

UNIVERSITÀ DEGLI STUDI DI PADOVA

FACOLTÀ DI INGEGNERIA

CORSO DI LAUREA IN INGEGNERIA DELL'INFORMAZIONE

TESI DI LAUREA TRIENNALE

---

**Studio in ambiente  
Matlab-Simulink del sistema di  
controllo di posizione del motore  
elettrico in corrente continua**

---

*Relatore:*  
Prof. Alessandro Beghi

*Autore:*  
Nicola Geromel

febbraio 2012



*Ringrazio la mia famiglia,  
per avermi permesso di arrivare a questo traguardo;*

*Ringrazio la mia ragazza per essere sempre al mio fianco;*

*Ringrazio gli amici che mi sono sempre stati vicini.*



# Indice

<b>Introduzione</b>	<b>IX</b>
<b>1 Modello del motore elettrico in corrente continua</b>	<b>1</b>
1.1 Cenni sul funzionamento . . . . .	1
1.2 Configurazione fisica . . . . .	2
1.3 Derivazione del modello matematico . . . . .	3
1.3.1 Funzione di trasferimento . . . . .	5
1.3.2 Spazio di stato . . . . .	6
1.3.3 Impostazione delle specifiche di progetto . . . . .	6
<b>2 Simulazioni delle risposte in catena aperta</b>	<b>9</b>
2.1 Simulazione in ambiente MatLab e risposte indiciali . . . . .	9
2.1.1 Risposta all' impulso . . . . .	10
2.1.2 Risposta al gradino unitario . . . . .	10
2.1.3 Risposta alla rampa . . . . .	11
2.1.4 Risposta alla sinusoide . . . . .	14
2.2 Simulazione in ambiente Simulink . . . . .	16
2.2.1 Commenti finali . . . . .	19
<b>3 Progetto del controllore</b>	<b>21</b>
3.1 Controllore PID . . . . .	21
3.1.1 Controllore proporzionale P . . . . .	24
3.1.2 Controllore proporzionale-derivativo PD . . . . .	28
3.1.3 Controllore proporzionale - integrativo PI . . . . .	34
3.2 Progetto del controllore tramite il luogo delle radici . . . . .	37
3.2.1 Polinomio completo . . . . .	38
3.2.2 Approssimazione dei poli dominanti . . . . .	43
3.3 Progetto del controllore tramite sintesi in frequenza . . . . .	49
3.3.1 Errore a regime . . . . .	50
3.3.2 Sovraelongazione . . . . .	51
3.3.3 Pulsazione di attraversamento . . . . .	51

## INDICE

---

3.3.4	Determinazione della rete corretttrice necessaria . . . . .	52
3.3.5	Rete corretttrice amplificatrice e anticipatrice . . . . .	54
<b>4</b>	<b>Analisi dinamica del sistema in catena chiusa</b>	<b>59</b>
4.1	Revisione del controllore PD - introduzione del PID . . . . .	59
4.2	Risposte indiciali in catena chiusa . . . . .	64
4.2.1	Risposta all' impulso . . . . .	64
4.2.2	Risposta alla rampa . . . . .	64
4.2.3	Risposta alla sinusoide . . . . .	66
4.3	Disturbi entranti nel sistema in punti diversi . . . . .	67
	<b>Bibliografia</b>	<b>I</b>

# Elenco delle figure

1.1	Motore elettrico in corrente continua . . . . .	2
1.2	Schema elettrico del motore in corrente continua . . . . .	2
1.3	Esempio di tempo di assestamento. . . . .	7
1.4	Esempio di overshoot. . . . .	7
2.1	Schema generale del motore in configurazione di catena aperta. . . . .	9
2.2	Risposta all' impulso in catena aperta. . . . .	11
2.3	Risposta al gradino in catena aperta. . . . .	12
2.4	Risposta alla rampa unitaria in catena aperta. . . . .	13
2.5	Confronto tra risposte del sistema a rampe con diverse pendenze. . . . .	14
2.6	Risposta alla sinusoide del sistema in catena aperta. . . . .	15
2.7	Rappresentazione in Simulink del motore elettrico DC. . . . .	16
2.8	Implementazione Simulink del sistema in catena aperta dopo l'uso di <i>create subsystem</i> . . . . .	17
2.9	Risposta del motore al gradino simulata con Simulink. . . . .	17
2.10	Risposta del motore alla rampa simulata con Simulink. . . . .	18
2.11	Risposta del motore alla sinusoide simulata con Simulink. . . . .	18
3.1	Schema generale del controllore PID. . . . .	22
3.2	Risposta al gradino con controllore proporzionale con $K_P = 2$ . . . . .	25
3.3	Confronto tra risposte al gradino ottenute da un controllore PID puramente proporzionale, con $K_P$ variabile. . . . .	26
3.4	Confronto tra risposte al gradino ottenute da un controllore PID puramente proporzionale, con $1 \leq K_P \leq 5$ . . . . .	27
3.5	Analisi del funzionamento del contributo derivativo. . . . .	28
3.6	Risposta del sistema con controllore PD, con $K_P = 1$ e $K_D$ variabile. . . . .	30
3.7	Risposta del sistema con controllore PD, con $K_P = 10$ e $K_D$ variabile. . . . .	31
3.8	Risposta del sistema con controllore PD, con $K_P = 100$ e $K_D$ variabile. . . . .	32

## ELENCO DELLE FIGURE

---

3.9	Dettaglio della risposta del sistema con controllore PD, con $K_P = 70$ e $0,2 \leq K_D \leq 1$ .	32
3.10	Risposta del sistema col controllore PD scelto.	33
3.11	Risposta del sistema con controllore PI, con $K_P = 1,5$ e $K_I$ variabile.	35
3.12	Risposta del sistema con controllore PI, con $K_P = 2,5$ e $K_I$ variabile.	36
3.13	Risposta del sistema con controllore PI, con $K_P = 3$ e $K_I$ variabile.	36
3.14	Configurazione del sistema per lo studio con il luogo delle radici.	37
3.15	Luogo delle radici del motore	38
3.16	Impostazione delle specifiche con Sisotool.	40
3.17	Aggiunta degli zeri al luogo con Sisotool.	41
3.18	Scelta del guadagno con Sisotool.	41
3.19	Risposta del sistema con il controllore trovato attraverso il luogo delle radici completo.	42
3.20	Luogo delle radici del motore.	44
3.21	Luogo delle radici del motore con i soli poli dominanti.	44
3.22	Imposizione delle specifiche di controllo nel luogo delle radici con i soli poli dominanti.	46
3.23	Risposta del motore (con i soli poli dominanti) con un compensatore proporzionale con guadagno pari a $K = 1,5$ .	47
3.24	Luogo delle radici del motore (con tutti i poli) con compensatore.	48
3.25	Risposta del motore a un gradino di tensione con il compensatore (3.17).	48
3.26	Configurazione del sistema usata per la sintesi in frequenza.	49
3.27	Diagrammi di Bode di modulo e fase di $\hat{W}(j\omega)$ .	53
3.28	Risposta del sistema con rete amplificatrice e anticipatrice.	56
3.29	Risposta del sistema con controllore determinato in frequenza.	57
3.30	Diagrammi di Bode del sistema controllato con il compensatore progettato in frequenza.	58
4.1	Schema a blocchi della configurazione di controllo con la presenza del disturbo.	59
4.2	Implementazione in Simulink del motore elettrico in DC.	61
4.3	Subsystem creato con Simulink.	61
4.4	Schema Simulink del sistema del motore con controllore PD.	62
4.5	Risposta del sistema con il controllore PD e un disturbo $T(t) = 0,1 Nm$ .	62



## ELENCO DELLE FIGURE

---

4.6	Risposta del sistema con il controllore PD e un disturbo $T(t) = 1 Nm$ . . . . .	63
4.7	Risposta del sistema con il controllore PID e un disturbo $T(t) = 0,1 Nm$ . . . . .	63
4.8	Sistema in catena chiusa con una rampa unitaria come input. . . . .	64
4.9	Sistema in catena chiusa con una rampa unitaria come input. . . . .	65
4.10	Sistema in catena chiusa con una sinusoide come input. . . . .	66
4.11	Risposta del sistema in catena aperta ad una sinusoide di frequenza $10 \frac{rad}{s}$ . . . . .	66
4.12	Schema Simulink contenente disturbi entranti in punti diversi della configurazione. . . . .	67
4.13	Risposta con $D1 = 0.1 Nm$ e $D2$ nullo. . . . .	68
4.14	Risposta con $D1$ nullo e $D2 = 0.1$ . . . . .	68



# Elenco delle tabelle

3.1	Effetti dell'inserimento nel luogo delle radici di poli e zeri. . . .	47
3.2	Scelta della rete corretttrice in base ai parametri $C$ e $\Delta\varphi$ . . . .	54



# Introduzione

Il concetto di automazione, nei nostri giorni, è radicato in profondità in moltissimi settori della produzione industriale e non. Esistono svariate applicazioni, anche tecnologicamente molto avanzate, a riguardo: sono il frutto di una ricerca nel settore molto intensa. L'ambito di applicazione è così vasto da avere dei confini sfumati, che si perdono nella collaborazione dell'automazione con altre branche della scienza: elettronica, meccanica, informatica, solo per citarne alcune.

Uno degli elementi fondamentali nell'automazione, e in particolare nei controlli automatici, è l'**attuatore**. Un attuatore è un dispositivo in grado di convertire un segnale di input (tipicamente elettrico) in un movimento di qualche tipo. Questo permette quindi di far compiere determinati movimenti a degli oggetti utilizzando dei segnali elettrici come comandi. Uno tra gli attuatori più diffusi è sicuramente il motore elettrico in corrente continua. Un dispositivo di questo genere, è quindi in grado di far ruotare il proprio albero di un certo angolo, e quindi di muovere un qualche organo meccanico che vi è attaccato. L'oggetto di questa tesina è proprio l'analisi e la simulazione di alcuni metodi matematici per il controllo di posizione di tale dispositivo (inteso come controllo dell'angolo di rotazione dell'albero del motore).



# Capitolo 1

## Modello del motore elettrico in corrente continua

In questo capitolo verrà spiegato il funzionamento del motore elettrico alimentato in corrente continua e sarà eseguita un'analisi con lo scopo di costruirne il modello matematico associato al fine di studiare il comportamento di tale dispositivo. Lo studio del modello matematico è il primo passo, essenziale per poter trattare poi il problema del controllo di posizione con i metodi classici dei controlli automatici.

### 1.1 Cenni sul funzionamento

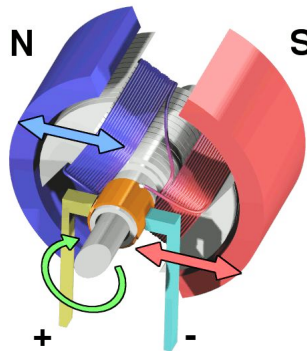
Il funzionamento del motore elettrico in corrente continua si basa sostanzialmente sulla combinazione tra alcuni principi dell'elettromagnetismo e della meccanica.

Si può pensare di suddividerlo in due parti: la parte statorica e quella rotorica. In figura 1.1 si notano le due parti: i due gusci esterni di materiale magnetico costituiscono lo statore mentre il nucleo centrale è il rotore.

Lo statore è sede dei due magneti permanenti che provocano un campo magnetico costante che agisce sul rotore. Si può immaginare quest'ultimo come un cilindro costituito da un numero elevato di spire di filo conduttore avvolte attorno ad un nucleo di materiale dielettrico, i cui estremi fuoriescono dal rotore stesso e sono collegati al circuito di eccitazione, chiamato *armatura*<sup>1</sup>. In figura si vedono chiaramente le due parti. L'armatura è costituita dalla serie tra una resistenza e un'induttanza. Ai capi di questo circuito viene applicata la tensione di ingresso, la quale è l'input che pilota l'intero

---

<sup>1</sup>In figura 1.1 si vedono solamente i due fili che fuoriescono dal rotore. L'armatura è il circuito che viene attaccato proprio a tali fili.

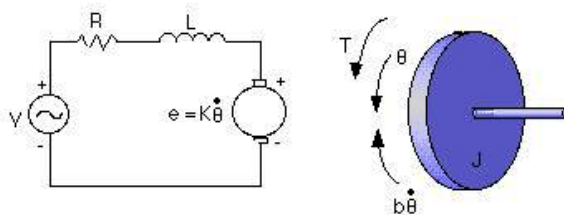


**Figura 1.1:** Motore elettrico in corrente continua

sistema. La combinazione tra il campo magnetico dello statore e quello generato dal passaggio della corrente nelle spire conduttrici del rotore dà luogo ad una interazione di forze che determina la rotazione del rotore stesso. Una descrizione più dettagliata del funzionamento del motore non è tra gli scopi di questa tesi. Le principali equazioni funzionali sono tuttavia trattate nella sezione (1.2).

## 1.2 Configurazione fisica

Come già detto, il motore elettrico in corrente continua viene spesso utilizzato come attuttore nell'ambito del controllo di sistemi meccanici. Lo schema funzionale che verrà utilizzato nell'analisi in seguito è quello riportato nella figura 1.2.



**Figura 1.2:** Schema elettrico del motore in corrente continua

Per continuare l'analisi nei paragrafi successivi, è necessario quantificare i parametri caratteristici di questo schema. Ecco allora l'elenco completo dei



valore dei parametri che saranno usati in seguito:

- Momento d'inerzia del rotore ( $J$ ):  $3,2284 \times 10^6 \frac{kgm^2}{s^2}$  ;
- Coefficiente di attrito del rotore ( $b$ ):  $3.5077 \times 10^{-6} Nms$ ;
- Costante di forza elettromotrice ( $K = K_e = K_m$ ):  $0,0274 \frac{Nm}{A}$  ;
- Resistenza ( $R$ ):  $4 \Omega$  ;
- Induttanza ( $L$ ):  $2,75 \times 10^{-6} H$  .

Si assume inoltre che sia il rotore che l'albero motore siano rigidi e indeformabili.

### 1.3 Derivazione del modello matematico

Come già detto nella sezione (1.1), c'è un legame matematico tra la tensione fornita in ingresso all'armatura e la velocità di rotazione dell'albero del motore. Chiaramente, a mano a mano che aumenta la tensione  $v(t)$  aumenta anche la corrente  $i(t)$  che scorre nel circuito. Ciò comporta un aumento anche della coppia  $C(t)$  all'albero del motore: le due quantità sono dunque proporzionali e la costante di proporzionalità è  $K_t$ , chiamata *costante di armatura*. Si ha dunque la relazione:

$$C(t) = K_t i(t)$$

La proporzionalità tra queste quantità è dovuta al fatto che a mano a mano che aumenta la corrente che scorre nelle spire del rotore, aumenta la forza magnetica generata: l'aumento di questa forza provoca quindi un aumento della coppia  $C(t)$  che fa girare il rotore. Dalla relazione  $\vec{F} = \vec{v} \wedge \vec{B}q$  si nota che l'aumento dell'intensità di corrente determina un aumento della velocità  $\vec{v}$  degli elettroni nelle spire, da cui si ha un aumento della forza  $\vec{F}$  sviluppatasi in tale processo. Le quantità  $\vec{B}$  e  $q$  rappresentano rispettivamente il campo magnetico dovuto allo statore e la carica elementare dell'elettrone, mentre  $\wedge$  è l'operatore prodotto vettoriale. L'altro legame fondamentale per questa analisi è quello tra la velocità di rotazione dell'albero del motore  $\dot{\theta}(t)$ <sup>2</sup> e la *forza contro elettromotrice* (FCEM)  $e(t)$  che si sviluppa nella parte rotorica. L'origine della FCEM va ricercata nella componente ortogonale alla velocità  $\vec{v}$  della particella. Si pensi alla traiettoria percorsa dall'elettrone durante il

---

<sup>2</sup>Chiaramente  $\dot{\theta}(t)$  è la velocità angolare, cioè la derivata della posizione angolare  $\theta(t)$ . Basta mettere un blocco integratore, quindi, all'uscita del sistema per ottenere la quantità cercata.

movimento: è un percorso elicoidale dato che è la somma di un moto rettilineo e uno rotatorio. In ogni istante la velocità di tale elettrone è scomponibile in due componenti, una tangenziale alla spira e una normale alla spira stessa. La presenza di questo secondo contributo di velocità genera una componente ortogonale della forza  $\vec{F}$  che tende a frenare gli elettroni e quindi a far passare meno corrente, rendendo la tensione effettiva inferiore a quella applicata. Anche in questo caso, quindi, le due quantità sono tra loro proporzionali. La costante di proporzionalità è  $K_e$ , detta *costante di forza controelettromotrice*. La relazione è la seguente:

$$e(t) = K_e \dot{\theta}(t)$$

Per semplificare l'analisi si assume che avvenga la conservazione delle potenze, ovvero che la potenza elettrica fornita  $P_e(t) = e(t)i(t)$  sia uguale alla potenza meccanica erogata  $P_m(t) = C(t)\dot{\theta}(t)$ , dove  $C(t)$  è stata definita prima e  $\dot{\theta}(t)$  è la velocità angolare di rotazione dell'albero motore. In questo contesto, di assenza di dispersione della potenza, si può allora scrivere

$$P_e(t) = P_m(t),$$

ovvero

$$e(t)i(t) = C(t)\dot{\theta}(t).$$

Da ciò segue che

$$K_e \dot{\theta}(t)i(t) = K_t i(t)\dot{\theta}(t),$$

per cui possiamo concludere che  $K_e = K_m = K$  in condizioni di conservazione della potenza. Questa semplificazione è utile nel senso che aiuta nell'analisi seguente. Bisogna però ricordare che è un risultato puramente teorico: nella realtà, dispersioni dovute all'effetto Joule e ad attriti interni sono sempre presenti.

Usando la legge di Kirchoff si scrive l'equazione della tensione tra l'armatura e la spira:

$$v(t) = L \frac{di}{dt}(t) + Ri(t) + e(t)$$

dove  $e(t)$  è appunto la forza controelettromotrice che causa una caduta di tensione nella spira,  $L$  e  $R$  sono rispettivamente l'induttanza e la resistenza del circuito di armatura.

Per quanto riguarda la parte meccanica del motore, usando il secondo principio della dinamica si può scrivere:

$$C(t) = Ki(t) = J\ddot{\theta}(t) + b\dot{\theta}(t)$$

in cui  $J$  è il momento di inerzia del rotore e  $b$  è la costante di attrito viscoso del rotore. Questa equazione esprime il fatto che la *coppia motrice* erogata

dal motore lavora contro il termine inerziale del motore stesso e a quello relativo all' attrito.

Mettendo insieme le due equazioni trovate si ottiene il sistema:

$$\begin{cases} J\ddot{\theta}(t) + b\dot{\theta}(t) = Ki(t) \\ L\frac{di}{dt}(t) + Ri(t) = v(t) - K\dot{\theta}(t) \end{cases} .$$

### 1.3.1 Funzione di trasferimento

Usando le trasformate di Laplace si arriva ad una rappresentazione nel dominio della frequenza del problema. Ricordando che la trasformata di  $\frac{dx}{dt}(t)$  è  $sX(s)$  si può scrivere:

$$\begin{cases} Js^2\Theta(s) + bs\Theta(s) = KI(s) \\ LsI(s) + RI(s) = V(s) - Ks\Theta(s) \end{cases} .$$

L'obbiettivo è quello di arrivare alla funzione di trasferimento tra l'ingresso del sistema  $V(s)$  e la sua uscita  $\Theta(s)$ . Raccogliendo  $s$  nel primo membro della prima equazione e  $I(s)$  nel primo membro della seconda si ottiene:

$$\begin{cases} s(Js + b)\Theta(s) = KI(s) \\ I(s)(Ls + R) = V(s) - Ks\Theta(s) \end{cases} .$$

Esplicitando  $I(s)$  dalla prima equazione, si ottiene  $I(s) = \frac{s(Js+b)\Theta(s)}{K}$ ; si sostituisce  $I(s)$  nella seconda equazione e raccogliendo  $\Theta(s)$  al primo membro risulta:

$$\Theta(s) \left( \frac{s(Ls + R)(Js + b)}{K} + sK \right) = V(s).$$

A questo punto, poiché a noi interessa la funzione di trasferimento ingresso-uscita del sistema, si calcola il rapporto tra  $\Theta(s)$  e  $V(s)$ , ottenendo così la f.d.t. cercata:

$$\frac{\Theta(s)}{V(s)} = \frac{K}{sK^2 + s(Ls + R)(Js + b)}. \quad (1.1)$$

Da questa scrittura si nota come il sistema abbia un polo nell'origine e il denominatore sia di terzo grado. Si può riscrivere la funzione di trasferimento sviluppando competamente il denominatore in modo da mettere in evidenza i coefficienti dei vari termini in  $s$ . Dopo qualche passaggio si ottiene quindi:

$$\frac{\Theta(s)}{V(s)} = \frac{K}{LJs^3 + (Lb + RJ)s^2 + (Rb + K^2)s}.$$

Questa scrittura sarà utile in seguito, quando si avranno bisogno proprio dei coefficienti dei termini in  $s$  per la simulazione con Simulink.

