

Università degli Studi di Padova

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE
CORSO DI LAUREA IN INGEGNERIA INFORMATICA

Progetto di un sistema di controllo per sospensioni in ambiente Matlab e Simulink

Laureando:
Stefano Cecchel

Relatore:
Prof. Alessandro Beghi

ANNO ACCADEMICO 2013/2014

Indice

1	Introduzione	5
2	Analisi del sistema fisico	7
2.1	Configurazione del sistema	7
2.2	Parametri del sistema	7
2.3	Equazioni del sistema e funzione di trasferimento	8
2.4	Specifiche di controllo	10
2.4.1	Requisiti per la sospensione	10
2.4.2	Introduzione del problema in MATLAB	11
3	Il sistema in catena aperta	13
3.1	Analisi della risposta del sistema	13
3.1.1	Implementazione SIMULINK	14
3.2	Una prima analisi	16
4	Controllo in catena chiusa	19
4.1	Controllore PID	21
4.1.1	Implementazione SIMULINK	22
4.2	Luogo delle radici di Evans	24
4.2.1	Implementazione SIMULINK	29
4.3	Diagrammi di Bode	29
4.3.1	Implementazione SIMULINK	34
5	Conclusioni	37

Capitolo 1

Introduzione

Mettersi alla guida di un autoveicolo è di certo un'esperienza molto frequente nella vita di tutti i giorni. Tuttavia se il veicolo in questione è ben ammortizzato l'esperienza risulta essere ben più piacevole.

Il sistema di sospensioni di un veicolo dev'essere infatti ben progettato in quanto, come appena detto, deve attutire gli effetti che un terreno irregolare o un manto stradale sconnesso avrebbero sulla nostra guida. Inoltre esso ha l'importante compito di preservare l'integrità di molte componenti interne del mezzo che andrebbero incontro ad un deterioramento ben più repentino causato dalle eccessive e continue vibrazioni.

In fase di progettazione occorre quindi scegliere il giusto equilibrio tra morbidezza e durezza: un ammortizzatore troppo tenue comprometterebbe infatti le prestazioni di guida provocando continue oscillazioni che necessiterebbero di un lungo intervallo di tempo per attenuarsi; viceversa un ammortizzatore estremamente rigido come appena citato conferirebbe pochissimo comfort alla guida nonostante garantisca una migliore aderenza al suolo.

Nel prossimo Capitolo si analizzerà la struttura della sospensione, qui rappresentata come un parallelo tra un molla ed uno smorzatore viscoso, applicato all'equivalente in massa di 1/4 di autobus, ricavandone la funzione di trasferimento.

Successivamente se ne studierà la risposta in catena aperta, definendo delle specifiche di progetto da rispettare. Tali specifiche saranno poi soddisfatte nel Capitolo 4 in cui, con tre differenti approcci, si sintetizzeranno tre controllori (numericamente differenti ma con gli stessi effetti sul sistema) che costituiranno il valore aggiunto del nostro ammortizzatore che diventerà così una sospensione attiva in grado, tramite la configurazione in catena chiusa, di controllare il sistema. Ogni sintesi di controllore verrà corredata dalla corrispondente implementazione in SIMULINK in modo da completare così la panoramica sulla metodologia di controllo applicata (controllore PID, luogo delle radici, sintesi di Bode).

Questa tesi si pone come obiettivo non tanto la definizione matematica di un vero

controllore per sospensioni, ma piuttosto (data la natura matematica della realtà) di definire in maniera chiara e precisa un approccio basilare con il quale affrontare molti problemi di sintesi tra loro analoghi per struttura.

Capitolo 2

Analisi del sistema fisico

2.1 Configurazione del sistema

Il modello fisico che viene qui analizzato rappresenta la sospensione di un autobus su una singola ruota (anche detto *quarter-car model*). Per semplicità di conti considereremo quindi le forze agenti sul sistema esattamente $1/4$ di quelle globali dell'intero veicolo. Suddividendo il mezzo in quattro parti identiche tra loro si esclude inoltre l'analisi di ulteriori fenomeni, quali rollo (oscillazioni intorno al proprio asse longitudinale) e beccheggio (intorno al proprio asse trasversale).

Il sistema è quindi ridotto all'essenziale: una molla ammortizzante (che chiameremo K_1) posta in parallelo con uno smorzatore viscoso (b_1) costituiscono la vera e propria sospensione. Nella nostra analisi aggiungiamo due ulteriori fattori (K_2 e b_2) che rappresenteranno l'effetto attenuante/smorzante causato dalla deformazione del pneumatico stesso.

2.2 Parametri del sistema

Sono qui riportati i valori dei coefficienti prima descritti che permetteranno un'analisi fedele alla realtà d'interesse, in questo caso la sospensione di $1/4$ della massa di un autobus ordinario.

- (M_1) Massa di $1/4$ di bus = $2500kg$;
- (M_2) Massa della sospensione = $320kg$;
- (K_1) Costante elastica del sistema di sospensione = $80000N/m$;
- (K_2) Costante elastica del pneumatico = $500000N/m$;
- (b_1) Costante di smorzamento del sistema di sospensione = $350Ns/m$;

- (b_2) Costante di smorzamento del pneumatico = $15020Ns/m$;
- (U) Forza applicata dal controllore che verrà progettato.

Non è stata presa in considerazione la massa del pneumatico in quanto poco influente sulla dinamica della struttura. Viene qui riportato una rappresentazione del modello semplificato di cui verrà studiato il comportamento.

Model of Bus Suspension System (1/4 Bus)

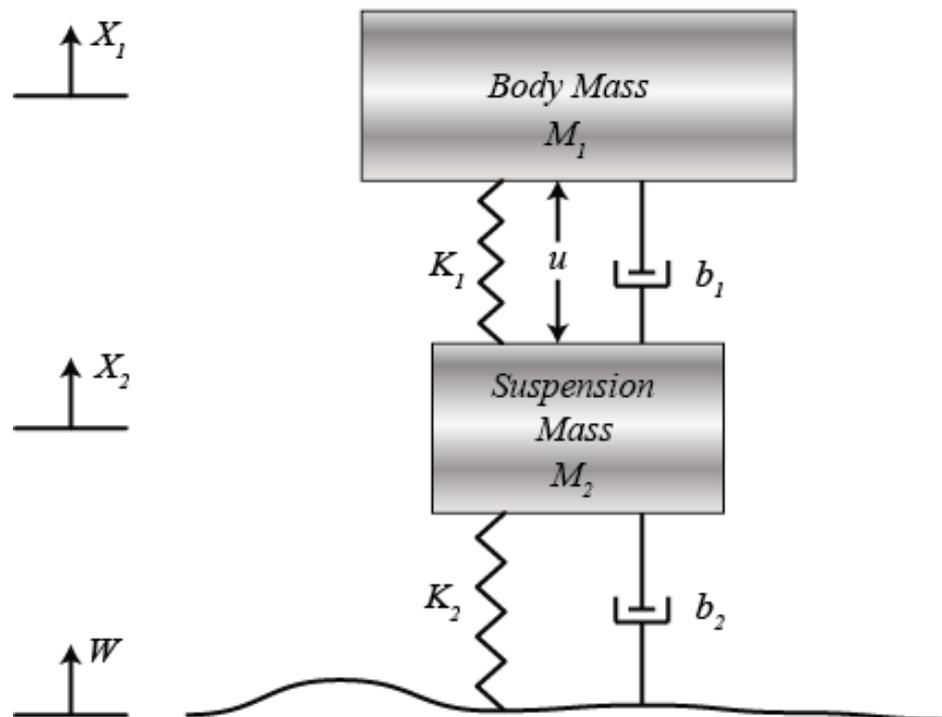


Figura 2.1: Modello fisico di una sospensione

2.3 Equazioni del sistema e funzione di trasferimento

In primo luogo occorre analizzare il sistema fisico ricavandone le equazioni della dinamica che lo descrive. Analizzando la Figura 2.1 scriviamo le due leggi di Newton applicate alle due masse M_1 ed M_2 :

$$M_1\ddot{X}_1 = -b_1(\dot{X}_1 - \dot{X}_2) - K_1(X_1 - X_2) + U \quad (2.1)$$

$$M_2\ddot{X}_2 = b_1(\dot{X}_1 - \dot{X}_2) + K_1(X_1 - X_2) + b_2(\dot{W} - \dot{X}_2) + K_2(W - X_2) - U \quad (2.2)$$

nelle quali W rappresenterà l'altezza del dosso incontrato dal veicolo.

Riscriviamo ora le precedenti equazioni avvalendoci della trasformata di Laplace sapendo che $\mathcal{L}[\dot{X}(t)] = sX(s) - X(0_-)$ e $\mathcal{L}[\ddot{X}(t)] = s^2X(s) - sX(0_-) - \dot{X}(0_-)$. Ciò renderà i conti più semplici passando dal dominio del tempo al dominio delle trasformate, nel quale operazioni complesse come la convoluzione di due segnali si trasformano in semplici prodotti.

