



UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

CORSO DI LAUREA TRIENNALE IN  
INGEGNERIA DELL'INFORMAZIONE

**Modellizzazione e controllo di un  
sistema Cruise Control per veicoli  
mobili**

*Relatore:*

PROF. LUCA SCHENATO

*Laureando:*

LEONARDO BIZZARO

2008879

Anno Accademico 2022/2023

Data di laurea 27/09/2023



## **Abstract**

La tesi consiste nella progettazione del sistema di controllo di un dispositivo Cruise Control per un veicolo mobile, in particolare un'automobile, il quale è in grado di far mantenere una velocità costante desiderata.

A partire dal modello, ricavato dal sistema fisico, si prosegue attraverso la progettazione di controllori PID in grado di correggerne il funzionamento in caso di disturbi esterni, al fine far rispettare certe specifiche di funzionamento prefissate.

L'obiettivo del progetto è di creare dei controllori in grado di rendere stabile il sistema che interagisce con il mondo esterno, valutandone il più efficiente.

Inoltre, si analizza il complesso dispositivo Adaptive Cruise Control, testandone il funzionamento attraverso simulazioni via software.



# Indice

<b>1</b>	<b>Analisi e modellizzazione del sistema</b>	<b>1</b>
1.1	Configurazione fisica . . . . .	2
1.2	Modello del sistema . . . . .	3
1.2.1	Funzione di trasferimento . . . . .	3
1.2.2	Spazio di stato . . . . .	4
1.3	Implementazione del modello in Matlab . . . . .	4
1.4	Analisi del sistema in catena aperta . . . . .	5
1.4.1	Risposta al gradino . . . . .	5
1.4.2	Diagramma di Bode della funzione di trasferimento . . . . .	6
1.4.3	Poli della funzione di trasferimento ad anello aperto . . . . .	7
<b>2</b>	<b>Progettazione del controllore</b>	<b>9</b>
2.1	Valutazione del sistema in catena aperta e requisiti progettuali . . . . .	9
2.2	Analisi del sistema in catena chiusa . . . . .	10
2.2.1	Introduzione al sistema in catena chiusa . . . . .	10
2.2.2	Schema del controllo in feedback e funzione di trasferimento . . . . .	10
2.3	Controllore PID . . . . .	11
2.3.1	Realizzazione del controllore su Matlab . . . . .	12
2.4	Implementazione controllore P . . . . .	13
2.5	Implementazione controllore PI . . . . .	14
2.6	Implementazione controllore PID . . . . .	17
2.7	Considerazioni sui diversi tipi di controllori . . . . .	19
<b>3</b>	<b>Simulazione del sistema su Simulink</b>	<b>21</b>
3.1	Considerazioni preliminari sul software Simulink . . . . .	21
3.2	Implementazione del modello su Simulink . . . . .	22
3.3	Simulazione del sistema in catena aperta su Simulink . . . . .	23
3.4	Simulazione del sistema in catena chiusa su Simulink . . . . .	24

---

3.4.1	Controllore P e PID . . . . .	25
3.4.2	Controllore PI . . . . .	27
3.5	Risposta del Cruise Control a salite o discese . . . . .	29
<b>4</b>	<b>Cruise Control Adattivo</b>	<b>35</b>
4.1	Introduzione al Cruise Control Adattivo . . . . .	35
4.2	Modellizzazione dei veicoli e dell'ACC . . . . .	36
4.3	Simulazione del funzionamento dell'ACC su Simulink . . . . .	37
4.3.1	Il sistema ACC mantiene la velocità impostata, con distanza minima tra le due auto soddisfatta . . . . .	38
4.3.2	Il sistema ACC mantiene la distanza di sicurezza, senza raggiungere la velocità impostata . . . . .	40
4.3.3	Il sistema ACC alterna il mantenimento della distanza di sicurezza e il raggiungimento della velocità impostata . . . . .	42
<b>5</b>	<b>Conclusione</b>	<b>45</b>
	<b>Bibliografia</b>	<b>47</b>

# Capitolo 1

## Analisi e modellizzazione del sistema

Il Cruise Control è un sistema elettronico presente nei veicoli mobili in grado di mantenere una velocità costante di riferimento (fornita dal guidatore) e sopprimere a eventuali disturbi inflitti dall'esterno. In particolare, ne esistono due tipi: *Adaptive Cruise Control* e *Cruise Control*.

Per quanto riguarda il *Cruise Control* classico, la sua funzione è di mantenere una velocità costante desiderata dal guidatore, che può essere aumentata o diminuita mediante due pulsanti facilmente raggiungibili dal diretto interessato.

Nel caso di *Adaptive Cruise Control*, il dispositivo è in grado di diminuire la velocità automaticamente in caso di presenza di ostacoli o rallentamenti del veicolo che precede. Il conducente, infatti, potrà scegliere attraverso un'adeguata strumentazione collocata nelle immediate circostanze della postazione di guida la velocità massima desiderata e la distanza minima da mantenere. Questo tipo di sistema prevede la presenza di un sensore montato nella parte anteriore della vettura (solitamente un radar) in grado di monitorare la distanza dagli ostacoli di fronte a sé.

Entrambi i sistemi prevedono la disattivazione immediata in caso di pressione del freno o per malfunzionamenti del veicolo.

Per ottenere dei dati ragionevoli, si è scelto di analizzare come veicolo un'automobile, dove l'inerzia delle ruote è trascurabile, assumendo che l'attrito con il suolo sia la forza che contrasta il moto della massa.

## 1.1 Configurazione fisica

Consideriamo il sistema precedentemente descritto, dove l'automobile è rappresentata attraverso una massa  $m$ , posizionata per semplicità su un piano perfettamente orizzontale.

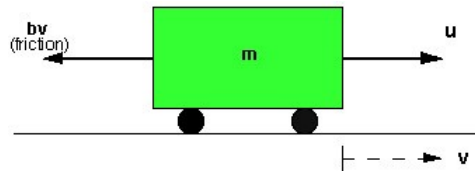


Figura 1.1: Sistema fisico

Lungo l'asse verticale, la forza peso  $F_P = mg$  viene compensata completamente dalla reazione vincolare data dal piano in cui avviene il movimento dell'automobile, dunque non sarà presente alcun contributo da parte dell'asse verticale nelle equazioni che regolano il moto del veicolo.

Lungo l'asse orizzontale, invece, sono presenti due forze che si contrastano:

- **Forza motrice**  $[N]$ , fornita dal motore, che verrà definita  $u$  in quanto ingresso del sistema di controllo.
- **Forza di attrito viscoso**  $[N]$  definita  $F_A = bv$ , dove  $v$   $[m/s]$  rappresenta la velocità del veicolo, mentre  $b$   $[Nsec/m]$  è il coefficiente di attrito viscoso che dipende dal materiale. Essa è causata dal contatto dell'automobile con la superficie sottostante durante il moto.

Dalla seconda legge di Newton Possiamo ricavare la seguente equazione di moto:

$$F_{TOT} = ma = u - bv \quad (1.1)$$

Per impostare dei valori verosimili di queste grandezze, si sono scelti i seguenti parametri:

$$\begin{aligned} m &= 1000kg \\ b &= 50Nsec/m \\ u &= 500N \end{aligned}$$



## 1.2 Modello del sistema

Per poter descrivere il modello del sistema a partire dallo schema fisico ricavato in precedenza vengono utilizzati i metodi illustrati durante il percorso di studi, necessari inoltre all'elaborazione Matlab e alla simulazione Simulink. Questi ci permettono di avere una relazione ingresso-uscita a tempo continuo, rispetto alla rappresentazione deterministica che ci dava la legge di Newton 1.1.

Le equazioni del modello si ricavano in modo immediato

$$\begin{cases} m\dot{v} + bv = u \\ y = v \end{cases} \quad (1.2)$$

dove, come indicato in precedenza, l'ingresso del sistema è la forza motore  $u$ , mentre come uscita si considera la velocità dell'automobile  $v$ , in quanto variabile destinata ad inseguire il riferimento fornito.

### 1.2.1 Funzione di trasferimento

Per trovare la funzione di trasferimento dell'intero sistema, è necessario ricorrere alla trasformata di Laplace dalle equazioni del modello 1.2. Come conseguenza di questo, possiamo assumere le **condizioni iniziali** del sistema **nulle**.

Le corrispondenti equazioni nella forma di Laplace sono le seguenti:

$$\begin{cases} msV(s) + bV(s) = U(s) \\ Y(s) = V(s) \end{cases} \quad (1.3)$$

Dove si può eseguire una semplice sostituzione dalla seconda equazione ottenendo una relazione diretta ingresso-uscita:

$$msY(s) + bY(s) = U(s) \quad (1.4)$$

Da cui si può ricavare in modo immediato la funzione di trasferimento del sistema:

$$W(s) = \frac{Y(s)}{U(s)} = \frac{1}{ms + b} \quad (1.5)$$

## 1.2.2 Spazio di stato

Per ottenere in modo esplicito la relazione ingresso-uscita di un modello, è possibile rappresentare il sistema attraverso un modello generale in spazio di stato del tipo:

$$\begin{cases} \dot{x}(t) = Fx(t) + Gu(t) \\ y(t) = Hx(t) + Ju(t) \end{cases} \quad (1.6)$$

dove le varie componenti sono elencate di seguito in modo generale:

- **F** *matrice dinamica*
- **G** *matrice di ingresso*
- **H** *matrice di uscita*
- **J** *matrice legame diretto ingresso-uscita*

Nel caso del sistema Cruise-Control preso in analisi le matrici sovraelencate corrispondono a dei coefficienti ricavabili in modo immediato dalle equazioni di modello 1.2, in modo esplicito:

$$\begin{aligned} F &= -b/m \\ G &= 1/m \\ H &= 1 \\ J &= 0 \end{aligned}$$

## 1.3 Implementazione del modello in Matlab

Si sceglie di utilizzare il software *Matlab* per elaborare e simulare il modello appena definito, a partire dal sistema fisico di figura 1.1 e dai parametri di partenza impostati.

```
% Definizione parametri del sistema
m = 1000; %[kg] massa del veicolo
b = 50; %[Ns/m] coefficiente di attrito viscoso
u = 500; %[N] forza motrice
r = 10; %[m/s] velocità di riferimento
```

**Figura 1.2:** Definizione parametri

Per quanto riguarda l'implementazione della funzione di trasferimento 1.5, si ricorre al comando *Matlab* **tf** che permette di definire un sistema a tempo continuo

a partire dalla sua funzione di trasferimento, la quale viene caratterizzata in modo esaustivo da numeratore e denominatore.

```
W = tf(1, [m b]) %funzione di trasferimento
W =
      1
-----
1000 s + 50
```

Figura 1.3: Definizione funzione di trasferimento

Per la definizione del modello in spazio di stato 1.6 è necessario l'utilizzo della routine `ss`, che permette la creazione del modello a partire dalle 4 matrici specifiche del sistema.

```
% Matrici del modello
F = -b/m; % matrice dinamica
G = 1/m; % matrice di ingresso
H = 1; % matrice di uscita
J = 0; % matrice legame diretto ingresso-uscita
```

Figura 1.4: Definizione matrici

A questo punto il passaggio successivo è immediato, grazie al comando sovraccitato

```
% Creazione del modello a partire dalle matrici
state_space = ss(F, G, H, J);
```

Figura 1.5: Definizione matrici

## 1.4 Analisi del sistema in catena aperta

Per poter osservare la risposta del sistema all'ingresso desiderato (nel caso preso in considerazione  $u = 500N$ ) si prosegue ad analizzare la risposta al gradino del sistema in catena aperta, valutare i poli della funzione di trasferimento 1.5 ed infine lo studio del diagramma di Bode.

### 1.4.1 Risposta al gradino

Grazie all'utilizzo di *Matlab*, è possibile visualizzare graficamente la risposta del sistema a un riferimento a gradino di  $500N$ , mediante la routine `step`. Il grafico che si ricava è il seguente:

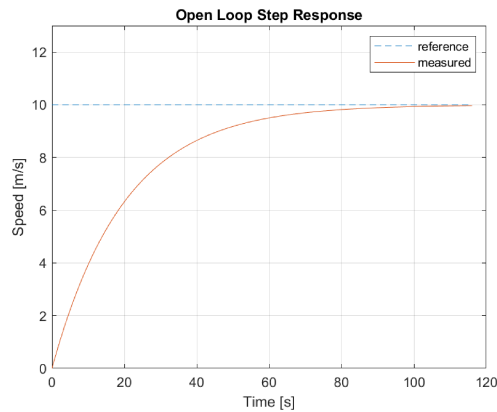


Figura 1.6: Risposta al gradino

Come si nota dalla figura 1.6, il modello matematico ricavato nelle sezioni precedenti è corretto, ma l'eventuale presenza di errori in uscita dal sistema o disturbi esterni comporterebbe una variazione significativa nei risultati desiderati. Inoltre, si noti che con una forza motrice di  $500N$  la velocità desiderata di  $10m/s$  viene raggiunta in circa  $90s$ .

