



**UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA**



**DIPARTIMENTO  
DI INGEGNERIA  
DELL'INFORMAZIONE**

**DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE**

**CORSO DI LAUREA IN INGEGNERIA INFORMATICA**

**PROGETTAZIONE DI UN SISTEMA DI CRUISE CONTROL ADATTIVO**

**Relatore: Prof. Sandro Zampieri**

**Laureando: Alessandro Barel**

**ANNO ACCADEMICO 2022 – 2023**

**Data di laurea 21 Luglio 2023**



# Abstract

In questa tesi verrà trattato il processo di sviluppo di un sistema di cruise control adattivo per un'automobile, utilizzando i metodi classici della teoria dei controlli automatici. Partendo dallo studio del movimento dell'automobile, ricaveremo le equazioni del sistema fisico a cui applicare il controllo. Svilupperemo poi un controllore che ci permetterà di regolare la velocità dell'auto ottenendo così un sistema di cruise control. Infine, introdurremo nel modello una seconda auto ed amplieremo il sistema precedente in modo da poter controllare la distanza tra i due veicoli. Il tutto verrà testato tramite i software simulativi MatLab e Simulink per verificare l'effettivo funzionamento del sistema e il rispetto delle specifiche prefissate.



# Indice

<b>1</b>	<b>Introduzione</b>	<b>9</b>
1.1	Introduzione al cruise control . . . . .	9
1.2	Introduzione al cruise control adattivo . . . . .	9
1.3	Obbiettivo della tesi . . . . .	10
<b>2</b>	<b>Controllo della velocità</b>	<b>11</b>
2.1	Sistema fisico e parametri del controllore . . . . .	11
2.2	Stabilità e controllo in catena aperta . . . . .	13
2.3	Controllo in catena chiusa . . . . .	17
2.4	Simulazioni . . . . .	21
<b>3</b>	<b>Controllo della distanza</b>	<b>25</b>
3.1	Sistema fisico e parametri del controllore . . . . .	25
3.2	Stabilità . . . . .	27
3.3	Controllo in catena chiusa . . . . .	28
3.4	Stabilità del sistema di controllo . . . . .	33
3.5	Simulazioni . . . . .	33
<b>4</b>	<b>Conclusioni</b>	<b>41</b>



# Elenco delle figure

2.1	sistema fisico . . . . .	11
2.2	polo delle funzioni di trasferimento . . . . .	14
2.3	risposta del controllore invertendo $W_{u,\dot{y}}(0)$ . . . . .	15
2.4	risposta spostando il polo . . . . .	16
2.5	effetto del disturbo . . . . .	17
2.6	risposta con $k_{pv} = 1500$ . . . . .	19
2.7	poli e zeri con $k_{iv} = 50$ . . . . .	20
2.8	risposta con $k_{iv} = 50$ . . . . .	20
2.9	modello in Simulink . . . . .	21
2.10	simulazione con $r_v = 10$ e $d = 0$ . . . . .	21
2.11	simulazione con $r_v = 0$ e $d = 0.1$ . . . . .	22
2.12	simulazione con $r_v = 10$ e $d = 0.1$ . . . . .	22
2.13	modello in Simulink con saturatore . . . . .	23
2.14	simulazione con $r_v = 20$ e $d = 0$ . . . . .	23
2.15	simulazione di $u$ con $r_v = 20$ e $d = 0$ . . . . .	24
3.1	sistema fisico con due automobili . . . . .	25
3.2	poli e zeri di $W_{vel}$ . . . . .	28
3.3	luogo delle radici . . . . .	30
3.4	risposta con $K_{pd} = -3$ . . . . .	30
3.5	poli e zeri del sistema con $K_{pd} = -3, K_{id} = -0.5, K_{dd} = -5$ . . . . .	32
3.6	risposta del sistema con $K_{pd} = -3, K_{id} = -0.5, K_{dd} = -5$ . . . . .	32
3.7	tabella di Routh di $\text{Den}(s)$ . . . . .	33
3.8	modello Simulink . . . . .	34
3.9	posizione delle due auto con $d = 0$ . . . . .	34
3.10	distanza tra le auto con $d = 0$ . . . . .	35
3.11	posizione delle due auto con $d = 0.1$ . . . . .	35
3.12	distanza tra le auto con $d = 0.1$ . . . . .	36
3.13	modello Simulink con saturatore . . . . .	36
3.14	posizione delle auto con saturazione e $\delta_0 = 50$ . . . . .	37
3.15	posizione delle auto con saturazione e $\delta_0 = 20$ . . . . .	37
3.16	distanza tra le due auto con saturazione e $\delta_0 = 50$ . . . . .	38
3.17	distanza tra le due auto con saturazione e $\delta_0 = 20$ . . . . .	38
3.18	posizione delle due auto con i guadagni aggiornati e $\delta_0 = 50$ . . . . .	39

---

3.19	distanza tra le due auto con i guadagni aggiornati e $\delta_0 = 50$	. . . . .	39
------	--	-----------	----



# Capitolo 1

## Introduzione

### 1.1 Introduzione al cruise control

Il sistema di controllo della velocità, noto anche come cruise control (o tempomat) è uno dei sistemi di assistenza alla guida più comuni sulle auto moderne. Ciò è dovuto sia alla sua grande utilità, che alla semplicità realizzativa che lo hanno reso ormai un optional di serie in quasi tutte le automobili. Il sistema permette al veicolo su cui è equipaggiato, di mantenere una velocità selezionata senza interventi sull'acceleratore da parte del guidatore. La nascita del cruise control avviene intorno agli anni '50 come semplice controllore proporzionale con feedback. È però all'inizio degli anni '80, con l'avvento dei microprocessori, che nasce il sistema nella sua configurazione attuale dove è realizzato tramite un controllore PI (Proporzionale -integrativo).

### 1.2 Introduzione al cruise control adattivo

Un'evoluzione più moderna del sistema precedentemente descritto è il controllo di velocità adattivo, noto anche come cruise control adattivo (o ACC, dall'inglese adaptive cruise control). La velocità del veicolo viene regolata in base alla distanza da quello che lo precede e varia in base al comportamento di quest'ultimo, cercando di mantenere costante lo spazio tra le auto. Questo sistema ha un'origine molto più recente rispetto al cruise control, con i primi modelli presenti sulle strade verso la fine degli anni '90. Presenta una diffusione molto meno estesa, dovuta soprattutto agli elevati costi di realizzazione causati dal numero e la tipologia dei sensori e, da una maggiore complessità generale del sistema. Oltre che ad un beneficio dal punto di vista del comfort di guida, il cruise control adattivo permette di aumentare di molto la sicurezza, evitando collisioni dovute a distrazione o variazioni improvvise della velocità del traffico. Sono inoltre sempre più studiati i sistemi di cruise control adattivo cooperativo, in cui, oltre che ai sensori di bordo, le vetture possono essere controllate mediante segnali provenienti da altri veicoli nelle vicinanze, con l'obiettivo di ridurre la congestione stradale.

## 1.3 Obiettivo della tesi

L'obiettivo di questa tesi sarà quindi quello di sviluppare un sistema di cruise control adattivo mediante le tecniche classiche della teoria dei controlli automatici, effettuando verifiche e simulazioni tramite i software MatLab e Simulink.

## Capitolo 2

# Controllo della velocità

In questo capitolo verrà discusso il sistema di controllo della velocità (cruise control). Inizieremo descrivendo il sistema fisico che descrive il moto dell'auto trovandone le funzioni di trasferimento. Ne studieremo la stabilità e applicheremo due tipologie di controllo: il controllo in catena in aperta ed il controllo in catena chiusa. Infine simuleremo il comportamento del controllore mediante Simulink considerando anche la saturazione degli ingressi.

### 2.1 Sistema fisico e parametri del controllore

Vogliamo studiare il moto rettilineo di un'auto lungo una strada. Per farlo utilizziamo il modello semplificato rappresentato in figura 2.1 dove i vari parametri rappresentano:

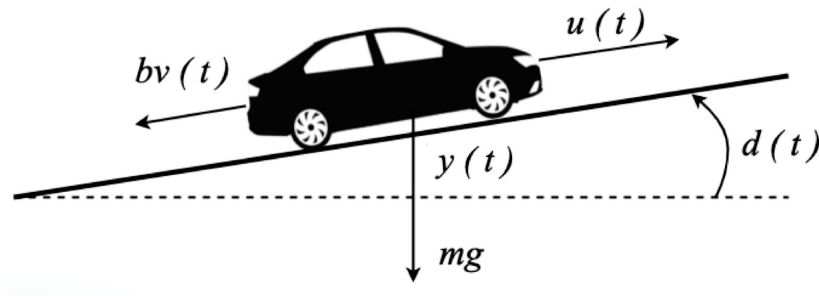


Figura 2.1: sistema fisico

- $y(t)$  : è la posizione dell'auto.
- $u(t)$ : è la forza orizzontale che agisce sull'auto. Questa è usata per rappresentare sia l'azione del motore che quella dei freni.
- $b$ : è il coefficiente di attrito. Questo ingloba sia l'effetto dell'attrito viscoso che quello di rotolamento tra la strada e i pneumatici.

- $g$ : è l'accelerazione di gravità.
- $m_1$ : è la massa dell'auto.
- $d(t)$ : è l'inclinazione della strada.

Il moto sarà quindi determinato da tre forze principali: la forza del motore  $u(t)$ , la forza di attrito  $F_a = b\dot{y}(t)$  e la forza peso parallela  $F_{//} = m_1g\sin(d(t))$ .

### Equazioni del moto

Una volta definito il modello fisico possiamo ricavare le equazioni del moto dell'automobile. Queste ultime ci serviranno per rappresentare le dipendenze tra: l'ingresso, il disturbo e l'uscita del nostro sistema che sono rispettivamente  $u$ ,  $d$  e  $\dot{y}$ .

$$m_1\ddot{y}(t) = u(t) - b\dot{y}(t) - m_1g\sin(d(t)) \quad (2.1)$$

La presenza del seno nell'equazione rende il sistema non lineare. È necessario effettuare una linearizzazione. Nel nostro caso possiamo sfruttare l'approssimazione per i piccoli angoli, in quanto,  $d(t)$  non avrà valori elevati.

$$m_1\ddot{y}(t) = u(t) - b\dot{y}(t) - m_1gd(t) \quad (2.2)$$

### Funzioni di trasferimento

Dalle equazioni del moto linearizzate possiamo poi ricavare le funzioni di trasferimento. Appliciamo alla 2.2 la trasformata di Laplace in modo da passare dal dominio del tempo a quello delle frequenze:

$$m_1s\dot{Y}(s) = U(s) - b\dot{Y}(s) - m_1gD(s) \quad (2.3)$$

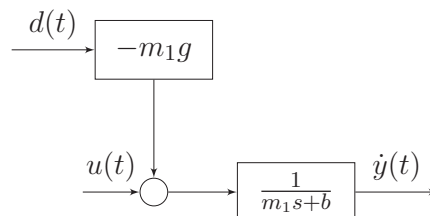
Con alcuni opportuni raccoglimenti possiamo poi evidenziare il contributo dei singoli ingressi verso l'uscita:

$$\dot{Y}(s) = \frac{1}{m_1s + b}U(s) + \frac{-gm_1}{m_1s + b}D(s) \quad (2.4)$$

Abbiamo ottenuto quindi due funzioni di trasferimento che descrivono il contributo degli ingressi  $u(t)$  e  $d(t)$  sull'uscita:

$$W_{u,\dot{y}}(s) = \frac{1}{m_1s + b}, W_{d,\dot{y}} = \frac{-gm_1}{m_1s + b} \quad (2.5)$$

Possiamo infine riassumere il sistema mediante il seguente schema a blocchi:



### Parametri del modello

Andiamo ora a specificare i valori dei parametri della funzione di trasferimento. Consideriamo un'auto di taglia media per cui un buon valore per la massa potrebbe essere  $m_1 = 1800\text{kg}$ . Per la forza  $u$  consideriamo invece valori nell'intervallo  $[-18000\text{N}, 18000\text{N}]$  che rappresentano la massima forza frenante e accelerante. Per il coefficiente di attrito, considerando un moto su asfalto, una buona approssimazione potrebbe essere  $b = 50 \frac{\text{Ns}}{\text{m}}$ . Infine, per l'inclinazione della strada, assumeremo valori di  $d$  di massimo  $0.1\text{rad}$ , che corrispondono ad un'inclinazione del 10%.

