

UNIVERSITA' DEGLI STUDI DI PADOVA

Dipartimento di Ingegneria dell'informazione

Corso di Laurea Triennale in Ingegneria Elettronica

Autore: Luigi Costa

Relatore: Alessandro Beghi

Tesi Laurea

**Progetto di un sistema di controllo di velocità  
di un veicolo in ambiente Matlab.**

Anno Accademico 2011 - 2012



## Indice

<b>INTRODUZIONE</b>	pag. 5
<b>CAPITOLO 1</b>	
<b>PRIMA FASE</b>	
<b>1.1 – SPECIFICHE DI PROGETTO E MODELLIZZAZIONE DEL SISTEMA</b>	pag. 7
1.1.1 – Specifiche di progetto	pag. 7
1.1.2 – Modello matematico del sistema	pag. 7
1.1.3 – Funzione di Trasferimento	pag. 8
<b>1.2 – ANALISI DEL SISTEMA IN CATENA APERTA in MATLAB</b>	pag. 8
1.2.1 – Analisi del sistema	pag. 8
<b>CAPITOLO 2</b>	
<b>SECONDA FASE – SOLUZIONE DEL PROBLEMA DEL CONTROLLO</b>	
<b>2.1 – NOTE SULLO SCHEMA A BLOCCHI DEL SISTEMA IN RETROAZIONE UNITARIA</b>	pag. 10
<b>2.2 – CONTROLLORE PID</b>	pag. 11
2.2.1 – Controllore Proporzionale in Matlab	pag. 13
2.2.2 – Controllore Proporzionale Integrato in Matlab	pag. 15
2.2.3 – Controllore Proporzionale Integrato Derivativo in Matlab	pag. 16
<b>2.3 – LUOGO DELLE RADICI</b>	pag. 18
2.3.1 – Introduzione	pag. 18
2.3.2 – Controllore Proporzionale con Luogo delle Radici in Matlab	pag. 19
2.3.3 – Rete Ritardatrice con Luogo delle Radici in Matlab	pag. 21
<b>2.4 STUDIO IN FREQUENZA</b>	pag. 24
2.4.1 – Introduzione	pag. 24
2.4.2 – Diagramma di Bode e risposta ad anello aperto in Matlab	pag. 27
2.4.3 – Controllore Proporzionale	pag. 28
2.4.4 – Rete Ritardatrice Studio in Frequenza ( <i>Bode</i> ) in Matlab	pag. 32
<b>CAPITOLO 3</b>	
<b>TERZA FASE – SIMULINK</b>	
<b>3.1 MODELLAZIONE CRUISE CONTROL SYSTEM IN SIMULINK</b>	pag. 37
3.1.1 – Costruzione Modello	pag. 37
3.1.2 – Implementazione controllore PI e PID con <i>Simulink</i>	pag. 39
3.1.3 – Implementazione Rete Ritardatrice (Luogo delle Radici) con <i>Simulink</i>	pag. 40
3.1.4 – Implementazione Rete Ritardatrice (Studio in frequenza) con <i>Simulink</i>	pag. 42
3.1.5 – Reiezione dei disturbi	pag. 44
<b>CAPITOLO 4</b>	
<b>CONCLUSIONI</b>	pag. 52



## INTRODUZIONE

Il controllo di velocità di un veicolo (*cruise control*), è un dispositivo molto comune nelle automobili al giorno d'oggi. Il suo scopo è di dare la possibilità al conducente di impostare una velocità di crociera attivando un semplice comando, in modo che egli non dovrà più agire fisicamente sul pedale dell'acceleratore per mantenere la vettura alla velocità desiderata.

Lo scopo del mio lavoro sarà di sintetizzare e analizzare un prototipo di *cruise control*, con specifiche di progetto assegnate, mediante l'utilizzo del programma Matlab.

La prima fase del progetto consiste nel derivare il modello matematico del sistema.

La seconda fase sarà di trovare una soluzione al problema del controllo mediante tre metodi diversi:

- Controllore PID
- Luogo delle Radici
- Studio in Frequenza

La terza fase è di valutare la Reiezione dei Disturbi, mediante l'utilizzo dello strumento Simulink.



## CAPITOLO 1 – PRIMA FASE

### 1.1 SPECIFICHE DI PROGETTO E MODELLIZZAZIONE DEL SISTEMA

#### 1.1.1 – Specifiche di progetto

Si vuole realizzare un sistema di controllo che, una volta azionato, porti il veicolo ad una velocità di crociera massima di 10 m/s. Quando si attiva il dispositivo, il motore sviluppa una forza di 500Newton che porta la vettura a raggiungere la velocità desiderata.

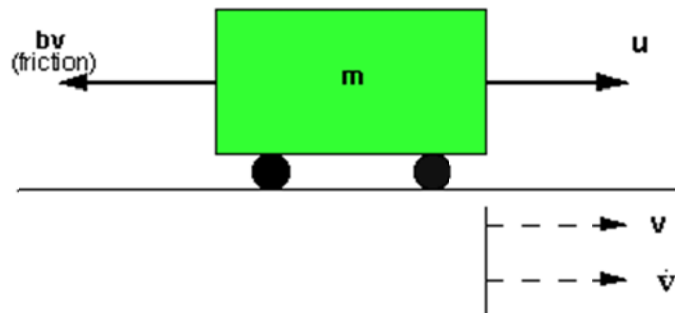
Si considera che l'automobile sia in grado di raggiungere tale velocità in un tempo inferiore ai 5secondi. Essendo un controllo di crociera, una sovralongazione del 10% del valore della velocità e un errore a regime del 2% risultano accettabili.

Riassumendo, le specifiche di progetto da rispettare saranno le seguenti:

- Tempo di salita  $< 5\%$  (*Rise time*)
- Sovraelongazione  $< 10\%$  (*Overshoot*)
- Errore a regime  $< 2\%$  (*Steady State error*)

#### 1.1.2 – Modello matematico del sistema

Trascurando l'inerzia delle ruote, e assumendo che l'attrito (proporzionale alla velocità della vettura) è quello che si oppone al movimento dell'automobile, il problema da analizzare si riduce alla massa semplice e al sistema smorzatore illustrato nella seguente figura.



[fig. 1]

Utilizzando la legge di Newton, si ottiene la seguente equazione che modella il sistema:

$$m\dot{v}(t) + dv(t) = u(t)$$

dove  $u$  è assunta come la forza impressa dal motore.

Inoltre, per tutto il proseguimento dell'analisi, si assumono i seguenti valori:

- $m = 1000\text{Kg}$  massa automobile
- $b = 50\text{Nsec/m}$  coefficiente d'attrito
- $u = 500\text{N}$  forza impressa dal motore

### 1.1.3 – Funzione di Trasferimento

Assumendo le condizioni iniziali nulle, mediante la Trasformata di Laplace, siamo in grado di ricavare la funzione di trasferimento riferita all'equazione che modella il sistema.

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

Essendo la velocità, l'uscita del nostro sistema, assumiamo  $V(s) = Y(s)$

$$msY(s) + bY(s) = U(s)$$

Si ottiene dunque la seguente funzione di trasferimento:

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

## 1.2 ANALISI DEL SISTEMA IN CATENA APERTA in MATLAB

### 1.2.1 – Analisi del sistema

Dopo aver determinato la funzione di trasferimento in catena aperta dal modello matematico del sistema, si procede ora all'analisi tramite *Matlab*.

Di seguito è riportato il codice *Matlab* che realizza la funzione di trasferimento.

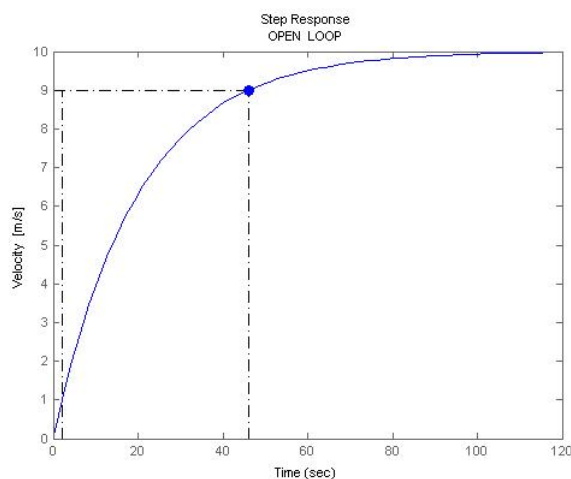
```
m=1000; % Massa del veicolo [kg]
b=50;   % Coefficiente d'attrito [Nsec/m]
u=500;  % Forza impressa al veicolo [N]

num= 1; % Numeratore della funzione
den=[m b]; % Denominatore della funzione
cruise=tf(num,den); % realizza la funzione di trasferimento
```

Analizziamo ora la risposta a gradino del sistema in catena aperta (*Open-Loop*), inserendo nel codice la seguente funzione *Matlab*:

```
step(u*cruise)
```

Eseguendo il codice si ottiene:



[fig. 2]

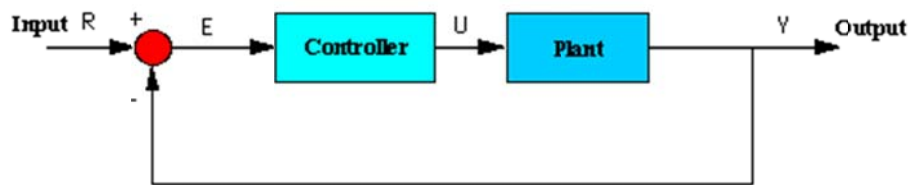


Osservando la risposta a gradino del sistema in catena aperta [fig. 2], si nota che l'auto raggiunge la velocità di 10 m/s a regime dopo un tempo maggiore di 100 secondi, e come marcato in figura, il *Rise time* è abbondantemente superiore rispetto alla specifica di progetto.

### **Conclusioni**

Analizzando la risposta a gradino del sistema in catena aperta, abbiamo appurato che esso così com'è non rispetta le specifiche di progetto. Per risolvere tale problema si ricorre a un controllo in retroazione unitario negativo. Inoltre, si osserva anche che il sistema da controllare è di tipo zero, in altre parole non presenta alcun polo nell'origine e questo comporta un errore a regime non nullo alla risposta a gradino. Osservazione di cui terremo conto più avanti.

Riportiamo di seguito un schema a blocchi di un tipico sistema di retroazione unitaria negativa.



[fig. 3]

La funzione di trasferimento del *Plant* è quella del sistema trovata in precedenza, mentre quella del *Controller* dovrà essere progettata in modo adeguato per soddisfare i criteri di progetto richiesti. Di seguito, nella seconda fase, ci occuperemo di realizzare tale *Controller*, mediante l'utilizzo di tre approcci diversi: PID, Luogo delle radici, Studio in frequenza.

## CAPITOLO 2

### SECONDA FASE – SOLUZIONE DEL PROBLEMA DEL CONTROLLO

#### 2.1 – NOTE SULLO SCHEMA A BLOCCHI DEL SISTEMA IN RETROAZIONE UNITARIA

Prima di procedere al dimensionamento del controllore, è doveroso fare alcune considerazioni sullo schema a blocchi introdotto in precedenza. [fig. 3]

Il *controller* definisce l'azione svolta per portare e mantenere a un valore prefissato un parametro fisico del *plant* che, nel nostro caso, è la velocità. Con  $r(t)$  viene indicato il valore che si vuole far assumere alla variabile controllata e con  $y(t)$  il valore effettivamente assunto da tale grandezza; possiamo così introdurre una funzione d'errore definita come:  $e(t) = r(t) - y(t)$ . Lo scopo dell'azione di controllo è di applicare la migliore scelta possibile della funzione  $u(t)$  detta variabile di controllo, in modo tale da soddisfare *Rise time*, *Overshoot*, *Steady state error*.

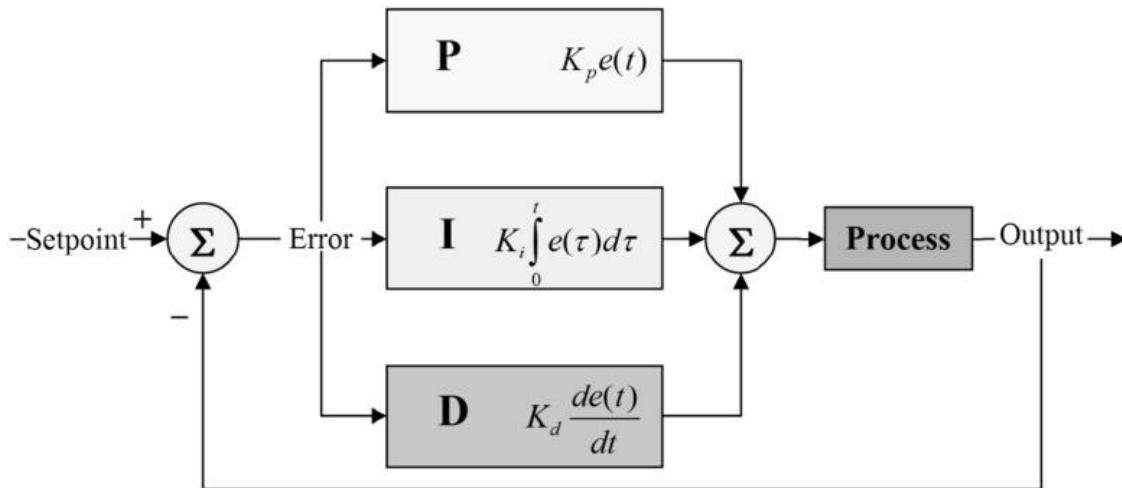
Ragionando ora in termini di Trasformata di Laplace, le principali componenti del sistema sono: il rivelatore di errore  $E(s) = R(s) - Y(s)$ ; il controllore che ha il compito di trasformare il segnale d'errore in un segnale  $U(s)$  che agisce sul processo sottoposto a controllo; un sensore posto all'interno del processo che misura la grandezza fisica da controllare fornendo il segnale  $Y(s)$ . Supponiamo che, sia il controllore, sia il processo, possano essere schematizzati come sistemi lineari e stazionari caratterizzati da una funzione di trasferimento che varrà rispettivamente,  $G(s)_C$  e  $G(s)_P$ . Va detto che lo schema mostrato in [fig.3] è semplificato; non mostra infatti, le sorgenti di rumore interne al sistema e le altre perturbazioni esterne che si sommano ai vari segnali. Una caratteristica essenziale di un buon controllore è di possedere una elevata "robustezza" rispetto alle fluttuazioni del sistema, mantenendo la grandezza controllata al valore desiderato anche alla presenza di eventuali piccole variazioni della funzione di trasferimento che caratterizza il processo. Utilizzando la ben nota relazione dei sistemi retroazionati, scriviamo la funzione di trasferimento del sistema ad anello chiuso:

$$T(s) = \frac{Y(s)}{U(s)} = \frac{G_C(s) * G_P(s)}{1 + G_C(s) * G_P(s)}$$

Il problema generale del controllo si riduce quindi a determinare, per una certa funzione di trasferimento del processo  $G(s)_P$ , la migliore funzione di trasferimento del controllore  $G(s)_C$  che ottimizza la  $T(s)$ .

## 2.2 – CONTROLLORE PID

Prima di procedere al dimensionamento del controllore PID, ne vediamo una rapida descrizione degli aspetti salienti. Per iniziare, di seguito è riportato lo schema a blocchi parallelo, inserito nella catena di retroazione al posto del *Controller*.



[fig. 4]

Il termine PID è l'acronimo che indica l'utilizzo combinato di tre funzioni di controllo di tipo Proporzionale, Integrale e Differenziale che danno origine al segnale di controllo  $u(t)$  seguente:

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt}$$

Ragionando in termini di Trasformata di Laplace si ottiene la seguente funzione:

$$U(s) = \left( K_p + K_i \frac{1}{s} + K_d s \right) E(s)$$

Dove  $K_p$  è il guadagno del controllore proporzionale,  $K_i$  quello dell' integratore e  $K_d$  del derivatore.

Un modo alternativo di scrivere l'equazione  $u(t)$  detto PID Standard è il seguente:

$$u(t) = K_p \left[ \frac{1}{T_i} \int e(t) dt + T_d \frac{de(t)}{dt} \right]$$

$K_p$  Guadagno Proporzionale

$T_i$  Costante di tempo dell'azione integrale (o di reset)  $T_i = K_p / K_i$

$T_d$  Costante di tempo dell'azione derivativa  $T_d = K_d / K_p$

### Significato delle tre azioni di controllo

- **azione proporzionale**
  - maggiore è l'errore, maggiore è l'azione di controllo
- **azione integrale**
  - errore nullo a segnali di riferimento o disturbi costanti
- **azione derivativa**
  - azione di controllo "preventiva"
  - anticipo di fase
  - **NOTA:**  
La realizzazione fisica dell'azione Derivativa è diversa dalla realizzazione teorica essendo la funzione di trasferimento impropria, costituita da uno zero nell'origine:

$$T_d s \rightarrow \frac{T_d s}{1 + \frac{T_d}{N} s}$$

Simile a una rete anticipatrice  
 $N = 5 \div 20$  per collocare polo all'esterno della banda d'interesse.

### OSSERVAZIONE:

La funzione di trasferimento complessiva del PID ottenuta in precedenza:

$$\frac{K_D s^2 + K_P s + K_I}{s}$$

risulta essere anch'essa impropria, quindi tale forma prende il nome di *PID Ideale*.  
Per realizzare fisicamente il PID, alla funzione di trasferimento ideale sarà quindi aggiunto, come nel caso del solo controllore Derivativo, un polo in alta frequenza.

### 2.2.1 – Controllore Proporzionale in Matlab

Come primo approccio al problema del controllo in retroazione, proviamo a vedere come si comporta il sistema in catena chiusa con solamente un controllore P.

