Universitá Degli Studi di Napoli
Federico II

Corso di Laurea in Ingegneria dell'Automazione

Corso di Sistemi di Controllo Multivariabile

Sistema di controllo per l'equilibrio di una sfera isotropa rotolante su un piano ad inclinazione variabile

Antonia Fiore M58/66

Anno Accademico 2012-2013

Indice

Introduzione **4**

1 Modello Matematico **6**
1.1 Linearizzazione del Modello 10
1.2 Modello del Sistema Linearizzato nello Spazio di Stato 11

2 Circuiteria di Controllo **12**

3 Modello Simulink **16**
3.1 Image Acquisition Toolbox 17
3.2 Controllo LQR 19
3.3 Comunicazione Via Seriale 23

4 Osservatore di Luenberger **26**

A Appendice **29**

Elenco delle figure

1 Vista dall'alto 4
2 Giunto Rotoidale 5
3 Attuatori 5

1.1 Vista laterale del sistema palla-piano 7

2.1 Launchpad-Texas Instruments 13
2.2 Servocomando Power HD-1900A 14
2.3 Componenti interni di un servocomando 15
2.4 PWM da inviare al servocomando 15

3.1 Andamento della posizione lungo l'asse x 21
3.2 Andamento della posizione lungo l'asse y 21
3.3 Andamento della velocitá stimata lungo l'asse x 22
3.4 Andamento della velocitá stimata lungo l'asse y 22
3.5 Schema Simulink 25

4.1 Schema Simulink - Osservatore 27

Introduzione

Obiettivo del presente progetto é la realizzazione di un tipico sistema multi-variabile con lo scopo di controllarne la stabilitá e l'inseguimento di un set prestabilito di traiettorie.

Figura 1: Vista dall'alto

Ai fini del controllo, si osservi che la pallina su un piano rigido é un oggetto non vincolato che si muove liberamente senza alcuna capacitá di riconoscere l'ambiente circostante: la pallina,quindi, non é capace di controllare da sola il suo movimento e questo rende il controllo del sistema molto piú complicato. Si tenga bene a mente che, a causa dell esigue risorse a disposizione, la realizzazione finale del progetto risulta molto rudimentale; per tale motivo, i risultati del controllo, seppur soddisfacenti, non sono del tutto ottimali.

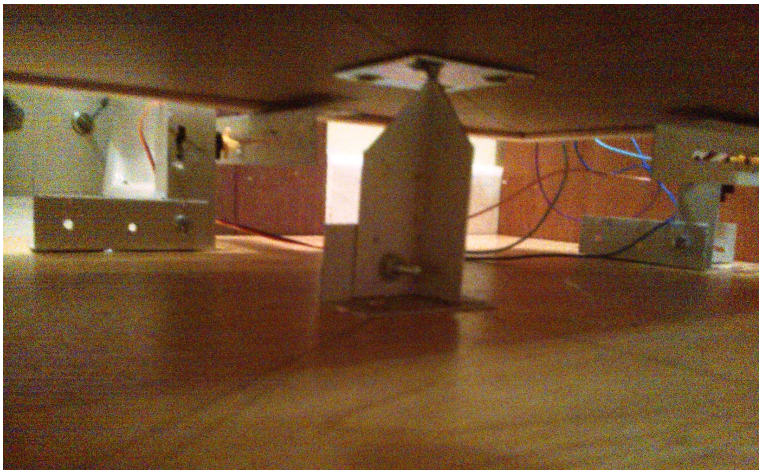


Figura 2: Giunto Rotoidale

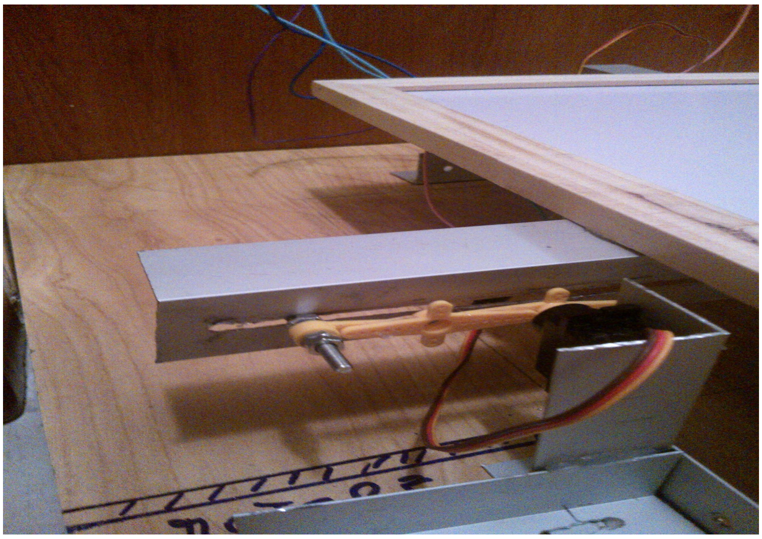


Figura 3: Attuatori

Di seguito viene fornita una spiegazione dettagliata su come é stato ricavato il modello del sistema palla-piano e di come é stata implementata la circuiteria di controllo.

Capitolo 1

Modello Matematico

Per semplificare il modello del sistema sono state assunte le seguenti ipotesi:

- Nessun slittamento della pallina.
- Pallina completamente simmetrica ed omogenea.
- Pallina e piano sempre in contatto.

Le equazioni dinamiche del sistema palla-piano sono state ricavate applicando l'equazione Lagrangiana al modello:

$$\frac{d}{dt}\frac{\partial T}{\partial \dot{q}_i} - \frac{\partial T}{\partial q_i} + \frac{\partial V}{\partial q_i} = Q_i \tag{1.1}$$

dove q_i indica la coordinata della direzione i-esima, T é l'energia cinetica del sistema, V l'energia potenziale e Q é la risultante delle forze agenti sul sistema.
Di seguito viene fornita una schematizzazione del sistema.