### 1.3.2 Spazio di stato

Una rappresentazione matriciale del sistema è utile in questo caso perchè è proprio quella necessaria per poter eseguire delle simulazioni con MatLab. Si fornisce quindi di seguito questa rappresentazione:

$$\frac{d}{dt} \begin{bmatrix} \theta \\ \dot{\theta} \\ i \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\frac{b}{J} & \frac{K}{J} \\ 0 & -\frac{K}{L} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ i \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{L} \end{bmatrix} V$$

$$y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ i \end{bmatrix}.$$

### 1.3.3 Impostazione delle specifiche di progetto

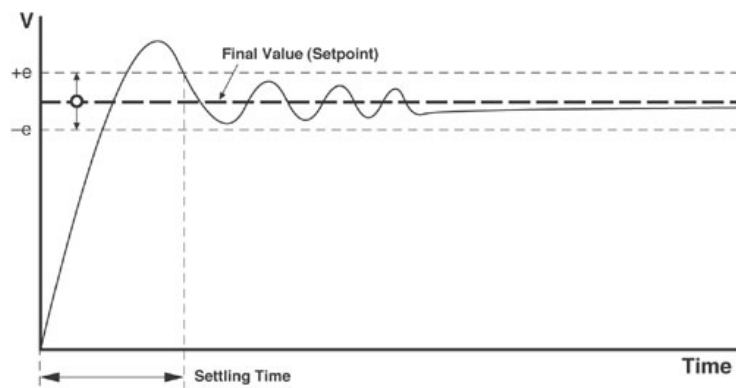
I vincoli di prestazioni imposti dall'utilizzo futuro di un tale sistema si traducono nelle cosiddette *specifiche di progetto*. In parole povere, la velocità di spostamento, la precisione nel posizionamento e la capacità di mantenere le prestazioni anche nel caso di disturbi esterni sono caratteristiche che si ricercano nel prodotto finito, e che a monte della fase di progettazione vengono interpretate in termini di controllo. Nello caso particolare di questo motore, le specifiche imposte sono le seguenti:

- tempo di assestamento minore di 40 ms;
- overshoot inferiore al 16 %;
- steady-state errore nullo;
- steady-state error dovuto a disturbi nullo.

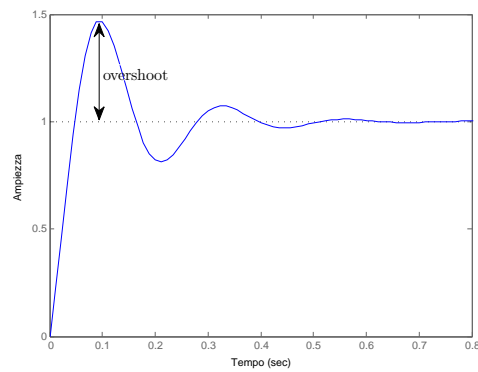
Un breve commento a questi requisiti: il **tempo di assestamento** è definito come il tempo impiegato da un dato segnale per portarsi definitivamente (cioè senza uscirne mai più) dentro ad un intorno (di dimensioni arbitrarie, ma sufficientemente piccole) del valore che deve assumere a regime: un esempio lo si vede in figura 1.3:

l'overshoot è la massima sovraelongazione che compie il segnale, durante la fase transitoria, al di sopra del valore che deve assumere a regime (un esempio lo si vede in figura 1.4, in cui il segnale sale di quasi il 50 % sopra al livello di regime che dovrebbe essere 1);

lo steady-state error (errore a regime) è la differenza tra il valore che il segnale avrebbe dovuto assumere nelle intenzioni del progettista e quello che



**Figura 1.3:** Esempio di tempo di assestamento.



**Figura 1.4:** Esempio di overshoot.

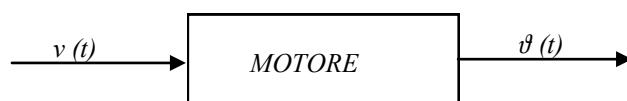
invece in realtà raggiunge una volta terminata la fase transitoria; lo steady-state error dovuto agli errori, infine, è l'errore che a regime si verifica sul segnale a causa di disturbi agenti sul sistema.



## Capitolo 2

# Simulazioni delle risposte in catena aperta

MatLab è un ambiente matematico che si presta molto bene alla simulazione di sistemi di controllo. In questo caso, lo si utilizza appunto per verificare il comportamento del motore in risposta ad alcuni input. Nella sezione (1.3.1) è stata calcolata la funzione di trasferimento (in catena aperta) del sistema, tra la tensione in ingresso all'armatura  $V(t)$  e la posizione angolare dell'albero  $\theta(t)$ . Nota quindi la funzione di trasferimento (3.13), con MatLab si riesce a vedere il comportamento del motore a seguito di un segnale di tensione come input. L'oggetto di questo capitolo è proprio quello di verificare il comportamento del motore in configurazione di catena aperta utilizzando come ingresso alcuni segnali notevoli. La schema a blocchi è riportato in figura 2.1.



**Figura 2.1:** Schema generale del motore in configurazione di catena aperta.

### 2.1 Simulazione in ambiente MatLab e risposte indiciali

Le risposte indiciali sono la risposta del sistema a segnali notevoli come input: i segnali che verranno analizzati sono l'impulso, il gradino unitario, la rampa e la sinusoidale.

Il primo passo è la definizione in Matlab delle variabili del motore e la creazione della FdT tra tensione e posizione. Questo è quello che fa il codice seguente:

```
J=3.2284E-6;  
b=3.5077E-6;  
K=0.0274;  
R=4;  
L=2.75E-6;  
numeratore=K;  
denominatore=[(J*L) ((J*R)+(L*b)) ((b*R)+K^2) 0];  
motore=tf(numeratore,denominatore);
```

La Fdt è costruita a partire dal numeratore e dal denominatore presi separatamente: questi vengono creati come dei vettori di coefficienti, in cui l'ordine determina il grado di ciascuno di essi nel polinomio corrispondente. Il comando *tf()* crea la FdT desiderata in base ai parametri fornitigli.

### 2.1.1 Risposta all' impulso

Il primo segnale che si prova a dare in ingressi al motore è un impulso di tensione. Per generarlo si usa il codice

```
impulse(motore)
```

con il quale si traccia la risposta del sistema *motore* a tale ingresso. L'esito della simulazione è riportato in figura 2.2.

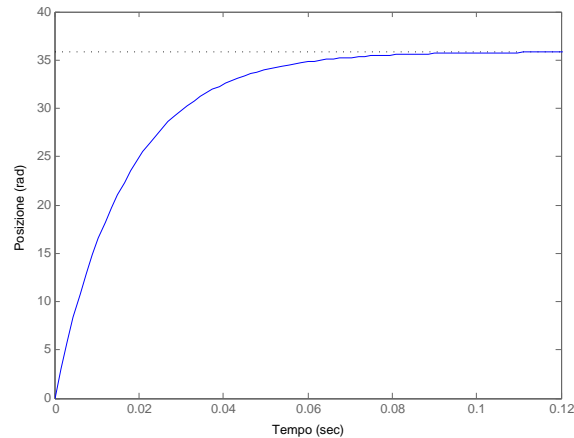
In questa simulazione l'impulso di tensione in ingresso mette in movimento il motore, ma questo dura per un tempo breve, al termine del quale il motore si ferma ma in una posizione diversa rispetto a quella iniziale chiaramente. Come si vede nel grafico, il movimento del motore dura circa 0,1 s: passato questo tempo l'albero è ruotato di poco più di 35 *rad*, ovvero ha fatto oltre 5 giri.

### 2.1.2 Risposta al gradino unitario

Per simulare il comportamento in catena aperta del motore con MatLab si utilizza il seguente codice:

```
t=(0:0.001:0.2);  
step(motor,t)
```





**Figura 2.2:** Risposta all' impulso in catena aperta.

in cui prima si definisce il vettore che determina gli istanti di tempo in cui calcolare la risposta, definendo durata (0, 2 s) e passo (0, 001 s), poi si procede a tracciare la risposta al gradino unitario attraverso il comando `step()`. Gli altri comandi

```
xlabel('Tempo');
ylabel('Posizione (rad)');
title(' ');
```

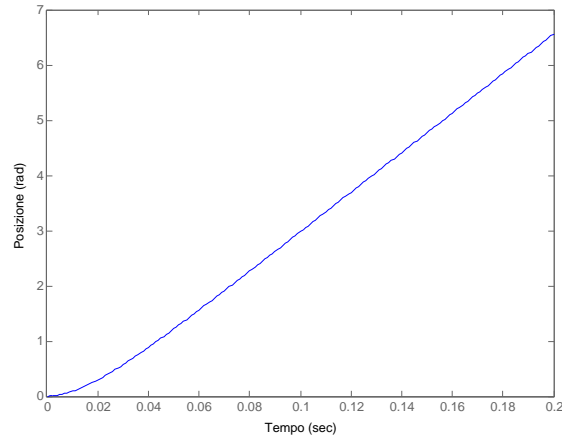
servono a impostare alcuni parametri del grafico.

La risposta è raffigurata in figura 2.3.

Dalla figura si nota che sotto l'input di una tensione costante pari a  $v(t) = 1V$  il rotore inizia a ruotare e dopo 0,2 s si è già spostato di circa 6,5 rad. La spiegazione fisica di questo comportamento va ricercata nel fatto che una tensione costante in ingresso implica una corrente costante nelle spire del rotore e dunque una forza che lo fa girare a **velocità costante**. Dopo un transitorio iniziale, infatti, si vede che la posizione angolare varia in maniera lineare.

### 2.1.3 Risposta alla rampa

Il terzo segnale notevole per il quale si simula il comportamento del sistema in catena aperta è la rampa. Il codice MatLab è il seguente:



**Figura 2.3:** Risposta al gradino in catena aperta.

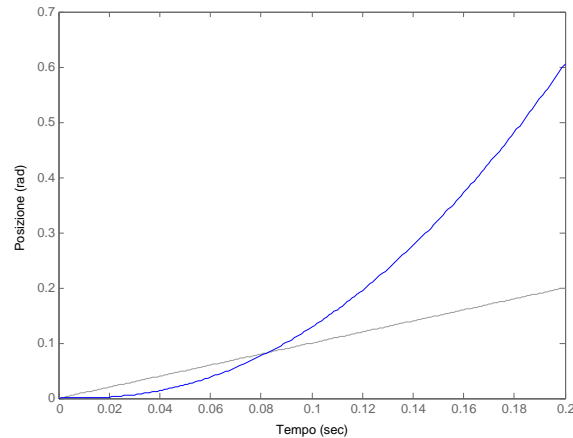
```
t=(0:0.001:0.2);
u=t;
lsim(motore,u,t);
```

Rispetto al codice usato per disegnare la risposta al gradino, le differenze sono poche. Qui si è utilizzato il comando *lsim()* che simula la risposta nel tempo del sistema che gli è passato come argomento (in questo caso il motore), utilizzando come ingresso il segnale *u* (che in questo caso è una rampa a pendenza unitaria), mentre il parametro *t* determina le caratteristiche della variabile temporale, come nel caso precedente. L'output di questo codice è il grafico riportato in figura 2.4.

In figura ci sono due linee: quella blu rappresenta l'andamento della risposta del sistema, mentre quella grigia è la rampa con pendenza unitaria. Come nella risposta al gradino, anche in questa si vede che il sistema non è stabile, perché la posizione angolare dell'albero motore continua ad aumentare, ovvero l'albero continua a girare senza mai fermarsi. Bisogna notare che un aumento lineare della tensione in ingresso, comporta un aumento lineare anche della velocità di rotazione dell'albero e dunque un andamento di tipo quadratico della posizione.

Al variare della pendenza della rampa chiaramente si ottengono risposte diverse. Con il codice MatLab seguente si riesce a fare un confronto tra rampe diverse in input:

```
PVect=[0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 100];
```



**Figura 2.4:** Risposta alla rampa unitaria in catena aperta.

```

for n=1:length(PVect)
    subplot(length(PVect)/2,2,n)
    u=t*PVect(n);
    lsim(motore,u,t)
    legend(strcat('P= ',num2str(PVect(n))),'boxoff',
            'Location','NorthWest');
    hold on
end
hold off

```

L'output è un insieme di 10 grafici (10 sono le pendenze contenute nel vettore PVect) ed è mostrato in figura 2.5.

Da questi andamenti si vede che a mano a mano che aumenta la pendenza della rampa, aumenta la velocità con la quale l'albero del motore si porta in posizioni angolari sempre maggiori, ovvero ruota più velocemente. Questo risultato tuttavia era immaginabile, in quanto si era detto che c'è un legame di proporzionalità diretta tra tensione in ingresso  $v(t)$  e velocità di rotazione  $\dot{\theta}(t)$ . L'aspetto più evidente comunque è ancora una volta l'instabilità del sistema in catena aperta.

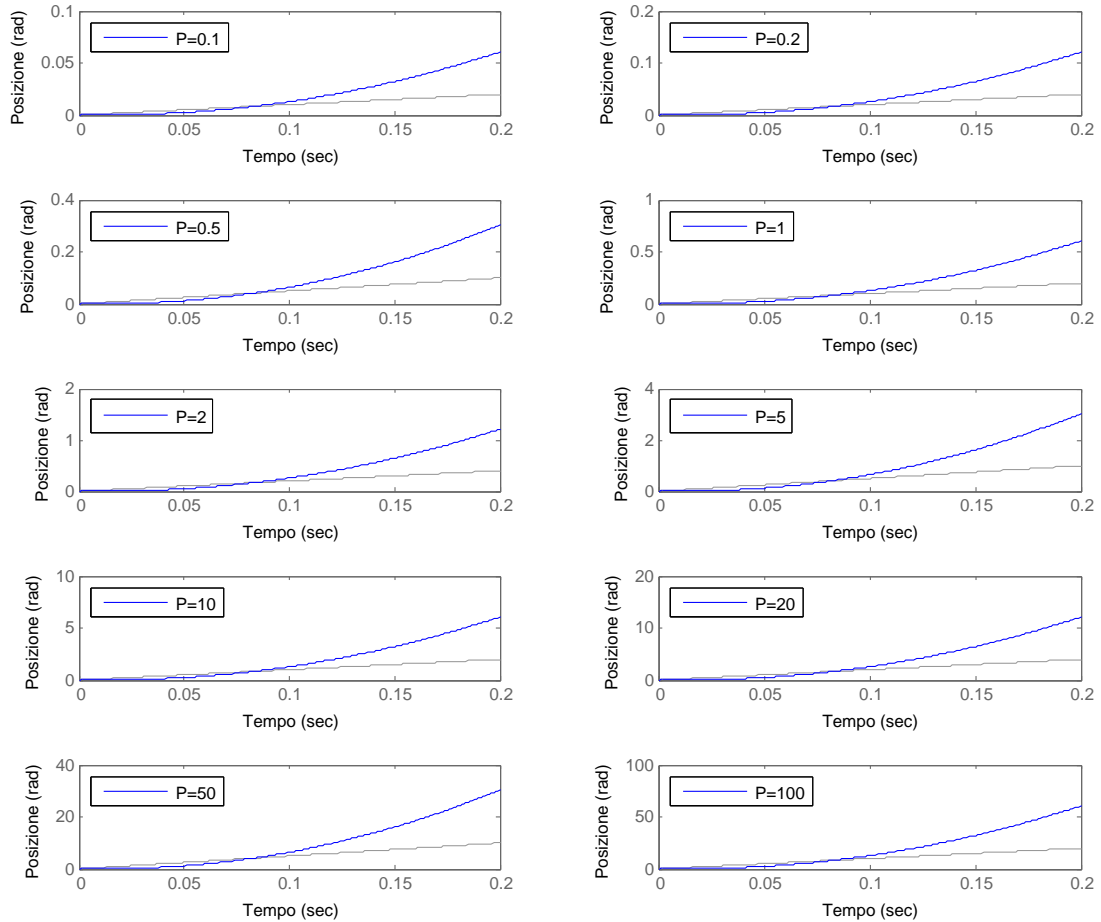


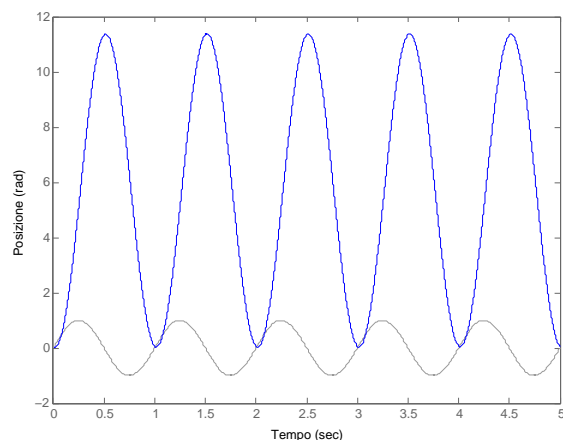
Figura 2.5: Confronto tra risposte del sistema a rampe con diverse pendenze.

### 2.1.4 Risposta alla sinusoide

L'ultimo segnale notevole che viene analizzato è la sinusoide. Un segnale di questo tipo come input, simula una tensione alternata come ingresso. Il codice MatLab da utilizzare per la simulazione è il seguente:

```
[u,t]=gensig('sin',1,5,0.001);
lsim(motore,u,t)
```

La novità introdotta in questa simulazione è il comando *gensig()* che genera un segnale sinusoidale con periodo 1 secondo e di durata 5 secondi, mentre il passo di distribuzione dei campioni temporali è 0,001 secondi (questo dato è invariato rispetto alle simulazioni precedenti).



**Figura 2.6:** Risposta alla sinusoide del sistema in catena aperta.

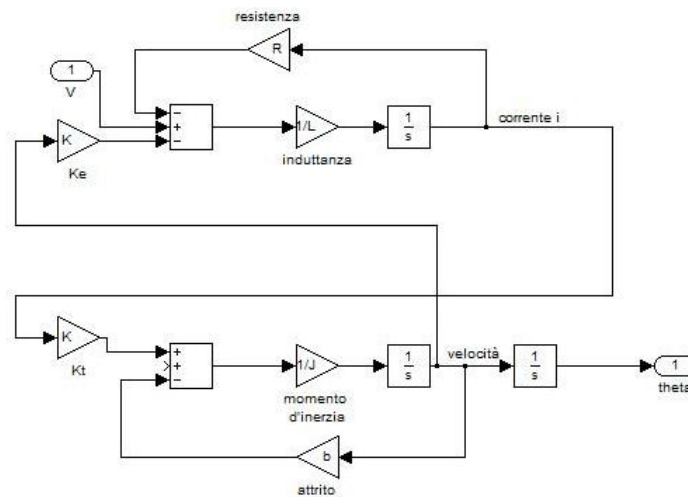
Il risultato della simulazione è la figura 2.6. Come negli altri casi, la linea blu è la risposta del sistema, mentre quella grigia è il segnale in ingresso. Con una sinusoide come segnale pilotante il comportamento del motore è particolare:  $v(t)$  oscilla nell'intervallo  $[-1, 1]$  V (circa) mentre l'albero ruota di circa 11 rad per poi tornare indietro, quando la tensione cambia di segno, fino alla posizione iniziale. Questo andamento è giustificato dal fatto che quando la tensione in ingresso cambia di segno, anche il verso di circolazione della corrente si inverte, e con esso anche quello del campo magnetico generato dalle spire. Il cambiamento di tutte queste quantità determina l'inversione anche delle forze in gioco tra statore e rotore, da cui un cambiamento del verso di rotazione dell'albero. Questo sistema non è asintoticamente stabile perchè non raggiunge mai una posizione di equilibrio, ma è limitato nel senso che la rotazione compiuta dal rotore si mantiene all'interno di un certo intervallo di posizioni angolari (qualitativamente tra gli 0 e gli 11 rad).

