggiunta la velocità desiderata di 10 m/s e nella pratica ciò non è desiderabile.

Vogliamo quindi modificare i valori delle costanti k_p e k_i in modo da migliorare la risposta al gradino del sistema.

Nella scelta di tali valori bisogna evitare di porre la costante integrativa k_i troppo elevata, in quanto ciò potrebbe introdurre una sovraelongazione e un carattere oscillatorio prima di raggiungere il valore a regime. Questo fenomeno è visibile nelle risposte al gradino riportate in Figura 3.5 e ottenute rispettivamente per $k_i = 300$ e $k_i = 1500$:

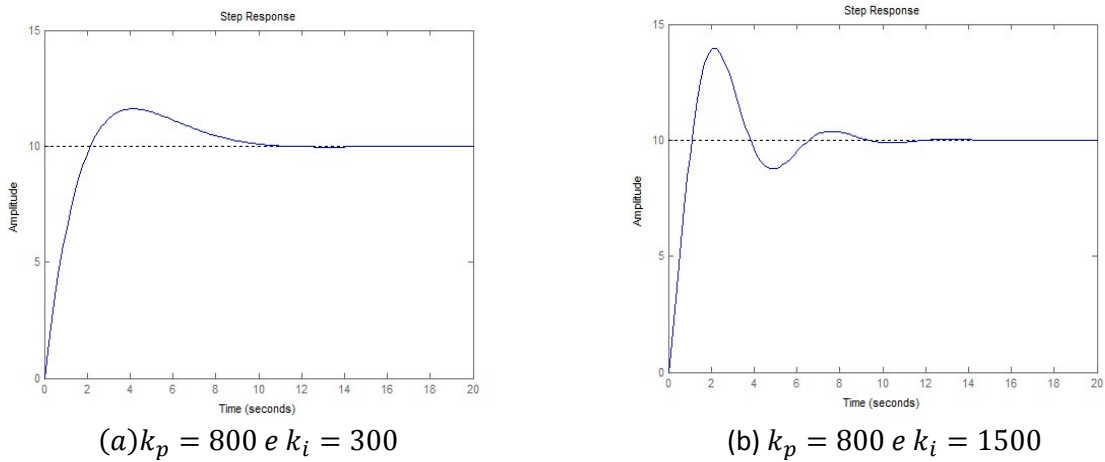


Figura 3.5: Risposta al gradino per elevati valori di k_i

Procedendo per tentativi, variando i valori di k_p e k_i partendo da valori piccoli della costante integrativa, si vede che per $k_p = 800$ e $k_i = 40$ la risposta al gradino, riportata in Figura 3.6, soddisfa le specifiche.

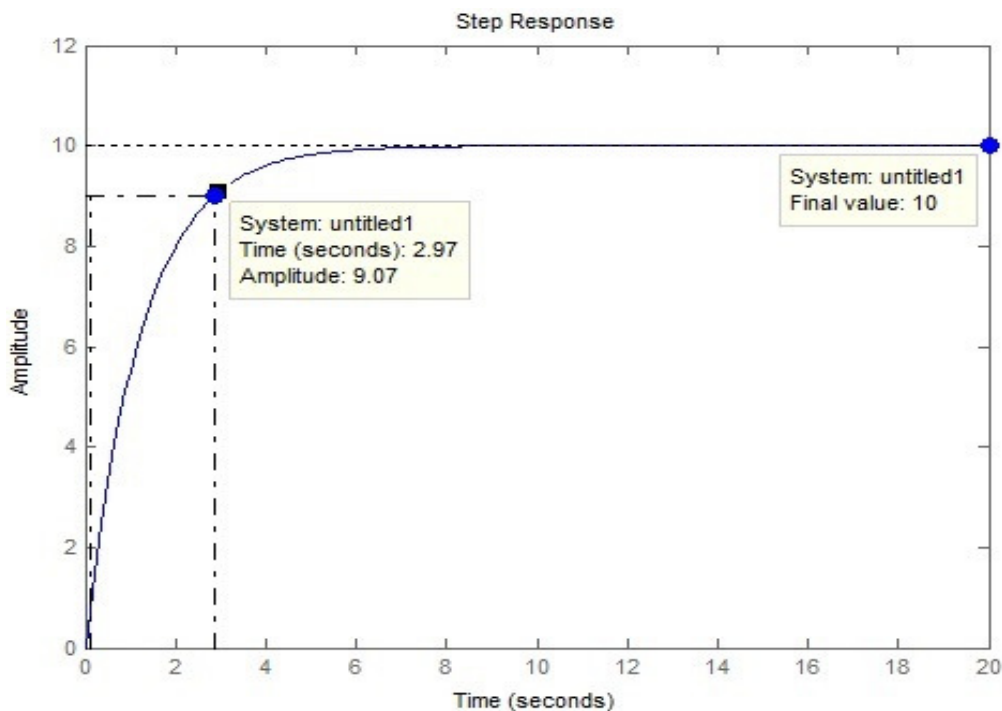


Figura 3.6: Risposta al gradino per $k_p = 800$ e $k_i = 40$

Il tempo di salita è pari a 2.97s e soddisfa quindi i vincoli progettuali e l'errore a regime, come ci aspettavamo, è nullo grazie all'azione dell'integratore. In questo caso inoltre il valore a regime di 10 m/s viene raggiunto con un tempo di assestamento accettabile.

3.2.3 – Controllore proporzionale-integrativo-derivativo PID

Pur essendo le specifiche richieste già soddisfatte da un controllore PI, vediamo cosa accade al comportamento del sistema se aggiungiamo al controllore precedente una componente derivativa k_d . Sostituendo la generica equazione (3.3) di un controllore PID nella (3.2) si ottiene la funzione di trasferimento globale del sistema:

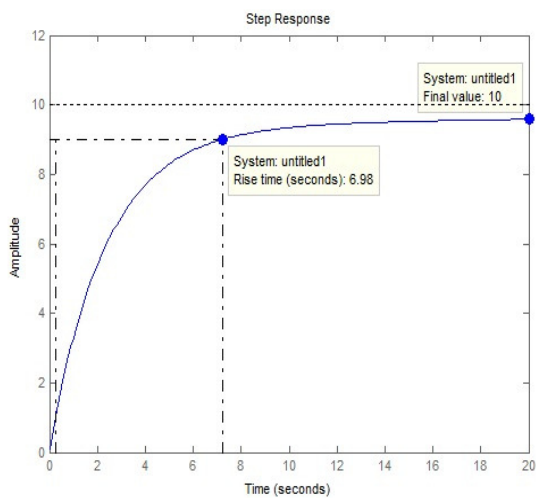
$$H(s) = \frac{k_d s^2 + k_p s + k_i}{(m + k_d) s^2 + (b + k_p) s + k_i} \quad (3.8)$$

Utilizziamo Matlab per tracciare le risposte al gradino di tale sistema corrispondenti a diversi valori delle costanti k_p , k_i e k_d , fino ad individuare per tentativi quei valori ottimali che soddisfano le specifiche di progetto. Creiamo il file-m contenente le seguenti istruzioni e lo eseguiamo variando i valori delle costanti:

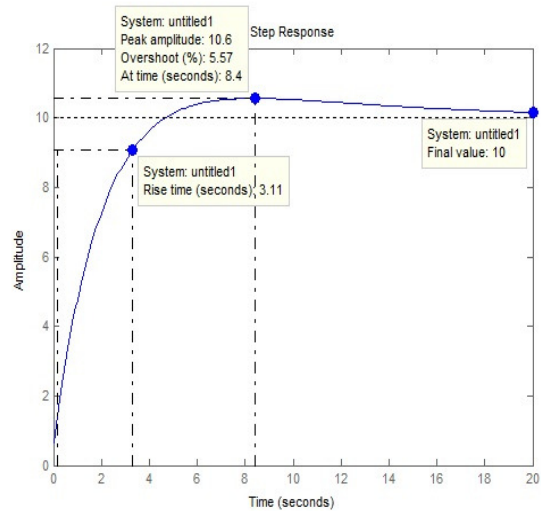
```

b = 50;
r = 10;      %Ingresso di riferimento (m/s)
m = 1000;
s = tf('s');
P_Cruise = 1/(m*s + b);
Kp = 400;   %Costante proporzionale
Ki=10;     %Costante integrativa
Kd=10;     %Costante derivativa
C = pid(Kp,Ki,Kd);   %Funzione di trasferimento del PID
T = feedback(C*P_Cruise,1);
t = 0:0.1:20;
step(r*T,t);
axis([0 20 0 12])

```



a) $K_p = 400, K_i = 10, K_d = 10$



b) $K_p = 600, K_i = 70, K_d = 70$

Figura 3.7: Risposta al gradino per diversi valori delle costanti

Una possibile combinazione dei valori dei parametri che porta il sistema a soddisfare le specifiche richieste è $K_p = 1000, K_i = 40$ e $K_d = 20$. La risposta al gradino corrispondente è riportata nella figura seguente e come si vede si porta al valore desiderato di 10 m/s con un tempo di salita inferiore a 5s, senza presentare alcuna sovraelongazione.

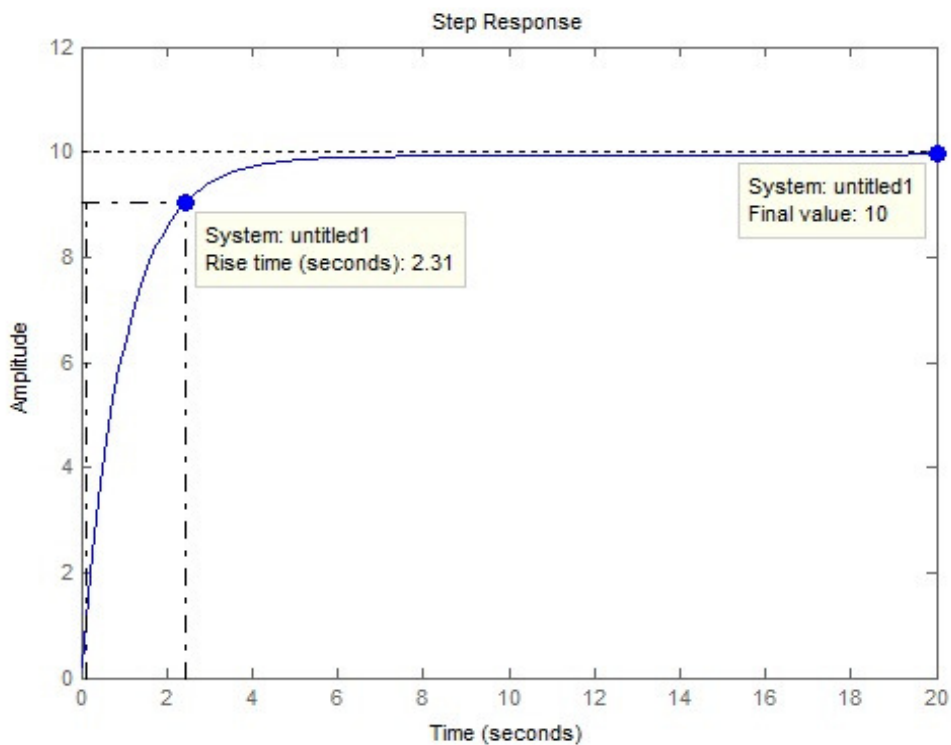


Figura 3.8: Risposta al gradino per $K_p = 1000, K_i = 40, K_d = 20$

3.2.4 – Osservazioni conclusive sui controllori PID

Come abbiamo osservato nel paragrafo 3.2.1, l'introduzione di un compensatore puramente proporzionale permette di rispettare le specifiche richieste solo in via teorica, in quanto non è fisicamente realizzabile. Invece l'utilizzo di un controllore proporzionale integrativo PI consente di ottenere buone prestazioni dinamiche del sistema; in particolare per i valori $k_p = 800$ e $k_i = 40$ si ottiene errore a regime nullo e tempo di salita di 2.97s, soddisfacendo le specifiche richieste.

Abbiamo inoltre verificato che anche con un controllore PID si è in grado di soddisfare i vincoli progettuali, ad esempio con i valori $K_p = 1000$, $K_i = 40$, $K_d = 20$. Dobbiamo però osservare che la sua funzione di trasferimento è impropria e quindi in fase di realizzazione necessita l'introduzione di un polo in alta frequenza che attenui, senza alterare il comportamento del sistema nella banda in cui opera, le componenti del segnale, tipicamente rumorose, che operano in alta frequenza. Tale accorgimento complica la realizzazione del controllore e per questo motivo è preferibile l'utilizzo di un controllore PI.