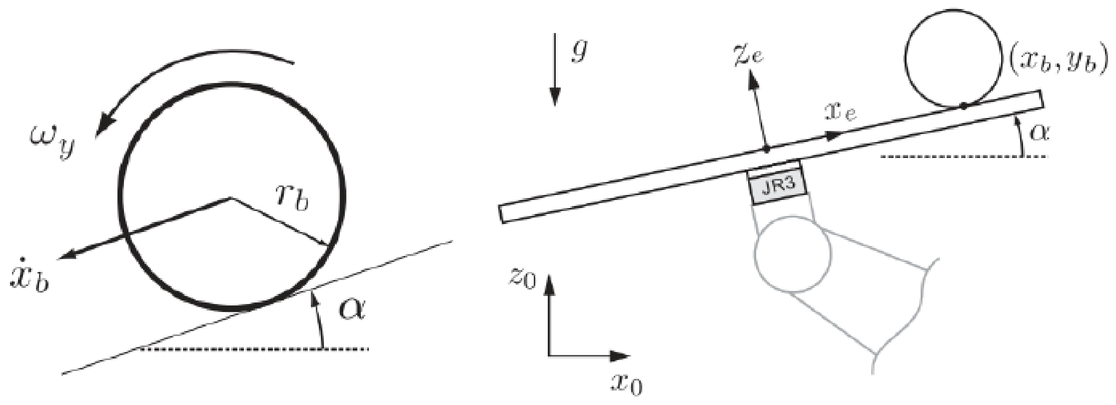


Figura 1.1: Vista laterale del sistema palla-piano

Come si evince dalla figura, il sistema possiede 4 gradi di libertá: due legati al movimento della palla e due legati all'inclinazione del piano; le coordinate generalizzate del sistema sono x_b e y_b per la posizione della palla e α e β per l'inclinazione del piano.
É importante osservare che l'origine del sistema di cooridnate $x - y$ coincide con il centro del piano.

Analizzeremo di seguito i contributi energetici dovuti sia alla rotazione della singola pallina che alla rotazione legata al movimento del piano solidale ad essa.

1. L'energia cinetica della singola palla T_{b1} é data dalla somma di due contributi: uno legato alla traslazione, l'altro alla rotazione intorno al centro di massa:

$$T_{b1} = \frac{1}{2} m_b \left(\dot{x}_b^2 + \dot{y}_b^2 \right) + \frac{1}{2} I_b \left(w_x^2 + w_y^2 \right) \tag{1.2}$$

   in cui : m_b é la massa della pallina e I_b il suo momento di massa d'inerzia; $\dot{x}_b$ e $\dot{y}_b$ sono le velocitá di traslazione della stessa lungo l'asse x e l'asse y.
   Le velocitá di traslazione e di rotazione sono legate dalle seguenti relazioni:

$$\dot{x}_b = r_b w_y \quad , \quad \dot{y}_b = r_b w_x \tag{1.3}$$

in cui r_b il raggio della pallina.
Sostituendo l'equazione (1.3) nell' equazione (1.2) si ottiene:

$$T_{b1} = \frac{1}{2}\left[m_b\left(\dot{x}_b^2 + \dot{y}_b^2\right) + \frac{I_b}{r_b^2}\left(\dot{x}_b^2 + \dot{y}_b^2\right)\right] = \frac{1}{2}\left(m_b + \frac{I_b}{r_b^2}\right)\left(\dot{x}_b^2 + \dot{y}_b^2\right) \tag{1.4}$$

2. L'energica cinetica della pallina solidale al piano ruotante, se si considera la pallina una massa puntiforme di coordinate (x_b,y_b), é data da:

$$\begin{aligned} T_{b2} &= \frac{1}{2}I_b\left(\dot{\alpha}^2 + \dot{\beta}^2\right) + \frac{1}{2}m_b\left(x_b\dot{\alpha} + \dot{\beta}y_b\right)^2 \\ &= \frac{1}{2}I_b\left(\dot{\alpha}^2 + \dot{\beta}^2\right) + \frac{1}{2}m_b\left(x_b^2\dot{\alpha}^2 + 2x_b\dot{\alpha}y_b\dot{\beta} + y_b^2\dot{\beta}^2\right) \end{aligned} \tag{1.5}$$

   in cui α e β sono le inclinazioni del piano rispettivamente lungo l'asse x e l' asse y e di conseguenza $\dot{\alpha}$ e $\dot{\beta}$ sono le velocitá di rotazione del piano.

In definitiva:

$$\begin{aligned} T_b &= T_{b1} + T_{b2} \\ &= \frac{1}{2}\left(m_b + \frac{I_b}{r_b^2}\right)\left(\dot{x}_b^2 + \dot{y}_b^2\right) + \frac{1}{2}I_b\left(\dot{\alpha}^2 + \dot{\beta}^2\right) + \frac{1}{2}m_b\left(x_b^2\dot{\alpha}^2 + 2x_b\dot{\alpha}y_b\dot{\beta} + y_b^2\dot{\beta}^2\right) \end{aligned} \tag{1.6}$$

Per quanto riguarda l'energia potenziale della pallina V_b relativa al piano orizzontale e alle sue possibile inclinazioni, si ha che:

$$V_b = m_b gh = m_b g\left(x_b sin\alpha + y_b sin\beta\right) \tag{1.7}$$

per cui l'equazione Lagrangiana che descrive il sistema palla-piano é :

$$L = T_b - V_b \tag{1.8}$$

Derivando la Lagrangiana rispetto alle componenti della posizione e della velocitá della pallina lungo l'asse x e y, si ottengono le equazioni costitutive cercate:

$$\frac{\partial T}{\partial \dot{x}_b} = \left(m_b + \frac{I_b}{r_b^2}\right)\dot{x}_b \quad , \quad \frac{\partial L}{\partial x_b} = m_b\left(x_b\dot{\alpha} + y_b\dot{\beta}\right)\dot{\alpha} \tag{1.9}$$

$$\frac{\partial T}{\partial \dot{y}_b} = \left(m_b + \frac{I_b}{r_b^2}\right)\dot{y}_b \quad , \quad \frac{\partial L}{\partial y_b} = m_b\left(x_b\dot{\alpha} + y_b\dot{\beta}\right)\dot{\beta} \tag{1.10}$$

Essendo il gradiente della Lagrangiana pari alla sommatoria dell forze esterne, si ha che:

$$\frac{d}{dt}\frac{\partial T}{\partial \dot{x}_b} - \frac{\partial L}{\partial \dot{x}_b} = \left(m_b + \frac{I_b}{r_b^2}\right)\ddot{x}_b - m_b\left(x_b\dot{\alpha} + y_b\dot{\beta}\right)\dot{\alpha} + m_b g sin\alpha = 0 \quad (1.11)$$

$$\frac{d}{dt}\frac{\partial T}{\partial \dot{y}_b} - \frac{\partial L}{\partial \dot{y}_b} = \left(m_b + \frac{I_b}{r_b^2}\right)\ddot{y}_b - m_b\left(y_b\dot{\beta} + x_b\dot{\alpha}\right)\dot{\alpha} + m_b g sin\beta = 0 \quad (1.12)$$

e si ottengono proprio le equazioni differenziali non lineari che descrivono il sistema palla-piano:

$$\left(m_b + \frac{I_b}{r_b^2}\right)\ddot{x}_b - m_b\left(x_b\dot{\alpha}^2 + y_b\dot{\alpha}\dot{\beta}\right) + m_b g sin\alpha = 0 \quad (1.13)$$

$$\left(m_b + \frac{I_b}{r_b^2}\right)\ddot{y}_b - m_b\left(y_b\dot{\beta}^2\right) + m_b g sin\beta = 0 \quad (1.14)$$

Le equazioni (1.17)-(1.18)mostrano il legame tra la stato della pallina e l'inclinazione del piano. Ovviamente, manipolare direttamente queste equazioni risulta molto difficile, per questo motivo di seguito verrá derivato un modello linearizzato nell'intorno di un punto di lavoro.