## 2.2 Simulazione in ambiente Simulink

Un modo alternativo per ottenere la risposta alla sinusoide del sistema in catena aperta è l'utilizzo del pacchetto *Simulink*, parte dell'ambiente MatLab. In primo luogo si crea lo schema a blocchi del sistema, composto da tre parti:

1. il blocco **sorgente**: genera il segnale di ingresso;
2. il blocco **motore**: contiene il modello matematico trovato del sistema fisico;
3. il blocco **oscilloscopio**: permette di visualizzare il risultato della simulazione.

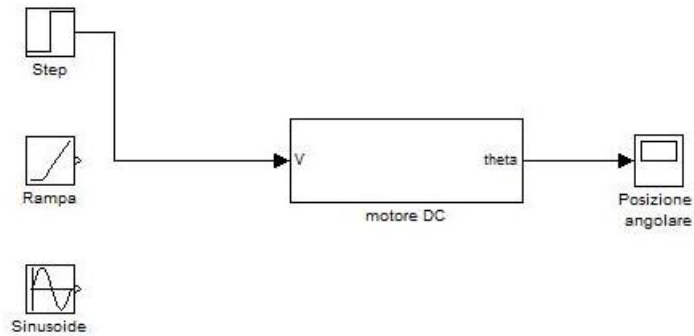
Lo schema del motore in Simulink è quello in figura 2.7.



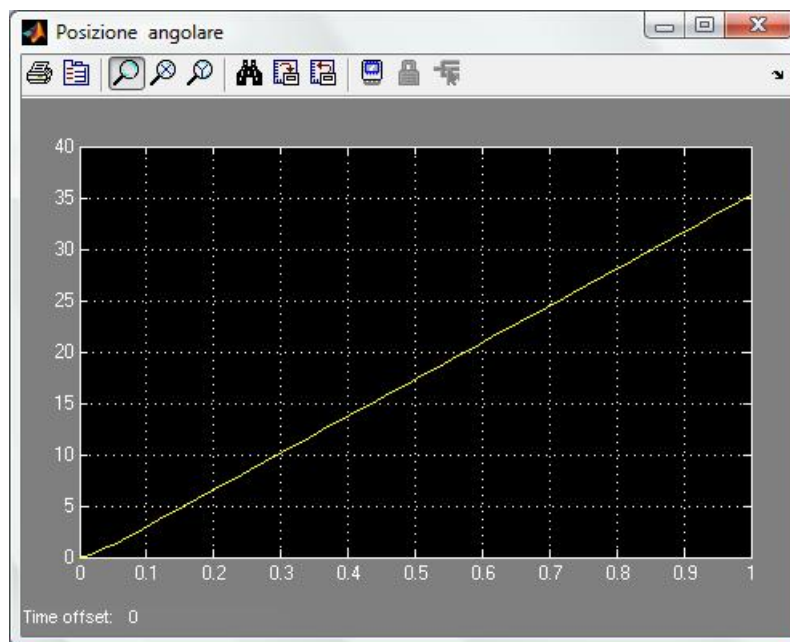
**Figura 2.7:** Rappresentazione in Simulink del motore elettrico DC.

Con la funzione *create subsystem* di Simulink è possibile compattare lo schema, mettendolo tutto dentro ad un contenitore dal quale fuoriescono la tensione di ingresso  $V$  e la posizione angolare in uscita. Il risultato è riportato in figura 2.8.

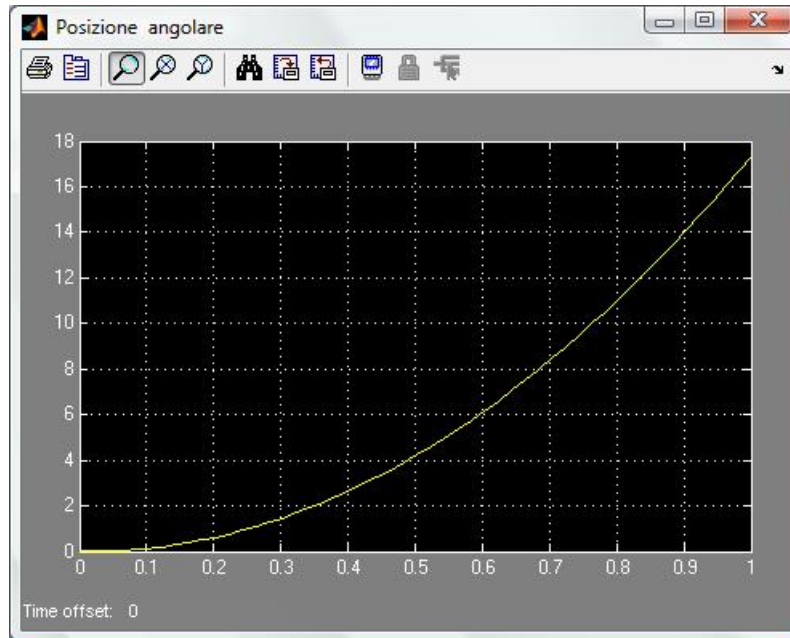
Nell'ultima figura si vede il risultato del comando: il modello del motore di figura 2.7 è contenuto nel blocco centrale, sulla sinistra si vedono i blocchetti che generano i vari segnali di input e sulla destra si trova l'oscilloscopio che permette di vedere la risposta del motore. I risultati delle simulazioni sono riportati nelle figure 2.9, 2.10 e 2.11.



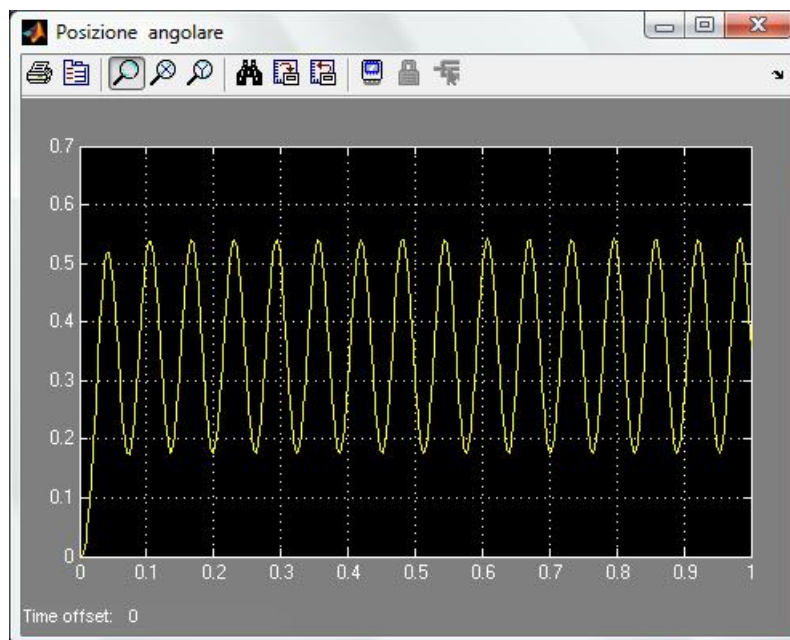
**Figura 2.8:** Implementazione Simulink del sistema in catena aperta dopo l'uso di *create subsystem*.



**Figura 2.9:** Risposta del motore al gradino simulata con Simulink.



**Figura 2.10:** Risposta del motore alla rampa simulata con Simulink.



**Figura 2.11:** Risposta del motore alla sinusoide simulata con Simulink.



### 2.2.1 Commenti finali

Dopo aver simulato il comportamento del sistema in catena aperta con diversi ingressi, la conclusione immediata è che lo schema in catena aperta in generale non è stabile e tantomeno quindi rispetta le specifiche di progetto. Parlare di rispetto dei requisiti in un sistema instabile è tuttavia banale. I tempi di assestamento non vengono rispettati perchè le risposte tendono a divergere e con esse crescono all'infinito anche gli errori <sup>1</sup>. Si rende quindi necessario l'utilizzo di schemi di controllo più complessi e, come molto spesso accade nel campo dei controllo automatici, la risposta a questi problemi è la retroazione. Il controllo in retroazione del motore è l'oggetto del prossimo capitolo.

---

<sup>1</sup>Lo steady-state error non è stato considerato in queste simulazioni perchè di disturbi si parlerà nel capitolo 4.



# Capitolo 3

## Progetto del controllore

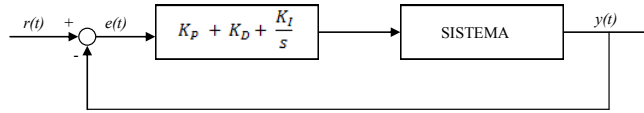
Il capitolo 2 si è concluso con la constatazione della necessità di utilizzare dei controllori in retroazione per stabilizzare il sistema, perché il semplice motore in catena aperta non è stabile. L'obbiettivo di questo capitolo è quindi quello di studiare alcuni metodi di controllo in catena chiusa per il sistema. I metodi proposti sono il **controllore PID**, il progetto tramite **studio del luogo delle radici** e quello tramite l'utilizzo delle **reti correttrici**.

Come nel capitolo precedente, è necessaria una configurazione del sistema, in cui definire i parametri del motore e le altre grandezze comuni a tutta la fase di progettazione. Il codice è quello riportato.

```
J=3.2284E-6;  
b=3.5077E-6;  
K=0.0274;  
R=4;  
L=2.75E-6;  
numeratore=K;  
denominatore=[(J*L) ((J*R)+(L*b)) ((b*R)+K^2) 0];  
motore=tf(numeratore,denominatore);
```

### 3.1 Controllore PID

La sintesi di un controllore PID è un procedimento euristico col quale si costruisce la funzione di trasferimento del controllore in un sistema in retroazione in modo da far sì che il sistema complessivo abbia prestazioni rispettose delle specifiche di progetto. Il controllore PID è la combinazione di tre funzioni di controllo, una di tipo Proporzionale, una Integrale e una Differenziale. Le tre parti di questo tipo di controllore intervengono in maniera diversa sull'errore  $e(t)$ . Lo schema generale è quello riportato in figura 3.1.



**Figura 3.1:** Schema generale del controllore PID.

Nello schema riportato,  $e(t) = r(t) - y(t)$ , dove  $r(t)$  è il segnale di riferimento (ovvero l'andamento che dovrebbe assumere l'uscita),  $e(t)$  l'errore di inseguimento e  $y(t)$  è chiaramente l'uscita. Il segnale  $e(t)$  rappresenta quindi la differenza tra l'andamento che l'uscita dovrebbe avere e quello che effettivamente ha nella realtà: essendo un errore l'obbiettivo, quindi, è quello di farlo tendere a 0 in qualche modo.

La forma generale della funzione di trasferimento del controllore PID è dunque:

$$C(s) = K_P + K_D s + \frac{K_I}{s}. \quad (3.1)$$

L'azione proporzionale (svolta dal primo termine della (3.1)) è la più semplice, e consiste nel correggere l'ingresso del sistema controllato  $u(t)$  in maniera proporzionale all'errore di inseguimento  $e(t)$ . L'effetto è quindi:

$$u_P(t) = K_P e(t). \quad (3.2)$$

L'azione integrativa invece incide sul comportamento a regime del sistema. Il termine  $\frac{K_I}{s}$  svolge un'azione integrativa dell'errore  $e(t)$  dal momento in cui il sistema è stato avviato all'istante attuale  $t$ . In questo modo mantiene memoria di quel che è accaduto nel passato e produce un contributo adeguato per l'ingresso di controllo:

$$u_I(t) = K_I \int_0^t e(\tau) d\tau. \quad (3.3)$$

L'azione derivativa infine interviene sul comportamento transitorio del sistema. Produce un contributo non nullo non appena l'errore inizia a variare, e questo permette una correzione dinamica dell'ingresso al variare dell'errore. Il risultato di questa azione di controllo è:

$$u_D(t) = K_D \frac{de}{dt}(t). \quad (3.4)$$

Questo contributo è quello più delicato, perché può portare facilmente a destabilizzare il sistema.

L'ingresso di controllo completo è dunque formato sommando queste tre azioni, ovvero :

$$u(t) = u_P(t) + u_I(t) + u_D(t) = K_P e(t) + K_I \int_0^t e(\tau) d\tau + K_D \frac{de}{dt}(t). \quad (3.5)$$

### 3.1.1 Controllore proporzionale P

Per la determinazione del contributo proporzionale, si pone  $K_I = 0$  e  $K_D = 0$ . Quello che si ottiene è uno schema retroazionato con controllo puramente proporzionale. Con l'introduzione di questo nuovo apparato di controllo, la funzione di trasferimento cambia, diventando appunto una funzione di trasferimento relativa ad un sistema in catena chiusa. La funzione di trasferimento del blocco motore  $M(s)$  è ancora quella calcolata nel capitolo 1:

$$M(s) = \frac{\Theta(s)}{V(s)} = \frac{K}{sK^2 + s(Ls + R)(Js + b)}. \quad (3.6)$$

Applicando le consuete regole dell'algebra degli schemi a blocchi, otteniamo che la funzione di trasferimento in catena chiusa  $W(s)$  è data da:

$$W(s) = \frac{C(s)M(s)}{1 + C(s)M(s)}, \quad (3.7)$$

dove  $C(s)$  è la funzione di trasferimento del blocco controllore. Sostituendo  $C(s) = K_P$  e ad  $M(s)$  l'espressione 3.13, dopo alcuni passaggi, si ottiene:

$$W(s) = \frac{K_P K}{K_P K + s(LJs^2 + (Lb + RJ)s + (Rb + K^2))}. \quad (3.8)$$

Ancora una volta ci si affida a MatLab per effettuare la simulazione. Per ottenere la funzione di trasferimento in catena chiusa si utilizza il comando `feedback()`. Il codice è il seguente:

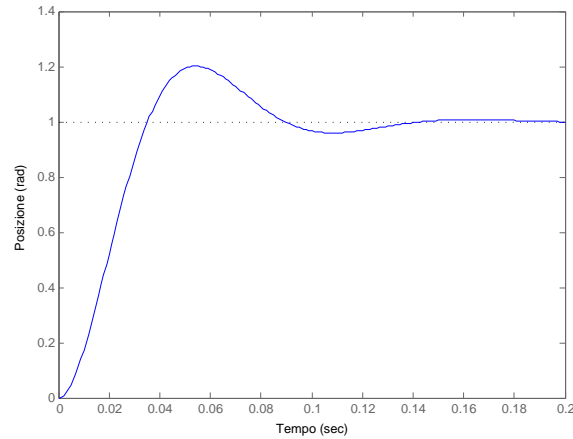
```
closed_loop_sys=feedback(controllore*motore,1);
```

Il comando `feedback()` ha come argomenti la FdT del ramo in catena diretta e quella del ramo di retroazione. Il suo output è proprio la 3.8. Una volta creata la funzione di trasferimento in retroazione è possibile verificare il comportamento del motore con un gradino unitario di tensione in input. Il codice per la simulazione è:

```
t=0:0.001:0.2;  
step(closed_loop_sys,t)
```

La risposta ottenuta è quella di figura 3.2.

Come si vede, la risposta presenta una sovraelongazione di circa 0,2 rad (che è pari a circa al 20%); anche il tempo di assestamento è superiore agli 0.04 secondi: queste specifiche dunque non sono soddisfatte. Il traguardo



**Figura 3.2:** Risposta al gradino con controllore proporzionale con  $K_P = 2$ .

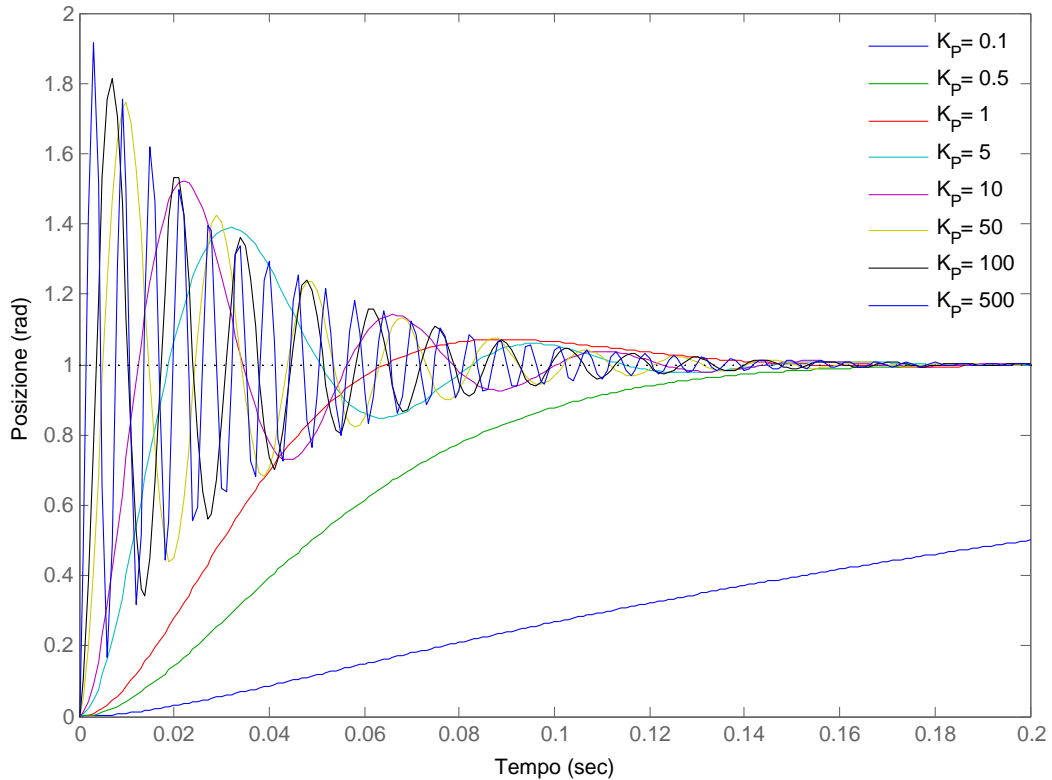
della stabilità è invece finalmente raggiunto: la risposta non è divergente al crescere di  $t$  e sembra che si stabilizzi attorno al valore 1 rad.

Per valutare il comportamento del sistema, è utile confrontare le risposte per diversi valori di  $K_P$ . Con il codice seguente si genera una serie di grafici che mostrano le risposte a variare di  $K_P$ :

```
KpVect=[0.1 0.5 1 5 10 50 100 500];
t=0:0.001:0.2;
for n=1:length(KpVect)
    controllore=KpVect(n);
    closed_loop_sys=feedback(controllore*motore,1);
    step(closed_loop_sys,t)
    xlabel('Tempo');
    ylabel('Posizione (rad)');
    title(' ');
    legend('K_{P}= 0.1', 'K_{P}= 0.5', 'K_{P}= 1', 'K_{P}= 5',
          'K_{P}= 10', 'K_{P}= 50', 'K_{P}= 100', 'K_{P}= 500');
    legend('boxoff');
    hold on
end
hold off
```

In questo script è stato creato il vettore  $KpVect$  contenente tutti i guadagni proporzionali da testare e poi prendendo un elemento alla volta del vettore è stato creato il sistema retroazionato corrispondente e simulata la sua risposta al gradino.

L'output prodotto è mostrato in figura 3.3.



**Figura 3.3:** Confronto tra risposte al gradino ottenute da un controllore PID puramente proporzionale, con  $K_P$  variabile.

In linea generale si nota un aumento del tempo di salita dell'uscita se  $K_P$  diminuisce, mentre se  $K_P$  aumenta la risposta presenta oscillazioni con frequenza e ampiezza sempre maggiori: emblematica è l'uscita per  $K_P = 500$ , in cui si evidenzia una sovralongazione di quasi il 100%.

Gli effetti generali del **contributo proporzionale** sono quindi una **diminuzione del tempo di salita**, un **aumento delle sovralongazioni**, e una **diminuzione dell'errore a regime** (rispetto ad un controllo in catena aperta).