La funzione di trasferimento è la seguente:

$$\frac{Y(s)}{R(s)} = \frac{K_p}{ms + (b + K_p)}$$

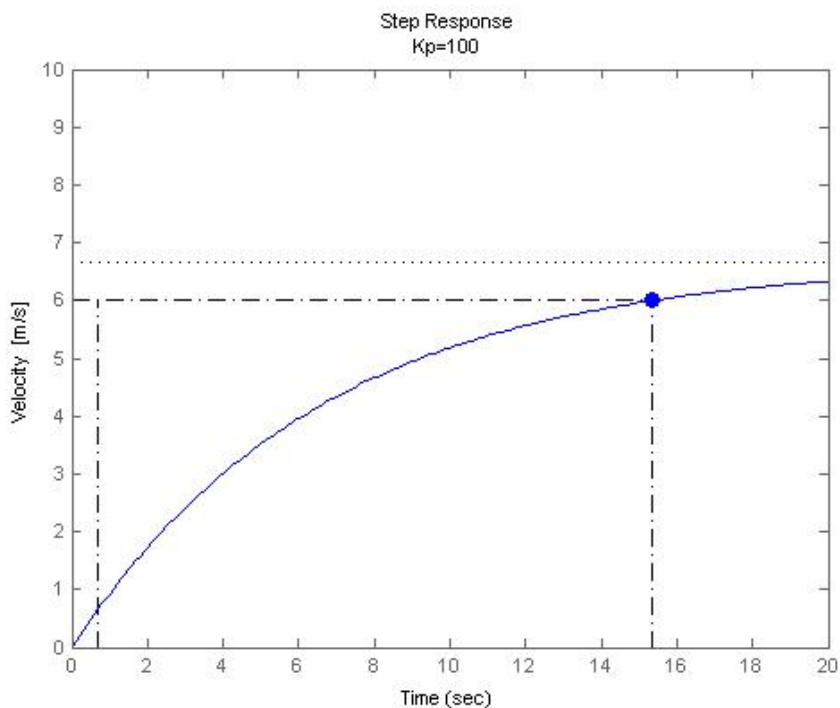
Di seguito è riportato il codice Matlab che realizza il sistema:

```
Kp = 100; % Guadagno Controllore Proporzionale

m = 1000; % Massa del veicolo [kg]
b = 50;   % Coefficiente d'attrito [Nsec/m]
r = 10;   % Velocita' di riferimento 10 [m/s]

num = [1]; % Numeratore
den = [m b]; % Denominatore
cruise = tf(num,den); % Funzione di Trasferimento Gp(s)
sys_cl=feedback(Kp*cruise,1); % Funzione W(s) catena chiusa
t=0:0.1:20;
step(r*sys_cl,t) % Risposta a gradino
axis([0 20 0 10])
```

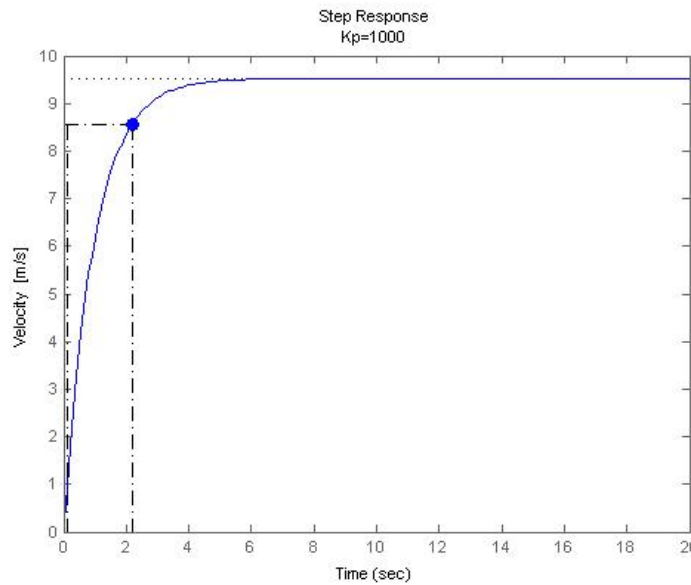
Eseguito il codice si ottiene:



[fig. 5]

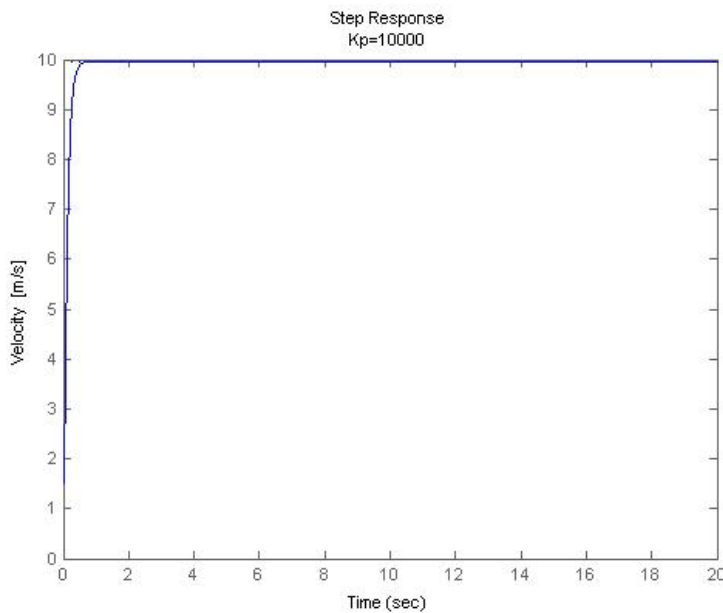
Come si può osservare dal grafico della risposta a gradino [fig. 5], né il *Rise time*, né lo *Steady State error* rispettano le specifiche di progetto; mentre, non compare il fenomeno di *Overshoot* poiché la funzione del sistema risulta del primo ordine.

Provando ad aumentare il guadagno proporzionale  $K_p=1000$  si ottiene il seguente grafico:



[fig. 6]

Come si può notare in [fig. 6], ora il *Rise time* rispetta le specifiche di progetto, mentre lo *Steady State error* non è ancora rispettato. Proviamo quindi con un guadagno  $K_p=10000$  e si ottiene il seguente grafico:



[fig. 7]

Osservando il grafico [fig. 7], si osserva che ora lo *Steady State error* è rispettare ampiamente la specifica di progetto. Si nota, però, che il *Rise time* è inferiore a 0.5 secondi, il che significherebbe che il *cruise control system* dovrebbe essere in grado di portare la vettura da 0 a 10 m/s in un tempo inferiore a 0.5 secondi il che è del tutto irrealistico.

### Conclusioni

Possiamo concludere, quindi, tenendo presente che il sistema da controllare è di tipo 0, che utilizzando solamente un controllore Proporzionale non riusciamo a risolvere il problema del controllo. La specifica di progetto del *Rise time*, è rispettata, mentre quella dello *Steady State error*, a meno di imporre un valore molto elevato del guadagno  $K_p$  (che renderebbe il sistema irrealizzabile dal punto di vista fisico), non è mai rispettata.

### 2.2.2 – Controllore Proporzionale Integrale in Matlab

Sapendo che l'azione di un controllore Integrale, ha la caratteristica di rendere nullo l'errore a regime alla risposta a gradino, proviamo a risolvere il problema utilizzando un controllore PI che è la combinazione di un' azione proporzionale ed una integrativa.

La funzione di trasferimento del sistema in catena chiusa con il controllore PI sarà la seguente:

$$\frac{Y(s)}{U(s)} = \frac{K_p s + K_i}{ms^2 + (b + K_p) + K_i}$$

Di seguito è riportato il codice Matlab che realizza il sistema:

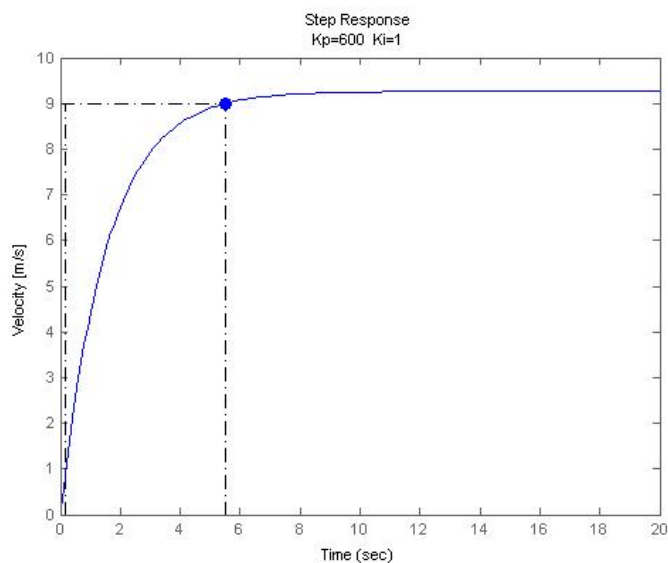
```
Kp = 600; % Guadagno Controllore Proporzionale
Ki = 1; % Guadagno Controllore Integratore

m = 1000; % Massa del veicolo [kg]
b = 50; % Coefficiente d'attrito [Nsec/m]
r = 10; % Velocità di riferimento [m/s]

num = [1]; % Numeratore Gp(s)
den = [m b]; % Denominatore Gc(s)
cruise = tf(num,den); % Gp(s) Dispositivo
contr=tf([Kp Ki],[1 0]); % Gc(s) Controllore PI

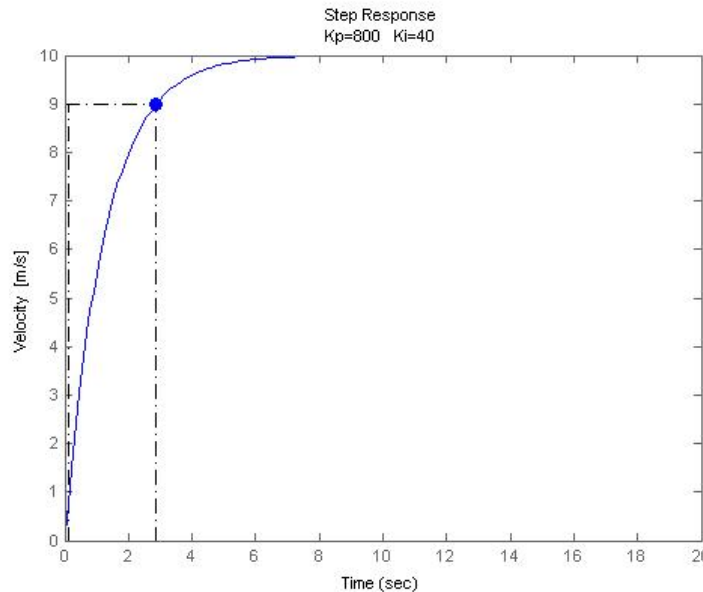
sys_cl=feedback(contr*cruise,1); % Y(s)/R(s)
t=0:0.1:20; % Intervallo campionamento asse del tempo
step(r*sys_cl,t) % Risposta a Gradino
axis([0 20 0 10]) % Intervallo valori degli assi
```

Considerando che un valore elevato del guadagno  $K_i$  potrebbe destabilizzare la risposta, meglio procedere per gradi e inizialmente impostiamo  $K_i=1$  e vediamo che si ottiene la seguente risposta:



[fig. 8]

Le specifiche di progetto non sono soddisfatte, pertanto andiamo ad aumentare entrambi i guadagni  $K_p$  e  $K_i$  ed osserviamo cosa accade.



[fig. 9]

Osservando il grafico [fig. 9], si può notare come ora tutte le specifiche di progetto (*Rise time*, *Overshoot*, *Steady State error*) siano rispettate.

### Conclusioni

Da questa analisi si può concludere che il problema del controllo risulta essere già ampiamente risolto mediante l'utilizzo di un controllore PI. L'impiego di una azione derivatrice, pertanto, risulta essere non necessaria.

Per completezza di studio, di seguito andremo a realizzare e analizzare la soluzione del problema del controllo, mediante l'utilizzo di un PID.

### 2.2.3 Controllore Proporzionale Integrativo Derivativo in Matlab

La funzione di trasferimento del sistema in catena chiusa con il controllore PID sarà la seguente:

$$\frac{Y(s)}{R(s)} = \frac{K_d s^2 + K_p s + K_i}{(m + K_d) s^2 + (b + K_p) s + K_i}$$

Di seguito è riportato il codice Matlab che realizza il sistema:

```
Kp = 800; % Guadagno Controllore Proporzionale
Ki = 40; % Guadagno Controllore Integratore
Kd = 40; % Guadagno Controllore Derivatore

m = 1000; % Massa del veicolo [kg]
b = 50; % Coefficiente d'attrito [Nsec/m]
r = 10; % Velocità di riferimento [m/s]

num = [1]; % Numeratore di Gp(s)
den = [m b]; % Denominatore di Gp(s)
cruise = tf(num,den); % Funzione Trasferimento Gp(s)

contr=tf([Kd Kp Ki],[1 0]); % Funzione Trasferimento DIP
```



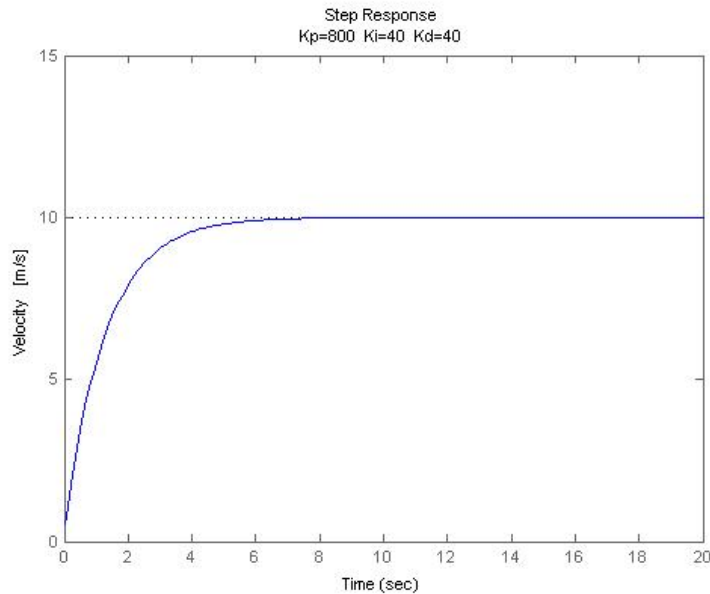
```

%ANELLO DI RETROAZIONE
sys_cl=feedback(contr*cruise,1);

t=0:0.1:20;      % asse del tempo con passo 0.1
step(r*sys_cl,t) % Risposta a gradino
axis([0 20 0 15])

```

Eseguito il codice si ottiene:



[fig. 10]

Si può notare dal grafico [fig. 10] come tutte le specifiche di progetto siano rispettate.

Per la scelta dei guadagni  $K_p$ ,  $K_i$  e  $K_d$  si è adottata una tecnica per tentativi aumentando in maniera adeguata i vari parametri; tenendo bene a mente che un valore elevato del guadagno  $K_i$  potrebbe destabilizzare la risposta. Inoltre, si osserva che non è stato possibile adottare il metodo di taratura *Zigler-Nichols* in catena chiusa, perché nel primo passo, quando si pongono a zero  $K_i$  e  $K_d$  e si fa variare il parametro  $K_p$  fino a trovare il valore critico dove avvengono fenomeni oscillatori, essi non compaiono, perché il sistema risulta del primo ordine.

### Conclusioni generali soluzione del controllo con PID

In conclusione, riassumendo, abbiamo appurato che il problema del *Cruise Control*, mediante l'utilizzo d'un solo Controllore Proporzionale non è possibile, poiché l'errore a regime non si riesce a ridurre in modo accettabile, a meno di scegliere un valore di  $K_p$  esageratamente elevato, tale da rendere irrealizzabile dal punto di vista fisico la sua realizzazione. Utilizzando, invece, un controllore Proporzionale Integrato il problema del *Cruise Control* risulta essere risolto già in modo soddisfacente, tanto da non rendere necessario l'utilizzo di un controllore PID. Per completezza di studio, abbiamo analizzato anche la soluzione mediante l'utilizzo di un controllore PID. Si osserva che nell'analisi con codice *Matlab*, è stato utilizzato per semplicità il modello Ideale del controllore PID. I risultati, com'era prevedibile, sono molto soddisfacenti mediante l'utilizzo di un controllore PID.

## 2.3 – LUOGO DELLE RADICI

### 2.3.1 Introduzione

Dopo aver visto un approccio alla soluzione del problema del controllo mediante controllore PID, vediamo ora una sintesi del controllo utilizzando il metodo del Luogo delle Radici.

Consideriamo un sistema con funzione di trasferimento (razionale propria)

$$G(s) = K \frac{p(s)}{q(s)}$$

con  $K$  parametro reale, mentre  $p(s)$  e  $q(s)$  polinomi a coefficienti reali tra loro coprimi.

Si vuole studiare, al variare del parametro  $K$ , la stabilità BIBO del sistema ottenuto per retroazione unitaria negativa del sistema dato, la cui funzione di trasferimento risulta:

$$W(s) = \frac{Kp(s)}{q(s) + Kp(s)}$$

Osserviamo che non vi è perdita di generalità nel supporre che  $p(s)$  e  $q(s)$  siano polinomi monici e quindi la rappresentazione adottata per  $G(s)$  sia quella di Evans così che  $K=K_E$ .

La coprimalità di  $p(s)$  e  $q(s)$  assicura per ogni valore di  $K \neq 0$ , anche la coprimalità di  $Kp(s)$  e  $q(s) + Kp(s)$  e, pertanto, la funzione di trasferimento  $W(s)$  sia espressa mediante una rappresentazione copriva. L'obiettivo è di valutare la posizione dei poli di  $W(s)$  nel piano complesso al variare del parametro reale  $K \neq 0$ .

Di conseguenza, per  $K \neq 0$  i poli di  $W(s)$  coincidono con gli zeri del polinomio:

$$d(s) := q(s) + Kp(s)$$

Il *Luogo delle Radici* è quindi l'insieme dei punti  $s$  appartenenti al piano complesso, che soddisfano l'equazione  $d(s) = q(s) + Kp(s) = 0$  al variare di  $K$  in  $\mathbb{R} \setminus \{0\}$ . Più precisamente, i punti che soddisfano tale equazione per valori di  $K > 0$  formano il cosiddetto *luogo positivo* delle radici, mentre quelli che la soddisfano per valori negativi del parametro  $K$  rappresentano il *luogo negativo*. L'unione dei due costituisce il *luogo completo*.