1.1 Linearizzazione del Modello

Partendo dalle equazioni (1.13)-(1.14) e sostituendo l'espressione generica del momento di massa d'inerzia di una sfera solida $I_{balla} = \frac{2}{5} mbr_b^2$ si ottiene:

$$m_b \left[\frac{7}{5} \ddot{x}_b - \left(x_b \dot{\alpha}^2 + y_b \dot{\alpha} \dot{\beta} \right) + gsin\alpha \right] = 0 \tag{1.15}$$

$$m_b \left[\frac{7}{5} \ddot{y}_b - \left(y_b \dot{\beta}^2 + x_b \dot{\alpha} \dot{\beta} \right) + gsin\beta \right] = 0 \tag{1.16}$$

Queste due equazioni possono essere linearizzate sotto le ipotesi per il piano di:

- Piccoli angoli di inclinazione (fino a ±5°):

$$\alpha \ll 5 \;\; e \;\; \beta \ll 5 \Rightarrow \;\; sin\alpha \simeq \alpha, \;\; sin\beta \simeq \beta$$

- Basse velocitá d'inclinazione:

$$|\dot{\alpha}| \ll 1 \;\; e \;\; |\dot{\beta}| \ll 1 \Rightarrow \;\; \dot{\alpha}\dot{\beta} \simeq 0, \;\; \dot{\alpha}^2 \simeq 0, \;\; \dot{\beta}^2 \simeq 0$$

Ottendo in tal modo le equazioni del modello linearizzato intorno al punto d'equilibrio $(x_b, y_b) = (0, 0)$:

$$\frac{7}{5} \ddot{x}_b + g\alpha = 0 \tag{1.17}$$

$$\frac{7}{5} \ddot{y}_b + g\beta = 0 \tag{1.18}$$

Linearizzando le equazioni (1.15)(1.16) si ottengono due equazioni differenziali indipendenti per l'asse x e per l'asse y che possono facilmente essere utilizzate per ottenere lo stato del sistema.
Assumendo che $\alpha(s)$ e $\beta(s)$ siano gli ingressi del sistema, le funzioni di trasferimento lungo l'asse x e lungo l'asse y risultano essere:

$$P_x(s) = \frac{X_b(s)}{\alpha(s)} = -\frac{g}{\frac{7}{5}s^2} \tag{1.19}$$

$$P_y(s) = \frac{Y_b(s)}{\beta(s)} = -\frac{g}{\frac{7}{5}s^2} \tag{1.20}$$

1.2 Modello del Sistema Linearizzato nello Spazio di Stato

Considerando le equazioni (1.17)-(1.18) é possibile ricavare le equazioni in forma di stato del sistema linearizzato, ottenuto con le dovute approssimazioni.
A tal proposito, definiamo il vettore dell variabili di stato:

$$X = [x_1, x_2, x_3, x_4]^T = [x_b, \dot{x}_b, y_b, \dot{y}_b]^T$$

il vettore degli ingressi:

$$U = [u_x, u_y]^T = [\alpha, \beta]^T$$

ottenendo in questo modo:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} X + \begin{bmatrix} 0 & 0 \\ -\frac{5}{7}g & 0 \\ 0 & 0 \\ 0 & -\frac{5}{7}g \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix} \tag{1.21}$$

Essendo le velocitá e le accelerazioni del piano molto basse ($|\dot{\alpha}| \ll 1$ e $|\dot{\beta}| \ll 1$), possiamo dividere il sitema in due sotto-sistemi che possono essere controllati indipendentemente l'uno dall' altro.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{5}{7}g \end{bmatrix} \begin{bmatrix} u_x \end{bmatrix} \tag{1.22}$$

$$\begin{bmatrix} \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{5}{7}g \end{bmatrix} \begin{bmatrix} u_y \end{bmatrix} \tag{1.23}$$

Le uscite del sistema, ovvero la posizione della pallina lungo l'asse x e lungo l'asse y saranno date da:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} X = \begin{bmatrix} x_1 \\ x_3 \end{bmatrix} \tag{1.24}$$

Capitolo 2

Circuiteria di Controllo

Per la realizzazione dell'elettronica di controllo, é stata utilizzata la scheda di sviluppo MSP-EXP430G2 LaunchPad fornita dalla Texas Instrument, le cui principali caratteristiche sono:

- Emulazione on-board per programmazione e debug;
- Socket DIP di alloggiamento per svariati microcontrollori della famiglia MSP430G2xx;
- LED e pulsanti general purpose programmabili;
- Oscillatore a controllo digitale (DCO) precalibrato in fabbrica che gira di default ad 1MHz, ma é possibile regolarne la frequenza fino a 16 MHz.

Il microcontrollore utilizzato é l' **MSP430G2553** aventi le seguenti caratteristiche:

Parametri	**MSP430G2553**
Frequenza (MHz)	16
Memoria Flash (kB)	16
Memoria SRAM (kB)	0.5
General Purpose I/O	24
Timer a 16 bit	2
Watchdog	Sí
USCI-A (UART/LIN/IrDA/SPI)	1
USCI-B (I2C e SPI)	1
ADC	SAR a 10-bit
Canali ADC	8
Range di temperatura (°C)	$-40 - 85$