3.3 – Analisi tramite luogo delle radici

Consideriamo il sistema retroazionato avente funzione di trasferimento:

$$H(s) = \frac{C(s) \cdot G(s)}{1 + C(s) \cdot G(s)}$$

dove $C(s) = k_p$ è un controllore proporzionale e $G(s) = \frac{p(s)}{q(s)}$, con $p(s)$ e $q(s)$ monici e coprime tra loro, rappresenta la funzione di trasferimento del sistema in catena aperta.

Il metodo del luogo delle radici permette la determinazione dei poli della funzione di trasferimento in catena chiusa $H(s)$ al variare del parametro k_p e quindi di determinare quei valori di tale parametro che garantiscono un certo comportamento del sistema, come ad esempio la BIBO-stabilità.

Sostituendo l'espressione di $G(s)$ e $C(s)$ nell'equazione (3.2) si ottiene:

$$H(s) = \frac{k_p \frac{p(s)}{q(s)}}{1 + k_p \frac{p(s)}{q(s)}} = \frac{k_p \cdot p(s)}{q(s) + k_p \cdot p(s)} \quad (3.9)$$

Il luogo delle radici è rappresentato dall'insieme dei punti che, al variare di k_p in $\mathbb{R} \setminus \{0\}$, annullano il denominatore di $H(s)$, ovvero $q(s) + k_p \cdot p(s)$.

Tale metodo permette quindi di individuare la posizione di tutti i poli della funzione di trasferimento in catena chiusa $H(s)$ a partire dalla conoscenza esatte di tutti gli zeri e i poli della funzione di trasferimento in catena aperta $G(s)$.

Tornando al nostro esempio, inseriamo l'espressione della risposta in catena aperta, ovvero $G(s) = \frac{1}{(m \cdot s + b)}$, nell'equazione (3.9) che esprime la risposta in catena chiusa $H(s)$, ottenendo:

$$H(s) = \frac{k_p}{m \cdot s + (k_p + b)} \quad (3.10)$$

Traduciamo ora le specifiche temporali imposte dal progetto in specifiche in frequenza, ovvero pulsazione naturale ω_n e fattore di smorzamento ξ .

Il fattore di smorzamento è legato alla sovraelongazione massima consentita M_p dall'approssimazione seguente:

$$M_p \approx e^{-\frac{\pi \cdot \xi}{\sqrt{1 - \xi^2}}} \quad (3.11)$$

Ricordando che nel nostro esempio $M_p = 10\% = 0.1$, tramite formula inversa si determina il fattore di smorzamento:

$$\xi \geq \sqrt{\frac{\ln^2(M_p)}{\pi^2 + \ln^2(M_p)}} = 0.59 \cong 0.6 \quad (3.12)$$

Per calcolare la pulsazione naturale ricorriamo invece all'approssimazione seguente, ricordando che nel nostro esempio desideriamo un tempo di salita t_r massimo di 5s:

$$\omega_n \geq \frac{1.8}{t_r} = 0.36 \text{ rad/s} \quad (3.13)$$

Ora, tramite Matlab, tracciamo il luogo delle radici evidenziando la regione del piano complesso in cui sono soddisfatti i vincoli sui parametri in frequenza espressi da (3.12) e (3.13). Creiamo un nuovo file-m contenente il codice:

```
m = 1000;
b = 50;
r = 10;
s = tf('s');
P_Cruise = 1/(m*s + b);
rlocus(P_Cruise);    %istruzione che grafica il luogo delle radici
axis([-0.6 0 -0.6 0.6]);
sgrid(0.6, 0.36);    %coeff. di smorzamento e pulsazione naturale
```


Eseguendo il file-m appena creato si ottiene il luogo delle radici in figura:

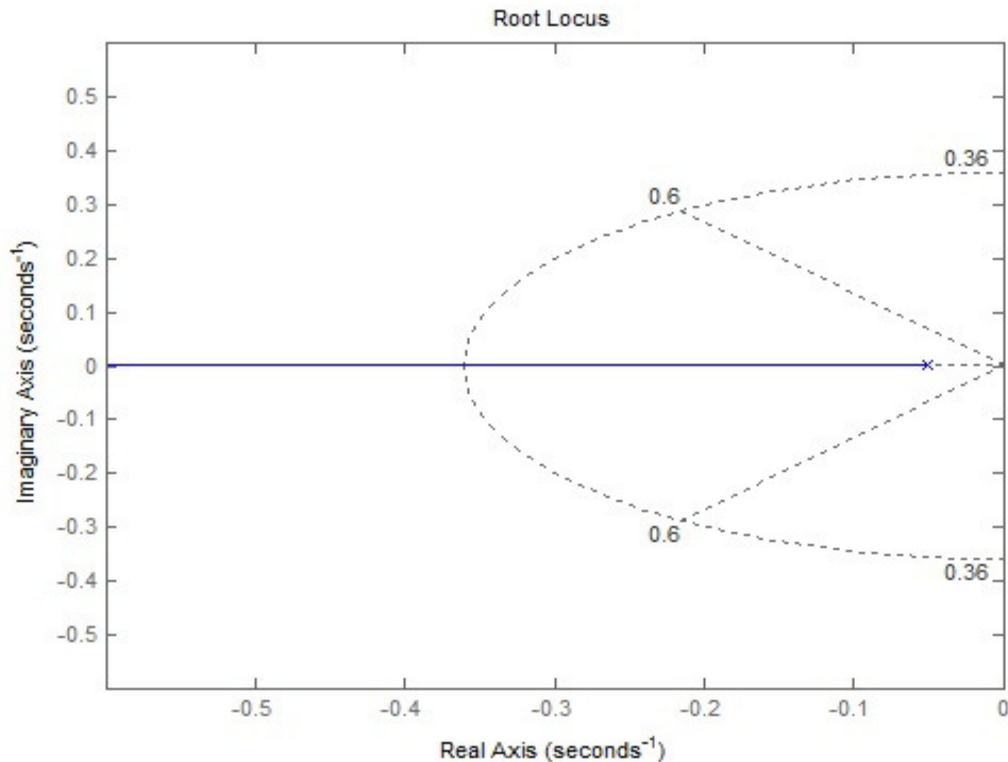


Figura 3.9: Luogo delle radici

L'unico polo presente, che sappiamo essere in -0.05 , è rappresentato correttamente con una x a sinistra dell'asse immaginario e non essendovi zero il ramo che parte dal polo si chiude all'infinito. In tale figura, inoltre, lungo i due segmenti tratteggiati il coefficiente di smorzamento vale esattamente 0.6 , mentre la semi-ellisse tratteggiata rappresenta i punti aventi pulsazione naturale 0.36 . Mantenendosi all'interno dei due segmenti il valore di ξ aumenta ed allo stesso modo man mano che ci si allontana dall'ellisse il valore della pulsazione naturale ω_n aumenta.

Se vogliamo conoscere il valore di k_p per una determinata posizione del polo, è possibile aggiungere alla fine del file-m l'utile istruzione:

```
[Kp,poles] = rlocfind(P_Cruise);
```

Non appena eseguiamo il file-m così modificato l'editor di Matlab ci chiede di selezionare nella finestra grafica un punto del luogo e ci fornisce in output nella Command Window il valore di k_p corrispondente.

Se proviamo ad esempio a selezionare un punto nell'intorno di -0.5 otteniamo come output:

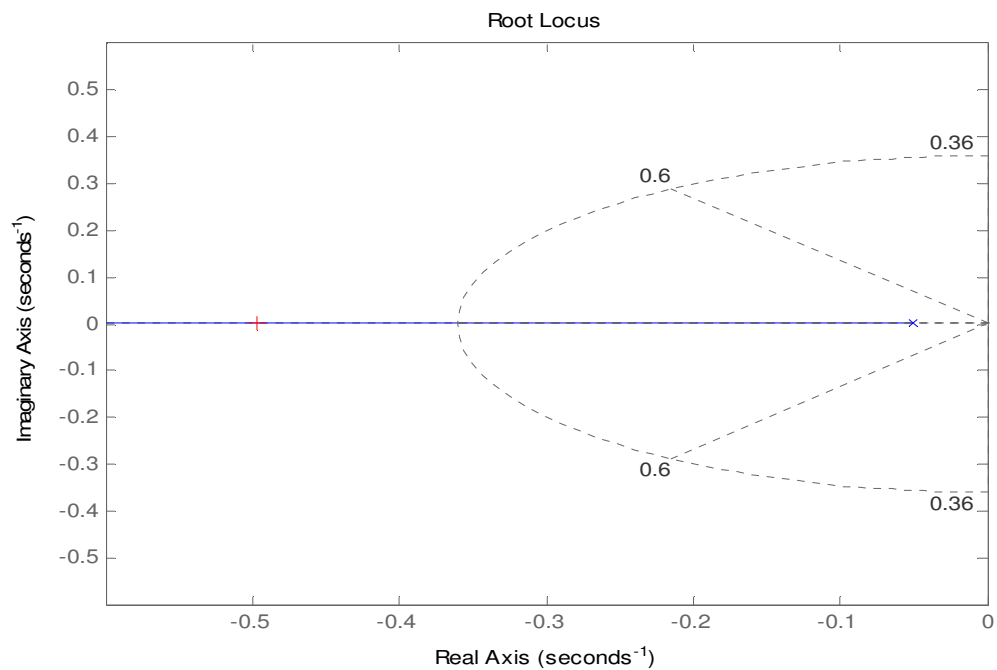


Figura 3.10: Punto del luogo selezionato

e la Command Window ci fornisce come risposta:

```
selected_point =
    -0.4969 + 0.0019i
Kp =
    446.9233
poles =
    -0.4969
```

Il valore di k_p ritornato può ora essere inserito direttamente nella funzione di trasferimento $H(s)$ in modo da poter ottenere la rispettiva risposta al gradino.

```
Kp = 446.9233;
T = feedback(Kp*P_Cruise, 1);
t = 0:0.1:20;
step(r*T, t);
```

axis([0 20 0 11]);

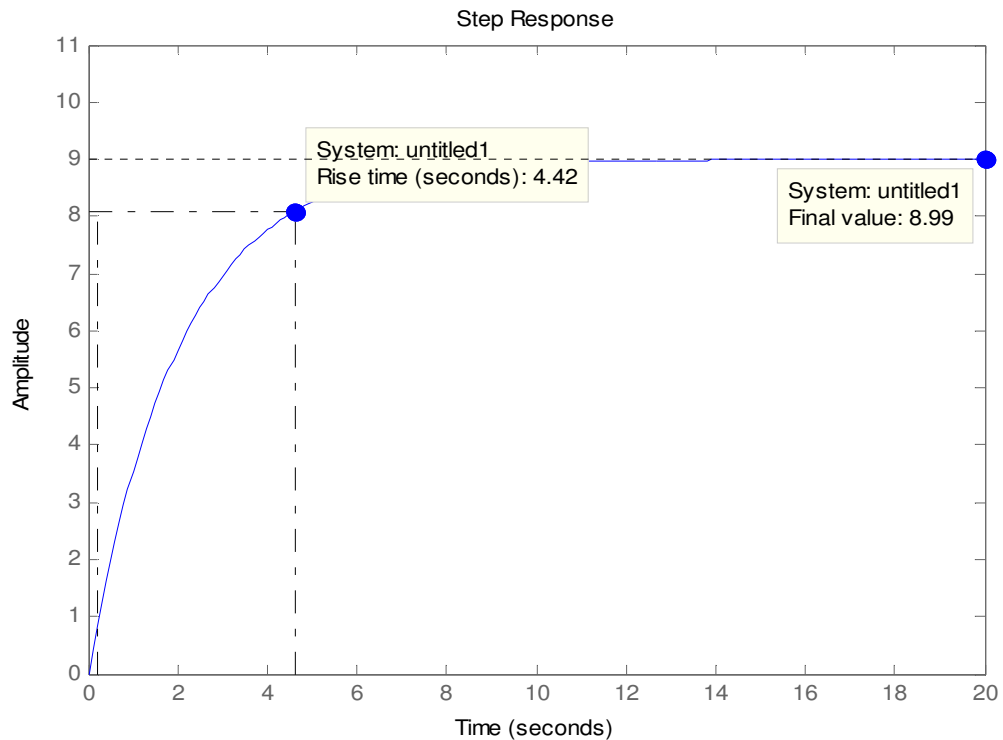


Figura 3.11: Risposta al gradino con $k_p = 446.9233$

Dalla figura si vede che la specifica sul tempo di salita viene rispettata, infatti, come richiesto, risulta inferiore a 5 secondi. L'errore a regime invece non rispetta le specifiche, in quanto risulta circa del 10%, mentre viene richiesto che sia inferiore al 2%.