Riformulando le precedenti espressioni in funzione di $X_1(s)$ e $X_2(s)$ otteniamo:

$$(M_1s^2 + b_1s + K_1)X_1(s) - (b_1s + K_1)X_2(s) = U(s) \quad (2.3)$$

$$(b_1s + K_1)X_1(s) + (M_2s^2 + (b_1 + b_2)s + (K_1 + K_2))X_2(s) = (b_2s + K_2)W(s) - U(s) \quad (2.4)$$

Ora è conveniente riassumere tali equazioni in un sistema in forma di stato per poter poi ricavarne la funzione di trasferimento tra uno degli ingressi di interesse e l'uscita.

$$\begin{bmatrix} (M_1s^2 + b_1s + K_1) & -(b_1s + K_1) \\ -(b_1s + K_1) & (M_2s^2 + (b_1 + b_2)s + (K_1 + K_2)) \end{bmatrix} \begin{bmatrix} X_1(s) \\ X_2(s) \end{bmatrix} = \begin{bmatrix} U(s) \\ (b_2s + K_2)W(s) - U(s) \end{bmatrix} \quad (2.5)$$

$$A = \begin{bmatrix} (M_1s^2 + b_1s + K_1) & -(b_1s + K_1) \\ -(b_1s + K_1) & (M_2s^2 + (b_1 + b_2)s + (K_1 + K_2)) \end{bmatrix} \quad (2.6)$$

$$\Delta = \det \begin{bmatrix} (M_1s^2 + b_1s + K_1) & -(b_1s + K_1) \\ -(b_1s + K_1) & (M_2s^2 + (b_1 + b_2)s + (K_1 + K_2)) \end{bmatrix} = (M_1s^2 + b_1s + K_1) \\ (M_2s^2 + (b_1 + b_2)s + (K_1 + K_2)) - (b_1s + K_1)(b_1s + K_1) \quad (2.7)$$

Possiamo esplicitare i conti calcolando l'inversa della matrice A , cioè A^{-1} che si ottiene semplicemente scambiando il membri sulla diagonale principale e invertendo il segno dei membri sulla diagonale secondaria dividendo il tutto per $\frac{1}{\Delta}$:

$$\begin{bmatrix} X_1(s) \\ X_2(s) \end{bmatrix} = \frac{1}{\Delta} \begin{bmatrix} (M_2s^2 + (b_1 + b_2)s + (K_1 + K_2)) & (b_1s + K_1) \\ (b_1s + K_1) & (M_1s^2 + b_1s + K_1) \end{bmatrix} \begin{bmatrix} U(s) \\ (b_2s + K_2)W(s) - U(s) \end{bmatrix} \quad (2.8)$$

Ora mettiamo in evidenza $U(s)$ e $W(s)$ così da poter poi ricavare le funzioni di trasferimento in maniera più immediata:

$$\begin{bmatrix} (M_2s^2 + b_2s + K_2) & (b_1b_2s^2 + (b_1K_2 + b_2K_1)s + K_1K_2) \\ (-M_1s^2) & (M_1b_2s^3 + (M_1K_2 + b_1b_2)s^2 + (b_1K_2 + b_2K_1)s + K_1K_2) \end{bmatrix} \begin{bmatrix} X_1(s) \\ X_2(s) \end{bmatrix} = \frac{1}{\Delta} \begin{bmatrix} U(s) \\ W(s) \end{bmatrix} \quad (2.9)$$

In prima analisi consideriamo come input del nostro sistema $U(s)$ supponendo l'assenza di sollecitazioni (per cui $W(s) = 0$) ottenendo la funzione di trasferimento $G_1(s)$:

$$G_1(s) = \frac{X_1(s) - X_2(s)}{U(s)} = \frac{(M_1 + M_2)s^2 + b_2s + K_2}{\Delta} \quad (2.10)$$

In seconda analisi supponiamo di voler analizzare la sola azione di disturbo, lasciando quindi $W(s)$ libero di agire e ponendo

$$G_2(s) = \frac{X_1(s) - X_2(s)}{W(s)} = \frac{-M_1b_2s^3 - M_1K_2s^2}{\Delta} \quad (2.11)$$

2.4 Specifiche di controllo

2.4.1 Requisiti per la sospensione

Come si può ben immaginare, le specifiche di controllo giocano un ruolo fondamentale nella definizione del problema di analisi di qualsiasi sistema dinamico. Esse infatti stabiliscono quali sono gli obiettivi che un buon controllista si pone e che intenderà raggiungere.

Nel caso delle sospensioni attive come già ricordato, lo scopo è quello di garantire un buon comfort di guida e allo stesso tempo evitare eccessive oscillazioni del veicolo: un buon compromesso tra due richieste antitetiche.

Analizzando nel dettaglio il nostro sistema dinamico, ci rendiamo conto anzitutto che la distanza $X_1 - W$ è di difficile misurazione e poiché riteniamo trascurabile la compressione del pneumatico (cioè la distanza $X_2 - X_1$) utilizzeremo come output la vera e propria compressione dell'ammortizzatore, ovvero $X_1 - X_2$.

Vogliamo progettare quindi un controllore che risponda alle seguenti specifiche relative all'output $X_1 - X_2$:

- sovrealongazione (*overshoot*) minore del 5%;
- tempo di assestamento (*settling-time*) minore di 5 secondi.

2.4.2 Introduzione del problema in Matlab

Dopo aver analizzato il problema da un punto di vista prettamente matematico, è giunta l'ora di riportare sotto forma di codice MATLAB i dati del nostro problema per poi analizzare la risposta del sistema ai vari ingressi.

Nella nostra analisi supporremo che il disturbo sia un semplice gradino di altezza 10cm. Secondo le specifiche di controllo prima riportate il nostro sistema dovrà rispondere in modo tale da avere una sovrealongazione massima di $\pm 5mm$ e attenuerà quasi completamente l'oscillazione in soli 5 secondi. Viene qui riportato il codice MATLAB che inserisce tutti i dati del nostro modello fisico sottoforma di dati con relative funzioni di trasferimento:

```

M1 = 2500;
M2 = 320;
K1 = 80000;
K2 = 500000;
b1 = 350;
b2 = 15020;

s = tf('s');
G1 = ((M1+M2)*s^2+b2*s+K2)/((M1*s^2+b1*s+K1)*(M2*s^2+(b1+b2)*s+
    (K1+K2))-(b1*s+K1)*(b1*s+K1));
G2 = (-M1*b2*s^3-M1*K2*s^2)/((M1*s^2+b1*s+K1)*(M2*s^2+
    (b1+b2)*s+(K1+K2))-(b1*s+K1)*(b1*s+K1));

```

Capitolo 3

Il sistema in catena aperta

3.1 Analisi della risposta del sistema

Dopo aver dato una sufficiente descrizione del modello fisico, dei valori dei parametri che lo descrivono e delle specifiche di controllo che si desiderano avere è giunta l'ora di analizzare la risposta del sistema in catena aperta. In tali condizioni il nostro modello non possiede alcuna retroazione, ovvero il valore di $u(t)$ non si modifica in funzione del valore dell'uscita. Uno schema dell'interazione che avviene in catena aperta è il seguente:

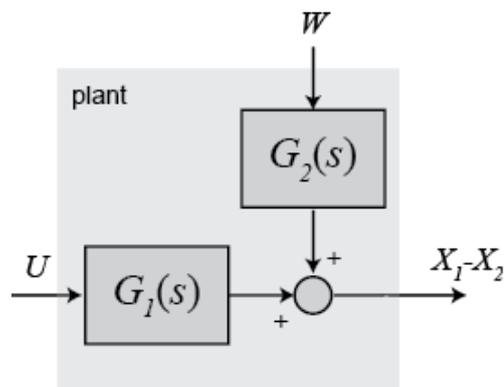


Figura 3.1: Schema a blocchi in catena aperta

Analizziamo la risposta ad un gradino di ampiezza 100 di $G_1(s)$ e successivamente la risposta ad un gradino di $0.1m$ di $G_2(s)$ tramite i seguenti comandi MATLAB:

```
figure, step(100*G1);  
grid on;
```

```
figure, step(0.1*G2);
grid on;
```

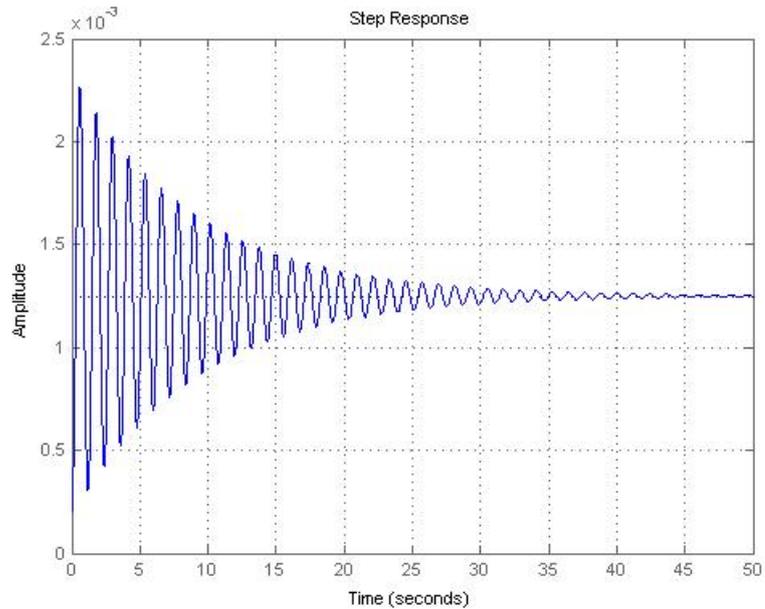


Figura 3.2: Risposta al gradino di $100 \cdot G_1(s)$

3.1.1 Implementazione Simulink

SIMULINK è senza dubbio uno dei più importanti applicativi che ci permettono di rappresentare una realtà fisica sottoforma di modello matematico, nonché di analizzarne l'evoluzione dinamica.