### 1.4.2 Diagramma di Bode della funzione di trasferimento

Per visualizzare il comportamento del sistema Cruise Control nel dominio della frequenza è sufficiente utilizzare la routine *bode* del *Control System Toolbox*, il quale è in grado di fornire un diagramma di Bode dettagliato della funzione di trasferimento 1.5:

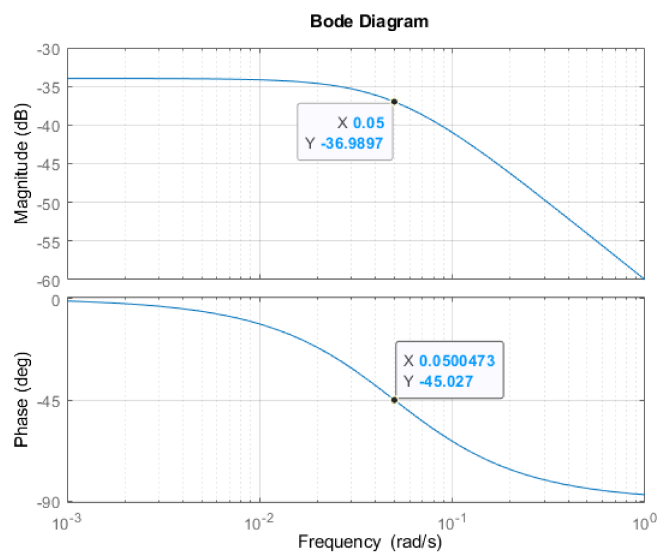


Figura 1.7: Diagramma di Bode

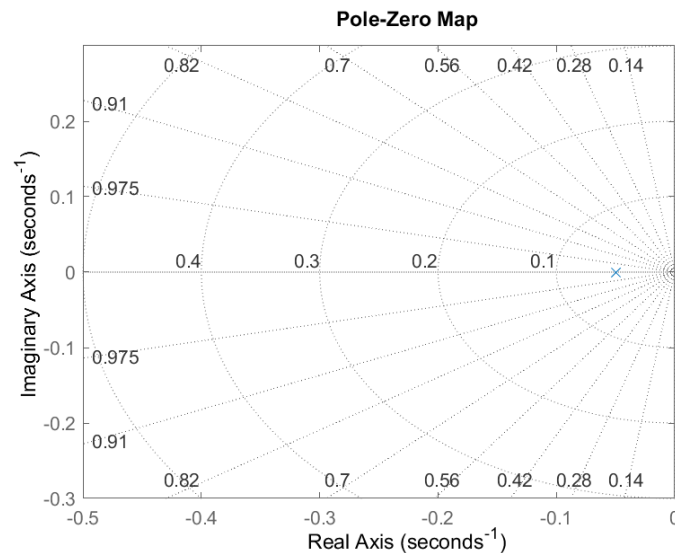
Si può notare che è presente un'attenuazione di  $3dB$  alla frequenza  $5 \cdot 10^{-2} rad/s$ ,

pulsazione alla quale la fase corrispondente è di circa  $-45^\circ$ . Per valori elevati di frequenze di lavoro, si noti che il modulo del diagramma di bode ha una pendenza di  $-20dB/dec$ .

### 1.4.3 Poli della funzione di trasferimento ad anello aperto

Per la rappresentazione di poli e zeri di una funzione di trasferimento, su *Matlab* è possibile ricorrere al comando *pzmap*, da cui è possibile osservare in modo chiaro le posizioni dei punti nel piano complesso.

In particolare, nella funzione di trasferimento del sistema 1.5 è presente un solo polo, di valore specifico  $-b/m$ .



**Figura 1.8:** Posizione dei poli della funzione di trasferimento

Come si può notare dalla figura 1.8, l'unico polo presente (rappresentato da una croce di colore blu) è collocato alla sinistra dell'asse immaginario, dunque con parte reale **negativa**. Da ciò si deduce che il sistema preso in analisi è BIBO-stabile, condizione per cui se l'ingresso del sistema  $u$  è limitato, anche l'uscita corrispondente  $y$  avrà carattere limitato nel tempo.



# Capitolo 2

## Progettazione del controllore

### 2.1 Valutazione del sistema in catena aperta e requisiti progettuali

Se si prende in considerazione la risposta al gradino  $u = 500N$  in figura 1.6, il sistema in catena aperta risulta poco reattivo alle sollecitazioni in ingresso, infatti il tempo di salita ha un valore di  $t_r = 43.94s$  mentre il tempo di assestamento al 2% vale  $t_{s2\%} = 78.24s$  (valori ricavati mediante la routine *stepinfo* di *Matlab*). Per un'applicazione pratica, essendo impossibilitati ad agire sui parametri del sistema fisico, è necessaria la progettazione di un sistema controllore capace di velocizzare il transitorio e di soddisfare eventuali specifiche progettuali.

Per la realizzazione del controllore appena citato, si prendono in considerazione alcuni requisiti progettuali da rispettare su sovraelongazione percentuale, tempo di salita, di assestamento ed errore percentuale massimo di uscita.

Per questo modello, si è scelto di avere un tempo di salita non superiore a  $5s$ , con una sovraelongazione percentuale massima del 10%. Per quanto riguarda l'errore percentuale a regime rispetto all'ingresso, si chiede un valore minore al 2%, con un tempo di assestamento al 2% massimo di  $15s$ . Infine, si è ipotizzato che l'automobile parta da ferma, con una velocità iniziale  $v(0) = 0m/s$ .

In modo sintetico:

$$\begin{aligned}t_r &< 5s \\t_{s2\%} &< 15s \\M_p &< 10\% \\e_r &< 2\%\end{aligned}$$

## 2.2 Analisi del sistema in catena chiusa

### 2.2.1 Introduzione al sistema in catena chiusa

In un modello di un sistema fisico, come quello in figura 1.1, si introduce la retroazione (o *feedback*), ovvero un meccanismo attraverso il quale una parte dell'output del sistema viene reinserita nel sistema stesso come input. Questo input retroazionale è utilizzato per influenzare e controllare il comportamento futuro del sistema.

Questo tipo di soluzione ha diversi scopi e vantaggi non secondari:

- **Controllo del sistema:** La retroazione è spesso utilizzata per controllare un sistema in modo da raggiungere un certo obiettivo o mantenere le sue variabili sotto controllo. Essa consente di rilevare differenze tra il comportamento attuale del sistema e il comportamento desiderato, quindi apporta le correzioni necessarie per avvicinarsi ai requisiti prefissati.
- **Stabilità:** La retroazione può migliorare la stabilità di un sistema. Quando un sistema diventa instabile, la retroazione può intervenire per correggere questa situazione e riportarlo a uno stato stabile.
- **Adattamento:** La retroazione può aiutare un sistema a adattarsi a cambiamenti dovuti a disturbi esterni o interni, consentendo di mantenere il funzionamento ottimale nonostante le variazioni delle condizioni di lavoro.
- **Riduzione dell'errore:** Con la retroazione, il sistema può ridurre l'errore tra l'output desiderato e l'output reale, migliorando così la precisione del sistema.

### 2.2.2 Schema del controllo in feedback e funzione di trasferimento

Nel caso più generale, si consideri un sistema dinamico con retroazione unitaria negativa, dove il processo è indicato con  $G(s)$  e il controllore con  $C(s)$

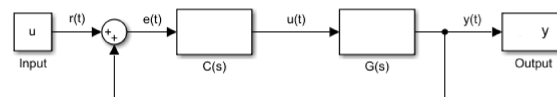


Figura 2.1: Controllo in retroazione unitaria



Dallo schema a blocchi in figura 2.1 è possibile ricavare una nuova funzione di trasferimento che lega ingresso-uscita, in particolare se consideriamo l'errore del sistema  $e(t)$ , quest'ultimo può essere scritto come:

$$e(t) = r(t) - y(t) \quad (2.1)$$

che nel dominio delle trasformate di Laplace corrisponde a

$$E(s) = R(s) - Y(s) \quad (2.2)$$

per la linearità della trasformazione.

A questo punto, la trasformata  $Y(s)$  dell'uscita può essere espressa come:

$$Y(s) = C(s)G(s)E(s) \quad (2.3)$$

e tramite la sostituzione dell'equazione 2.2 in 2.3 si ottiene la relazione ingresso-uscita a tempo continuo:

$$Y(s) = C(s)G(s)(R(s) - Y(s)) \quad (2.4)$$

Data la relazione 2.4 è possibile definire immediatamente la nuova funzione di trasferimento del sistema in catena chiusa:

$$\frac{Y(s)}{R(s)} = \frac{C(s)G(s)}{1 + C(s)G(s)} := H(s) \quad (2.5)$$

Da cui si nota chiaramente l'effetto del controllore introdotto  $C(s)$ : quest'ultimo infatti ha lo scopo di modificare il valore del guadagno di Bode del sistema in catena aperta visto nel capitolo 1 e di introdurre eventuali poli o zeri nel sistema in catena chiusa.

Queste modifiche vengono applicate allo scopo di ottenere le specifiche richieste in sezione 2.1.

## 2.3 Controllore PID

Per rendere il sistema ottimale e idoneo alle caratteristiche richieste, si decide di procedere con l'utilizzo di controllori di tipo **PID** (Proporzionale-Integrale-

Derivativo) con equazione:

$$C(s) = K_P + \frac{K_I}{s} + K_D s \quad (2.6)$$

Questo è un tipo di controllore utilizzato per regolare il comportamento di un sistema dinamico. Si basa su tre componenti principali che lavorano insieme per produrre una risposta di controllo che cerca di mantenere il sistema il più vicino possibile al suo punto di riferimento desiderato.

Ecco una breve spiegazione di ciascuna componente:

- **Proporzionale (P):** La componente proporzionale produce un'uscita proporzionale all'errore attuale  $e(t)$ . In altre parole, misura la differenza tra  $y(t)$  e il valore desiderato  $r(t)$  e produce un segnale di controllo proporzionale a questa discordanza. L'azione proporzionale è responsabile di ridurre l'errore presente.
- **Integrale (I):** La componente integrale tiene conto degli errori passati e somma tutte le differenze  $e(t)$ . Questa azione riduce gli errori persistenti o a lungo termine che potrebbero non essere completamente corretti dall'azione proporzionale da sola. L'azione integrale agisce per ridurre il costo cumulativo dell'errore.
- **Derivativo (D):** La componente derivativa misura la velocità di variazione dell'errore nel tempo. Questa componente è utile per prevedere come l'errore  $e(t)$  sta cambiando e può essere utilizzata per anticipare come il sistema si comporterà in futuro. L'azione derivativa aiuta a ridurre il transitorio e a stabilizzare rapidamente il sistema.

Nelle applicazioni pratiche, è importante distinguere i controllori fisicamente realizzabili da quelli puramente ideali. In particolare, non è possibile costruire un controllore che presentino nella loro funzione di trasferimento  $C(s)$  un numero di zeri maggiore al numero di poli, in quanto rappresentanti di funzioni razionali improprie.

### 2.3.1 Realizzazione del controllore su Matlab

Per la scrittura e simulazione su Matlab di un controllore PID, è sufficiente esprimere i guadagni  $K_P$ ,  $K_I$  e  $K_D$  come variabili e scrivere esplicitamente la funzione di trasferimento  $C(s)$  come nell'equazione 2.6.

```

Kp = 1; % Proporzionale
Kd = 1; % Derivativo
Ki = 1; % Integrale
s = tf([1 0], 1);
C = Kp + Ki/s + Kd*s;

```

Figura 2.2: Definizione PID su Matlab

dove a scopo dimostrativo sono stati inseriti dei guadagni unitari per ognuna delle componenti.

## 2.4 Implementazione controllore P

Come prima soluzione, si decide di procedere alla realizzazione del controllore di tipo proporzionale  $\mathbf{P}$ , in particolare  $C(s) = K_P$ .

Sostituendo l'espressione di Laplace del controllore nella funzione di trasferimento del controllore 2.5, dove  $G(s)$  corrisponde a  $W(s)$  in 1.5, si ottiene:

$$H(s) = \frac{K_P}{ms + b + K_P} \quad (2.7)$$

che può essere implementata su *Matlab* mediante i seguenti comandi:

```

s = tf([1 0], 1);
H = Kp/(m*s+b+Kp);

```

Figura 2.3: Definizione  $H(s)$  su *Matlab*

Come primo tentativo per osservare il comportamento del sistema in catena chiusa 2.7, si imposta un valore di  $K_P = 150$ . Mantenendo in ingresso il gradino desiderato  $u = 500N$ , la risposta avrà questo carattere:

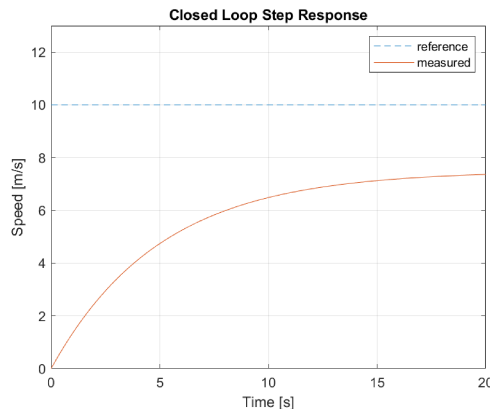


Figura 2.4: Risposta di  $H(s)$  al riferimento  $r = 10m/s$

dove si nota un'evidente diminuzione del tempo di salita, a fronte però di un errore percentuale a regime molto alto, visibilmente distante dal 2% richiesto nei requisiti progettuali.

A fronte di questo, l'unica soluzione possibile per rendere il sistema ottimale è procedere per tentativi aumentando il guadagno proporzionale  $K_P$ . Con questo metodo, il sistema in catena chiusa è ottimale con un valore di  $K_P = 3500$ , la cui risposta al gradino è la seguente:

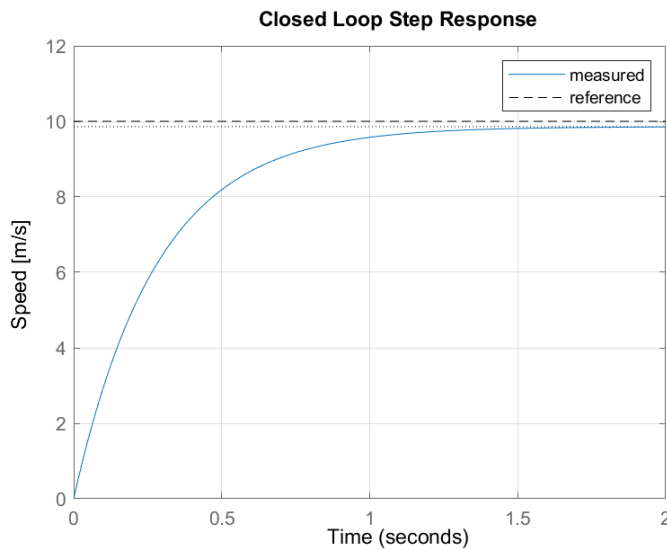


Figura 2.5: Risposta del sistema in catena chiusa  $H(s)$  con  $K_P = 3500$

In questo caso particolare, le specifiche di sistema sono rispettate, in quanto  $t_r = 0.62s$ ,  $t_{s2\%} = 1.1s$ ,  $M_p = 0\%$  e  $e_r = 1.5\%$  (calcolati mediante la routine *stepinfo* di *Matlab*).