### Specifiche di controllo

Prima di proseguire con la progettazione del controllore è necessario indicare le specifiche che esso dovrà rispettare. Mediante alcuni test è stato possibile osservare che a potenza massima l'automobile riusciva a raggiungere, in un tempo inferiore ai 5 secondi, valori di velocità intorno ai  $40\text{m/s}$  (circa  $145\text{km/h}$ ). Possiamo quindi considerare ragionevole un tempo di salita  $T_s < 5\text{s}$ . Inoltre ammettiamo una sovraelongazione massima  $S < 10\%$  ed un errore a regime  $e_s < 5\%$ .

## 2.2 Stabilità e controllo in catena aperta

In questo paragrafo verrà trattata la progettazione di un controllore in catena aperta per il controllo della velocità. In catena aperta l'ingresso  $u(t)$  prodotto dal controllore sarà basato unicamente sul modello fisico del sistema e sulla conoscenza del riferimento  $r_v(t)$  che vogliamo che sia seguito. Questo controllo è apprezzato per la sua semplicità. Tuttavia, la mancanza di un monitoraggio dell'uscita potrebbe causare scostamenti dal riferimento nel caso di un modello non del tutto esatto oppure nel caso di una variazione di alcuni parametri.

### Stabilità

Prima di iniziare la progettazione è importante verificare la stabilità del nostro sistema. Per fare ciò possiamo studiare la posizione dei poli delle funzioni di trasferimento, lo facciamo usando MatLab che fornirà il grafico in figura 2.2.

```
m1=1800; %massa del veicolo
b=50; %coefficiente di attrito
g=9.81; %accelerazione di gravita
s=tf('s'); %istruzione per inserire s nelle formule
Wuy=1/(m1*s+b); %funzione di trasferimento per l'ingresso
Wdy=-m1*g/(m1*s+b); %funzione di trasferimento per l'uscita
figure();
pzmap(Wuy) %poli e zeri di Wuy
figure()
pzmap(Wdy) %poli e zeri di Wdy
```

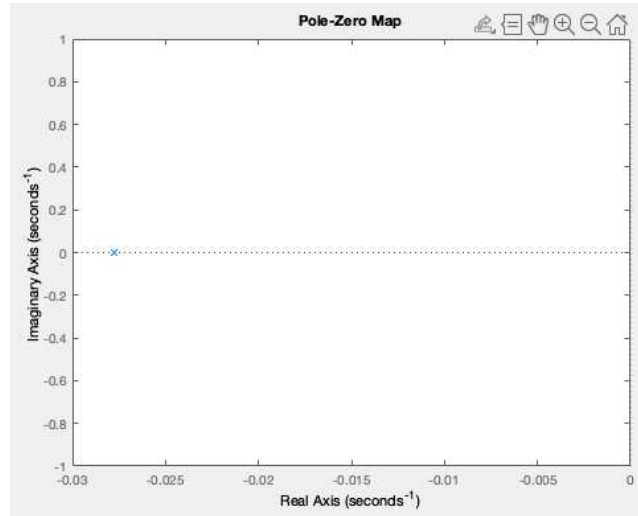
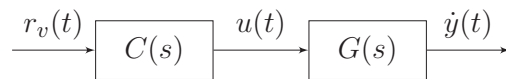


Figura 2.2: polo delle funzioni di trasferimento

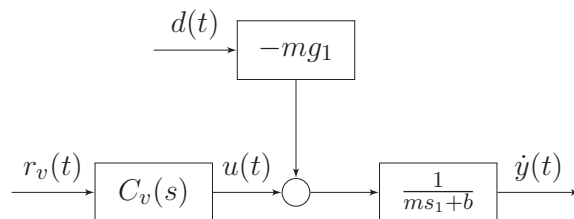
Possiamo osservare che entrambe le funzioni di trasferimento presentano un polo reale negativo, in  $-\frac{b}{m_1}$ . Questo ci permette di affermare che il sistema è BIBO stabile.

### Controllo in catena aperta

Una volta determinata la stabilità possiamo procedere con il controllo in catena aperta. Lo schema di base del controllo è il seguente:



dove  $G(s)$  rappresenta la funzione di trasferimento del sistema fisico, mentre  $C(s)$  quella del controllore. Adattandolo al nostro sistema abbiamo quindi:



dove  $C_v(s)$  rappresenta il controllore che stiamo progettando. Vogliamo fare in modo che l'uscita segua il riferimento. Un modo per ottenere questo risultato è invertire il guadagno in continua:

$$C_v(s) = \frac{\alpha}{W_{u,\dot{y}}(0)} = b\alpha \quad (2.6)$$

Con cui ricaviamo la seguente funzione di trasferimento per il sistema in catena aperta:

$$W_{r,y}(s) = C_v(s)W_{u,y}(s) = \frac{b\alpha}{m_1s + b} \quad (2.7)$$

Utilizziamo MatLab per simulare la risposta del sistema, useremo un valore del riferimento  $r_v = 10$ .

```
alpha=1;      %coefficiente proporzionale
Wuy=1/(m1*s+b); %fdt tra u e y
Cv=alpha*b(m1*s+b); %controllore
Wry=Cv*Wuy;   %funzione di trasferimento r e y
rv=10;        %riferimento
step(rv*Wry); %risposta al gradino
```

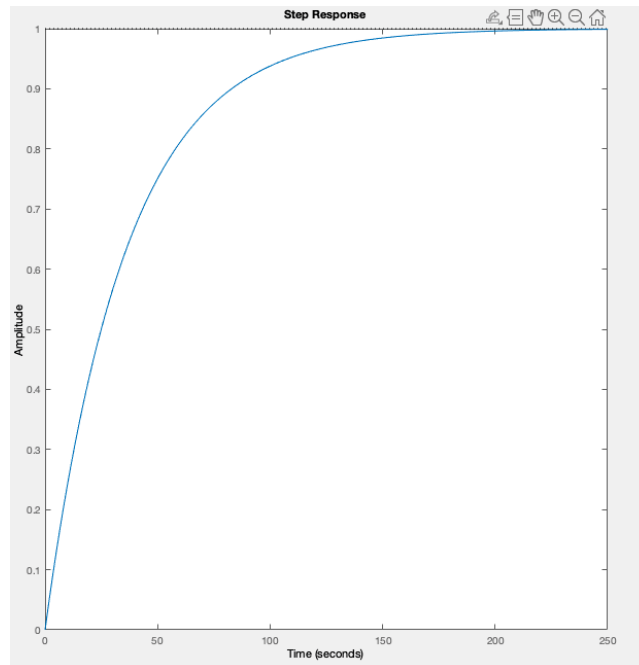


Figura 2.3: risposta del controllore invertendo  $W_{u,y}(0)$

Nella figura 2.3 possiamo osservare che l'uscita riesce a seguire il riferimento senza sovraelongazione ed errore a regime, allo stesso tempo però, il tempo di salita è molto oltre le specifiche. Questo è causato dal polo  $-\frac{b}{m_1}$ . Per velocizzare il sistema posso usare un controllore che ne modifica la posizione. Un esempio potrebbe essere:

$$C_v(s) = \alpha \frac{m_1s + b}{1 + s\tau} \rightarrow W_{r,y} = \alpha \frac{1}{1 + s\tau} \quad (2.8)$$

```

alpha=1;%coefficiente proporzionale
tau=1;%costante di tempo del controllore
Wuy=1/(m1*s+b); %fdt tra u e y
Cv=alpha*((m1*s+b)/(1+tau*s)); %controllore
Wry=Cv*Wuy; %fdt tra r e y
rv=10; %riferimento
step(rv*Wry); %risposta al gradino

```

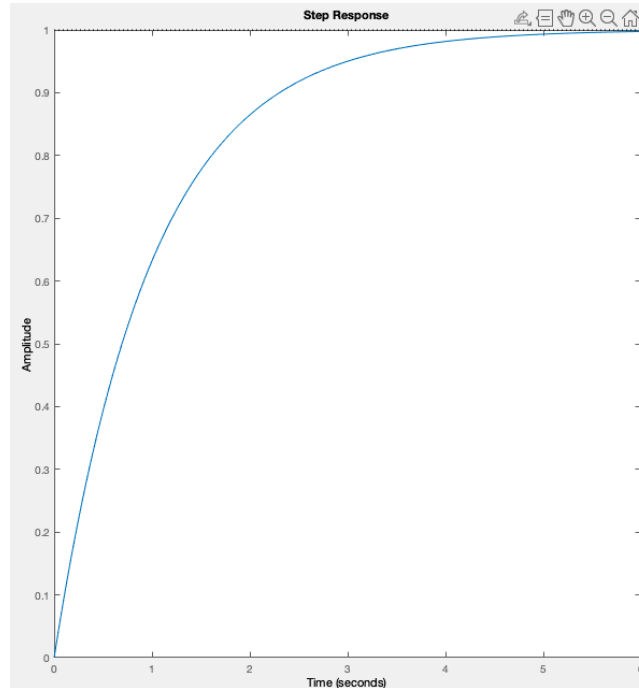


Figura 2.4: risposta spostando il polo

Osservando la figura 2.4 notiamo che il tempo di salita sarà ora pari a 3.9s che rientra nelle specifiche.