Il modello viene schematizzato tramite un diagramma a blocchi in cui le frecce rappresentano le relazioni tra questi ultimi.

Nel caso in questione, ovvero nell'analisi delle sospensioni di un *quarter-car model*, decidiamo di utilizzare SIMULINK per rappresentare il sistema in catena aperta tramite opportuni blocchi integratori ed un blocco derivatore i quali forniranno rispettivamente velocità e posizioni. I vari blocchi di guadagno rappresentano i coefficienti di ammortizzatori e molle opportunamente definiti in MATLAB:

```
m1 = 2500;
m2 = 320;
k1 = 80000;
```

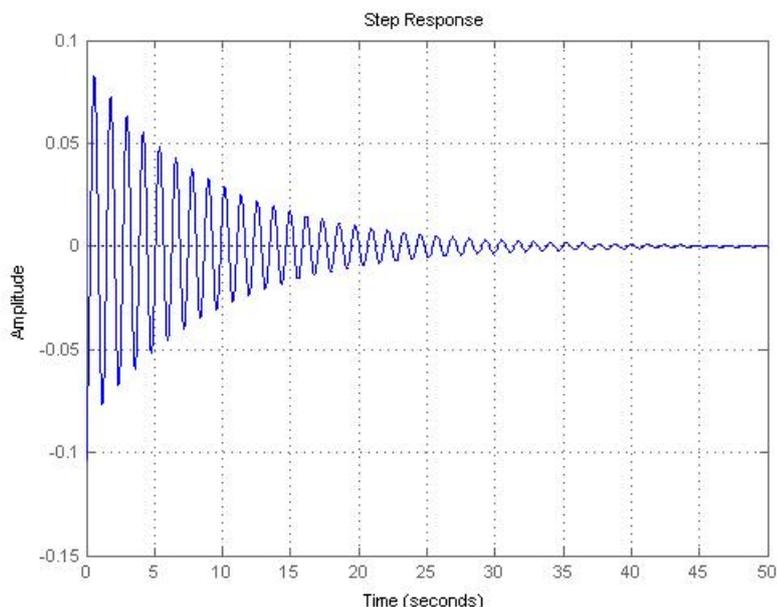


Figura 3.3: Risposta al gradino di ampiezza 10cm di $G_2(s)$

```
k2 = 500000;
b1 = 350;
b2 = 15020;
```

Avvalendoci della legge di Newton applicata a ciascuna delle due masse le uguaglianze che regolano le relazioni tra accelerazione, velocità e posizione sono le seguenti:

$$\begin{aligned} \iint \frac{d^2 x_1}{dt^2} &= \int \frac{dx_1}{dt} = x_1 \\ \iint \frac{d^2 x_2}{dt^2} &= \int \frac{dx_2}{dt} = x_2 \end{aligned} \quad (3.1)$$

Ciò che ne consegue è lo schema di Figura 3.4.

Successivamente tramite il comando `PLAY` di `SIMULINK` possiamo visualizzare grazie allo `scope` collocato all'uscita di un blocco sommatore/sottrattore l'andamento temporale in seguito ad un impulso indiciale di U della nostra uscita di interesse, ovvero $X_1 - X_2$. Il risultato è riportato in Figura 3.5 e come possiamo notare è lo stesso risultato ottenuto in `MATLAB` in Figura 3.2.

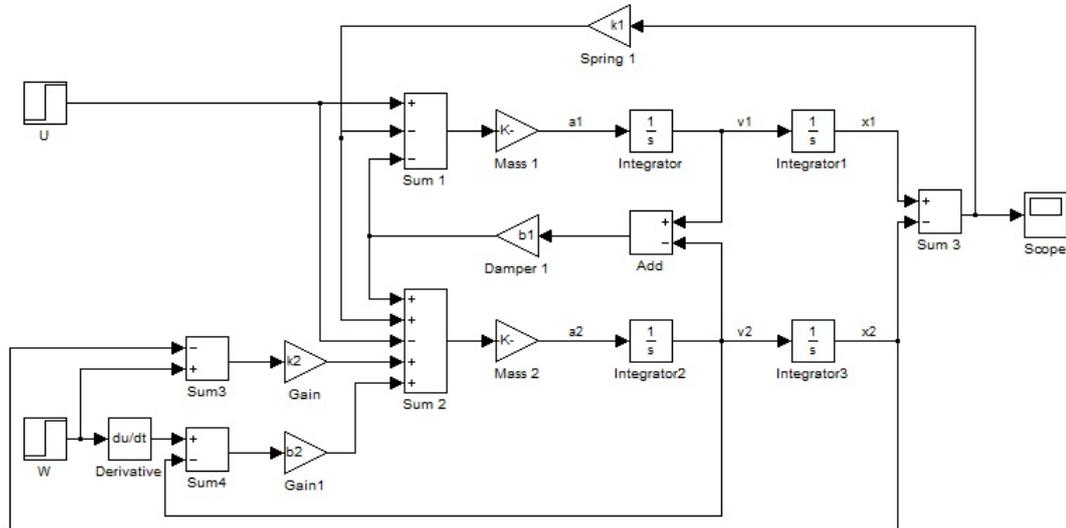


Figura 3.4: Schema a blocchi in SIMULINK

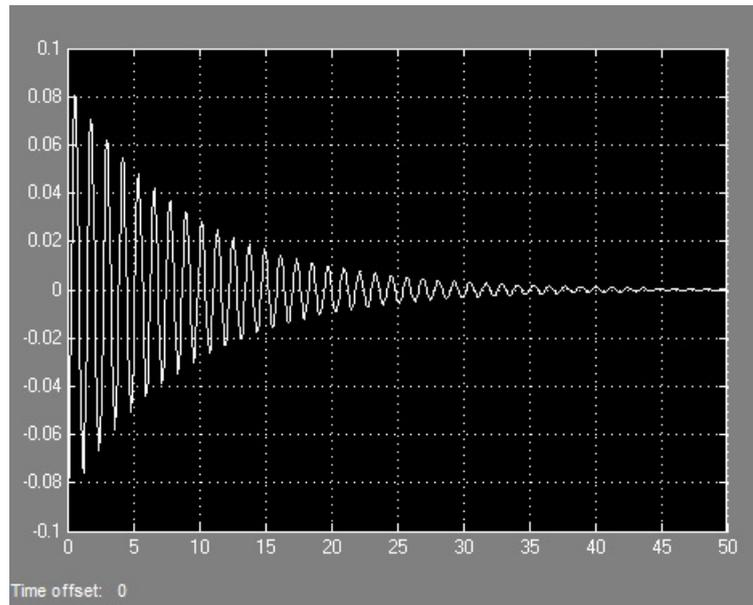


Figura 3.5: Risposta in catena aperta calcolata tramite SIMULINK

3.2 Una prima analisi

Come notiamo immediatamente dai grafici prodotti tramite MATLAB e SIMULINK, la sovralongazione risulta superare i limiti imposti ed il tempo di assestamento

risulta essere estremamente lungo (si aggira attorno ai 40 secondi). Viaggiare in un autobus con tali caratteristiche non può che risultare scomodo in quanto le oscillazioni si ripercuoterebbero a lungo sia sul conducente che sui passeggeri, per questo decidiamo di ricorrere ad un controllore che agirà in catena chiusa conferendo più comfort alla guida.

Capitolo 4

Controllo in catena chiusa

Per soddisfare le specifiche di controllo che abbiamo imposto sia sul tempo di assestamento che sulla sovraelongazione è necessario ricorrere alla retroazione in catena chiusa.

La configurazione in catena chiusa viene riportata in Figura 4.1: è stato aggiunto un controllore $C(s)$ che regolerà l'ingresso U in funzione della differenza $e = r - y$ (la sottrazione è data dalla presenza di una retroazione unitaria negativa). L'ingresso del nostro controllore $C(s)$ è infatti diventato $r - (X_1 - X_2)$ con $r = 0$, quindi $X_2 - X_1$. La potenza della retroazione sta proprio nel fatto di conoscere il valore dell'uscita e tramite questo regolare opportunamente l'ingresso.