Ricaviamo allora la risposta al gradino, in modo analogo a quanto appena fatto, per punti diversi del luogo, in modo da vedere come il sistema si comporta man mano che il polo si sposta lungo il ramo.

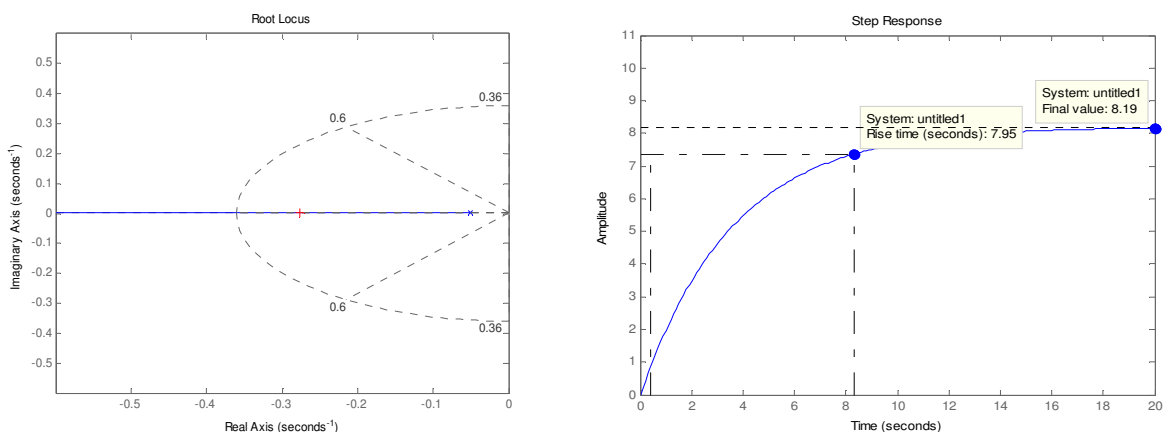


Figura 3.12: Luogo delle radici e risposta al gradino per il punto $-0.2765 + 0.0019i$ con $k_p=226.54$

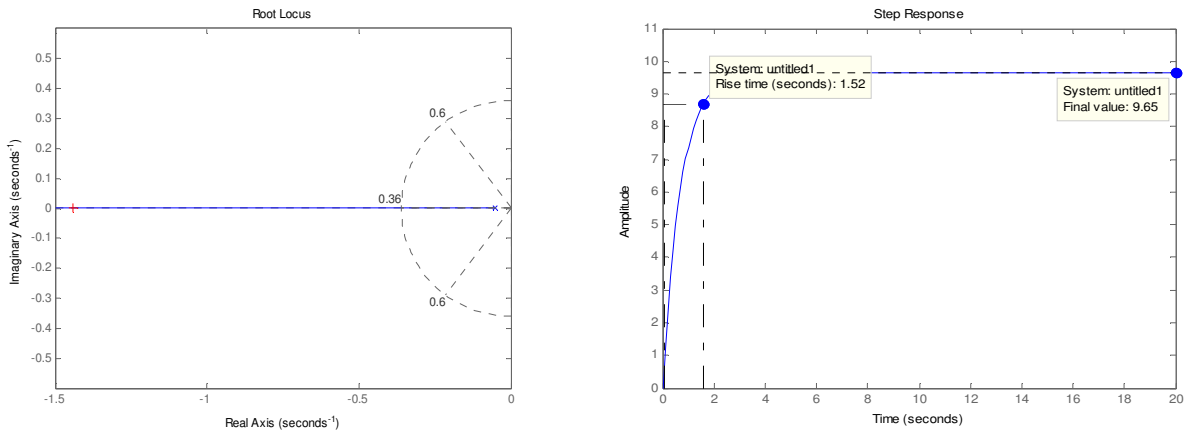


Figura 3.13: Luogo delle radici e risposta al gradino per il punto $-1.4414 + 0.0019i$ con $k_p=1391.4$

Man mano che il polo si allontana dall'asse immaginario e che quindi il guadagno k_p aumenta, si osserva che sia il tempo di salita sia l'errore a regime diminuiscono, ricadendo nel caso del controllore proporzionale P, che come già analizzato presenta un limite legato alla realizzazione fisica. Per raggiungere un errore a regime sufficientemente basso e un tempo di salita che soddisfi le specifiche pur rimanendo fisicamente ottenibile manteniamo il guadagno proporzionale k_p non eccessivamente elevato ed aggiungiamo al sistema una rete ritardatrice.

3.3.1 - Introduzione della rete ritardatrice (phase LAG)

Per ridurre l'errore a regime possiamo aggiungere al sistema una rete ritardatrice avente funzione di trasferimento:

$$C(s) = \frac{s+z_0}{s+p_0} \quad (3.14)$$

dove la pulsazione del polo ω_p è inferiore alla pulsazione dello zero ω_z . L'introduzione di una rete ritardatrice è utile per ridurre l'errore a regime, infatti si può dimostrare che tale errore cala di un fattore $\frac{z_0}{p_0}$.

Consideriamo ora il sistema in retroazione unitaria del nostro esempio, costituito dalla rete ritardatrice, dal controllo proporzionale e dalla funzione di trasferimento $G(s)$ del sistema in catena aperta. Si ottiene facilmente la seguente funzione di trasferimento in catena chiusa:

$$H(s) = \frac{k_p \cdot s + k_p \cdot z_0}{m \cdot s^2 + (b + m \cdot p_0 + k_p) \cdot s + (b \cdot p_0 + k_p \cdot z_0)} \quad (3.15)$$

Come già accennato, l'utilizzo di una rete ritardatrice permette di ridurre l'errore a regime di un fattore $\frac{z_0}{p_0}$, quindi poniamo $z_0 = 0.3$ e $p_0 = 0.03$ in modo da ridurre l'errore di un fattore 10.

Tracciamo ora il luogo delle radici di $H(s)$ utilizzando i comandi Matlab:

```
z0 = 0.3;
p0 = 0.03;
C_lag = (s+z0)/(s+p0);
rlocus(C_lag * P_Cruise)    % Rappresentazione grafica del luogo
axis([-0.6 0 -0.4 0.4])
sgrid(0.6,0.36);          % Imposta i parametri di smorzamento e pulsazione
```

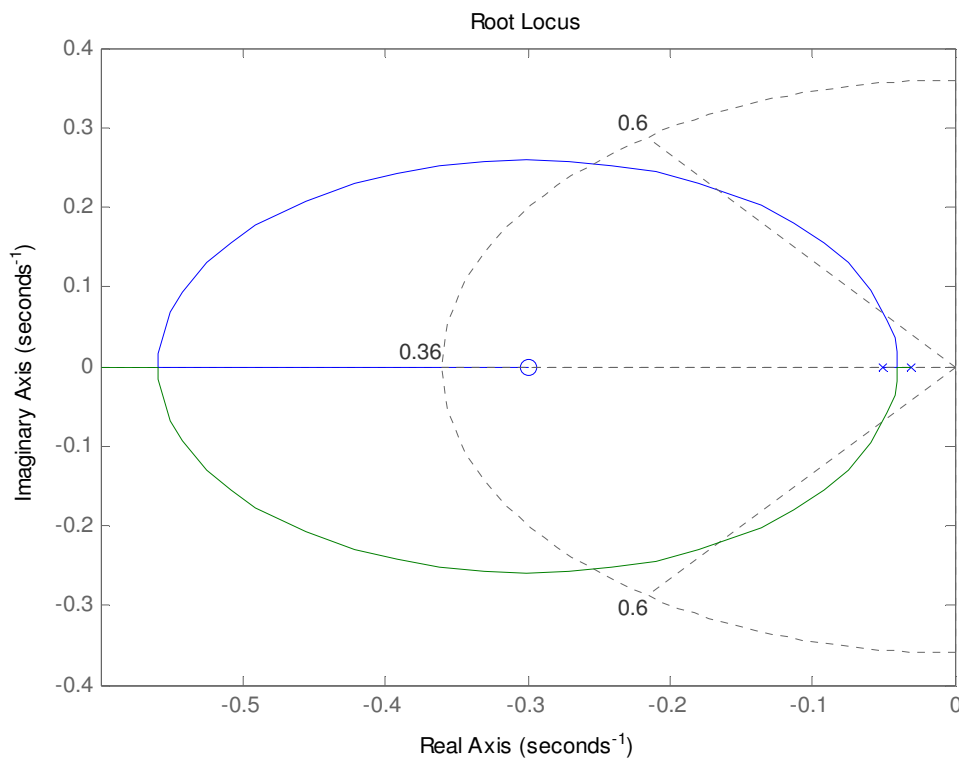


Figura 3.14: Luogo delle radici con rete ritardatrice

Utilizziamo ora nuovamente il comando `rlocfind`, aggiungendo l'istruzione :

```
[Kp,poles] = rlocfind(C_lag*P_Cruise)
```

e selezioniamo ad esempio un punto del luogo nell'intorno di -0.4, come mostrato in Figura 3.13.

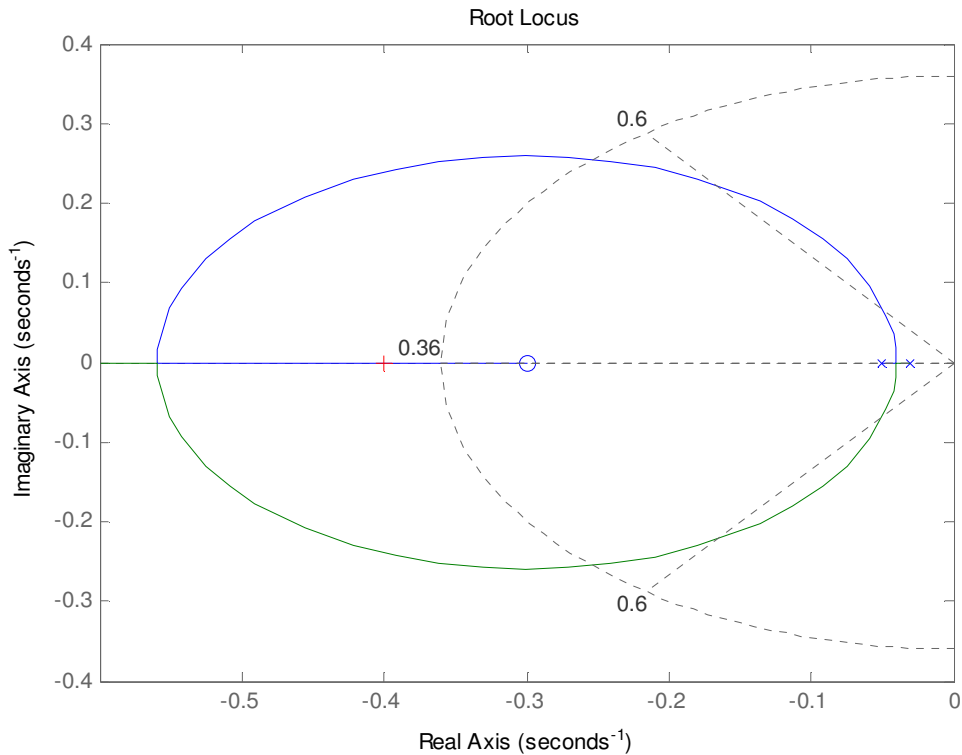


Figura 3.15: Punto del luogo selezionato

Nella Command Window otteniamo come output:

selected_point =

-0.4002 - 0.0037i

Kp =

1.2929e+003

poles =

-0.9725

-0.4004

quindi il valore di k_p corrispondente al punto selezionato è 1293.9 e lo possiamo utilizzare come già visto in precedenza per rappresentare la risposta al gradino corrispondente:

`Kp = 1293.9;`

`T = feedback(C_lag*Kp*P_Cruise,1);` %istruzione che implementa la f.d.t.