Verifichiamo ora l'effetto che ha il disturbo  $d(t)$  sul valore dell'uscita. Ricordiamo la funzione di trasferimento  $W_{d,y} = \frac{-m_1 g}{m_1 s + b}$ . Utilizziamo il seguente codice MatLab dove simuliamo la risposta al gradino con un inclinazione di 0.1rad ( $\approx 10\%$  di inclinazione). Esso produrrà il grafico in figura 2.5.

```

Wdy=-m1*g/(m1s+b); %funzioni di trasferimento tra d e y
d=0.1; %inclinazione della strada
step(d*Wdy); %risposta al gradino

```



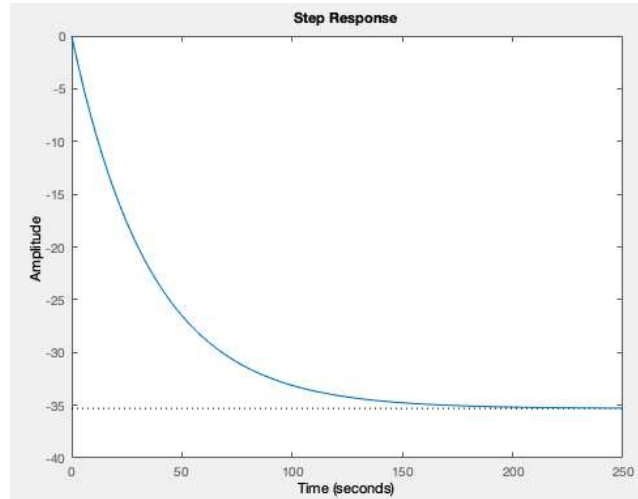


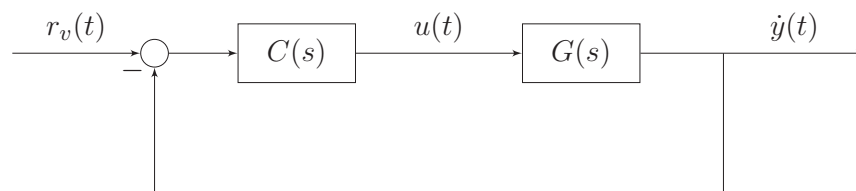
Figura 2.5: effetto del disturbo

### Conclusioni

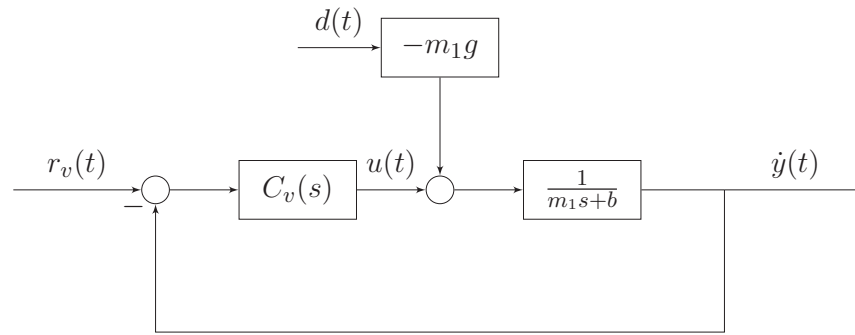
Come abbiamo potuto osservare, utilizzando un controllore in catena aperta siamo riusciti a progettare un controllore di velocità capace di rispettare le specifiche. Allo stesso tempo però, il mancato monitoraggio dell'uscita fa sì che gli effetti del disturbo, della variazione dei parametri o eventuali errori sul modello non vengano corretti, portando a scostamenti dal riferimento. Un controllore di questo genere non è quindi la soluzione più adatta.

## 2.3 Controllo in catena chiusa

I problemi riscontrati finora possono essere risolti utilizzando il controllo in catena chiusa, nel quale, a differenza del caso precedente, viene tenuto conto del valore dell'uscita. Questo metodo verifica istante per istante l'errore di inseguimento, calcolando la differenza tra il valore dell'uscita e quello del riferimento. In questo modo vengono automaticamente corretti gli effetti dei disturbi (anche senza misurarli), ed eventuali errori dovuti al modello o alla variazione dei parametri. Lo schema di base del controllo in catena chiusa è il seguente:



che applicato al nostro sistema fisico diventa:



Nel controllo in catena chiusa possono essere usati un grande numero di controllori diversi. Nella nostra analisi ci concentreremo sui controllori della famiglia PID, ovvero: proporzionale, integrativo, derivativo.

### Controllore proporzionale P

Il controllore proporzionale  $C_v(s) = K_{pv}$  è uno dei controllori più semplici ma allo stesso tempo più utilizzati. La sua uscita consiste semplicemente nell'errore di inseguimento moltiplicato per il coefficiente proporzionale  $K_{pv}$ . Aumentando il valore di  $K_{pv}$  aumenta la risposta del nostro sistema, ma se il valore è troppo grande, il sistema può diventare instabile e oscillare. Se il valore di  $K_{pv}$  è troppo piccolo invece è probabile che il nostro sistema non riesca a seguire il riferimento. Le funzioni di trasferimento del nostro sistema diventano:

$$W_{r,\dot{y}}(s) = \frac{C_v(s)W_{u,\dot{y}}(s)}{1 + C_v(s)W_{u,\dot{y}}(s)} = \frac{K}{m_1s + b + K} \quad (2.9)$$

$$W_{d,\dot{y}}(s) = \frac{W_{d,\dot{y}}(s)}{1 + C_v(s)W_{u,\dot{y}}(s)} = \frac{-m_1g}{m_1s + b + K} \quad (2.10)$$

Per trovare un valore di  $K_{pv}$  adatto alle specifiche sfruttiamo il luogo delle radici. Questa tecnica ci permette di osservare lo spostamento dei poli della funzione di trasferimento al variare del parametro  $K_{pv}$ . Per ottenere un controllore che rispetti le specifiche posso tracciare nel luogo delle radici il coefficiente di smorzamento  $\xi$  e la pulsazione naturale  $\omega_n$  che ci indicano rispettivamente: il grado di sovraelongazione ed il tempo di salita del sistema.

$$\xi \leq \sqrt{\frac{\ln^2 S}{\pi^2 + \ln^2 S}} \approx 0.6, \omega_n = \frac{1.8}{T_s} \approx 0.36 \quad (2.11)$$

Utilizziamo quindi uno script MatLab che permette di interagire con il luogo delle radici scegliendo la posizione del polo. Poi, per il valore corrispondente di  $K_{pv}$  viene simulata la risposta del sistema al gradino con  $r_v = 10$ .

```
figure();
rlocus(Cv*Wuy); %luogo delle radici
```

```

axis([-2 0 -1 1])
sgrid(0.6,0.36); %per mostrare nel grafico xi e wn
%istruzione per interagire con il luogo
[K,poles]=rlocfind(Cv*Wuy)
Cv=K; %controllore
%funzione di trasferimento ingresso-uscita
Wry=Cv*Wuy/(1+Cv*Wuy);
figure();
step(rv*Wry); %risposta al gradino
stepinfo(rv*Wry) %comando per info su risposta al gradino

```

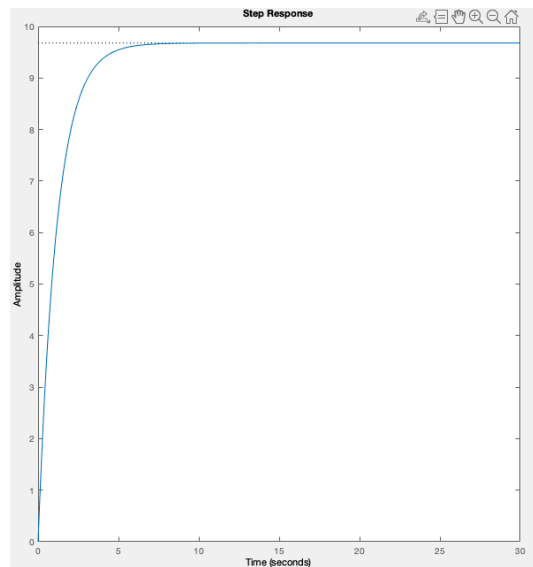


Figura 2.6: risposta con  $k_{pv} = 1500$

Ci accorgiamo che il sistema segue le specifiche solo per valori di  $K_{pv}$  molto elevati, per esempio  $K_{pv} = 4000$ , con tempi di salita ben poco realistici per la tipologia di auto che stiamo considerando. Un guadagno così elevato richiederebbe al motore una forza maggiore rispetto a quella che è capace di generare. Dobbiamo quindi selezionare un valore più basso. Vediamo in figura 2.6 che  $K_{pv} = 1500$  permette di avere tempi di salita più realistici a discapito però del valore a regime che sarà minore del voluto. Da questo possiamo dedurre che un controllore proporzionale non è sufficiente per il nostro scopo. Siamo costretti a usarne uno più complesso.

### Controllore Proporzionale Integrativo PI

Il controllore proporzionale integrativo  $C_v(s) = K_{pv} + \frac{K_{iv}}{s}$  è un altro controllore della famiglia dei PID. Combina un' azione proporzionale ed una integrativa ed in generale riesce a fornire prestazioni migliori rispetto al semplice controllore proporzionale. L'integratore agisce accumulando l'errore passato permettendo la correzione dell'errore a regime nella risposta, aumentando la precisione del sistema. Le funzioni di

trasferimento con il PI diventano:

$$W_{r,\dot{y}}(s) = \frac{C_v(s)W_{u,\dot{y}}(s)}{1 + C_v(s)W_{u,\dot{y}}(s)} = \frac{K_{pv}s + K_{iv}}{m_1s^2 + (K_{pv} + b)s + K_{iv}} \quad (2.12)$$

$$W_{d,\dot{y}}(s) = \frac{W_{d,\dot{y}}(s)}{1 + C_v(s)W_{u,\dot{y}}(s)} = \frac{-m_1gs}{m_1s^2 + (K_{pv} + b)s + K_{iv}} \quad (2.13)$$

Come valore del parametro proporzionale  $K_{pv}$  possiamo usare 1500 che abbiamo visto darci buone prestazioni riguardanti il tempo di salita e la sovraelongazione. Rimane quindi da ricavare  $K_{iv}$ . Possiamo farlo con un processo simile al precedente: guardiamo come varia la posizione dei poli al variare del parametro e scegliamo un valore consono con le specifiche. Dalla figura 2.7 e 2.8 possiamo notare che, con un valore di  $K_{iv} = 50$ , i poli della funzione di trasferimento sono entro la zona desiderata per la sovraelongazione. Inoltre il sistema presenta valori entro le specifiche per tempo di salita (2.6s) e l'errore a regime (0.5%).

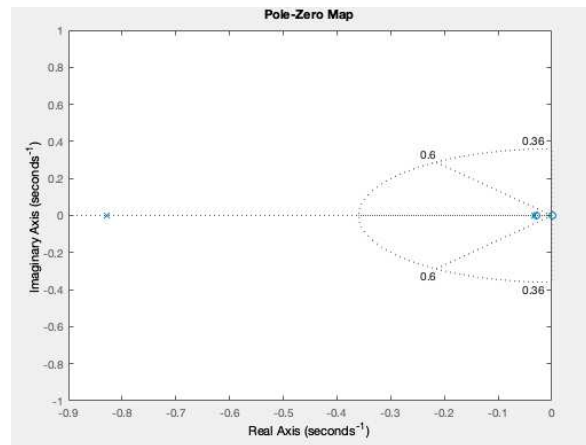


Figura 2.7: poli e zeri con  $k_{iv} = 50$

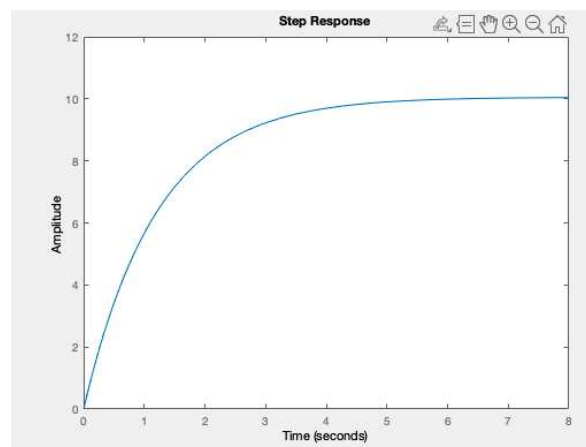


Figura 2.8: risposta con  $k_{iv} = 50$

## 2.4 Simulazioni

Una volta determinato il controllore possiamo iniziare ad effettuare delle simulazioni sul comportamento del nostro sistema. Queste verranno fatte mediante il software Simulink che ci consente di rappresentare il sistema mediante uno schema a blocchi e di simulare la risposta al variare degli ingressi (figura 2.9).