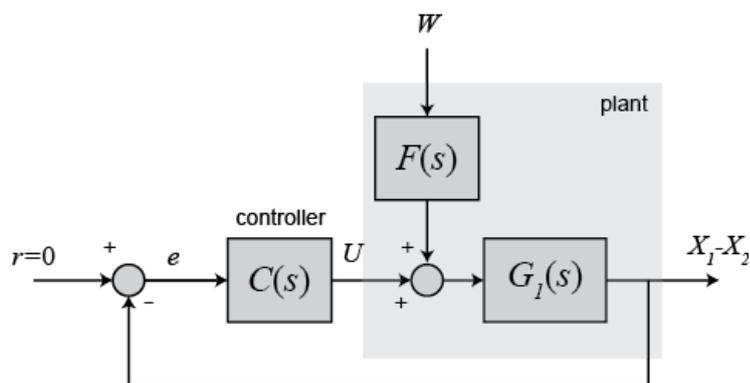


Figura 4.1: Configurazione del sistema in catena chiusa

Ricordando le Equazioni 2.10 e 2.11 del Capitolo 2 che definivano $G_1(s)$ e $G_2(s)$ applicate alla Figura 4.1 deduciamo che:

$$F(s)G_1(s) = G_2(s) \quad (4.1)$$

e volendo ricavare $F(s)$

$$F(s) = \frac{G_2(s)}{G_1(s)} \quad (4.2)$$

Possiamo ora creare un nuovo `m-file` nel quale ridefinire le variabili del sistema e creare la funzione di trasferimento di nostro interesse, ovvero $F(s)$:

```

m1 = 2500;
m2 = 320;
k1 = 80000;
k2 = 500000;
b1 = 350;
b2 = 15020;

nump=[(m1+m2) b2 k2];
denp=[(m1*m2) (m1*(b1+b2))+(m2*b1) (m1*(k1+k2))+(m2*k1)+(b1*b2)
      (b1*k2)+(b2*k1) k1*k2];
G1=tf(nump,denp);

num1=[-(m1*b2) -(m1*k2) 0 0];
den1=[(m1*m2) (m1*(b1+b2))+(m2*b1) (m1*(k1+k2))+(m2*k1)+(b1*b2)
      (b1*k2)+(b2*k1) k1*k2];
G2=tf(num1,den1);

numf=num1;
denf=nump;
F=tf(numf,denf);

```

Abbiamo utilizzato la funzione `tf()` che riceve in input due parametri, rispettivamente numeratore e denominatore della funzione di trasferimento che va a creare. Abbiamo inoltre definito `nump`, `denp`, `num1`, `den1` quali numeratori e denominatori delle funzioni di trasferimento, di $G_1(s)$ e di $G_2(s)$. Abbiamo poi ricavato $F(s)$ con la formula ricavata nell'Equazione 4.2.

La sintesi del controllore verrà effettuata con tre metodi differenti, ovvero:

- Controllore PID
- Luogo delle radici di Evans
- Risposta in frequenza

che verranno analizzati nelle prossime sezioni.

4.1 Controllore PID

P.I.D. è un acronimo che rappresenta i tre tipi di guadagno offerti da questa tipologia di controllore: Proporzionale, Integrativo, Derivativo. La duttilità di tale controllore risulta notevole potendo manipolare a piacimento ciascuna delle tre componenti per ottenere gli effetti desiderati. La struttura di un controllore PID è la seguente:

$$C(s) = K_p + \frac{K_i}{s} + sK_d = \frac{s^2K_d + sK_p + K_i}{s} \quad (4.3)$$

Le influenze di ciascun blocco sull'uscita del sistema sono schematizzate nella seguente tabella:

Coefficiente	Tempo di salita	Sovraelongazione	Tempo di assestamento	Errore a regime
K_p	Decresce	Cresce	Cambia poco	Decresce
K_i	Decresce	Cresce	Cresce	Eliminato
K_d	Cambia poco	Decresce	Decresce	Cambia poco

I tre blocchi del controllore sono posti in parallelo, affinché i loro effetti vadano a sommarsi secondo la configurazione di Figura 4.2.

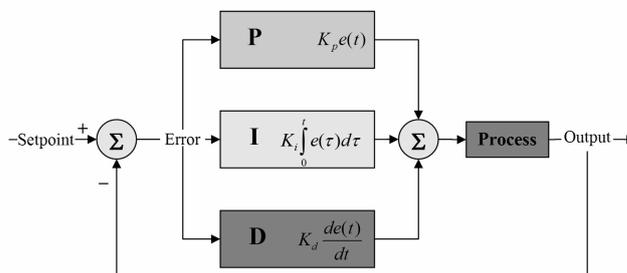


Figura 4.2: Schema di un controllore PID

Ogni blocco quindi opera per uno specifico scopo:

- K_p è un semplice controllo proporzionale al modulo dell'errore, reagisce sulla prontezza del sistema a catena chiusa. Inoltre un suo aumento, oltre a diminuire l'errore a regime ad un gradino, porta il sistema ad essere meno stabile;
- K_i annulla l'errore a regime ma tende a rendere meno stabile il sistema;
- K_d compensa le rapide oscillazioni dell'errore, aumenta lo smorzamento e consente, quindi, a pari stabilità, di aumentare l'azione proporzionale e la prontezza.

Nel nostro caso non avendo specifiche di controllo relative all'errore a regime del sistema possiamo porre a zero il coefficiente relativo al blocco integratore (cioè K_i) andando a modificare solamente i guadagni del blocco proporzionale e di quello derivatore.

Scriviamo in MATLAB il codice per creare il nostro controllore PID assegnando come valori iniziali di K_p e K_d un fattore pari a 100000:

```
Kp = 100000;
Ki = 0;
Kd = 100000;
C = pid(Kp, Ki, Kd);
sys=F*feedback(F*G1, C);
t=0:0.05:5;
step(0.1*sys,t)
grid on;
title('0.1m*Step response')
```

Nel codice il comando `pid()` crea un controllore PID con i parametri passati in input, mentre `feedback()` crea una configurazione in retroazione unitaria negativa tra le due funzioni di trasferimento impostate come argomento. Il risultato dell'esecuzione del codice è riportato in Figura 4.3. Come possiamo immediatamente notare, il problema principale (il tempo di assestamento) è stato drasticamente ridotto a 2.5 secondi, ma la sovraelongazione rimane al di sopra dei 5mm richiesti.

Facendo riferimento alla tabella riportata ad inizio Sezione, volendo apportare una modifica alla sola sovraelongazione andiamo a modificare il valore K_d . Procedendo euristicamente, cioè testando vari valori del parametro, consideriamo valida la soluzione $K_d = 70000$. Riportiamo in Figura 4.4 il risultato con tale valore.

Finalmente il risultato ottenuto è quanto richiesto dalle specifiche: *overshoot* minore di 5mm e *settling time* minore di 5s.

4.1.1 Implementazione simulink

Risolto il problema in MATLAB è possibile verificare la correttezza del risultato trovato anche tramite SIMULINK. Tramite la funzione `subsystem` è possibile comprimere tutto lo schema di Figura 3.4 in un unico blocco per il nostro diagramma. Aggiungendo i vari componenti del controllore PID otteniamo il seguente schema a blocchi: Impostando sul prompt di MATLAB i comandi di assegnazione delle costanti otteniamo il risultato di Figura 4.6.

```
Kp = 100000;
Ki = 0;
Kd = 70000;
```