`%retroazionata`

`t=0:0.1:20;`

`step(r*T,t)` %rappresentazione grafica della risposta al gradino

axis([0 20 0 12])

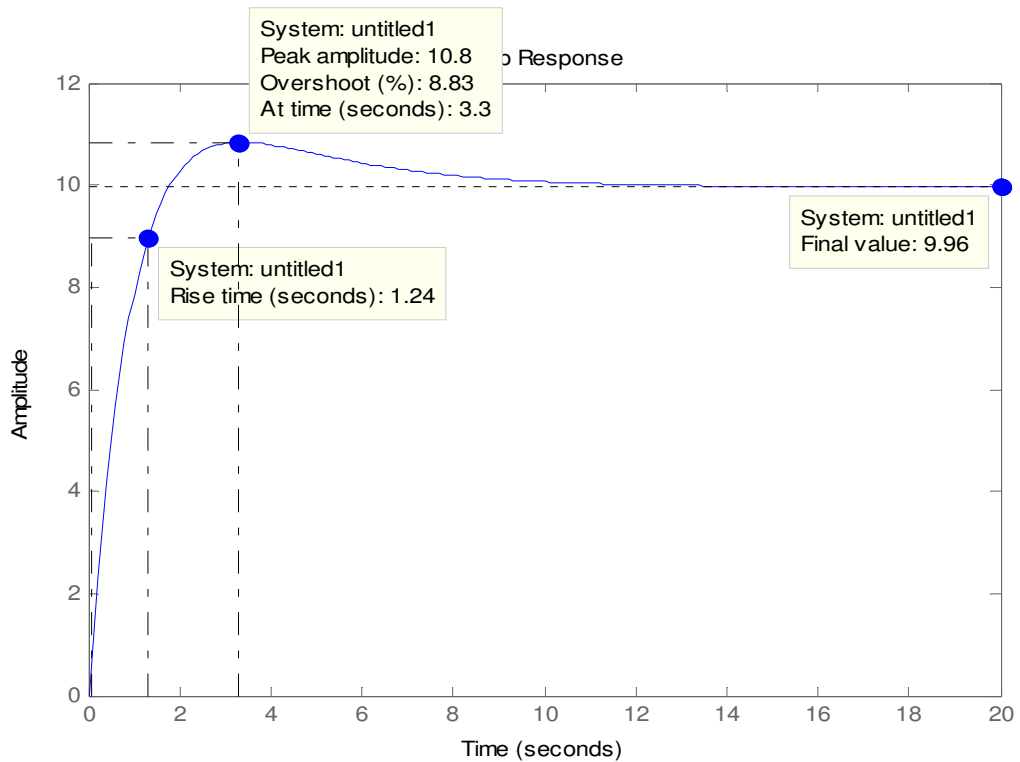


Figura 3.16: Risposta al gradino corrispondente al punto del luogo selezionato

L'introduzione di una rete ritardatrice ha migliorato il comportamento dinamico del sistema, in quanto sono rispettati i vincoli progettuali sul tempo di salita e sull'errore a regime. Bisogna però porre ancora attenzione a non ottenere valori del tempo di salita troppo bassi e quindi non ottenibili fisicamente. Si osserva inoltre la presenza di una sovraelongazione in prossimità di 3.3 secondi dovuta allo zero introdotto dalla rete ritardatrice, ma questa è accettabile in quanto risulta inferiore al 10% e quindi rispetta le specifiche assegnate.

3.3.2 - Osservazioni conclusive sul luogo delle radici

Tramite il metodo del luogo delle radici abbiamo ancora una volta osservato come l'introduzione di un solo guadagno proporzionale non sia sufficiente a soddisfare le specifiche richieste. Infatti l'incremento di tale guadagno permette una riduzione dell'errore a regime a cui corrisponde però una drastica diminuzione del tempo di salita, che rende il sistema fisicamente irrealizzabile. Per risolvere questo problema abbiamo allora introdotto una rete ritardatrice, che è in grado di ridurre l'errore a regime e di diminuire il tempo di salita mantenendolo però a valori che rendano il sistema implementabile. Infine abbiamo osservato che la presenza di uno zero nella rete ritardatrice ha introdotto una sovraelongazione, comunque accettabile in quanto rispetta i vincoli progettuali.

3.4 - Analisi in frequenza

3.4.1 - Introduzione all'analisi in frequenza

Fino ad ora abbiamo studiato la dinamica del sistema in funzione di soli parametri temporali, come tempo di salita, errore a regime e sovraelongazione. Vogliamo ora studiare il comportamento del sistema nel dominio della frequenza, analizzando la risposta in frequenza $W(j\omega)$ ottenuta dalla funzione di trasferimento $W(s)$ tramite il cambio di variabile $s = j\omega$.

La risposta in frequenza può essere rappresentata graficamente attraverso diverse tipologie di diagrammi, di cui noi consideriamo le due più utilizzate.

Una prima è data dai diagrammi di Bode, che rappresentano in scala semilogaritmica e in funzione della pulsazione l'andamento del modulo della risposta in frequenza espresso in dB e dell'argomento della risposta in frequenza espresso in gradi o radianti. I diagrammi di Bode permettono di ricavare in modo immediato i valori di alcuni parametri fondamentali per lo studio della risposta in frequenza, ovvero:

- pulsazione di attraversamento ω_a di $G(s)$, ovvero quella pulsazione positiva in corrispondenza della quale il diagramma di Bode del modulo interseca l'asse delle ascisse: $|G(j\omega_a)|_{db} = 0$;
- fase di attraversamento φ_a di $G(s)$, ovvero il valore dell'argomento corrispondente alla pulsazione di attraversamento: $\varphi_a = \arg(G(j\omega_a))$;
- margine di fase di $G(s)$, ovvero la grandezza $m_\varphi = \pi + \varphi_a$.

Una ulteriore rappresentazione grafica spesso utilizzata è data dal diagramma di Nyquist, dove viene rappresentato l'andamento della funzione complessa $W(j\omega)$ nel piano complesso con una curva parametrizzata dalla pulsazione ω . Come vedremo in seguito il diagramma di Nyquist risulta di grande utilità per lo studio della BIBO-stabilità del sistema in catena chiusa.

Prima di proseguire con lo studio della risposta in frequenza in catena aperta ed in catena chiusa è utile definire i seguenti parametri:

- banda passante (a 3dB) B_p : individua l'intervallo di pulsazioni $[0, B_p]$ in cui si ha che $|W(j\omega)|_{db} \geq |W(0)|_{db} - 3dB$
- pulsazione di risonanza ω_r : pulsazione positiva in corrispondenza della quale il modulo assume valore massimo, ovvero $|W(j\omega_r)| = \max_{(\omega>0)} |W(j\omega)|_{db}$;
- picco di risonanza relativo M_{rel} : rappresenta il rapporto tra il modulo della risposta in frequenza in corrispondenza della pulsazione ω_r e $|W(0)|$, ovvero $M_{rel} = \left[\frac{|W(j\omega_r)|}{|W(0)|} \right]_{db}$.

3.4.2 - Considerazioni in catena aperta e catena chiusa

Per effettuare l'analisi in frequenza del sistema consideriamo, come fatto per il luogo delle radici, un controllore proporzionale $C(s) = k_p$. La funzione di trasferimento di tale sistema, che abbiamo già ricavato, risulta:

$$W(s) = C(s) \cdot G(s) = \frac{k_p}{m \cdot s + b}$$

Se in particolare consideriamo $k_p = 1$ otteniamo il sistema già analizzato nel Capitolo 2 e di cui abbiamo già verificato la BIBO-stabilità in catena aperta. Per studiarne invece la stabilità in catena chiusa ricorriamo al criterio ridotto di Nyquist, che esprime la relazione:

$$N = n_{W+} - n_{H+} \quad (3.16)$$

dove N rappresenta il numero di giri che il diagramma di Nyquist di $W(s)$ compie attorno al punto critico $-1+j0$ (contati con segno positivo se compiuti in senso antiorario e negativo se in senso orario), n_{W+} rappresenta il numero di poli a parte reale negativa della funzione di trasferimento $W(s)$ del sistema in catena aperta e n_{H+} rappresenta il numero di poli a parte reale negativa della funzione di trasferimento $H(s)$ del sistema retroazionato.

Per valutare il numero di giri N tracciamo il diagramma di Nyquist della funzione di trasferimento $W(s)$ tramite i comandi Matlab:

```
Kp=1;
```

```
nyquist(Kp*P_Cruise);
```

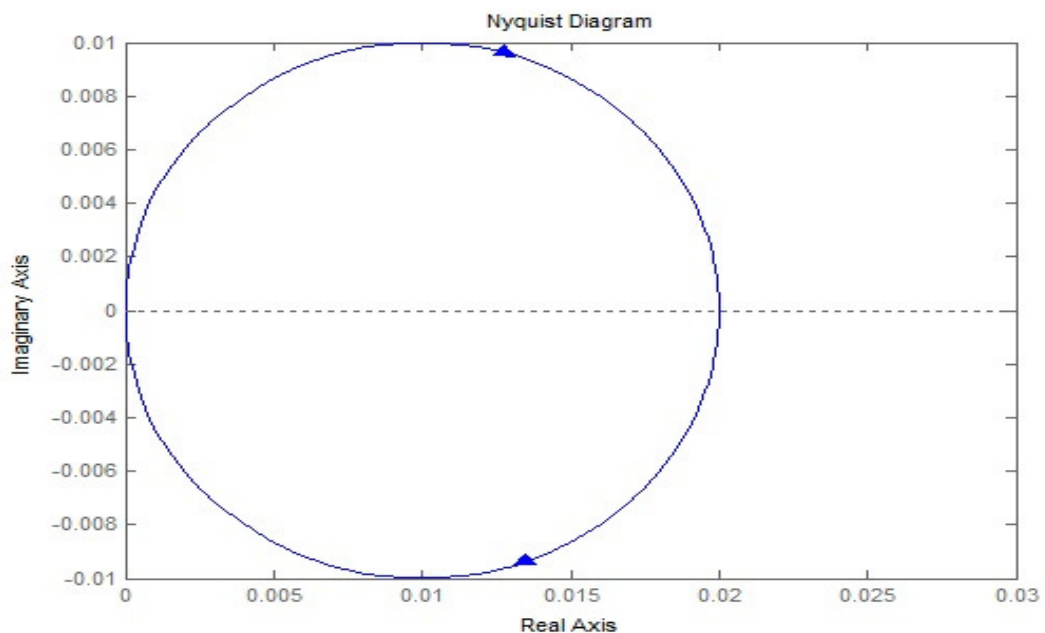


Figura 3.17: Diagramma di Nyquist di $W(S)$

Dal diagramma ottenuto si vede subito che il punto critico $-1+j0$ non viene circondato e quindi $N = 0$. Inoltre, essendo la funzione di trasferimento in catena aperta $W(s)$ a noi nota, sappiamo che $n_{W+} = 0$ e quindi segue immediatamente che $n_{H+} = 0$, il che significa che il sistema retroazionato risulta BIBO-stabile.

Tracciamo adesso il diagramma di Bode della funzione di trasferimento $W(s)$ tramite il comando Matlab:

```
Kp = 1;
bode(Kp*P_Cruise);
```

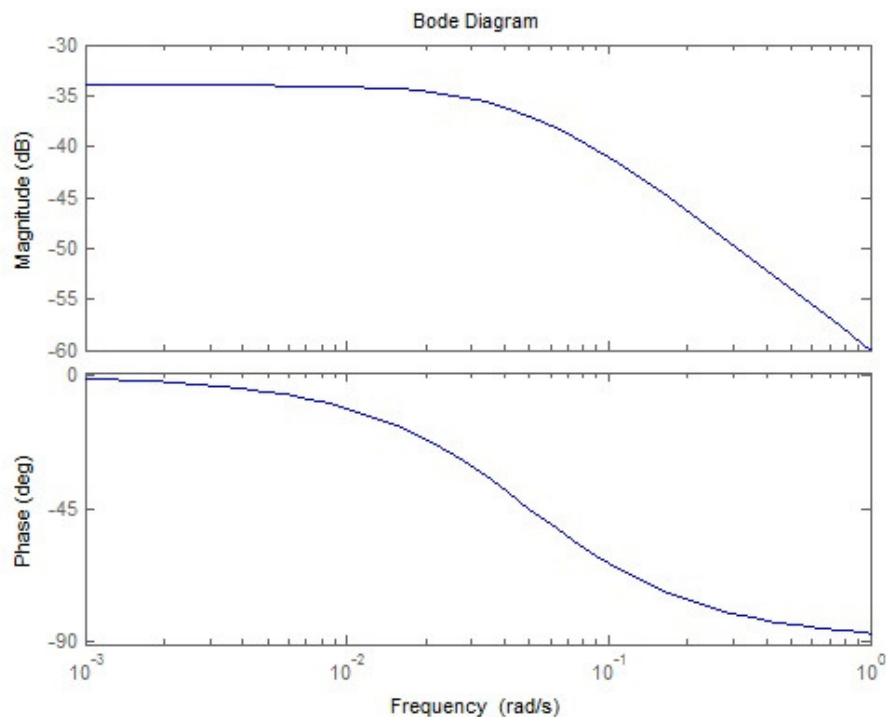


Figura 3.18: Diagrammi di Bode di $W(s)$

L'errore a regime può essere calcolato attraverso la seguente formula:

$$e_{rp} = \frac{1}{1+M_{\omega \rightarrow 0}} \cdot 100 = \frac{1}{1+0.02} \cdot 100 \approx 98\% \quad (3.17)$$

in quanto dal diagramma di Bode del modulo si vede che $M_{\omega \rightarrow 0} = -34dB = 0.02$.