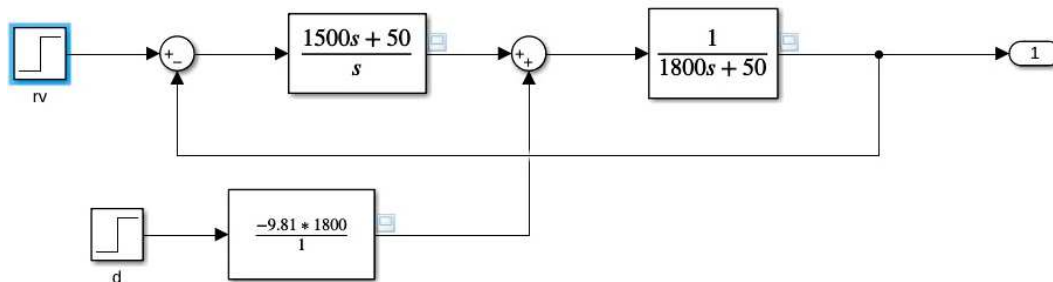


Figura 2.9: modello in Simulink

### Riferimento e disturbo

Per prima cosa verifichiamo le simulazioni fatte in precedenza con MatLab, simulando con un riferimento  $r_v = 10m/s$ .

Come possiamo vedere in figura 2.10 l'uscita segue il comportamento desiderato rimanendo entro le specifiche. Controlliamo ora l'efficacia del sistema nell'attenuare il disturbo. Simuliamo con un valore dell'inclinazione di  $0.1rad$ .

L'effetto del disturbo (figura 2.11), a differenza che nel controllo in catena aperta, viene azzerato. Verifichiamo il comportamento unendo i due effetti. Come possiamo osservare, (figura 2.12) il controllore riesce a portare comunque l'auto alla velocità desiderata.

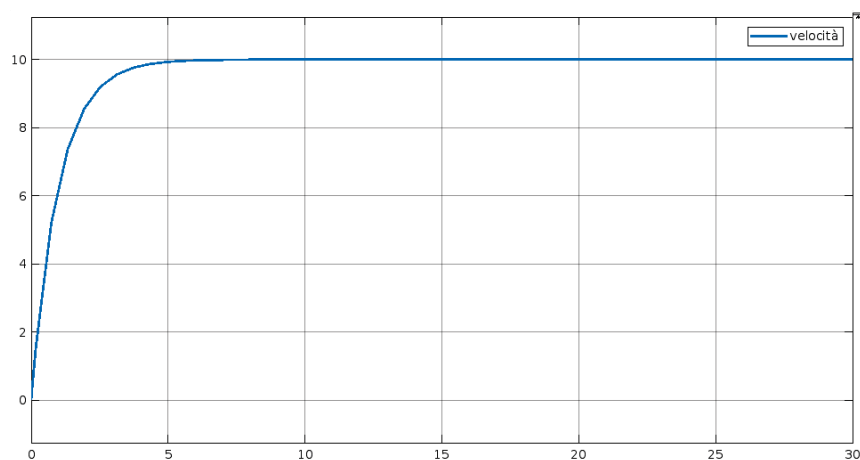
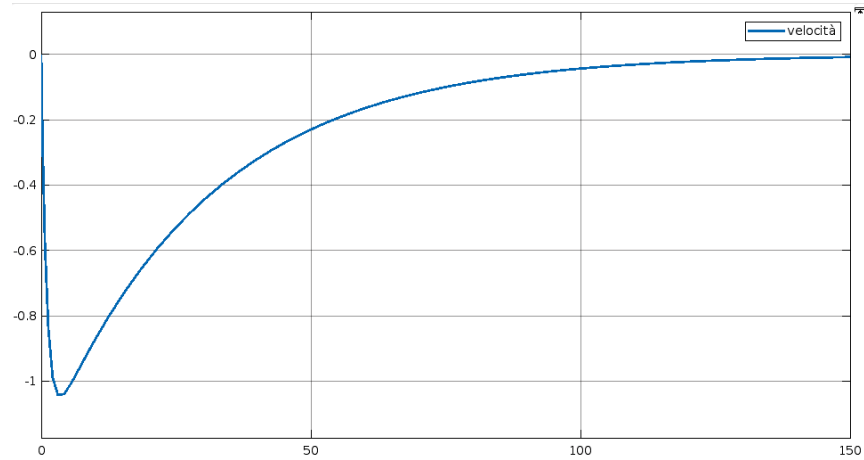
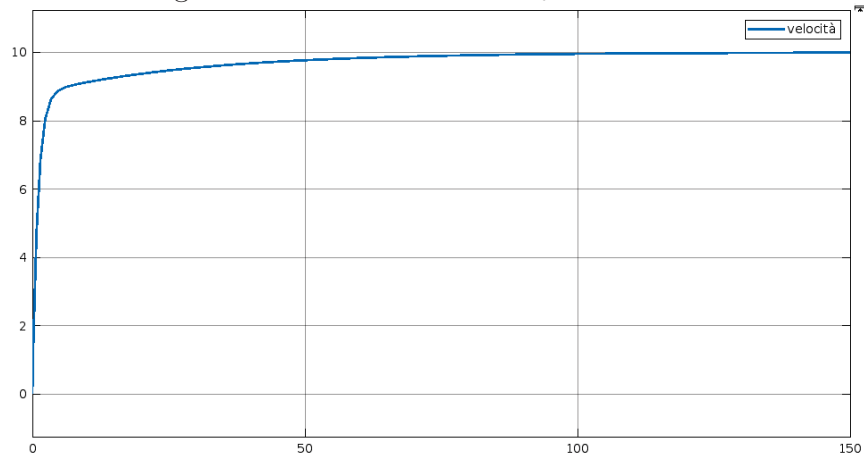


Figura 2.10: simulazione con  $r_v = 10$  e  $d = 0$

Figura 2.11: simulazione con  $r_v = 0$  e  $d = 0.1$ Figura 2.12: simulazione con  $r_v = 10$  e  $d = 0.1$ 

### Saturazione

All'aumentare del valore del riferimento  $r_v$ , aumenterà anche la forza richiesta dal motore che non può andare oltre i  $18000N$  permessi dalle specifiche. Per realizzare quindi, un controllore che sia più realistico, possiamo inserire nel nostro sistema un blocco saturatore (figura 2.13) che ci permette di limitare l'uscita del controllore PI entro i valori permessi.

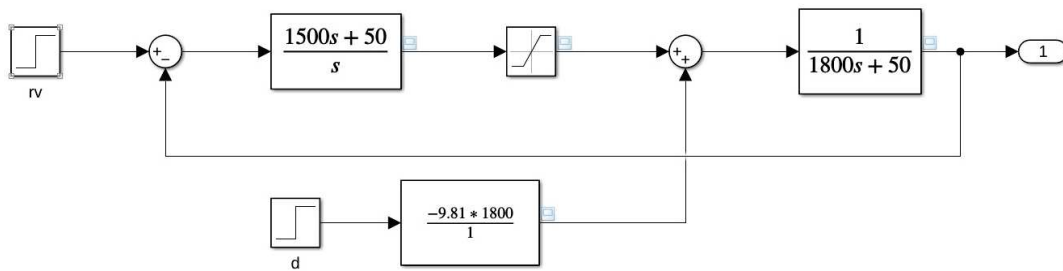


Figura 2.13: modello in Simulink con saturatore

Tramite Simulink mostriamo sia la velocità che l'uscita del controllore con un valore del riferimento pari a  $20\text{m/s}$  (figura 2.14 e 2.15).

Con un tale valore di  $r_v$  il nostro controllore produce un ingresso desiderato (in blu) maggiore del valore massimo consentito che viene limitato (in rosso) dal saturatore. In ogni caso le prestazioni rimangono entro le specifiche.

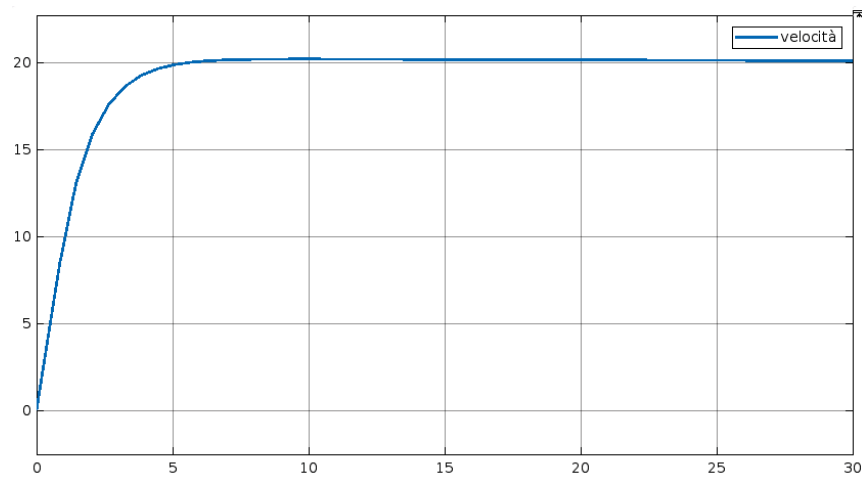


Figura 2.14: simulazione con  $r_v = 20$  e  $d = 0$

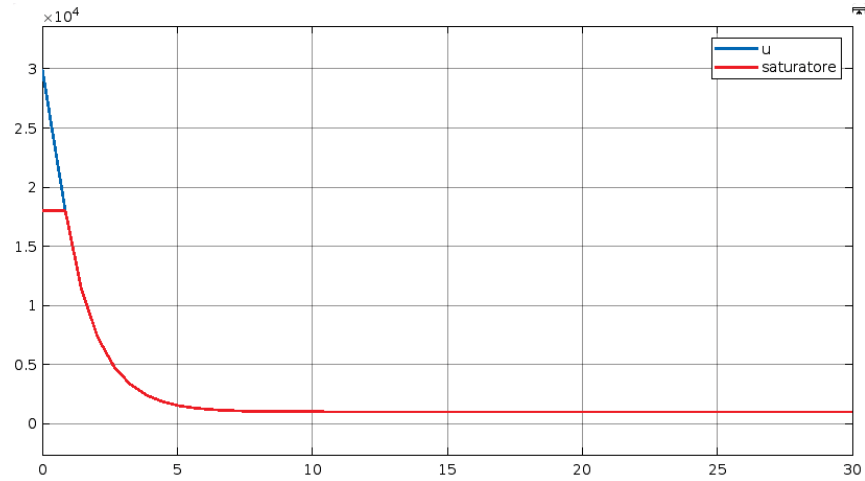


Figura 2.15: simulazione di  $u$  con  $r_v = 20$  e  $d = 0$



## Capitolo 3

# Controllo della distanza

In questo capitolo verrà tratta la progettazione di un controllore per la distanza tra due automobili. Amplieremo il sistema per il controllo della velocità aggiungendo una seconda auto. Utilizzeremo poi le metodologie del precedente capitolo per sviluppare un controllore capace di effettuare il mantenimento della distanza. Il tutto verrà verificato mediante simulazioni effettuate tramite il software Simulink.