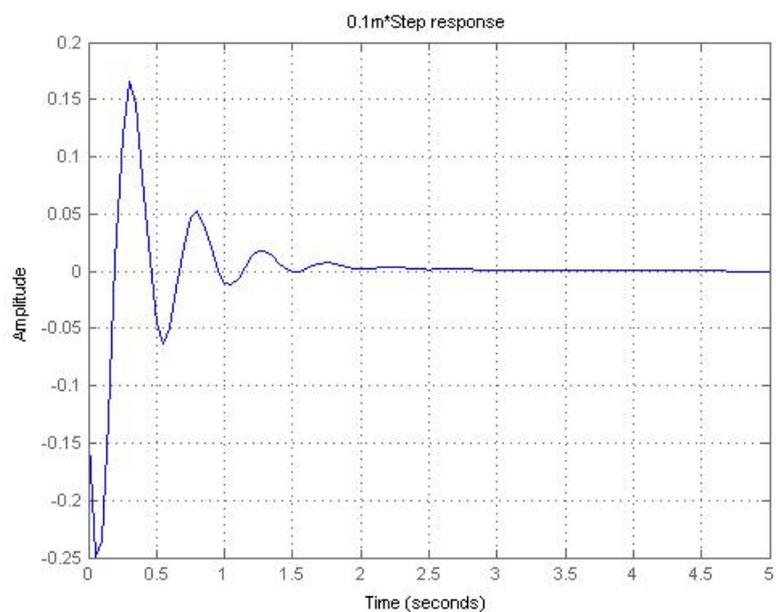


Figura 4.3: Risposta in catena chiusa a $0.10m$ con controllore PID

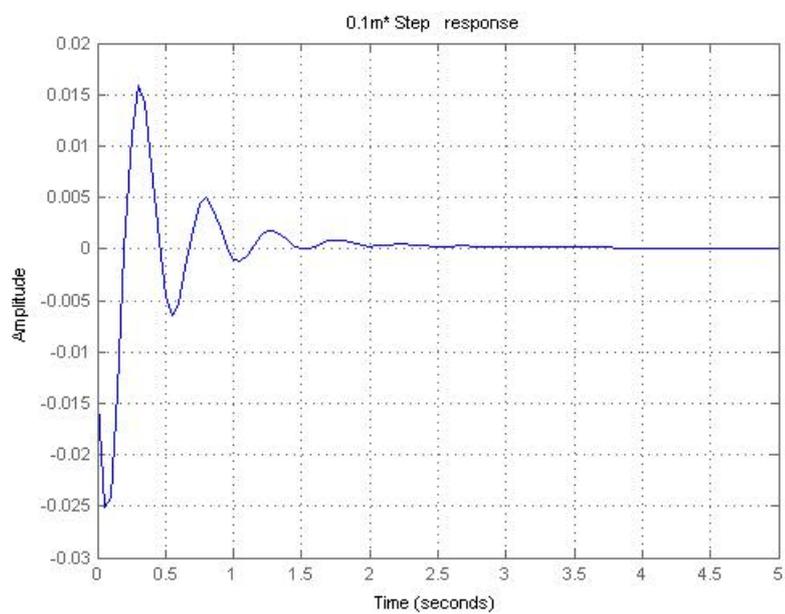


Figura 4.4: Risposta in catena chiusa a $0.10m$ con controllore PID modificato

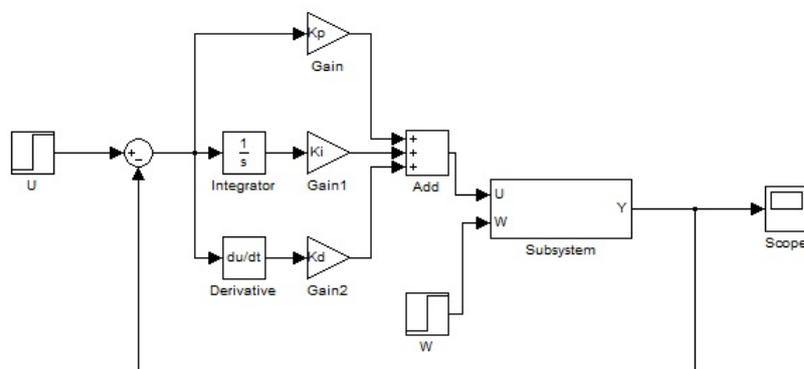


Figura 4.5: Sistema in SIMULINK con controllore PID

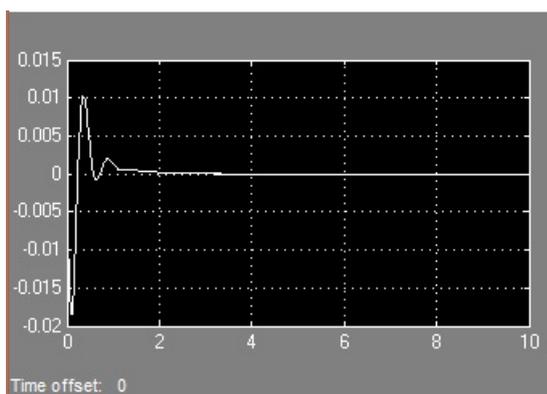


Figura 4.6: Risposta del sistema con controllore PID calcolata in SIMULINK

4.2 Luogo delle radici di Evans

Il luogo delle radici è una rappresentazione grafica sul piano complesso delle soluzioni del denominatore della funzione di trasferimento in catena chiusa del nostro sistema al variare di un fattore moltiplicativo K . Modificando tale valore infatti i poli della funzione si sposteranno in maniera continua lungo dei cosiddetti 'rami'. Il nostro obiettivo è quello di acquisire per via grafica con l'ausilio di MATLAB un buon valore di K per cui le specifiche di controllo vengano soddisfatte. Riferendoci al frammento di codice riportato ad inizio del capitolo, procediamo ricavandoci

anzitutto le radici del denominatore $denp$ della funzione di trasferimento in catena aperta tramite l'apposito comando `roots()`.

```
R = roots(denp)

R =

-23.9758 +35.1869i
-23.9758 -35.1869i
-0.1098 + 5.2504i
-0.1098 - 5.2504i
```

Dalle quattro radici notiamo che la coppia di poli dominanti è $-0.1098 \pm 5.2504i$ e come ben sappiamo poli vicini all'asse immaginario conferiscono al sistema un tempo di salita lento. Prima di procedere, ricaviamo il cosiddetto rapporto di smorzamento ζ tramite l'opportuna formula:

$$\zeta = -\log(\%OS/100)/\sqrt{\pi^2 + \log(\%OS/100)^2} \quad (4.4)$$

In tale formula $\%OS$ rappresenta appunto l'*overshoot* espresso in %, nel nostro caso $\%OS = 5$. Possiamo rappresentare ora in MATLAB il luogo delle radici tramite l'opportuno comando `rootlocus()` e tramite il comando `sgrid()` creare dei tratteggi che delimitano le aree nelle quali le specifiche di controllo sono esaurite.

```
rlocus(G1)
z=-log(0.05)/sqrt(pi^2+(log(0.05)^2))
sgrid(z,0)
```

Quanto prodotto è riportato in Figura 4.7.

Notiamo immediatamente che il luogo delle radici al variare di K non è mai contenuto nella zona di interesse. Possiamo però inserire nella nostra configurazione un cosiddetto filtro *notch* con due poli e due zeri, in modo da modificare l'andamento dei rami del luogo fino ad avere dei valori di K per i quali il nostro sistema risponderà come richiesto. Sapendo che nel grafico del luogo delle radici i rami 'escono' dai poli per 'entrare' negli zeri, proviamo ad inserire due poli in $5 \pm \frac{1}{2}i$ in modo da 'attirare' i due zeri vicini all'asse, e due zeri in -20 e -40 che faranno lo stesso per i poli della $G_1(s)$.

```
z1=5+0.5i;
z2=5-0.5i;
p1=20;
p2=40;
numc=conv([1 z1],[1 z2]);
denc=conv([1 p1],[1 p2]);
C=tf(numc,denc);
```

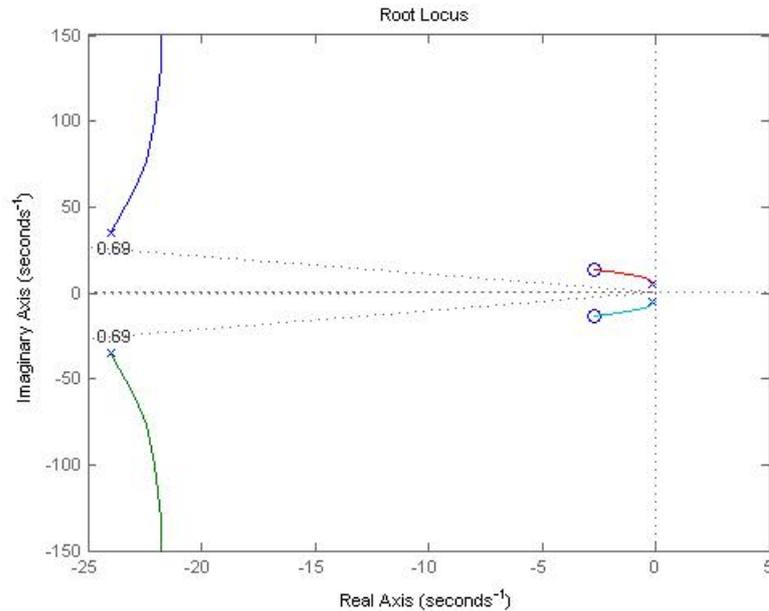


Figura 4.7: Luogo delle radici di $K \cdot G_1(s)$ in catena chiusa

```
rlocus(C*G1)
```

Il risultato è riportato in Figura 4.8.