Nella nostra sintesi, restringeremo l'attenzione al luogo positivo.

Il grafico del luogo delle radici contiene tanti rami quanti sono i poli di  $W(s)$  contati con le loro molteplicità.

Al variare del parametro  $K$  vi sono dei valori critici, per i quali avvengono delle transizioni nel comportamento del sistema (ad esempio si ha stabilità BIBO per  $K_{cr,1} < K < K_{cr,2}$  e instabilità per  $K < K_{cr,1}$  e per  $K > K_{cr,2}$ ); tali valori critici, si hanno in corrispondenza dell'attraversamento da parte dei rami del luogo delle radici dell'asse immaginario (situazione in cui uno o più poli passano dalla regione di stabilità, o viceversa).

### 2.3.2 Controllore Proporzionale con Luogo delle Radici in Matlab

Ricordiamo che la funzione di trasferimento  $G_p(s)$  del nostro sistema è la seguente:

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

Le specifiche di progetto del *Rise time* ( $t_r$ ) e dell' *Overshoot* ( $M_p$ ) impongono dei vincoli ben precisi sulla pulsazione naturale  $\omega_n$  e sul coefficiente di smorzamento  $\xi$  in base alle seguenti relazioni:

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

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

Essendo la specifica di progetto  $T_r < 5\%$ , si ottiene  $\omega_n = 0.36$ .

$M_p$ , (differenza tra valore aggiunto e valore finale, espresso in %), da specifiche di progetto dev'essere minore del 10% ovvero, massimo valore aggiunto 1m/s, impone che  $\xi = 0.6$ .

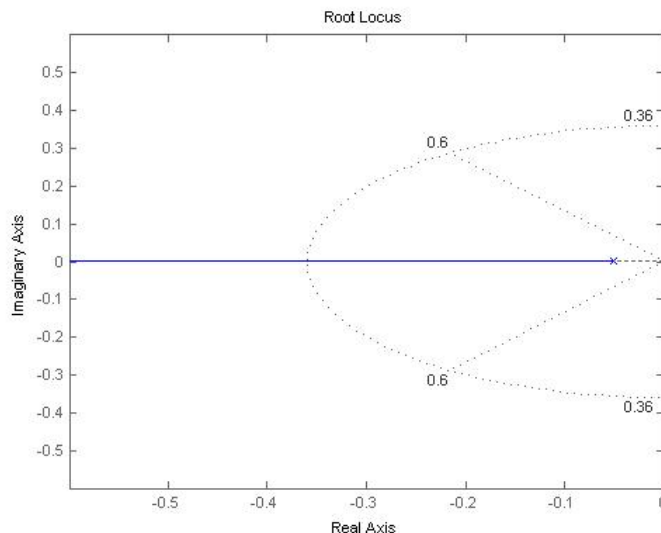
Per prima cosa si traccia il *Luogo delle Radici* di  $G_p(s)$ , e di seguito si riporta il relativo codice *Matlab*:

```
m = 1000; % Massa del veicolo [kg]
b = 50;   % Coefficiente d'attrito [Nsec/m]
r = 10;   % Velocità di riferimento 10 [m/s]

num = [1]; % Numeratore Gp(s)
den = [m b]; % Denominatore Gp(s)
cruise = tf(num,den); % Funzione di Trasferimento Gp(s)

rlocus(cruise) % Traccia il luogo delle radici Gp(s)
axis([-0.6 0 -0.6 0.6]);
sgrid(0.6, 0.36) % Traccia nel grafico i vincoli di progetto
```

eseguendo il codice si ottiene il seguente *Luogo delle Radici*:



[fig. 11]

Osservando il grafico [fig. 11] si nota subito che il Luogo delle Radici, essendo  $G_p(s)$  una funzione del primo ordine, è costituito da un solo polo, che è posizionato nel semipiano sinistro del piano complesso. Possiamo quindi, in primo luogo, già affermare che per qualsiasi valore di  $K > 0$  il sistema sarà sempre comunque BIBO stabile.

Nel Luogo delle Radici [fig. 11], grazie all'utilizzo della funzione *Matlab* `sgrid(0.6, 0.36)`, sono stati marcati i limiti di confine per la pulsazione naturale  $\omega_n = 0.36$  e il coefficiente di smorzamento  $\xi = 0.6$ .

In un secondo momento è stata aggiunta la seguente parte di codice per completare la sintesi del controllore:

```
[Kp, poles]=rlocfind(cruise) % Scelta del Kp e relativo polo

%_____ SISTEMA RETRAZIONATO_____%
sys_cl=feedback(Kp*cruise,1);

t=0:0.1:20;
step(r*sys_cl,t) % Risposta a gradino
axis([0 20 0 10])
```

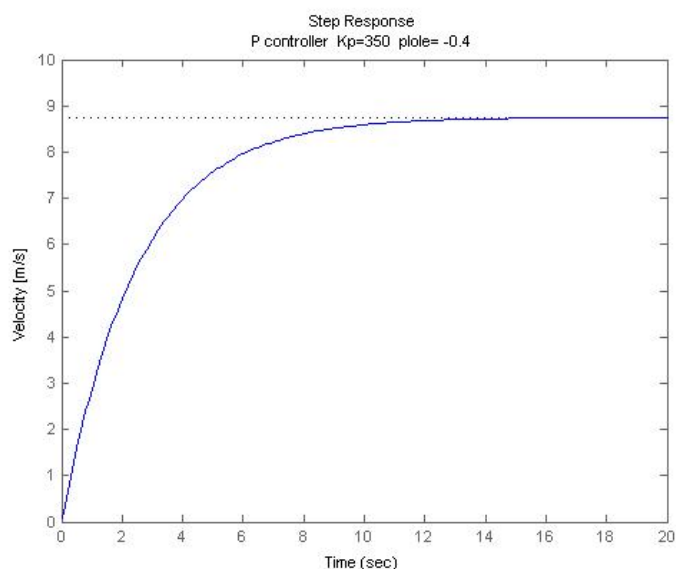
La funzione `[Kp, poles]=rlocfind(cruise)`, una volta che il programma ha tracciato il Luogo delle Radici di  $G_p(s)$ , permette di posizionarsi tramite un cursore su di esso, ed una volta scelto il punto del piano, verrà restituito il corrispettivo valore di  $K_p$  e del *polo* ad esso corrispondente.

Affinché tutte le specifiche di progetto siano rispettate, bisognerà scegliere un valore della pulsazione di risonanza  $\omega_n > 0.36$ , ovvero all'esterno della semi-ellisse tratteggiata nel Luogo delle Radici. Invece, per quanto riguarda il coefficiente di smorzamento, per fare in modo che sia  $\xi > 0.6$  bisognerà restare all'interno dell'angolo formato dalle due rette tratteggiate.

Il valore  $K_p$  trovato, viene poi in seguito utilizzato nel codice *Matlab*, dalla funzione `sys_cl=feedback(Kp*cruise,1)`, che realizza il sistema in catena chiusa con il Controllore P, di cui ne riportiamo la funzione di trasferimento.

$$\frac{Y(s)}{R(s)} = \frac{K_p}{ms + (b + K_p)}$$

Per finire è poi visualizzata la risposta a gradino di cui riportiamo un esempio per  $K_p = 350$  e rispettivo polo in  $-0.4$



[fig. 11]

### Conclusioni

Dalla risposta a gradino [fig. 11] si può notare come le specifiche di progetto del *Rise time* e dell'*Overshoot* sono ampiamente rispettate, mentre quella dello *Steady State error* risulta molto inferiore al 10%. Possiamo concludere, come abbiamo già visto in precedenza, che solamente con l'utilizzo di un Controllore Proporzionale, il problema del *cruise control system* non risulta essere soddisfatto.

### 2.3.3 Rete Ritardatrice con Luogo delle Radici in Matlab

Per ridurre lo *Steady State error* proviamo ora ad aggiungere al sistema una Rete Ritardatrice (*Lag controller / phase-lag compensator*). Utilizzando il metodo del *Luogo delle Radici* la forma della Rete Ritardatrice, viene scritta nel seguente modo:

$$G_L(s) = \frac{s + Z_o}{s + P_o}$$

Dove la grandezza dello zero  $Z_o$  è maggiore di quella del polo  $P_o$ .

Una *Lag* tende a spostare l'intero Luogo delle Radici nel piano complesso verso destra, il che non è desiderato. Per tale motivo, il polo e lo zero devono essere accostati (solitamente vicino all'origine) in modo da non modificare sensibilmente la risposta transitoria o caratteristiche di stabilità del sistema. Se da una parte una *Lag* non migliora la risposta transitoria a causa del suo effetto negativo, essa può migliorare la risposta a regime dell'intero sistema. Ciò avviene nel seguente modo; alle alte frequenze, il controllore *Lag* avrà un guadagno unitario, mentre, alle basse frequenze, il guadagno sarà dato dal rapporto  $Z_o/P_o$ , che è maggiore di 1. Tale rapporto moltiplicherà la costante  $K_p$ , e l'errore a regime sarà diminuito di un fattore pari a tale rapporto  $Z_o/P_o$ .

Per tracciare il Luogo delle Radici e farne un'analisi utilizzeremo le tre funzioni *Matlab* (`rlocus`, `sgrid`, `rlocfind`) utilizzate in precedenza per il controllore P.

Di seguito è riportato il codice *Matlab* che realizza il sistema con la Lag:

```
m = 1000; % Massa del veicolo [kg]
b = 50;   % Coefficiente d'attrito [Nsec/m]
r = 10;   % Velocità di riferimento [m/s]

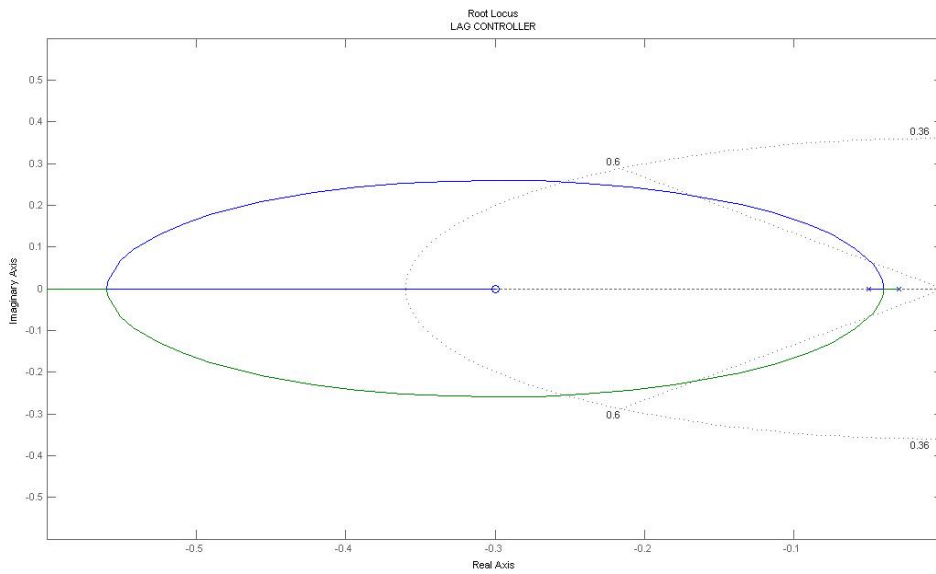
num = [1]; % Numeratore Gp(s)
den = [m b]; % Denominatore Gp(s)
cruise = tf(num,den); % Funzione di trasferimento Gp(s)

%__RETE RITARDATRICE__%
Zo=0.3;
Po=0.03;
contr=tf([1 Zo],[1 Po]); % Funzione trasferimento Lag

%_____ ROOT LOCUS_____%
rlocus(contr*cruise); % Traccia il luogo delle radici
axis([-0.6 0 -0.6 0.6]);
sgrid(0.6, 0.36) % Traccia nel grafico i vicoli di progetto
[Kp, poles]=rlocfind(contr*cruise) Scelta del Kp e relativi poli

%_____ SISTEMA RETRAZIONATO_____%
sys_cl=feedback(Kp*contr*cruise,1);
t=0:0.1:20;
step(r*sys_cl,t) % Risposta a gradino
axis([0 20 0 12])
```

Il Luogo delle Radici che si ottiene è il seguente:



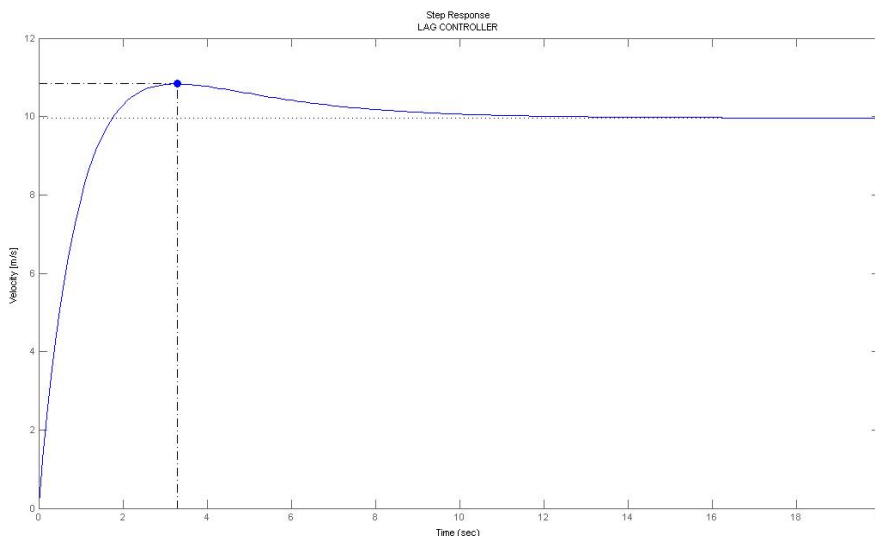
[fig. 12]

Come si può osservare dal grafico [fig. 12], all'aumentare del valore di  $K$ , i due poli si avvicinano al loro baricentro e vi collasano uscendo così con due rami dall'asse reale, uno dei essi va nello zero (ramo in blu) mentre l'altro tende a  $\infty$  (ramo in verde). Anche con l'introduzione della Rete *Lag*, tutti i rami non intersecano mai l'asse immaginario del piano complesso, quindi, per qualsiasi valore di  $K > 0$ , il sistema sarà sempre stabile. Nel Luogo delle Radici, sono stati tracciati mediante la funzione `sgrid` i vincoli di progetto sulla pulsazione  $\omega_n = 0.36$  e coefficiente di smorzamento  $\xi = 0.6$ .

La funzione di trasferimento ad anello-chiuso che si ottiene risulta la seguente:

$$\frac{Y(s)}{U(s)} = \frac{K_p s + K_p P_o}{ms^2 + (b + P_o + K_p)s + (bP_o + K_p Z_o)}$$

Con i valori  $Z_o = 0.3$  e  $P_o = 0.03$  scelti, e posizionandosi nel Luogo delle Radici intorno al punto  $-0.4$  scelto in precedenza anche per il controllore P, si ottiene la seguente risposta a gradino:



[fig. 13]

### ***Conclusioni***

Con un primo sguardo al grafico [fig. 12], in base alle considerazioni fatte sul rapporto  $Z_o/P_o$  l'errore a regime con i valori imposti viene attenuato di un fattore 10, tale fatto risulta ben visibile nel grafico. Si può notare che lo zero introdotto alla Rete *Lag* ha fatto comparire un fenomeno di *Overshoot* contenuto che rispetta la sua specifica di progetto.

Si può concludere, quindi, che tutte le specifiche di progetto (*Rise time*, *Overshoot*, *Steady State error*) sono rispettate, ovvero, mediante l'utilizzo di una Rete *Lag* il problema del controllo risulta soddisfatto pienamente.

## 2.4 STUDIO IN FREQUENZA

### 2.4.1 Introduzione

La *Risposta in Frequenza* è una rappresentazione della risposta del sistema ad ingressi sinusoidali a frequenze variabili.

- In regime permanente:

$$\begin{aligned}u(t) &= A \sin(\omega_0 t + \varphi_0) \\y(t) &= |G(j\omega_0)| A \sin(\omega_0 t + \varphi_0 + \Phi(\omega_0))\end{aligned}$$

- $y(t)$  a regime è ancora una sinusoide con:
  - stessa frequenza  $\omega_0$
  - ampiezza amplificata/attenuata di un fattore  $|G(j\omega_0)|$
  - fase aumentata/diminuita di un fattore  $\Phi(\omega_0)$
  - in sostanza la *Risposta in Frequenza* è definita come la differenza di ampiezza e di fase tra ingresso e uscita di sinusoidi.

- $G(j\omega)$ : funzione di *RISPOSTA ARMONICA* o *RISPOSTA IN FREQUENZA*

- Si ottiene valutando la funzione di trasferimento in  $s=j\omega$  (muovendosi lungo l'asse immaginario), con  $\omega \geq 0$

- Legame con la risposta impulsiva  $g(t)$ :

$$g(t) \xrightarrow{\quad L \quad} G(s) \xrightarrow{\quad s=j\omega \quad} G(j\omega)$$

- Quindi  $G(j\omega)$  è determinabile univocamente da  $G(s)$  e viceversa

- Forniscono caratterizzazioni equivalenti del sistema

- La risposta armonica può essere calcolata sperimentalmente

- applico  $u(t) = A \sin(\omega_0 t + \varphi)$
- lascio esaurirsi il transitorio
- misuro  $y(t) = A' \sin(\omega_0 t + \varphi')$
- assumo  $|G(j\omega_0)| = A'/A$  e  $\Phi(\omega_0) = \varphi' - \varphi$
- ripeto per altre frequenze
- costruisco per punti  $G(j\omega)$

La *Risposta in Frequenza* di un sistema può essere visualizzata in due modi diversi: attraverso il diagramma di Bode o tramite il diagramma di Nyquist.

$$G(j\omega) = |G(j\omega)| e^{j \arg G(j\omega)}$$

### **Diagrammi di Bode**

Due diagrammi separati:

- $|G(j\omega)|$  in funzione di  $\omega$
- $\arg G(j\omega)$  in funzione di  $\omega$

### **Diagramma di Nyquist**

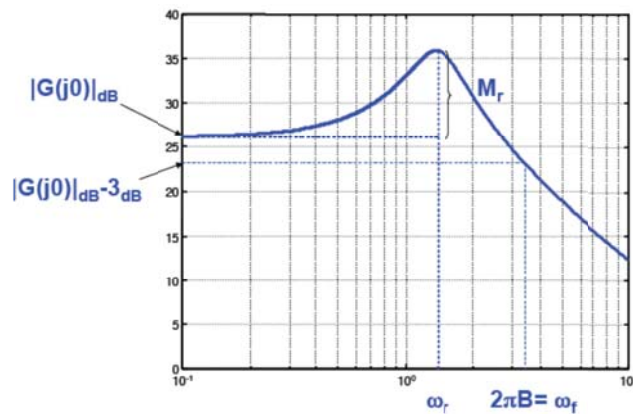
- Unica curva nel piano complesso parametrizzata in  $\omega$  :
- $\text{Im}G(j\omega)$  in funzione di  $\text{Re}G(j\omega)$



Il metodo dello studio della Risposta in Frequenza può essere meno intuitivo rispetto ad altri metodi che abbiamo analizzato in precedenza. Tuttavia, presenta alcuni vantaggi, soprattutto in situazioni reali, come funzioni di modellazione di trasferimento di dati fisici.