### 3.1 Sistema fisico e parametri del controllore

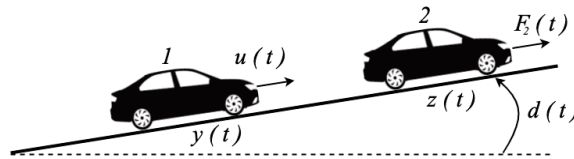


Figura 3.1: sistema fisico con due automobili

Vogliamo controllare la distanza tra due veicoli, per praticità chiameremo l'auto controllata "host" e quella da seguire "target". Inserisco l'auto target nel sistema fisico utilizzato per il controllo della velocità.

#### Equazioni del moto

Dal sistema fisico otteniamo le seguenti equazioni del moto:

$$\begin{cases} m_1 \ddot{y}(t) = u(t) - b\dot{y}(t) - m_1 g \sin(d(t)) \\ m_2 \ddot{z}(t) = f_2(t) - b\dot{z}(t) - m_2 g \sin(d(t)) \\ \delta(t) = z(t) - y(t) \end{cases} \quad (3.1)$$

Dove  $y$  è la posizione dell'auto host mentre  $z$  e  $f_2$  rappresentano rispettivamente la posizione e la forza prodotta dal motore di quella target. La distanza tra le due verrà invece indicata con  $\delta$ .

### Funzioni di trasferimento

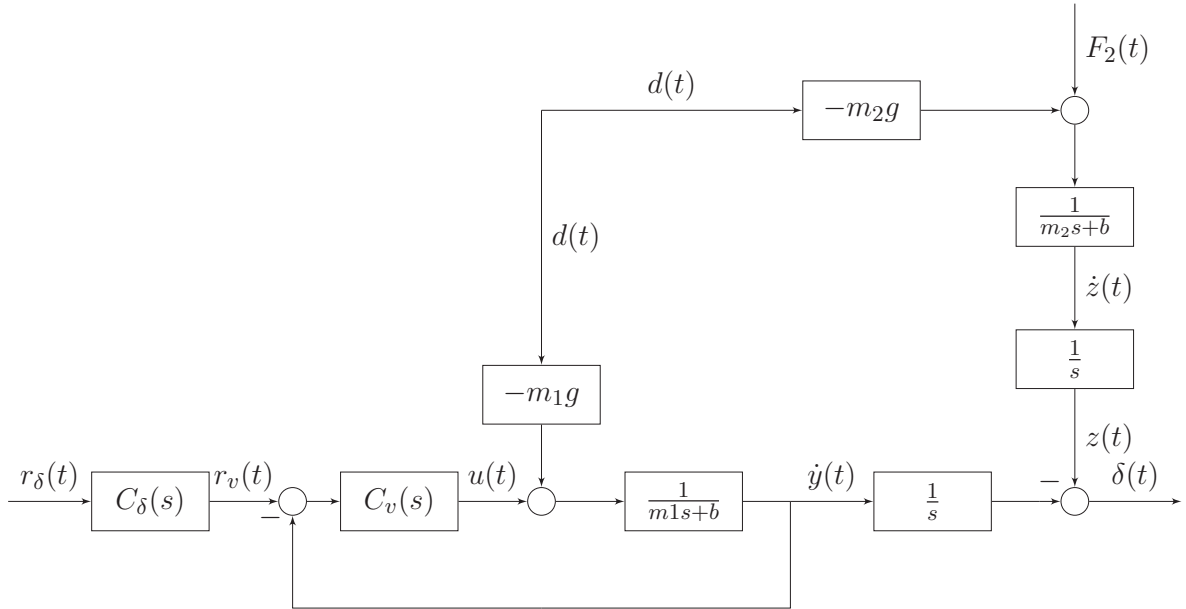
Come visto in precedenza, applicando la trasformata di Laplace e delle opportune sostituzioni, ricaviamo un'equazione che mostra le relazioni tra gli ingressi e l'uscita.

$$\Delta(s) = \frac{1}{s(m_2s + b)}F_2(s) + \left[\frac{m_1g}{s(m_1s + b)} - \frac{m_2g}{s(m_2s + b)}\right]D(s) - \frac{1}{s(m_1s + b)}U(s) \quad (3.2)$$

L'uscita in questo sistema sarà rappresentata da  $\Delta(s)$ , in quanto la distanza è la variabile che vogliamo controllare. Dalla 3.2 possiamo quindi estrarre le funzioni di trasferimento per ogni ingresso del sistema:

$$W_{f,\delta} = \frac{1}{s(m_2s + b)}, W_{u,\delta} = \frac{1}{s(m_1s + b)}, W_{d,\delta} = \frac{m_1g}{s(m_1s + b)} - \frac{m_2g}{s(m_2s + b)} \quad (3.3)$$

Possiamo inserire queste funzioni all'interno dello schema a blocchi per il controllo della velocità. Otteniamo il seguente diagramma per il sistema di controllo, dove  $C_\delta$  rappresenta il controllore che vogliamo sviluppare.



### Parametri del controllore

Per i parametri del modello utilizzeremo gli stessi valori usati in precedenza. Riassumendo:

- $m_1 = 1800kg$ .
- $m_2 = 1800kg$ .

- $-18000 < u(t) < 18000$ .
- $-18000 < f_2(t) < 18000$ .
- $b = 50 \frac{Ns}{m}$ .
- $d < 0.1rad$ .

Eseguendo dei test con l'auto a potenza massima vediamo che in un tempo inferiore ai 5s riesce a raggiungere una distanza di circa 100m. Un tempo di salita  $T_s < 5s$  è quindi ragionevole. Per la sovraelongazione e l'errore a regime invece, prendiamo rispettivamente valori massimi del 10% e del 5%.

### Parametri per i test

Per controllare il comportamento dell'auto target, oltre ad  $f_2$ , sarà necessario aggiungere la distanza iniziale  $\delta_0$  e la velocità  $v_0$ . Per come è costruito il sistema  $v_0$  può rappresentare due concetti diversi:

- La velocità iniziale del target, se consideriamo che l'host parta da ferma.
- La differenza di velocità tra le due auto a  $t = 0$ .

Considerando il nostro utilizzo, risulterà più conveniente la seconda interpretazione. Dobbiamo definire un range per i precedenti parametri, poiché il valore influisce sulle prestazioni del sistema. Valori realistici potrebbero essere

- $\delta_0 < 150m$
- $v_0 < 20m/s$

In quanto non verrà mai richiesto al sistema di inseguire un'auto molto lontana o che procede ad una velocità di gran lunga superiore a quella attuale. Nelle simulazioni per determinare il controllore useremo i seguenti valori dei parametri:

- $f_2 = 500N$
- $r_\delta = 10m$
- $\delta_0 = 50m$
- $v_0 = 10m/s$

## 3.2 Stabilità

Il nuovo sistema da controllare è rappresentato dalla funzione di trasferimento in retroazione del controllo della velocità:

$$W_{vel} = \frac{C_v W_{r,\dot{y}}}{1 + C_v W_{r,\dot{y}}} = \frac{K_{pv}s + K_{iv}}{m_1 s^2 + (K_{pv} + b)s + K_{iv}} \quad (3.4)$$

Ne verifichiamo la stabilità mediante il seguente script MatLab:

```
m1=1800; %massa del veicolo
%parametri controllore velocita
Kpv=1500;
Kiv=50;
s=tf('s');
%funzione di trasferimento controllore velocita
Wvel=(Kpv*s+Kiv)/(m1*s^2+(Kpv+b)*s+Kiv);
pzmap(Wvel); % poli e zeri di Wvel
```

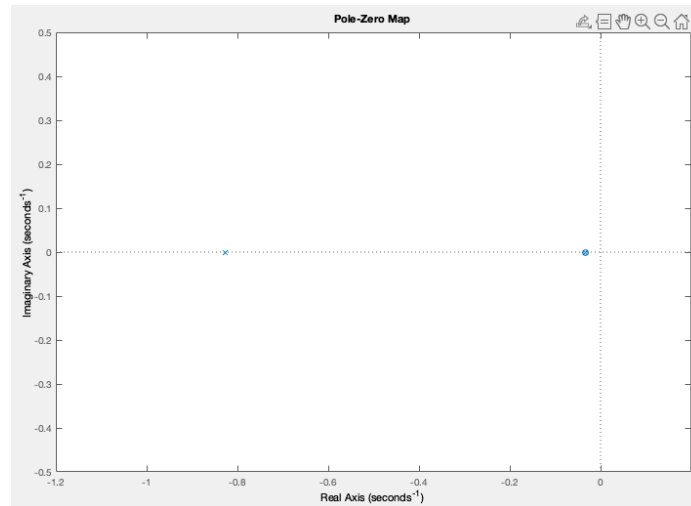
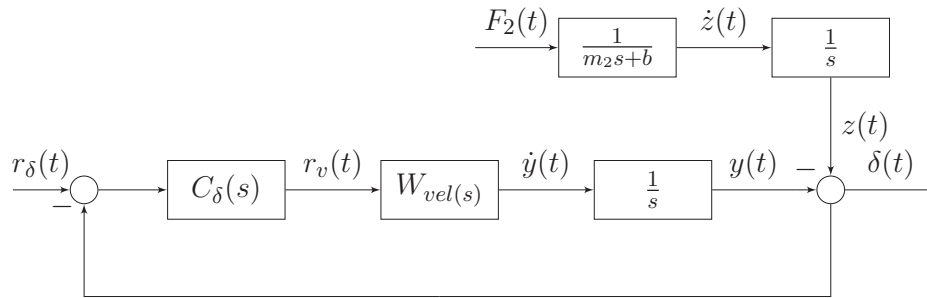


Figura 3.2: poli e zeri di  $W_{vel}$

Come possiamo osservare nella figura 3.2  $W_{vel}$  presenta poli reali minori di zero, quindi, possiamo affermare che è BIBO stabile.

### 3.3 Controllo in catena chiusa

Inizieremo la progettazione del controllore direttamente con il controllo in catena chiusa. Come visto nel capitolo precedente questa configurazione consente di ottenere prestazioni migliori in quanto, permette la reiezione dei disturbi e la correzione dell'errore di inseguimento. Estendiamo lo schema generale definito in precedenza al nuovo sistema. Per semplificare la rappresentazione il sistema di controllo della velocità viene sostituito con  $W_{vel}$ . Sono inoltre stati rimossi i disturbi, il cui effetto verrà verificato più tardi con Simulink.