Verifichiamo tale risultato osservando la risposta al gradino, che ricaviamo tramite i comandi Matlab:

```
r = 10;
W = feedback(Kp*P_Cruise,1);
step(r*W);
```

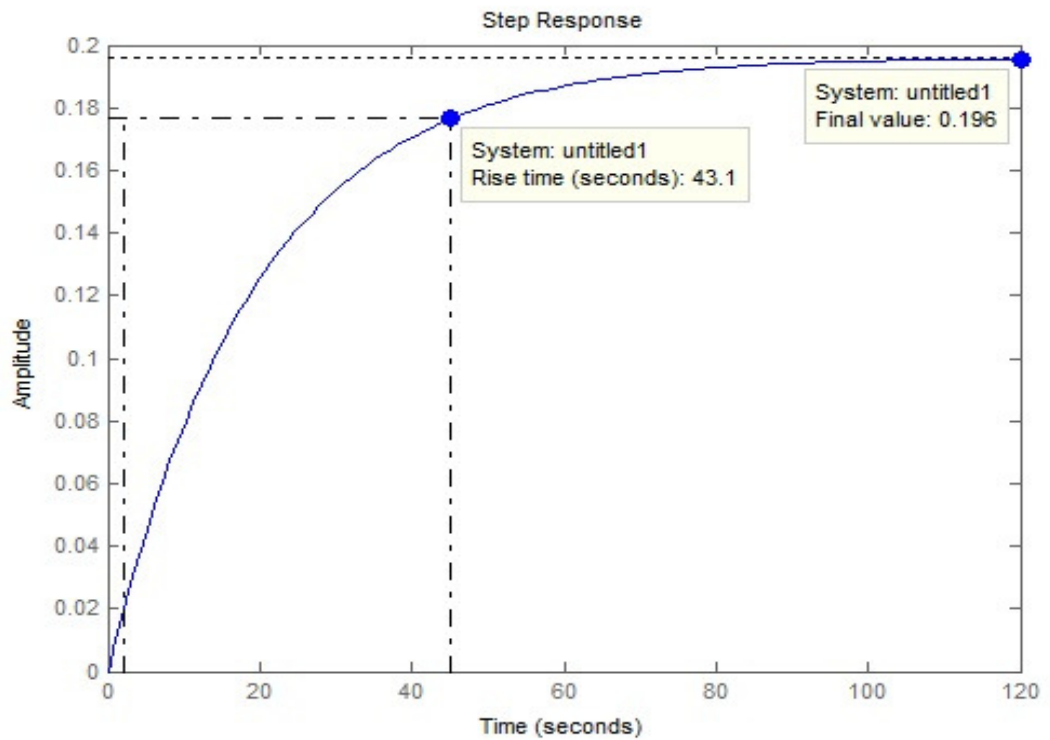


Figura 3.19: Risposta al gradino di $W(s)$ in catena chiusa

Per rispettare i vincoli di progetto vogliamo però ottenere un errore a regime inferiore al 2%. Ricaviamo quindi un valore di $M_{\omega \rightarrow 0}$ che comporti un errore a regime che rispetti le specifiche.

$$e_{rp} < 2\% \Rightarrow \frac{1}{1 + M_{\omega \rightarrow 0}} < 2\% \Rightarrow M_{\omega \rightarrow 0} > 49$$

Esprimendo tale valore in dB si ottiene:

$$M_{\omega \rightarrow 0_{dB}} = 33.8dB \quad (3.18)$$

Dobbiamo quindi alzare il diagramma del modulo di almeno:

$$|33.8 - (-34)|_{dB} = 67.8dB \quad (3.19)$$

il che si ottiene ponendo il guadagno $k_p > 2454.71$. Scegliamo ad esempio $k_p = 2500$.

Tracciamo il diagramma di Bode della funzione di trasferimento così ottenuta per vedere come il diagramma del modulo sia effettivamente stato traslato verso l'alto in modo da garantire il valore di $M_{\omega \rightarrow 0}$ cercato. Evidenziamo inoltre la pulsazione di attraversamento e il margine di fase:

$K_p = 2500$;

`margin(Kp*P_Cruise);`

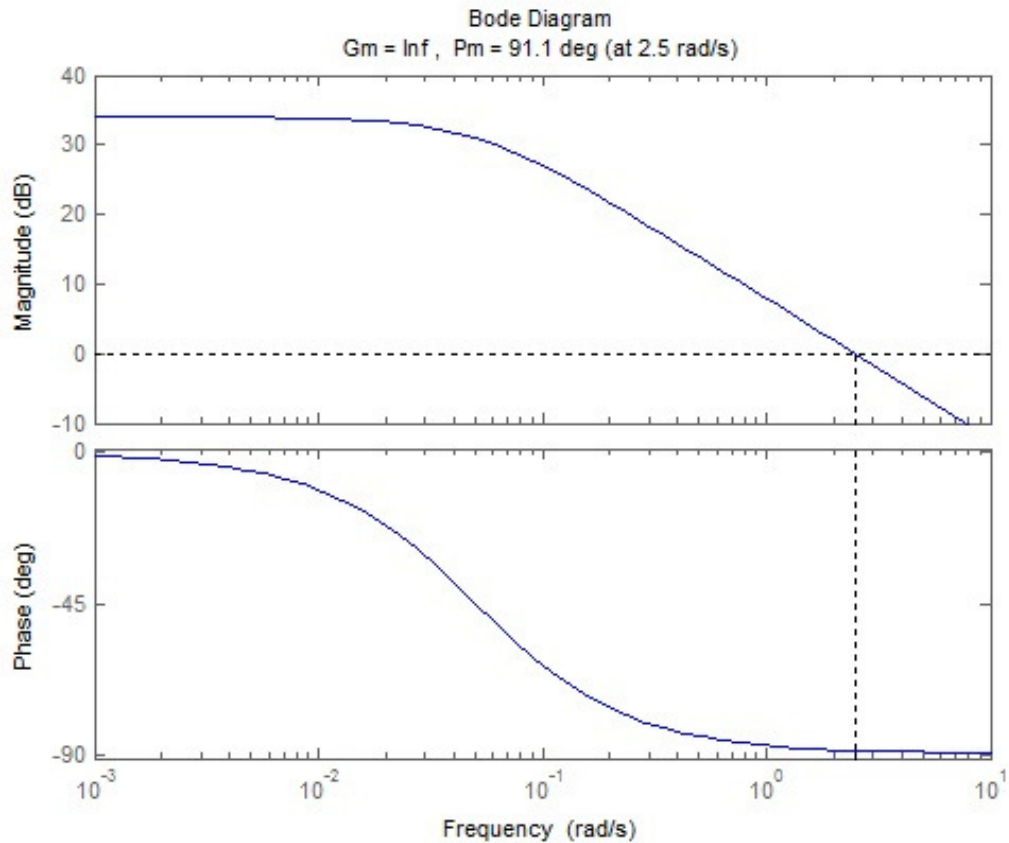


Figura 3.20: Diagrammi di Bode per $K_p = 2500$

Il diagramma in figura mostra un margine di fase di circa 90° e nell'intorno della pulsazione di attraversamento $\omega_a = 2.5 \text{ rad/s}$ il diagramma del modulo scende con una pendenza sufficiente a assicurare che il diagramma non si avvicini più all'asse delle ascisse. Ciò garantisce che vi sia una distanza sufficiente del diagramma di Nyquist dal punto critico $-1+j0$ e quindi la BIBO-stabilità del sistema.

Analizziamo ora invece la risposta al gradino che si ottiene per $k_p = 2500$, mostrata in Figura 3.19. Si osserva che il sistema ancora una volta rispetta tutti i vincoli di progetto, garantendo errore a regime inferiore al 2% e tempo di salita inferiore a 5 secondi. Il tempo di salita risulta però troppo piccolo e, come già visto nei casi precedenti, ciò compromette la realizzabilità del sistema.

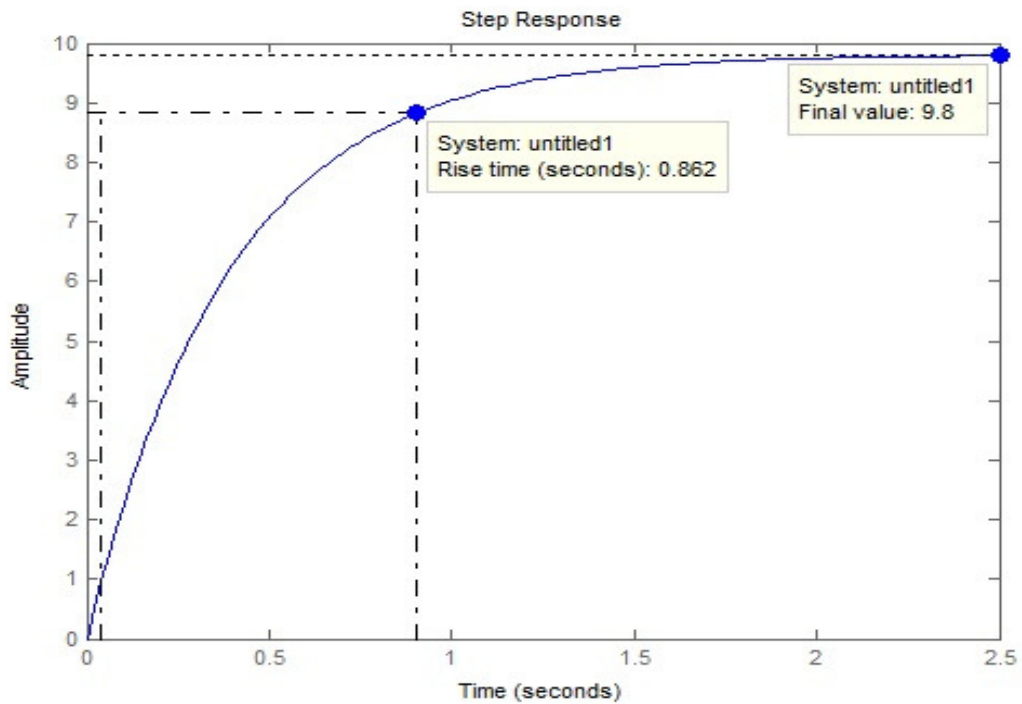


Figura 3.21: Risposta al gradino in catena chiusa per $K_p = 2500$

3.4.3 - Introduzione della rete ritardatrice

Una rete ritardatrice può essere descritta in frequenza dall'equazione seguente, ricavata in pochi passaggi dalla (3.14):

$$C_{lag}(s) = \frac{1}{\alpha} \left(\frac{1 + \alpha T s}{1 + T s} \right) \quad (3.20)$$

dove $\alpha = p_0/z_0$ e $T = 1/p_0$. Come già discusso in precedenza, in una rete ritardatrice si ha che la pulsazione del polo p_0 è inferiore alla pulsazione dello zero z_0 , quindi $\alpha < 1$.

Il diagramma di Bode di tale rete è quello mostrato in Figura 3.20 ed evidenzia un polo alla frequenza $\omega_p = 1/T$ ed uno zero alla frequenza $\omega_z = 1/\alpha T$.