*Specifiche nel dominio della frequenza:*

- Caratterizzazione della risposta al gradino nel dominio del tempo
  - Permette di formulare delle specifiche di controllo in termini di  $M_p$ ,  $t_r$ ,  $t_s$
- Caratterizzazione della risposta armonica: si suppone che il sistema abbia comportamento simile a quello di un sistema del secondo ordine
  - Pulsazione di risonanza  $\omega_r$ : pulsazione in cui si ha  $\max |G(j\omega)|$
  - Picco di risonanza  $M_r: |G(j\omega_r)| - |G(j0)|$
  - Banda passante  $\omega_{BW} = 2\pi B$ : pulsazione in cui  $|G(j\omega)|$  è diminuito di 3dB rispetto a  $|G(j0)|$



[fig. 14]

*Correlazione tra specifiche nel dominio del tempo e specifiche nel dominio della frequenza:*

Relazioni approssimate

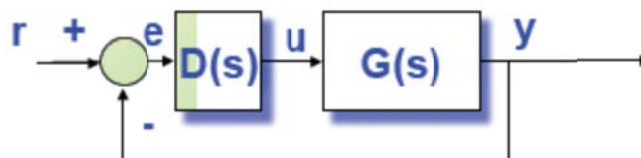
- $t_r$  piccolo  $\Rightarrow$  sistema “pronto”  $\Rightarrow$  “passano” frequenze elevate  $\Rightarrow B$  elevata

$$t_r B \approx 0.4$$

- Sistema poco smorzato ( $\zeta$  piccolo)  $\Rightarrow M_p$  grande  $\Rightarrow M_r$  grande

$$M_p \approx (M_r / |G(0)|) - 1$$

### **SINTESI DI BODE**



[fig. 15]

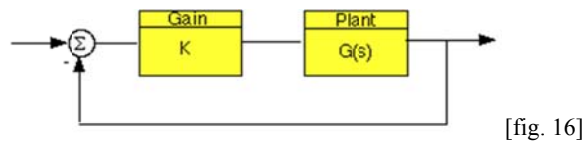
La Sintesi di Bode è basata sulla formulazione di alcune specifiche del sistema a catena chiusa, in termini di caratteristiche del termine in catena diretta  $L(s) = D(s)G(s)$ . E' detta anche sintesi per tentativi (richiede in genere diverse iterazioni).

Le specifiche sono: errore a regime  $|e_r, k| < \epsilon$  in risposta all'ingresso canonico con trasformata  $1/s^{k+1}$  (e nullo per indici  $< k$ ); margine di fase (per  $L(s)$ )  $PM > PM^*$ ; pulsazione di attraversamento (per  $L(s)$ )  $\omega_c = \omega^*$ .

$M_p$  nel sistema retroazionato e  $PM$  di  $G(s)$  sono legati tra di loro in maniera inversamente proporzionale (se  $PM$  cresce allora  $M_p$  cala). Infatti, un  $M_p$  elevato implica che il sistema retroazionato abbia poli vicini all'asse immaginario, di conseguenza è "vicino" all'instabilità e quindi possiede un  $PM$  piccolo. Nella sintesi di Bode si cerca di ottenere una sovralongazione soddisfacente agendo su  $PM$ . Tra la  $\omega_c$  di  $L(s) = D(s)G(s)$  e la  $B$  di  $T(s)$  sistema complessivo retroazionato, esiste un legame di proporzione diretta:  $\omega_c$  cresce se e solo se  $B$  cresce.

**NOTE: Margine di Fase PM e Margine di Guadagno GM**

Consideriamo il seguente sistema:

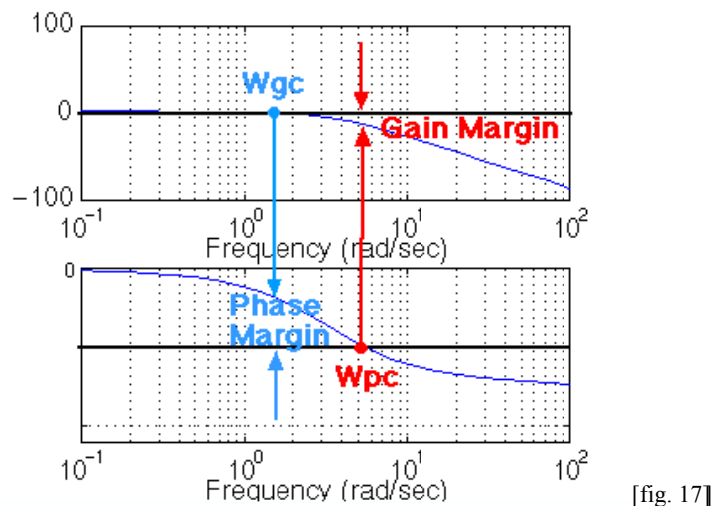


dove  $K$  è una variabile (costante) che indica il guadagno, mentre  $G(s)$  è il *Plant (impianto/sistema)* in esame.

Il *Margine di Guadagno* è definito come la variazione del guadagno ad anello aperto richiesta per rendere il sistema instabile. I sistemi con margini di guadagno elevati sono in grado di sopportare grandi cambiamenti nei parametri del sistema prima di diventare instabile in anello chiuso. Si tenga presente che, il guadagno unitario, è pari a un guadagno a zero in dB.

Il *Margine di Fase* è definito come la variazione di spostamento di fase ad anello aperto richiesta per rendere un sistema ad anello chiuso instabile. Esso, in particolare, serve ad indicare quanto il sistema è "robusto", ovvero immune ai ritardi.

Il *Margine di Fase*, nel diagramma di Bode, è la differenza di fase tra la curva di fase e -180 gradi nel punto corrispondente alla frequenza in cui vi è un guadagno di 0dB (the gain cross over frequency,  $\omega_{gc}$ ). Analogamente, il *Margine di Guadagno* è la differenza tra la curva dell'ampiezza del modulo e 0dB in corrispondenza alla frequenza dove la fase è di -180 gradi (the phase cross over frequency,  $\omega_{pc}$ ). Tali margini sono riportati nella seguente figura.



## 2.4.2 Diagramma di Bode e risposta ad anello aperto in Matlab

Il primo passo per risolvere il problema del controllo utilizzando la *Risposta in Frequenza*, in particolare i *Diagrammi di Bode*, è quello di verificare se il sistema in catena aperta alla risposta a gradino risulti stabile. Di seguito riportiamo lo schema a blocchi in *catena-aperta* e relativa funzione di trasferimento.



[fig. 18]

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

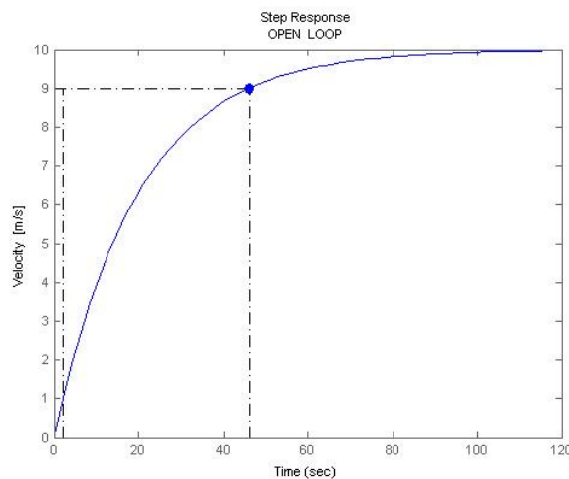
Codice *Matlab*:

```
Kp=1; % Guadagno unitario ANELLO APERTO
m =1000; % Massa del veicolo [kg]
b=50; % Coefficiente d'attrito [Nsec/m]
u=500; % Forza impressa al veicolo [N]

num= 1; % Numeratore della funzione
den=[m b]; % Denominatore della funzione
cruise=tf(num,den); % Funzione di trasferimento Gp(s)
contr = Kp; % Controllore

step(u*contr*cruise) % Risposta a gradino OPEN-LOOP
```

eseguendo il codice si ottiene:



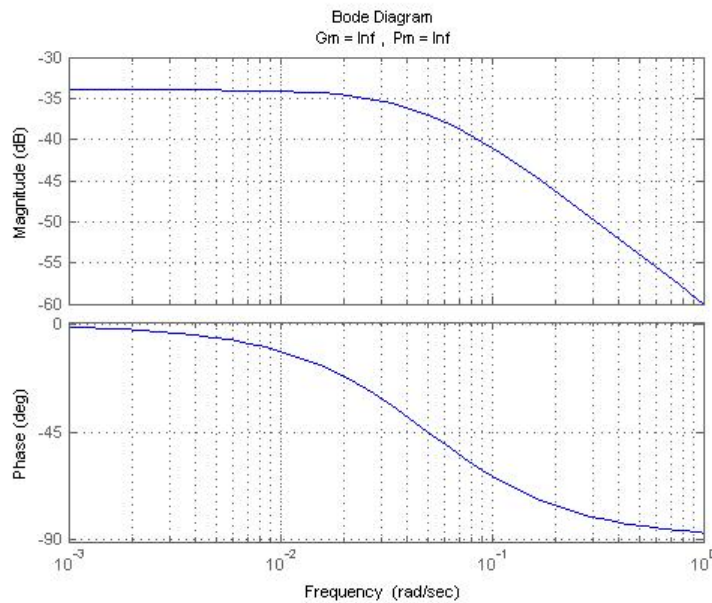
[fig. 19]

Come si può osservare dal grafico [fig. 19] la risposta a gradino del sistema in catena aperta è stabile, possiamo tracciare ora i *Diagrammi di Bode* utilizzando il seguente comando *Matlab*

```
margin(contr*cruise)
```

che, oltre a tracciarne i relativi diagrammi di *Modulo* e *Fase*, ne riporta i valori di PM e GM.

Eseguito il codice si ottengono quindi i seguenti diagrammi:



[fig. 20]

### 2.4.3 Controllore Proporzionale

Per analizzare quale e come sarà la risposta del sistema in catena-chiusa, in base alle informazioni dei *Diagrammi di Bode* del sistema in catena-aperta, focalizziamo l'attenzione su alcuni concetti:

- Il sistema deve essere stabile in *catena-aperta*, se abbiamo intenzione di progettare il controllore mediante i *Diagrammi di Bode* come già accennato in precedenza.
- Se  $WGC < WPC$  allora il sistema in *catena-chiusa* sarà stabile.
- Per i sistemi del secondo ordine, in *catena-aperta*, il coefficiente di smorzamento  $\xi$  viene approssimato dalla seguente relazione:

$$\xi = \frac{PM(deg)}{100}$$

Questo se PM è compreso tra 0 e 60 gradi. Altrimenti bisognerà procedere con cautela.

- Nella sintesi di progetto si assume in modo approssimativo  $B_\omega \approx \omega_n$
- Si assume inoltre la  $B_\omega$ , dove il sistema attenua il guadagno di  $M_{dB} = -6 \sim -7.5dB$
- Il *Rise time* è approssimato con la seguente relazione:

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

- Per lo *Steady State error* viene assunta la seguente approssimazione:

$$Steady State error = \frac{1}{1+(magnitudo)} 100\% \quad \text{dove: } 10^{\left(\frac{MdB}{20}\right)} = Magnitudo$$

Dopo questa premessa, prendiamo ora in considerazione il *Diagramma di Bode* [fig. 20] in *catena-aperta* tracciato in precedenza per  $K_P=1$ .

Da un primo sguardo possiamo approssimare che  $PM \approx 155^\circ$ , di conseguenza si avrà  $\xi = 1.55$ .

Mentre il guadagno in bassa frequenza possiamo considerarlo  $M_{dB} = -35dB$ ; quindi, utilizzando le relazioni espresse in precedenza si avrà  $Magnitude = 0.01778$  da cui un *Steady State error*  $\approx 98\%$ .

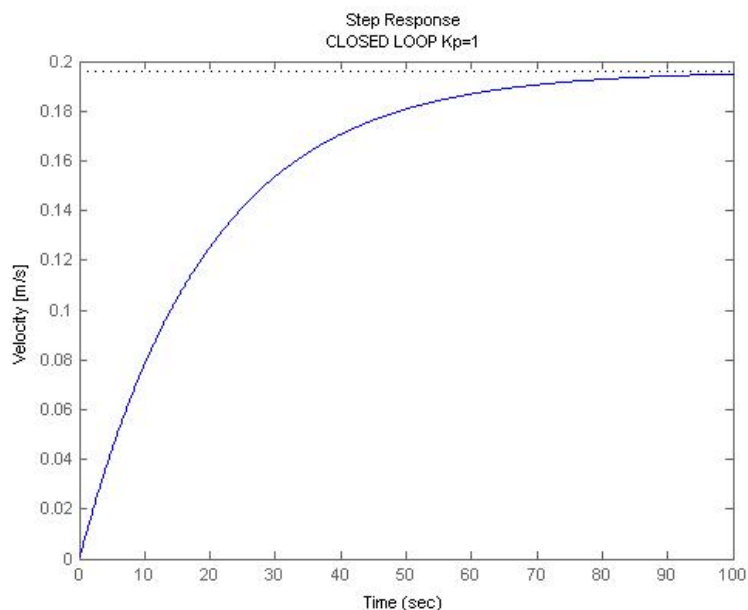
Per quanto riguarda il  $Tr$  possiamo a priori prevedere che sarà molto elevato.

Andiamo quindi a vedere quale sarà la *Risposta a Gradino* del sistema in catena-chiusa eseguendo il seguente codice Matlab:

```
Kp = 1; % Guadagno Proporzionale unitario
m = 1000; % Massa del veicolo [kg]
b = 50; % Coefficiente d'attrito [Nsec/m]
r = 10; % Velocità di riferimento [m/s]

num = [1]; % Numeratore Gp(s)
den = [m b]; % Denominatore Gp(s)
cruise = tf(num,den); % Funzione di Trasferimento Gp(s)
contr = Kp;

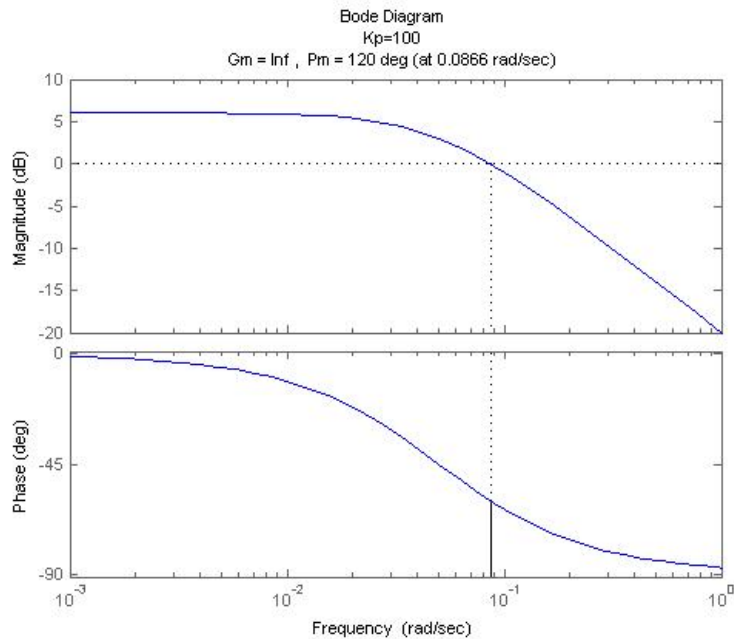
%_____ SISTEMA RETRAZIONATO_____%
sys_cl=feedback(contr*cruise,1);
t=0:0.1:20;
step(r*sys_cl,t) % Risposta a gradino CLOSED LOOP
axis([0 20 0 10])
```



[fig. 21]

Come si può osservare dal grafico [fig. 21] lo *Steady State error*  $\approx 98\%$ , non sono presenti fenomeni di sovraelongazione, e il  $Tr$  veramente elevato. Pertanto, in catena-chiusa, il sistema così com'è non va assolutamente bene. Bisognerà, quindi, aumentare la banda passante  $B_\omega$  e incrementare il guadagno  $M_{dB}$  in bassa frequenza. Questo (ragionando in termini *Bode*) consiste nel traslare verso l'alto il diagramma, e si ottiene incrementando il parametro  $K_P$ .