Dallo schema a blocchi otteniamo le seguenti funzioni di trasferimenti per gli ingressi:

$$W_{r_\delta, \delta}(s) = \frac{-W_{vel} C_\delta \frac{1}{s}}{1 - W_{vel} C_\delta \frac{1}{s}}, W_{f_2, \delta} = \frac{\frac{1}{m_2 s + b} \frac{1}{s}}{1 - W_{vel} C_\delta \frac{1}{s}} \quad (3.5)$$

Prima di iniziare la ricerca del controllore è importante fare una precisazione sul valore dei guadagni. Nel calcolo della distanza viene invertito il segno di  $y$ , ciò fa sì che il controllore  $C_\delta$  avrà un'azione inversamente proporzionale sull'uscita. Per ottenere questo comportamento, come vedremo, utilizzeremo guadagni negativi.

### Controllore proporzionale P

Iniziamo la ricerca del controllore dal caso più semplice, il controllore proporzionale. Utilizziamo nuovamente uno script MatLab per selezionare il valore di  $K_{pd}$  interagendo con il luogo delle radici. Simuleremo poi il comportamento del sistema con il valore scelto ed i parametri precedentemente specificati.

```
%parametri fisici
m1=1800;
m2=1800;
b=50;
g=9.81;
d=0;
%parametri secondo veicolo
F2=500;
delta0=50;
v0=10;
%riferimento
rd=10;
%parametri controllore velocita
Kpv=1500;
Kiv=50;
%fdt controllo velocita
Wvel= (kpv*s+kiv)/(m1*s^2 + (kpv +b)*s +kiv);
%fdt auto target
Wfz= (F2*(1/(m2*s+b)) + v0)/s +delta0;
rlocus(Wvel/s); % luogo delle radici
sgrid(0.6,2);
axis([-1.5 0.5 -0.9 0.9]);
```

```

[k,poles]=rlocfind(Wvel)
Kpd=-k;
Cd=Kpd; %controllore
Wry=Cd*Wvel*(1/s); %fdt tra r e y
%retroazione prima auto
Wrdist=-Wry/(1-Wry);
%retroazione seconda auto
Wfdist= Wfz/(1-Wry)
figure();
step(Wfdist+rd*Wrdist); %risposta al gradino
axis([-1 30 -1 50]);

```

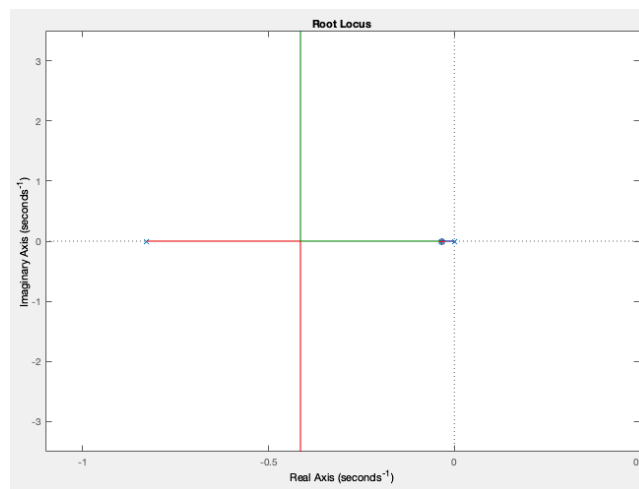
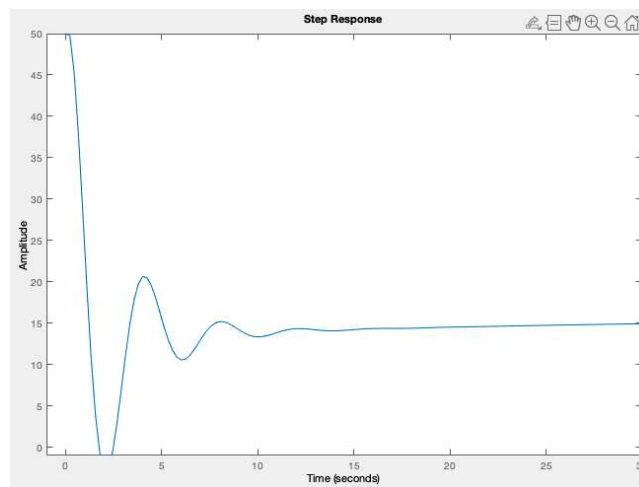


Figura 3.3: luogo delle radici

Figura 3.4: risposta con  $K_{pd} = -3$

Come possiamo vedere dalle 3.3 e 3.4 troviamo che con valori di  $K_{pd}$  vicini a  $-3$  otteniamo un tempo di salita entro le specifiche desiderate di  $5s$ . Notiamo però che l'errore a regime e la sovraelognazione sono ben oltre il livello sperato. È necessario utilizzare un controllore più complesso.

### Controllore proporzionale integrativo Derivativo PID

Per ridurre contemporaneamente sia l'errore a regime che la sovraelongazione del sistema, possiamo usare un controllore Proporzionale Integrativo Derivativo. Il PID possiede tutte le qualità del controllore PI, usato in precedenza, in aggiunta ad una componente Derivativa :  $PID(S) = K_p + \frac{K_i}{s} + K_d S$ . La parte derivativa aiuta a compensare le variazioni dell'errore, fornendo una risposta proporzionale alla sua variazione istantanea nel tempo. Questo permette di migliorare la risposta del sistema, riducendo l'errore e le oscillazioni indesiderate. Per il parametro  $K_{pd}$  teniamo il valore trovato al punto precedente; rimangono da determinare  $K_{id}$  e  $K_{dd}$ . Per farlo useremo nuovamente la tecnica basata sulla posizione dei poli della funzione di trasferimento, già usata nel controllo della velocità.

```

m1=1800;
m2=1800;
b=50;
g=9.81;
d=0;
%parametri secondo veicolo
F2=500;
z0=50;
v0=10;
%riferimento
rd=10;
%parametri controllore velocita
Kpv=1500;
Kiv=50;
%sistema controllo velocita
Wvel= (kpv*s+kiv)/(m1*s^2 + (kpv +b)*s +kiv)
%controllo seconda auto
Wfz= (F2*(1/(m2*s+b)) + v0)/s +z0;
%controllore della distanza
Kpd=-3;
Kid=-0.5;
Kdd=-5;
Cd=Kpd+Kid/s+Kdd*s;
pzmap(1/(1-Wvel/s*Cd));
axis([-6 2 -0.2 0.2]);
sgrid(0.6,8);
Wry=Cd*Wvel*(1/s);
%retroazione prima auto
Wrdist=-Wry/(1-Wry);
%retroazione seconda auto

```

```

Wfdist= Wfz/(1-Wry)
figure();
step(Wfdist+rd*Wrdist);
axis([-1

```

Troviamo che  $K_{id} = -0.5$  e  $K_{dd} = -5$ , come vediamo dalla figure 3.5 fanno si che i poli siano entro la zona desiderata per la sovraelongazione. Inoltre, simulando la risposta (figura 3.6) vediamo che l'uscita raggiunge il valore desiderato rimanendo entro le specifiche.

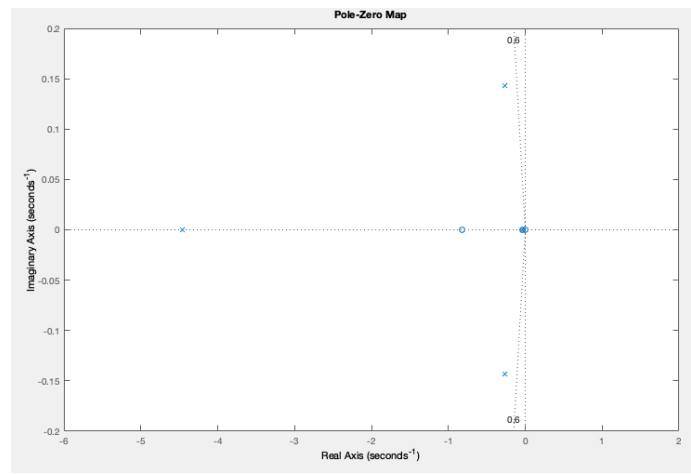


Figura 3.5: poli e zeri del sistema con  $K_{pd} = -3$ ,  $K_{id} = -0.5$ ,  $K_{dd} = -5$

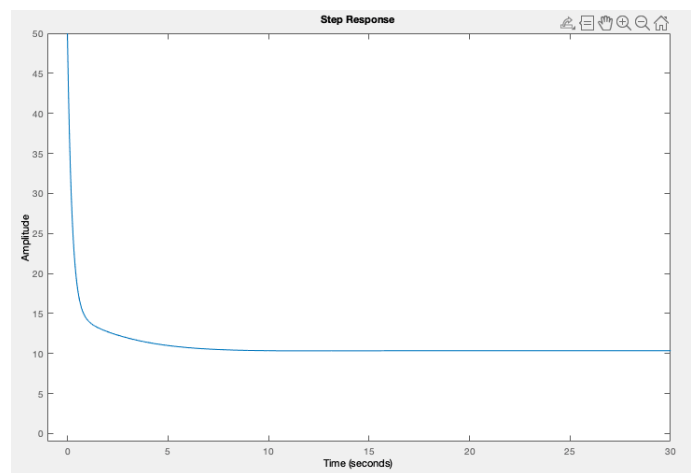


Figura 3.6: risposta del sistema con  $K_{pd} = -3$ ,  $K_{id} = -0.5$ ,  $K_{dd} = -5$



### 3.4 Stabilità del sistema di controllo

Una caratteristica molto importante dei sistemi di controllo è la stabilità. Vogliamo evitare l'insorgere di uscite esponenziali, in quanto, possono compromettere la sicurezza ed il funzionamento del dispositivo progettato. Per determinare la stabilità utilizzeremo il criterio di Routh. Questo metodo ci permetterà di verificare se il polinomio in esame è un polinomio di Hurwitz, cioè stabile. Calcoliamo i denominatori delle funzioni di trasferimento  $W_{r,\delta}$  e  $W_{f_2,\delta}$ . Dopo una serie di passaggi e sostituendo i valori dei parametri otteniamo per entrambe il seguente polinomio:

$$Den(s) = 1800s^4 + 9050s^3 + 4800s^2 + 900s + 25 \quad (3.6)$$

Come possiamo osservare  $Den(s)$  ha tutti i coefficienti non nulli e con lo stesso segno. Soddisfa il criterio di Cartesio, condizione necessaria per la stabilità. Partendo da questo polinomio costruiamo la tabella di Routh, che vediamo in figura 3.7.

4	1800	4800	25
3	9050	900	
2	4621	25	
1	851		
0	25		

Figura 3.7: tabella di Routh di  $Den(s)$

Vediamo che gli elementi della prima colonna sono tutti non nulli e con lo stesso segno. Il polinomio soddisfa il criterio di Routh, possiamo quindi affermare che il sistema è stabile.

### 3.5 Simulazioni

Passiamo ora alla verifica mediante Simulink del funzionamento del sistema appena progettato, utilizzeremo il modello in figura 3.8. Come possiamo vedere, allo schema in catena chiusa, sono stati aggiunti i parametri per il controllo della seconda auto ed i blocchi per gestire il disturbo.