Dato che la sintesi del controllore PID è un processo piuttosto empirico nel senso che funziona ad approssimazioni successive, dai grafici ottenuti si può dire che il controllore proporzionale da buoni risultati se  $K_P$  è maggiore di 1 ma minore di 5. In questo range le specifiche non sono assolutamente rispettate, ma il comportamento non è molto distante da quello cercato. Negli

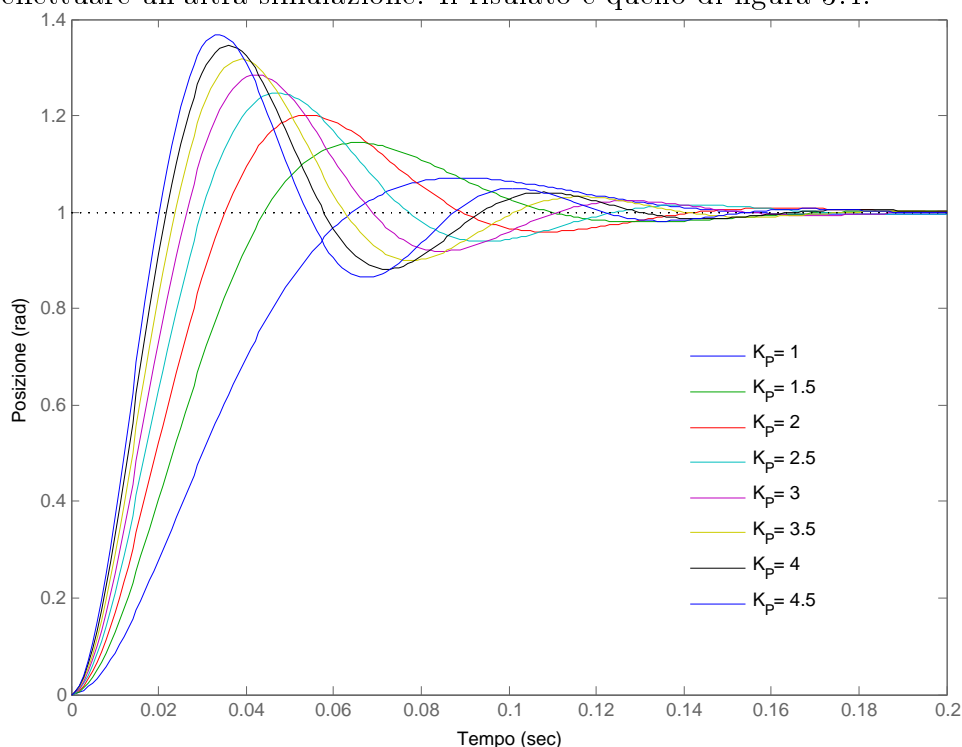


altri casi, invece, la risposta è pessima.

Per avere una visione più precisa della situazione con  $1 \leq K_P \leq 5$  si può definire il vettore di guadagni

`KpVect2=[1 1.5 2 2.5 3 3.5 4 4.5]`

ed effettuare un'altra simulazione. Il risultato è quello di figura 3.4.



**Figura 3.4:** Confronto tra risposte al gradino ottenute da un controllore PID puramente proporzionale, con  $1 \leq K_P \leq 5$ .

In questo caso si vede come per  $K_P = 1$  e  $K_P = 2$  la risposta abbia un overshoot entro il 20 % mentre nei casi con  $K_P$  maggiore la sovranelongazione sia eccessiva. Il tempo di salita cala all'aumentare del guadagno, a conferma di quanto detto prima. In ogni caso, tuttavia, la posizione asintotica raggiunta è quella corretta.

Ricapitolando, per un controllore PID puramente proporzionale, a mano a mano che aumenta il guadagno  $K_P$  aumentano la frequenza delle oscillazioni nel transitorio e l'overshoot, mentre calano il tempo di salita e l'errore. Per superare questi problemi è necessario aggiungere altre azioni sull'errore.

### 3.1.2 Controllore proporzionale-derivativo PD

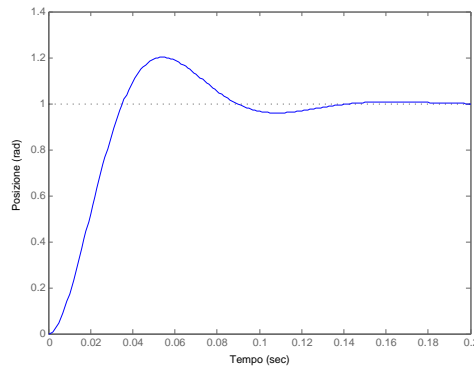
Gli esiti della simulazione del controllore puramente proporzionale non sono stati soddisfacenti. Per migliorare la situazione si introduce allora un controllore PD in cui il termine integrale è annullato ponendo  $K_I = 0$ . La funzione di trasferimento del controllore è in questo caso:

$$C(s) = K_P + K_D s. \quad (3.9)$$

Nel dominio del tempo, l'ingresso di controllo  $u(t)$  ha la forma:

$$u(t) = K_P e(t) + K_D \frac{de}{dt}(t). \quad (3.10)$$

L'effetto del termine derivativo è sostanzialmente quello di migliorare il transitorio della risposta. Per capirne il funzionamento si prenda come esempio la figura 3.5.



**Figura 3.5:** Analisi del funzionamento del contributo derivativo.

La posizione di riferimento per questo sistema è  $1 \text{ rad}$ , mentre la linea blu rappresenta l'andamento della posizione angolare del motore. Nel tratto iniziale del movimento, l'errore (cioè la distanza verticale tra il riferimento e la risposta del motore) è grande e positivo, ma sta diminuendo a mano a mano che il motore ruota (ci si avvicina al riferimento), per cui la sua derivata è negativa. Ciò implica che l'azione derivativa fornisce un contributo negativo al controllore, ovvero frena il motore mentre si sta avvicinando alla posizione corretta. Quando il motore è in posizione chiaramente il contributo è nullo. A causa della propria inerzia, tuttavia, il motore non si ferma una volta arrivato al riferimento, bensì va oltre: nel tratto ascendente della sovraelongazione, l'azione proporzionale dà un contributo sempre maggiore all'aumentare dell'errore per cercare di riportare il motore in posizione e quella derivativa fa lo stesso: dato che l'errore sta crescendo, la sua derivata

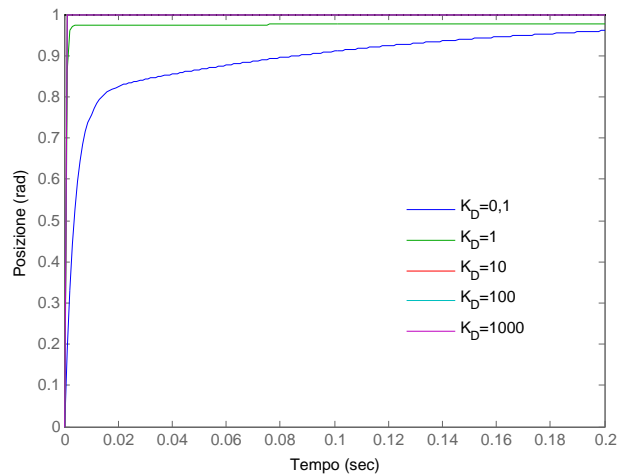
è positiva quindi dall'azione derivativa si ha un ulteriore contributo positivo per  $u(t)$ . Nel tratto discendente dell'overshoot invece la derivata è negativa e l'effetto è quello di rallentare il motore per evitare che, ancora a causa della sua inerzia, oltrepassi nuovamente il riferimento. Il risultato netto dell'azione derivativa è quindi quello di un miglioramento del comportamento nel transitorio della risposta, un'eliminazione delle sovraelongazioni ma talvolta un leggero aumento del tempo di salita. Questo è un aspetto che bisogna considerare con attenzione: dato che nella fase iniziale della risposta l'errore è grande, il contributo dell'azione derivativa è negativo (perché l'errore sta diminuendo) e proporzionale all'errore e questo può portare a rallentare il motore se il coefficiente  $K_D$  è troppo elevato in quanto si va a ridurre molto l'azione proporzionale nel segnale  $u(t)$  complessivo.

Fatte alcune considerazioni generali sul controllore PD, si passa ora alla pratica. Per questa simulazione il codice MatLab utilizzato è:

```
KpVect=[0.1 1 10 100 1000];
KdVect=[0.1 1 10 100 1000];
t=0:0.001:0.2;
for n=1:length(KpVect)
    figure(n);
    Kp=KpVect(n);
    for m=1:length(KdVect)
        controllore=tf([KdVect(m) Kp],[0 1]);
        sys=feedback(controllore*motore,1);
        step(sys,t)
        hold on
        xlabel('Tempo');
        ylabel('Posizione (rad)');
        title(' ');
        legend('K_{D}=0,1', 'K_{D}=1', 'K_{D}=10', 'K_{D}=100',
              'K_{D}=1000');
        legend('boxoff');
    end
end
```

Si riportano solo le risposte ottenute per  $K_P = 1$ ,  $K_P = 10$  e  $K_P = 100$ , nelle figure 3.6, 3.7 e 3.8.

Quello che si osserva è che il controllore PD presenta un transitorio molto breve. Per valori elevati del coefficiente derivativo  $K_D$  le prestazioni del sistema sono buone in termini di tempo di assestamento, overshoot ed errore



**Figura 3.6:** Risposta del sistema con controllore PD, con  $K_P = 1$  e  $K_D$  variabile.

a regime. A mano a mano che aumenta  $K_P$  si notano delle sovraelongazioni sempre più marcate, il che fa supporre che per questo tipo di controllore vada scelto un valore non troppo alto del coefficiente proporzionale. I benefici introdotti dal **coefficiente derivativo** sono allora una **diminuzione delle sovraelongazioni** e un **transitorio più veloce**.

Con questa prima simulazione si è verificato il comportamento generale del sistema con un controllore PD. Si nota, inoltre, che per bassi valori di  $K_P$  e  $K_D$  il sistema non raggiunge la posizione corretta, almeno in tempi brevi. Da tutto ciò si può supporre che per un controllore PD è meglio scegliere dei valori di  $K_P$  inferiori a 100 e per  $K_D$  va bene un valore non superiore a 1 o 2. Si preferisce scegliere un valore alto di  $K_P$  e molto basso di  $K_D$  per motivi di stabilità: scegliendo un valore troppo alto del coefficiente derivativo, infatti, in alcuni casi si rischia di avere comportamenti divergenti del sistema a seguito di input particolari.

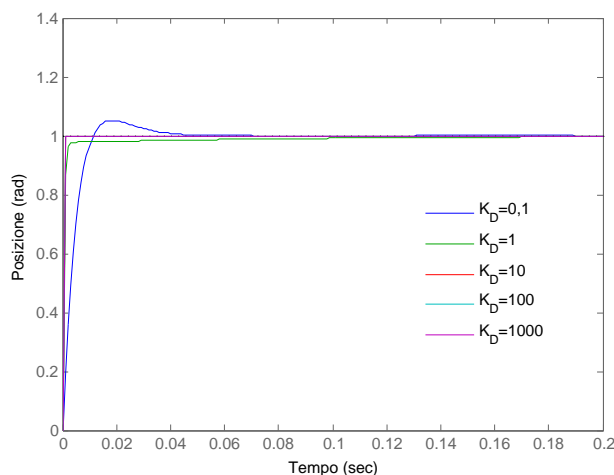
Si può testare il comportamento del motore con un controllore formato da parametri proporzionale e derivativo che stanno intorno ai valori appena indicati.

Simulando la risposta per i vettori

```
KpVect=[50 60 70 80 90 100];
KdVect=[0.2 0.4 0.6 0.8 1];
```

si nota come con questi valori diano prestazioni ottime. In figura 3.9 è riportato il caso più interessante tra quelli analizzati.

Per  $K_D$  troppo basso si osserva una sovraelongazione eccessiva (circa del 20 %), mentre con valori leggermente superiori la risposta rispetta tutte le



**Figura 3.7:** Risposta del sistema con controllore PD, con  $K_P = 10$  e  $K_D$  variabile.

specifiche imposte. Il tempo di assestamento è pari a circa  $25 \text{ ms}$ , l'overshoot è inferiore al 10 % e non c'è errore a regime.

La scelta finale ricade allora sui seguenti coefficienti:

$$K_P = 70, K_D = 0,4.$$

La risposta col controllore PD scelto è quella in figura 3.10:

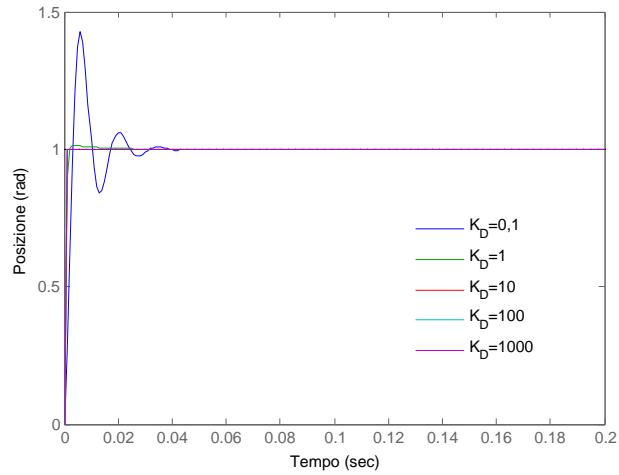
La struttura, quindi, del controllore PD fin qui trovato è:

$$C(s) = 70 + 0,4s \quad (3.11)$$

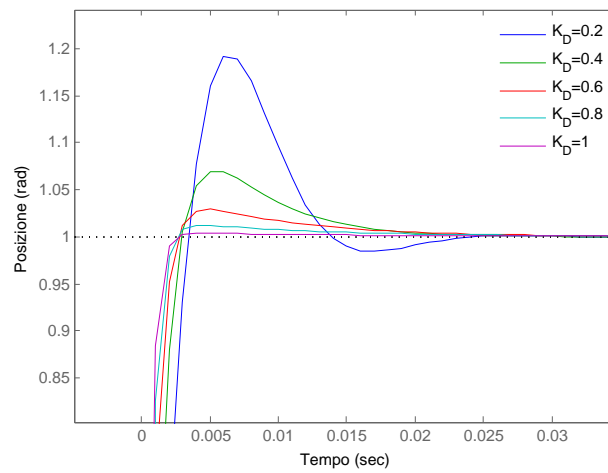
che nel dominio del tempo genera un ingresso di controllo pari a:

$$u(t) = 70e(t) + 0,4 \frac{de}{dt}(t)$$

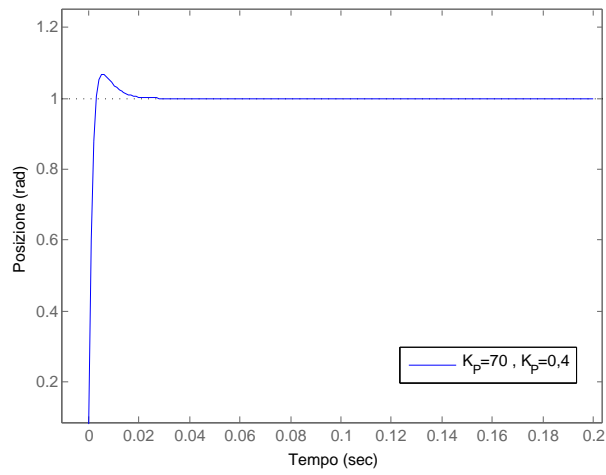
Non si è valutata, tuttavia, la specifica relativa all'errore a regime dovuto al disturbo perchè non si è ancora analizzato questo aspetto. Quello che emergerà sarà un comportamento non conforme alle specifiche perchè il semplice controllore PD non è in grado di rimediare ai disturbi: sarà necessario l'intervento dell'azione integrativa. Gli effetti dei disturbi e il comportamento del sistema in loro presenza sarà l'oggetto del capitolo successivo.



**Figura 3.8:** Risposta del sistema con controllore PD, con  $K_P = 100$  e  $K_D$  variabile.



**Figura 3.9:** Dettaglio della risposta del sistema con controllore PD, con  $K_P = 70$  e  $0,2 \leq K_D \leq 1$ .



**Figura 3.10:** Risposta del sistema col controllore PD scelto.

### 3.1.3 Controllore proporzionale - integrativo PI

Un'ulteriore tipo di controllore è quello proporzionale-integrativo, in cui si mantiene  $K_D = 0$  rendendo nullo il contributo del termine derivativo. La funzione di trasferimento del controllore PID è adesso:

$$C(s) = K_P + \frac{K_I}{s}. \quad (3.12)$$

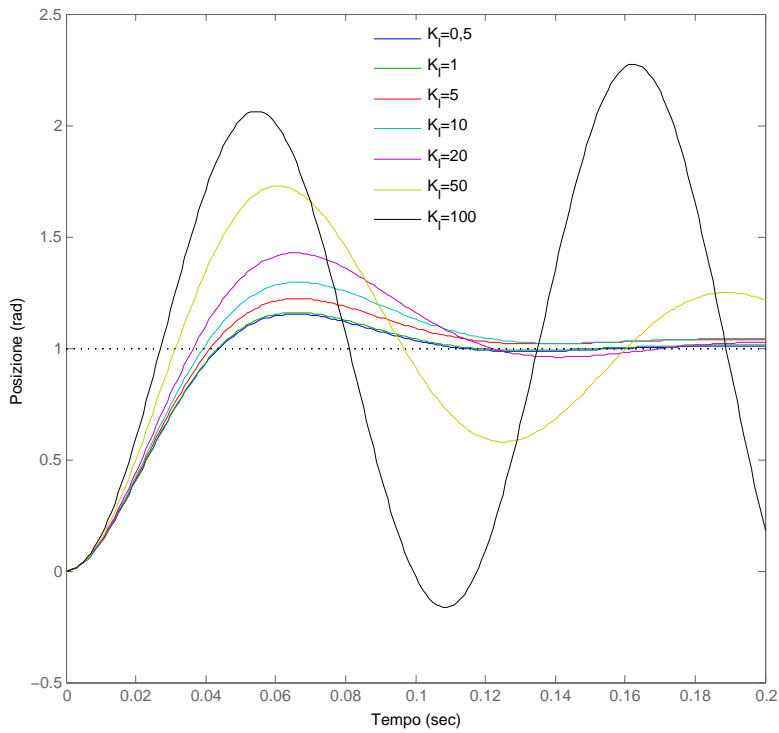
Il risultato ottenuto nella sezione (1.1.1), è che i valori migliori di  $K_P$  sono quelli compresi tra 1 e 5 (a grandi linee). Per testare il nuovo comportamento del sistema con l'introduzione del termine integrale, si utilizza una simulazione MatLab in cui si fanno variare alcuni valori di  $K_P$  nel range indicato prima, e corrispondentemente si provano anche alcuni valori per  $K_I$ . Il codice necessario è il seguente:

```
KiVect=[0.5 1 5 10 20 50 100];
t=0:0.001:0.2;
step(sys,t)
for n=1:length(KiVect)
    controllore=tf([Kp KiVect(n)], [1 0]);
    sys=feedback(controllore*motore,1);
    step(sys,t)
    xlabel('Tempo');
    ylabel('Posizione (rad)');
    title(' ');
    legend('Kp=1.5', strcat('K_{I}= ',num2str(KiVect(n))),
           'Location','NorthEast');
    legend('boxoff');
    hold on
end
hold off
```

I grafici che si ottengono sono quelli che vanno da figura 3.11 a figura 3.13.

Dall'osservazione di tutti questi grafici si nota che a mano a mano che  $K_I$  aumenta, aumentano molto anche le sovraelongazioni nel transitorio, sia in ampiezza che in frequenza. L'aumento invece di  $K_P$  causa un aumento dell'overshoot nei termini con  $K_I$  minore. Confrontando ad esempio il grafico ottenuto per  $K_P = 1,5$  con quello per  $K_P = 3$  si vede come la sovraelongazione della curva per  $K_I$  minore passa da un valore quasi accettabile per le specifiche di progetto imposte a uno decisamente superiore agli 0,2 radianti, cioè un overshoot maggiore del 20%. Gli effetti del **contributo integrale**





**Figura 3.11:** Risposta del sistema con controllore PI, con  $K_P = 1, 5$  e  $K_I$  variabile.

sono infatti di far **diminuire il tempo di salita**, **aumentare le sovraelongazioni** e di **eliminare l'errore a regime**.

Con nessuna delle combinazioni di coefficienti  $K_P - K_I$  tuttavia si raggiungono delle prestazioni accettabili da parte del sistema. Il controllore PI non è adatto al controllo di questo motore.