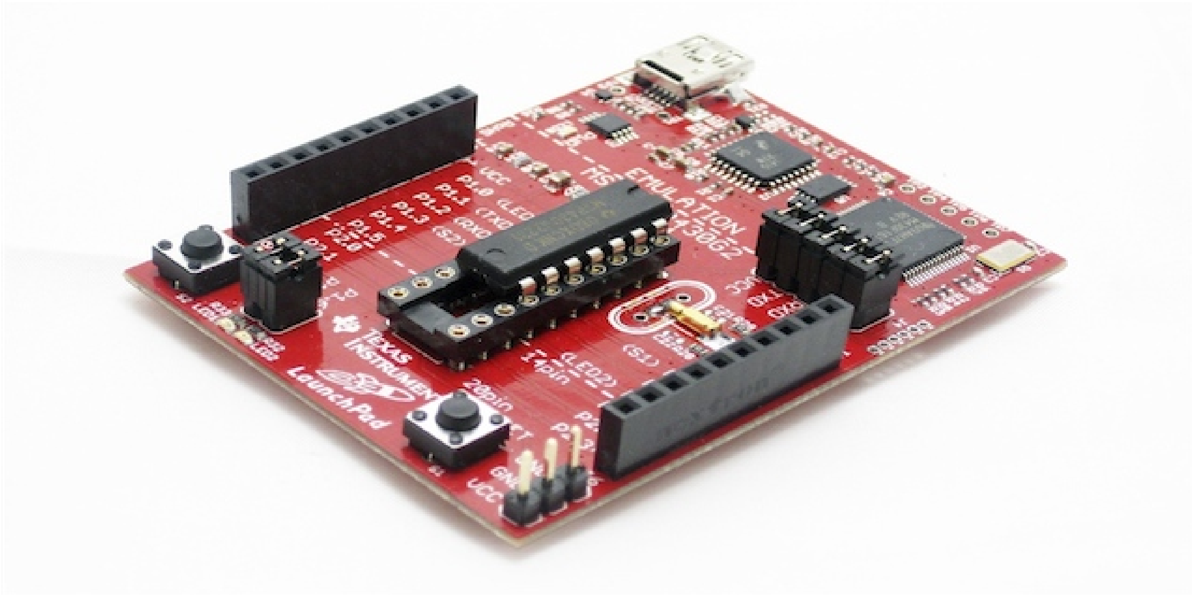


Figura 2.1: Launchpad-Texas Instruments

Come si evince dalla tabella, questo microcontrollore mette a disposizione il modulo UART per la comunicazione seriale con il PC ed il modulo TIMER per generare l'onda PWM necessaria per il controllo dei servocomandi. Infine, l'ambiente di sviluppo utilizzato per programmare la schedina é stato il *Code Composer Studio* fornito in licenza dalla stessa Texas Instrument.

Per quanto riguarda gli attuatori, la scelta é ricaduta sull'utilizzo di due servocomandi Power HD-1900A aventi le seguenti caratteristiche:

Caratteristiche	**HD-1900A**
Dimensioni	$23.0 \times 29.0 \times 12.2$ mm
Peso	9.0 g
Alimentazione	4.8 - 6.0 V
Momento Torcente	1.5 kg-cm
Velocitá	0.08 sec/60°

Il funzionamento di un servocomando é molto semplice: esso viene alloggiato all'interno di un contenitore in cui vengono posizionati anche un motore elettrico in corrente continua completo di riduzione meccanica, un potenziometro per la misura della posizione dell'albero ed un sistema di controllo in retroazione per la posizone dell'asse di uscita.

Figura 2.2: Servocomando Power HD-1900A

Il segnale di comando per questo tipo di attuatori é un'onda quadra inviata ripetutamente, il cui periodo é di $20ms$ (frequenza 50Hz) il cui fronte alto ha una durata compresa tra i 0.5 e i 2.5 ms.
Ció é dovuto al fatto che i servocomandi utilizzati hanno un'escursione di 180°e quindi ruotano di:

- 90°a destra con un impulso di 0.5 ms
- 90°a sinistra con un impulso di 2.5 ms

Per generare questo tipo di segnale é stato utilizzato il **Timer A** messo a disposizione dal microcontrollore MSP430G2553 utilizzato.