Possiamo ora effettuare uno zoom nella zona più delicata del grafico, quella nella quale giacciono i poli dominanti. Tramite il comando `[k,poles]=rlocfind(C*G1)` possiamo con l'ausilio del puntatore selezionare un punto a piacimento del luogo delle radici e MATLAB in automatico fornirà a noi i valori di K e di tutti i poli corrispondenti a tale punto. Scegliamo in tale grafico un punto tale che massimizzi l'avvicinamento dei poli alla regione di nostro interesse: la posizione dei poli verrà indicata con un simbolo `+`.

```
[k,poles]=rlocfind(C*G1)
Select a point in the graphics window
```

```
selected_point =
```

```
    -4.2773 + 4.3789i
```

```
k =
```

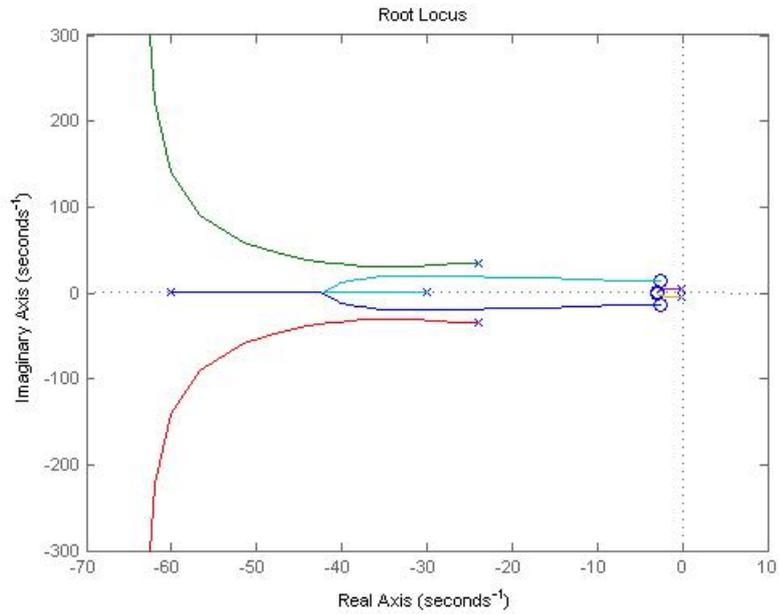


Figura 4.8: Luogo delle radici di $K \cdot G_1(s)$ con controllore $C(s)$

4.0070e+06

poles =

```

1.0e+02 *

-0.4445 + 1.2526i
-0.4445 - 1.2526i
-0.0543 + 0.1143i
-0.0543 - 0.1143i
-0.0421 + 0.0434i
-0.0421 - 0.0434i
    
```

Il risultato è rappresentato in Figura 4.9.

Non resta altro che applicare il valore $k = 4 \cdot 10^6$ al nostro controllore $C(s)$ e vedere se effettivamente il risultato è quanto atteso.

```

sys=F*feedback(G1,k*C);
t=0:0.01:2;
step(0.1*sys,t)
title('0.1m*Step response')
    
```

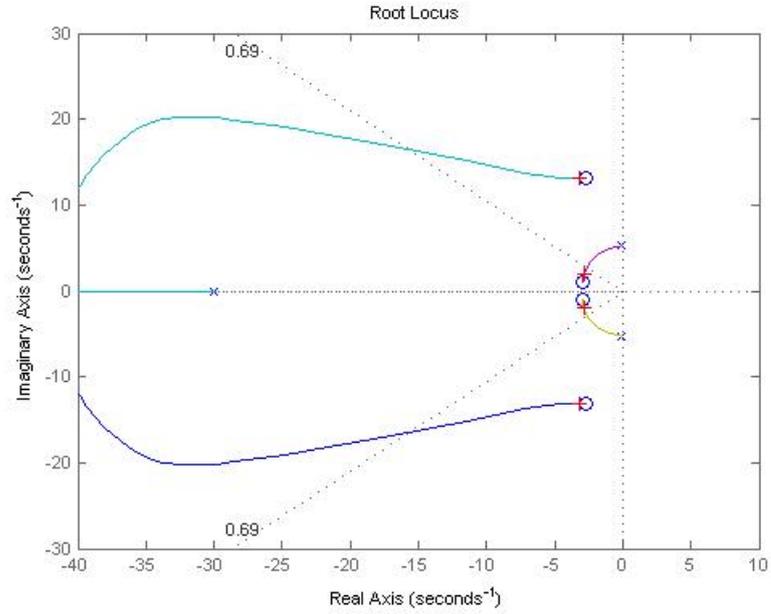


Figura 4.9: Zoom del luogo delle radici di $K \cdot G_1(s)$ con controllore $C(s)$ e posizione dei poli per K fissato

grid on

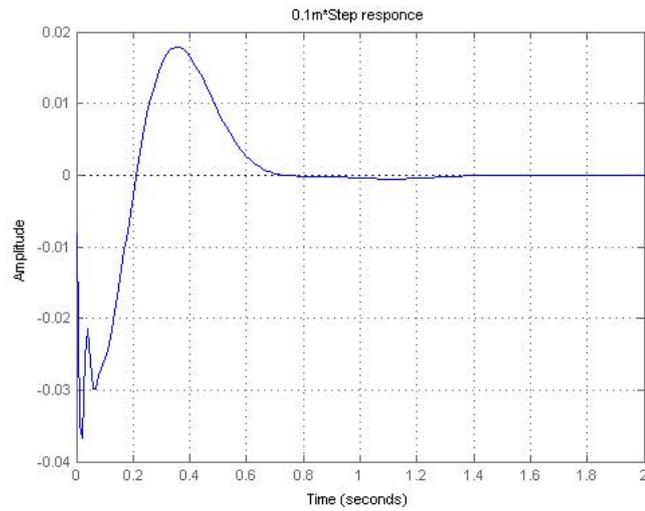


Figura 4.10: Risposta in catena chiusa a $0.10m$ con compensatore *notch*

I risultati sono soddisfacenti: in Figura 4.10 la sovraelongazione non supera di molto i $4mm$ e in circa 2 secondi vengono smorzate quasi completamente le oscillazioni.

4.2.1 Implementazione Simulink

Per implementare questo nuovo compensatore in SIMULINK calcoliamo i coefficienti della funzione di trasferimento dello stesso semplicemente svolgendo operazioni algebriche:

$$K \cdot \frac{(s + 5 - \frac{i}{2})(s + 5 + \frac{i}{2})}{(s + 20)(s + 40)} = K \cdot \frac{4s^2 + 40s + 101}{4s^2 + 40s + 3200} \quad (4.5)$$

con $K = 4 \cdot 10^6$. Creando un blocco che rappresenti la funzione di trasferimento ed un blocco di guadagno con valore K ricaviamo lo schema di Figura 4.11.

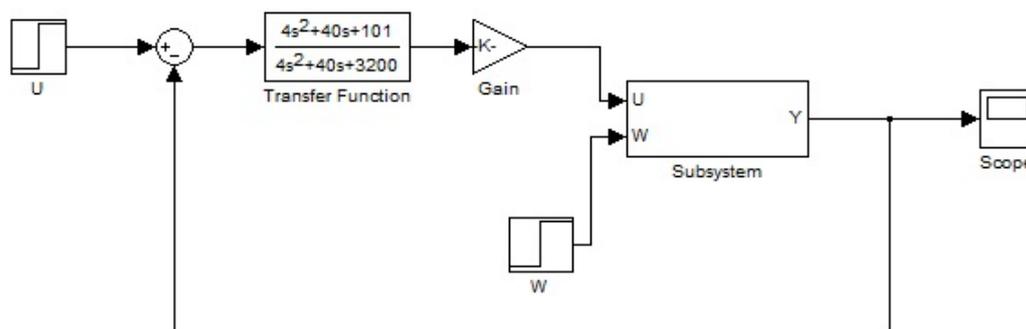


Figura 4.11: Implementazione SIMULINK del sistema con controllore calcolato tramite luogo delle radici

La risposta ad un gradino di $10cm$ viene riportata nella Figura 4.12.

La risposta del sistema è in accordo con quella prodotta in MATLAB, rispettando le specifiche di progettazione.

4.3 Diagrammi di Bode

I diagrammi di Bode sono utilizzati per rappresentare la risposta in frequenza di un sistema al variare della pulsazione ω . Consistono in due grafici che rappresentano rispettivamente l'ampiezza (o modulo) e la fase della funzione complessa. Una volta definite la funzione di trasferimento $G_1(s)$ e tutte le costanti è possibile rappresentarli in MATLAB tramite i comandi:

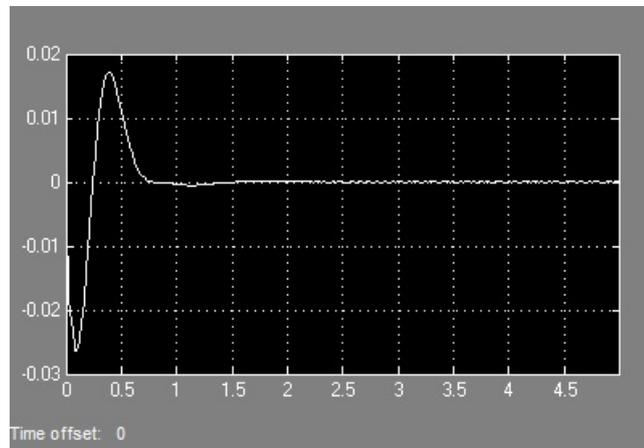


Figura 4.12: Risposta del sistema in SIMULINK controllato tramite luogo delle radici

```
w = logspace(-1,2);
bode(G1,w)
```

Ottenendo quanto riportato in Figura 4.13.