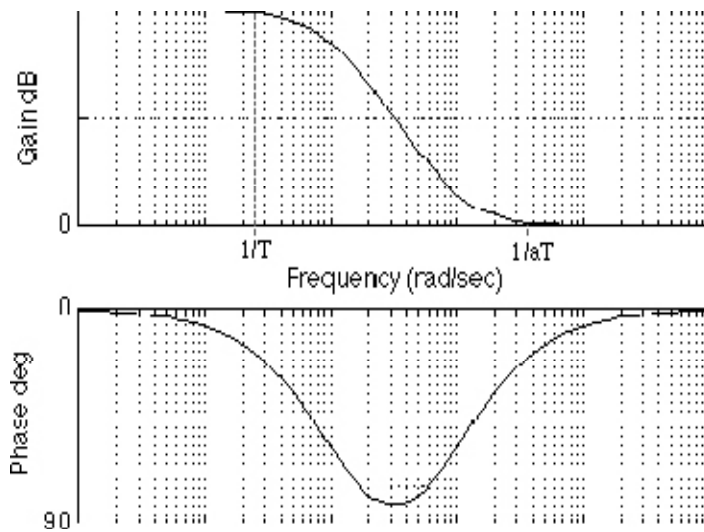


Figura 3.22: Diagramma di Bode di una rete ritardatrice

Dal diagramma del modulo si osserva come la rete ritardatrice causa a basse frequenze un incremento del guadagno pari a:

$$M_{lag} = \frac{\omega_p}{\omega_z} = \frac{1/T}{1/\alpha T} = \alpha \quad (3.21)$$

portando ad una riduzione dell'errore a regime di un fattore α ; ad alte frequenze invece il guadagno è unitario e non varia quindi il comportamento del sistema.

Dal diagramma della fase invece si osserva come la rete ritardatrice aggiunga una fase negativa del valore massimo di -90° tra il polo e lo zero e bisogna quindi porre molta attenzione a far in modo che il nuovo margine di fase soddisfi il requisito essenziale di stabilità. Per evitare che tale requisito non sia soddisfatto si cerca di posizionare lo zero sufficientemente prima della nuova pulsazione di attraversamento, esprimibile dalla relazione:

$$\omega_A' = \frac{1}{T\sqrt{\alpha}} \quad (3.22)$$

Vediamo ora di tradurre le specifiche temporali di progetto in vincoli sui parametri frequenziali, per poi ricavare quei valori di α e T che li soddisfano.

Come abbiamo già visto, possiamo ottenere un vincolo sul guadagno di Bode a partire dalla specifica sull'errore a regime attraverso la relazione:

$$e_{rp} < 2\% \Rightarrow \frac{1}{1+M_{\omega \rightarrow 0}} < 2\% \Rightarrow M_{\omega \rightarrow 0} > 49 \Rightarrow M_{dB} > 33.8 \text{ dB} \quad (3.23)$$

Unendo il risultato appena ottenuto alla (3.21) si ottiene:

$$\alpha = 10^{-\left|\frac{33.8 \text{ dB}}{20}\right|} \approx 0.02 \quad (3.24)$$

Dalla specifica sul tempo di salita ricaviamo invece una banda passante ed un pulsazione di attraversamento pari a:

$$B_{\omega} = \omega_n = \frac{1.8}{5} \approx 0.36 \text{ rad/s} \quad (3.25)$$

da cui si ottiene il valore di T:

$$T = \frac{10}{\alpha \omega_n} \approx 1389 \quad (3.26)$$

Possiamo infine legare il margine di fase m_{ϕ} al coefficiente di smorzamento ξ , che abbiamo già ricavato nella (3.12) a partire dal vincolo sulla sovraelongazione, tramite la relazione:

$$m_{\phi} \approx 100 \xi = 60^{\circ} \quad (3.27)$$

Per avere uno smorzamento pari ad almeno 0.6 dobbiamo quindi ottenere un margine di fase di almeno 60° .

Ora che abbiamo trovato dei valori di α e T che soddisfano le specifiche, tracciamo il diagramma di Bode del sistema in catena aperta formato dalla rete ritardatrice, da un controllore proporzionale con guadagno $k_p = 1000$ scelto per tentativi e dal nostro sistema di partenza G(s).

Kp=1000;

a = 0.02;

t = 1389;

C_lag = (1/a)*((1+a*t*s)/(1+t*s)); %Funzione di trasferimento della rete ritardatrice

margin(Kp*C_lag*P_Cruise); % traccia i diagrammi di Bode evidenziando il margine di

% fase

Il diagramma ottenuto, rappresentato in Figura 3.21, mostra un margine di fase di circa 90° , valore che garantisce la stabilità del sistema e che, essendo maggiore di 60° , soddisfa il vincolo sul coefficiente di smorzamento e quindi sulla sovraelongazione.

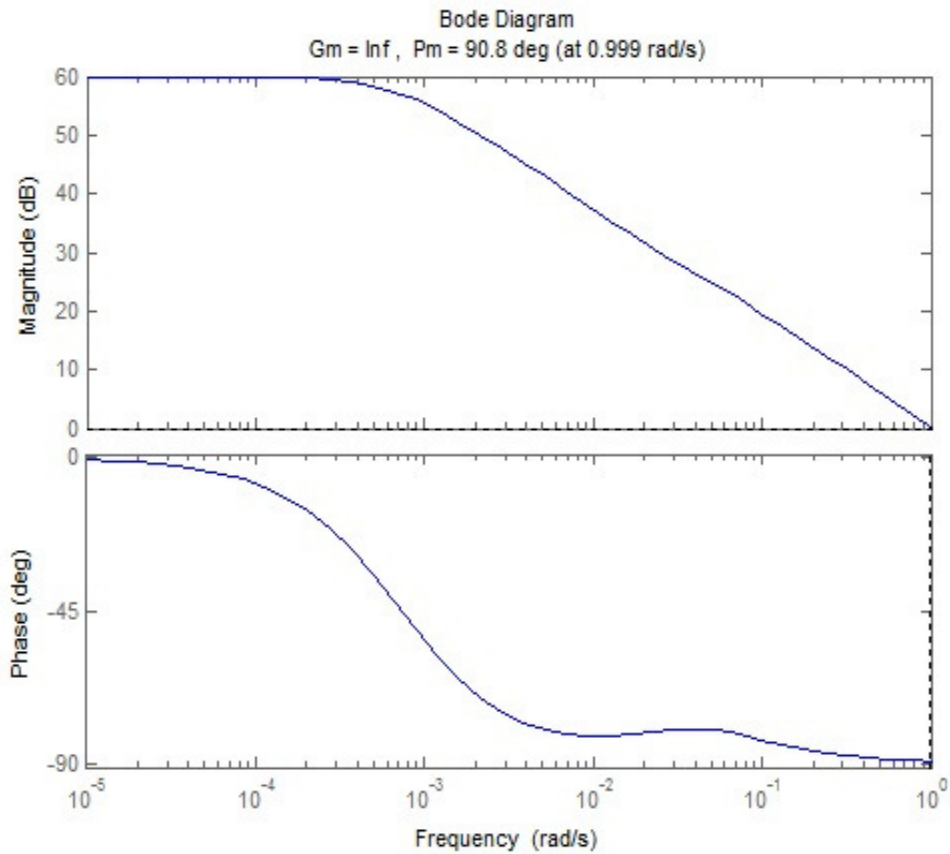


Figura 3.23: diagramma di Bode del sistema con rete attenuatrice

Tracciamo ora la risposta al gradino del sistema in catena chiusa per verificare che tutte le specifiche siano rispettate:

$r = 10;$

$T = \text{feedback}(K_p * C_{\text{lag}} * P_{\text{Cruise}}, 1);$

$\text{step}(r * T);$

Come si vede dalla risposta al gradino ottenuta in Figura 3.22, il valore a regime viene raggiunto con errore quasi nullo e con un tempo di salita di 2.29s, quindi i vincoli progettuali sono pienamente rispettati.

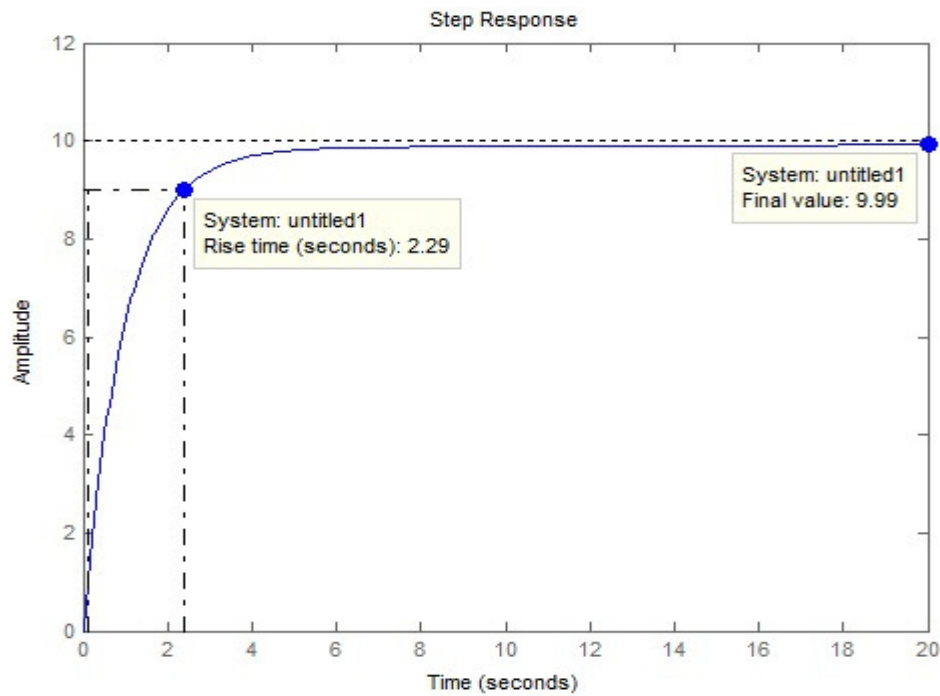


Figura 3.24: Risposta al gradino del sistema con rete attenuatrice

3.5 – Osservazioni conclusive sull’analisi in frequenza

L’analisi in frequenza ci ha permesso di verificare la BIBO-stabilità del sistema retroazionato tramite il criterio di Nyquist. In seguito, come abbiamo già constatato nel corso di questo elaborato, abbiamo visto come un controllore puramente proporzionale che soddisfi i vincoli di progetto non sia realizzabile, in quanto porta ad ottenere tempi di salita fisicamente improponibili. Come già effettuato per il luogo delle radici abbiamo risolto tale problema introducendo una rete ritardatrice, che permette di ridurre l’errore a regime pur mantenendo tempi di salita accettabili.

Capitolo 4

Simulazione mediante simulink

4.1 - Introduzione all'ambiente Simulink

Simulink è un'estensione di Matlab che permette la costruzione e la simulazione di modelli.

Tale pacchetto applicativo mette a disposizione un'interfaccia grafica che permette di rappresentare i modelli tramite diagrammi a blocchi, i cui singoli componenti possono essere estratti da una vastissima libreria di blocchi elementari. Una volta posizionate e collegate tra loro tramite tecnica "drag and drop" tutte le componenti del modello, è possibile settarne correttamente i parametri per poi avviarne la simulazione, i cui risultati sono osservabili tramite appositi blocchi visualizzatori.

In questo capitolo ci occuperemo inizialmente di costruire il diagramma a blocchi corrispondente al modello matematico che rappresenta il nostro sistema in catena aperta e di simularne il comportamento. Successivamente andremo ad arricchire il nostro modello nelle modalità che abbiamo già analizzato in Matlab nei capitoli precedenti con lo scopo di migliorare le prestazioni dinamiche del sistema.

4.2 – Simulazione del sistema in catena aperta

Per prima cosa implementiamo in Simulink il sistema in catena aperta, operazione fondamentale per poter poi progettare un qualunque tipo di controllore che migliori le prestazioni del sistema in modo da soddisfare le specifiche richieste.

Consideriamo il modello analizzato nel capitolo 1, descritto dall'equazione (1.1):

$$m \cdot \dot{v} = u - b \cdot v$$

mantenendo inalterati i valori dei parametri che vi compaiono, ovvero:

- massa $m = 1000$ Kg;
- coefficiente di attrito $b = 50$ Ns/m;
- forza di controllo $u = 500$ N.

Disegniamo ora in Simulink lo schema a blocchi corrispondente a tale modello.

Per prima cosa inseriamo il blocco centrale, che genera in uscita la velocità integrando l'accelerazione. Poiché in ingresso abbiamo però una forza u , per la seconda legge di Newton dobbiamo dividere tale forza per la massa m , in modo da ottenere l'accelerazione del veicolo ($v(t) = \int \frac{u(t)}{m} dt$).