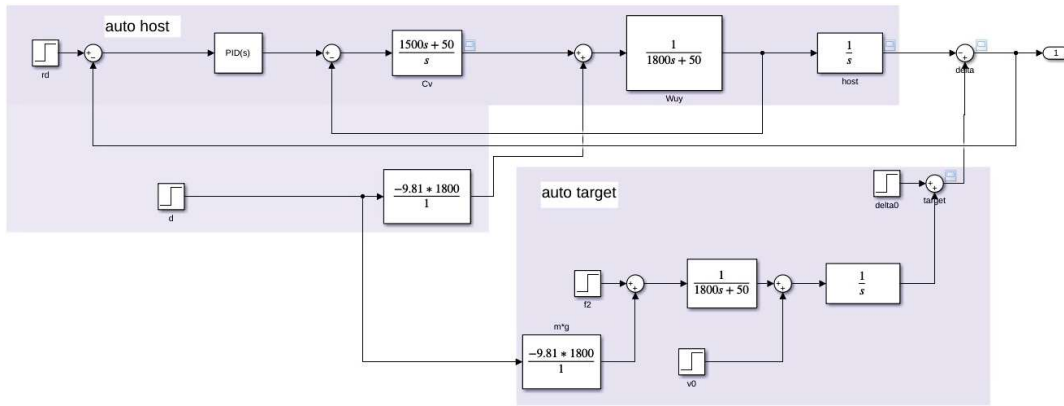


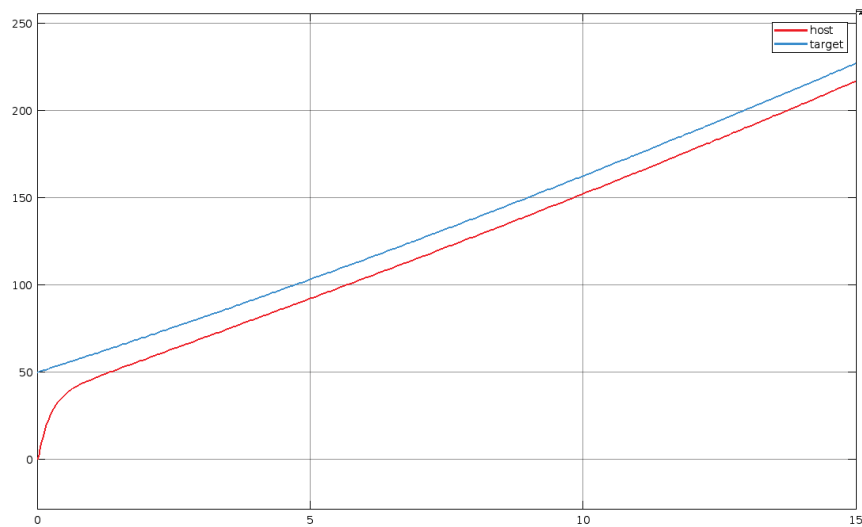
Figura 3.8: modello Simulink

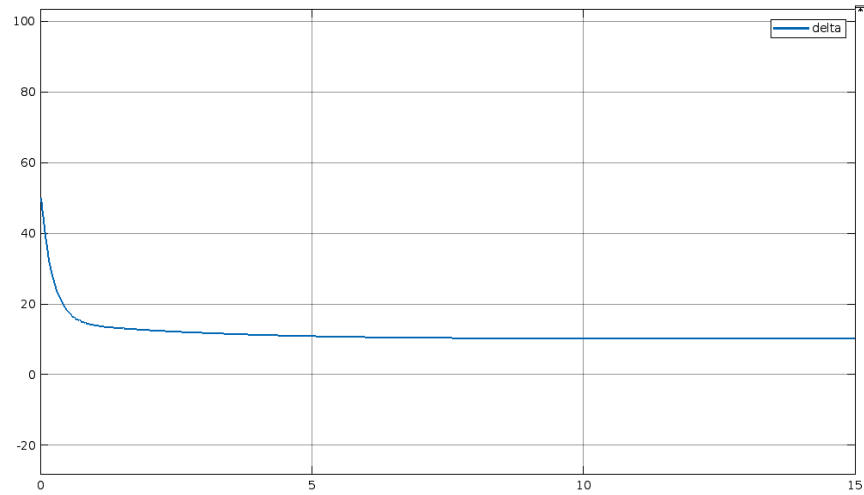
### Riferimento e disturbo

Simuliamo per prima cosa la capacità del sistema di seguire il riferimento, utilizziamo gli stessi parametri usati su MatLab:

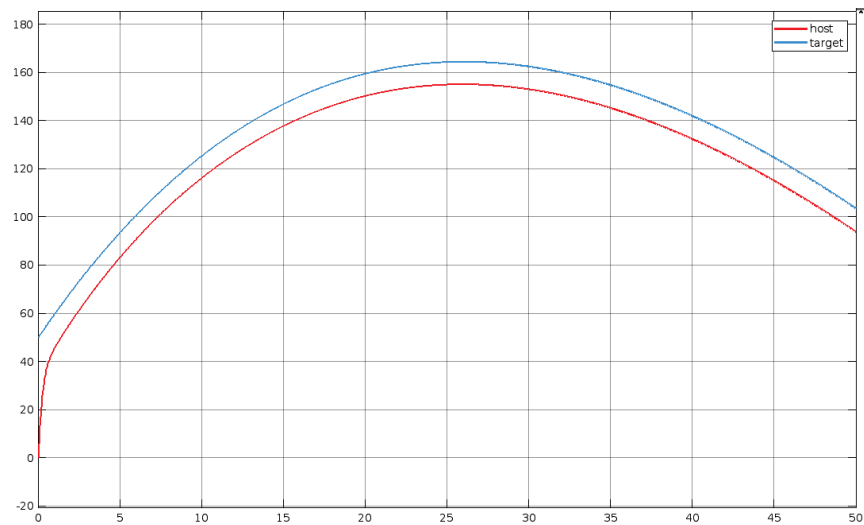
- $f_2 = 500N$
- $r_\delta = 10m$
- $\delta_0 = 50m$
- $v_0 = 10m/s$

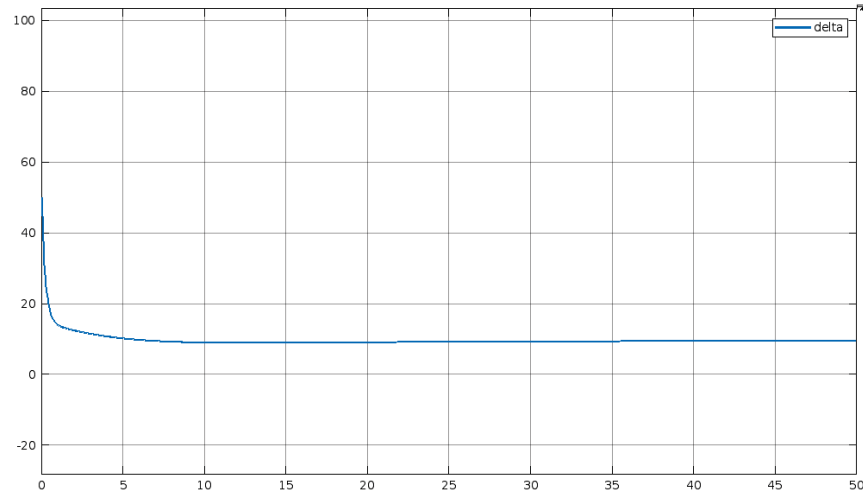
Mostriamo il risultato nelle figure 3.9 e 3.10, che rappresentano rispettivamente la posizione delle due auto e la loro distanza.

Figura 3.9: posizione delle due auto con  $d = 0$

Figura 3.10: distanza tra le auto con  $d = 0$ 

Possiamo notare che l'uscita rimane coerente con i risultati precedenti. Verifichiamo ora la capacità di reiezione del disturbo, aggiungendo alla simulazione precedente un'inclinazione della strada  $d = 0.1rad$ . Come possiamo vedere, dai grafici 3.11 e 3.12, la presenza del disturbo non influisce sulla capacità del sistema nel mantenere l'uscita desiderata. Possiamo notare infatti che, nonostante l'auto target rallenti e addirittura inizi ad indietreggiare, l'host si mantiene comunque alla distanza di  $10m$ .

Figura 3.11: posizione delle due auto con  $d = 0.1$

Figura 3.12: distanza tra le auto con  $d = 0.1$ 

### Saturazione

Come ultima simulazione andiamo a considerare l'effetto della saturazione. Come avveniva nel controllo della velocità, anche in questo caso, al variare degli ingressi potrebbero essere richiesti al nostro sistema valori di  $u(t)$  molto elevati. Andiamo quindi ad inserire nel nostro modello Simulink un saturatore (figura 3.13).

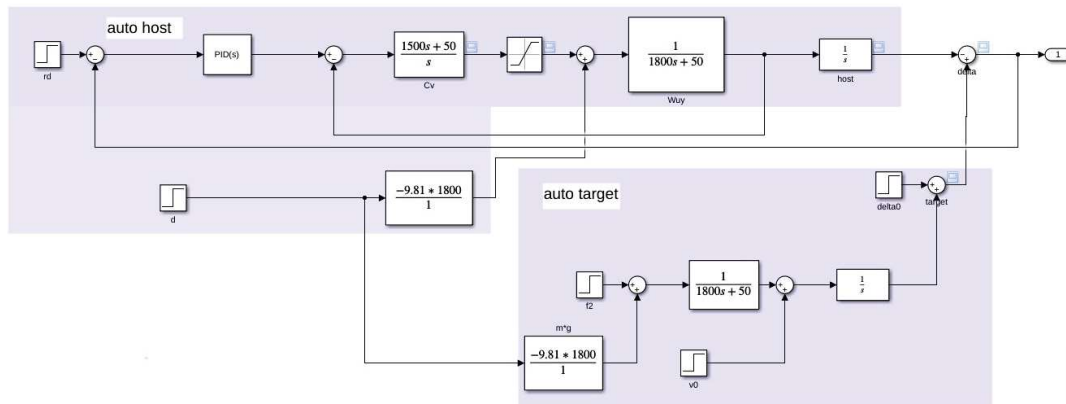
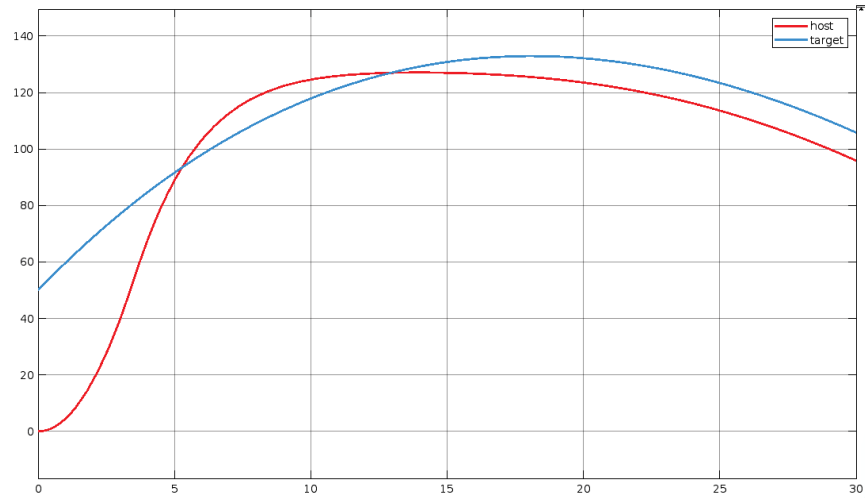
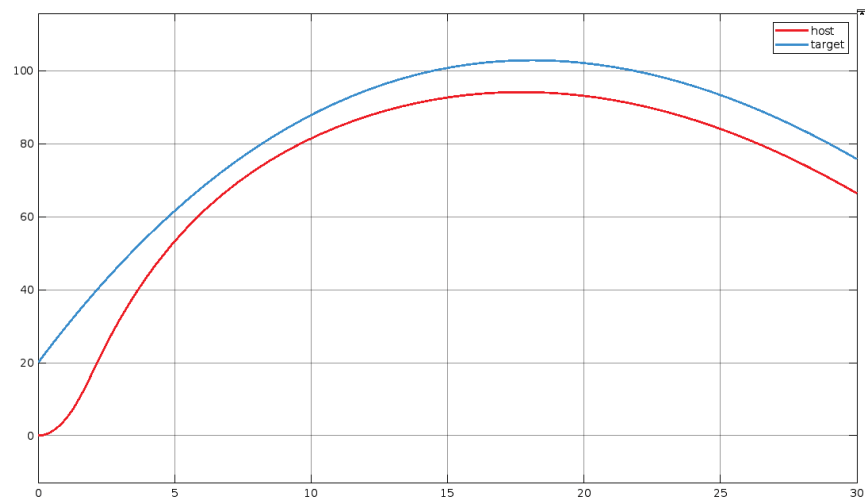