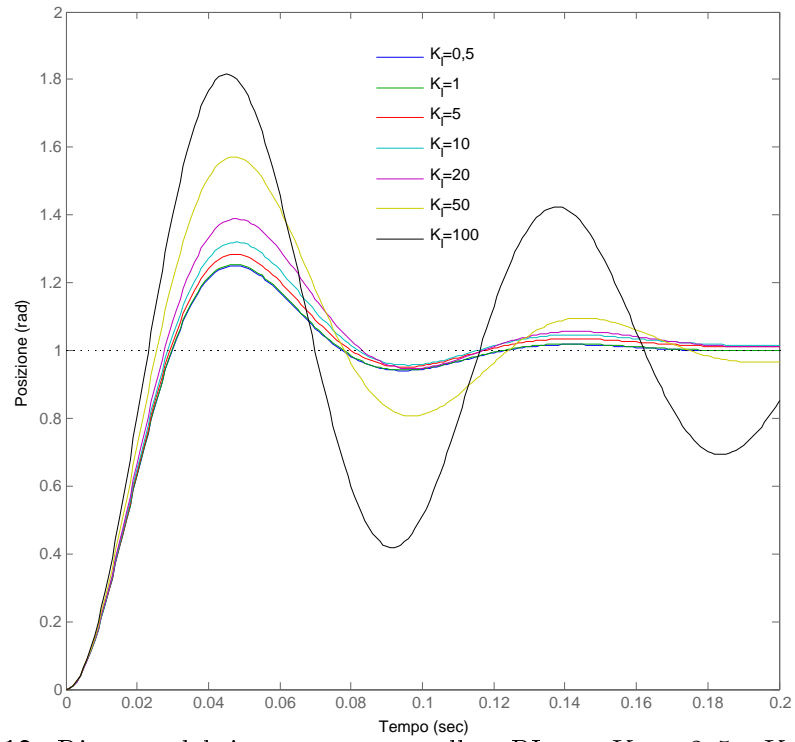


Figura 3.12: Risposta del sistema con controllore PI, con  $K_P = 2, 5$  e  $K_I$  variabile.

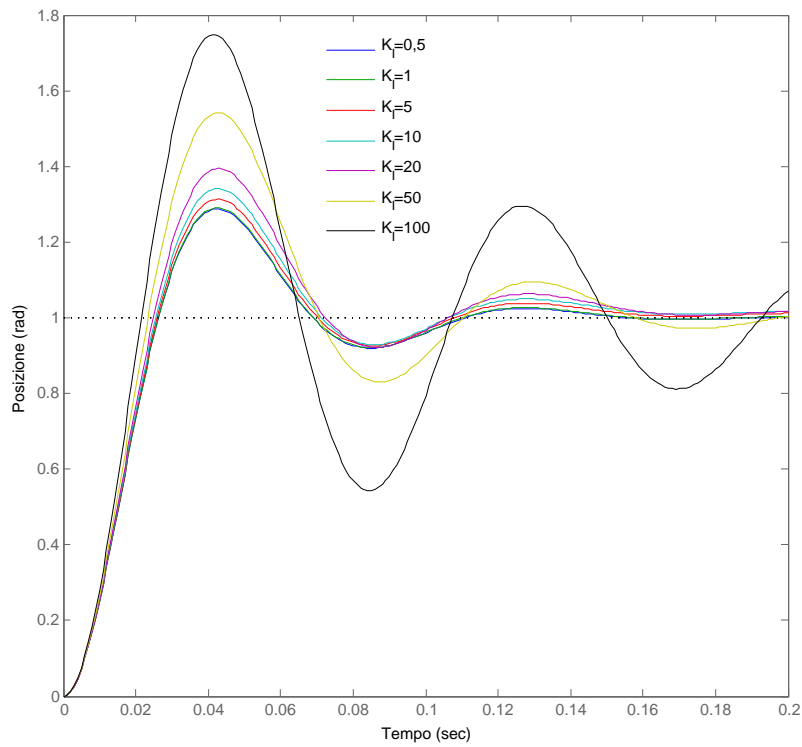
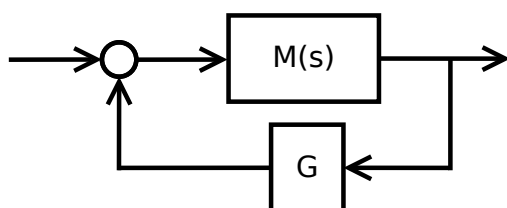


Figura 3.13: Risposta del sistema con controllore PI, con  $K_P = 3$  e  $K_I$  variabile.

## 3.2 Progetto del controllore tramite il luogo delle radici

Il metodo del luogo delle radici è un procedimento grafico che permette di studiare lo spostamento dei poli di una FdT ad anello chiuso al variare di un parametro reale  $G$ <sup>1</sup>, che sostanzialmente rappresenta un guadagno puramente proporzionale messo nel ramo di retroazione del sistema.

La configurazione è quella di figura 3.14.



**Figura 3.14:** Configurazione del sistema per lo studio con il luogo delle radici.

La FdT in catena chiusa relativa allo schema riportato è

$$\frac{M(s)}{1 + GM(s)}$$

ma considerando che  $M(s)$  è una frazione del tipo  $\frac{n(s)}{d(s)}$ , dopo qualche passaggio si scopre che il polinomio da studiare (che è il denominatore della FdT in catena chiusa) è  $d(s) + Gn(s)$ , cioè

$$LJs^3 + (Lb + RJ)s^2 + (Rb + K^2)s + GK .$$

Si passa ora al problema pratico. Le specifiche di progetto imposte per il sistema erano:

- tempo di assestamento  $T_a < 0,04$  secondi;
- overshoot  $< 16\%$ ;
- steady-state error nullo;
- steady-state error dovuto agli errori nullo;

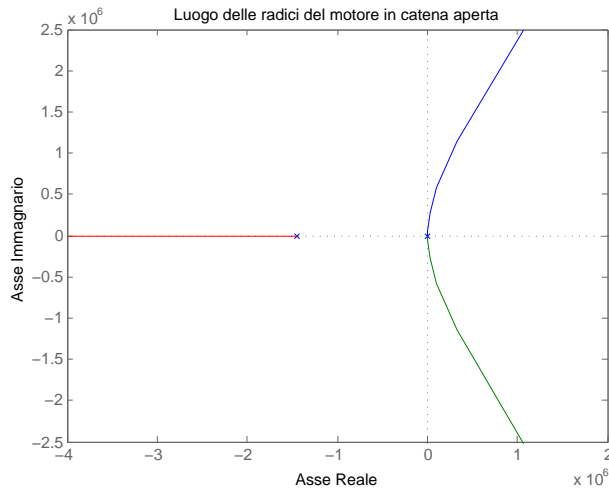
Ci sono due strade percorribili in a questo punto: si può procedere con il polinomio di terzo grado oppure si può ricorrere all'approssimazione dei poli dominanti.

<sup>1</sup>In letteratura il parametro utilizzato nello studio del luogo delle radici è  $K$ , ma in questa tesi  $K$  indica la costante di FCEM, per cui si utilizza  $G$  per evitare confusione.

### 3.2.1 Polinomio completo

In questa sezione si tratta il polinomio nel suo completo, senza trascurare le sue radici meno influenti al comportamento del sistema. Con le seguenti istruzioni MatLab si ottiene il luogo delle radici in questione, che è riportato nella figura sottostante.

```
rlocus(motor)
Title('Luogo delle radici del motore in catena aperta');
xlabel('Asse Reale');
ylabel('Asse Immaginario');
```



**Figura 3.15:** Luogo delle radici del motore

Si nota subito che i tre poli della funzione di trasferimento sono reali. Osservando inoltre la scala degli assi nel luogo, ci si rende conto che c'è un polo in bassissima frequenza, mentre gli altri due sono molto vicini tra loro, al punto che sembrano essere uno solo situato nell'origine: il polinomio è tuttavia di terzo grado, quindi ha tre radici e tre sono anche i rami distinti nel luogo. Guardando il luogo delle radici si nota anche come i due rami che nascono dai due poli vicini all'origine, vadano nel semipiano positivo del piano complesso: causano instabilità del sistema. Con il comando:

```
R=roots(den)
```

si ottengono le radici del polinomio, le quali vengono memorizzate nel vettore  $R$ . Il risultato del comando è appunto

$$R = \begin{matrix} 0 \\ -1.4545e+06 \\ -59.2260 \end{matrix}$$

in cui si ha la conferma di quanto si vede nel luogo tracciato riguardo la posizione dei poli.

Le specifiche sulle prestazioni del motore possono essere ora tradotte in specifiche adatte a essere imposte al luogo delle radici: verranno definite infatti delle regioni del luogo tali che se i poli sono posizionati all'interno delle stesse allora le prestazioni del sistema sono conformi a quelle desiderate. Nella pratica si impongono le relazioni seguenti:

- sovraelongazione  $S = e^{-\frac{\pi}{\tan\theta}}$ ;
- tempo di assestamento  $T_a = \frac{3}{\sigma}$ .

dove  $\theta$  definisce una regione angolare del piano e  $\sigma$  è la parte reale dei poli e costituisce un vincolo sull'asse reale del piano. Imponendo i valori dati dalle specifiche

$$S < 0,16, \quad T_a < 0,04$$

si trovano le regioni di piano cercate. Il vincolo sul massimo overshoot ammissibile genera una regione angolare per l'appartenenza dei poli mentre la specifica sul tempo di assestamento crea una retta verticale nel luogo delle radici.

Un modo semplice per imporre le specifiche di progetto è l'utilizzo del pacchetto Sisotool di Matlab. Come si vede in figura 3.16

impostando il *settling time* e il *percent overshoot* si ottiene una delimitazione del luogo che indica la regione in cui i poli devono stare per fare in modo che il sistema rispetti le specifiche. Si imposta allora il settling time  $< 40 \text{ ms}$  e il percent overshoot  $< 16\%$ . La linea nera che si vede nel luogo (nel grafico a sinistra), è proprio il risultato di questa imposizione e la regione da evitare è quella ombreggiata. Come si vede, il sistema non è stabile, perchè ci sono due rami che vanno nel semipiano complesso positivo. Un'ulteriore conferma arriva dalla risposta del sistema al gradino, che si vede nel grafico a sinistra di figura 3.16: il comportamento del motore è divergente nel tempo.

Il pacchetto Sisotool permette di modificare il luogo aggiungendo poli e/o zeri (reali o complessi coniugati) in modo da deviare i rami a piacimento fino a far stare tutto il luogo nelle zone desiderate. Il tutto controllando in maniera real-time la risposta del motore.

Si comincia con l'aggiunta di una coppia di zeri complessi vicino ai due poli

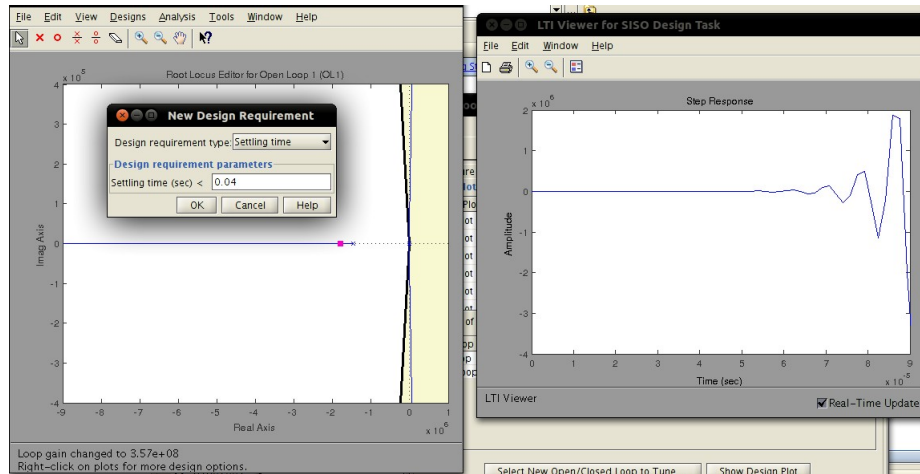


Figura 3.16: Impostazione delle specifiche con Sisotool.

che divergono in modo da deviare i rami a stare nella parte buona del piano. L'operazione è riportata in figura 3.17.

Con l'aggiunta degli zeri il luogo è cambiato notevolmente: i due rami che prima erano divergenti verso il semipiano positivo, ora girano verso quello negativo, come si voleva: vanno a finire nei due zeri che sono stati aggiunti. Anche il comportamento del motore è cambiato totalmente: ora è convergente alla posizione di riferimento.

Resta da modificare il valore del guadagno nel ramo di retroazione fino ad ottenere prestazioni soddisfacenti. Questa operazione si fa andando a modificare manualmente posizione degli zeri e guadagno fino ad ottenere il risultato cercato. Si sa che i rami del luogo delle radici sono ottenuti al variare di  $G$ , cioè del guadagno di retroazione. Ad ogni valore di  $G$  corrisponde una posizione dei poli nel piano e i rami stanno proprio a rappresentare lo spostamento degli stessi. Anche in questo caso Sisotool è d'aiuto. Trascinando il quadratino viola lungo i rami, si vanno a selezionare tutti gli infiniti valori di  $G$  e in corrispondenza di ognuno di questi si può verificare il comportamento del sistema. Dopo qualche simulazione si trova il valore che permette di rispettare le specifiche. Una considerazione importante emerge dalle simulazioni. Per valori di  $G$  inferiori all'ordine di grandezza  $10^7$ , il comportamento del sistema è più simile a uno di primo grado, mentre per valori superiori assomiglia al comportamento del sistema del secondo ordine.

Gli zeri che vengono aggiunti al luogo e la scelta del guadagno danno origine infine al seguente controllore:

$$C(s) = 2 \times 10^8 \left( 6.25 \times 10^{-12} s^2 + 5 \times 10^{-6} s + 1 \right)$$

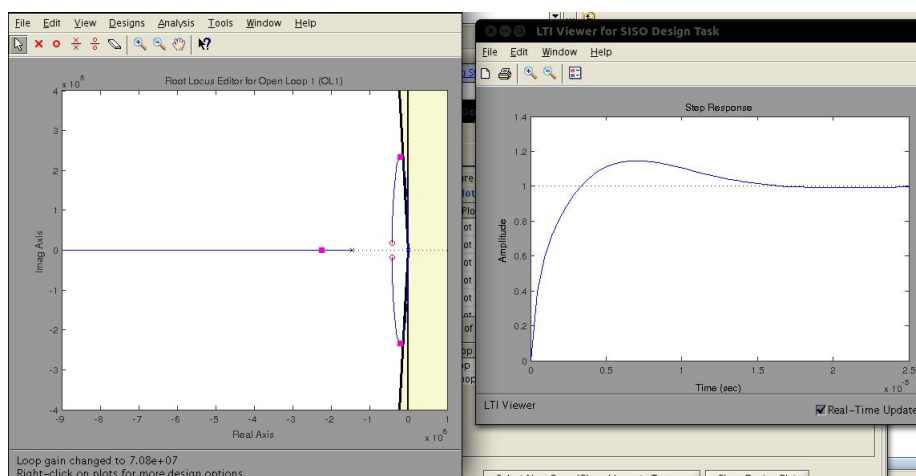


Figura 3.17: Aggiunta degli zeri al luogo con Sisotool.

le cui radici (che sono i due zeri complessi coniugati che sono stati aggiunti) sono:

$$s_1 = -4 \times 10^5 + 10000i \quad s_2 = -4 \times 10^5 - 10000i ,$$

ela risposta del motore con il compensatore così trovato è quella di figura 3.18.

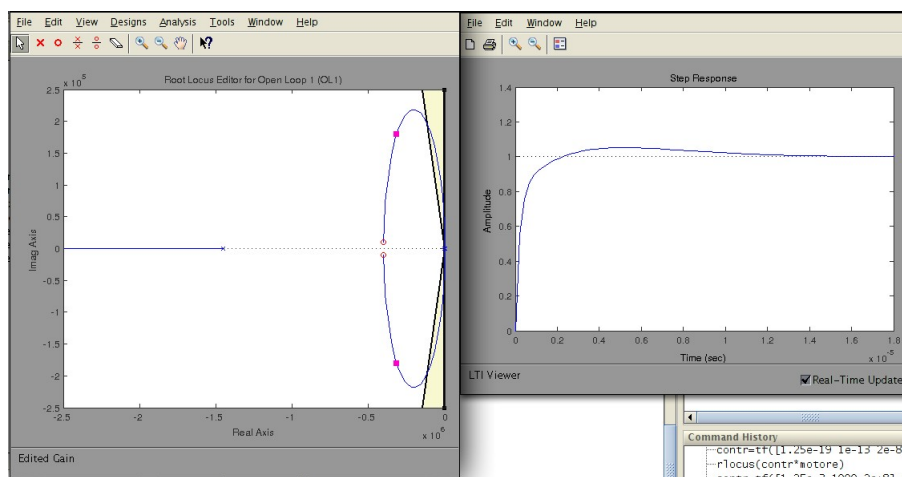
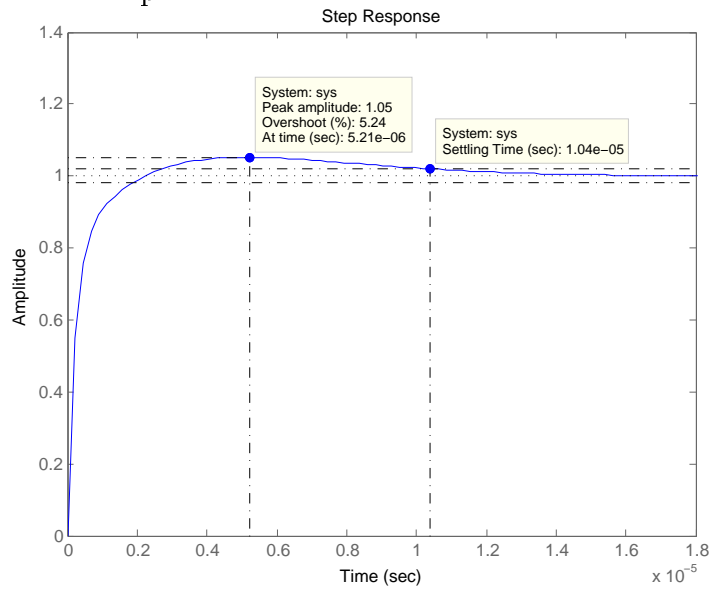


Figura 3.18: Scelta del guadagno con Sisotool.

Per costruire il compensatore si può usare Matlab. Con i seguenti comandi

```
contr=tf([1.25e-3 1000 2e+8],1);
sys=feedback(contr*motore);
step(sys)
```

si costruisce il controllore e si verifica la risposta del sistema: questa è riportata in figura 3.19 e in essa sono indicati anche il tempo di assestamento e la massima sovraelongazione così ottenuti. Entrambi i valori soddisfano abbondantemente le specifiche.



**Figura 3.19:** Risposta del sistema con il controllore trovato attraverso il luogo delle radici completo.



### 3.2.2 Approssimazione dei poli dominanti

Una soluzione alternativa nella progettazione del controllore tramite luogo delle radici è quella di utilizzare l'**approssimazione dei poli dominanti**. Data la funzione di trasferimento del motore

$$M(s) = \frac{\Theta(s)}{V(s)} = \frac{K}{sK^2 + s(Ls + R)(Js + b)}. \quad (3.13)$$

si era visto nella sezione precedente che i poli della FdT del motore sono:

```
R =  
      0  
 -1.4545e+06  
 -59.2260
```

Quello che si nota è che il secondo polo è trascurabile rispetto a quello nell'origine e a al terzo perchè si trova in bassissima frequenza, quindi il suo contributo al comportamento del sistema è praticamente ininfluenza se non viene sollecitato da guadagni molto elevati. Per queste ragioni si può ometterlo, ottenendo così un sistema del secondo o grado.

Con la serie di comandi

```
rlocus(motor)  
Title('Luogo delle radici del motore in catena aperta');  
xlabel('Asse Reale');  
ylabel('Asse Immaginario');
```

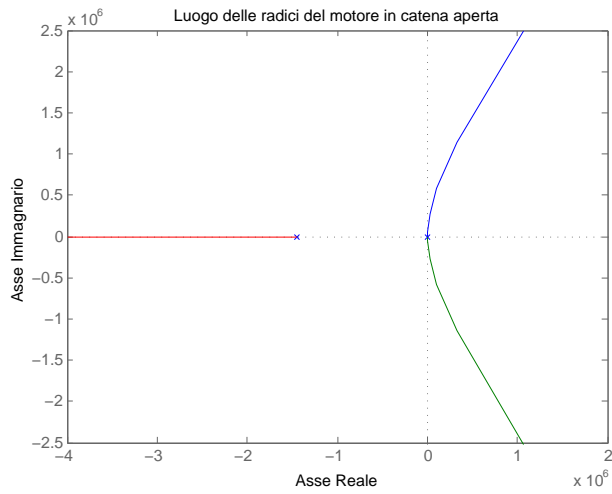
ci si rende conto della situazione. Questo script infatti produce il grafico del luogo delle radici della funzione di trasferimento *motor* che è la 3.13. Il risultato è quello di figura 3.20.