Nonostante il sistema soddisfi i requisiti prefissati, è chiaro che non è fisicamente realizzabile, in quanto un passaggio di velocità da  $v(0) = 0m/s$  a  $v(t_s) = 9.85m/s$  (circa  $35.46km/h$ ) in un tempo di  $1.1s$  comporta costi di costruzione quasi insostenibili ed inutili allo scopo dello sviluppo di un dispositivo Cruise-Control.

## 2.5 Implementazione controllore PI

Come osservato precedentemente in figura 2.4, il problema del tempo di salita e overshoot viene risolto quasi immediatamente, senza grandi valori del guadagno  $K_P$ ; resta però il problema della diminuzione dell'errore percentuale a regime  $e_r$ , risolto in precedenza con un guadagno molto elevato, come in figura 2.5.

Per risolvere questo problema, si decide di aggiungere al controllore  $C(s)$  una

componente integrale **I**, si veda la sezione 2.3 per la motivazione di questo cambiamento.

La nuova espressione del controllore sarà:

$$C(s) = K_P + \frac{K_I}{s} \quad (2.8)$$

mentre la nuova funzione di trasferimento del sistema in catena chiusa risulta:

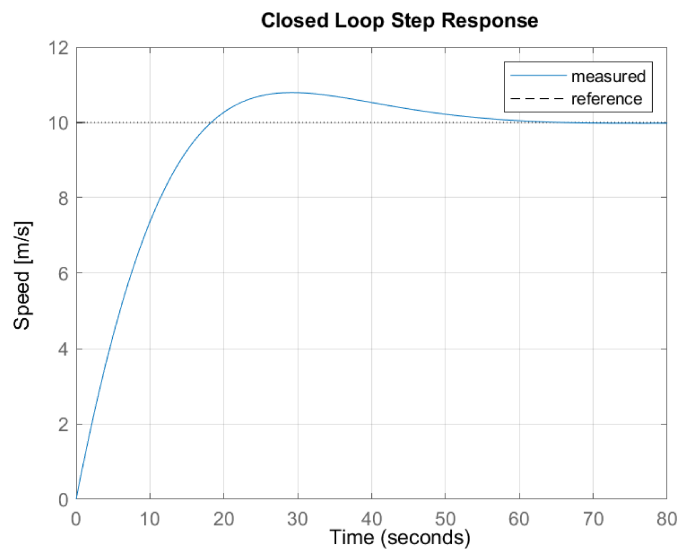
$$H(s) = \frac{K_P s + K_I}{m s^2 + (K_P + b)s + K_I} \quad (2.9)$$

Per la sua scrittura su *Matlab* si procede come per il controllore di tipo **P**, più esplicitamente:

```
Kp = 100;
Ki = 10;
s = tf([1 0], 1);
H = (Kp*s + Ki)/(m*s^2 + (b+Kp)*s + Ki);
```

**Figura 2.6:** Scrittura di  $H(s)$  su *Matlab*

dove sono stati assegnati dei valori di test a  $K_I$  e  $K_P$ , rispettivamente 10 e 100. Con questi guadagni la risposta del sistema in catena chiusa al gradino  $r = 10m/s$  ha il seguente grafico:



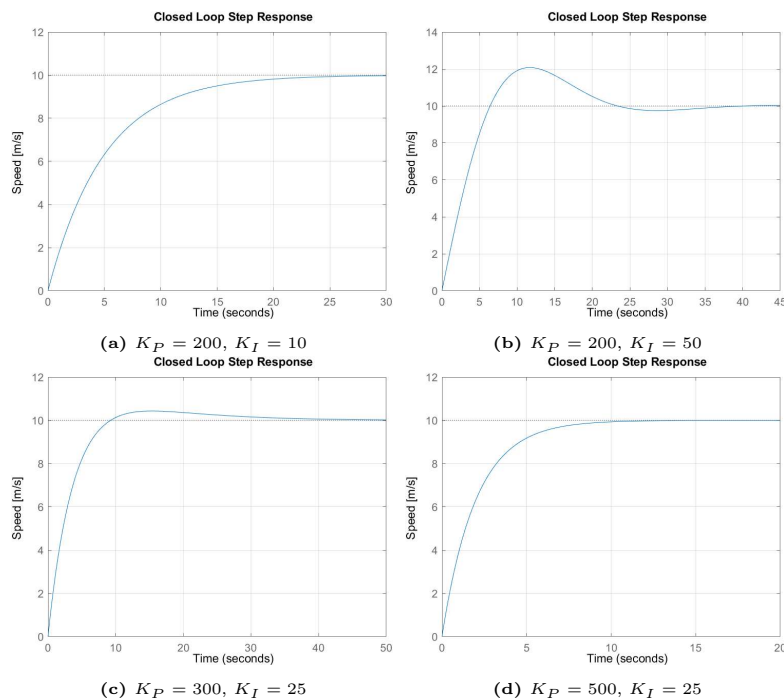
**Figura 2.7:** Risposta di  $H(s)$  al riferimento  $r = 10m/s$

con l'inserimento di una parte integrale, sarà presente una certa sovraelongazione percentuale, in questo caso specifico ottima in quanto minore del limite consentito 20%. Non è possibile dire lo stesso del tempo di assestamento al 2%, infatti con

i valori di figura 2.6 il suo valore è di  $t_{s2\%} = 50.72s$ .

Per rendere il controllore **PI** ottimale si procede a modificare manualmente i parametri  $K_I$  e  $K_P$ , tenendo conto che aumentando l'azione proporzionale il tempo di salita  $t_r$  diminuisce, ma può portare a un sistema fisicamente non realizzabile. Le stesse considerazioni valgono per l'azione integrale, che con l'aumento del guadagno può provocare un overshoot troppo elevato.

Il risultato della variazione del valore dei guadagni è riportato in seguito:



**Figura 2.8:** Risposta del sistema in catena chiusa  $H(s)$  con diversi valori di guadagno

Si osserva dai primi tre grafici 2.8a, 2.8b, 2.8c che i guadagni scelti non soddisfano le richieste di progetto. In particolare:

- Per le risposte 2.8a, 2.8c il tempo di salita e di assestamento sono troppo elevati, nonostante la sovraelongazione percentuale sia ottima.
- Per il comportamento 2.8b, è evidentemente visibile un picco che sfora il valore di  $12m/s$ , di conseguenza l'overshoot avrà un valore  $M_p > 20\%$ , non accettabile.

A differenza dei grafici descritti precedentemente, la risposta 2.8d risulta rispettare tutti i requisiti progettuali impostati, infatti  $M_p = 0\%$ ,  $t_r = 4.39s$ ,  $t_{s2\%} = 7.82s$  e l'errore percentuale  $e_r = 0.1\%$ . Grazie a un tempo di salita sensibilmente maggiore a  $1s$ , inoltre, il sistema risulta fisicamente realizzabile.

## 2.6 Implementazione controllore PID

Allo scopo di ridurre il transitorio e aumentare la velocità di stabilizzazione del sistema, è possibile introdurre un'azione derivativa **D** (si veda sezione 2.3). Il termine aggiunto appare esplicitamente nell'espressione di Laplace del controllore:

$$C(s) = K_P + \frac{K_I}{s} + K_D s \quad (2.10)$$

Per quest'ultima soluzione, la funzione di trasferimento  $H(s)$  del sistema in catena chiusa (derivante dalla forma in equazione 2.5) risulta essere:

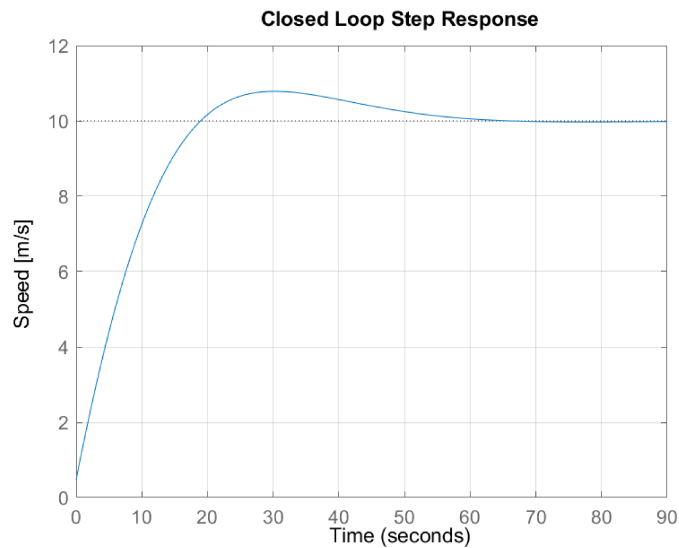
$$H(s) = \frac{K_D s^2 + K_P s + K_I}{(m + K_D) s^2 + (b + K_P) s + K_I} \quad (2.11)$$

la cui implementazione in *Matlab* corrispondente è

```
Kp = 100;
Ki = 10;
Kd = 50;
s = tf([1 0], 1);
H = (Kd*s^2 + Kp*s + Ki)/((m+Kd)*s^2 + (b+Kp)*s + Ki);
```

**Figura 2.9:** Definizione di  $H(s)$  su *Matlab*

Impostando dei guadagni iniziali di  $K_P = 100$ ,  $K_I = 10$ ,  $K_D = 50$ , si ottiene la seguente risposta al gradino di riferimento  $r = 10m/s$ :

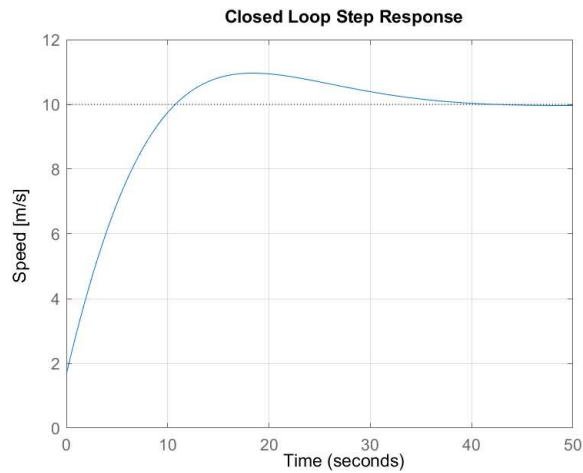


**Figura 2.10:** Risposta di  $H(s)$  al riferimento  $r = 10m/s$

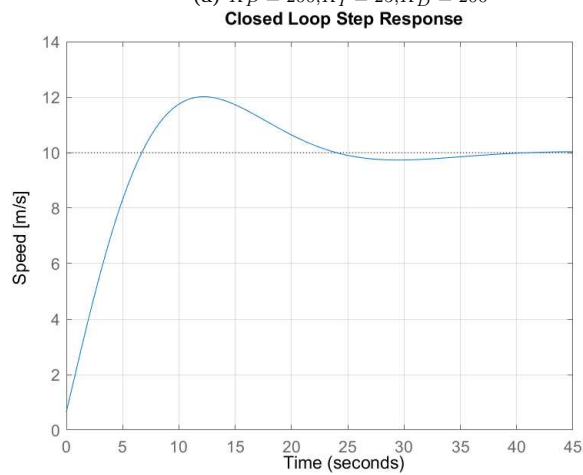
Anche in questo terzo caso, è necessario modificare i parametri per tentativi, in modo che il sistema in catena chiusa di equazione 2.11 rispetti le specifiche di

progetto.

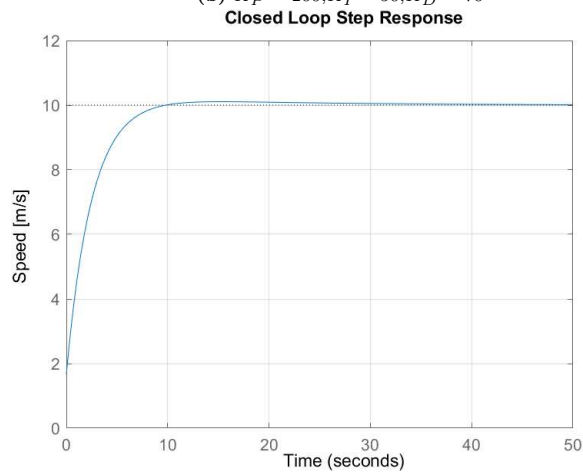
Procedendo con la modifica dei valori di guadagno delle varie componenti, si trovano le seguenti risposte al gradino:



(a)  $K_P = 200, K_I = 25, K_D = 200$



(b)  $K_P = 200, K_I = 50, K_D = 70$



(c)  $K_P = 500, K_I = 30, K_D = 200$

**Figura 2.11:** Risposta al gradino del sistema in catena chiusa  $H(s)$  con diversi valori di guadagno del controllore PID  $C(s)$



Prendendo in analisi i tre grafici di figura 2.11 si possono fare le seguenti considerazioni:

- La risposta 2.11a presenta un overshoot accettabile (minore del 20%) a discapito di un tempo di assestamento molto elevato, superiore ai 30s.
- La risposta 2.11b riporta una sovraelongazione percentuale critica, infatti il picco si avvicina al valore  $12m/s$ , il che equivale a un  $M_p \simeq 20\%$ , non accettabile secondo le specifiche di progetto. Presenta inoltre un tempo di assestamento troppo elevato, come nel caso precedente.
- L'ultima risposta 2.11c, con i guadagni  $K_P$ ,  $K_I$ ,  $K_D$  scelti, risulta essere ottimale. In particolare, l'overshoot vale  $M_p = 1.06\%$ , il tempo di salita  $t_r = 4.91s$  (minore del massimo consentito di 5s), con un tempo di assestamento al 2% di  $t_{s2\%} = 7.76s$  e un errore percentuale a regime  $e_r \simeq 0.1\%$ , grazie all'introduzione dell'azione derivativa **D** (valori esatti ricavati tramite la routine *stepinfo* di *Matlab*).