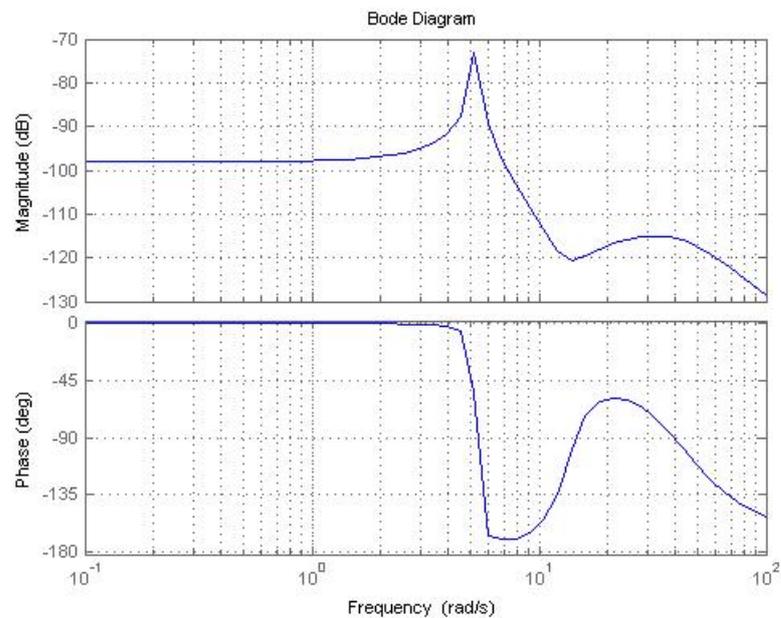


Figura 4.13: Diagrammi di Bode di $G_1(s)$

Notiamo anzitutto che non è presente una pulsazione di attraversamento (cioè un valore di ω per il quale il diagramma dei moduli valga $0dB$) per cui possiamo traslare l'intero grafico verso l'alto senza apportare modifiche all'andamento della curva aggiungendo un guadagno statico K con opportuni accorgimenti. Infatti il valore di un guadagno espresso in dB vale esattamente $K_{dB} = 20\log(K)$. Sapendo ciò e dovendo traslare verso l'alto il diagramma di quasi $100dB$, aggiungiamo in primo luogo un guadagno K pari a $10^{\left(\frac{100}{20}\right)}$ cioè 100000 .

```
K=100000;
bode(K*G1,w)
```

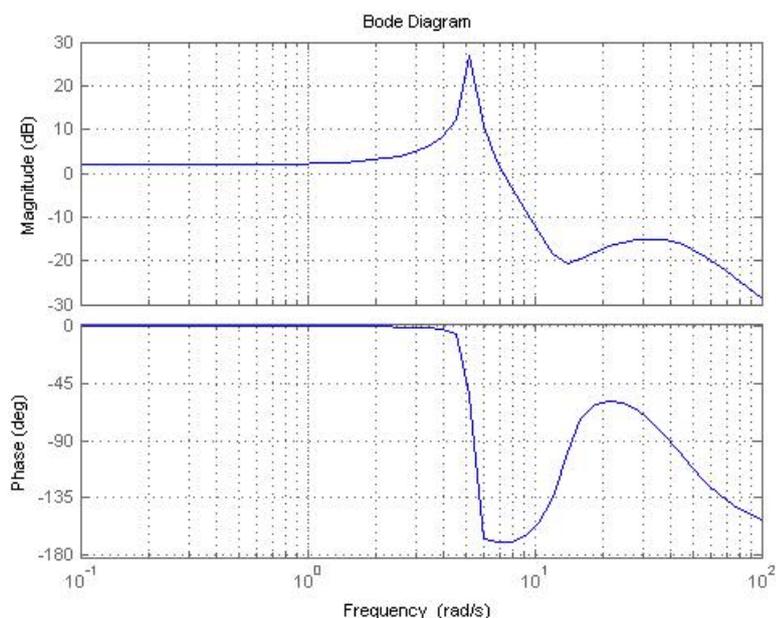


Figura 4.14: Diagrammi di Bode di $K \cdot G_1(s)$

Tramite il comando `margin(K*G1)` è possibile visualizzare il valore della pulsazione di attraversamento e del corrispondente margine di fase. In questo caso specifico $\omega_a = 7.29rad/s$ mentre $m_\phi = \pi - \angle(K \cdot G_1(\omega_a)) = 9.71deg$.

Dalla teoria sappiamo che un ampio margine di fase implica una ridotta sovraelongazione, per cui ci proponiamo di aggiungere circa 140 gradi alla corrispondente pulsazione ω_a . La via più semplice per effettuare ciò è applicare una cascata di due reti anticipatrici, le quali hanno la caratteristica di aumentare il margine di fase al massimo di 90 gradi ciascuna. La struttura di una rete anticipatrice è la

seguinte:

$$C(s) = K \cdot \frac{1 + Ts}{1 + aTs} \cdot \frac{1 + Ts}{1 + aTs} = K \cdot \frac{(1 + Ts)^2}{(1 + aTs)^2} \quad (4.6)$$

Con $T > 0$ e $0 < a < 1$. Procedendo per *step*:

- definiamo il margine di fase necessario che in questo caso sono 140 gradi, per cui avremo due controllori in serie ciascuno con guadagno 70 gradi;
- determiniamo la frequenza sulla quale vogliamo interagire aggiungendo il margine di fase, ovvero $\omega_a = 7rad/s$;
- calcoliamo infine le costanti a , T della nostra rete anticipatrice tramite opportune formule.

Il valore a viene scelto in modo da massimizzare l'utilizzo del recupero di fase offerto dalla rete anticipatrice:

$$a = \frac{1 - \sin(70^\circ)}{1 + \sin(70^\circ)} = 0.03109 \quad (4.7)$$

Determiniamo poi i valori a ed aT :

$$T = \frac{1}{\omega_a \sqrt{a}} = \frac{1}{7\sqrt{0.03109}} = 0.81012 \quad (4.8)$$

$$aT = \frac{a}{\omega_a \sqrt{a}} = \frac{\sqrt{a}}{\omega_a} = \frac{\sqrt{0.03109}}{7} = 0.02519 \quad (4.9)$$

Abbiamo così ottenuto tutti i parametri del nostro controllore che avrà quindi la seguente struttura:

$$C(s) = K \cdot \frac{(1 + Ts)^2}{(1 + aTs)^2} = 100000 \cdot \frac{(1 + s(0.81012))^2}{(1 + s(0.02519))^2} \quad (4.10)$$

In Figura 4.15 è riportato il diagramma di Bode finale, con margine di fase aumentato.

Qui di seguito viene riportato inoltre il codice MATLAB necessario alla creazione del compensatore e alla visualizzazione del grafico.

```

a = (1-sin(70/180*pi))/(1+sin(70/180*pi));
K=100000;
W=7;
T=1/(W*sqrt(a));
aT=sqrt(a)/W;
numc = conv([T 1], [T 1]);
    
```

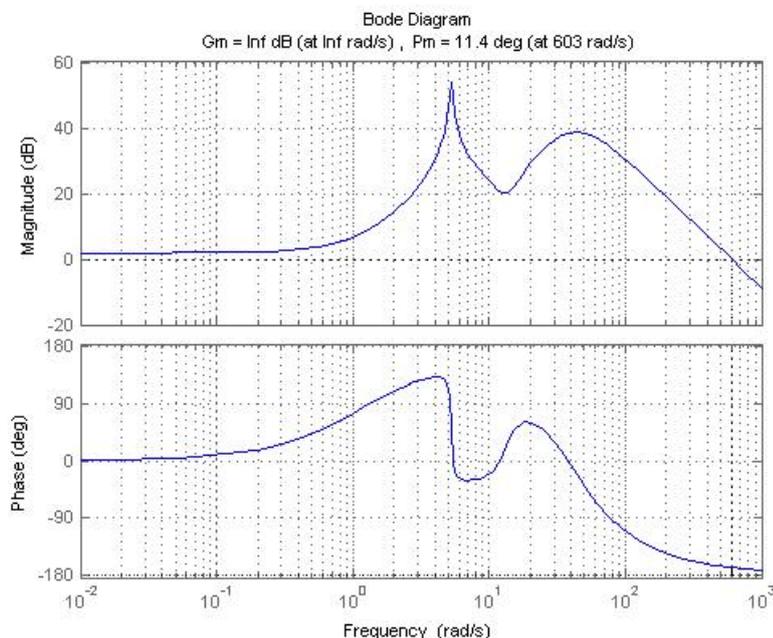


Figura 4.15: Diagrammi di Bode di $K \cdot C(s) \cdot G_1(s)$

```
denc = conv([aT 1], [aT 1]);
C = tf(numc,denc);
margin(K*C*G1)
grid on
```

Per verificare la correttezza del nostro lavoro è dunque necessario applicare il compensatore al nostro sistema in catena chiusa e verificare se la risposta indiciale rispetterà le specifiche di progettazione.