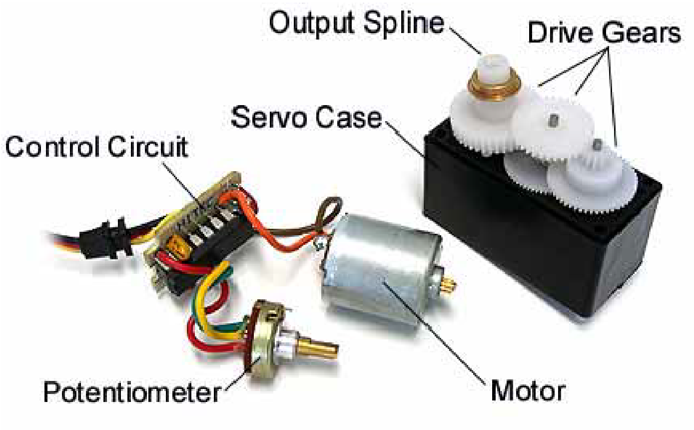


Figura 2.3: Componenti interni di un servocomando

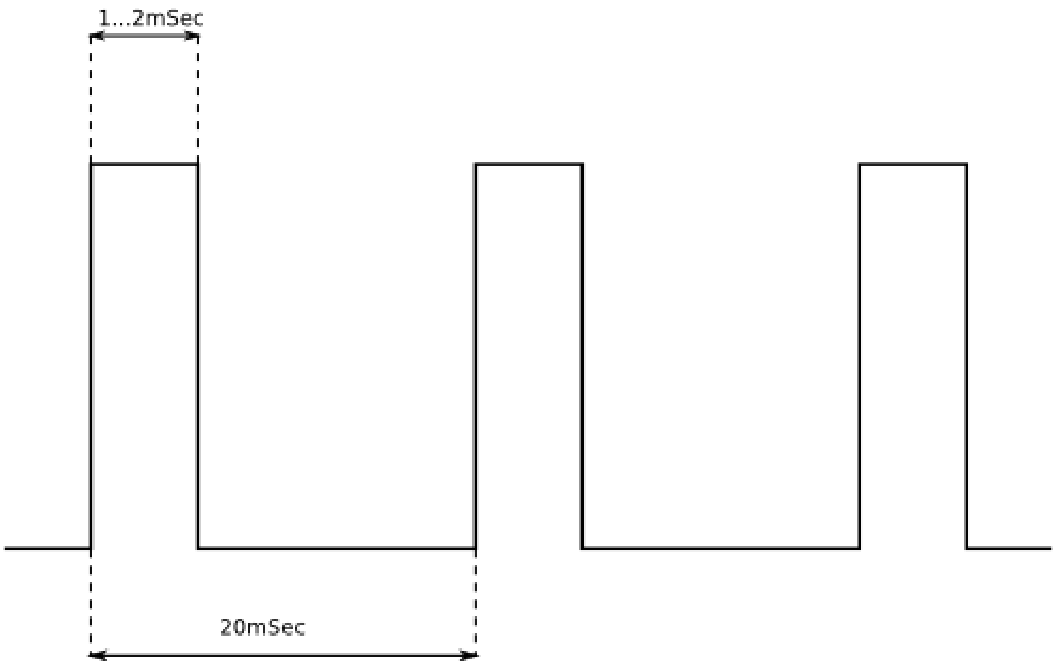


Figura 2.4: PWM da inviare al servocomando

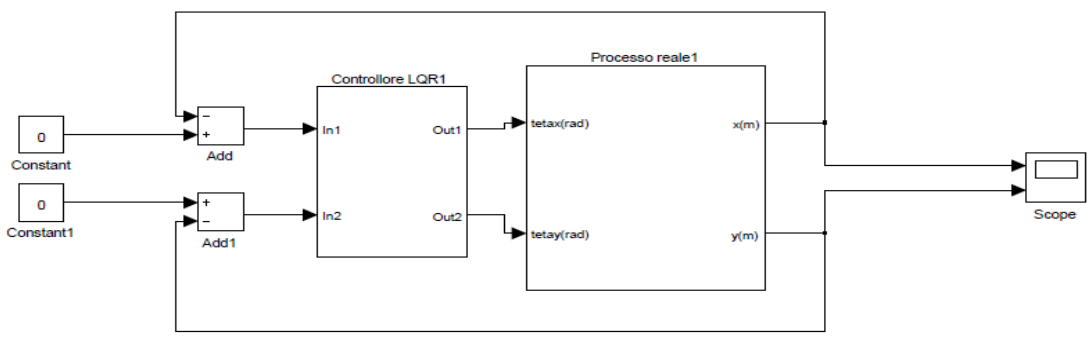
Capitolo 3

Modello Simulink

Il controllo del sistema in oggetto é stato implementato attraverso il tool *Simulink* fornito in licenza dalla Mathworks.
In particolare, esso consta di 3 macroblocchi:

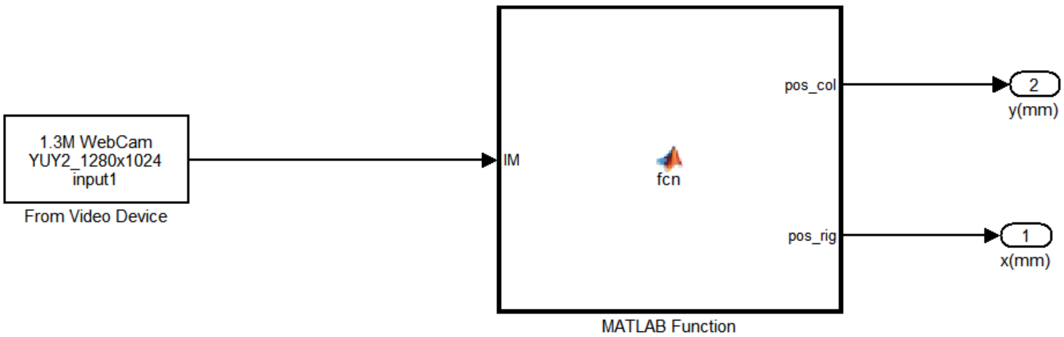
- il primo che esegue l'acquisizione dell'immagine, l'elabora e fornisce in uscita la posizione della pallina.
- il secondo che computa la legge di controllo in retroazione di stato e invia in uscita i valori degli angoli di posizione
- il terzo che comunica al microcontrollore la posizione da dare al motore via seriale, ovvero un valore in millisecondi che codifica la posizione dell'albero.

Di seguito analizzeremo nel dettaglio ciascun macroblocco.

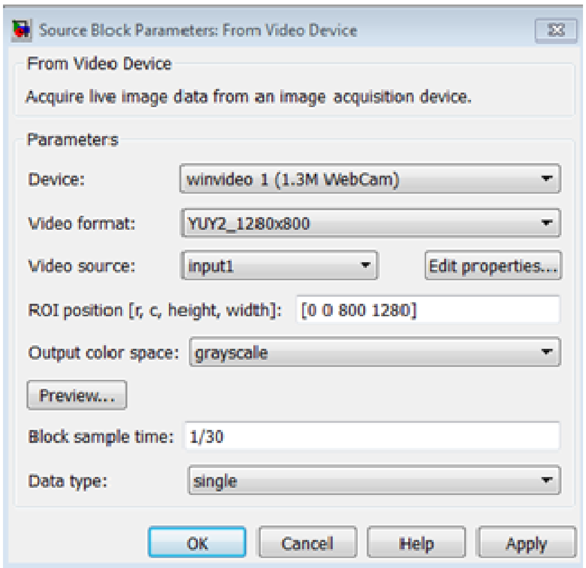


3.1 Image Acquisition Toolbox

Per acquisire l'immagine e identificare la posizione della pallina, é stato utilizzato il toolbox messo a disposizione dalla Mathworks che permette di elaborare un'immagine acquisita mediante una semplice webcam collegata al pc.
L'algoritmo di identificazione della posizione (ed in particolare del centro della pallina) viene riportato nell'*Appendice A*.
Di seguito si riporta lo schema Simulink:



In particolare, il blocco **From Video Device** consente di acquisire immagini e dati video in streaming da dispositivi come telecamere e frame-grabber e di poterli manipolare in ambiente Simulink.
La finestra di comando di tale blocco permette, inoltre, di impostare la dimensione in pixel dell'immagine (600×800), la ROI *region of interest* e la modalitá di output utilizzata per i colori dell'informazione acquisita.



Tale blocco restituisce in output una matrice contenente valori compresi tra 0 e 255 codificata in scala di grigi, avendo impostato la modalit *grayscale*. Dopodiché l'immagine entra in una Matlab Function che esegue le seguenti operazioni:

- Converte l'immagine in bianco e nero;
- Ricava la posizione della pallina attraverso un algoritmo basato sul fatto che l'immagine acquisita é una matrice di elementi pari a 0, eccetto che nella regione dove é posizionata la pallina in cui gli elementi sono pari a 1.

  A tale scopo:

  - Si é costruito un vettore di tanti elementi quante sono le righe della matrice e a ciascun elemento é stata associata la somma degli elementi contenuti nella riga della matrice corrispondente.
  - Attraverso una selezione a soglia, é stato individuato l'indice di riga con maggiore concentrazione di 1 (colore bianco) e tale indice é stato posto pari alla coordinata della pallina lungo l'asse x.
  - In maniera del tutto analoga, é stata trovata la posizione della pallina lungo l'asse y.