## 2.7 Considerazioni sui diversi tipi di controllori

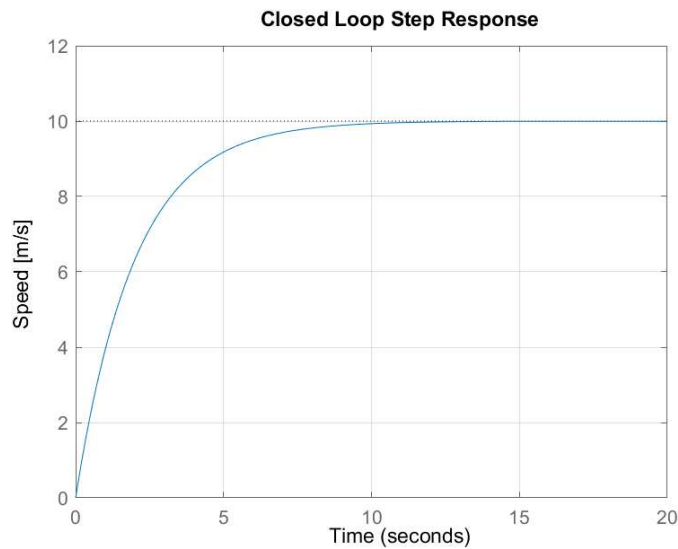
Analizzando le diverse soluzioni al problema della progettazione del controllore  $C(s)$  viste nelle sezioni 2.4, 2.5 e 2.6, si conclude che per ognuno dei tre casi si è riusciti ad ottenere un sistema in catena chiusa  $H(s)$  che rispetti i requisiti progettuali indicati in sezione 2.1.

Nonostante questo, è importante evidenziare vantaggi e svantaggi delle varie implementazioni:

- Il controllore puramente proporzionale **P** permette di soddisfare le specifiche molto velocemente, infatti con un valore elevato del guadagno  $K_P$  il tempo di salita e di assestamento diminuiscono fino ad arrivare a grandezze di pochi secondi. Il problema è la realizzabilità pratica, in quanto l'accelerazione del veicolo è eccessiva (come osservato precedentemente in sezione 2.4), comportando dei costi di produzione insostenibili.
- Il controllore **PID** è una buona soluzione allo scopo; dopo alcune modifiche ai guadagni delle varie componenti si riesce a rispettare i requisiti progettuali con ampio margine. Lo svantaggio di questa risoluzione è la presenza di 3 componenti distinte nel controllore, che comportano un costo di produzione maggiore rispetto ad altri dispositivi, per esempio il controllore **PI**.

- Il controllore **PI** è la soluzione ottima per la realizzazione del Cruise-Control, infatti, come si osserva in sezione 2.5, con i valori di guadagno  $K_P$  e  $K_I$  ricavati il sistema rispetta pienamente le specifiche progettuali e risulta realizzabile nella pratica, con dei costi di produzione minore rispetto ai diretti concorrenti.

Viene riportata di seguito la risposta in catena chiusa del sistema  $H(s)$  realizzato con controllore **PI** ottimale:



**Figura 2.12:** Risposta di  $H(s)$  al riferimento  $r = 10m/s$  con controllore **PI**

e i relativi parametri di valutazione ricavati da *Matlab* sono:

$$t_r = 4.39s$$

$$t_{s2\%} = 7.82s$$

$$M_p = 0\%$$

$$e_r \simeq 0.1\%$$

# Capitolo 3

## Simulazione del sistema su Simulink

### 3.1 Considerazioni preliminari sul software Simulink

Per un'analisi più accurata e verosimile del modello del Cruise-Control studiato nei capitoli precedenti, è possibile effettuare delle simulazioni virtuali tramite il software *Simulink* presente all'interno di *Matlab*.

In particolare, *Simulink* è un ambiente di programmazione visuale basato su blocchi, progettato principalmente per la modellazione, la simulazione e l'analisi di sistemi dinamici. Con il suo utilizzo, è possibile creare modelli di sistemi dinamici usando una rappresentazione grafica delle equazioni matematiche che descrivono il comportamento del sistema nel dominio del tempo o della frequenza. Alcune delle principali attività che si possono svolgere con questo software sono le seguenti:

- Modellazione di sistemi dinamici: è possibile costruire modelli di sistemi fisici o di controllo utilizzando una vasta gamma di blocchi funzionali predefiniti. I blocchi rappresentano elementi come segnali, sistemi, sensori, attuatori, controllori, ecc., e possono essere collegati insieme per creare il modello completo del sistema. Nel caso specifico preso in considerazione, il modello del Cruise Control.
- Simulazione: Una volta creato il modello, si possono eseguire simulazioni per comprendere come il sistema si comporta nel tempo o nella frequenza. Questo consente di studiare il comportamento dinamico del sistema, verificarne le prestazioni, identificare potenziali problemi e valutare diverse strategie di controllo.

- Controllo automatico: *Simulink* può essere utilizzato per progettare e implementare controllori PID, controllori avanzati e altre tecniche di controllo automatico. È inoltre utile per ottimizzare i parametri dei controllori, valutare le prestazioni e analizzare il comportamento del sistema in presenza di regolazione automatica.

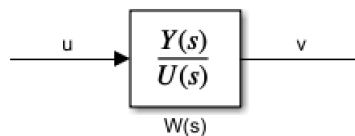
## 3.2 Implementazione del modello su Simulink

Per poter rappresentare il modello riferito al sistema fisico 1.1 su *Simulink* è necessario sviluppare una struttura a blocchi, usufruendo dei vari tipi di blocco disponibili all'interno dell'ambiente di programmazione.

Nel caso specifico preso in considerazione, dato che è stata già ricavata in precedenza la funzione di trasferimento  $W(s)$  in sezione 1.2, per comodità riportata:

$$W(s) = \frac{Y(s)}{U(s)} = \frac{1}{ms + b} \quad (3.1)$$

è possibile utilizzare il blocco **Transfer Fcn** di Simulink per rappresentare l'intero sistema attraverso la sua funzione di trasferimento:



**Figura 3.1:** Rappresentazione *Simulink* della funzione di trasferimento  $W(s)$

dove  $u$  rappresenta la forza motrice del motore in Newton  $[N]$ , mentre l'uscita corrisponde alla velocità  $v$  in  $[m/s]$ .

Per poter inserire  $W(s)$  all'interno del blocco, è stato necessario l'utilizzo della routine **tfdata** di *Matlab* che permette di ricavare due vettori dalla funzione di trasferimento, corrispondenti rispettivamente a numeratore e denominatore di essa (i quali saranno in seguito inseriti manualmente all'interno dei *Block Parameters* del blocco *Transfer Fcn*), più esplicitamente:

```
W = tf(1, [m b]) %funzione di trasferimento
[numW, denW] = tfdata(W, 'v');
```

**Figura 3.2:** ricavare numeratore e denominatore dalla funzione di trasferimento  $W(s)$  su *Matlab*

A questo scopo era possibile anche sviluppare l'implementazione a blocchi seguendo direttamente l'equazione di Newton 1.1, dove ogni termine verrà rappresentato

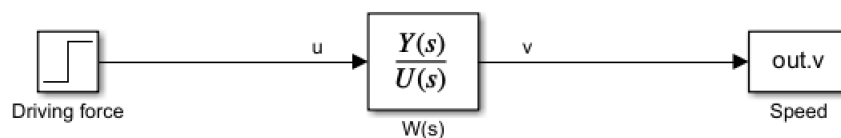
tramite un blocco *gain* con l'uso di nodi sommatore/sottrattori. Per lo scopo seguito, non è conveniente questo tipo di soluzione, in quanto causa un'occupazione di spazio maggiore rispetto al semplice blocco *Transfer fcn* di figura 3.1.

### 3.3 Simulazione del sistema in catena aperta su Simulink

Per la simulazione via software del sistema in catena aperta rappresentato dalla funzione di trasferimento  $W(s)$  di equazione 1.5 o 3.1, è necessario completare la configurazione dello schema *Simulink* di figura 3.1 con due parti aggiuntive:

- Una componente che permetta di simulare l'ingresso  $u$  a gradino del sistema in catena aperta, dove il valore è specificato in sezione 1.1, per comodità riportato:  $u = 500N$ . A questo scopo, si è scelto di utilizzare il blocco **step**, il quale permette di simulare un segnale a gradino di valore variabile, applicato in un certo istante del periodo di simulazione (per comodità si è scelto  $1s$ ).
- Un collegamento in uscita che renda possibile l'elaborazione del segnale  $v$  direttamente sul software *Matlab*, nel quale grazie al comando **plot** si è in grado di elaborarlo e visualizzarne l'andamento temporale. Il blocco **To Workspace** risulta ottimo, in quanto permette di salvare l'uscita  $v$  di  $W(s)$  in una variabile *Structure with Time*, da cui è facilmente ricavabile vettore di tempi e corrispondente vettore delle velocità istantanee.

Completando la struttura di figura 3.1 con i blocchi appena descritti, il sistema in catena aperta su *Simulink* risulterà:



**Figura 3.3:** Schema *Simulink* del sistema in catena aperta

Per quanto riguarda il processo di test, è importante premettere che il software *Simulink*, una volta configurato il modello e i parametri di simulazione, calcola il comportamento del sistema nel tempo, basandosi sulle equazioni matematiche e sulle relazioni definite dai blocchi e dai segnali presenti. Nella pratica, è in grado di sviluppare il comportamento del sistema su un lasso di tempo eventualmente

molto ampio in pochi secondi.

Per la simulazione del sistema in catena aperta di figura 3.3, si è scelto di impostare un tempo di simulazione di  $T = 150s$ , in quanto, come già osservato in sezione 1.4, con tempi minori non si è in grado di osservare la completa risposta al gradino del sistema.

È riportata di seguito il grafico della velocità  $v$  simulata, in  $m/s$ :

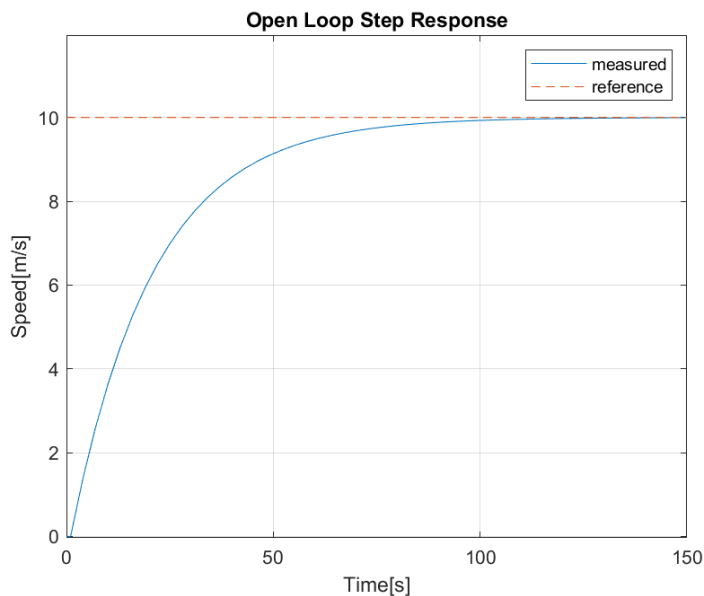


Figura 3.4: Simulazione su *Simulink* del sistema in catena aperta

dove si nota immediatamente la somiglianza alla risposta ottenuta tramite *Matlab* di figura 1.6. Infatti, il sistema in catena aperta descritto tramite blocchi di figura 3.3 non soddisfa i requisiti progettuali impostati, in particolare il tempo di salita e di assestamento superano di molto i valori limite di 5 e 10 secondi.

### 3.4 Simulazione del sistema in catena chiusa su Simulink

Avendo osservato in precedenza, in particolare in sezione 2.4 l'importanza di realizzare sistemi di controllo fisicamente realizzabili, per rendere il più verosimile possibile la simulazione, è necessario l'inserimento di un **satratore**.

Questa componente, rappresentata su *Simulink* dal blocco *Saturation*, serve a limitare la forza erogabile in uscita dal controllore a un valore massimo consentito. Per ricavare questo valore, sapendo che per passare da una velocità iniziale  $v(0) = 0m/s$  a una velocità desiderata  $r = 10m/s$  è necessario restare sotto il tempo di

salita  $t_r < 5s$ , supponiamo che l'accelerazione massima consentita sia di  $a_{max} = 2.5m/s^2$  (supponendo che il  $t_r$  reale sia minore del valore limite). A questo punto per trovare la forza massima erogata si esegue un semplice passaggio:

$$u_{max} = ma_{max} = 2500N \quad (3.2)$$

In questo modo, il veicolo non potrà erogare una forza motrice maggiore di quella appena ricavata, impedendo così accelerazioni improvvise molto brusche.

### 3.4.1 Controllore P e PID

Dopo le considerazioni conclusive sul tipo di controllore da utilizzare fatte in sezione 2.7, si è scelto di procedere con la realizzazione del sistema in catena chiusa  $H(s)$  tramite sistema di controllo **PI**.

Nonostante questo, a scopo dimostrativo è riportata nella sottosezione corrente l'implementazione *Simulink* del sistema in catena chiusa con controllo tramite **P** e **PID**.

Per quanto riguarda il sistema retroazionato con controllo proporzionale **P**, sarà necessario inserire un blocco *Gain* con valore  $K_P = 3500$  (da sezione 2.4) in serie alla funzione di trasferimento  $W(s)$  rappresentata dal *Transfer Fcn* di figura 3.3. Per realizzare la struttura in catena chiusa, è sufficiente ricorrere all'utilizzo di un operatore *Sum* con i collegamenti adeguati.

