

Capitolo 6

Controllo lineare

6.1 Elementi del sistema di controllo

In questo capitolo si studieranno varie strategie di controllo per migliorare le prestazioni del sistema di visual servoing ad anello chiuso presentato nel paragrafo 5.2.

Partendo dallo schema a blocchi di figura 5.1 ed analizzando sia la struttura del sistema hardware sia l'architettura del software si è potuto realizzare uno schema funzionale dell'intero sistema. Attraverso quindi l'utilizzo del pacchetto di simulazione *Simulink* dell'ambiente *Matlab* si è passati dallo schema funzionale ad un modello digitale (rappresentato in figura 6.1¹) che ha permesso di simulare il comportamento dinamico dell'intero sistema di asservimento visuale.

Per ottenere una tale schematizzazione si sono dovute considerare solo quelle grandezze e quelle trasformazioni che condizionano la dinamica del sistema, tralasciando tutti gli altri dettagli implementativi.

In questo schema viene considerato come ingresso il movimento dell'oggetto rappresentato dal segnale \mathbf{q} e come variabile di uscita la posizione \mathbf{p} del braccio nello spazio del robot (posizione assoluta). In realtà la vera variabile di uscita, intesa come variabile misurata, sarebbe l'immagine ripresa dalla telecamera ovvero il segnale \mathbf{y} .

¹Un ulteriore modello è presente in [22].

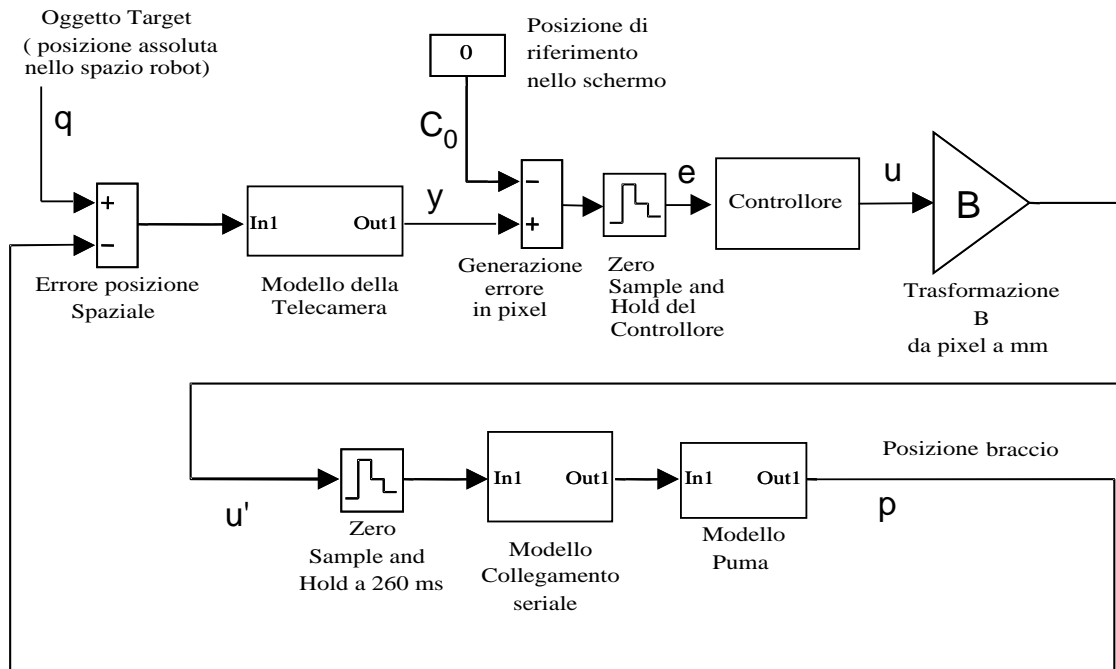


Figura 6.1: Schema logico dell'intero sistema.

Modello della telecamera

Questo blocco riassume diverse operazioni realizzate dall'insieme degli elementi costituito da: telecamera, frame grabber, libreria MIL, IMEL e programma di controllo. La telecamera campiona un'immagine ogni $T_{frame} = 33ms^2$ ed introduce una cambiamento di scala (dipendente dalla distanza a cui è posto l'oggetto ripreso ed eventualmente dalla diversa dimensione del pixel nei due assi) che può semplicemente essere riassunto da un guadagno chiamato B^{-1} .

Il blocco di passaggio da pixel a millimetri, costituito da un semplice guadagno rappresentato da B, effettua l'operazione inversa delle telecamera. Il valore di B^3 viene ricavato attraverso la procedura di auto-calibrazione descritta nel

²Questo è il sample time della telecamera che come vedremo coincide praticamente con il sample time del robot Puma 260.

³L'esempio qui riportato è unidimensionale, nella realtà però trattando una immagine a due dimensioni si opererà con grandezze vettoriali per cui il guadagno B è rappresentato da una matrice 2×2 .

paragrafo 3.4.2; questa operazione annulla la trasformazione introdotta dal *blocco telecamera* e permette di poter progettare il controllore indipendentemente dalla particolare telecamera che si utilizza⁴.

Zero-Order Hold a sample time

Questo blocco rappresenta l'intervallo temporale che trascorre tra due successivi campionamenti della variabile $e(t)$, questi campionamenti vengono concretamente effettuati dalla routine software che implementa il controllore.

Controllore

Questo è il blocco che realizza il controllo del sistema e condiziona quindi la dinamica del braccio robotico. La sua implementazione è puramente software e segue fedelmente la formulazione matematica sviluppata in 6.3.1.

Zero-Order Hold a sample time di 260ms

Questo blocco riassume il segnale di controllo spedito dalla seriale del PC verso il controllore Unival. Essendo il tempo di aggiornamento di questo segnale di circa 260 – 290 ms considereremo in tale intervallo temporale il segnale costante, tale segnale può essere semplicemente modellato da un *sample and hold* con un *sample time* che chiameremo STS (Sample time seriale) appunto di 260 ms.

Modello Collegamento Seriale

Questo blocco deve modellare il ritardo dovuto al trasferimento dei dati con la seriale che come già detto ammonta circa a 260 ms.

Modello Puma

Questo è il blocco fondamentale di tutto lo schema, determina il *sample time* dal quale vengono generati tutti gli altri blocchi. Dovendo rappresentare il Puma 260 operante in modalità *Alter Mode* cumulativa, ricordando che l'aggiornamento delle variazioni della posizione avviene ogni 32 ms, si prenderà tale tempo come

⁴Il discorso vale anche per l'orientamento con cui la telecamera è montata sull'organo terminale del braccio.

sample time T_S . Si può allora rappresentare il blocco come mostrato in figura 6.2.

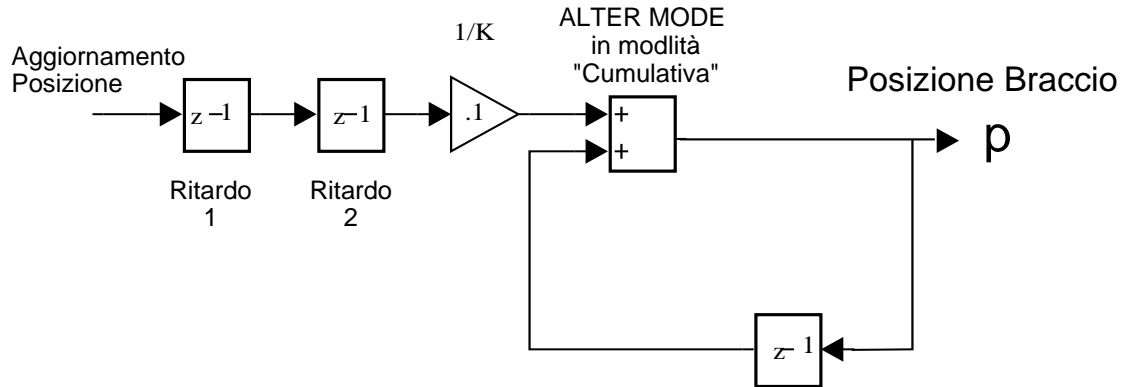


Figura 6.2: Modello del Puma 260 in Alter Mode.

I due blocchi di ritardo all'ingresso (ognuno dei quali con un T_S di 32 ms), modellano il ritardo temporale dovuto al calcolo della traiettoria da parte dell'Unival che ammonta a circa 70ms. L'anello di retroazione invece rappresenta lo schema di funzionamento dell'Alter Mode in modalità cumulativa. La attenuazione di 10 volte ($K=10$ nel programma Val II) dell'aggiornamento della posizione permette di ottenere il raggiungimento della posizione desiderata dopo un tempo pari a $T_S \cdot K = 320ms$.

Notare che, essendo STS di 260 ms, il controllore Unival riceverà per questo intervallo un valore costante che determinerà a sua volta una velocità costante nel movimento del Robot Puma.

Avendo preso come *sample time* di riferimento la tempistica del Puma 260 ($T_S = 32ms$) è possibile rappresentare il ritardo della seriale secondo lo schema di figura 6.3.