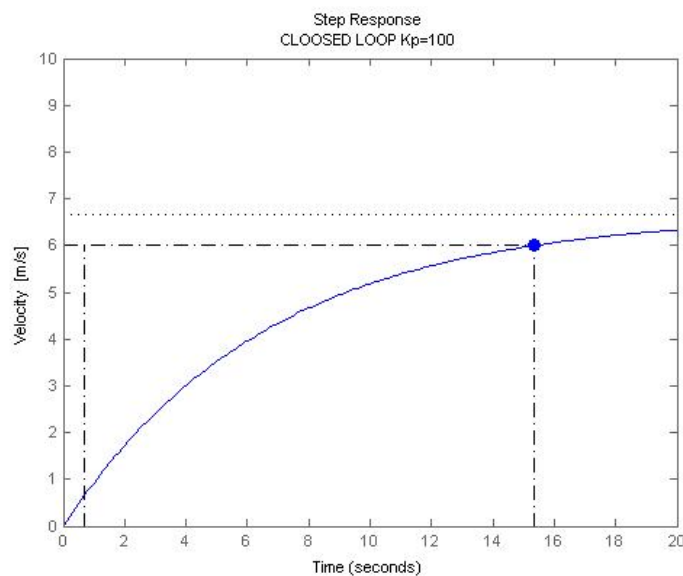
Proviamo quindi ad aumentare il guadagno proporzionale  $K_P$  di un valore pari a 100 e vediamo cosa accade.



[fig. 21]

Osservando i *Diagrammi di Bode* [fig. 21] ottenuti per  $K_P = 100$ , si nota subito che il diagramma del modulo si è traslato verso l'alto. Ora il guadagno in bassa frequenza è circa  $M_{dB} = 6dB$ , di conseguenza  $magnitude \approx 2$  e quindi lo *Steady State error*  $\approx 33\%$ , ovvero un errore di circa 3.3 m/s. Per la banda passante assumiamo  $B_\omega \approx 0.1$  rad/sec che comporterà un  $Tr \approx 18$  secondi; mentre, per quanto riguarda il coefficiente di smorzamento, essendo  $PM = 120^\circ$ , si ottiene  $\xi = 1.2$ .

Andiamo ora a vedere cosa accade alla *Risposta a Gradino* del sistema in *catena-chiusa* per  $K_P = 100$

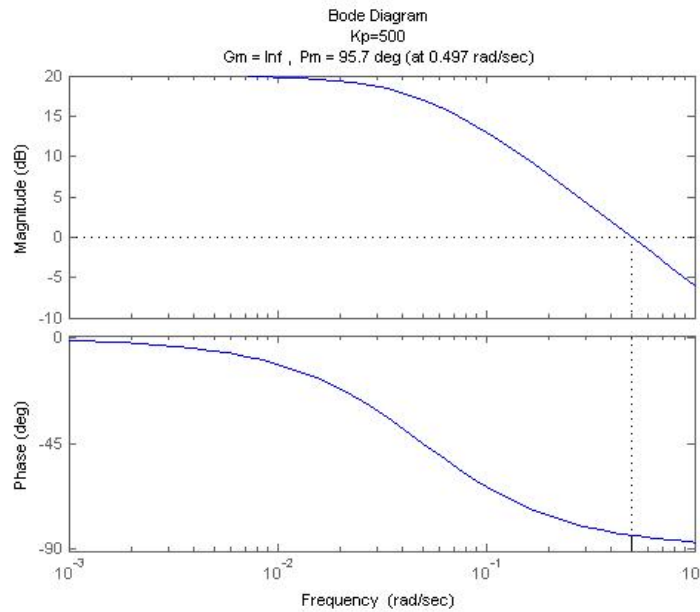


[fig. 22]

Da un primo sguardo alla [fig. 22], si può notare come lo *Steady State error* risulta circa 33%, mentre il *Rise time* risulta  $Tr \approx 14.6$  (si tenga presente che le formule adottate sono abbastanza approssimate). Inoltre, come previsto, in base al valore  $\xi = 1.2$  vi è l'assenza di *Overshoot*.

In conclusione, per  $K_P = 100$  le specifiche sul *Rise Time* e *Steady State error* non sono ancora essere rispettate.

Aumentiamo quindi ulteriormente il guadagno proporzionale e poniamo  $K_p = 500$  e vediamo cosa accade:

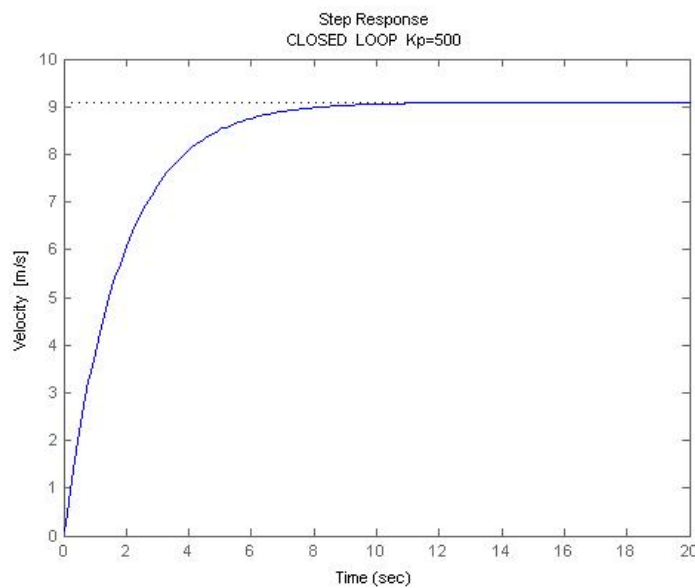


[fig. 23]

Dai *Diagrammi di Bode* [fig. 23], si può osservare che il diagramma del modulo si è traslato ulteriormente verso l'alto. Ora il guadagno in bassa frequenza è circa  $M_{dB} = 20dB$ , di conseguenza  $magnitude \approx 10$  e quindi lo *Steady State error*  $\approx 9\%$ , ovvero un errore di circa 1 m/s.

Per la banda passante assumiamo  $B_\omega \approx 0.6$  rad/sec, che comporterà un  $Tr \approx 3$  secondi; mentre, per quanto riguarda il coefficiente di smorzamento, essendo  $PM \approx 96^\circ$ , si ottiene  $\xi \approx 0.96$ .

Andiamo ora a vedere quindi cosa accade alla *Risposta a Gradino* del sistema in *catena-chiusa* per  $K_p = 500$



[fig. 23]



### Conclusioni

Dalla *Risposta a Gradino* in catena-chiusa ottenuta [fig. 23], si può notare che tutte le previsioni fatte sulle specifiche di progetto (*Rise time*, *Overshoot*, *Steady State error*) si sono verificate.

Assenza di *Overshoot*, *Rise time*  $T_r \approx 3$  secondi e *Steady State error*  $\approx 9\%$ .

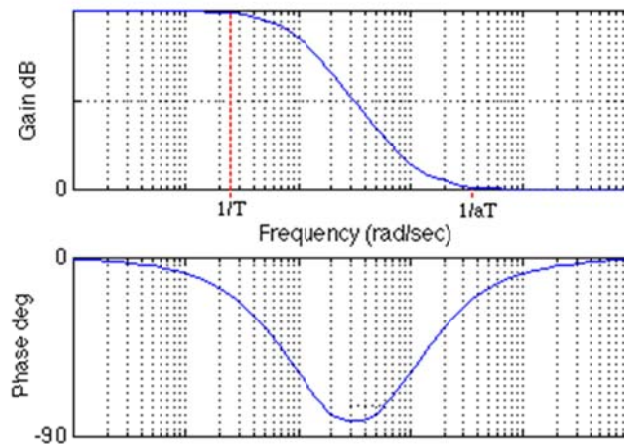
I valori ottenuti per il *Rise time*, e l' *Overshoot* risultano così rispettare le nostre specifiche di progetto, ovvero *Rise time*  $< 5\%$  e *Overshoot*  $< 2\%$ ; mentre, ciò che non risulta ancora rispettato è lo *Steady State error*. Per fare ciò, bisognerebbe continuare ad aumentare il guadagno proporzionale, di contro otteniamo un valore del *Rise time* sempre più piccolo, tale da raggiungere un valore irrealistico dal punto di vista fisico. Possiamo concludere, come già si era ottenuto in precedenza, che il problema del controllo mediante l'utilizzo di un solo controllore Proporzionale non è realizzabile. Di seguito, per tentare di risolvere il problema, proveremo ad utilizzare una *Rete Lag*.

### 2.4.4 Rete Ritardatrice Studio in Frequenza (Bode) in Matlab

Per il dimensionamento mediante lo *Studio in Frequenza* di una Rete Ritardatrice (*Lag controller*), la forma della funzione di trasferimento risulta scritta nella seguente maniera:

$$G_L(s) = \frac{1}{a} \frac{1 + aTs}{1 + Ts} \quad (a < 1)$$

I relativi *Diagrammi di Bode* risultano essere del tipo seguente:



[fig. 24]

Si può osservare [fig. 24] come le due frequenze di taglio siano posizionate in  $1/T$  e  $1/aT$ .

L'azione della Rete *Lag*, è quella di aggiungere guadagno alle basse frequenze con  $M_{db} = a$ . L'effetto di tale guadagno, è quello di ridurre in *catena-chiusa* lo *Steady State error* di un fattore pari al valore di  $a$ . Poiché per frequenze medio – alte, il guadagno del *compensatore Lag* è unitario, la risposta al transitorio e la stabilità non sono influenzate più tanto.

L'effetto collaterale della *Lag* è la fase negativa che è aggiunta tra le frequenze  $1/T$  e  $1/aT$ . Ciò dipende dal valore di  $a$ , un'aggiunta fino ad un valore di  $-90^\circ$  può essere accettabile. Bisogna, però, tenere presente che il *margin di fase PM* dell'intero sistema con *Lag* è ancora accettabile.

Dopo questa breve introduzione, procediamo ora con il dimensionamento della Rete Ritardatrice.

Per determinare i valori di  $a$  e di  $T$  in base alle specifiche di progetto, anziché la via analitica, si è scelta la via per tentativi. Tenendo presente che lo *Steady State error* cresce in base al valore di  $a$ , scegliamo  $a = 0.05$ . La scelta del valore  $T$  viene fatta tenendo presente che le due frequenze di taglio non devono essere vicine; poniamo quindi  $T = 700$ .



Riportiamo di seguito il codice *Matlab* che realizza il tutto:

```

Kp = 600; % Guadagno Proporzionale

m = 1000; % Massa del veicolo [kg]
b = 50;   % Coefficiente d'attrito [Nsec/m]
r = 10;   % Velocità di riferimento [m/s]

num = [1]; % Numeratore di Gp(s)
den = [m b]; % Denominatore di Gp(s)
cruise = tf(num,den); % Funzione di Trasferimento di Gp(s)

%___RETE LAG___%
a = 0.05;
T = 700;
numlag = [a*T 1]; % Numeratore di GL(s)
denlag = a*[T 1]; % Denominatore di GL(s)

%___CONTROLLER___%
contr = Kp*tf(numlag,denlag); % Kp*GL(s)

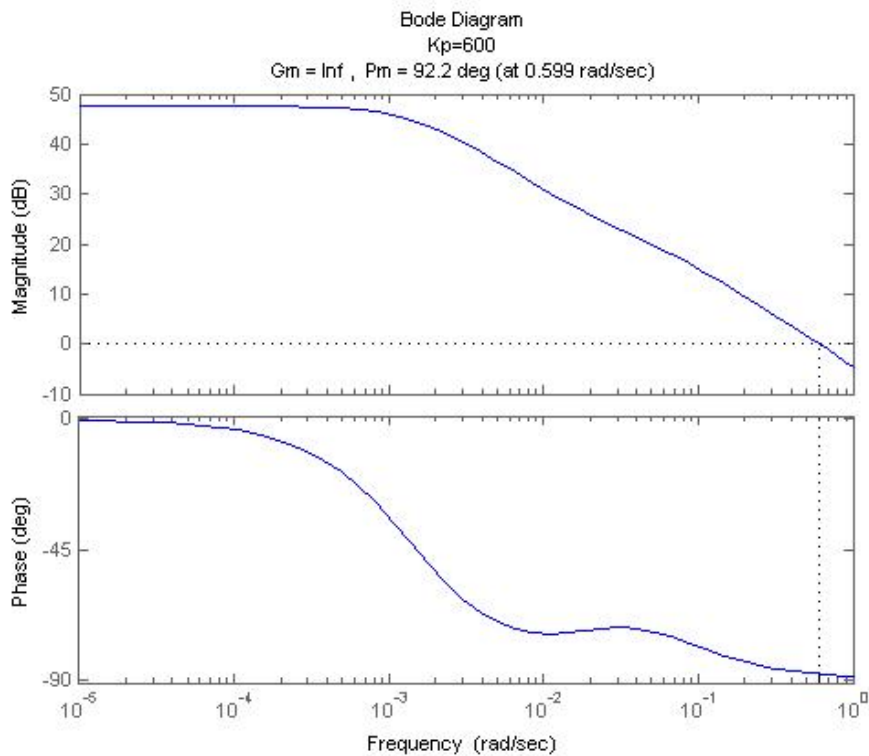
%Diagrammi di Bode OPEN-LOOP
margin(contr*cruise)

%___ CATENA RETROAZIONE___%
sys_cl=feedback(contr*cruise,1);
t=0:0.1:20;

%___RISPOSTA a GRADINO IN CATENA-CHIUSA
step(r*sys_cl,t)
axis([0 20 0 10])

```

I *Diagrammi di Bode* che si ottengono risultano i seguenti:



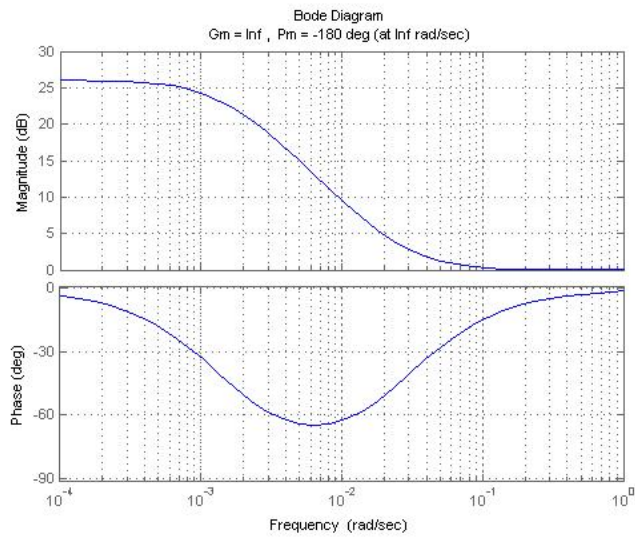
[fig. 25]

Per completezza di seguito riportiamo i *Diagrammi di Bode* della sola Rete *Lag* con i parametri da noi scelti, ottenuta eseguendo il seguente codice Matlab:

```
%____RETE LAG____%
a = 0.05;
T = 700;
numlag = [a*T 1]; % Numeratore di GL(s)
denlag = a*[T 1]; % Denominatore di GL(s)

lag=tf(numlag,denlag); % Funzione di Trasferimento GL(s)

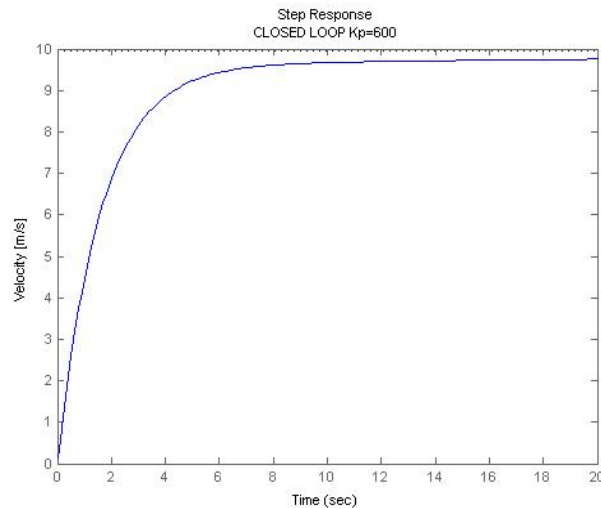
margin(lag) % Diagrammi di Bode con margini PM GM
```



[fig. 26]

Osservando il grafico [fig. 25] si nota subito che il guadagno in bassa frequenza è ora un valore più elevato e in questo caso  $M_{dB} \approx 50dB$ ; in base alle relazioni espresse in precedenza si ottiene che lo *Steady State error*  $\approx 0.3\%$ . Per la banda passante si assume  $B_{\omega} \approx 0.6 \text{ rad/sec}$  che comporterà un  $T_r \approx 3$  secondi; mentre, per quanto riguarda il coefficiente di smorzamento, essendo  $PM \approx 92^\circ$ , si ottiene  $\xi = 0.92$ .

Riportiamo di seguito il grafico della *Risposta a Gradino* in *catena-chiusa*:



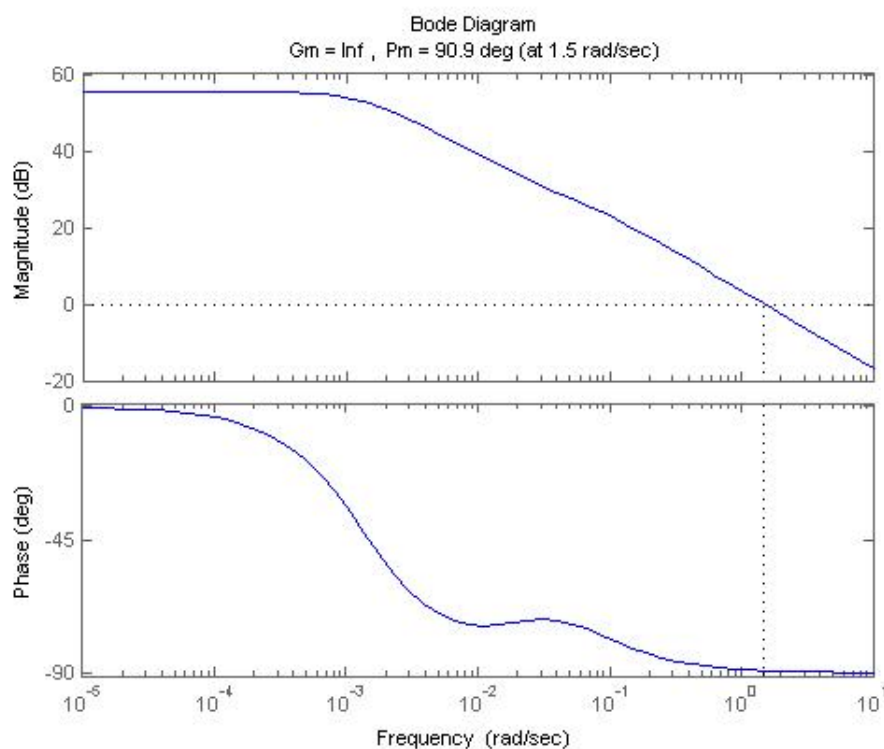
[fig. 27]

Dal grafico [fig. 27] della *Risposta a Gradino* in *catena-chiusa*, si può constatare che i valori preventivamente e ottenuti dal analisi dei *Diagrammi di Bode* in *catena-aperta* sono stati rispettati. Inoltre, possiamo aggiungere che le nostre tre specifiche di progetto (*Rise Time*, *Steady State error*, *Overshoot*) risultano ora essere tutte rispettate. Tuttavia il Tempo di Assestamento (*Settling time*), una specifica che sin ora non abbiamo mai preso in considerazione nella nostra analisi, e che si ottiene dalla seguente relazione:

$$T_s = \frac{4.6}{\xi \omega_n}$$

risulta essere  $T_s \approx 8.33$  secondi quindi troppo elevato.