Lo schema *Simulink* è il seguente:

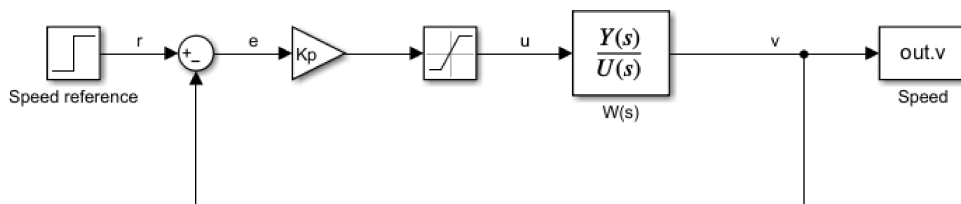


Figura 3.5: Schema *Simulink* del sistema in catena chiusa con controllore **P**

E il relativo andamento della velocità dell'automobile simulato corrisponde alla figura 3.6, con riferimento di velocità  $r = 10m/s$  applicato all'istante  $t = 1s$  su una simulazione della durata complessiva di  $5s$ .

Come già osservato in precedenza in sezione 2.4, il sistema rispetta a pieno le specifiche, nonostante la presenza di un piccolo errore percentuale a regime (più accentuato rispetto alla simulazione numerica su *Matlab*), ma il tempo di salita ridotto (nonostante la pendenza di salita sia stata ridotta grazie alla saturazione)

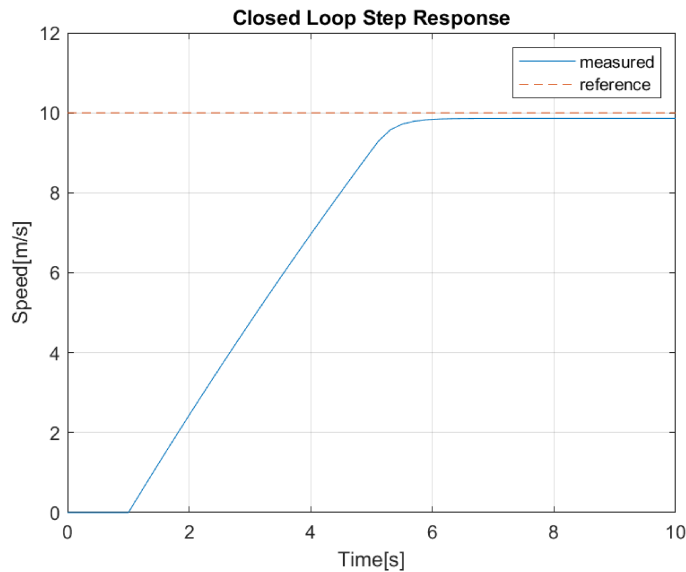


Figura 3.6: Risposta del sistema in catena chiusa in *Simulink* a  $r = 10\text{m/s}$  con controllore **P**

rende il dispositivo non realizzabile nella pratica, a causa dei costi molto alti per la produzione.

Per l'implementazione su *Simulink* del **PID** è necessario inserire in serie alla funzione di trasferimento  $W(s)$  un parallelo seguito da un blocco *Sum*, che comprenda tutti e tre i guadagni (rispettivamente  $K_P = 500$ ,  $K_I = 30$ ,  $K_D = 200$ , da sezione 2.6) e dei blocchi *Integrator* e *Derivative*.

Questi ultimi due blocchi non presenti nel controllore **P** servono alla realizzazione della parte integrativa **I** e derivativa **D**.

Nella pratica, lo schema a blocchi del sistema in catena chiusa è indicato dalla figura seguente:

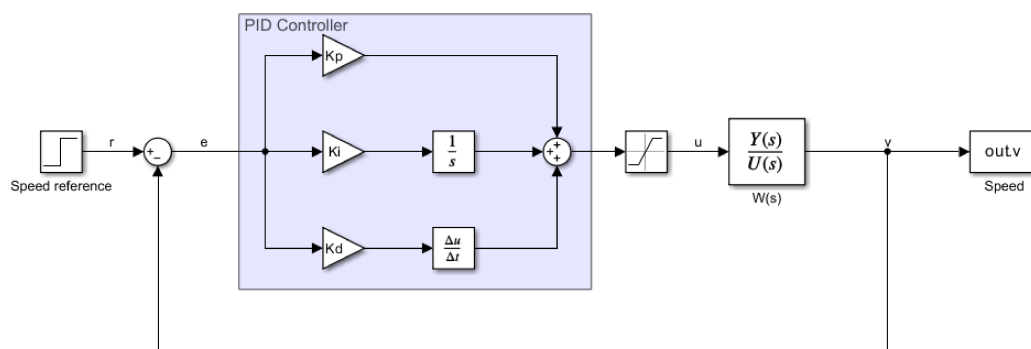


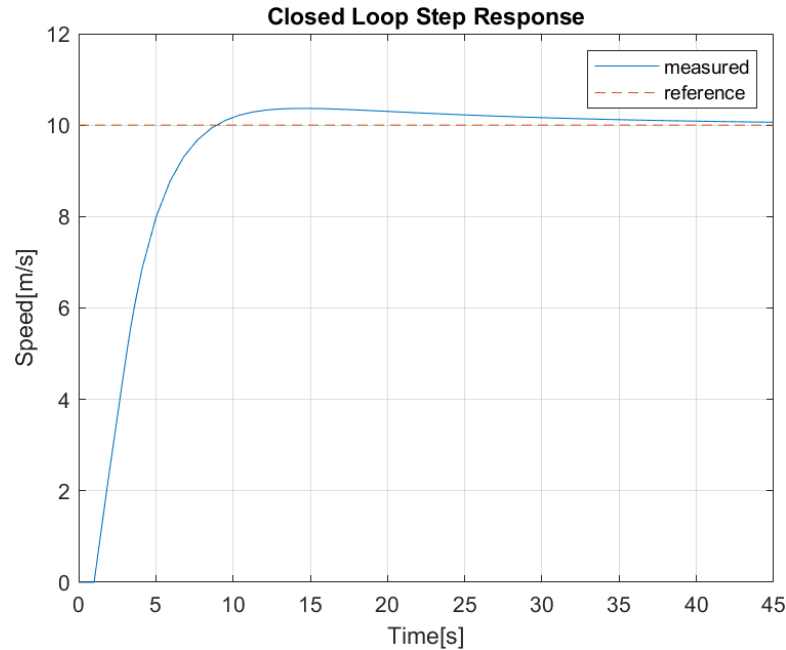
Figura 3.7: Schema *Simulink* del sistema in catena chiusa con controllore **PID**

dove è evidenziata in un'area di colore blu la sintesi a blocchi del controllore sopracitato, comprensivo di ognuna delle principali componenti.

Per la simulazione della velocità  $v$  dell'automobile si è scelto un tempo di simula-



zione di  $T = 45s$ , in cui il riferimento di velocità a gradino è applicato all'istante temporale  $t = 1s$ .



**Figura 3.8:** Risposta del sistema in catena chiusa in *Simulink* a  $r = 10m/s$  con controllore **PID**

Come si può osservare in figura 3.8, con l'inserimento della parte integrale la risposta del sistema in catena chiusa al gradino presenta una sovralongazione percentuale, assente nel sistema precedente di figura 3.5.

Inoltre, confrontando con la risposta numerica generata da *Matlab* di figura 2.11c, si nota come il tempo di assestamento  $t_{s2\%}$  sia sensibilmente maggiore nella risposta *Simulink*, mentre tempo di salita, overshoot ed errore a regime restino abbastanza invariati. Questo disallineamento è dovuto principalmente dall'uso dei blocchi *Integrator* e *Derivative* nello schema 3.7, nonché dalla saturazione del valore di  $2500N$ .

### 3.4.2 Controllore PI

Come da sezione 2.7, per la realizzazione pratica del dispositivo Cruise-Control si è deciso di utilizzare un controllore di tipo **PI** (Proporzionale-Integrale) per il sistema in catena chiusa  $H(s)$  di equazione 2.9. È quindi opportuno riportare il relativo schema a blocchi all'interno dell'ambiente di lavoro *Simulink*: la parte relativa al controllore sarà analoga a quella vista in figura 3.7, a meno del termine derivativo comprensivo di guadagno  $K_D$  e blocco *Derivative*.

Il sistema risulta dunque:

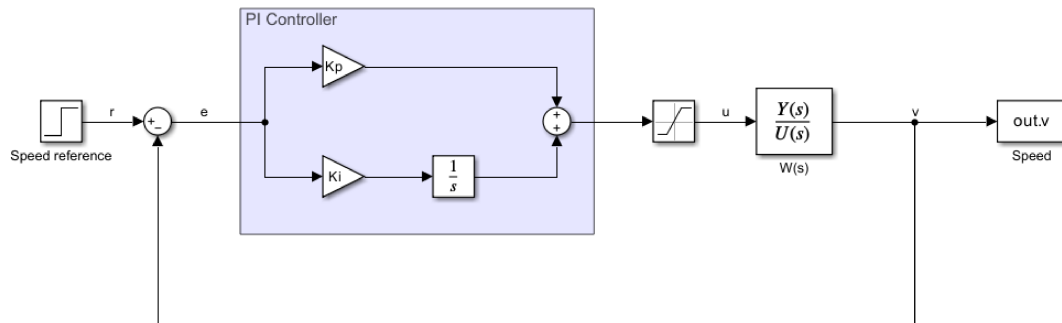


Figura 3.9: Schema *Simulink* del sistema in catena chiusa con controllore **PI**

dove i guadagni sono quelli ricavati in sezione 2.5, di valore specifico  $K_P = 500$  e  $K_I = 25$ .

Procedendo come nei casi presi in analisi precedentemente, con l'uso combinato di *Matlab* e *Simulink* si ottiene la seguente risposta al gradino di riferimento  $r = 10\text{m/s}$ :

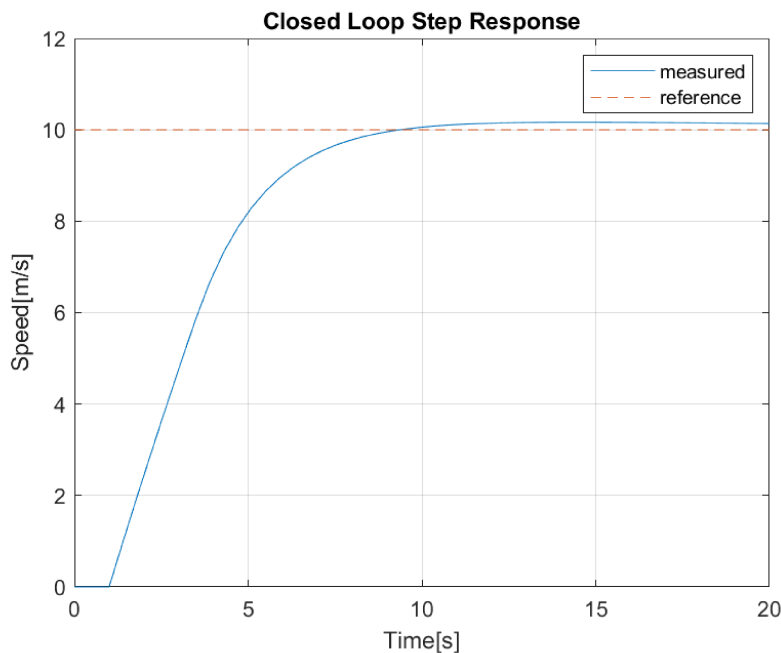


Figura 3.10: Risposta del sistema in catena chiusa in *Simulink* a  $r = 10\text{m/s}$  con controllore **PI**

Per verificare che il sistema in catena chiusa con controllore **PI** è ottima, si riprendono i valori ricavati, da confrontare con i requisiti progettuali di sezione 2.1.

$$\begin{aligned}
 t_r &= 4.59\text{s} (< 5\text{s}) \\
 t_{s2\%} &= 8.07 (< 15\text{s}) \\
 M_p &= 1.66\% (< 20\%) \\
 e_r &\simeq 0\% (< 2\%)
 \end{aligned}$$

In seguito sono riportati i parametri di valutazione della risposta in figura cal-

colati mediante la routine *stepinfo* di *Matlab*, dopo aver esportato i risultati di simulazione da *Simulink*.

```

RiseTime: 4.5890
TransientTime: 8.0709
SettlingTime: 8.0709
SettlingMin: 9.1819
SettlingMax: 10.1662
Overshoot: 1.6617
Undershoot: 0
Peak: 10.1662
PeakTime: 14.3000

```

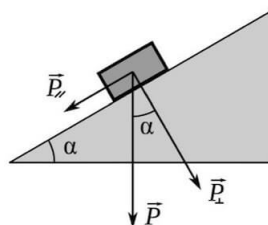
**Figura 3.11:** parametri della risposta al gradino di figura 3.10 estratti da *Matlab*

Se preso in considerazione lo stesso sistema in catena chiusa  $H(s)$  con risposta numerica (calcolata su *Matlab*) di figura 2.8d, si conclude che l'andamento è quasi analogo, con parametri di valutazione della risposta molto simili. Le piccole variazioni, come negli schemi *Simulink* precedenti sono dovuti alla procedura di simulazione del software e all'inserimento del blocco di saturazione.

### 3.5 Risposta del Cruise Control a salite o discese

Per rendere la simulazione ancora più interessante, verificando che il sistema in catena chiusa controllato da **PI** funzioni correttamente, è opportuno simulare il moto del veicolo lungo tratti di strada in salita o in discesa.