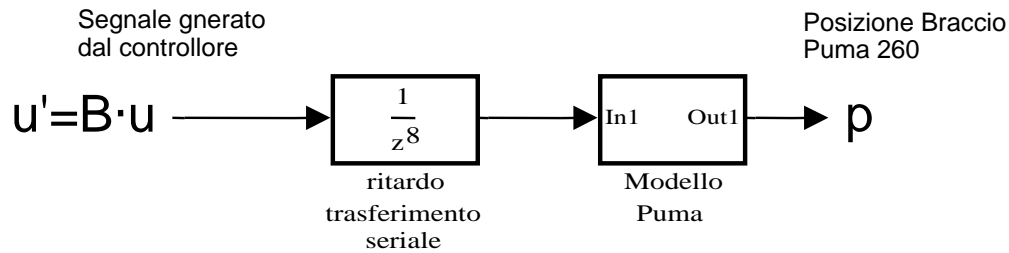


Figura 6.3: Modello del sistema comprensivo del ritardo della seriale.

6.2 Risposta del sistema ad anello aperto

È stato sollecitato il sistema reale lungo un solo asse con un impulso a gradino di ampiezza 1 mm e per un tempo pari circa a 2.6 secondi. Analizzando la risposta si è potuto ricavare il guadagno del sistema ad anello aperto $K_{open} \cong 3.15^5$ peraltro confermata dalla simulazione tramite il modello appena presentato (figura 6.4, grafico di destra). Il valore di questo guadagno deriva dal rapporto tra l'unità di tempo su cui si calcola la variazione dell'uscita (1 sec.) ed il sample-time del Puma 260 ($T_s = 0.032\text{sec.}$) considerando anche l'attenuazione introdotta dal coefficiente K ($1/K = 0.1$), risulta quindi $\frac{1\text{sec.}}{K \cdot T_s} = 3.15$.

La somma dei ritardi dovuti all'acquisizione ed elaborazione dell'immagine, alla trasmissione dei dati tramite seriale, all'elaborazione del controllore Unival per il calcolo della traiettoria, porta ad un ritardo nel tempo di risposta che come si può vedere dalla figura 6.4 di sinistra oscilla intorno ai 350 ms e, come rilevato da innumerevoli prove, non scende mai al di sotto dei 320ms.

Nota

Tutti i grafici riportati da ora in avanti si riferiranno al solo asse x del sistema di riferimento della telecamera, per l'asse y varranno considerazioni del tutto identiche.

⁵Osservando il grafico di sinistra della figura 6.4, compare una discrepanza rispetto ai risultati ricavati in [16] dovuta all'attenuazione introdotta dal coefficiente K.

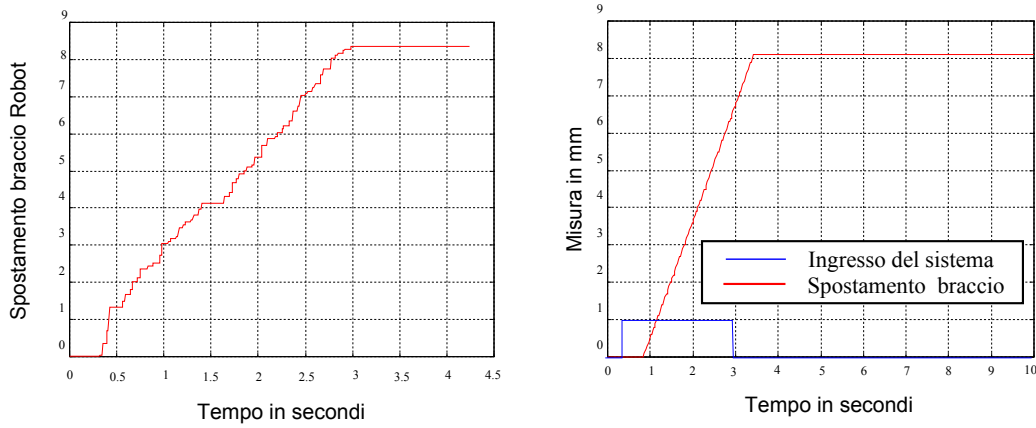


Figura 6.4: Risposta ad anello aperto del sistema.

6.3 Risposta del sistema ad anello chiuso

Analizziamo ora l'*errore* del sistema quando viene chiusa la retroazione visiva prendendo come indice di bontà del controllo il *tempo di assestamento* T_a . L'*errore* è lo scostamento della posizione dell'oggetto ripreso dalla telecamera (segnale *Out1* del modello telecamera) rispetto ad una posizione di riferimento (in figura 6.1 indicato come c_0) e viene convertito da pixel in mm attraverso la matrice B.

6.3.1 Controllo digitale

Un generico controllo PID (figura 6.5) nel dominio dei tempi è così espresso

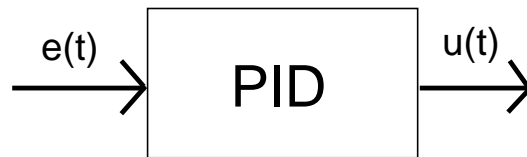


Figura 6.5: Rappresentazione schematica di un PID.

$$u(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right) \quad (6.1)$$

trasformando secondo Laplace si ottiene

$$U(s) = K_p \left(1 + \frac{1}{T_i \cdot s} + T_d \cdot s \right) E(s) \quad (6.2)$$

essendo in questa forma la componente derivativa non fisicamente realizzabile si introduce un polo ad alta frequenza (determinato dal valore di N che in genere è compreso tra 3 e 10)

$$U(s) = K_p \left(1 + \frac{1}{T_i \cdot s} + \frac{T_d \cdot s}{1 + \frac{T_d \cdot s}{N}} \right) E(s) \quad (6.3)$$

A questo punto è possibile dare una rappresentazione discreta (digitale) della 6.3 utilizzando le approssimazioni chiamate “*differenza in avanti*” per la parte integrale

$$s = \frac{z - 1}{T} \quad (6.4)$$

e della “*differenza all'indietro*” per il termine derivativo⁶

$$s = \frac{z - 1}{T \cdot z} \quad (6.5)$$

otteniamo sostituendo la 6.4 e la 6.5 nella 6.3 un controllore PID in forma digitale espresso dalla funzione di trasferimento $D(z)$

$$D(z) = \frac{U(z)}{E(z)} = K_p \left(1 + \frac{T}{T_i(z - 1)} + \frac{T_d}{T + \frac{T_d}{N}} \frac{z - 1}{\left[z - \frac{T_d}{NT + T_d} \right]} \right) \quad (6.6)$$

T rappresenta il periodo di campionamento (sample time) del controllore. Nel nostro caso T è limitato inferiormente dalla velocità di acquisizione delle immagini da parte del binomio telecamera-frame grabber, per cui non ha senso che sia inferiore a $T \approx 33ms$.

6.3.2 Controllo proporzionale

Partiamo dall'analisi dei risultati ottenuti mediante un semplice controllore proporzionale, espresso dalla relazione

$$D(z) = \frac{U(z)}{E(z)} = K_p \quad (6.7)$$

⁶Vedere a questo proposito [26] pag. 150,153,250.

ricavata tramite la soppressione delle componenti derivativa ed integrale dalla 6.6. In figura 6.6 vengono presentate diverse risposte del sistema ad un ingresso a gradino per differenti valori del guadagno K_p .

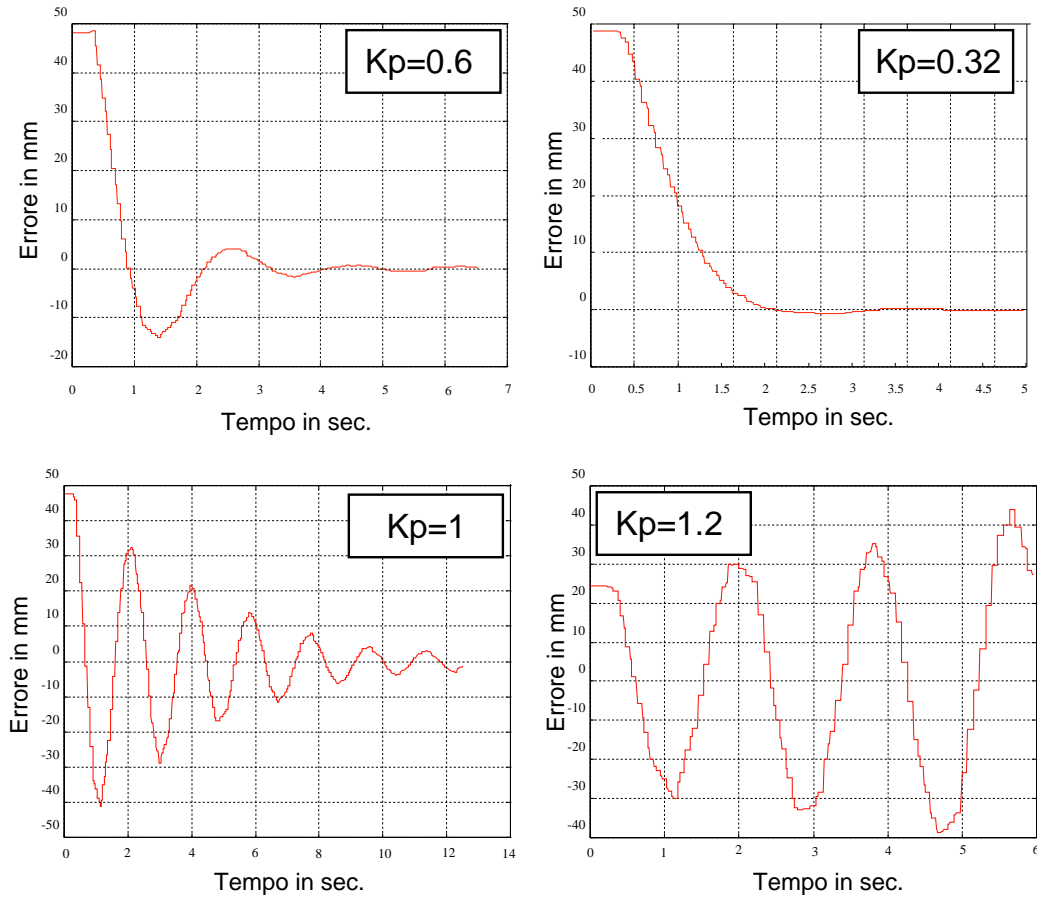


Figura 6.6: Andamento dell'errore nel tempo in risposta ad un impulso.