Creiamo in MATLAB il sistema in retroazione con il noto comando

```
sys = F*feedback(G1,K*C);
```

avendo già definito le funzioni F , G_1 e C in precedenza. Simuliamo ancora una volta la risposta al gradino di ampiezza $0.1m$ e visualizziamone il grafico.

```
t=0:0.01:5;
step(0.1*sys,t)
axis([0 5 -0.01 0.01])
grid on
```

In Figura 4.16 e specifiche di controllo sono soddisfatte. Per rendere il sistema più pronto alle sollecitazioni è possibile aumentarne il guadagno K del controllore

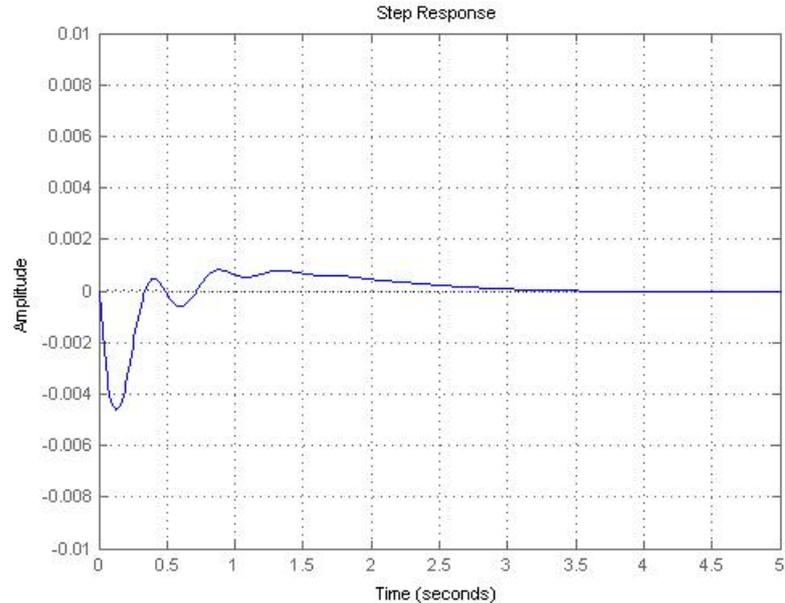


Figura 4.16: Risposta del sistema ad un gradino di ampiezza $0.1m$

per ottenere una maggiore *performance* del sistema.

Definiamo il nuovo guadagno di $C(s)$ come $6C(s)$ e visualizziamone il grafico.

```

numc = 6*conv([T 1], [T 1]);
denc = conv([aT 1], [aT 1]);
C = tf(numc,denc);
sys = F*feedback(G1,K*C);
t=0:0.01:5;
step(0.1*sys,t)
axis([0 3 -.004 .004])
grid on
    
```

Notiamo in Figura 4.17 che la sovralongazione, come atteso, è diminuita quasi al di sotto dell'1% e che il *settling time* non supera i 3 secondi. Ancora una volta, con un approccio differente, siamo riusciti a far rispettare le specifiche di controllo richieste sfruttando la retroazione negativa.

4.3.1 Implementazione Simulink

Una volta trovata la funzione di trasferimento $C(s)$ nell'Equazione 4.9, possiamo velocemente implementarla nel nostro schema SIMULINK tramite una retroazione unitaria. I parametri del blocco della funzione di trasferimento sono semplicemente

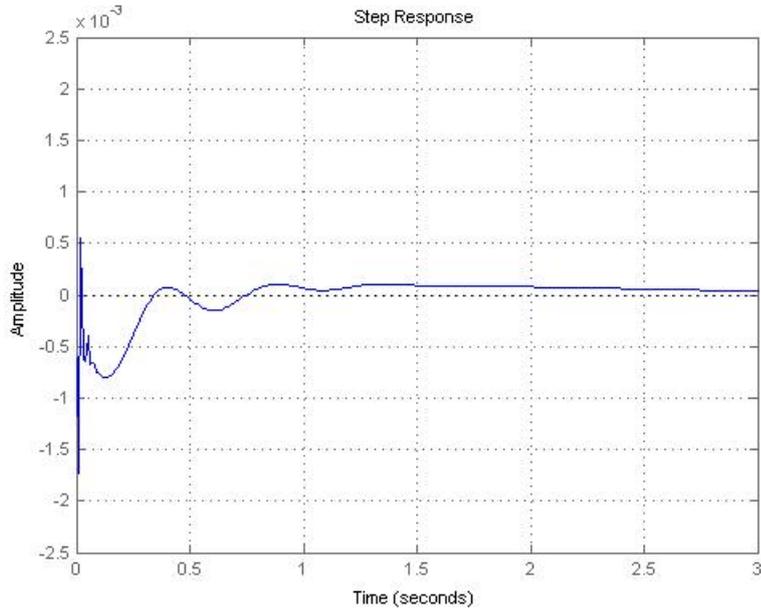


Figura 4.17: Risposta del sistema ad un gradino di ampiezza $0.1m$ con maggior guadagno

stati calcolati espandendo in polinomio la funzione $C(s)$. Nello schema inoltre è stato separato il parametro $K = 100000$, oltre che per fini grafici, per poter studiare possibili risposte del sistema variando tale parametro senza dover di volta in volta modificare tutti i coefficienti della funzione. Per calcolare la risposta ad un disturbo di $10cm$ abbiamo posto a zero l'ingresso U e inserito come valore di ampiezza di $W = 0.1$. Il risultato è riportato nella Figura 4.19.

La risposta del sistema risulta la stessa calcolata in MATLAB come da previsione.

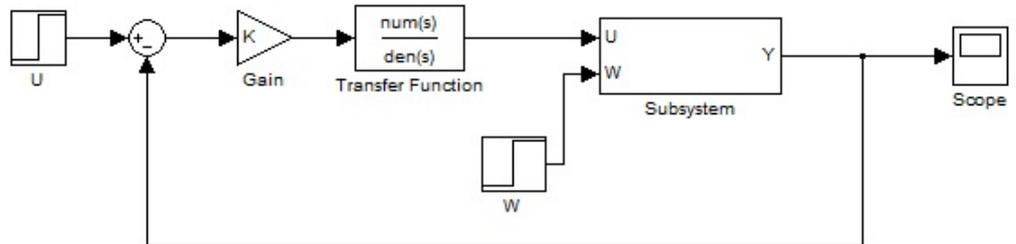


Figura 4.18: Implementazione SIMULINK del sistema con controllore calcolato tramite diagrammi di Bode

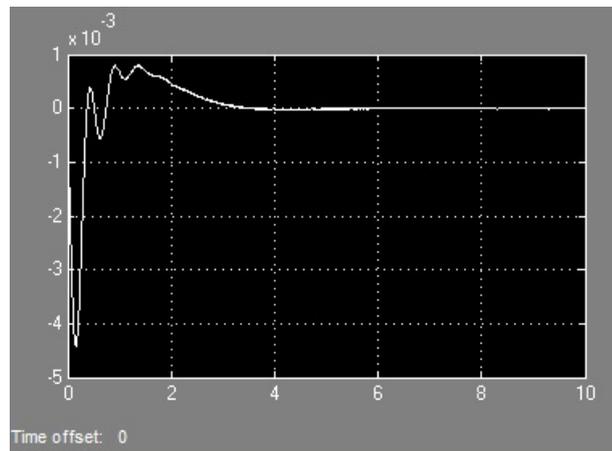


Figura 4.19: Risposta del sistema in SIMULINK con controllore calcolato tramite diagrammi di Bode

Capitolo 5

Conclusioni

Questo elaborato è stato strutturato con l'intento di fornire un primo abbozzo di sintesi di un controllore tramite tre metodologie differenti. Le specifiche di controllo in tutti i tre casi sono state rispettate e sono state fornite le linee guida per la costruzione di una funzione di trasferimento che comportasse a quegli obiettivi. Come già introdotto nel Capitolo 2, numerose sono state le approssimazioni e le assunzioni fatte in merito al sistema che è stato analizzato: anzitutto l'analisi del sistema come *quarter car model* ci ha impedito di studiare fenomeni più complessi relativi alla struttura dell'intero veicolo ed inoltre il disturbo studiato risulta essere quello di un singolo gradino di ampiezza 10cm , per questo viene a mancare un effetto realistico della risposta del sistema in quanto un fondo stradale molto sconnesso presenta numerosi affossamenti di ampiezze differenti i cui effetti, seppur attenuandosi, vanno a sommarsi nel tempo.

In questa tesi si è posto un particolare accento sulla corrispondenza biunivoca tra i codici MATLAB e i diagrammi a blocchi SIMULINK. Se MATLAB infatti ci permette di utilizzare funzioni molto importanti senza le quali non saremmo riusciti a costruire in semplici passaggi un controllore efficiente, SIMULINK ha tutto il vantaggio di poter studiare la risposta del sistema creato con un approccio molto più grafico, limitandosi alla scrittura dei coefficienti costanti senza dover ricavare dal sistema in forma di stato le equazioni differenziali che regolano lo stesso.

In ogni caso l'analisi effettuata rappresenta un'ottima base per lo studio del problema presentato ed un primo approccio ideale a molti problemi di sintesi, utile a definire delle linee generali che andranno sempre più raffinandosi. Lo scopo di un controllista è infatti questo: partendo da un problema di sintesi di controllo la soluzione finale non è ricavata immediatamente ma tramite un processo iterativo e sempre più sgrezzato dalle approssimazioni, fino a giungere ad un compromesso tra idealità, realtà e tempi di studio.

Bibliografia

- 1 Control Tutorials for MATLAB and SIMULINK. [<http://www.engin.umich.edu/class/ctms>]
- 2 Controllo a retroazione di sistemi dinamici, Volume I, G. F. Franklin, J. D. Powell, A. Emami-Naeini, EdiSES.
- 3 Dispensa di Controlli Automatici, Edizioni Libreria Progetto, Padova.