Per simulare l'andamento del fondo stradale su *Simulink*, si è scelto di procedere rappresentando la pendenza attraverso la componente di forza peso parallela e contrastante al movimento dell'automobile, espressa in  $[N]$ . È riportato di seguito lo schema utile per rappresentare questo tipo di forza:



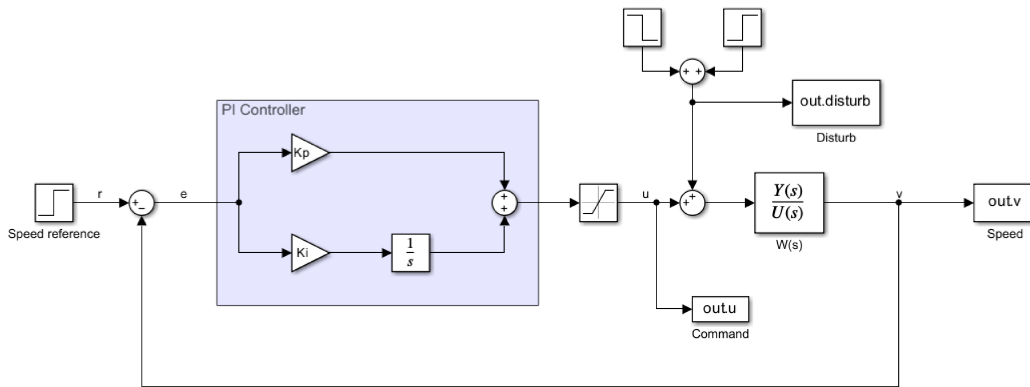
**Figura 3.12:** Rappresentazione delle forze agenti sulla massa del veicolo in una situazione di pendenza, con angolo  $\alpha$

Per trovare la componente parallela  $F_P$  che contrasta la forza motrice, è necessario eseguire il seguente passaggio aritmetico, dipendente dall'ampiezza dell'angolo  $\alpha$ :

$$F_P = mg\cos(\alpha) \quad (3.3)$$

Come intervallo di valori per l'angolo  $\alpha$ , si è scelto un minimo di  $0^\circ$  e un massimo di  $10^\circ$ , considerando che la pendenza massima del *Passo Pordoi* (salita molto famosa di un valico alpino delle Dolomiti, a un'altitudine di  $2239m$ ) è del  $10\%$ , corrispondente a  $5.71^\circ$ , valore molto minore rispetto al limite imposto.

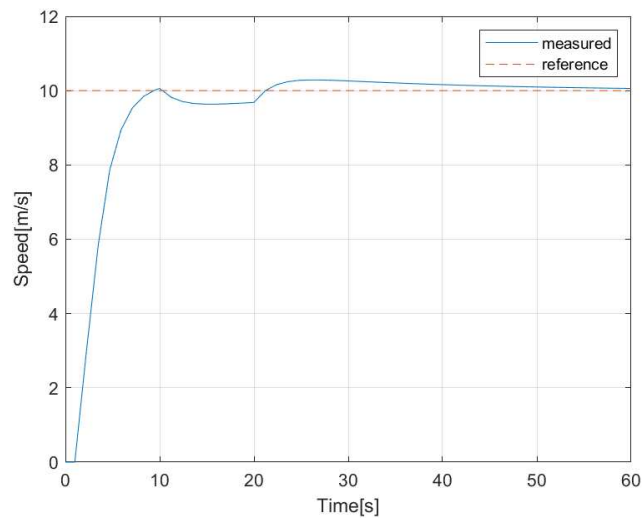
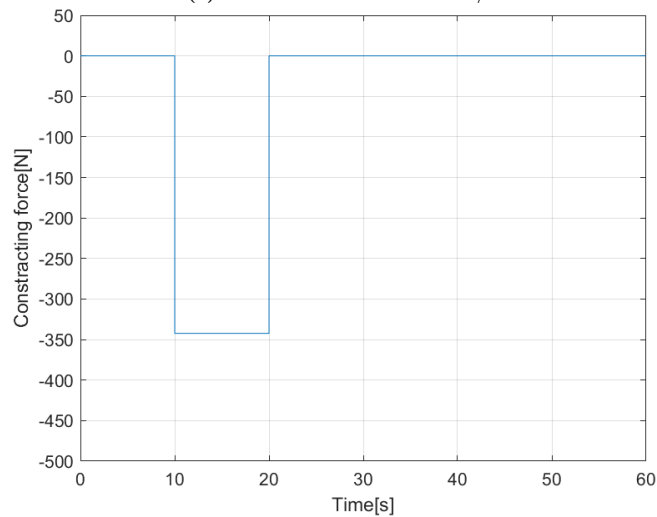
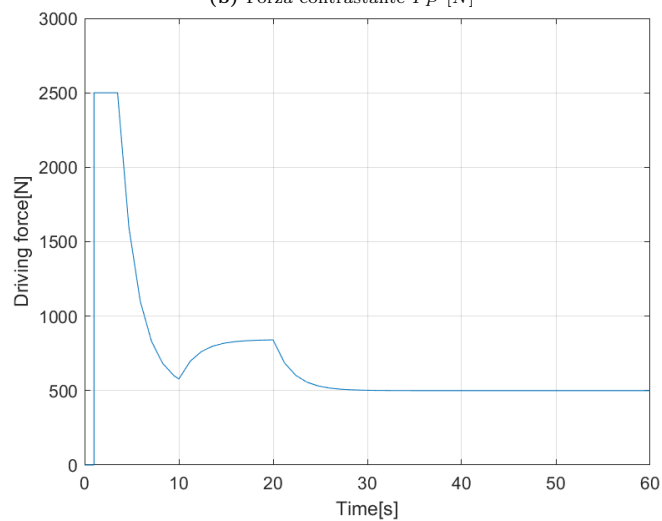
Per inserire la forza contrastante di equazione 3.3 all'interno dello schema a blocchi di figura 3.9, si utilizza un sommatore *Sum* dove in ingresso andranno due blocchi *Step* opportunamente collegati per simulare una salita/discesa di pendenza desiderata, che iniziano in un istante di tempo prefissato durante la simulazione del sistema. Il nuovo schema a blocchi risulta:

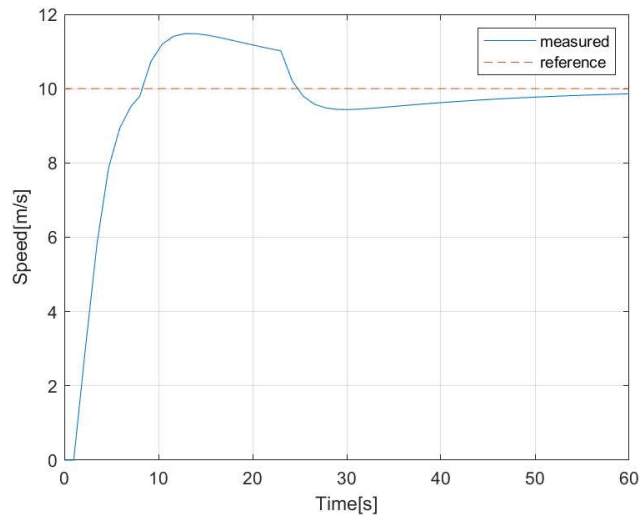
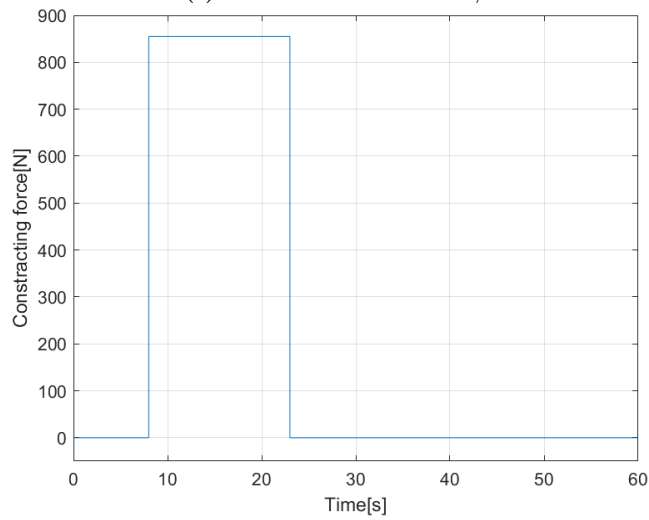
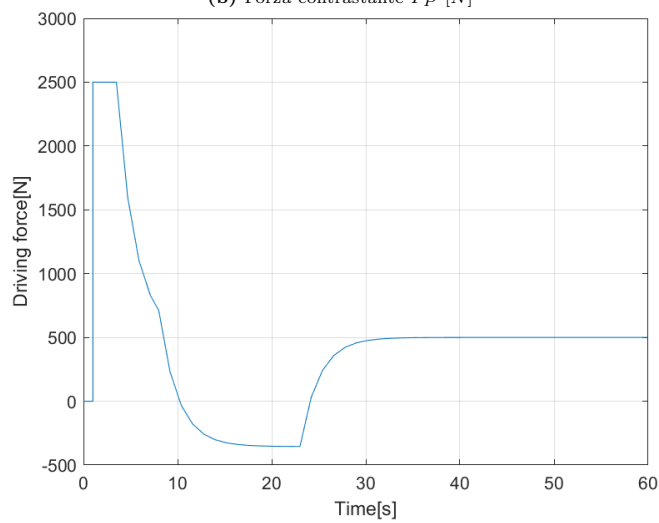


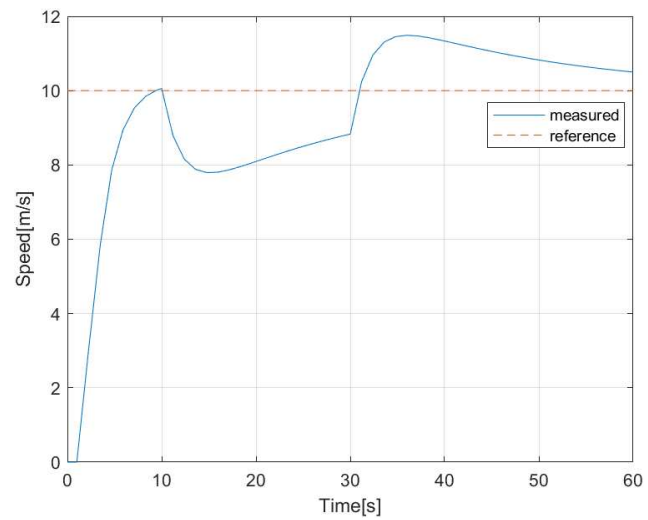
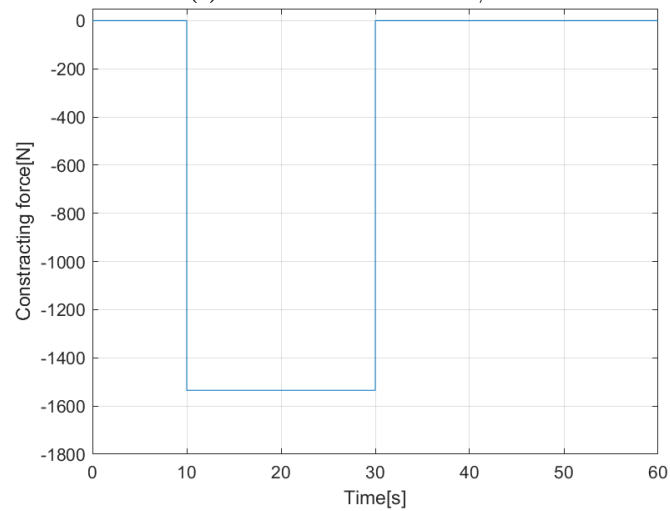
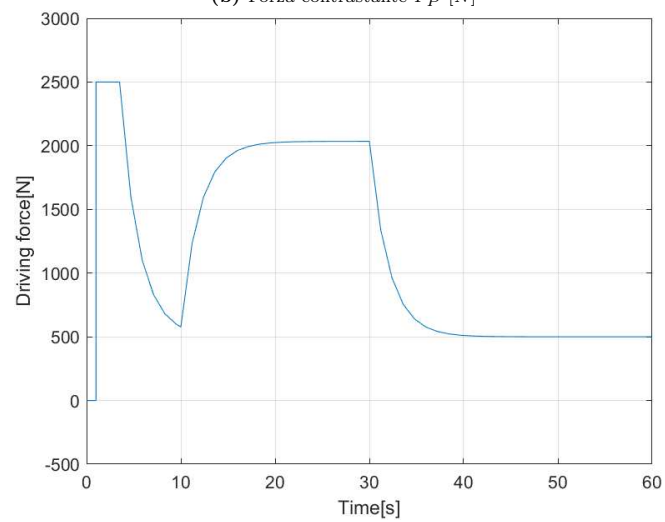
**Figura 3.13:** Schema *Simulink* del sistema in catena chiusa in presenza di salite o discese durante il percorso.

Dove resta presente il blocco di saturazione che limita la forza motrice a un valore massimo di  $2500N$  (ricavato in precedenza in sezione 3.4), mentre sono stati inseriti dei nuovi blocchi *To Workspace* per visualizzare l'andamento dei segnali interessanti su *Matlab*. Nella parte in alto a destra di figura 3.13 è possibile vedere il modo in cui viene simulata la pendenza della strada percorsa dall'automobile. Sono riportate di seguito alcune simulazioni con diversi valori di angolo  $\alpha$ , escludendo il caso banale di  $\alpha = 0^\circ$ , corrispondente al grafico di figura 3.10 (forza contrastante  $F_P = 0N$ ).