Figura 4.1: Blocco integratore

Inseriamo poi la forza di ingresso. Volendo noi analizzare la risposta al gradino del sistema, in quanto ci permette di osservarne le prestazioni dinamiche, consideriamo come segnale di ingresso un gradino di ampiezza u . Inoltre inseriamo il blocco rappresentante le forze che si oppongono al moto, il cui effetto verrà sottratto dalla forza di ingresso tramite un blocco sommatore:

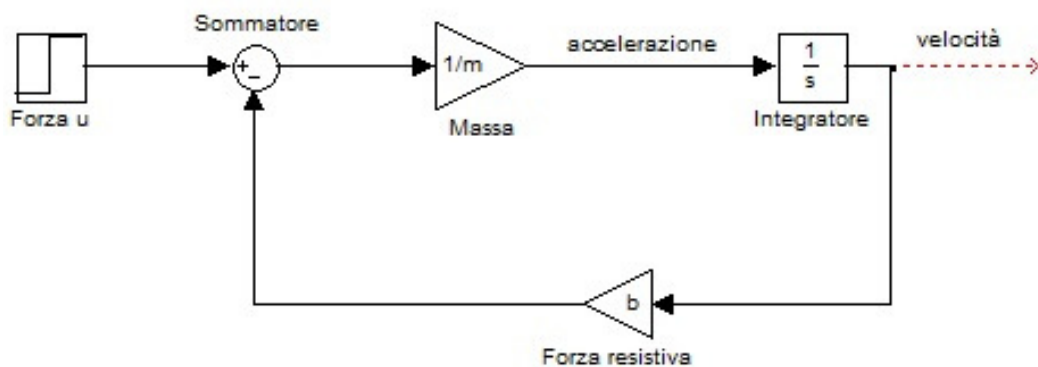


Figura 4.2: Schema a blocchi del sistema in catena aperta con ingresso a gradino

Infine inseriamo in uscita un blocco che permetta di visualizzare graficamente l'andamento della risposta al gradino del sistema in catena aperta:

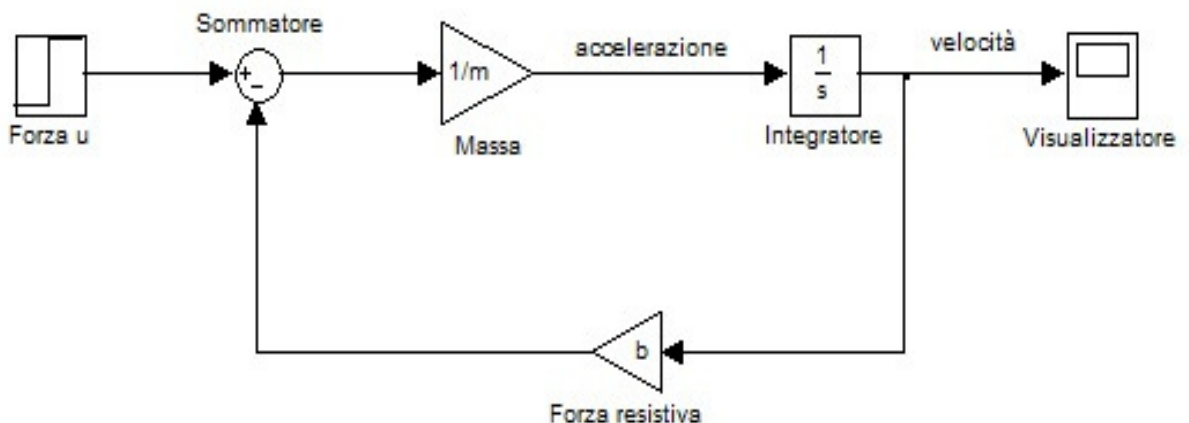


Figura 4.3: Schema a blocchi dopo l'aggiunta del blocco visualizzatore

Prima di procedere con la simulazione del modello, impostiamo come tempo di stop 120s nel menù raggiungibile in “Simulation” → “Configuration Parameters”. A questo punto inseriamo i parametri del sistema nella Command Window di Matlab ed avviamo la simulazione tramite il comando “start simulation”, ottenendo attraverso il blocco visualizzatore la risposta al gradino riportata:

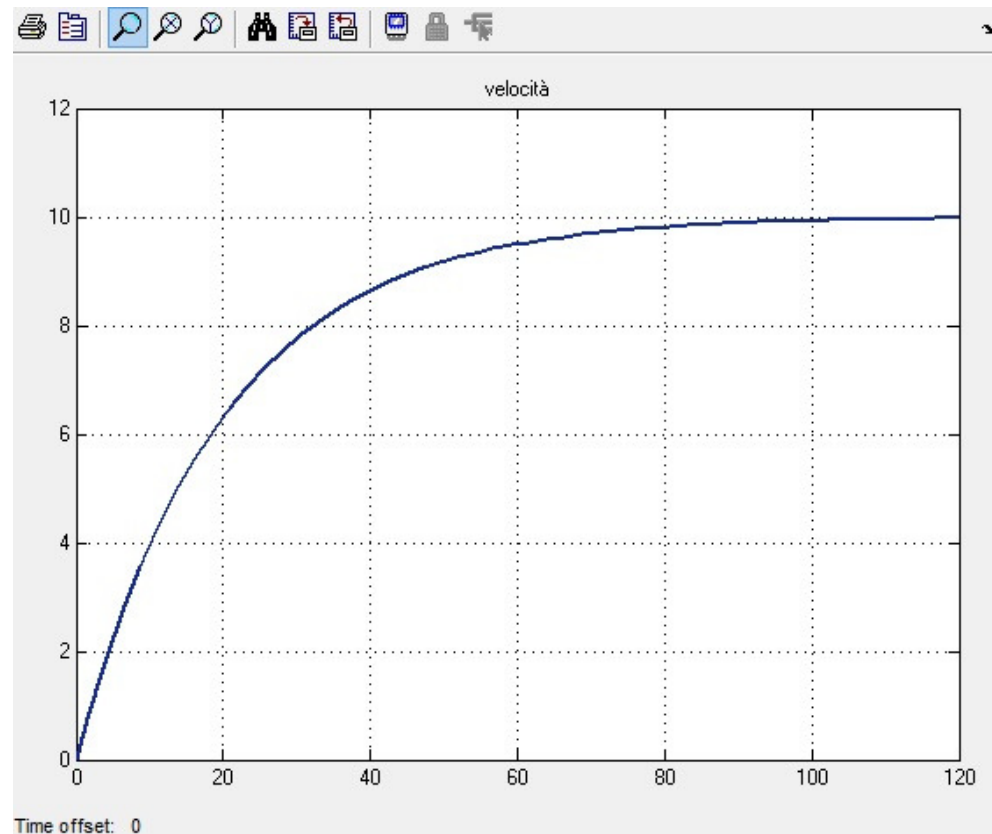


Figura 4.4: Risposta al gradino in catena aperto del sistema simulato

Il comportamento ottenuto dal sistema simulato è identico a quello ottenuto in Figura 2.1 dal calcolo della risposta ad anello aperto in Matlab nel capitolo 2. Come abbiamo già osservato, il valore a regime ottenuto è nullo, ma il tempo di salita è molto lontano dal rispettare le specifiche richieste. Per ridurre tale tempo possiamo introdurre uno dei controllori che abbiamo già progettato ed analizzato in Matlab nei capitoli precedenti, ovvero controllore P, PI, PID o rete attenuatrice.

4.3 - Creazione del sottosistema in catena aperta

Prima di proseguire con l'implementazione del sistema in catena chiusa comprensivo di controllore, risulta utile rappresentare con un unico blocco di tipo subsystem l'intero schema del sistema in catena aperta che abbiamo appena realizzato. In questo modo lo schema a blocchi del sistema complessivo retro azionato risulterà graficamente più leggero e leggibile.

Per fare ciò, modifichiamo per prima cosa gli ingressi del nostro sistema in catena aperta, sostituendo la forza u in ingresso con il blocco “In 1” ed il visualizzatore con il blocco “Out 1”, che rappresentano rispettivamente il nodo di ingresso e di uscita del sottosistema che stiamo creando.

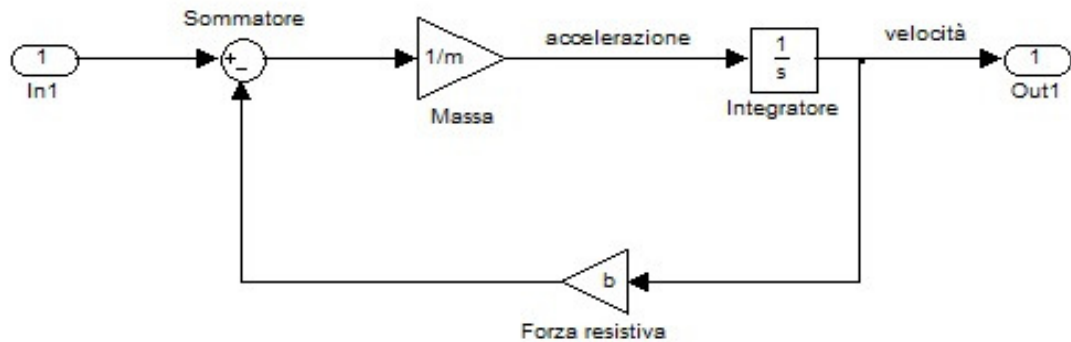


Figura 4.5: Schema a blocchi del sottosistema creato

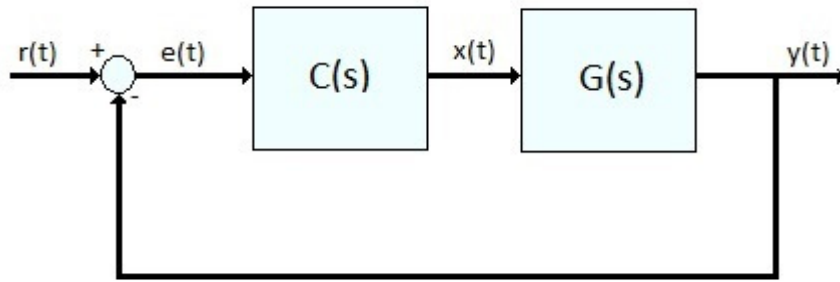
Successivamente inseriamo nella Command Window di Matlab il seguente codice:

```
m = 1000;
b = 50;
u = 500;
[A, B, C, D] = linmod('CruiseSubsystem')    %CruiseSubsystem e' il nome del file del
                                             modello simulink
Cruise_ss = ss(A, B, C, D);
```

che crea il sottosistema assegnandogli l'identificativo “Cruise_ss”.

4.4 – Simulazione del sistema in catena chiusa

Siamo ora pronti per implementare e simulare in ambiente Simulink il nostro sistema in catena chiusa comprensivo di controllore. In questa analisi ci concentreremo solamente su controllore PI e rete ritardatrice, in quanto abbiamo visto come un controllore puramente proporzionale P che soddisfi le richieste di progetto non sia fisicamente realizzabile e un controllore PID, avendo funzione di trasferimento impropria, necessita dell'aggiunta di un polo in alta frequenza che ne complica la realizzazione pratica. Nei paragrafi seguenti faremo riferimento allo schema di controllo in retroazione in Figura 3.1, che riportiamo per comodità.



Ricordiamo che $C(s)$ rappresenta la funzione di trasferimento del controllore, $G(s)$ la funzione di trasferimento del nostro sistema in catena aperta e $r(t)$ il segnale di riferimento, che nel nostro caso è la velocità che si desidera ottenere a regime in uscita. Per implementare $G(s)$ utilizziamo il blocco di tipo Subsystem che abbiamo realizzato nel paragrafo 4.3, mentre per generare la velocità di riferimento ricorriamo a un blocco Step, che fornisce un segnale a gradino.

4.4.1 - Simulazione con controllore PI

In Simulink un controllore con azione proporzionale – integrativa può essere implementato inserendo un blocco “Transfer Function”, in cui impostiamo come funzione di trasferimento la (3.6).