Notare che nei casi in cui il sistema retroazionato sia stabile (escluso quindi il caso con $K_p = 1.2$) l'errore a regime è nullo. Questo è un comportamento desiderato ed è dovuto al fatto che il robot Puma 260 è assimilabile ad un integratore⁷, cosa per altro evidente dal modello di figura 6.2.

Si sottolinea il caso in cui $K_p = 0.32$ guadagno sopra il quale si iniziano ad avere delle oscillazioni intorno al valore di regime ed in cui il *tempo di assestamento* vale circa 1.5 sec. Cercando di migliorare la dinamica dell'errore, applicando un tuning manuale, all'aumentare di K_p si ottiene un *tempo di salita*

⁷Vedi per questa considerazione sistemistica [21, 22].

inferiore ma il tempo T_a inizia ad allungarsi, questo è il limite delle prestazioni ottenibili con un semplice controllo proporzionale.

Per migliorare le cose si deve introdurre un *derivatore* cioè bisogna effettuare un controllo di tipo *PD*.

6.3.3 Controllore PD

Dovendo realizzare un controllore proporzionale-derivativo, si elimina dalla 6.6 la sola componente integrale ottenendo

$$D(z) = \frac{U(z)}{E(z)} = K_p \left(1 + \frac{T_d}{T + \frac{T_d}{N}} \cdot \frac{z - 1}{\left[z - \frac{T_d}{NT + T_d} \right]} \right) \quad (6.8)$$

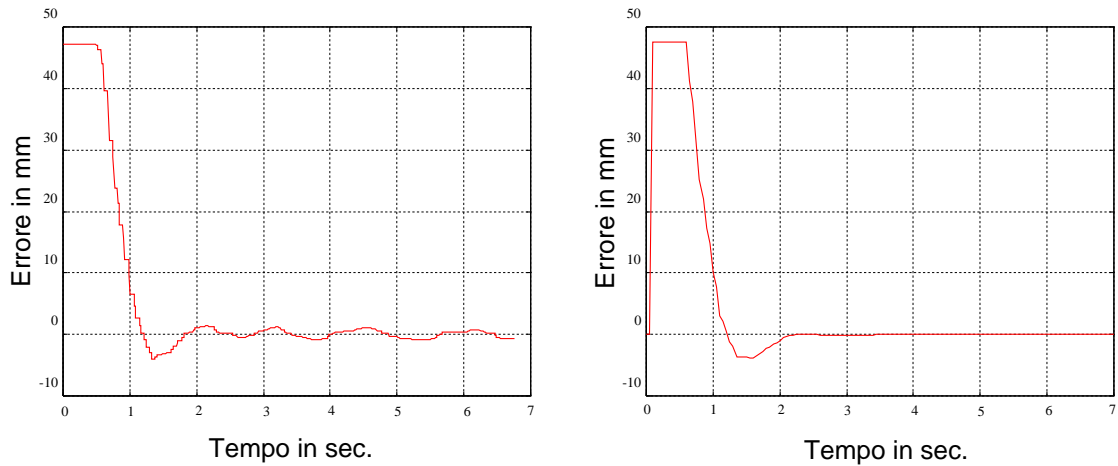


Figura 6.7: Prestazioni regolatore PD e corrispondente modello simulato.

ponendo per compattezza

$$\begin{aligned} \gamma &= \frac{T_d}{N \cdot T + T_d} \\ \beta &= N \cdot \gamma \end{aligned} \quad (6.9)$$

si può riscrivere la 6.8 più semplicemente come

$$U = K_p \left(1 + \frac{\beta(z - 1)}{z - \gamma} \right) E \quad (6.10)$$

facendo il minimo comune multiplo e raccogliendo opportunamente

$$U \cdot z - U \cdot \gamma = K_p (z \cdot (1 + \beta) - (\gamma + \beta)) E \quad (6.11)$$

moltiplicando ambo i membri per z^{-1} e ricordando che questo termine *ritarda di un periodo* il campione a cui è applicato, possiamo ricavare la forma *alle differenze* della 6.11 pronta per essere implementata al calcolatore; per un generico istante k si ottiene definendo le seguenti grandezze

$$\begin{aligned} q_0 &= K_p(1 + \beta) \\ q_1 &= -K_p(\gamma + \beta) \\ r_1 &= \gamma \end{aligned} \quad (6.12)$$

la seguente equazione alle differenze

$$U_k = U_{k-1} \cdot r_1 + E_k \cdot q_0 + E_{k-1} \cdot q_1 \quad (6.13)$$

Implementando direttamente la 6.13 si è ottenuto un regolatore con l'andamento mostrato nella figura 6.7 di sinistra, dove con $T = 0.070s$, $N = 10$, $K_p = 0.5$, $T_d = 0.17s$ si ottiene un $T_a \approx 0.8$ secondi, nel grafico di destra invece è mostrata la simulazione del corrispondente modello.

Il *tempo di assestamento* rispetto al caso proporzionale (con $K_p = 0.6$ vedi fig.6.6), si è praticamente dimezzato e questo evidenzia l'importanza dell'aggiunta di una parte derivativa nel controllore.

6.3.4 Controllore PID

La dinamica fin qui studiata si riferisce ad un ingresso a gradino, cioè ad un movimento repentino dell'oggetto da inseguire. Il sistema, grazie alla sua implicita parte integrale rappresentata dal Robot in Alter Mode, tende ad inseguire e raggiungere ogni oggetto *fermo* e presente nel campo visivo della telecamera. Questo vuol dire che se un oggetto compie uno spostamento finito, dopo un breve transitorio il robot posiziona la telecamera sulla verticale dell'oggetto. Questo non è più vero quando l'oggetto invece di fermarsi continua a muoversi con velocità costante (caso per altro molto comune su un nastro trasportatore), in tale eventualità l'errore a regime del sistema non si annulla e sia il controllore proporzionale che il PD non hanno più un comportamento soddisfacente⁸.

⁸Vedere il paragrafo 6.5.6 per i dettagli matematici.

Infatti scrivendo la funzione di trasferimento (discreta) del robot rappresentato in figura 6.2 risulta

$$A(z) = z^{-2} \cdot \frac{K^{-1} \cdot z}{z-1} \cdot \frac{T_s}{T_{i_{alter}}} \quad (6.14)$$

Dove $T_s = 32ms$ è il sample time del sistema mentre $T_{i_{alter}}$ indica la costante di integrazione (infatti il sistema è un integratore) anch'essa pari a $T_{i_{alter}} = 32ms$.

La 6.14 rappresenta un sistema di tipo 1, ovvero un sistema che presenta un polo in $z=1$ ⁹. Questa è una condizione sufficiente per annullare l'errore $\mathbf{e}(\mathbf{t})$ a regime con un ingresso a gradino, ma non basta per un *ingresso a rampa*, cioè quando il sistema insegue un oggetto che si muove a velocità costante.

Per risolvere questo problema basta aumentare il *tipo di sistema* introducendo un integratore nel controllore e quindi realizzare un PID. In questo modo il blocco complessivo formato dal controllore e dal robot diviene un *sistema di tipo 2* e si annulla a regime sia l'errore di posizione che di velocità.

Per progettare quindi un PID, si procede come fatto precedentemente per il PD, partendo dalla 6.6 chiamando per semplicità di notazione

$$\begin{aligned} \alpha &= \frac{T}{T_i} \\ \beta &= N \cdot \gamma \\ \gamma &= \frac{T_d}{N \cdot T + T_d} \end{aligned}$$

si ricava la relazione

$$U = \left(\frac{K_p(1 + \beta) \cdot z^2 - K_p(1 + \gamma - \alpha + 2\beta) \cdot z + K_p(\gamma - \alpha\gamma + \beta)}{(z-1) \cdot (z-\gamma)} \right) E \quad (6.15)$$

e chiamando ancora

$$\begin{aligned} q_0 &= K_p(1 + \beta) \\ q_1 &= -K_p(1 + \gamma - \alpha + 2\beta) \\ q_2 &= K_p(\gamma - \alpha\gamma + \beta) \\ r_1 &= \gamma + 1 \\ r_2 &= \gamma \end{aligned}$$

⁹Nel dominio di Laplace (variabile s) il *tipo di sistema* è determinato dal numero di poli nell'origine, vedere [25].

si ottiene dopo alcuni passaggi il controllore sotto forma di equazioni alle differenze. Per un generico istante k si può allora scrivere

$$U_k = U_{k-1} \cdot r_1 - U_{k-2} \cdot r_2 + E_k \cdot q_0 + E_{k-1} \cdot q_1 + E_{k-2} \cdot q_2 \quad (6.16)$$

La 6.16 insieme alla 6.13 sono state direttamente utilizzate nell'implementazione software dei due tipi di regolatori appena descritti.