- Salita di pendenza  $\alpha = 2^\circ$  all'istante  $t = 10s$ , della durata  $T = 10s$ , su una simulazione totale di  $60s$ . [figura 3.14]
- Discesa di pendenza  $\alpha = 5^\circ$  all'istante  $t = 8s$ , della durata  $T = 15s$ , su una simulazione totale di  $60s$ . [figura 3.15]
- Salita di pendenza  $\alpha = 9^\circ$  all'istante  $t = 10s$ , della durata  $T = 20s$ , su una simulazione totale di  $60s$ . [figura 3.16]

(a) Velocità dell'automobile in  $m/s$ (b) Forza contrastante  $F_P$  [N](c) Forza motrice in uscita dal controllore  $u$  [N]**Figura 3.14:** Simulazione salita di pendenza  $\alpha = 2^\circ$

(a) Velocità dell'automobile in  $m/s$ (b) Forza contrastante  $F_P$  [N](c) Forza motrice in uscita dal controllore  $u$  [N]**Figura 3.15:** Simulazione discesa di pendenza  $\alpha = 5^\circ$

(a) Velocità dell'automobile in  $m/s$ (b) Forza contrastante  $F_P$  [N](c) Forza motrice in uscita dal controllore  $u$  [N]**Figura 3.16:** Simulazione salita di pendenza  $\alpha = 9^\circ$

Osservando le figure, si nota come il sistema Cruise Control realizzato sia in grado di sopperire ad eventuali variazioni del manto stradale, più o meno velocemente in base all'angolo pendenza di quest'ultimo. Infatti, come si nota in figura 3.16, l'automobile ci mette molto tempo a ritornare alla velocità di riferimento rispetto ai casi 3.15 e 3.14, questo a causa dell'angolo quasi limite  $\alpha = 9^\circ$  (poco verosimile rispetto agli altri valori).

Nell'applicazione pratica, tuttavia, è importante evidenziare come sia altamente sconsigliato l'uso del dispositivo di mantenimento di velocità in caso di salite o discese critiche, in quanto mediante acceleratore e freno manuali si ha un controllo maggiore dell'automobile.



# Capitolo 4

## Cruise Control Adattivo

Per completezza, è dedicato un breve capitolo alla spiegazione e al funzionamento del sistema di guida Adaptive Cruise Control citato in introduzione di capitolo 1, molto più complesso e costoso del Cruise Control classico preso in esame nei capitoli precedenti.

### 4.1 Introduzione al Cruise Control Adattivo

Il Cruise Control Adattivo, noto anche come ACC (*Adaptive Cruise Control*), è una tecnologia avanzata utilizzata nei veicoli per migliorare la comodità e la sicurezza durante la guida su autostrade e strade a velocità costante. A differenza del dispositivo tradizionale, che mantiene costante la velocità impostata dal conducente, l'ACC regola automaticamente la velocità del veicolo per mantenere una distanza di sicurezza preimpostata rispetto al veicolo che lo precede.

Le funzioni principali di questo sistema elettronico sono:

- Rilevamento del veicolo anteriore: il sistema utilizza sensori come radar, lidar o telecamere per monitorare il traffico e rilevare i veicoli che si trovano davanti.
- Monitoraggio della distanza: il sistema calcola in modo continuo la distanza tra il veicolo in cui è installato il sistema di guida e il veicolo che lo precede.
- Regolazione automatica della velocità: in base alla distanza rilevata e alla velocità del veicolo anteriore, l'ACC può accelerare o decelerare automaticamente in modo da mantenere la distanza di sicurezza preimpostata.

- Riporto alla velocità preimpostata: una volta che il veicolo che precede si sposta o accelera, il Cruise Control Adattivo può riportare il veicolo alla velocità preimpostata dal conducente, se non sono presenti ostacoli nelle vicinanze.

Questo tipo di dispositivo elettronico può contribuire a ridurre l'affaticamento del conducente, migliorare l'efficienza del carburante e, soprattutto, aumentare la sicurezza stradale mantenendo automaticamente una distanza di sicurezza appropriata dai veicoli circostanti.

## 4.2 Modellizzazione dei veicoli e dell'ACC

A differenza di quanto visto in precedenza in sezione 1.2, è necessario in questo caso definire il modello dell'automobile (vale per entrambi i veicoli) secondo la dinamica che lega accelerazione e velocità. La funzione di trasferimento relativa corrisponde a:

$$G(s) = \frac{1}{s(0.5s + 1)} \quad (4.1)$$

che approssima le dinamiche di inerzia del veicolo e le componenti di accelerazione. Per quanto riguarda il sistema ACC, esso viene modellizzato usando il blocco *Adaptive Cruise Control System Block* di *Simulink*, il quale al suo interno presenta il collegamento opportuno di funzioni complesse con un blocco *MPC Controller*, facente parte del *Model Predictive Control Toolbox*.

Le componenti interessanti sono le interfacce di ingresso e uscita del blocco, gli **input** infatti sono:

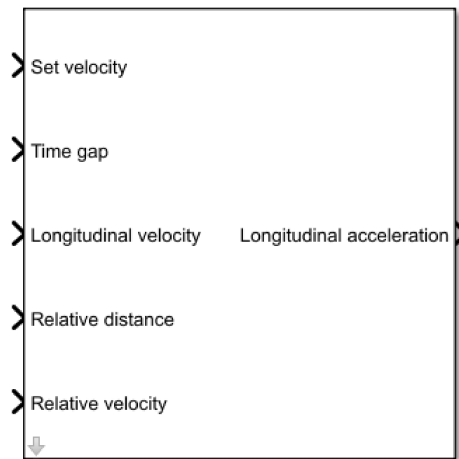
- $v_{set}$  velocità desiderata
- $T_{gap}$  distanza temporale tra i veicoli
- $v_{ego}$  velocità del veicolo dotato di ACC (*ego car*)
- $d_{rel}$  distanza relativa dall'auto che precede (*lead car*)
- $v_{rel}$  velocità relativa della *lead car*

L'unico **output** del blocco *Adaptive Cruise Control System Block* è l'accelerazione fornita dall'ACC al veicolo principale, indicato come *ego car*.

Per il calcolo della distanza di sicurezza tra i veicoli, l'equazione è la seguente:

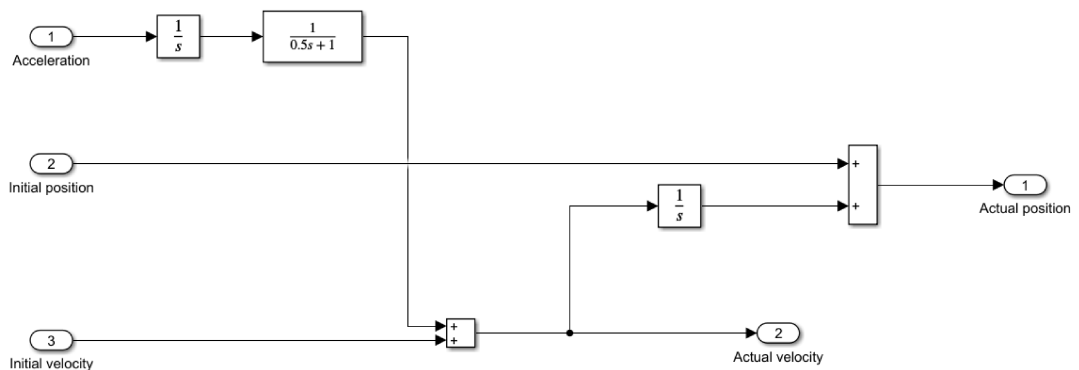
$$d_{safe} = d_{default} + T_{gap}v_{ego} \quad (4.2)$$

dove il termine  $d_{default}$  corrisponde alla distanza di sicurezza impostata. Su *Simulink* il blocco appena descritto apparirà come di seguito:



**Figura 4.1:** blocco *Adaptive Cruise Control System Block* del *Model Predictive Control Toolbox* di *Simulink*.

Per quanto riguarda lo schema a blocchi *Simulink* del modello delle vetture, è necessario l'utilizzo di blocchi integratori per ricavare non solo accelerazioni, ma valori di velocità e posizione. Ne risulterà lo schema seguente, valido sia per *ego car* che per *lead car*:



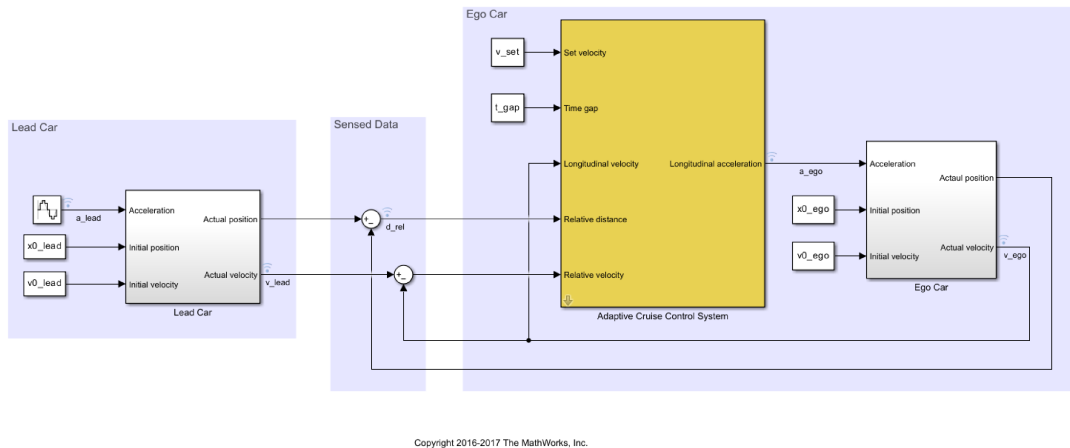
**Figura 4.2:** Schema *Simulink* che rappresenta l'automobile, in ingresso accelerazione, posizione e velocità iniziali, in uscita posizione e velocità attuali.

## 4.3 Simulazione del funzionamento dell'ACC su Simulink

Per la simulazione di funzionamento corretto del sistema Adaptive Cruise Control, si fa riferimento all'esempio presente all'interno del *Model Predictive Control*

*Toolbox* di *Simulink*, per realizzare i collegamenti adeguati tra il modello di figura 4.2 e il blocco di figura 4.1.

Il sistema, completo di *lead car*, *ego car* e ACC risulta:



**Figura 4.3:** Schema *Simulink* di collegamento del modello 4.2 con il blocco ACC.

dove si possono notare chiaramente i parametri che devono essere inseriti manualmente dall'esterno, in particolare i valori di input verranno salvati tramite variabili definite su *Matlab*.

Inizialmente vengono definiti i valori che non andranno modificati durante le varie simulazioni, in particolare:

- accelerazione massima e minima del veicolo dotato di ACC, rispettivamente  $2.5m/s^2$  e  $-2.5m/s^2$  (come visto nell'introduzione di sezione 3.4):  
`amax_ego = 2.5;`  
`amin_ego = -2.5;`
- distanza temporale  $T_{gap}$  e distanza di default in metri  $d_{default}$  tra i veicoli  
`D_default = 10;`  
`t_gap = 1.4;`

### 4.3.1 Il sistema ACC mantiene la velocità impostata, con distanza minima tra le due auto soddisfatta

La prima simulazione prende in analisi il caso in cui il veicolo munito di Adaptive Cruise Control (*ego car*) riesce a raggiungere e mantenere la velocità desiderata  $v_{set}$  in quanto la distanza minima di sicurezza tra le due automobili è soddisfatta. I parametri Matlab di simulazione sono riportati di seguito:

`x0_lead = 40;`

$v0\_lead = 25$ ; ( $90km/h$ )

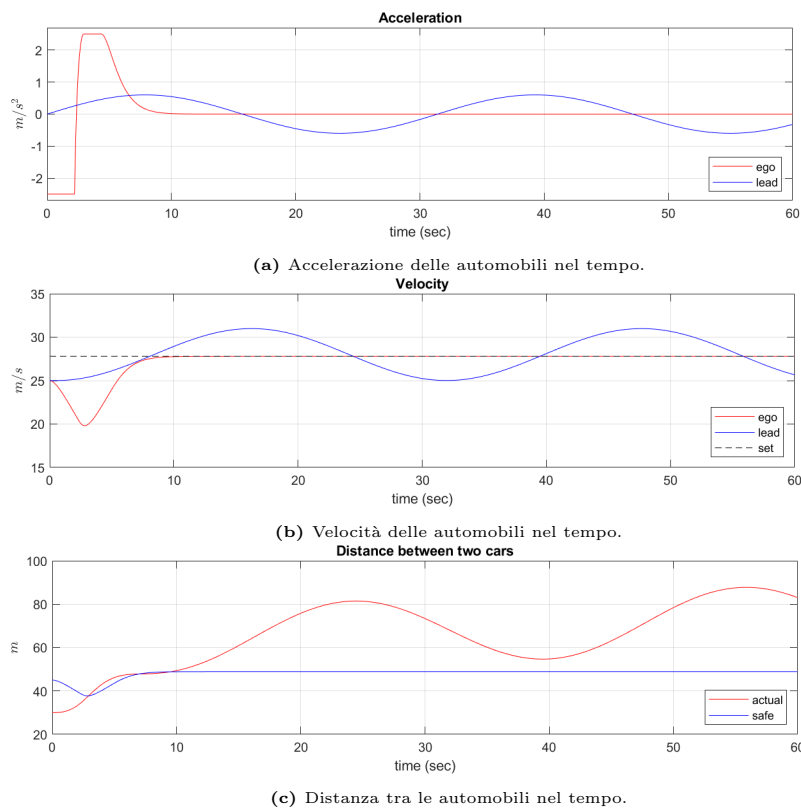
$x0\_ego = 10$ ;

$v0\_ego = 25$ ; ( $90km/h$ )

$v\_set = 27.8$ ; ( $100km/h$ )

dove velocità e posizioni iniziali sono espresse rispettivamente in  $[m/s]$  e  $[m]$ . Inoltre, per rendere la simulazione ancora più interessante, si decide di supporre che l'accelerazione dell'auto che precede (*lead car*) sia di tipo sinusoidale, così da poter permettere alla velocità di oscillare attorno al valore iniziale.

In figura sono riportati i risultati ottenuti, dopo aver eseguito una simulazione su un periodo temporale di 60s:



**Figura 4.4:** *Ego car* mantiene la velocità impostata con distanza di sicurezza tra le auto soddisfatta.

Analizzando la figura 4.4, si evidenzia l'istante iniziale in cui la distanza minima non è soddisfatta, per cui l'ACC provvede immediatamente a ridurre la velocità imponendo un'accelerazione negativa, fino a quando non è possibile attivare la componente di controllo di velocità per mantenere  $v_{set}$ . Infatti, dopo circa 10s, l'automobile che precede comincia ad allontanarsi (figura 4.4c) per cui il veicolo principale può procedere a una velocità costante impostata di  $100km/h$  (figura 4.4b), dove non dovrà più preoccuparsi della distanza minima di sicurezza.