3.2 Controllo LQR

Nel secondo macroblocco viene computata la legge di controllo del sistema. Si osservi che il sistema originario non é intrinsecamente disaccoppiato, ma in seguito alla linearizzazione, i termini di accoppiamento scompaiono e di conseguenza é stato possibile risolvere il problema del controllore con le tecniche utilizzate per sistemi SISO, quali: l'assegnamento dei poli e il Linear Quadratic Regulator.
In realtá, per poter utilizzare la tecnica **LQR** é necessario avere a disposizione tutto lo stato del sistema e data la mancanza di un apposito sensore, la velocitá della pallina é stata calcolata come velocitá media tra due istanti di campionamento successivi:

$$v_{ball} = \dot{x}_b = \frac{x_b(k) - x_b(k-1)}{\Delta T} \tag{3.1}$$

In maniera del tutto analoga, é stata calcolata la velocitá lungo l'asse y.
Una volta ottenuta la velocitá della pallina, lo stato del sistema linearizzato é a completa disposizione, essendo la posizione fornita dal sistema di *Image Acquisition* di cui sopra.
A tal punto, l' applicazione della tecnica di controllo LQR risulta del tutto immediata.
Innanzitutto, é stata verificata la controllabiltiá e l'osservabilitá del sistema, calcolando il rango della matrice di controllabilitá e di osservabilitá:

$$rankC = \begin{bmatrix} B & AB & A^2B & \cdots & A^{n-1}B \end{bmatrix} \tag{3.2}$$

$$rankO = \begin{bmatrix} C & CA & CA^2 & \cdots & CA^{n-1} \end{bmatrix}^T \tag{3.3}$$

entrambi pari all'ordine del sistema linearizzato.

Una volta verificate tali proprietá, si é passato a discretizzare il sistema utilizzando un passo di campionamento pari a $= 1/22$, essendo questo il tempo minimo necessario per eseguire tutti i calcoli.
Come giá detto, per il controllo della pallina sul piano, é stata utilizzata la tecnica in retroazione di stato *Linear Quadraic Regualator* che, oltre a garantire la stabilitá del sistema retroazionato, fornisce la legge di controllo ottima $u(t)$ che porta il sistema dallo stato iniziale $x(t_0)$ allo stato finale $x(t_f)$ mantenendo lo stato del sistema vicino all'origine dello spazio di stato, minimizzando un indice di costo quadratico del tipo:

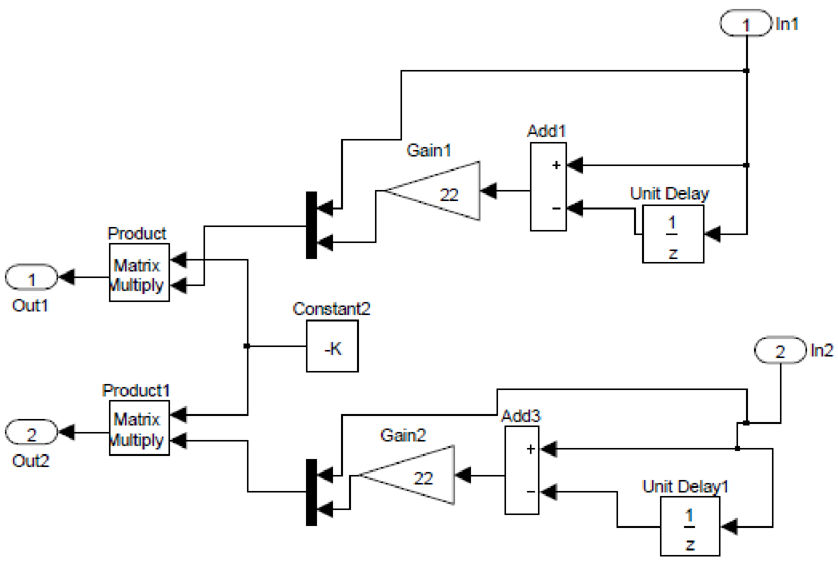
$$J = \int_{t_0}^{tf} x(t)^T Q x(t) + u(t)^T R u(t)\, dt \tag{3.4}$$

in cui **Q** é la matrice di peso dello stato, **R** é la matrice di peso della variabile di controllo.
Nel nostro caso, tenendo conto delle numerose approssimazioni legate al modello e delle innumerevoli incertezze legate alla struttura meccanica, sono state scelte matrici di peso piuttosto elevate cosí da garantire una migliore stabilitá:

$$Q = \begin{bmatrix} 100 & 0 & 0 & \\ 0 & 100 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 100 \end{bmatrix}$$

$$R = \begin{bmatrix} 50 & 0 \\ 0 & 50 \end{bmatrix}$$

Il significato fisico per cui sono state scelte matrici diagonali é legato semplicemente al fatto che lo scostamento dello stato del sistema dall'origine dello spazio di stato ha peso identico per ciascuna componente.



La matrice dei guadagni $K = R^{-1}B^TP$ é stata immediatamente ricavata utilizzando il comando **lqrd** che restituisce la matrice P definita positiva, soluzione dell'equazione di Riccati per sistemi discretizzati (vedi Appendice A):

$$PBR^{-}1B^TP = PA + A^TP + Q \tag{3.5}$$

Nel nostro caso:

$$K = \begin{bmatrix} -1.0980 & -1.2326 & 0.0000 & -0.0000 \\ 0.0000 & -0.0000 & -1.0980 & -1.2326 \end{bmatrix} \tag{3.6}$$

e dalla struttura di tale matrice si evince che il controllore agisce sul sistema come se fosse costituito da due sottosistemi disaccoppiati e ció favorisce una notevole semplificazione nell'implementazione della logica di controllo sul nostro microcontrollore. I risutltati ottenuti sono i seguenti:

Figura 3.1: Andamento della posizione lungo l'asse x

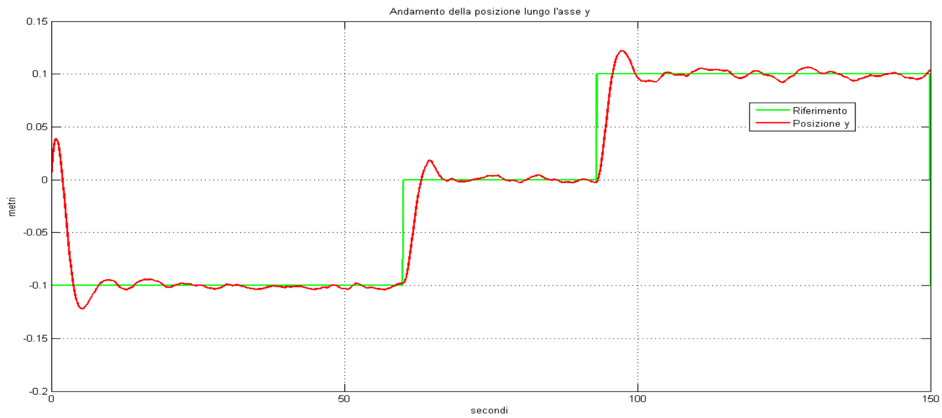


Figura 3.2: Andamento della posizione lungo l'asse y

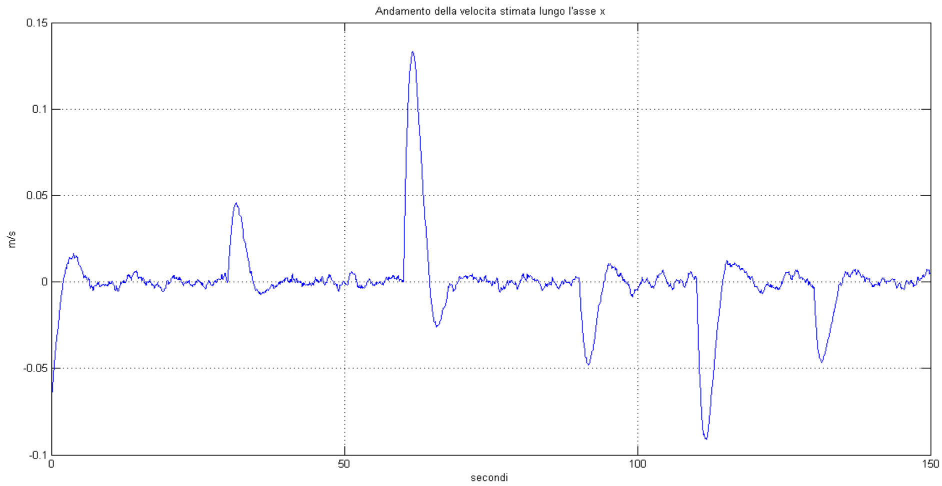


Figura 3.3: Andamento della velocitá stimata lungo l'asse x

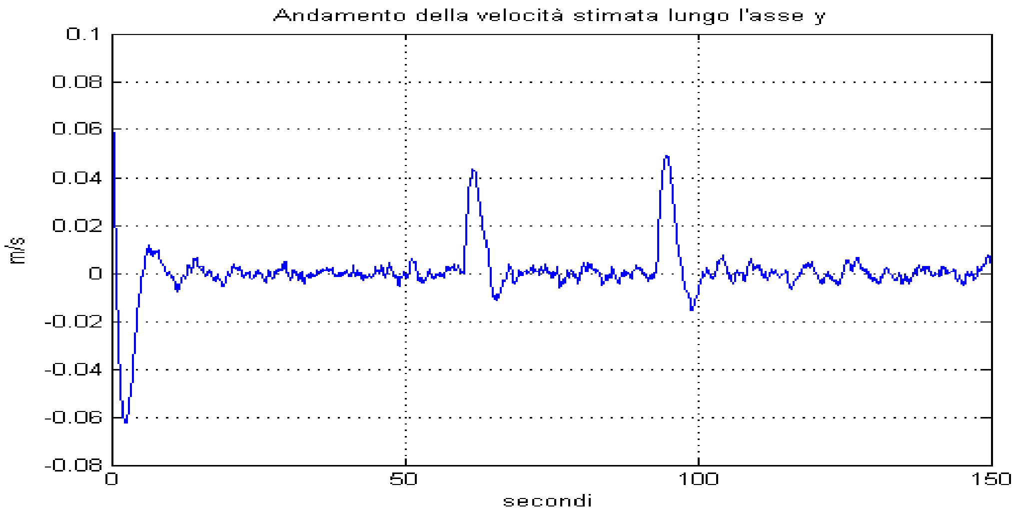
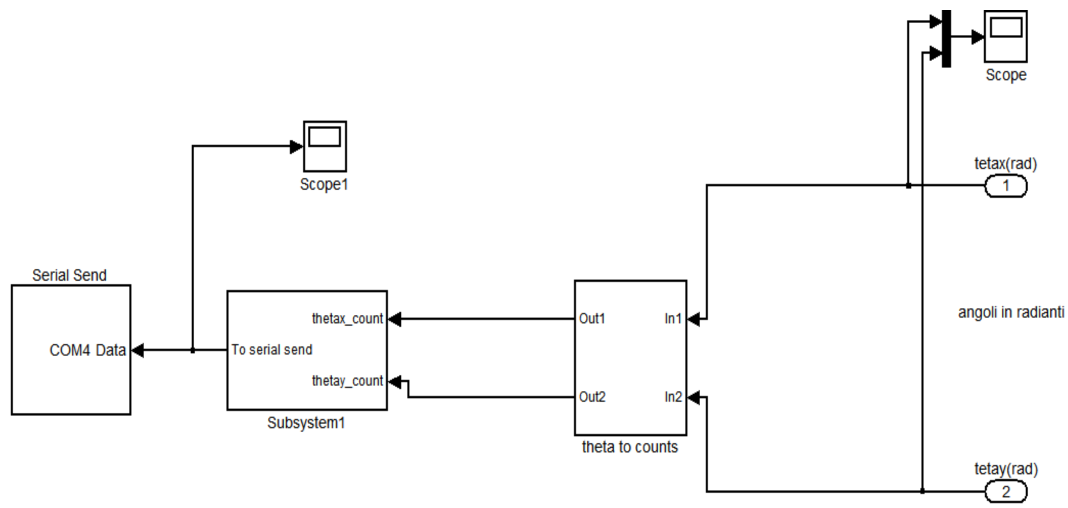


Figura 3.4: Andamento della velocitá stimata lungo l'asse y

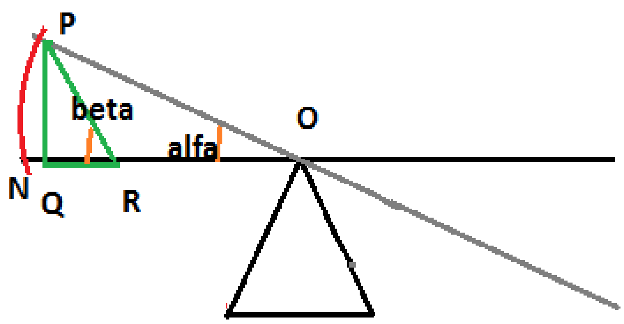
3.3 Comunicazione Via Seriale

Quest'ultimo macroblocco ha il compito di inviare ai motori un valore che ne codifica la posizione dell'albero chiudendo, cosí, il loop di controllo.