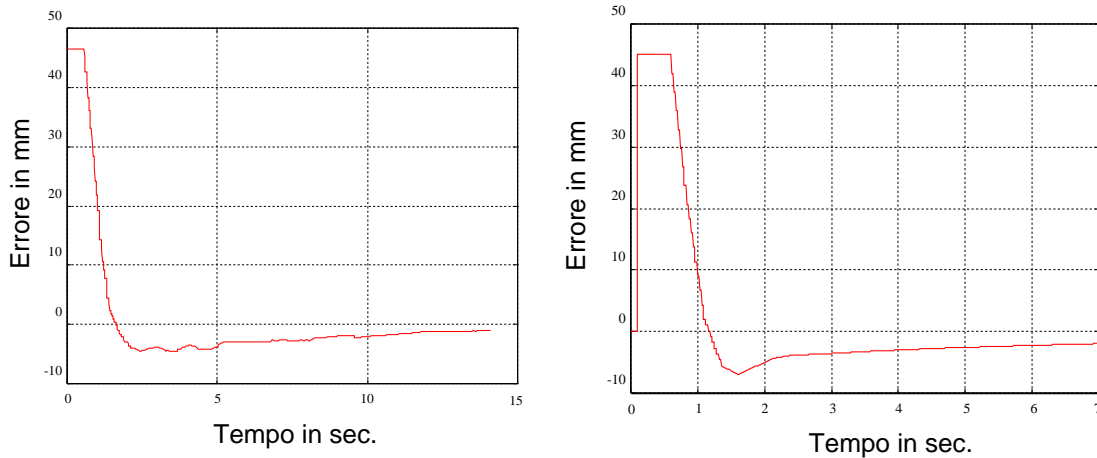


Figura 6.8: Esempio di risposta ad un gradino con un regolatore PID.

In figura 6.8 il grafico di sinistra indica l'errore commesso utilizzando un regolatore PID con parametri $T = 0.13s$, $K_p = 0.4$, $T_i = 4s$, $T_d = 0.17s$, $N = 10$ con un ingresso a gradino di $45mm$, nel grafico di destra invece si trova la corrispondente simulazione. Appare evidente che le prestazioni sono notevolmente peggiorate riguardo il *tempo di assestamento* T_a dell'errore di posizione rispetto ad un *target* fermo, ma osservando la figura 6.9 dove sono graficate le traiettorie nello spazio del robot sia dell'*end-effector* che del *target*, si può notare l'annullamento dell'errore di posizione dovuto all'azione integrale, infatti dopo circa 6 secondi dal momento in cui l'oggetto inizia a muoversi, la posizione del *target* e quella dell'*end-effector* (telecamera) coincidono.

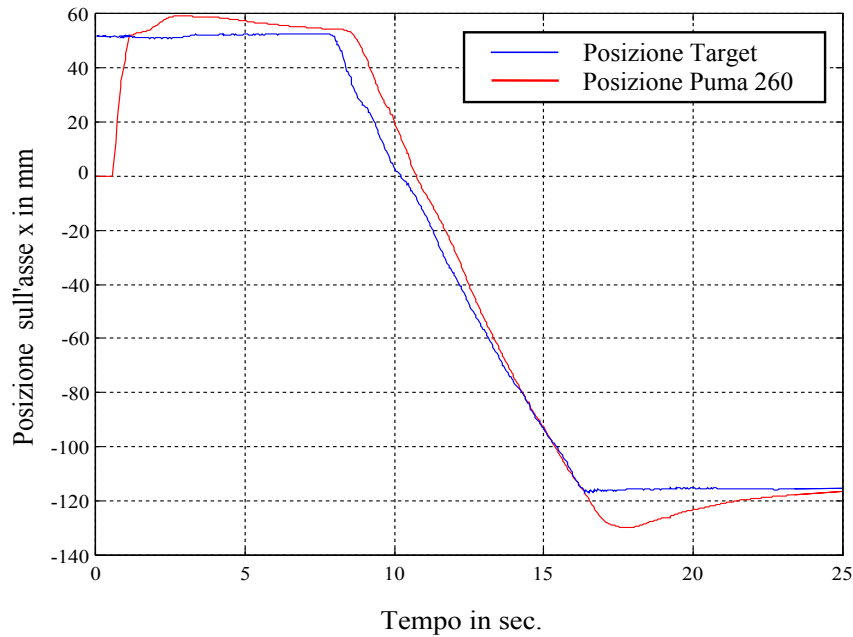


Figura 6.9: Risposta del PID ad una rampa.

6.4 Profilo di velocità

Come sottolineato più volte, la banda di controllo del robot è limitata superiormente dalla velocità di comunicazione della seriale che permette di poter aggiornare la posizione del robot ogni 260 ms circa. Per come è stato implementato l'aggiornamento della posizione del robot, il dato inviato al controllore Unival è in realtà una velocità, infatti inviando ad un istante j un aggiornamento della posizione di S_j millimetri, otterremo

$$V_j = \frac{S_j \text{ mm}}{320 \text{ msec}} \quad (6.17)$$

dove V_j è la velocità espressa in mm per millisecondi che terrà il robot per i prossimi 320 millisecondi¹⁰. Nell'eventualità che l'oggetto da inseguire abbia un movimento discontinuo, la componente derivativa del regolatore introduce una risposta impulsiva del braccio robotico ($V_j \gg V_{j+1}$) che tende a sollecitare in maniera considerevole la telecamera montata sull'organo terminale del Puma 260, per cui un tipico profilo di velocità ottenuto dal controllo implementato attraverso il programma VAL II di nome "Goa" è mostrato in figura 6.10.

¹⁰Valore discendente dalla relazione $K \cdot T_S = 320ms$ dove $K=10$ e $T_S=32ms$.

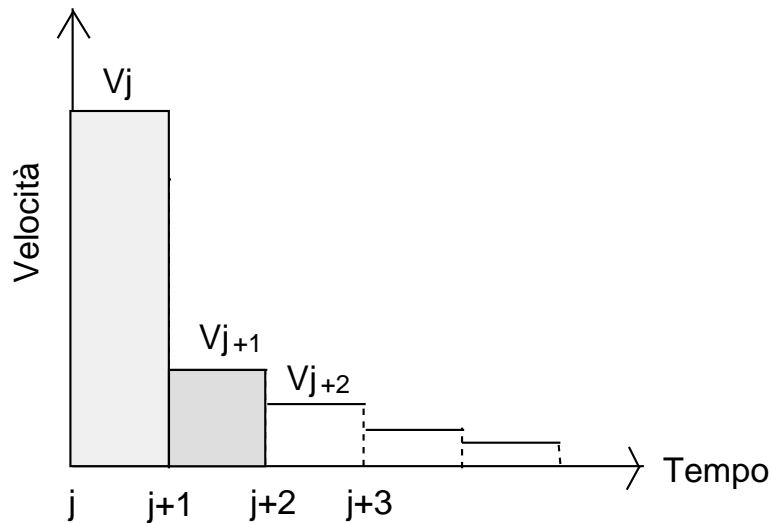


Figura 6.10: Tipico profilo di velocità del programma *Goa*.

Quindi, come appena accennato, variazioni così repentine di velocità ($V_j \gg V_{j+1}$) introducono effetti dinamici imprevisti rappresentati da oscillazioni del blocco braccio-telecamera. In generale tali effetti non hanno influenza sulla stabilità del sistema questo perché il tempo di assestamento di tali oscillazioni è notevolmente inferiore alle tempistiche del controllo (260ms), ma una loro eliminazione porterebbe comunque ad un movimento più omogeneo con una diminuzione delle sollecitazioni che il sistema deve assorbire e quindi con una conseguente diminuzione dell'usura degli elementi che lo compongono¹¹.

6.4.1 Raccordo continuo di velocità

Una soluzione a questo problema è stata ottenuta con un intervento sul profilo di velocità raccordando, attraverso una variazione lineare, due velocità differenti. Per raggiungere tale scopo è necessario poter aggiornare repentinamente il valore della velocità del braccio, risultato ottenibile quindi solo spostando a valle della seriale parte del controllo. Questo ha aumentato leggermente la complessità del programma di controllo velocità realizzata in VAL II, ma solo così è stato possibile calcolare in real-time le accelerazioni (decelerazioni) opportune.

¹¹Uno studio di questo tipo diventerebbe necessario nel momento in cui la velocità del controllo diventasse paragonabile al tempo di assestamento delle oscillazioni dovute all'inerzia del blocco braccio-telecamera.

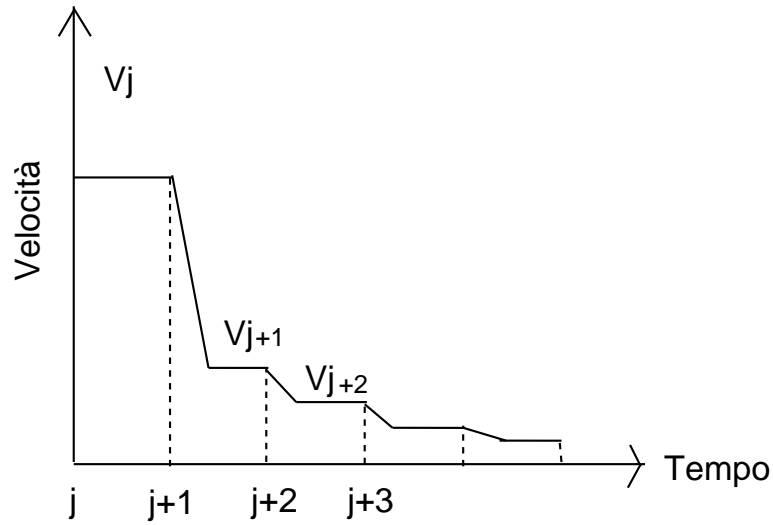


Figura 6.11: Profilo di velocità con raccordi lineari.