### 4.3.2 Il sistema ACC mantiene la distanza di sicurezza, senza raggiungere la velocità impostata

La seconda simulazione si riferisce al caso in cui all'automobile dotata di Adaptive Cruise Control (*ego car*) sia fornita una  $v_{set}$  tale da avvicinarsi troppo all'auto che precede (*lead car*), in modo tale da attivare la componente di mantenimento della distanza di sicurezza a discapito del raggiungimento della velocità impostata.

I parametri *Matlab* per la simulazione di questo tipo di situazione sono i seguenti:

$x0\_lead = 40;$

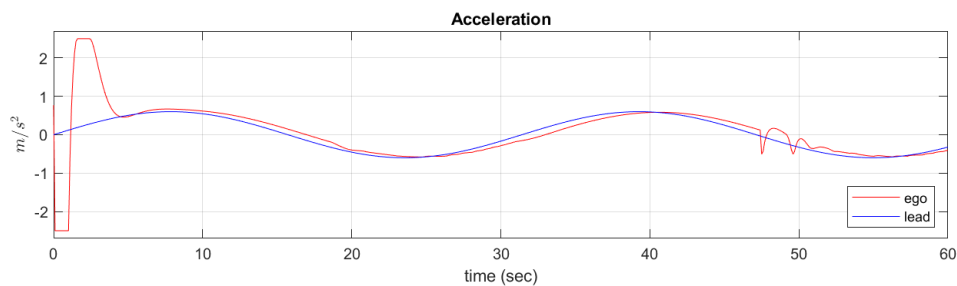
$v0\_lead = 22.2;$  ( $80km/h$ )

$x0\_ego = 10;$

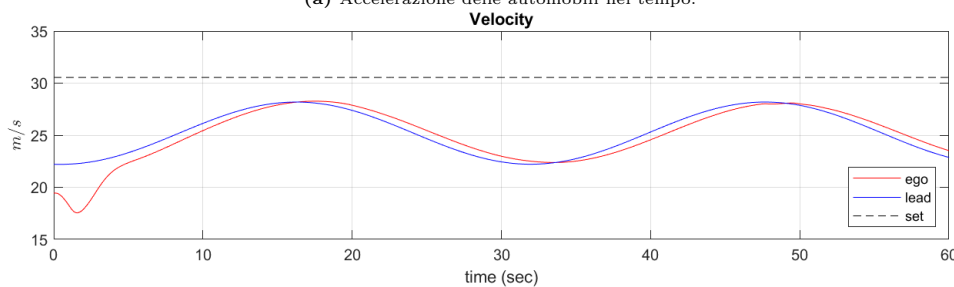
$v0\_ego = 19.44;$  ( $70km/h$ )

$v\_set = 30.56;$  ( $110km/h$ )

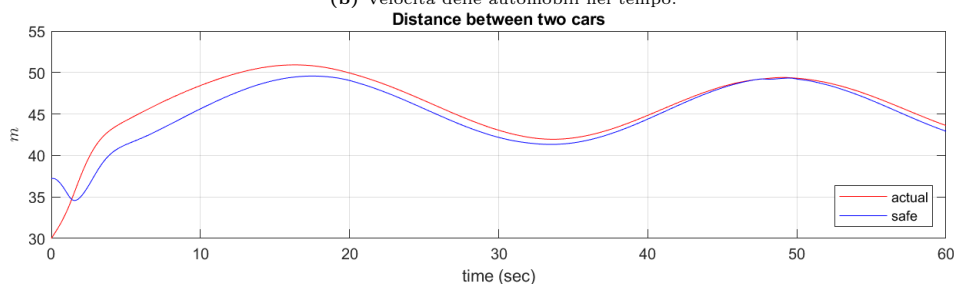
Come nella sottosezione precedente, si è scelto uno spazio temporale di 60s, dove i grafici ottenuti tramite *Matlab* risultano:



(a) Accelerazione delle automobili nel tempo.



(b) Velocità delle automobili nel tempo.



(c) Distanza tra le automobili nel tempo.

**Figura 4.5:** *Ego car* e *lead car* mantengono la distanza minima di sicurezza, non è possibile raggiungere la velocità di viaggio impostata.

Analizzando i tre grafici di figura 4.5, si nota come nel caso precedente la decelerazione iniziale data dall'ACC per soddisfare la distanza minima; dopo pochi secondi, l'automobile *ego car* prova a raggiungere la velocità impostata  $v_{set}$  ma è obbligata a limitarla a causa dell'avvicinamento con il veicolo *lead car*, infatti dopo circa 10s, le automobili iniziano a viaggiare quasi a distanza di sicurezza (figura 4.5c) con andamento della velocità analogo (figura 4.5b), ricordando che l'accelerazione di *lead car* è di tipo sinusoidale.

Nel grafico dell'accelerazione di figura 4.5a, dopo circa 50s si notano dei piccoli picchi discontinui di decelerazione, questo è dovuto principalmente al raggiungimento della distanza  $d_{safe}$  e all'utilizzo di sensori radar per la misurazione, i quali comportano un certo ritardo temporale.

### 4.3.3 Il sistema ACC alterna il mantenimento della distanza di sicurezza e il raggiungimento della velocità impostata

L'ultima simulazione tratta lo scenario più complesso, infatti il dispositivo di controllo di velocità ACC dovrà alternare l'inseguimento della velocità impostata  $v_{set}$  con il mantenimento della distanza di sicurezza  $d_{safe}$ , perciò ci si aspetta di vedere un grafico della velocità che aumenta e diminuisce nel tempo, saturando in certi intervalli temporali al valore di velocità impostato. Sono riportati di seguito i parametri di simulazione scelti:

$x0_{lead} = 50;$

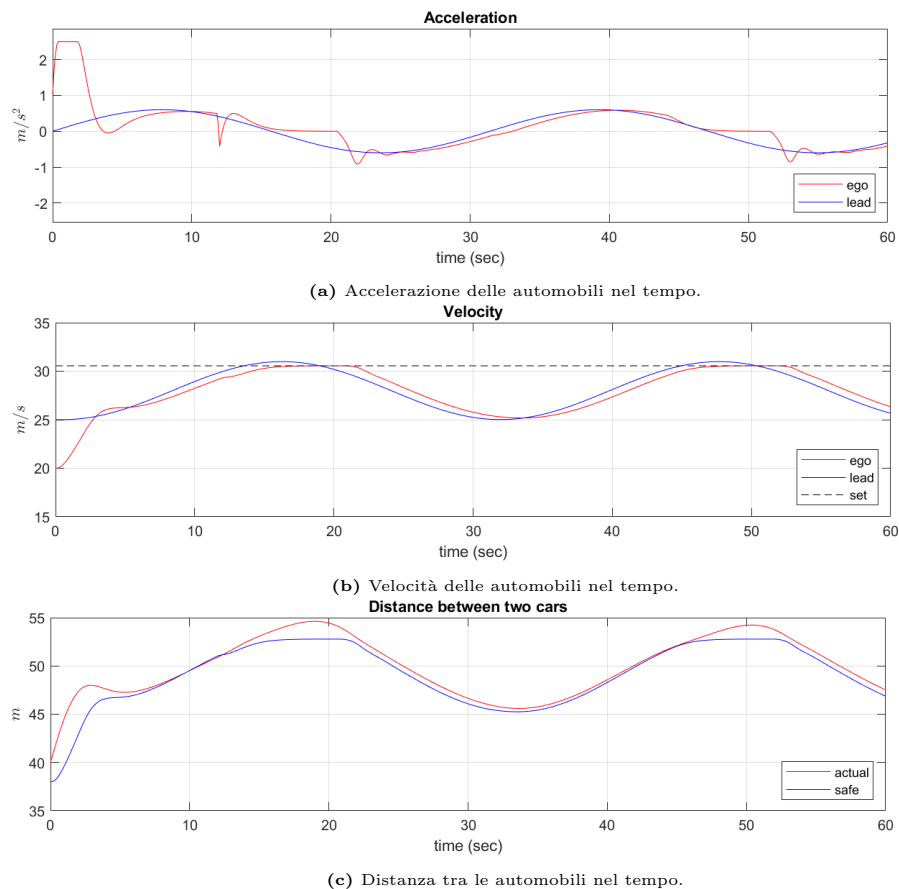
$v0_{lead} = 25; (90km/h)$

$x0_{ego} = 10;$

$v0_{ego} = 20; (72km/h)$

$v_{set} = 30.56; (110km/h)$

L'andamento delle grandezze interessanti è:



**Figura 4.6:** *Ego car* raggiunge la velocità impostata  $v_{set}$ , rallentando quando la distanza con *lead car* è vicina al valore  $d_{safe}$



Esaminando la figura 4.6, si vede immediatamente la differenza con le simulazioni precedenti. Quest'ultima, infatti, mette in evidenza la saturazione della velocità di *ego car* (figura 4.6b) nel caso in cui la distanza tra le due automobili sia troppo bassa: si confrontino gli stessi istanti temporali di figura 4.6c, specialmente verso i 20s e i 50s della simulazione.

Infine, per quanto riguarda l'accelerazione, sono presenti molti più picchi di salita e discesa, in quanto l'ACC deve alternare il controllo di mantenimento di velocità  $v_{set}$  con il controllo della distanza di sicurezza  $d_{safe}$ .



# Capitolo 5

## Conclusione

A partire dal sistema fisico illustrato in figura 1.1, è stato possibile ricavare il modello matematico in sezione 1.2, in particolare la funzione di trasferimento relativa alla relazione ingresso-uscita tra forza motrice  $[N]$  e velocità dell'automobile  $[m/s]$ .

In seguito, è stata analizzata la stabilità del sistema in catena aperta in sezione 1.4, concludendone la stabilità BIBO, nonostante ci fosse il bisogno di rispettare certe specifiche progettuali, introdotte in sezione 2.1.

Per rendere il sistema ottimale, si è scelto di procedere attraverso il controllo in retroazione negativa del veicolo mediante controllori di tipo **PID**, in particolare si sono ricavati guadagni adatti allo scopo per ognuno dei tre controllori presi in esame in capitolo 2. Come conseguenza, si è concluso che, per la realizzazione di un sistema Cruise Control classico, il controllore più adatto è il **PI** (Proporzionale-Integrale): la sua robustezza e il costo minore rispetto a un **PID** lo rendono la scelta migliore allo scopo, come analizzato in sezione 2.7. Il controllore di tipo puramente proporzionale, invece è risultato inadatto in quanto i costi per la produzione sarebbero stati insostenibili.

Per dimostrare l'efficienza del sistema in catena chiusa, in capitolo 3 si è fatto utilizzo del software *Simulink* per la simulazione del corretto funzionamento del dispositivo Cruise Control, implementato attraverso i 3 tipi di controllori citati precedentemente. In particolare, in sezione 3.5 è stata eseguita una simulazione di utilizzo del dispositivo nelle situazioni in cui il manto stradale presenta salite o discese: il sistema ha risposto in maniera ottimale, è sconsigliato però l'uso su questo tipo di strade, infatti in caso di pendenze elevate, è richiesta quasi per tutto l'intervallo temporale una forza motrice massima, che può portare a dei consumi di carburante molto alti nonché un pericolo in caso di presenza di osta-

coli nella carreggiata.

Infine, in capitolo 4 si è trattata l'innovativa tecnologia riguardante il Cruise Control Adattivo (ACC). Per questo scopo, è stata introdotta una nuova funzione di trasferimento  $G(s)$  (di equazione 4.1) in gradi di mettere in relazione accelerazione e velocità di ognuna delle due automobili. Per quanto riguarda invece il blocco ACC, è stato utilizzato su *Simulink* il blocco *Adaptive Cruise Control System Block* che permette il funzionamento corretto del sistema, comprensivo di collegamento coi modelli di entrambe le automobili.

Si sono eseguite anche in questo caso delle simulazioni via software, riferite a tre situazioni diverse possibili durante l'uso dell'assistente di guida, i cui ottimi risultati sono visualizzabili nelle figure 4.4, 4.5, 4.6.

Si può constatare perciò che grazie all'uso dell'ACC sulle automobili si è in grado di espandere il funzionamento del classico Cruise Control, grazie all'implementazioni di sensori radar o lidar per mantenere una distanza di sicurezza dal veicolo che precede durante la corsa.

In conclusione, si può affermare come il sistema elettronico Cruise Control sia un dispositivo molto utile e affidabile durante la guida, le cui evoluzioni rendono l'automobilista sempre più sicuro all'interno dell'abitacolo, oltre a condurre sempre di più verso un orizzonte di guida completamente autonoma.

# Bibliografia

- [1] M.Bisiacco, M.E. Valcher. *Controlli automatici*. Libreria Progetto Padova, Padova, 2018.
- [2] *Modeling a Cruise Control System*, <https://instruct.uwo.ca/engin-sc/391b/CTM/examples/cruise/cc.html>, The University of Michigan, last consultation: 24/08/2023.
- [3] *Adaptive Cruise Control System Using Model Predictive Control*, <https://it.mathworks.com/help/mpc/ug/adaptive-cruise-control-using-model-predictive-controller.html>, last consultation: 24/08/2023.
- [4] *Dati del Passo Pordoi*, <https://strada.bicilive.it/cicloturismo/passopordoi-bici-da-corsa/#:~:text=Dati%20del%20Passo%20Pordoi%20da,%25%20e%20massima%20del%209%25.>, last consultation: 24/08/2023.
- [5] *Cruise Control*, [https://it.wikipedia.org/wiki/Cruise\\_control](https://it.wikipedia.org/wiki/Cruise_control), last consultation: 24/08/2023.