Si nota la posizione sull'asse reale dei poli come era stato detto prima. Osservando la scala degli assi inoltre, si vede come il polo che si vuole trascurare sia molto lontano dagli altri due, che addirittura sono indistinguibili e sembrano posizionati entrambi nell'origine.

Con il comando

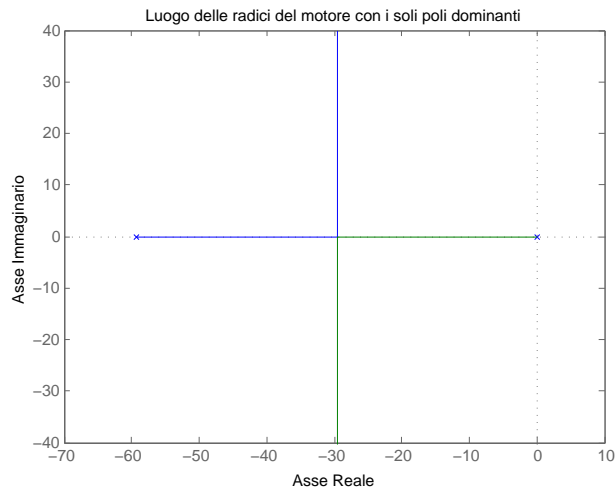
```
den2=deconv(den,[1/max(abs(R)) 1]);  
motore_solo_dom=tf(num,den2);
```

si va a fare questa semplificazione: *deconv()* fa la divisione del denominatore della funzione di trasferimento del motore per il polinomio contenente il polo da trascurare (che è il più grande in modulo) ottenendo così il denominatore *den2* che è di secondo grado. Fatto ciò, si crea la nuova funzione di



**Figura 3.20:** Luogo delle radici del motore.

trasferimento del motore con il solito comando  $tf()$ . Come si vede nel luogo di figura 3.21 l'approssimazione non ha influito sui due poli rimasti, proprio come si voleva.



**Figura 3.21:** Luogo delle radici del motore con i soli poli dominanti.

Il modello di secondo grado a cui si è arrivati ha dunque la forma

$$s^2 + 2\xi\omega_n s + \omega_n^2, \quad (3.14)$$

con i poli che sono  $p_1 = 0$  e  $p_2 = -59, 2260$ .

Si pone problema di tradurre le specifiche date in una forma adatta all'uso

tramite luogo delle radici. Il punto fondamentale è che i vincoli imposti determineranno l'insieme dei poli della funzione di trasferimento adatti a questo tipo di progetto del controllore. Nella sezione precedente questa operazione era stata fatta con l'aiuto del pacchetto Sisotool di Matlab; in questa invece si procederà in maniera diversa, più manuale. La traduzione di queste informazioni avviene mediante il calcolo di due parametri, il **coefficiente di smorzamento**  $\xi$  e la **pulsazione naturale**  $\omega_n$ . Queste quantità sono legate al sistema (3.14) e alle sue radici secondo le relazioni

$$p_{1,2} = \sigma^2 + \omega^2, \quad \xi\omega_n = \sigma, \quad \omega_n^2 = \sigma^2 + \omega^2.$$

Dalla formula

$$M_p = e^{-\pi\left(\frac{\xi}{\sqrt{1-\xi^2}}\right)}$$

in cui  $M_p$  è il picco di sovralongazione (che in questo caso è pari a 0,16 V per un segnale in ingresso di ampiezza unitaria e con un overshoot massimo del 16 %) si ricava che:

$$\xi = \sqrt{\frac{\left(\frac{\ln M_p}{\pi}\right)^2}{1 + \left(\frac{\ln M_p}{\pi}\right)^2}}. \quad (3.15)$$

Per calcolare la pulsazione naturale invece, si impone la relazione

$$T_a = \frac{3}{\xi\omega_n} \leq 0,04 \text{ s}$$

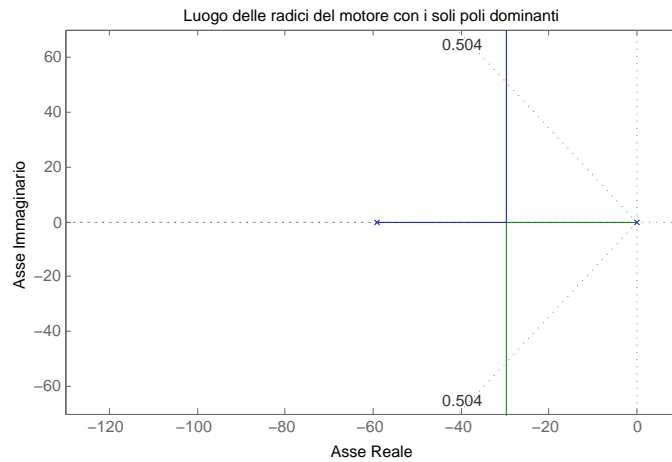
tra il tempo di assestamento  $T_a$ , il coefficiente di smorzamento e la pulsazione naturale. Esplicitando  $\omega_n$  si ottiene che quest'ultima appunto vale:

$$\omega_n \geq \frac{3}{0,04\xi} \frac{\text{rad}}{\text{s}}. \quad (3.16)$$

Con i valori delle specifiche risulta:  $\xi = 0,5039$  e  $\omega_n \geq 148,8485$ . Una volta calcolati questi parametri si può procedere col tracciamento del luogo delle radici e delle regioni di piano che soddisfano le specifiche. Il codice è:

```
rlocus(motore_solo_dom)
sgrid(z,Wn)
```

Il comando *sgrid()* traccia nel piano complesso le regioni nelle quali dovrebbero stare i poli per soddisfare le specifiche, date attraverso lo smorzamento ( $\xi=z$ ) e la pulsazione naturale ( $\omega_n=Wn$ ). Il risultato prodotto è il luogo di figura 3.22.

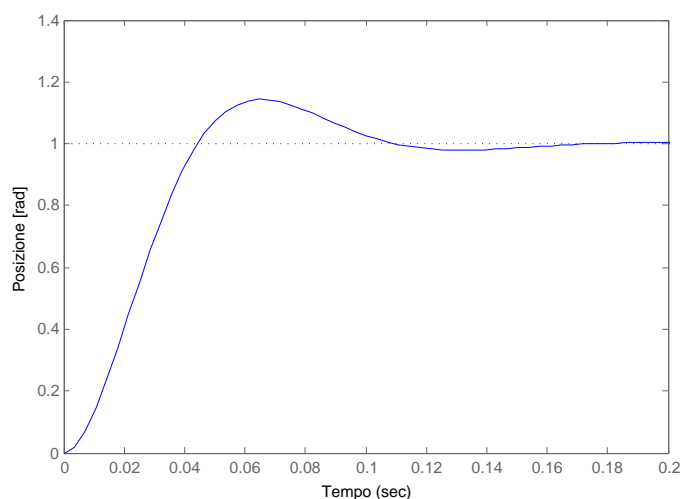


**Figura 3.22:** Imposizione delle specifiche di controllo nel luogo delle radici con i soli poli dominanti.

Nella figura 3.22 si vede come la parte interessante dal punto di vista del rispetto delle specifiche è quella racchiusa tra le due linee oblique, ma soprattutto che per piccoli valori del guadagno, i poli vi rimangono dentro. La funzione *rlocfind()* permette di selezionare tramite un cursore un punto di interesse nel luogo delle radici ottenendone il guadagno e i poli. Attraverso questa funzione si può verificare che il guadagno massimo che si può scegliere in modo da rimanere all'interno delle regione voluta è circa  $K = 1,5$ . Per valori superiori si attraversano i limiti. Si verifica rapidamente con Sisotool che un valore così basso del guadagno non permette nemmeno di avvicinarsi alle prestazioni cercate: nella figura 3.23, si vede che sia il tempo di assestamento che il picco di overshoot sono molto superiori ai valori indicati nelle specifiche. Si manifesta quindi la necessità di utilizzare un compensatore più complesso del semplice guadagno proporzionale in modo da migliorare le prestazioni del sistema.

In questa fase si nota il fatto che la presenza di un polo reale lontano dall'asse immaginario del piano complesso aumenta la velocità di risposta del sistema (minor tempo di salita) però introduce delle sovraelongazioni. Inserendo dunque uno zero reale molto vicino all'origine si va a compensare gli effetti dei poli dominanti, migliorando il transitorio. La situazione è riassunta in tabella 3.1.

Seguendo quanto riportato in tabella 3.1, dopo alcune prove e simulazioni si arriva finalmente al compensatore che permette di far rispettare le specifiche di controllo. Il risultato è quindi:



**Figura 3.23:** Risposta del motore (con i soli poli dominanti) con un compensatore proporzionale con guadagno pari a  $K = 1,5$ .

**Tabella 3.1:** Effetti dell'inserimento nel luogo delle radici di poli e zeri.

	<b>Polo reale</b>	<b>Zero reale</b>
<b>Vicino all'origine</b>	rallenta la risposta	migliora il transitorio eliminando le oscillazioni
<b>Lontano dall'origine</b>	aumenta la velocità delle risposta introduce oscillazioni	aumenta le oscillazioni nel transitorio

$$C(s) = 40 \left( 1 + 0,02s + 0,0001s^2 \right), \quad (3.17)$$

in cui è stato inserita la coppia di zeri complessi coniugati  $z_{1,2} = -100 \pm 5i$ : con questa coppia di zeri, i rami del luogo sono attirati verso la parte buona del piano. Il luogo delle radici che si ottiene con questo compensatore è quello di figura 3.24.

La risposta del motore è infine quella in figura 3.25.

Dalla figura 3.25 si vede come le specifiche siano soddisfatte. Il tempo assestamento è molto inferiore ai  $40 \text{ ms}$  richiesti, così come è quasi assente l'overshoot. Anche il valore asintotico è in linea con le prestazioni desiderate.

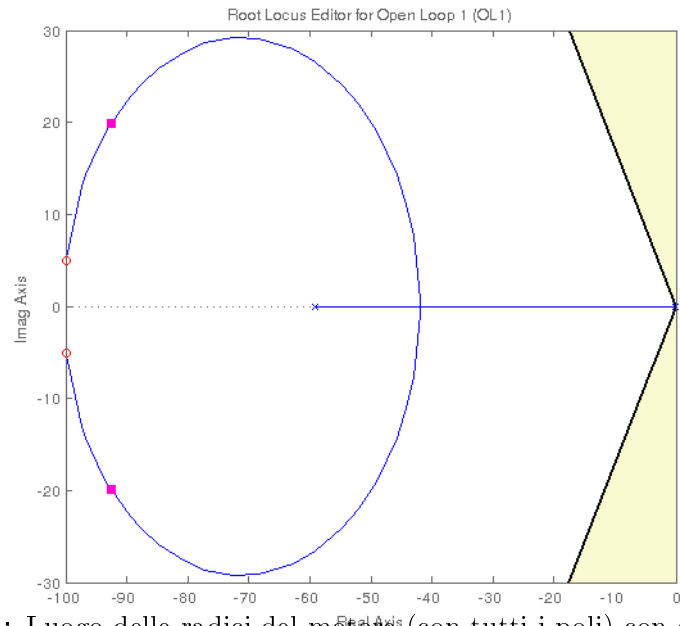


Figura 3.24: Luogo delle radici del motore (con tutti i poli) con compensatore.

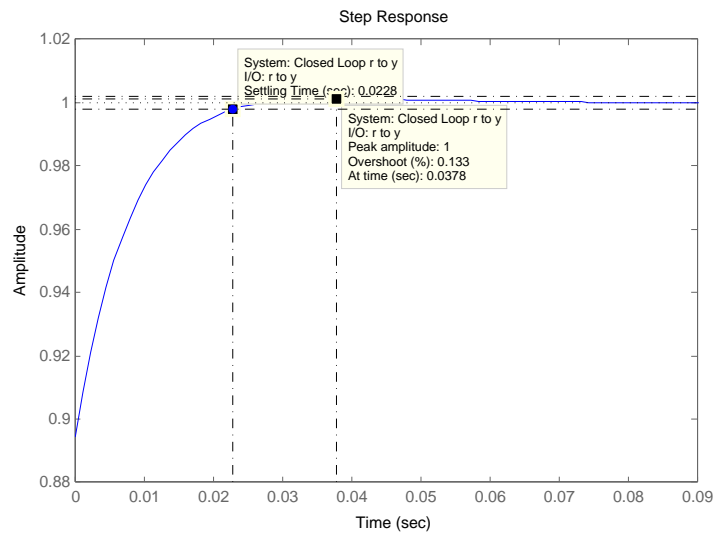
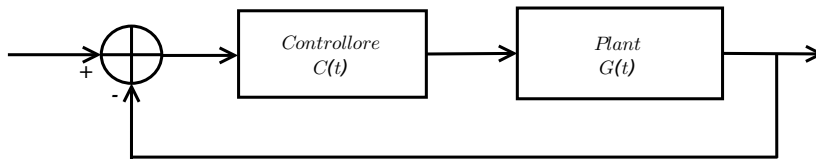


Figura 3.25: Risposta del motore a un gradino di tensione con il compensatore (3.17).

### 3.3 Progetto del controllore tramite sintesi in frequenza

In questa sezione verrà fatta una sintesi del controllore per il motore con il metodo della sintesi in frequenza, basato sullo studio dei diagrammi di Bode. La configurazione a cui si farà riferimento in seguito è quella di figura 3.26.



**Figura 3.26:** Configurazione del sistema usata per la sintesi in frequenza.

La tecnica della sintesi in frequenza consiste nel tradurre le specifiche date in *specifiche in frequenza*, ovvero in specifiche utilizzabili nei diagrammi di Bode. Fatta la traduzione, queste vengono imposte al sistema in catena aperta avente funzione di trasferimento  $W(s) = C(s)G(s)$ , dove  $C(s)$  è la funzione di trasferimento del controllore (incognita) e  $G(s)$  è chiaramente quella del motore. In questo modo, si determina  $C(s)$  in modo tale che il sistema in catena aperta  $W(s)$  soddisfi le specifiche di progetto.

Si ricorda che le specifiche assegnate per il motore sono:

- tempo di assestamento  $T_a \leq 40 \text{ ms}$ ;
- sovraelongazione inferiore al 16% ( $S = 0,16$ );
- errore a regime nullo;
- errore a regime dovuto ai disturbi nullo.

A questo punto è necessario fare un'ipotesi di lavoro: al fine di determinare che tipo di sistema sarà  $W(s)$  si impone errore a regime nullo a gradino di tensione e si accetta un errore molto piccolo a seguito di una rampa come input. Si sceglie questo valore arbitrariamente piccolo, cioè sarà pari a  $\epsilon = 0,001$ . Questa scelta impone allora che  $W(s)$  sarà un sistema di tipo uno.

### 3.3.1 Errore a regime

La tecnica di sintesi oggetto di questa sezione prevede che le **funzioni di trasferimento** interessante siano in **forma di Bode**. La rappresentazione cercata è allora:

$$W(s) = \frac{K_W}{s^{h_W}} \bar{W}(s) , \text{ con } \bar{W}(0) = 1$$

$$C(s) = \frac{K_C}{s^{h_C}} \bar{C}(s) , \text{ con } \bar{C}(0) = 1$$

$$G(s) = \frac{K_G}{s^{h_G}} \bar{G}(s) , \text{ con } \bar{G}(0) = 1$$

Da queste espressioni e dalla relazione  $W(s) = C(s)G(s)$  si deduce che:

$$W(s) = \frac{K_C K_G}{s^{h_C + h_G}} \bar{C}(s) \bar{G}(s) .$$

Si ottiene infine:

$$K_W = K_C K_G$$

$$h_W = h_C + h_G$$

$$\bar{W}(s) = \bar{C}(s) \bar{G}(s) , \text{ con } \bar{W}(0) = 1$$

Nelle relazioni appena trovate, i termini incogniti sono quelli relativi a  $C(s)$  ovvero  $K_C$  e  $h_C$ . Questi termini vanno determinati a partire dagli altri che sono noti in quanto  $K_G$  e  $h_G$  sono rispettivamente il guadagno di Bode e il numero di poli nell'origine della funzione di trasferimento del motore (che è nota) e  $K_W$  e  $h_W$  vanno dedotti a partire dalle specifiche in frequenza. Esplicitando le incognite si ottiene dunque:

$$K_C = \frac{K_W}{K_G} , \quad h_C = h_W - h_G .$$

La forma di Bode di  $G(s)$  in particolare è la seguente:

$$G(s) = \frac{K}{s \left( (Js + b)(Ls + R) + K^2 \right)} = \frac{K}{s} \frac{1}{(Rb + K^2) \left( 1 + \frac{RJ + Lb}{Rb + K^2} s + \frac{JL}{Rb + K^2} s^2 \right)}$$

da cui si vede che

$$K_G = \frac{K}{Rb + K^2} , \quad h_G = 1 .$$



Per quanto detto prima sulla specifica sull'errore a regime, per avere una risposta con errore nullo al gradino ci vuole un sistema di tipo  $1^2$ , ovvero con  $h_W = 1$ . Il valore del guadagno di Bode di  $W(s)$  è invece pari a  $K_W = \frac{1}{\epsilon}$  con  $\epsilon = 0,001$  cioè  $K_W = 1000$ . Da tutto ciò si arriva a dire che

$$K_C = \frac{1000}{\frac{K}{Rb+K^2}} = \frac{1000(Rb + K^2)}{K} = 27,91 \quad (3.18)$$

$$h_C = h_W - h_G = 0 \quad (3.19)$$

### 3.3.2 Sovraelongazione

La specifica sulla sovraelongazione può essere convertita in frequenza sfruttando l'intuizione: è infatti facile pensare che un sistema con sovraelongazioni marcate sia molto vicino all'instabilità e quindi abbia un margine di fase piccolo. Esiste quindi una proporzionalità inversa tra  $S$  e  $m_\phi$ :

$$S \text{ aumenta} \iff m_\phi \text{ diminuisce} .$$

Una stima del valore del margine di fase è allora:

$$m_\phi = 1 - 0,8S .$$

Dalla specifica sull' overshoot si ha che  $S = 0,16$ , per cui

$$m_\phi = 0,872 \text{ rad} . \quad (3.20)$$

### 3.3.3 Pulsazione di attraversamento

La specifica sulla pulsazione di attraversamento si può ricavare attraverso un breve ragionamento. In generale si nota che per sistemi abbastanza regolari un aumento della banda passante coincide con una maggiore prontezza (minor tempo di salita) del sistema. Esiste inoltre la relazione  $w_a \simeq 5B$  dove  $B$  è appunto la banda passante e  $w_a$  chiaramente la pulsazione di attraversamento. Da queste due osservazioni si deduce che il tempo di salita e la pulsazione di attraversamento sono tra loro inversamente proporzionali. Una stima della relazione tra queste due grandezze è data da

$$w_a \simeq \frac{2}{T_s} .$$

---

<sup>2</sup>Un sistema di tipo uno ha un polo nell'origine.

Tra le specifiche assegnate per le prestazioni del motore non ce n'è una relativa al tempo di salita, ma si può ipotizzare un valore sulla base anche dell'informazione sul tempo di assestamento (il quale sarà sicuramente maggiore). Da qui in poi dunque si pone  $T_s = 5 \text{ ms}$  che implica

$$w_a = 400 \frac{\text{rad}}{\text{s}} . \quad (3.21)$$

### 3.3.4 Determinazione della rete correttiva necessaria

Una volta tradotte tutte le specifiche è possibile procedere al calcolo dei parametri  $C$  e  $\Delta\varphi$  che serviranno per decidere quale rete correttiva è necessaria per il motore.