In figura 6.11 è evidenziato il profilo di velocità generato dal programma VAL II di nome *goa1*, questo programma impone per un certo tempo una accelerazione (decelerazione) costante, permettendo al braccio robotico di non subire sollecitazioni impulsive ma controllate. Notare che il programma calcola tale profilo imponendo che lo spazio percorso, dall'inizio del generico intervallo di controllo j fino al trascorrere di 320ms, sia comunque S_j ¹².

¹²Per i dettagli matematici vedere l'appendice B.

6.5 Il predittore

Un normale controllore come quelli fino ad ora trattati, non è in grado di poter compensare adeguatamente gli effetti derivanti da sistemi con un grosso ritardo presenti lungo l'anello di controllo. Le prestazioni ottenute sono state condizionate dal basso guadagno statico K_p ($K_p < 1$) necessario per garantire la stabilità del sistema in retroazione¹³.

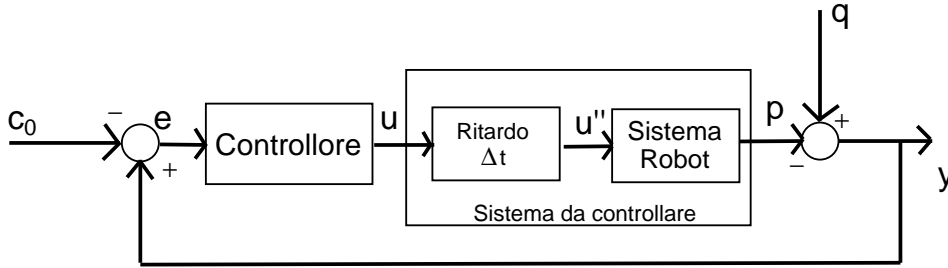


Figura 6.12: Schema di principio del controllo effettuato.

In figura 6.12 viene mostrata una versione più particolareggiata dello schema di figura 6.1 mettendo in risalto le sole grandezze che condizionano direttamente il controllo e trascurando invece i cambiamenti di coordinate indotti dalla telecamera (\mathbf{B} e \mathbf{B}^{-1}) di conseguenza risulta u' coincidente con u .

Una volta acquisita l'immagine ad un dato tempo t^* e ricavato l'errore

$$e(t^*) = (q(t^*) - p(t^*)) - c_0$$

c_0 e $q(t^*)$ verranno considerati a valore costante¹⁴ ma, mentre c_0 è determinato dal progettista, il valore di $q(t^*)$ è incognito.

Tramite il regolatore è allora possibile generare il segnale di controllo $u(t^*)$, tutto questo si ottiene in un tempo trascurabile (10ms-20ms) vista la considerevole potenza di calcolo a disposizione. Tuttavia, prima che il segnale $u(t^*)$ giunga al robot, deve trascorrere un tempo considerevole Δt a causa del ritardo della seriale, nel frattempo il robot avrà compiuto uno spostamento rispetto alla posizione iniziale $p(t^*)$, dovuto ai valori passati del segnale di controllo u , raggiungendo quindi una posizione $p(t^* + \Delta t)$. Per cui al tempo $t^* + \Delta t$ il sistema si

¹³A questo proposito vedere il paragrafo 4.7 di [25] per quantificare l'effetto della presenza di "sistemi a ritardo finito" sui margini di fase, ampiezza e sul guadagno statico K_p .

¹⁴Per ora quindi l'oggetto da inseguire viene ritenuto fermo.

appresta ad eseguire una correzione u'' della traiettoria $p(t)$ valutando di essere nella posizione $p(t^*)$ ed invece trovandosi in una posizione effettiva $p(t^* + \Delta t)$.

L'errore e_p commesso nella valutazione della posizione diviene allora

$$e_p = e(t^* + \Delta t) - e(t^*) = B \cdot (p(t^* + \Delta t) - p(t^*)) \quad (6.18)$$

Dove B è la costante moltiplicativa dovuta al cambiamento di scala introdotto dalla telecamera¹⁵, ma come anticipato si trascurerà tale trasformazione ponendo $B = 1$.

Per annullare questo errore bisogna utilizzare un blocco funzionale in grado di poter prevedere una volta acquisito un errore $e(t^*)$ quale sarà la posizione p del robot all'istante $t^* + \Delta t$ e calcolare di conseguenza la correzione u'' adeguata. Notando che, come già sottolineato, il comando u'' arriva al robot solo dopo un tempo Δt questo suggerisce che **in tale intervallo temporale la posizione dell'end-effector è condizionata solo dai valori passati del segnale u''** e conoscendo il valore di questi ecco allora che diviene possibile il calcolo a priori di una posizione futura del robot.

Per poter realizzare concretamente tutto ciò bisogna compiere due passi, per primo realizzare un sistema dinamico in grado di poter riprodurre il comportamento del sistema reale a parità degli ingressi, secondo lo schema di figura 6.13

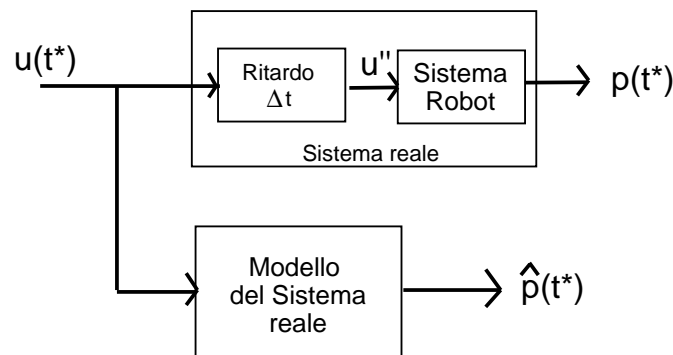


Figura 6.13: Utilizzo del *modello del sistema reale*.

di modo che

$$\hat{p}(t) \equiv p(t) \quad \forall t$$

¹⁵Vedere a questo proposito il discorso fatto a pag.86.

Di tale sistema è naturalmente accessibile ad ogni istante t oltre che l'uscita \hat{p} anche lo stato che verrà indicato con $\hat{x}_0(t)$.

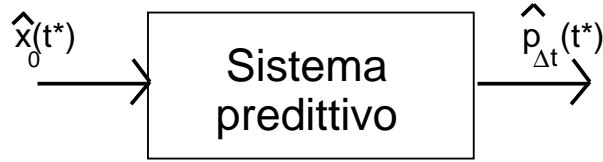


Figura 6.14: Secondo sistema componente del predittore.

L'ultimo passo invece consiste nella realizzazione di un secondo sistema, indicato in figura 6.14, che introduca la possibilità¹⁶ ad un generico istante t^* , partendo dalla condizione iniziale $\hat{x}_0(t^*)$ con uscita $\hat{p}_0(t^*)$ ¹⁷, di poter far evolvere il proprio stato in maniera tale da riprodurre il trascorrere dell'intervallo Δt ed ottenere così all'uscita $\hat{p}_{\Delta t}(t^*)$ la stima di p al tempo $t + \Delta t$, di modo che

$$\hat{p}_{\Delta t}(t^*) \equiv p(t^* + \Delta t) \quad (6.19)$$

A questo punto, seguendo lo schema di figura 6.15, è possibile ottenere la variazione Δe da apportare all'errore $e(t^*)$

$$\Delta e = \hat{p}_0(t^*) - \hat{p}_{\Delta t}(t^*) \quad (6.20)$$

Un sistema dinamico con queste caratteristiche viene chiamato *predittore* e, se utilizzato secondo lo schema di figura 6.15, ha la peculiarità di annullare la 6.18 attraverso la 6.20 dove al posto di $e(t^*)$ si pone $e(t^*) - \Delta e$, infatti risulta

$$e_p = e(t^* + \Delta t) - (e(t^*) - \Delta e) = p(t^* + \Delta t) - (p(t^*) - (\hat{p}_0(t^*) - \hat{p}_{\Delta t}(t^*))) \quad (6.21)$$

termine di valore nullo se le stime del predittore sono attendibili.

Uno spunto, per ottenere un tale sistema dinamico, arriva dalle simulazioni condotte nell'ambiente *Simulink*, dove si è potuto verificare attraverso una analisi comparativa tra simulazioni e prove reali, la validità del modello realizzato¹⁸.

¹⁶ Utilizzata ogni qualvolta si invia una nuova correzione al Puma 260.

¹⁷ Cioè dallo stato corrente del *modello del sistema reale*.

¹⁸ Vedere le figure 6.6, 6.7, 6.8.