Lo schema a blocchi che si ottiene è il seguente:

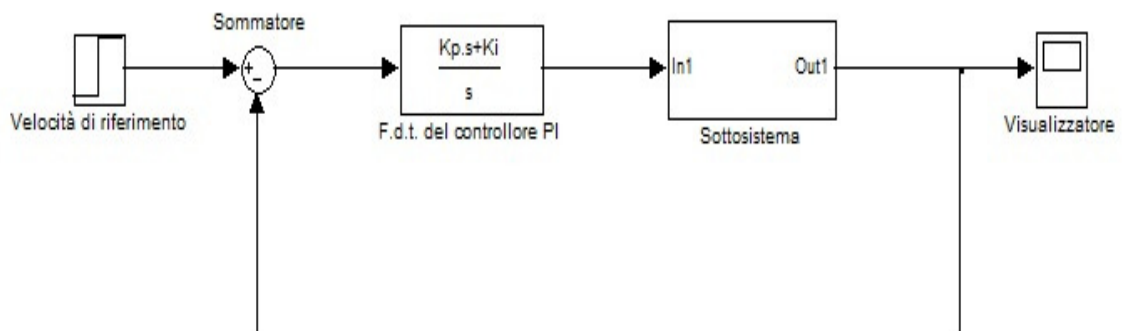


Figura 4.6: Schema a blocchi del sistema retroazionato con controllore PI

Per la simulazione utilizziamo i valori di k_i e k_p ottimali ottenuti nel paragrafo 3.2.2, che come abbiamo visto portano il sistema a soddisfare le specifiche richieste.

Dopo aver inserito i valori dei parametri del sistema e di k_i e k_p tramite i comandi Matlab:

$m = 1000;$

$b = 50;$

$r = 10;$

$k_p = 1000$;

$k_i = 50$;

avviamo la simulazione, ottenendo la risposta al gradino riportata nella figura seguente.

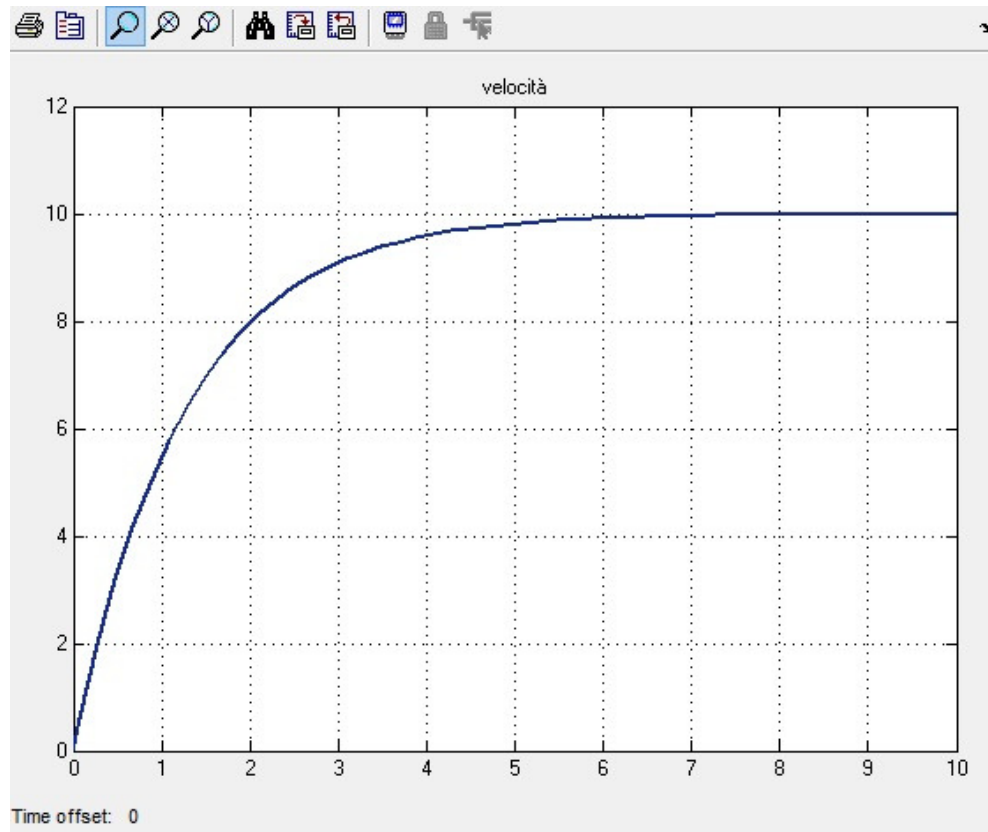


Figura 4.7: Risposta al gradino del sistema retroazionato con controllore PI

Come era prevedibile, la risposta al gradino è identica a quella che avevamo ottenuto in Figura 3.6 tramite l'analisi in Matlab e come abbiamo già discusso sono soddisfatti tutti i requisiti dinamici.

4.4.2 - Simulazione con rete ritardatrice

Come già visto in precedenza, l'introduzione di una rete ritardatrice permette di ridurre l'errore a regime del sistema retroazionato con controllore puramente proporzionale di guadagno k_p . Ricordiamo che la funzione di trasferimento generica di una rete ritardatrice è del tipo:

$$C_{lag}(s) = \frac{s + z_0}{s + p_0}$$

Per realizzare in Simulink il controllore $C(s)$ complessivo inseriamo un blocco di guadagno k_p e un blocco Transfer Function avente come funzione di trasferimento quella della rete ritardatrice.

Otteniamo quindi lo schema a blocchi complessivo riportato di seguito.

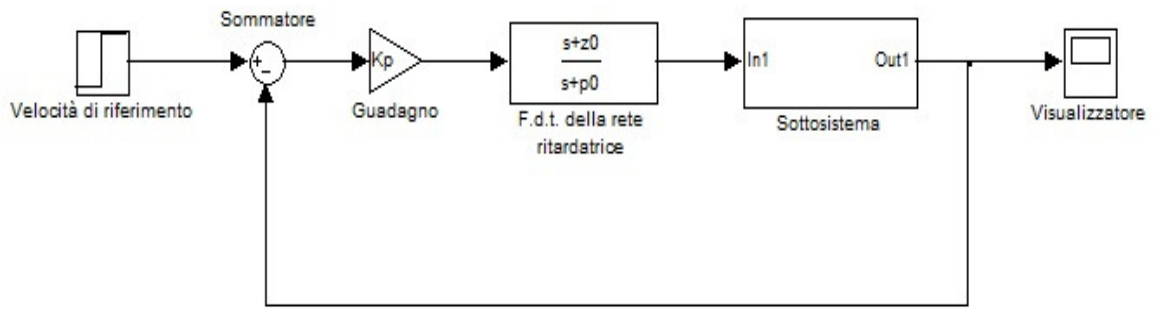


Figura 4.8: Schema a blocchi del sistema retroazionato con rete ritardatrice

Simuliamo ora il sistema utilizzando sia i valori del guadagno K_p , dello zero z_0 e del polo p_0 ottenuti attraverso lo studio del luogo delle radici nel paragrafo 3.3.

Inseriamo in Matlab tali valori:

$K_p = 1293.9$;

$z_0 = 0.3$;

$p_0 = 0.03$;

ed avviamo la simulazione, ottenendo la risposta al gradino:

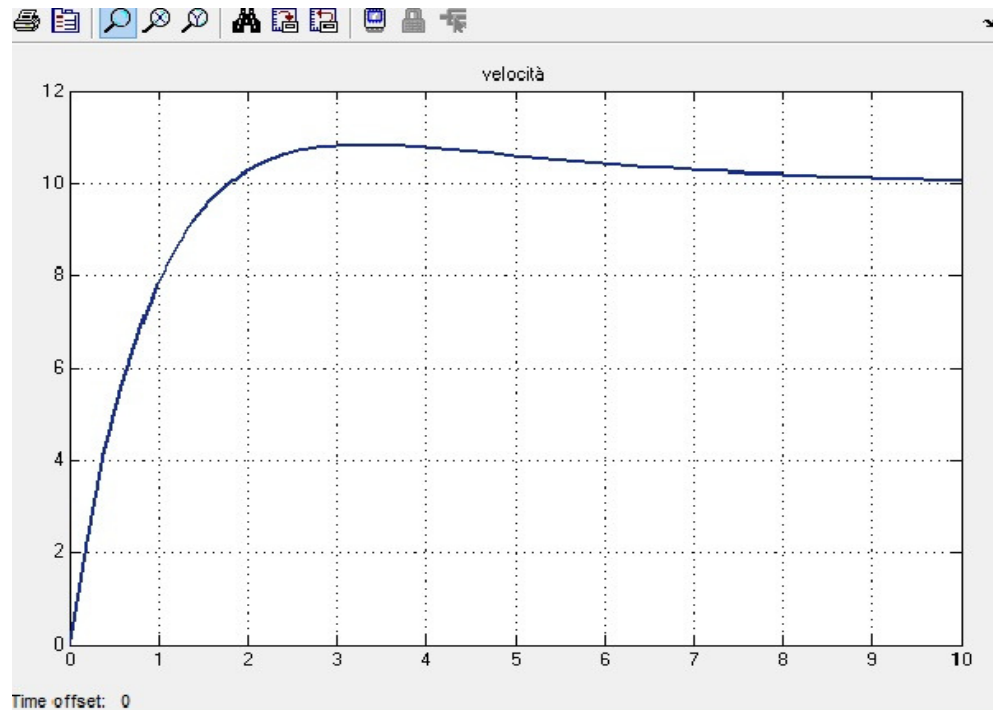


Figura 4.9: Risposta al gradino del sistema con rete ritardatrice per $K_p=1293.9$, $z_0=0.3$ e $p_0=0.03$

Com'era prevedibile la risposta al gradino è identica a quella che avevamo ottenuto in Matlab riportata in Figura 3.16. Come già osservato, abbiamo ottenuto un errore a regime nullo ed il tempo di salita soddisfa le specifiche ed è fisicamente ottenibile. Inoltre la sovraelongazione introdotta è accettabile in quanto inferiore al 10%.

4.5 – Osservazioni conclusive sulla simulazione del sistema

Tramite la simulazione in ambiente Simulink del sistema, sia in catena aperta che in catena chiusa, abbiamo confermato i stessi risultati ricavati dall'analisi in ambiente Matlab.

Osservando le risposte al gradino ottenute dalla simulazione abbiamo infatti constatato che sia l'utilizzo di un controllore proporzionale-integrativo, sia l'introduzione di un controllore proporzionale abbinato ad una rete ritardatrice, portano il sistema retroazionato complessivo a raggiungere i requisiti di progetto. Inoltre abbiamo ancora una volta osservato come la rete ritardatrice introduca una piccola sovraelongazione a causa dello zero, che risulta però inferiore al 10% e soddisfa quindi i vincoli progettuali.

Capitolo 5

Conclusioni

In questo elaborato abbiamo modellizzato, analizzato e simulato un sistema di *Cruise control*, prima in catena aperta e poi in catena chiusa con l'aggiunta di controllori appositamente progettati per migliorare le prestazioni dinamiche del sistema. Valuteremo ora i vari tipi di controllori sulla base di alcune considerazioni che si riferiscono più ad un aspetto pratico.

Per prima cosa valutiamo le prestazioni dei vari controllori esaminati, ovvero se sono in grado di soddisfare le specifiche imposte dal progetto. Come abbiamo dedotto nel corso di questo elaborato l'unico controllore a non soddisfare i vincoli imposti è il compensatore proporzionale P, in quanto comporta una risposta eccessivamente pronta che richiederebbe in fase di realizzazione impulsi di forza difficilmente compatibili con il motore di un autoveicolo. I controllori PI e PID e la rete ritardatrice, invece, ci hanno permesso di ottenere un buon comportamento dinamico del sistema, portando al soddisfacimento dei requisiti. Riepiloghiamo qui i risultati ottenuti per tali tre tipi di controllore:

Controllore	t_r	e_{regime}	sovraelongazione
PI	2.97s	0	assente
PID	2.31s	0	assente
Rete ritardatrice (luogo delle radici)	1.24s	± 0.04	8.83%
Rete ritardatrice (analisi in frequenza)	2.29s	± 0.01	assente

Ulteriore elemento di valutazione è costituito dalla semplicità di progetto, legata al costo finale del controllore. Ricordiamo che, in generale, il compensatore PID comporta qualche ulteriore passo in fase di realizzazione, dovuto all'aggiunta della componente derivativa e del polo in alta frequenza.

Bibliografia

[1] M.Bisiacco, M.E. Valcher, Controlli automatici, Libreria Progetto Padova, Padova, 2008;

[2] M.Bisiacco, S.Braghetto, Teoria dei sistemi dinamici, Esculapio, Bologna, 2010;

[3] Cruise control,
<http://ctms.engin.umich.edu/CTMS/index.php?example=CruiseControl>, consultato nel
settembre 2013