Figura 3.13: modello Simulink con saturatore

L'aggiunta di quest'ultimo porta ad un peggioramento delle prestazioni del sistema, soprattutto riguardante la sovralongazione. Avremo un'uscita entro le specifiche solamente in casi in cui  $\delta_0$  non sarà molto elevata. Questo è osservabile nelle seguenti immagini(3.14-3.17), dove è paragonato il comportamento con valori pari a 20 e 50 m.

Figura 3.14: posizione delle auto con saturazione e  $\delta_0 = 50$ Figura 3.15: posizione delle auto con saturazione e  $\delta_0 = 20$

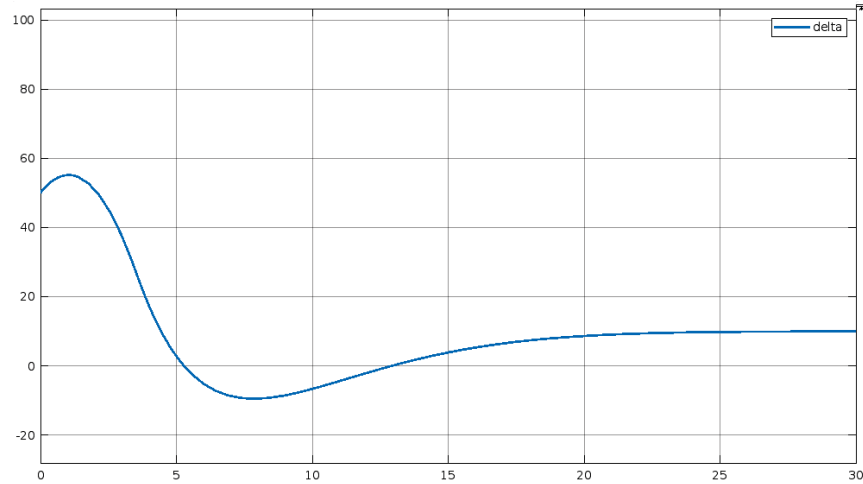


Figura 3.16: distanza tra le due auto con saturazione e  $\delta_0 = 50$

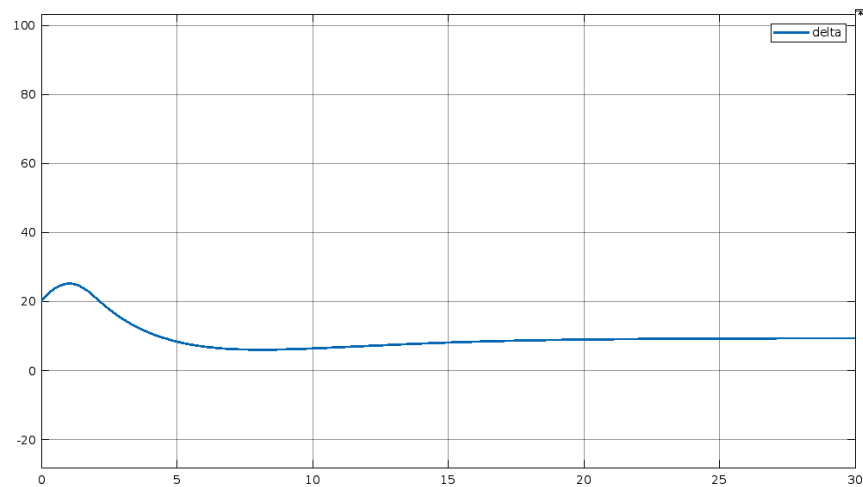


Figura 3.17: distanza tra le due auto con saturazione e  $\delta_0 = 20$

Un miglioramento può essere raggiunto aumentando i valori dei parametri proporzionale e derivativo del controllore. Ponendo ad esempio  $K_{pd} = -30$  e  $K_{dd} = -50$  riusciamo a ridurre significativamente la sovraelongazione che però sarà sempre oltre il valore desiderato del 10%, come possiamo osservare in figura 3.18 e 3.19. Aumentando ulteriormente i coefficienti, le prestazioni tendono a migliorare, a discapito però della qualità di guida che sarà caratterizzata da un susseguirsi di brusche frenate e accelerate.

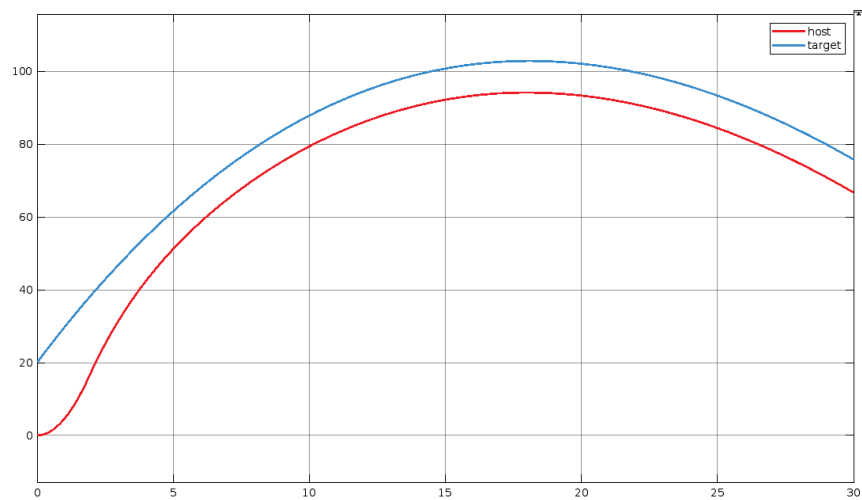


Figura 3.18: posizione delle due auto con i guadagni aggiornati e  $\delta_0 = 50$

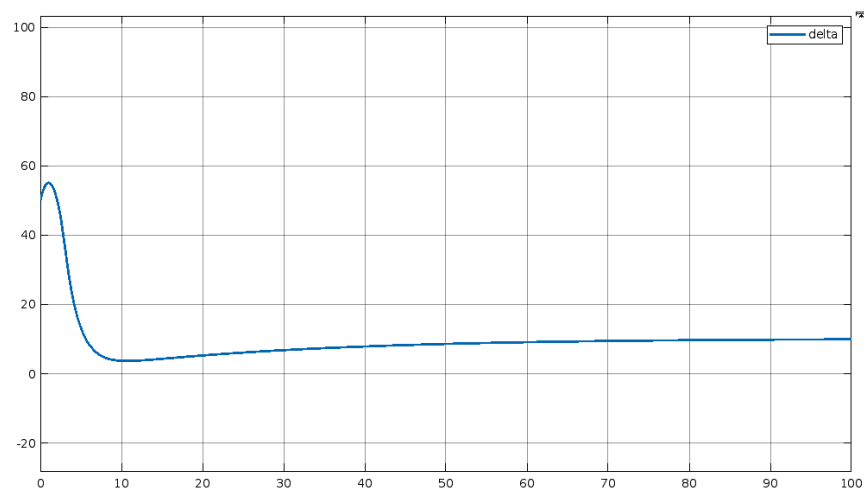


Figura 3.19: distanza tra le due auto con i guadagni aggiornati e  $\delta_0 = 50$





## Capitolo 4

# Conclusioni

Ricapitolando il lavoro effettuato, abbiamo iniziato la realizzazione del controllore per la distanza, sviluppando un sistema di controllo per la velocità. Siamo giunti alla conclusione che utilizzare un controllore PI in catena chiusa fosse la soluzione migliore, il che è stato confermato dalle simulazioni effettuate su Simulink, anche in presenza del saturatore. Abbiamo poi inglobato tale controllore in un sistema più ampio per realizzare un controllore della distanza tra due auto. Questo è stato ottenuto mediante un controllore PID in catena chiusa, dove anche in questo caso le simulazioni hanno dimostrato l'efficacia dell'approccio scelto. L'aggiunta del saturatore ha purtroppo peggiorato sensibilmente le prestazioni del controllore, portando un grande aumento della sovraelongazione. Questo problema può essere parzialmente risolto modificando i guadagni del PID. Per ottenere i risultati desiderati, anche in saturazione, sarà necessario l'utilizzo di tecniche e controllori più avanzati rispetto a quelli visti in questa tesi. Oltre che all'effettivo funzionamento sono molto importanti da considerare altri aspetti come la stabilità e l'effettiva realizzabilità dei controllori. Riguardo la stabilità, essa è stata verificata mediante il criterio di Routh, non dobbiamo quindi preoccuparci dell'insorgere di uscite tendenti a valori infiniti. Per la realizzabilità invece, avendo utilizzato componenti standard come i controllori della famiglia PID, il nostro sistema risulta essere realizzabile ed implementabile.



# Bibliografia

- [1] Marcello Calzavara. Progetto di un sistema di cruise control in ambiente matlab/simulink.
- [2] Duc Lich Luu, Ciprian Lupu, and Thien Van Nguyen. Design and simulation implementation for adaptive cruise control systems of vehicles. In *2019 22nd International Conference on Control Systems and Computer Science (CSCS)*, pages 1–6, 2019.
- [3] Worrawut Pananurak, Somphong Thanok, and Manukid Parnichkun. Adaptive cruise control for an intelligent vehicle. In *2008 IEEE International Conference on Robotics and Biomimetics*, pages 1794–1799, 2009.
- [4] A. Shaout and M.A. Jarrah. Cruise control technology review. *Computers and Electrical Engineering*, 23(4):259–271, 1997.
- [5] Lingyun Xiao and Feng Gao. A comprehensive review of the development of adaptive cruise control systems. *Vehicle System Dynamics*, 48(10):1167–1192, 2010.