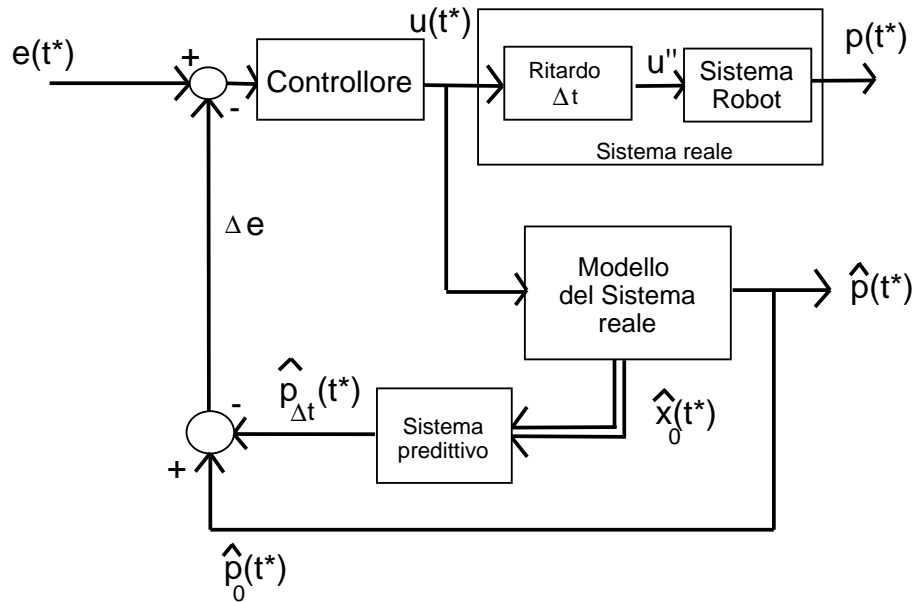


Figura 6.15: Schema di utilizzo del *predittore* .

Infatti, pur trascurando nel modello creato l'errore introdotto dal ritardo variabile della seriale sulla traiettoria del robot, le discrepanze tra le simulazioni ed il comportamento del sistema reale sono minime. Questo risultato è molto importante perché permette di poter implementare prima il modello di figura 6.13 e poi lo schema del “*predittore*” di figura 6.15.

6.5.1 Spazio degli stati

Per mezzo della funzione di trasferimento 6.14 si è data una rappresentazione matematica di tipo ingresso uscita del sistema robot Puma 260, ora invece si esprimerà tale sistema attraverso un modello a variabili di stato comprensivo anche del ritardo della seriale.

Un generico modello lineare a coefficienti costanti¹⁹ ed a tempo discreto può essere descritto nella seguente forma

$$\begin{cases} \vec{x}_{k+1} = \mathbf{A} \cdot \vec{x}_k + \mathbf{B} \cdot \vec{u}_k \\ \vec{p}_k = \mathbf{C} \cdot \vec{x}_k + \mathbf{D} \cdot \vec{u}_k \end{cases} \quad (6.22)$$

¹⁹Se non si utilizza un algoritmo di controllo del raccordo di velocità descritto nel paragrafo 6.4, il nostro sistema può essere considerato lineare ed a coefficienti costanti.

Nel caso specifico di questa tesi, i modelli rappresentati in figura 6.2 e 6.3 sono stati implementati con le seguenti matrici

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{K} & 1 \end{pmatrix} \quad \mathbf{B} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\mathbf{C} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad \mathbf{D} = \begin{pmatrix} 0 \end{pmatrix}$$

La dimensione della matrice \mathbf{A} è strettamente legata al ritardo complessivo del sistema di figura 6.3 che è di circa $320ms$ ed al sample time $T_s = 32ms$ del Puma in Alter Mode, notare che questo sistema riproduce la funzione del modello di figura 6.13 precedentemente descritto.

Attraverso il modello 6.22 è quindi possibile, dati uno stato iniziale \vec{x}_k ed il valore dell'ingresso u_k , conoscere l'uscita p agli istanti $(k, k + 1, \dots, k + m)$, dove m è un intero strettamente legato al ritardo del sistema, concetto che verrà approfondito nel prossimo paragrafo.

6.5.2 Modello del predittore

Analizzando le matrici che caratterizzano il sistema studiato si può notare che $\mathbf{D} = 0$ cioè non esiste un legame istantaneo tra ingresso u_k ed uscita \vec{p}_k e questo indica allora che l'uscita del sistema è condizionata solo dai valori passati dell'ingresso u , ma a sua volta tale segnale non varia ad ogni iterazione k perché per i tempi di latenza della seriale è possibile aggiornare il segnale u_k solo (circa) ogni $260ms$, in tale tempo il modello dinamico del Puma indicato dalla 6.22 ha eseguito $m_c = \frac{260ms}{32ms} \approx 8$ iterazioni per cui si avranno sequenze di ingressi formate da pacchetti di otto valori uguali.

Per maggior chiarezza si presenta un esempio “tipo” dei valori dello stato \vec{x}_k e dell’uscita p_k del sistema al giungere dalla seriale di un nuovo ingresso u_k

$$\vec{x}_k = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \end{pmatrix}_k = \begin{pmatrix} u_{k-1} \\ u_{k-2} \\ u_{k-3} \\ u_{k-4} \\ u_{k-5} \\ u_{k-6} \\ u_{k-7} \\ u_{k-8} \\ u_{k-9} \\ x_{10_{k-1}} + \frac{u_{k-10}}{K} \end{pmatrix} \equiv \begin{pmatrix} u_{k-1} \\ u_{k-1} \\ u_{k-1} \\ u_{k-1} \\ u_{k-1} \\ u_{k-1} \\ u_{k-1} \\ u_{k-1} \\ u_{k-1} \\ x_{10_{k-1}} + \frac{u_{k-9}}{K} \end{pmatrix} \quad (6.23)$$

mentre per l’uscita risulta semplicemente il termine scalare

$$p_k = x_{10_k} \quad (6.24)$$

È quindi possibile ottenere in maniera deterministica l’uscita del sistema ad un generico istante $k + m$ con $m \leq 10$ addirittura senza conoscere l’ingresso presente u_k ²⁰. Partendo da uno stato iniziale del tipo indicato in 6.23 possiamo scrivere

$$\vec{x}_{k+m} = \mathbf{A}^m \cdot \vec{x}_k \quad (6.25)$$

Mentre considerando l’ingresso presente u_k si possono predire le uscite fino ad un massimo 18 iterazioni future per un ammontare di $18 \cdot T_s = 576ms$ ²¹. Partendo come prima dallo stato iniziale indicato dalla 6.23 possiamo ottenere

$$\begin{aligned} \vec{x}_{k+m} = & \mathbf{A}^m \cdot \vec{x}_k + \mathbf{A}^{m-1} \cdot \mathbf{B} \cdot u_k + \mathbf{A}^{m-2} \cdot \mathbf{B} \cdot u_{k+1} + \dots \\ & + \mathbf{A} \cdot \mathbf{B} \cdot u_{k+m-1} + \mathbf{B} \cdot u_{k+m} \end{aligned} \quad (6.26)$$

Essendo il ritardo complessivo del sistema di circa 320ms ecco allora risultare $m = \frac{320ms}{32ms} = 10$, per tale valore è possibile utilizzare la semplice 6.25 al posto

²⁰Infatti l’ingresso u_k impiega $m = 10$ iterazioni per iniziare a condizionare il valore dell’uscita p .

²¹Questo valore deriva dalle transizioni effettuate dal sistema durante il tempo di controllo ($m_c = 8$) più quelle dovute al ritardo che sono appunto 10.

della piú generica ma complessa 6.26 ed ottenere quindi *una predizione della posizione p del Robot Puma 260 nel momento in cui questo inizierà ad eseguire il comando u_k appena inviatogli.*

La matrice \mathbf{A}_p del *predittore* con $m=10$ istanti di predizione diviene allora

$$\mathbf{A}_p = \mathbf{A}^{10} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{K} & \frac{1}{K} & \frac{1}{K} & \frac{1}{K} & \frac{1}{K} & \frac{1}{K} & \frac{1}{K} & \frac{1}{K} & \frac{1}{K} & 1 \end{pmatrix} \quad (6.27)$$

ed attraverso la 6.25 si ottiene lo stato

$$x_{10_{k+10}} = x_{10_k} + \frac{1}{K}(m_c \cdot u_{k-1} + ((m - m_c) \cdot u_{k-9})) \quad (6.28)$$

e conseguentemente si ricava l'uscita p_{k+10} utilizzando la 6.24, con queste si è riusciti a dare una rappresentazione matematica del *sistema predittivo* schematizzato in figura 6.14, dove $p_{\Delta t}$ coincide con p_{k+10} .

Il risultato ottenuto non è ancora direttamente utilizzabile come predizione, infatti il Puma "ragiona" in modalità cumulativa cioè interpreta i dati ricevuti come scostamenti dalla posizione attuale e non come posizioni assolute, per cui bisogna ricavare la variazione dell'uscita

$$\Delta e = p_k - p_{k+10} \quad (6.29)$$

Il valore di Δe è la correzione da apportare all'errore $e(t^*)$ rilevato tramite la telecamera come raffigurato in figura 6.15, dove \hat{p}_0 coincide con p_k .

Entrambi i sistemi appena descritti possono essere riuniti sotto un'unica

rappresentazione del tipo 6.22 con le seguenti matrici

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{K} & 1 & 0 \end{pmatrix} \quad \mathbf{B} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\mathbf{C} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \frac{1}{K} & \frac{1}{K} & \frac{1}{K} & \frac{1}{K} & \frac{1}{K} & \frac{1}{K} & \frac{1}{K} & \frac{1}{K} & \frac{1}{K} & \frac{1}{K} & 1 \end{pmatrix} \quad \mathbf{D} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Dove, sempre in riferimento alla 6.22, il primo elemento dell'uscita \vec{p}_k rappresenta \hat{p}_0 , mentre il secondo $p_{\Delta t}$.

6.5.3 Predittore di Smith

Lo schema realizzativo appena sviluppato può essere ricondotto attraverso piccoli cambiamenti alla struttura del *Predittore di Smith*²². Infatti riconducendo la correzione del segnale errore Δe all'uscita $y(t)$ di figura 6.12 ed adattando opportunamente i segni dei blocchi sommatori, è possibile dare una schematizzazione più intuitiva dell'intero sistema come mostrato in figura 6.16.