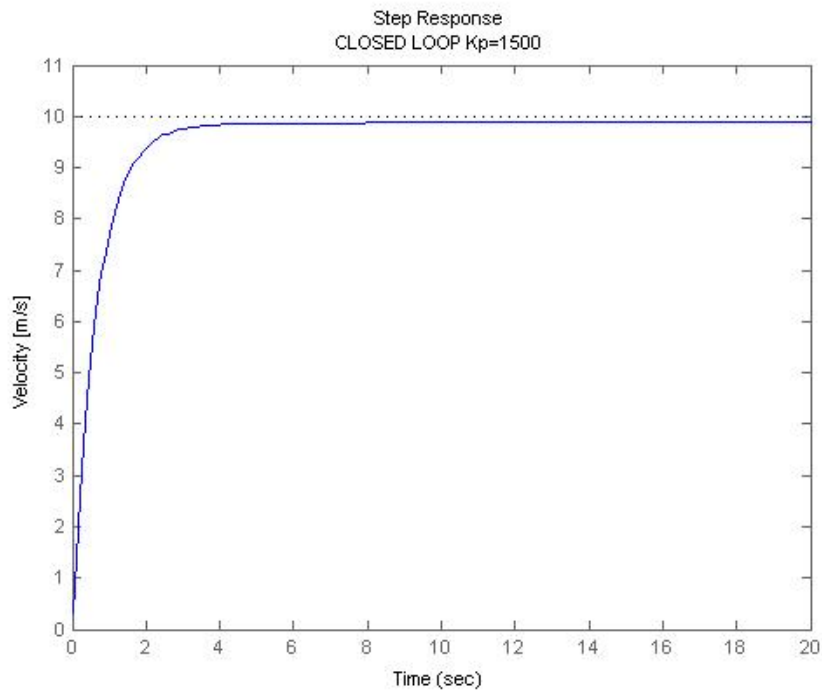
Proviamo quindi ad aumentare nuovamente il parametro  $K_p = 1500$ , e vediamo cosa accade:



[fig. 28]

Osservando il grafico [fig. 28] si nota subito che il guadagno in bassa frequenza è ora un valore ancora più elevato e in questo caso  $M_{dB} \approx 55$ dB; in base alle relazioni espresse in precedenza si ottiene che lo *Steady State error*  $\approx 0.177\%$ . Per la banda passante si assume  $B_\omega \approx 1.5$  rad/sec che comporterà un  $Tr \approx 1.5$  secondi; mentre, per quanto riguarda il coefficiente di smorzamento, essendo  $PM \approx 91^\circ$ , si ottiene  $\xi = 0.921$ . Il *Settling time*  $T_s \approx 3.3$  secondi.

Riportiamo di seguito il grafico della *Risposta a Gradino* in *catena-chiusa*:



[fig. 29]

### **Conclusioni**

Dal grafico [fig. 29] della *Risposta a Gradino* in *catena-chiusa*, si può constatare che i valori preventivamente ottenuti dall'analisi dei *Diagrammi di Bode* in *catena-aperta* sono stati rispettati. Inoltre, possiamo aggiungere che le nostre tre specifiche di progetto (*Rise Time*, *Steady State error*, *Overshoot*) risultano ora essere tutte rispettate; il Tempo di Assestamento (*Settling time*) risulta anch'esso migliorato notevolmente essendo ora  $T_r \approx 3.3$  secondi.

In conclusione, possiamo affermare che la Rete *Lag*, analizzata e sintetizzata mediante l'utilizzo del metodo della *Risposta in frequenza*, risolve con successo il nostro problema del controllo.

## CAPITOLO 3

### TERZA FASE – SIMULINK

#### 3.1 MODELLAZIONE *CRUISE CONTROL SYSTEM* IN SIMULINK

Simulink è un'estensione grafica di *MATLab* per la modellazione e simulazione dei sistemi.

Di seguito, andremo a riproporre alcuni dei sistemi analizzati e realizzati in codice *Matlab*, del *Cruise Control* a partire dalla determinazione del modello matematico che ne descrive il sistema fisico.

##### 3.1.1 Costruzione Modello

Riportiamo di seguito l'equazione matematica che modella il nostro sistema fisico preso in esame, ottenuta all'inizio del nostro studio:

$$m\dot{v}(t) + dv(t) = u(t)$$

e da questa ne ricaviamo la relativa presentazione grafica tramite simulink che per comodità riscriviamo nel seguente modo:

$$m \frac{dv}{dt} = u - bv$$

dalla quale partiremo per realizzare lo schema.

Riportiamo anche i tre parametri adottati per il nostro studio:

$m = 1000\text{Kg}$  massa automobile

$b = 50\text{Nsec/m}$  coefficiente d'attrito

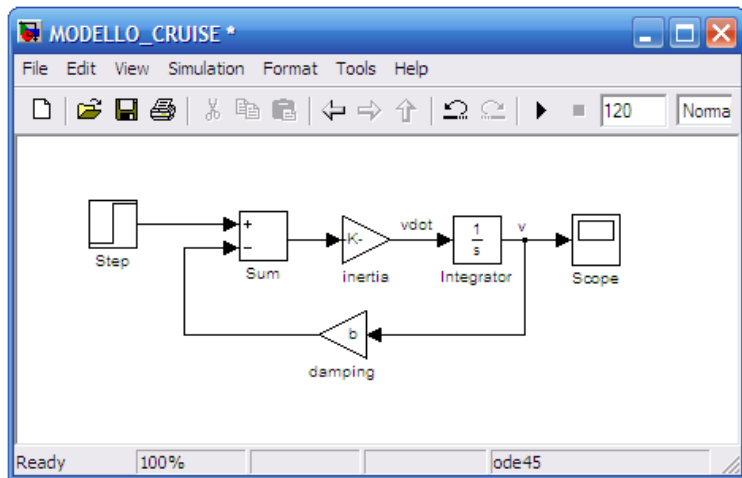
$u = 500\text{N}$  forza impressa dal motore

Il sistema viene modellato sommando le forze che agiscono sulla massa ed integrando l'accelerazione.

A seguire si presentano i vari passaggi in modo schematico per la realizzazione del modello:

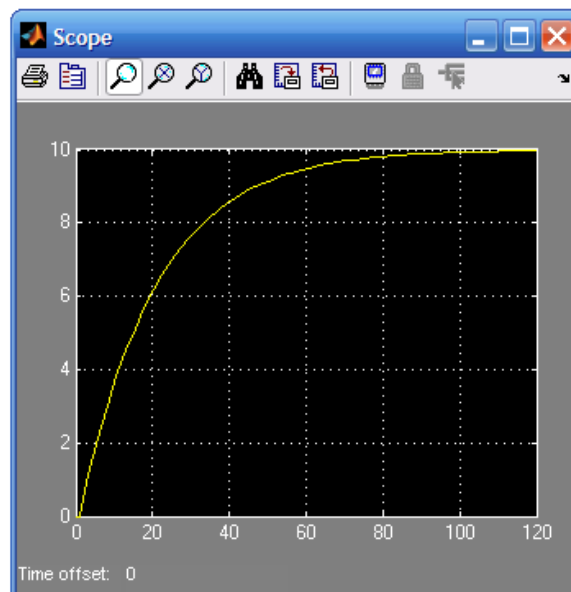
- Si modella l'integrale dell'accelerazione, inserendo un blocco integratore.
- Poiché la somma delle forze viene divisa per la massa, si inserisce un blocco Gain caratterizzato da un guadagno  $1/m$  (*inertia*)
- Si introducono le forze mediante un blocco sommatore per la somma di  $u$  e  $b$
- Un blocco Gain retroazionato di guadagno  $b$  (*damping*)
- La Forza  $u$  che viene impressa alla massa del veicolo tramite un blocco step che simula l'ingresso a gradino (cliccando sopra si impostano i parametri valore iniziale zero finale  $u$ )
- Inoltre, per poter visualizzare il valore assunto dalla variabile d'uscita che nel nostro caso risulta essere la velocità, si inserisce un blocco Scope

Di seguito viene riportato lo schema del sistema in *Simulink* ottenuto:



[fig. 30]

Prima di lanciare la simulazione, nel menu *simulation* si configurano i tempi di simulazione impostando un tempo max di 120 secondi. Cliccando su *start*, inizia così il processo e il blocco *Scope*, dopo aver impostato *autoscale*, visualizzerà a schermo la seguente risposta:



[fig. 31]

Osservando [fig. 31] si può notare che la *Risposta a Gradino*, in *catena-aperta* è la stessa ottenuta in precedenza con il *codice Matlab*.

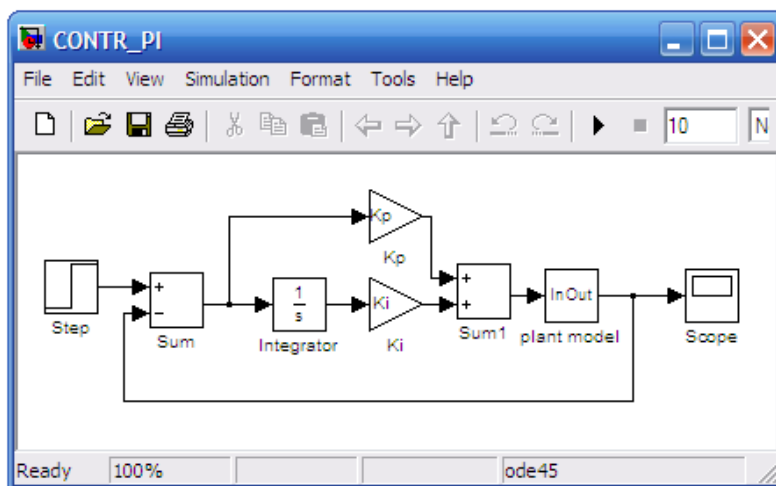
### 3.1.2 Implementazione controllore PI/PID con *Simulink*

Dalle analisi e sintesi del problema del controllo fatte in precedenza, si è ottenuto che il controllore PI (Proporzionale Integrale) risulta soddisfare le specifiche di progetto e risolvere in maniera abbastanza ottimale il problema del *Cruise Control*. Vediamo quindi di seguito la sua modellizzazione tramite *Simulink*

Richiamiamo di seguito la funzione di trasferimento con controllore PI:

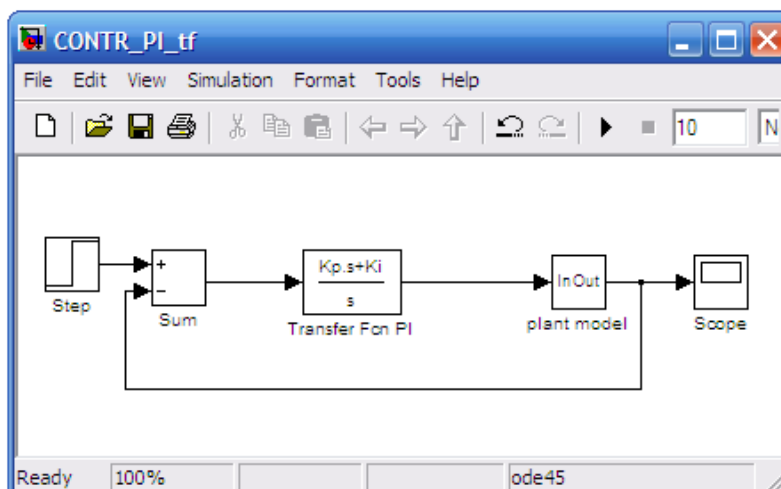
$$\frac{Y(s)}{U(s)} = \frac{K_p s + K_i}{ms^2 + (b + K_p) + K_i}$$

Come prima cosa utilizziamo un blocco Subsystem nel quale copieremo il modello in catena-aperta ottenuto in precedenza (*Plant model*) Dopodichè procediamo con l'inserimento di vari blocchi fino ad ottenere il seguente schema finale:



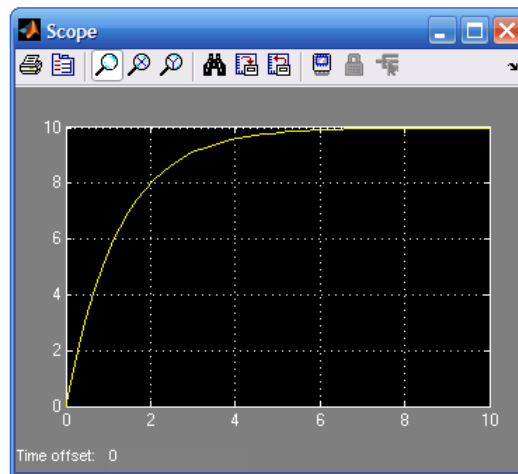
[fig. 32]

Utilizzando sempre il Subsystem e otteniamo il modello Controllore PI definitivo:



[fig. 33]

Se si impostano  $K_p=800$  e  $K_i=40$ , valori ottenuti nella precedente analisi, e se si lancia la simulazione della risposta a gradino, il blocco Scope visualizzerà la seguente traccia:

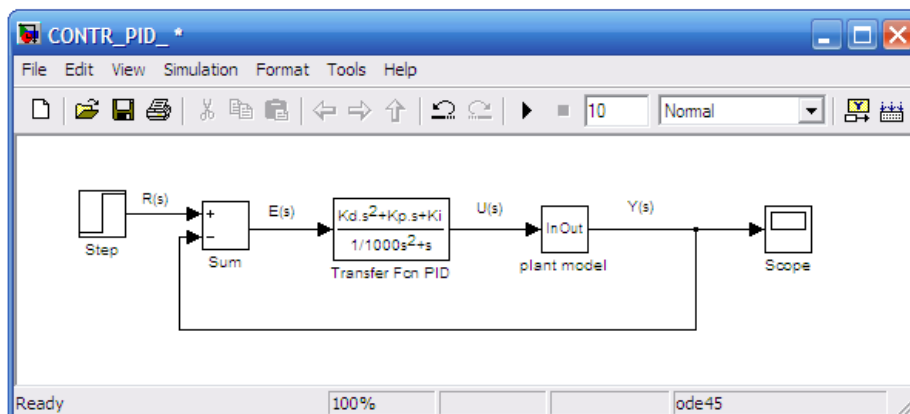


[fig. 34]

La traccia visualizzata dallo *Scope* [fig. 34] è uguale a quella che si era ottenuta in precedenza con il codice *Matlab*.

### Controllore PID con Simulink:

Per la simulazione del PID è necessario aggiungere un polo in alta frequenza e rendere reale la funzione, altrimenti Simulink non permette la simulazione.



[fig. 35]



Se si impostano  $K_p=800$ ,  $K_i=40$  e  $K_d=40$  (valori ottenuti nella precedente analisi), e se si lancia la simulazione della risposta a gradino, il blocco Scope visualizzerà la seguente traccia:



[fig. 36]

La traccia visualizzata dallo *Scope* [fig. 36] risulta uguale a quella che si era ottenuta in precedenza con il codice *Matlab*

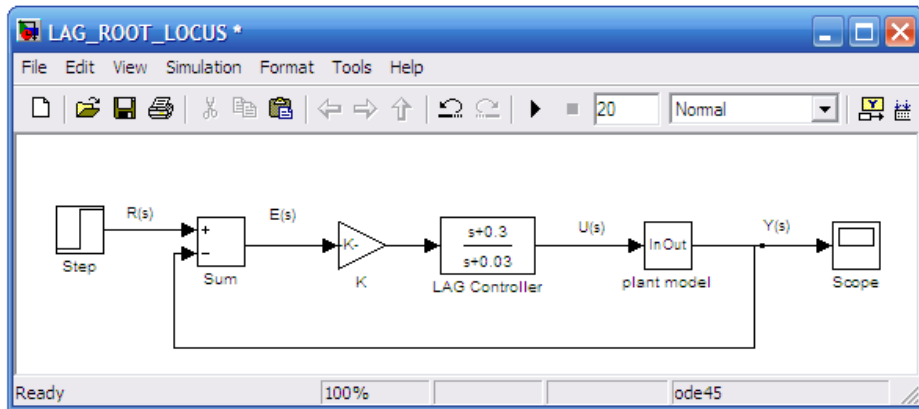
### 3.1.3 Implementazione Rete Ritardatrice (Luogo delle Radici) con *Simulink*

Analizziamo ora in *Simulink*, la soluzione del controllo tramite la Rete Lag ottenuta in precedenza con il Luogo delle Radici.

Ricordiamo di seguito la Funzione di Trasferimento *Rete Lag* e la relativa Funzione di Trasferimento in *catena-chiusa* ottenuta con tale controllo.