Le informazioni che sono state raccolte fino a questo punto della sintesi sono raccolte nelle relazioni

$$W(s) = \hat{W}(s)\bar{C}(s)$$

dove

$$\hat{W}(s) = \frac{K_C}{s^{h_C}} G(s)$$

è la parte nota mentre  $\bar{C}(s)$  è quella libera.

Con i valori calcolati prima si ottiene

$$\hat{W}(s) = \frac{27,91}{s^0} G(s) = 27,91 G(s)$$

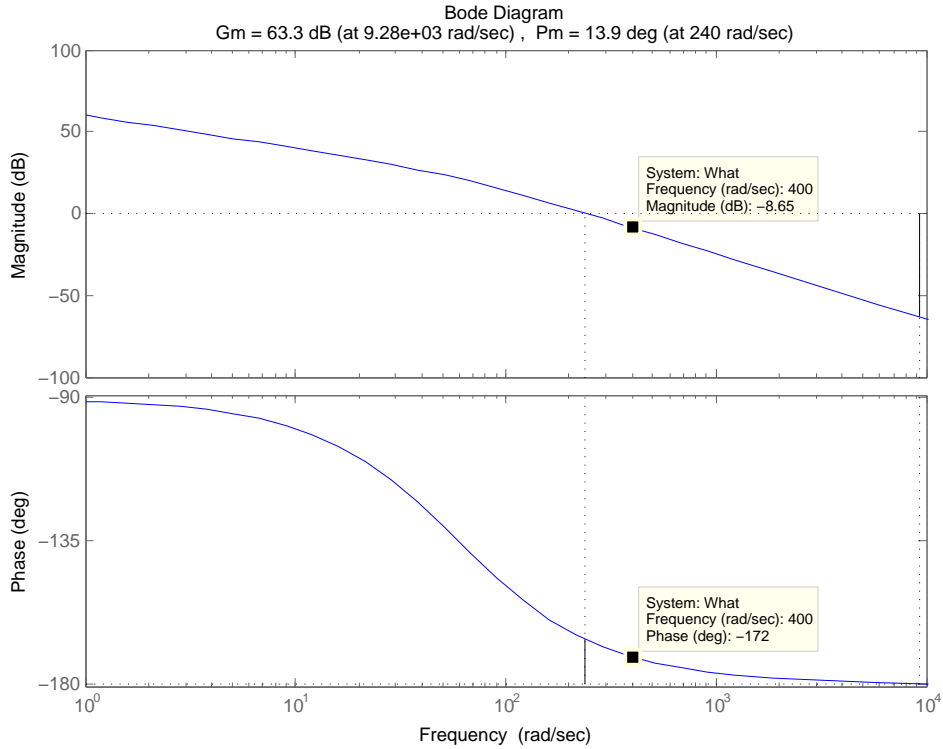
i cui diagrammi di Bode sono quelli in figura 3.27 e sono ottenuti direttamente dalla FdT  $\hat{W}(s)$  con il comando **margin()**.

La pulsazione di attraversamento e il margine di fase richiesti sono:

- $\omega_a = 400 \frac{\text{rad}}{\text{s}}$ ;
- $m_\varphi = 0,872 \text{ rad} \simeq 49,96 \text{ grad}$ .

Dall'osservazione dei diagrammi di figura 3.27 si vede invece che il controllore fin qui ottenuto ha le seguenti caratteristiche:

- $\omega_a^0 = 240 \frac{\text{rad}}{\text{s}}$  è la pulsazione di attraversamento corrente, ovvero tale che  $|\hat{W}(j\omega_a^0)| = 1$ .
- $m_\varphi^0 = \pi + \angle\hat{W}(j\omega_a) = 180 - 172 = 8 \text{ grad}$  è il margine di fase corrente.



**Figura 3.27:** Diagrammi di Bode di modulo e fase di  $\hat{W}(j\omega)$ .

Dato che i valori non sono conformi a quelli richiesti, si utilizza la parte ancora libera del controllore per modificare quella già nota. A tal fine si calcola  $\Delta\varphi$ :

$$\Delta\varphi = m_\varphi - m_\varphi^0 = 49,96 - 8 = 41,96 \text{ grad} > 0 . \quad (3.22)$$

Per definizione, l'informazione sulla pulsazione di attraversamento permette di dire che

$$|W(j\omega_a)| = 1 \Leftrightarrow |\hat{W}(j\omega_a)||\bar{C}(j\omega_a)| = 1$$

da cui si definisce

$$C = |\bar{C}(j\omega_a)| = \frac{1}{|\hat{W}(j\omega_a)|} .$$

Dalla figura 3.27 si vede che  $|\hat{W}(j400)|_{dB} = -8,65dB$  che corrisponde a un guadagno  $|\hat{W}(j400)| = 0,136$  in scala lineare. Si ottiene quindi

$$C = 7,328 > 1 . \quad (3.23)$$

Dai valori ottenuti e dalla figura 3.27 si nota come la funzione di trasferimento abbia una pulsazione di attraversamento inferiore rispetto a quella voluta e

anche che il margine di fase è anch'esso minore al valore desiderato. Il fatto che  $\Delta\varphi > 0$  significa appunto che il sistema ha poca fase rispetto alla desiderata. Dalla tabella 3.2 si vede immediatamente che è necessaria una rete in grado di amplificare il modulo e anticipare la fase di  $\hat{W}(j\omega)$ .

**Tabella 3.2:** Scelta della rete correttrice in base ai parametri  $C$  e  $\Delta\varphi$ .

	$C > 1$	$C < 1$
$\Delta\varphi > 0$	amplificare e anticipare	attenuare e amplificare
$\Delta\varphi < 0$	amplificare e ritardare	attenuare e ritardare

Il contenuto della tabella 3.2 dice intuitivamente che se il modulo della funzione di trasferimento è troppo in alto, attenuando viene tirato verso il basso, in modo da farlo passare per la pulsazione di attraversamento desiderata. Questa informazione è rappresentata dal parametro  $C$ .  $\Delta\varphi$  invece esprime il fatto che se la funzione di trasferimento corrente è abbondante di fase allora si può stare tranquilli per quanto riguarda la stabilità mentre se il margine di fase è scarso allora bisogna aumentarlo per avvicinarsi alle prestazioni cercate.

### 3.3.5 Rete correttrice amplificatrice e anticipatrice

Da quanto detto nella sezione precedente si è arrivati alla conclusione che la FdT a disposizione è scarsa in quanto a margine di fase e necessità di essere amplificata in modo che la sua pulsazione di attraversamento coincida con quella desiderata ai fini della conformità con le specifiche. Bisogna dunque tirare verso l'alto il diagramma del modulo. Il modello di rete correttrice è:

$$\bar{C}(s) = \frac{1 + sT}{1 + saT} \text{ con } 0 < a < 1 \text{ e } T > 0.$$

Per determinare i parametri  $a$  e  $T$  si procede come segue: imponendo  $q = \tan(\Delta\varphi)$  si risolve l'equazione di secondo grado nell'incognita  $a$

$$C^2(C^2q^2 + C^2 - 1)a^2 + C^2q^2a + (q^2 - C^2 + 1) = 0$$

e si trovano le due radici di tale polinomio. Per scegliere quella corretta si prende quella che rispetta il vincolo  $0 < a < 1$ . Questa operazione si fa In Matlab con il codice riportato di seguito.

```
C=7.238;
A=C^2*((C^2)*(q^2)+(C^2)-1);
```

$$\begin{aligned} B &= 2 \cdot (C^2) \cdot (q^2); \\ E &= (q^2) - (C^2) + 1; \\ \Delta &= (B^2) - 4 \cdot A \cdot E; \\ a_1 &= (-B - \sqrt{\Delta}) / (2 \cdot A); \\ a_2 &= (-B + \sqrt{\Delta}) / (2 \cdot A); \end{aligned}$$

in cui A, B ed E sono i tre coefficienti dell'equazione. Le radici sono:

$$\begin{aligned} a_1 &= \\ &-0.1105 \end{aligned}$$

$$\begin{aligned} a_2 &= \\ &0.0932 \end{aligned}$$

Per quanto detto prima si sceglie  $a = 0.0932$ . Per calcolare T invece, dopo avere trovato a, si usa la relazione:

$$T^2 = \frac{1}{\omega_a^2} \frac{C^2 - 1}{1 - a^2 C^2}.$$

Il risultato è:

$$\begin{aligned} T &= \\ &0.0243 \end{aligned}$$

La parte libera del controllore così trovato è allora

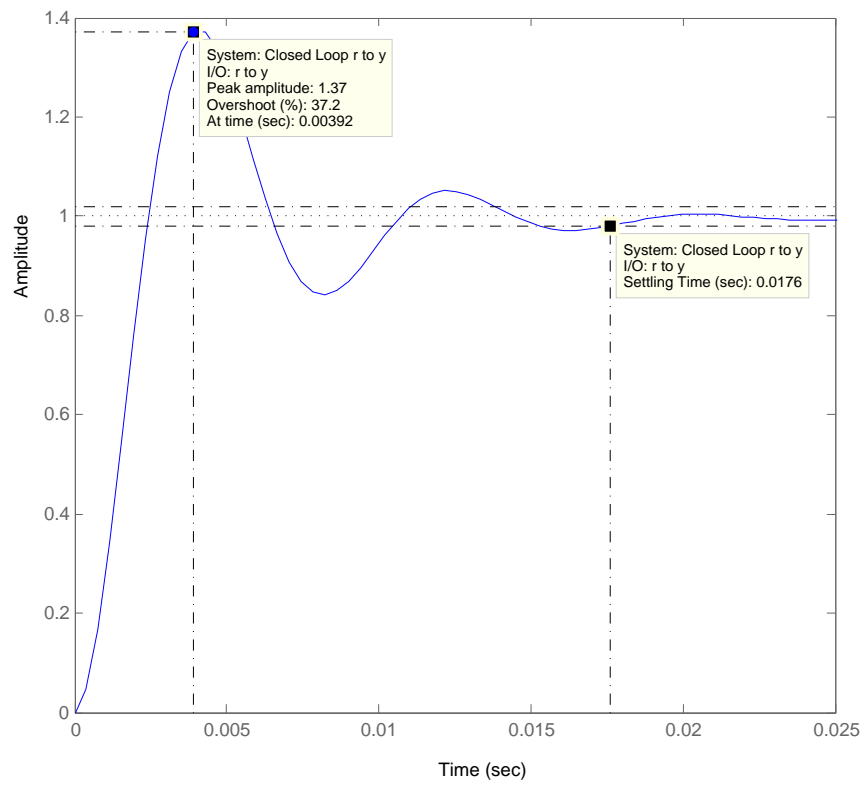
$$\bar{C}(s) = \frac{1 + s0,0243}{1 + s0.00226}$$

La risposta del sistema con il questo controllore è quella riportata in figura 3.28.

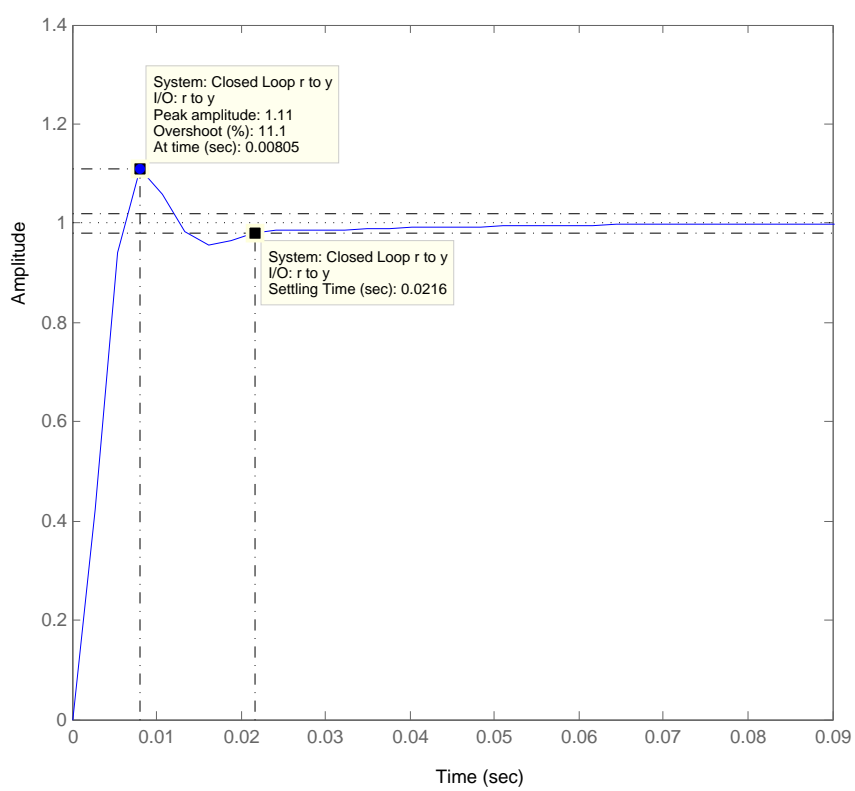
Si vede come la specifica sul tempo di assestamento sia ora rispettata mentre quella sull'overshoot non ancora. Aggiungendo un guadagno statico pari a 0,3 a  $\bar{C}(s)$  si ottiene una risposta soddisfacente (figura 3.29).

Nei diagrammi di Bode del sistema con  $Fdt \frac{W(s)}{1+W(s)}$  sono riportati in figura 3.30.

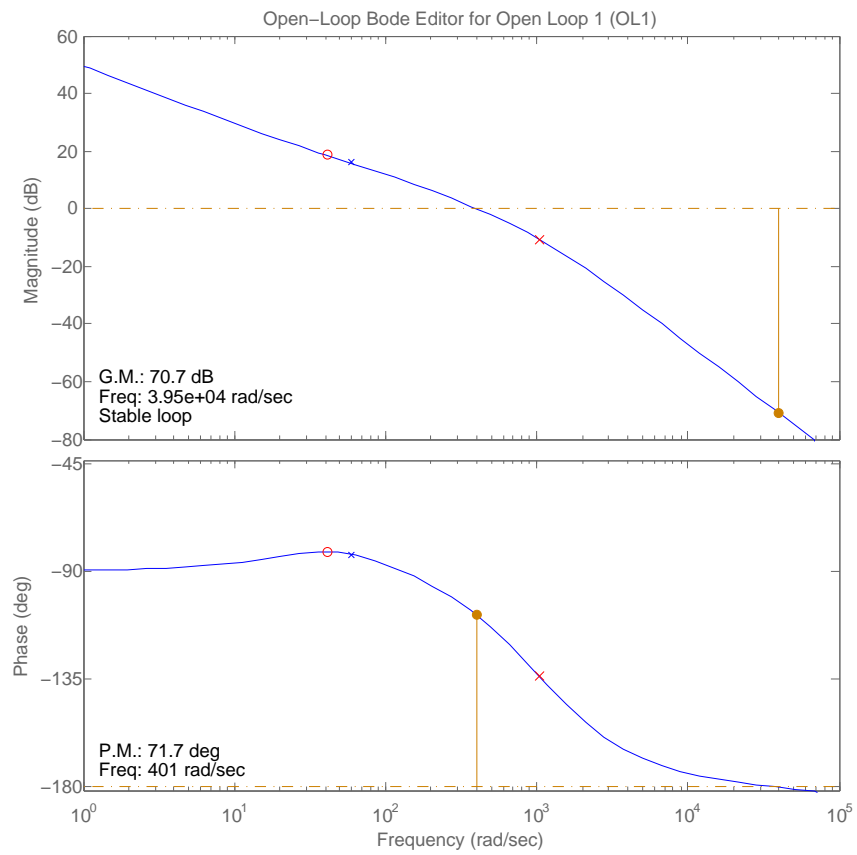
Si vede che finalmente  $W(s)$  passa per la pulsazione di attraversamento  $\omega_a = 400 \frac{rad}{s}$  con un margine di fase che è ora superiore al minimo richiesto in quanto vale circa 71 grad contro i circa 50 richiesti.



**Figura 3.28:** Risposta del sistema con rete amplificatrice e anticipatrice.



**Figura 3.29:** Risposta del sistema con controllore determinato in frequenza.



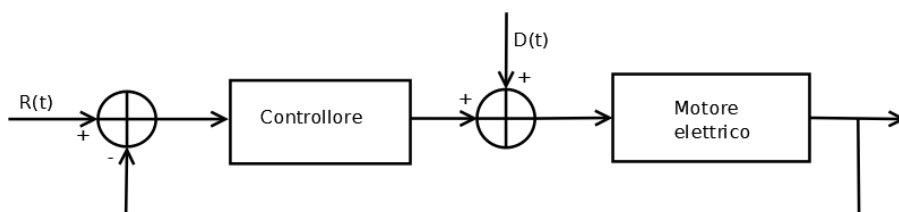
**Figura 3.30:** Diagrammi di Bode del sistema controllato con il compensatore progettato in frequenza.



## Capitolo 4

# Analisi dinamica del sistema in catena chiusa

In questo capitolo si andrà a simulare il comportamento del motore (in configurazione di catena chiusa) a seguito di disturbi che entrano in diversi punti del sistema. Verrà fatta una simulazione con alcuni segnali notevoli in ingresso all'armatura, verificando se la risposta del sistema così ottenuta è ancora rispettosa delle specifiche di controllo imposte nel capitolo precedente. Lo schema di riferimento è quello riportato in figura 4.1.



**Figura 4.1:** Schema a blocchi della configurazione di controllo con la presenza del disturbo.

In questo capitolo, tuttavia, si verificheranno gli effetti di disturbi che entrano anche in altri punti del sistema e non solo come somma all'ingresso di controllo.

### 4.1 Revisione del controllore PD - introduzione del PID

Come anticipato nella sezione sul controllore PD nel capitolo precedente, questo tipo di controllo è in grado di far rispettare la specifiche al motore

ma non di rimediare alla presenza di disturbi provenienti dall'esterno. Si ricorda che le funzioni di trasferimento del controllore e del motore sono rispettivamente

$$C(s) = 70 + 0,4s, \quad M(s) = \frac{K}{sK^2 + s(Ls + R)(Js + b)}.$$

La prima considerazione da fare quando si parla di disturbi è la seguente: tutta la progettazione fatta fino ad ora, non ha mai preso in considerazione il fatto che viene attaccato un carico all'albero del motore. Questa situazione è molto realistica chiaramente, perchè il motore è utilizzato per movimentare un qualche tipo di apparecchiatura collegata al suo albero. Un carico all'albero costituisce sostanzialmente una coppia che tende a lavorare contro quella erogata dal motore.

Per tenere conto di questo fatto è necessario modificare le equazioni caratteristiche del motore: l'equazione che esprime il secondo principio della dinamica (ovvero la relazione tra tutti momenti in gioco) era:

$$J\ddot{\theta}(t) + b\dot{\theta}(t) = Ki(t)$$

in cui  $Ki(t)$  è la coppia motrice erogata dal motore e gli altri due sono le coppie che lavorano contro il motore, ovvero il termine inerziale (proporzionale all'accelerazione angolare) e quello relativo all'attrito viscoso (proporzionale alla velocità angolare). Per modellizzare la presenza di un'ulteriore coppia resistente dovuta al carico, si introduce il termine  $T(t)$ , che esprime la coppia che il carico esercita sull'albero del motore. La nuova equazione diventa:

$$J\ddot{\theta}(t) + b\dot{\theta}(t) + T(t) = Ki(t). \quad (4.1)$$

Per simulare il comportamento del sistema in questo caso, si ricorre a Simulink. Il modello del motore è quello di figura 4.2.

Tutto questo schema ha due ingressi, ovvero la tensione di armatura  $V$  e la coppia resistente al carico  $T$ . L'uscita è la posizione angolare dell'albero. Dopo aver creato un subsystem con Simulink si ottiene uno schema in cui sono evidenziati solo ingressi e uscite, mentre tutto il nucleo centrale è nascosto. Il risultato è riportato in figura 4.3.

Lo scopo di questa sezione è quello di evidenziare i problemi del controllore PD nel gestire gli errori. Si inserisce dunque nello schema, il controllore descritto dall'equazione  $C(s) = K_P + K_Ds = 70 + 0,4s$ . come mostrato in figura 4.4.

Simulando il comportamento del sistema con questo controllore (con un disturbo di  $0,1 Nm$ ) si ottiene la risposta di figura 4.5.