6.5.4 Prestazioni del predittore

Si analizzeranno ora i risultati ottenuti implementando lo schema di controllo rappresentato in figura 6.15 utilizzando un semplice regolatore proporzionale con guadagno $K_p = 1$.

La prestazione, come per i regolatori precedentemente studiati, è calcolata

²²Vedere [26, 16, 31].

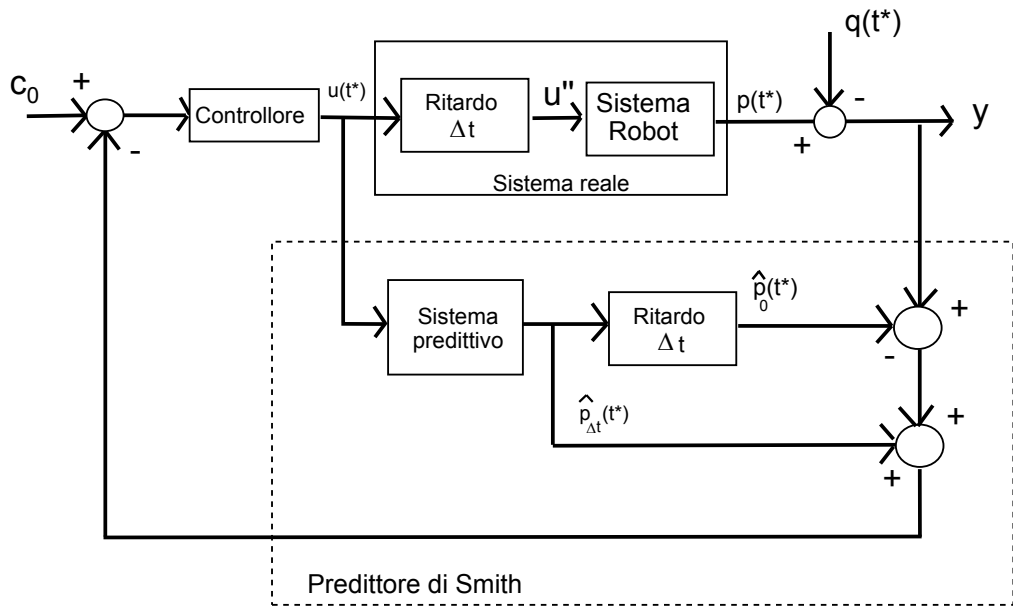


Figura 6.16: Schema funzionale del *predittore di Smith*.

attraverso il tempo di assestamento T_a dell'errore in risposta ad un gradino di circa 45mm (tale andamento è graficato in figura 6.17)

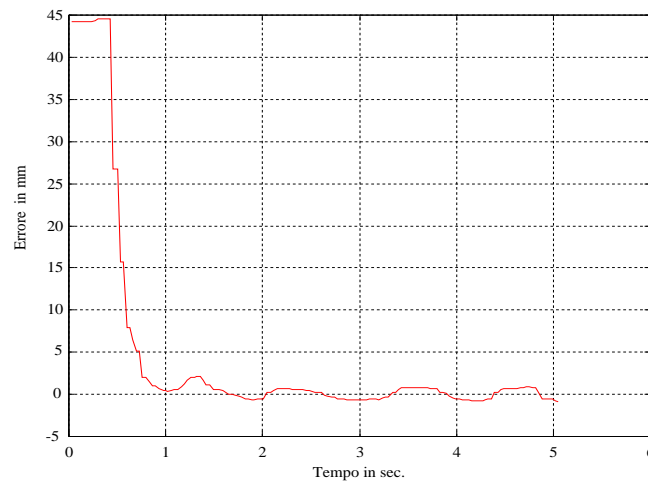


Figura 6.17: Prestazione del controllo con il solo predittore.

Appare subito evidente come l'utilizzo di un regolatore proporzionale insieme al predittore generi una risposta di gran lunga migliore alla corrispondente risposta del caso studiato nel paragrafo 6.3.2. Notare come tale andamento assomigli

alla risposta del controllo PD, questo non deve sorprendere dato che a suo modo il derivatore tende ad anticipare il valore futuro dell'errore, ma mentre quest'ultimo per dare la stima futura utilizza l'ultimo e penultimo valore dell'errore (vedere l'equazione 6.13), il predittore utilizza invece, secondo la formulazione data nella 6.28, il penultimo ed il terzultimo valore inviato al robot.

Questa differenza permette di passare da $T_a \approx 0.8sec$ del regolatore PD al $T_a \approx 0.5sec$ con il predittore, nel raggiungimento di un oggetto fermo.

6.5.5 Limiti di utilizzo del predittore

Analizzando invece il comportamento del sistema di figura 6.16 caratterizzato da un controllore PID (con parametri $T = 0.130s$, $K_p = 0.5$, $T_i = 1s$, $T_d = 0.1s$, $N = 10$) con un target che si muove a velocità costante, in base a quanto detto in 6.3.4, appare un risultato inatteso, il sistema, pur avendo come controllore un PID, compie un errore di velocità non nullo (come mostrato nella prova realizzata in figura 6.18)

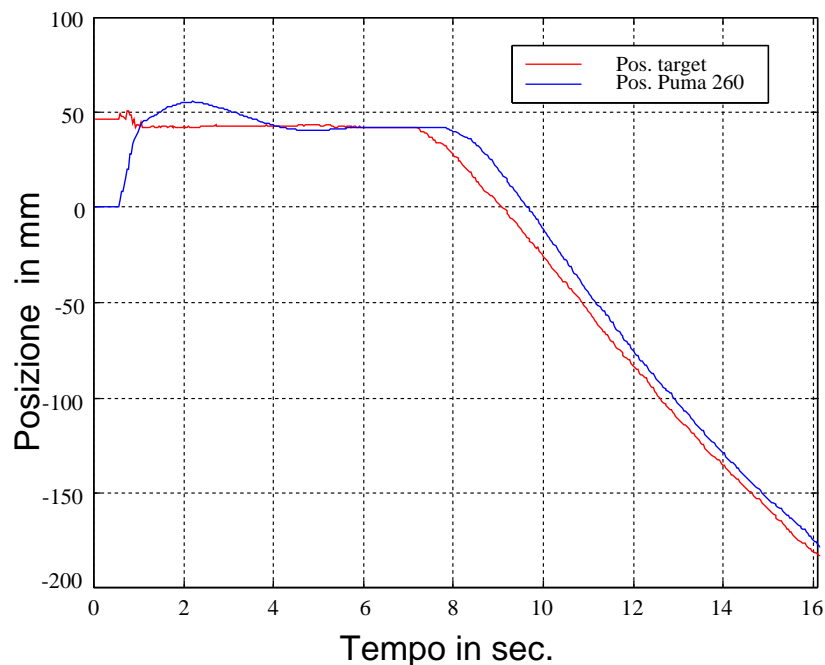


Figura 6.18: Prova con l'utilizzo di un predittore e regolatore PID.

Per capire come mai l'inserimento del predittore vanifichi l'utilizzo di un re-

golatore PID si faccia riferimento allo schema del predittore di Smith riproposto in figura 6.19 (ponendo attenzione all'inversione dei segni dei nodi sommatori).

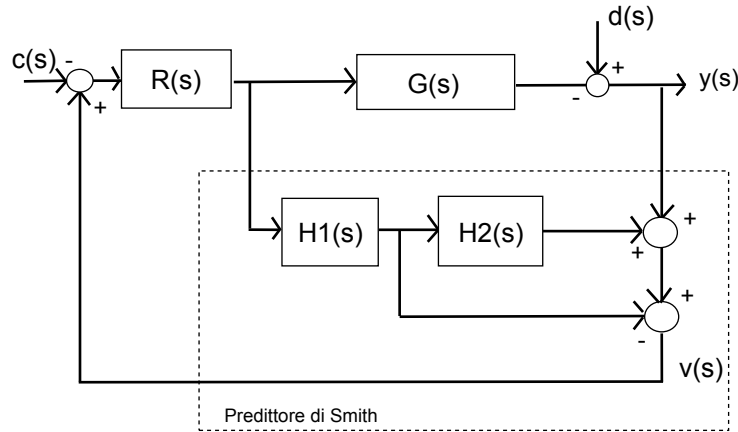


Figura 6.19: Schema in variabile s del predittore di Smith.

Tutte le grandezze saranno rappresentate attraverso la trasformata di Laplace per facilitare i calcoli, per cui scrivendo la f.d.t. dell'intero sistema formato da controllore, seriale e Robot Puma 260, in un unico termine si ottiene

$$G(s) = \frac{e^{-T_0 s}}{T_{ir} \cdot s} \quad (6.30)$$

entro il quale è stata modellata l'azione integrale del Robot $(\frac{1}{T_{ir} \cdot s})^{23}$ ed il ritardo della seriale T_0 (termine $e^{-T_0 s}$). È possibile ora ottenere immediatamente la H_1 e H_2 , che saranno rispettivamente

$$H_1(s) = \frac{1}{T_{ir} \cdot s} \quad H_2 = e^{-T_0 s} \quad (6.31)$$

di modo che sia

$$G = H_1 \cdot H_2 \quad (6.32)$$

Per ultimo definendo la f.d.t. del regolatore come un semplice integratore²⁴

$$R_1(s) = \frac{1}{s} \quad (6.33)$$

²³Il parametro T_{ir} rappresenta la costante di integrazione del robot.