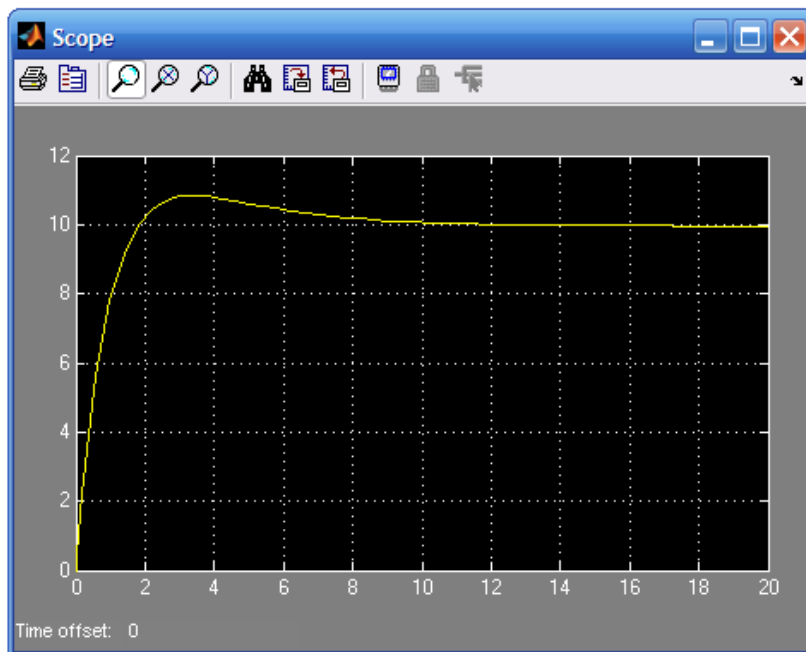
$$G_L(s) = \frac{s + Z_o}{s + P_o}$$

$$\frac{Y(s)}{U(s)} = \frac{K_p s + K_p P_o}{ms^2 + (b + P_o + K_p)s + (bP_o + K_p Z_o)}$$



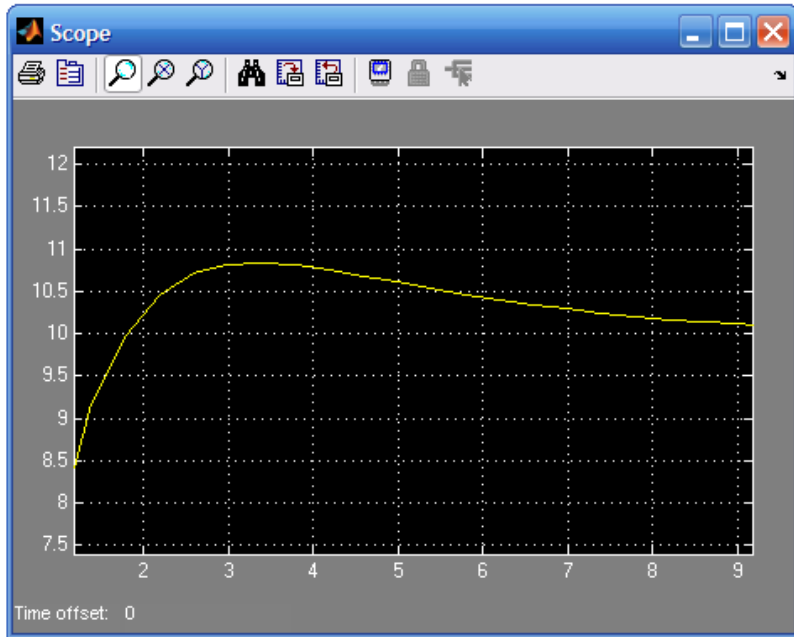
[fig. 37]

Lanciamo la simulazione e vediamo cosa visualizza lo Scope



[fig. 38]

Osservando [fig. 38], si nota che la *Risposta a Gradino in catena-chiusa*, visualizzata dallo *Scope*, risulta essere uguale a quella ottenuta in precedenza. Inoltre, in [fig. 39], grazie alla funzione zoom dello scope, possiamo osservare meglio come il fenomeno dell' *Overshoot* risulti rispettare le specifiche.



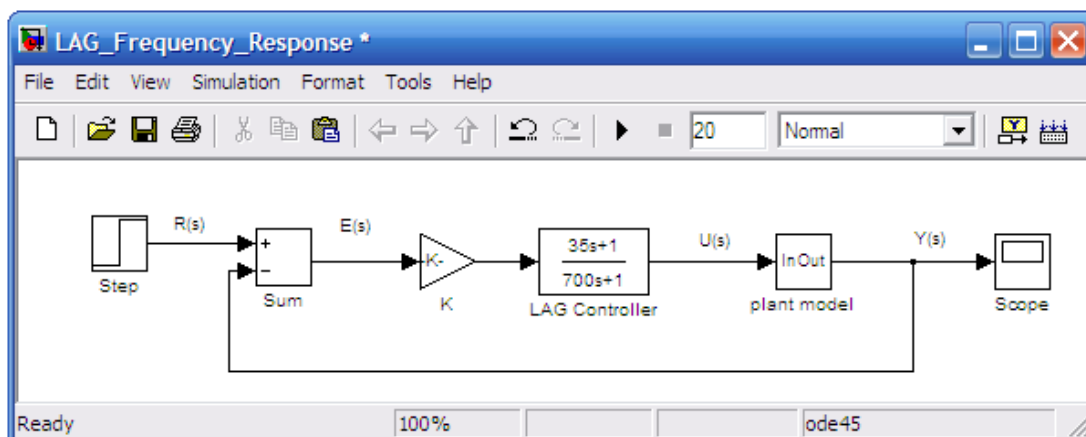
[fig. 39]

### 3.1.4 Implementazione Rete Ritardatrice (Studio in frequenza) con *Simulink*

Analizziamo ora in *Simulink*, la soluzione del controllo tramite la Rete Lag ottenuta in precedenza con lo *Studio in Frequenza*.

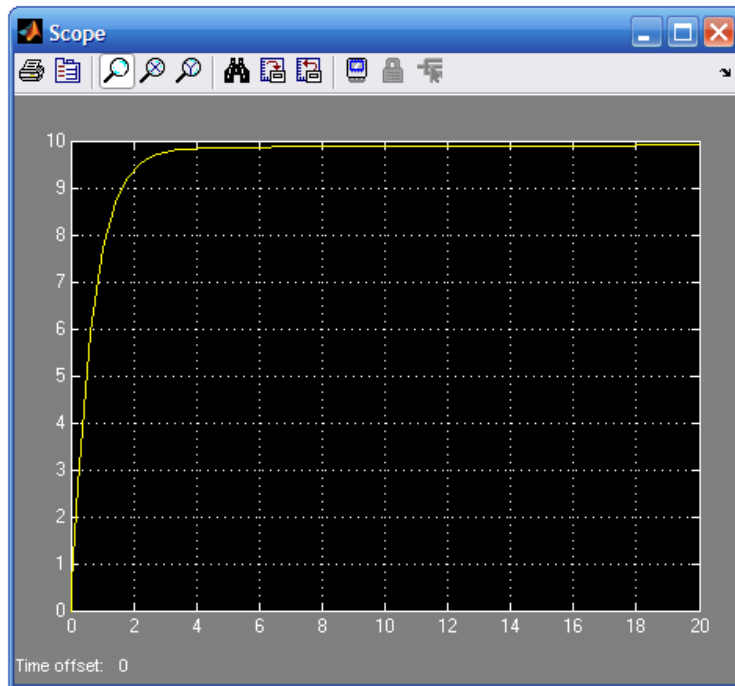
Ricordiamo di seguito la Funzione di Trasferimento *Rete Lag* :

$$G_L(s) = \frac{1}{a} \frac{1 + aTs}{1 + Ts} \quad (a < 1)$$



[fig. 40]

Lanciamo la simulazione e vediamo cosa accade:



[fig. 41]

Osservando [fig. 41], si nota che la *Risposta a Gradino* in *catena-chiusa*, visualizzata dallo *Scope*, risulta essere uguale a quella ottenuta in precedenza.

### 3.1.5 Reiezione dei disturbi

Vogliamo ora osservare come si comportano alla presenza di un disturbo i vari controllori analizzati finora. Per fare ciò, prendiamo in considerazione, come disturbo, la variazione di pendenza della strada. In base alle relazioni del *Piano Inclinato*, sappiamo che la forza peso si scompone in una componente parallela al piano, la quale si oppone al moto mediante la seguente relazione:

$$F = mg \sin \theta$$

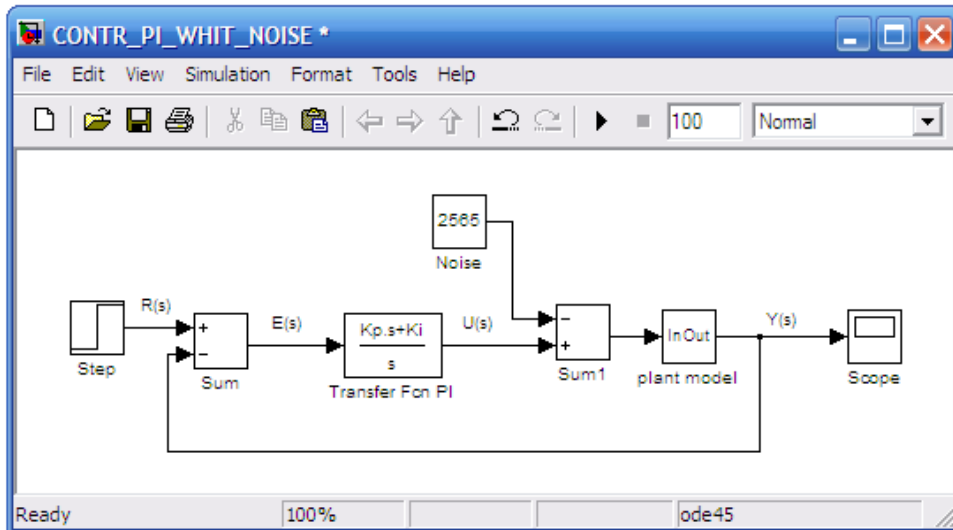
dove  $m$  è la massa,  $g$  l'accelerazione di gravità ( $g \approx 9.8\text{m/s}^2$ ),  $\sin\theta$  l'angolo.

Nel nostro caso la massa dell'automobile è  $m = 1000\text{Kg}$ , per tanto si può osservare che la variazione di  $1^\circ$  della pendenza comporta una forza  $F \approx 171\text{N}$ .

Secondo questa relazione, se consideriamo una strada con una pendenza di  $15^\circ$ , avremo una forza che si oppone al moto dell'automobile pari circa a  $2565\text{N}$ . Di seguito andremo a osservare come rispondono a tale disturbo i controllori da noi presi in considerazione.

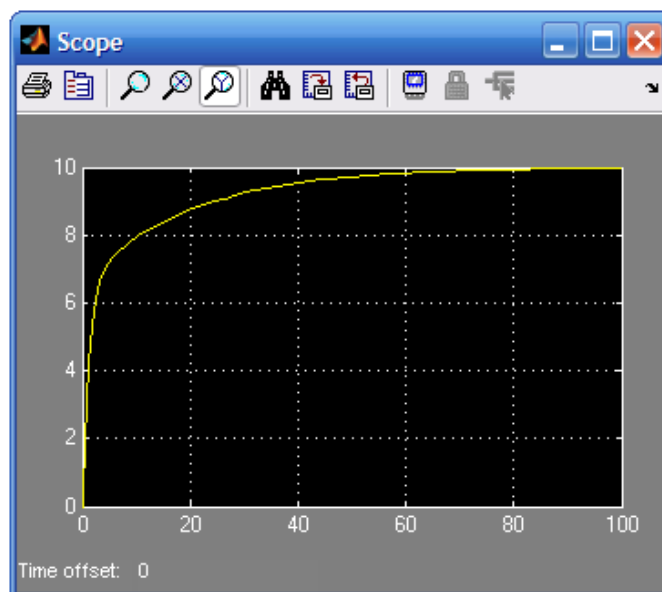
## Controllore PI

Con la presenza del disturbo lo schema è il seguente:



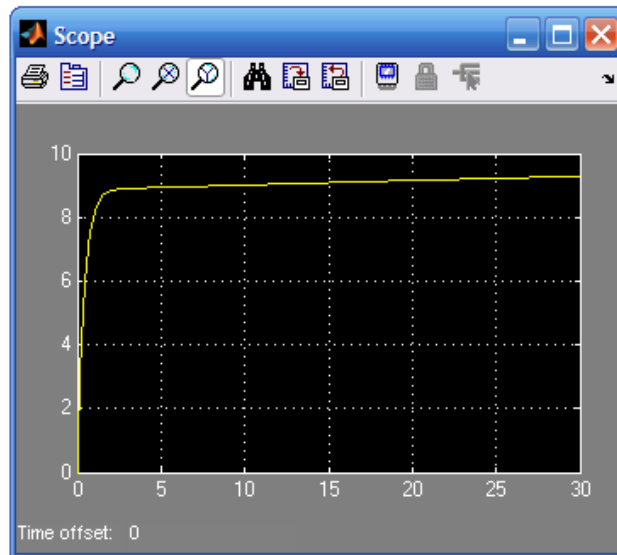
[fig. 42]

Lanciamo la simulazione e vediamo cosa accade:



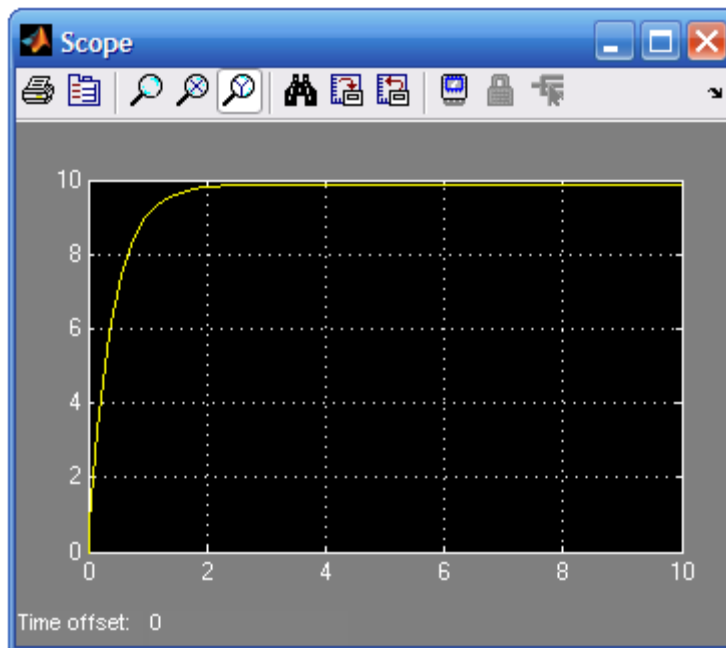
[fig. 43]

Osservando [fig. 43], *Risposta a Gradino in catena-chiusa* con presenza di un disturbo, si nota subito come ora il *Rise Time* è  $T_r \approx 25$  secondi, ben maggiore dei 5 secondi assegnati dalle specifiche; mentre, le specifiche dello *Steady State error* e dell' *Overshoot* sono ancora rispettate. Tale situazione con disturbi si migliora di molto solamente per  $K_p \approx 2500$  [fig. 44].



[fig. 44]

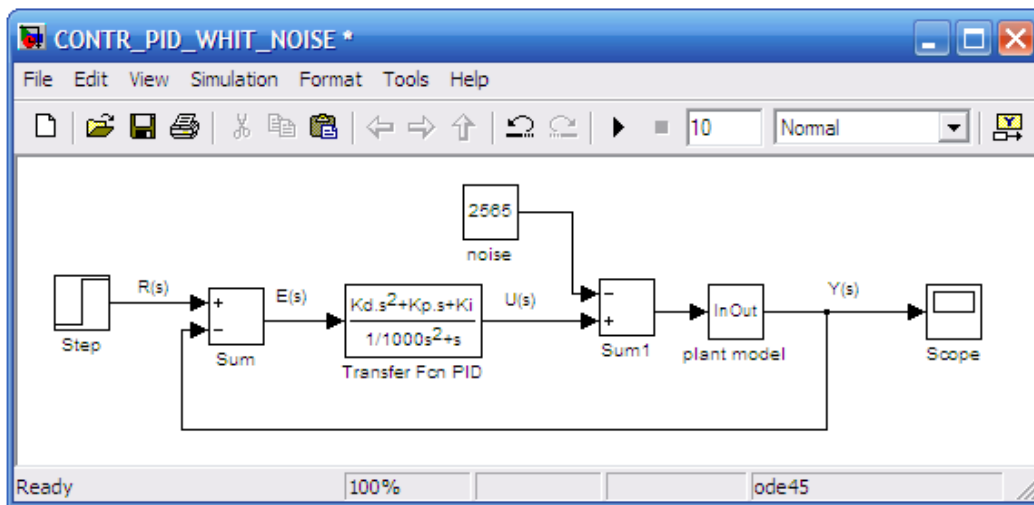
Tale valore di  $K_p$ , però, renderebbe, come già accennato in precedenza, il *Rise time* irrealistico dal punto di vista fisico quando il sistema è esente dai disturbi; nel nostro esempio, coincide con una pendenza nulla della strada, come possiamo notare dalla simulazione riportata in [fig. 45].