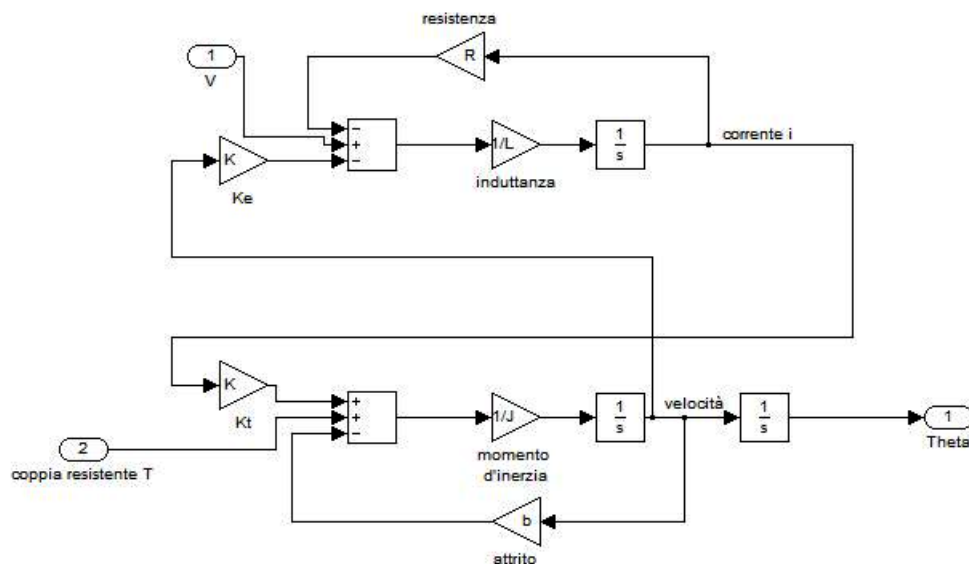


Figura 4.2: Implementazione in Simulink del motore elettrico in DC.

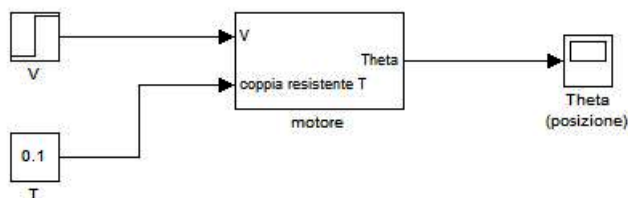


Figura 4.3: Subsystem creato con Simulink.

Quello che si vede è che il controllore non è in grado di posizionare correttamente il motore, ma a causa del disturbo questo si assesta intorno agli  $1,2 \text{ rad}$ . L'errore a regime dovuto ai disturbi è quindi non nullo. A mano a mano che l'intensità della coppia motrice aumenta, il motore si allontana sempre di più dalla posizione desiderata. Per  $T(t) = 1 \text{ Nm}$  ad esempio, il motore si posiziona a circa  $3 \text{ rad}$ .

Per rimediare a questa situazione, come già anticipato, è necessario usare anche l'azione integrativa del controllore PID. Il termine integrale infatti, somma all'ingresso di controllo l'area dell'errore, e questo da un ulteriore contributo di tensione a  $u(t)$ .

Dopo alcune simulazioni in Simulink, si è arrivati alla seguente configurazione

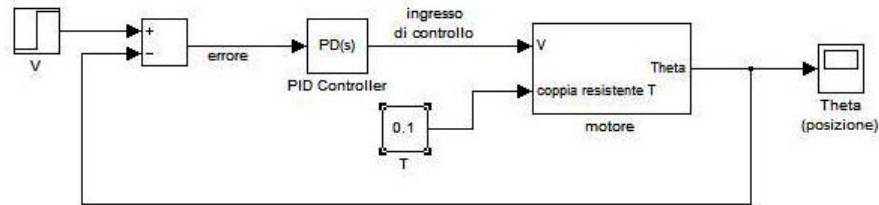


Figura 4.4: Schema Simulink del sistema del motore con controllore PD.

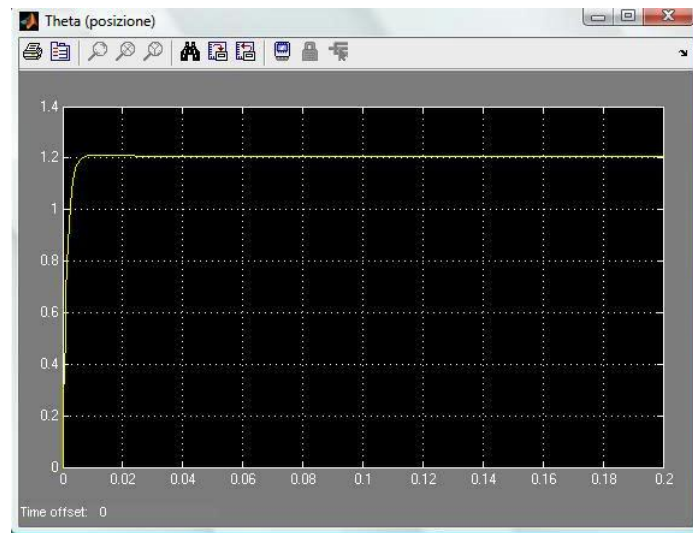


Figura 4.5: Risposta del sistema con il controllore PD e un disturbo  $T(t) = 0,1 \text{ Nm}$ .

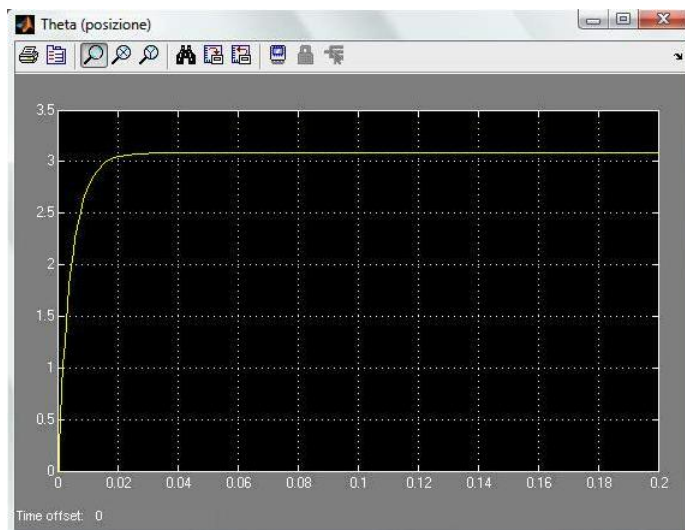
de controllore PID:

$$K_P = 2000 , K_I = 10000 , K_D = 4 .$$

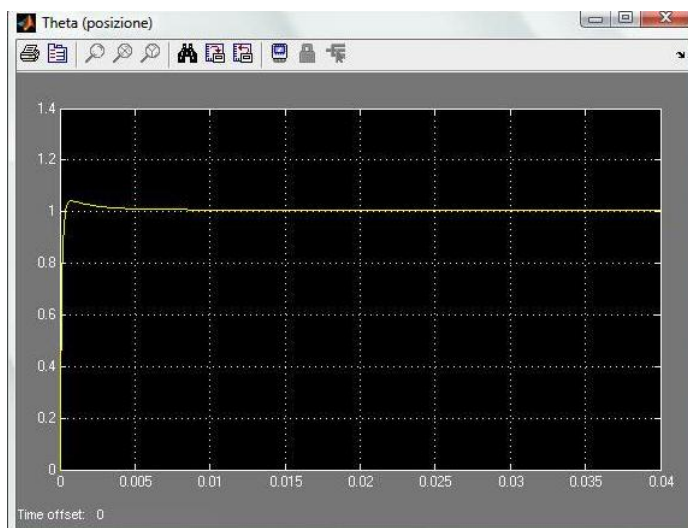
Il comportamento del sistema in questo caso è mostrato in figura

Finalmente il motore riesce a posizionarsi correttamente e con esso anche il carico. Il controllore PID definitivo ha allora l' espressione

$$C(s) = 2000 + 4s + \frac{10000}{s} . \tag{4.2}$$



**Figura 4.6:** Risposta del sistema con il controllore PD e un disturbo  $T(t) = 1 Nm$ .



**Figura 4.7:** Risposta del sistema con il controllore PID e un disturbo  $T(t) = 0,1 Nm$ .

## 4.2 Risposte indiciali in catena chiusa

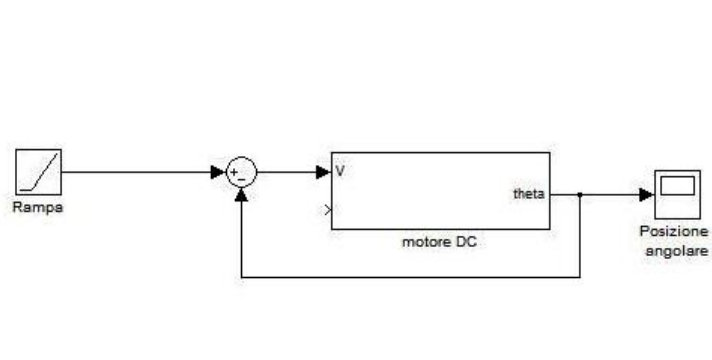
In questa sezione verranno riportate le risposte del sistema in catena chiusa con alcuni segnali notevoli in ingresso: si simulerà il comportamento per l'impulso, la rampa e a sinusoidale. Per quanto riguarda la risposta al gradino, questa è già stata abbondantemente trattata nel terzo capitolo. Questo tipo di prova è utile per verificare come si comporta il sistema in situazioni particolari come appunto sono quelle in cui la tensione in input è data come un impulso piuttosto che come una tensione alternata.

### 4.2.1 Risposta all' impulso

Avendo a disposizione il modello Simulink del motore è molto semplice verificarne la risposta all' impulso.

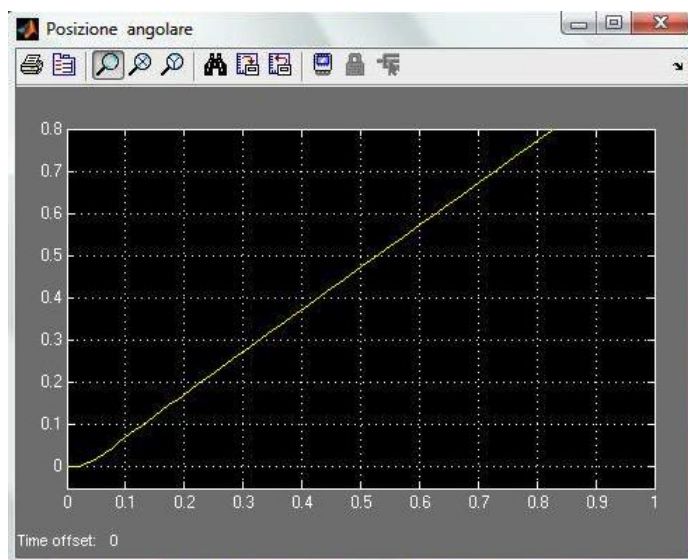
### 4.2.2 Risposta alla rampa

Con il modello Simulink mostrato in figura 4.8 si riesce facilmente a ricavare l'andamento della risposta.



**Figura 4.8:** Sistema in catena chiusa con una rampa unitaria come input.

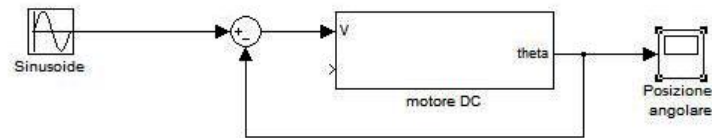
La risposta è quella di figura 4.9.



**Figura 4.9:** Sistema in catena chiusa con una rampa unitaria come input.

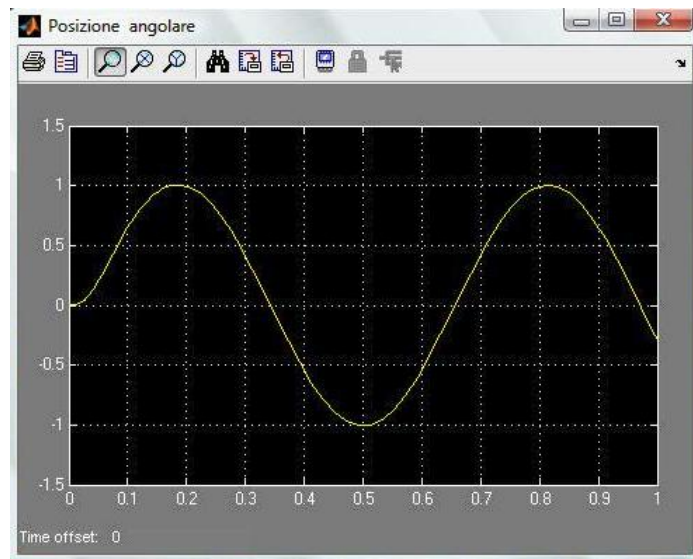
### 4.2.3 Risposta alla sinusoide

Cambiando il segnale di input nel modello Simulink si ottiene lo schema mostrato in figura 4.10 .



**Figura 4.10:** Sistema in catena chiusa con una sinusoide come input.

La risposta è mostrata in figura 4.11.



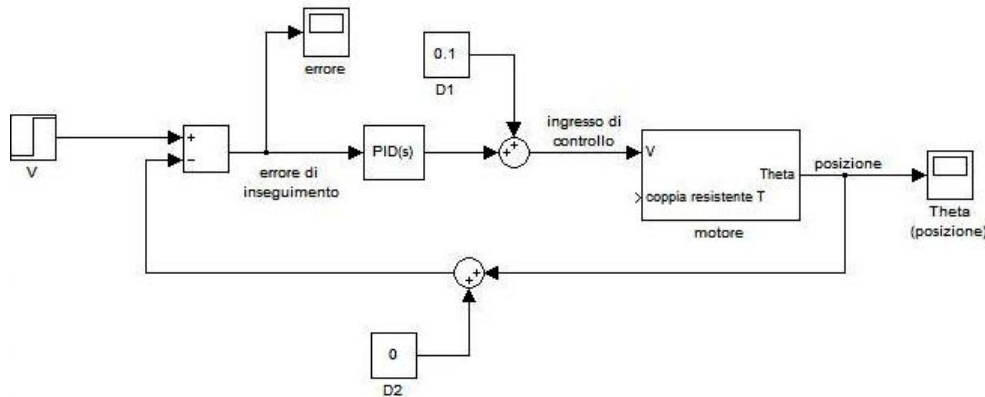
**Figura 4.11:** Risposta del sistema in catena aperta ad una sinusoide di frequenza  $10 \frac{rad}{s}$ .

Facendo variare la frequenza del segnale si ottengono chiaramente risposte diverse.



### 4.3 Disturbi entranti nel sistema in punti diversi

In questa sezione si simulerà il comportamento del sistema nel caso in cui ci siano disturbi entranti in punti diversi del sistema. Lo schema Simulink utilizzato è quello di figura 4.12.



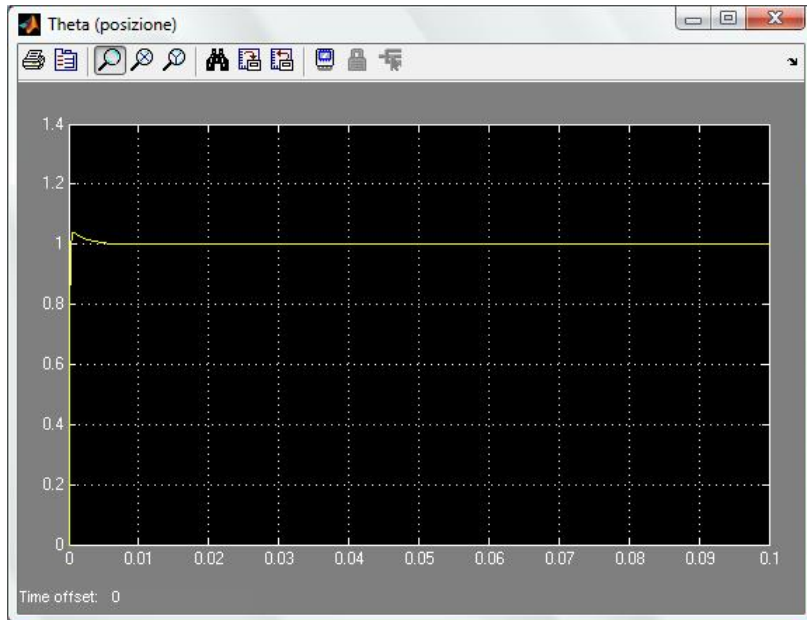
**Figura 4.12:** Schema Simulink contenente disturbi entranti in punti diversi della configurazione.

Nel caso in cui il disturbo  $D1 \neq 0$  e  $D2 = 0$  l'esito della simulazione è quello riportato in figura 4.13.

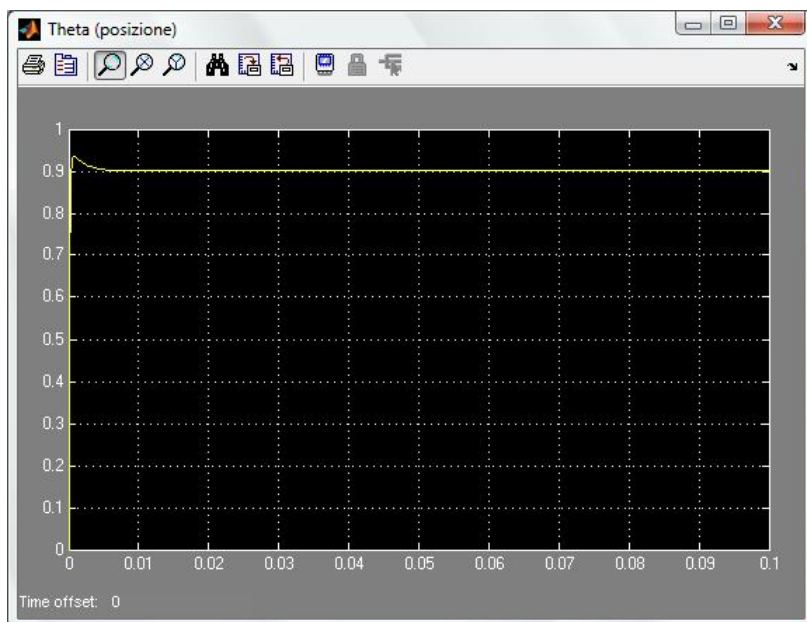
Il sistema è evidentemente in grado di rimediare ad un disturbo entrante in quella posizione.

Più delicata è la situazione in cui  $D1 = 0$  e  $D2 \neq 0$ . In questo caso la risposta ha l'andamenti riportato in figura 4.14.

Si vede che l'albero ruota di  $0,9 \text{ rad}$ . Questo errore è causato dal fatto che il disturbo di valore 0.1 si sottrae direttamente al riferimento, il quale diventa dunque un gradino di ampiezza  $0,9 \text{ V}$  anziché  $1 \text{ V}$ . Il sistema non vede la presenza del disturbo, quindi, perchè questo agisce direttamente sul riferimento iniziale.



**Figura 4.13:** Risposta con  $D1 = 0.1 \text{ Nm}$  e  $D2$  nullo.



**Figura 4.14:** Risposta con  $D1$  nullo e  $D2 = 0.1$ .

# Conclusione

Lo scopo di questa tesina è lo studio di diversi sistemi di controllo per il motore elettrico. Pur senza entrare nel dettaglio e senza usare tecniche particolari di controllo, le difficoltà non sono state poche. I problemi sono stati diversi: il primo ad essere affrontato è stato la modellizzazione del motore, cioè il passaggio dal funzionamento fisico ad un modello matematico che lo descrivesse. Il secondo problema (e sicuramente il più complicato) è stato mettere in pratica quello che ho imparato sui controlli automatici nel corso di questi primi tre anni di università. Il passaggio dai problemi affrontati a lezione in maniera del tutto didattica ad un'applicazione reale non è banale. Uno scopo secondario di questa tesina, infine, è stato quello di imparare ad usare Matlab e a scrivere utilizzando l'ambiente Latex. Il risultato è soddisfacente e senz'altro è stata un'esperienza utile dalla quale ho imparato molto sul mondo dei controlli automatici.



# Bibliografia

- [1] S.Zampieri ,Dispensa di controlli Automatici, Edizioni Libreria Progetto Padova, 2011;
- [2] R.C. Dorf, R.H.Bishop, Controlli Automatici, Pearson, 2010;
- [3] <http://www.engin.umich.edu/class/ctms/>, Control Tutorials for Ma- tlab and Simulink.