²⁴Comprendere la parte proporzionale e derivativa complicherebbe i calcoli e non cambierebbe la sostanza del discorso.

si può calcolare la f.d.t dell'intero sistema di controllo avendo come ingressi i segnali $d(s)$ e $c(s)$ e come uscita il segnale $y(s)$ ²⁵, ponendo allora

$$\begin{cases} y = d - G \cdot u \\ v = y + u \cdot H_1 \cdot H_2 - u \cdot H_1 \\ u = (v - c) \cdot R \end{cases} \quad (6.34)$$

dopo pochi passaggi si ricava che

$$y = \frac{1 + R \cdot H_1 - H_1 \cdot H_2 \cdot R}{1 + R \cdot H_1 + G \cdot R - H_1 \cdot H_2 \cdot R} \cdot d + \frac{R \cdot G}{1 + R \cdot H_1 + G \cdot R - H_1 \cdot H_2 \cdot R} \cdot c$$

utilizzando la 6.32, risulta

$$y = \frac{1 + R \cdot H_1 - H_1 \cdot H_2 \cdot R}{1 + R \cdot H_1} \cdot d + \frac{R \cdot G}{1 + R \cdot H_1} \cdot c \quad (6.35)$$

Rappresentando il segnale c un riferimento costante (posizione sullo schermo rispetto alla quale si calcola l'errore), possiamo annullarlo senza pregiudicare i risultati che si otterranno. A questo punto si è ottenuta una $G_t(s)$ che descrive il comportamento dell'uscita $y(s)$ in funzione del segnale di ingresso $d(s)$.

$$G_t(s) = \frac{y(s)}{d(s)} = \frac{1 + R \cdot H_1 - H_1 \cdot H_2 \cdot R}{1 + R \cdot H_1} \quad (6.36)$$

Calcolando l'errore a regime della $y(s)$, si ottiene

$$\lim_{s \rightarrow 0} s \cdot y(S) = \lim_{s \rightarrow 0} s \cdot G_t(S) \cdot d(s) \quad (6.37)$$

sostituendo i valori della 6.31 e della 6.33, risulta

$$\lim_{s \rightarrow 0} s \cdot \frac{T_{ir} \cdot s^2 + 1 - H_2}{T_{ir} \cdot s^2 + 1} \cdot d(s) \quad (6.38)$$

Si sviluppi ora in serie di Maclaurin il termine $H_2(s)$

$$H_2 = e^{-T_0 \cdot s} = 1 - T_0 \cdot s + \frac{(T_0 \cdot s)^2}{2} + \dots \quad (6.39)$$

troncando al 2° termine e sostituendo nella 6.38 si ottiene

$$\lim_{s \rightarrow 0} s \cdot \frac{T_{ir} \cdot s^2 + T_0 \cdot s - \frac{(T_0 \cdot s)^2}{2}}{T_{ir} \cdot s^2 + 1} \cdot d(s) \quad (6.40)$$

²⁵Per una notazione più compatta si trascurerà la scrittura della variabile s .

La 6.40 permette di calcolare l'errore a regime commesso dal sistema in funzione di una particolare traiettoria $d(s)$ del target.

Nel caso in cui il target è fermo, $d(s)$ risulta

$$d(s) = \frac{d_0}{s} \quad (6.41)$$

sostituendo nella 6.40 e calcolando il limite si ricava un errore a regime nullo, risultato questo già anticipato nel paragrafo 6.5.4.

Analizzando invece il caso in cui l'oggetto si muove a velocità costate, in tale circostanza $d(s)$ ha la seguente forma

$$d(s) = \frac{d_0}{s^2} \quad (6.42)$$

calcolando il limite 6.40, si ottiene

$$\lim_{s \rightarrow 0} s \cdot \frac{T_{ir} \cdot s^2 + T_0 \cdot s - \frac{(T_0 \cdot s)^2}{2}}{T_{ir} \cdot s^2 + 1} \cdot \frac{d_0}{s^2} = T_0 \cdot d_0 \quad (6.43)$$

conferma matematica del comportamento indesiderato dello schema di controllo con il predittore.

Proponendo in figura 6.20 la corrispondente simulazione (realizzata con un modello Simulink) della prova reale precedentemente mostrata in figura 6.18, appare subito evidente l'errore che viene commesso nell'inseguimento di un oggetto che si muove a velocità costante.

Essendo il ritardo del sistema il solito $T_0 = 0.32sec$ e muovendosi il target con una velocità di circa $d_0 = \dot{q} = 26mm/sec$ si ottiene dalla 6.43 che l'errore commesso è

$$T_0 \cdot d_0 = 8.32mm \quad (6.44)$$

che è effettivamente molto prossimo al valore raffigurato nel grafico dell'errore in figura 6.20.

6.5.6 Errore del PD con un target in movimento

Ora si presenterà, per un eventuale confronto, l'errore commesso da uno schema di controllo, già presentato nei paragrafi 6.3.1 e 6.3.3, che utilizzi un regolatore PD.

Il regolatore avrà allora la seguente f.d.t.

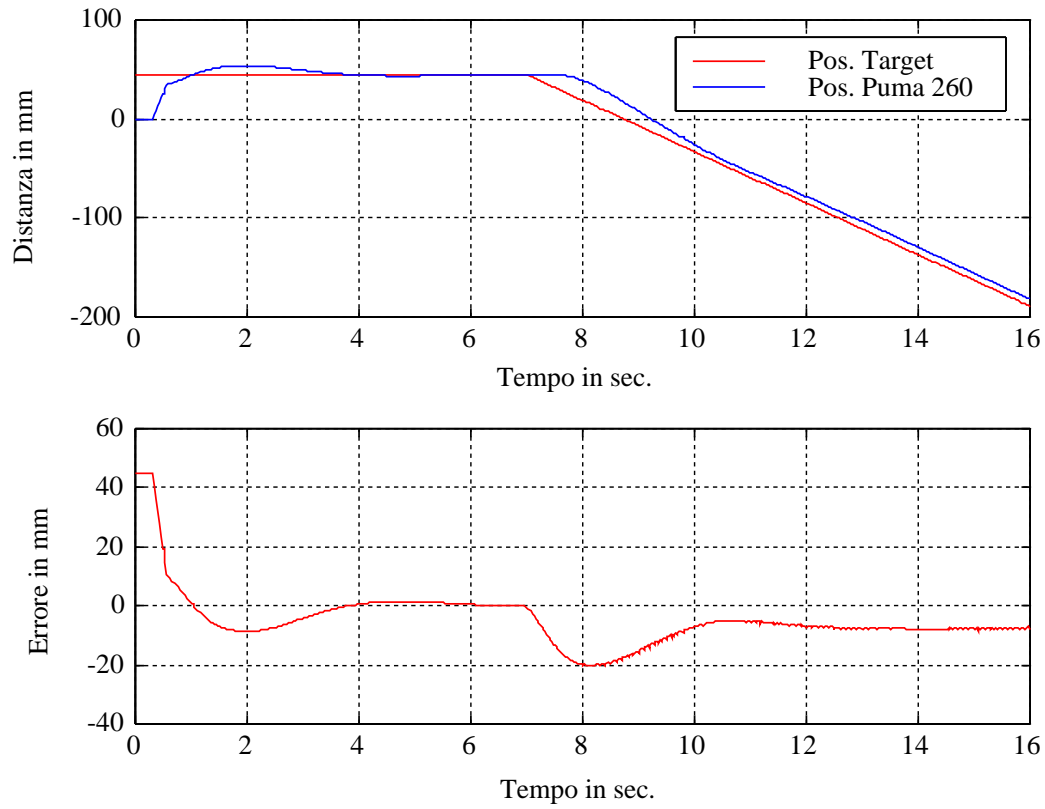


Figura 6.20: Simulazione schema di controllo predittore con PID.

$$R(s) = K_p \cdot \left(1 + \frac{\beta \cdot (z - 1)}{z - \gamma}\right) \quad (6.45)$$

mentre il sistema da controllare (vedere il paragrafo 6.3.4)

$$G(s) = \frac{z^{-10}}{z - 1} \cdot \frac{1}{K} \quad (6.46)$$

Sotto queste ipotesi calcolando l'errore di velocità e_v di $y(z)$

$$e_v = \lim_{z \rightarrow 1} (z - 1) \cdot y(z)$$

si ottiene dopo alcuni passaggi

$$e_v = \lim_{z \rightarrow 1} \frac{(z - 1)^2 \cdot (z - \gamma)}{(z - 1) \cdot (z - \gamma) + z^{-10} \cdot \frac{K_p}{K} \cdot (z - \gamma + \beta \cdot (z - 1))} \cdot d(z) \quad (6.47)$$

Il sample time della telecamera è $T_{frame} = 33.ms^{26}$ per cui un oggetto che si muove a velocità costante potrà essere così rappresentato

$$d(z) = \frac{d_0 \cdot T_{frame}}{(z - 1)^2} \quad (6.48)$$

sostituendo la 6.48 nella 6.47 e calcolando il limite si ottiene l'errore a regime commesso utilizzando un regolatore PD

$$e_v = \frac{d_0 \cdot T_{frame} \cdot K}{K_p} \quad (6.49)$$

risultato del tutto analogo alla 6.43 con l'unica differenza che la 6.49 ha diversi parametri definibili dal progettista mentre T_0 è una grandezza non manipolabile perché dipendente dal vincolo rappresentato dal ritardo di trasmissione della seriale.

²⁶ Vedere il paragrafo 6.1.