[fig. 45]

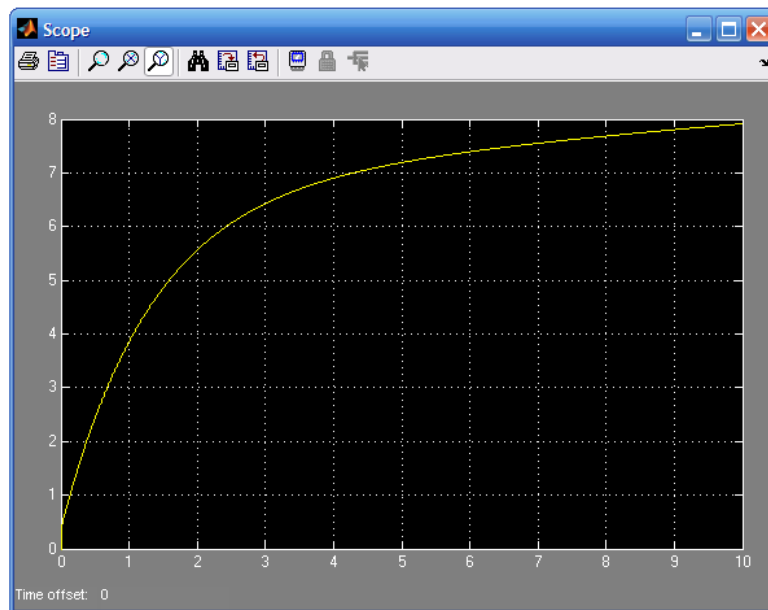
## Controllore PID

Con la presenza del disturbo lo schema è il seguente:



[fig. 46]

Lanciamo la simulazione e vediamo cosa accade:

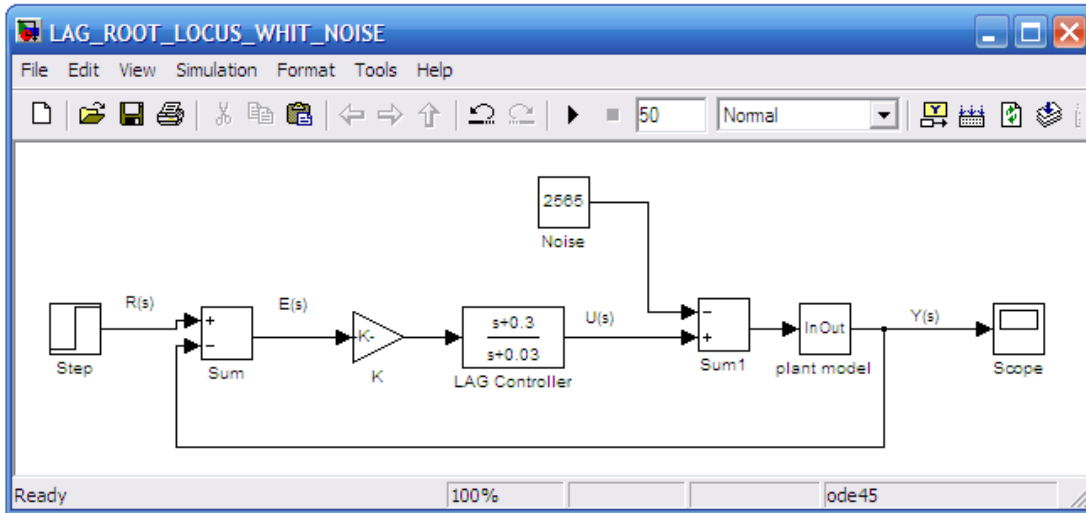


[fig. 47]

Osservando [fig. 47], *Risposta a Gradino in catena-chiusa* con presenza di un disturbo, si nota subito come ora il *Rise Time* risulti molto elevato  $T_r \approx 25$  secondi, ma soprattutto si nota che il sistema raggiungerà la velocità di regime in un tempo molto lungo. Per tale motivo, proviamo a settare nuovamente i valori  $K_p$ ,  $K_i$  e  $K_d$  per tentare di ridurre tale problema, che in assenza di disturbi non si presenta.

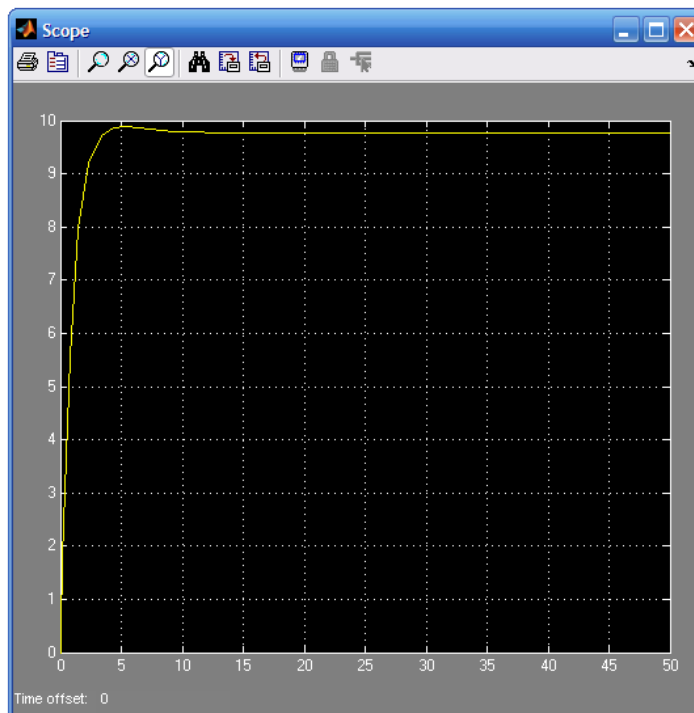
### Rete LAG con *Luogo delle Radici*

Con la presenza del disturbo lo schema è il seguente:



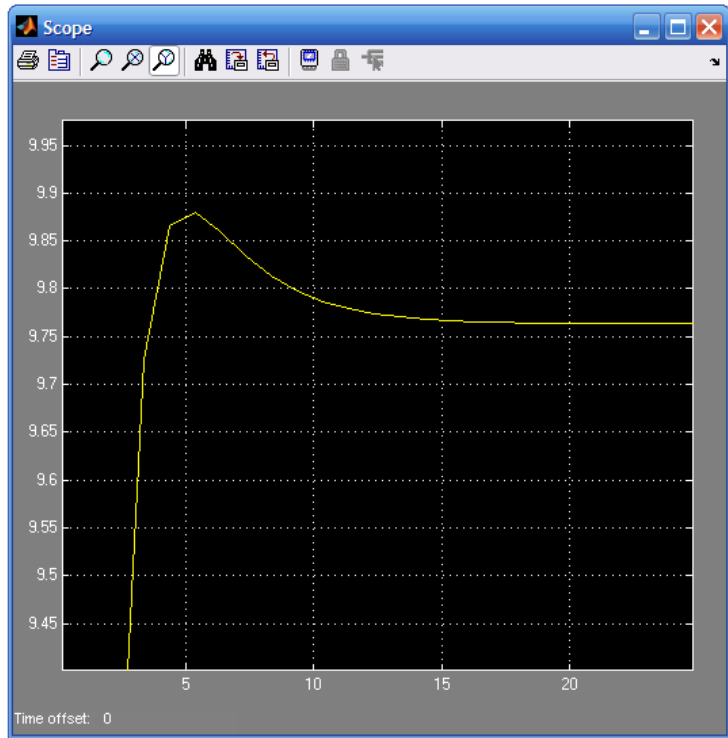
[fig. 48]

Il valore di  $K = 1291$  è quello rilevato con il metodo del *Luogo delle Radici*. Lanciamo la simulazione e vediamo cosa accade:

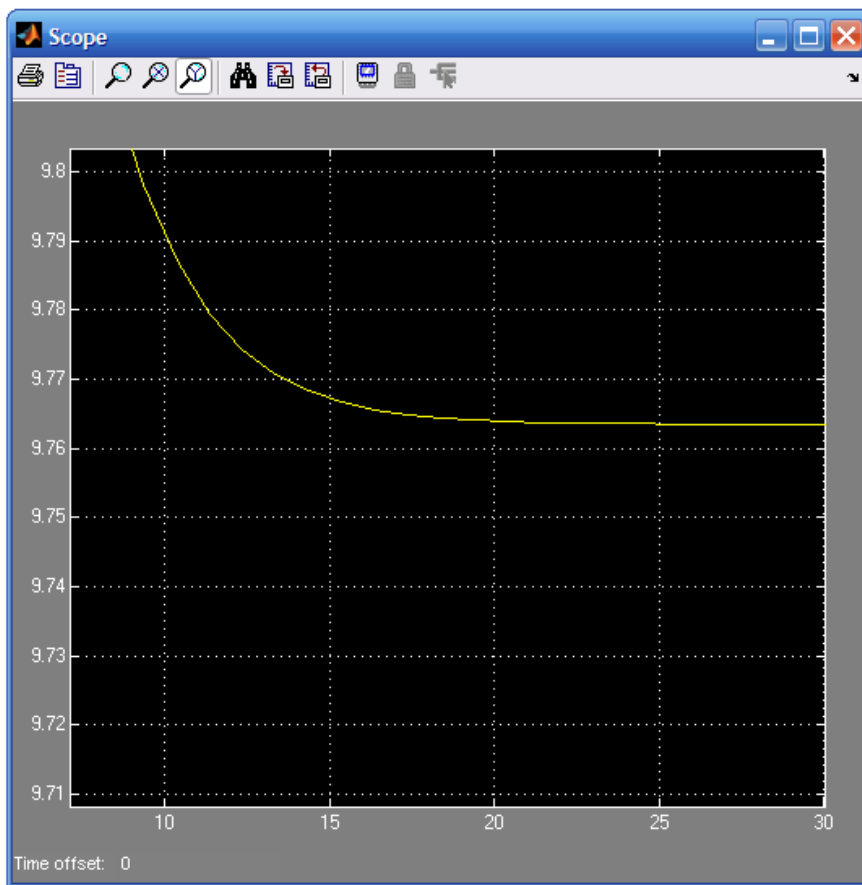


[fig. 49]





[fig. 50]

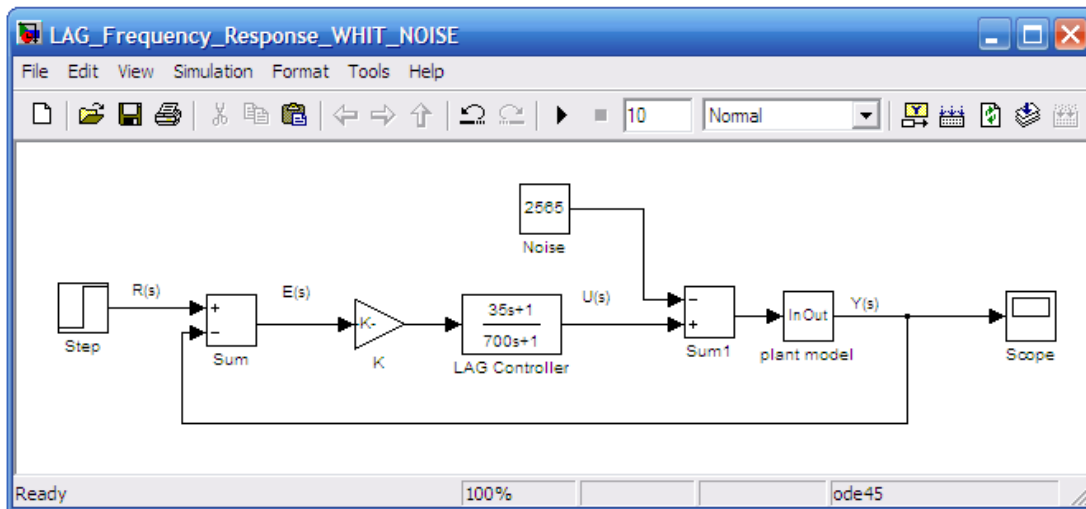


[fig. 51]

Osservando i grafici [fig. 49, 50, 51], possiamo notare come la *Rete Lag*, ottenuta con il metodo del *Luogo delle Radici*, risponda in modo molto performante al disturbo in esame. Il *Rise time* è infatti  $T_r \approx 2.2$  secondi; la sovraelongazione, introdotta dalla *Rete Lag* raggiunge un valore  $M_p \approx 9.88$  quindi la specifica sull' *Overshoot* è rispettata; lo *Steady State error*, che da specifica deve essere inferiore al 2%, risulta di poco superiore, e come si può notare in [fig. 51] la velocità a regime si assesta attorno alla  $V \approx 9.765$  m/s, che consiste in uno 0.4% in più rispetto al valore di specifica 2%.

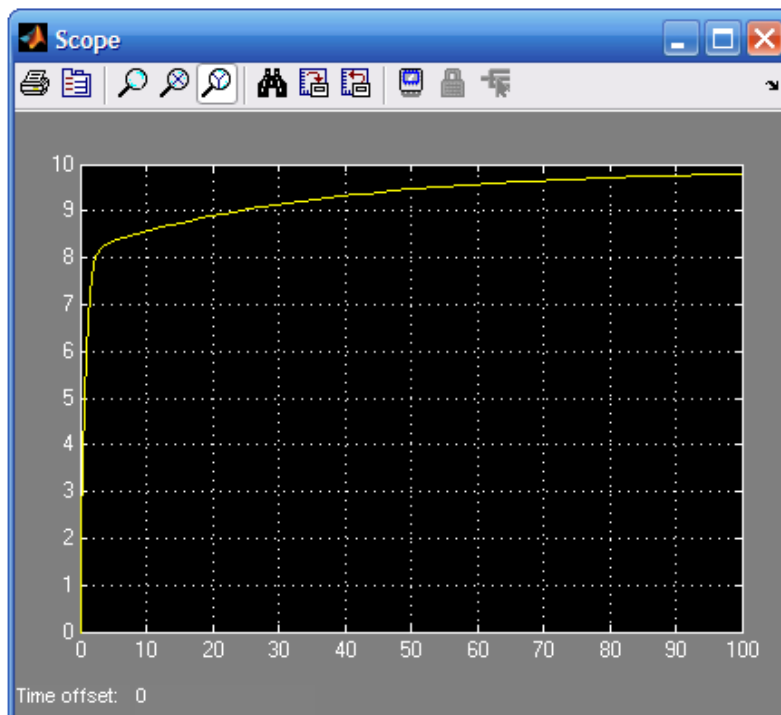
### Rete LAG con Studio in Frequenza

Con la presenza del disturbo lo schema risulta essere il seguente:

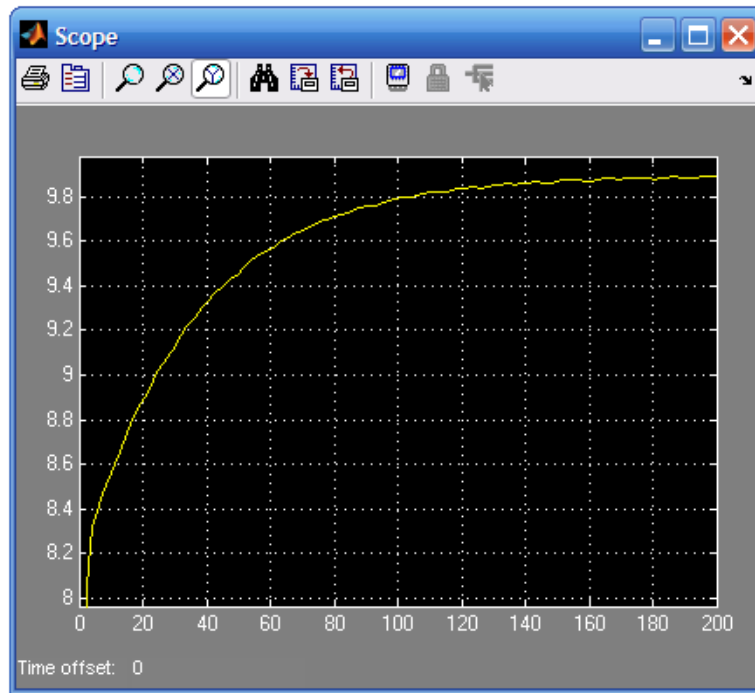


[fig. 52]

Lanciamo la simulazione e vediamo cosa accade:



[fig. 53]



[fig. 54]

Osservando [fig. 53, 54], *Risposta a Gradino in catena-chiusa* con presenza di un disturbo, si nota subito come il *Rise Time* risulti molto elevato,  $T_r \approx 25$  secondi, ma soprattutto che il sistema raggiungerà la velocità di regime (rispettando la specifica dello *Steady state error*), in un tempo molto lungo  $T_s \approx 100$  secondi.

## CAPITOLO 4 CONCLUSIONI

Lo scopo del mio lavoro era di sintetizzare e analizzare un prototipo di *cruise control*, con specifiche di progetto assegnate, mediante l'utilizzo del programma Matlab.

Per raggiungere lo scopo sono stati affrontati tre metodi di approccio alla soluzione del problema:

- Controllore PID
- Luogo delle Radici
- Studio in Frequenza

In un momento successivo, mediante lo strumento Simulink, se ne è realizzata una modellizzazione e simulazione, valutando anche la loro reiezione dei disturbi.

Giunti a questo punto non resta che mettere a confronto le tre soluzioni ottenute.

Con il metodo del controllore PID, si è potuto osservare come il problema del controllo risultasse già risolto mediante l'utilizzo di un controllore PI, anziché usare un PID. Pertanto, in fase di progetto e di scelta, considererei di adottare la soluzione con controllore PI, essendo più semplice da realizzare. Questo perché la funzione di trasferimento del PI non risulta essere impropria e quindi non necessita alcuna aggiunta di poli in alta frequenza. D'altro canto bisogna osservare però che dalla simulazione di un disturbo, mediante Simulink, in entrambi i controllori si è evidenziato un forte incremento del tempo di risposta  $T_r \approx 25$  secondi. Lo *Steady State error* veniva raggiunto e rispettato, ma in un tempo  $T_s \approx 100$  secondi.

Col *Luogo delle Radici* siamo giunti alla conclusione che il controllore che risolve il problema risulta essere una Rete Ritardatrice (*Lag*). Rispetto, però, ai controllori PI e PID, la sua *Risposta a Gradino* in *catena-chiusa* presenta un lieve *Overshoot*, che rientra comunque nei parametri di progetto. In questo caso, dalla simulazione di un disturbo mediante Simulink, si è potuto notare come il controllore fosse molto performante nella *Reiezione dei Disturbi*.

Mediante l'utilizzo dello *Studio in Frequenza* si è riscontrato che il controllore che risolve il problema risulta essere, ancora una volta, una Rete Ritardatrice (*Lag*). In questo caso, però, la *Risposta a Gradino* in *catena-chiusa* non presenta fenomeni di *Overshoot*. Dalla simulazione di un disturbo, mediante Simulink, si è potuto notare, un forte incremento del tempo di risposta  $T_r \approx 25$  secondi. Lo *Steady State error* veniva raggiunto e rispettato, ma in un tempo  $T_s \approx 100$  secondi.

Posso concludere, in base alle considerazioni fatte in precedenza, che la scelta migliore per risolvere il problema del controllo, risulta ricadere sulla *Rete Lag* ottenuta con il metodo del *Luogo delle Radici*.