Data la struttura del piano risulta evidente che ad una rotazione dei servo-



comandi corrisponde una ben precisa rotazione del piano, o meglio del punto del piano vincolato all'albero in uscita dei servocomandi. Per questo motivo é stato necessario identificare con precisione il fattore di conversione tra una coordinata e laltra del sistema vincolato. Con riferimento alla seguente figura dove con α si indica l'inclinazione del piano e con β l'inclinazione del



servocomando, si é posto:

$$\begin{aligned}
\bar{ON} &= l && (3.7)\\
\bar{NR} &= m && (3.8)\\
\bar{PR} &= m && (3.9)\\
\bar{QR} &= m \cdot cos\beta && (3.10)\\
\bar{NQ} &= m\,(1-cos\beta) && (3.11)\\
\bar{QO} &= \bar{ON}-\bar{NQ} = l - m\,(1-cos\beta) && (3.12)\\
\bar{PQ} &= m \cdot sen\beta && (3.13)
\end{aligned}$$

Di conseguenza:

$$\alpha = f(\beta) = \arctan \frac{msin\beta}{l - m\,(1-cos\beta)} \tag{3.14}$$

Quello che a noi interessa, in realtá, é la relazione duale, ovvero $\beta = g(\alpha)$ che si ottiene costruendo la funzione α in ambiente Matlab mediante un polinomio interpolante del 5°grado:

```
beta=-pi/2:0.01:pi/2;
alpha=atan2(m*sin(beta),l-m*(1-cos(beta)));
plot(alpha,beta);
```

Una volta determinato l'angolo d'inclinazione del piano, é necessario convertire tale informazione in termini di durata del duty-cicle dell'onda quadra da inviare al microcontrollore per pilotare il servocomando e portare l'albero di uscita nella posizione desiderata.
Per trovare tale relazione, si é fatto riferimento al datasheet dei servocomandi dal quale é stato ricavato che la durata dell'impulso in millisecondi corrispondenti al fine corsa alto e al fine corsa basso sono rispettivamente di $0.6ms$ e $2ms$.
A tal punto, mettendo in relazione la rotazione completa dell'albero con l'intervallo di tempo impiegato:

$$(180° - 0°) : (2ms - 0.6ms) = \theta_d : \Delta t \tag{3.15}$$

é stata ricavata la relazione:

$$\Delta t = \frac{1.4ms \cdot \theta_d}{180°} \tag{3.16}$$

a cui deve essere sottratta la quantitá di offset di $0.6ms$ per ottenere il corretto duty-cicle.

In definitiva, la relazione che permette di convertire l'angolo desiderato nell'intervallo di ampiezza temporale da inviare al servocomando é:

$$\delta t = \frac{1.4ms \cdot \theta_d}{180^\circ} + 0.6 \tag{3.17}$$

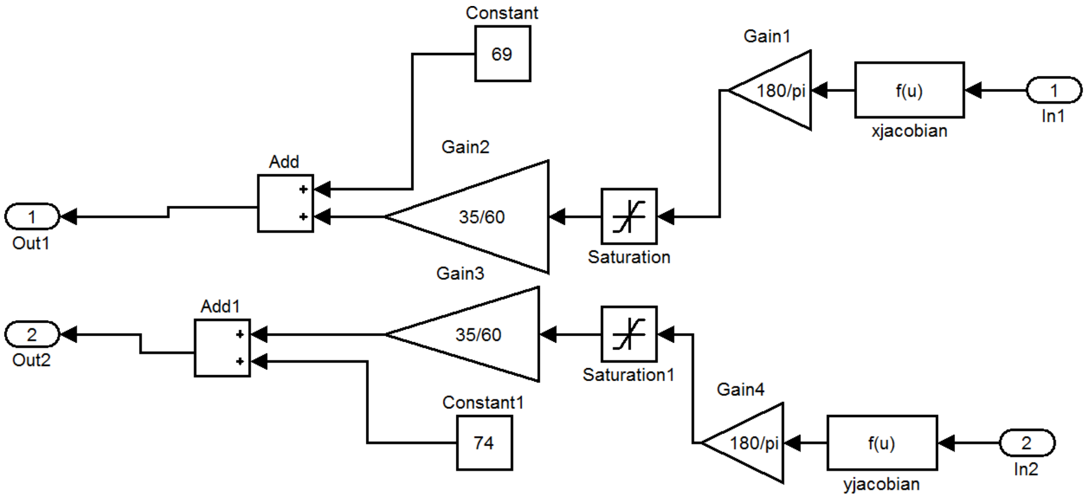


Figura 3.5: Schema Simulink

Capitolo 4

Osservatore di Luenberger

Data l'assenza di un opportuno sensore che potesse misurare la velocitá della pallina, lo stato del sistema, é stato in prima approssimazione ricavato derivando la posione mediante un'equazione alle differenze.
Successivamente, per poter avere un'informazione piú corretta é stato utilizzato l' *Osservatore di Luenberger*, sistema dinamico in grado di stimare lo stato del sistema sulla base delle sole informazioni disponibili: ingressi e uscite.
La struttura dell'osservatore di Luenberger é:

$$\dot{\xi} = A\xi + Bu + L(y - C\xi); \tag{4.1}$$

in cui il guadagno L é stato scelto in modo tale che gli autovalori della matrice $(A - LC)$ siano molto piú veloci degli autovalori della matrice $(A + BK)$ in modo tale che il sistema **Errore di Stima** $\dot{e} = (A - LC)e$ converga a 0 (e quindi $\xi \to x$) piú velocemente rispetto alle dinamiche del sistema.
Nel nostro caso, i poli dell'osservatore sono stati posti in [-30,-31,-34,-36]:

```
L=place(A',C_t',[-30 -31 -34 -36])'
```

Per la legge di controllo stavolta é stata utilizzata la tecnica di *Assegnamento degli Autovalori* attraverso il comando **place** che restituisce un guadagno K tale da assegnare i poli nelle posizioni desiderate:

```
K=place(A,B,[-4 -5 -3 -6]);
```

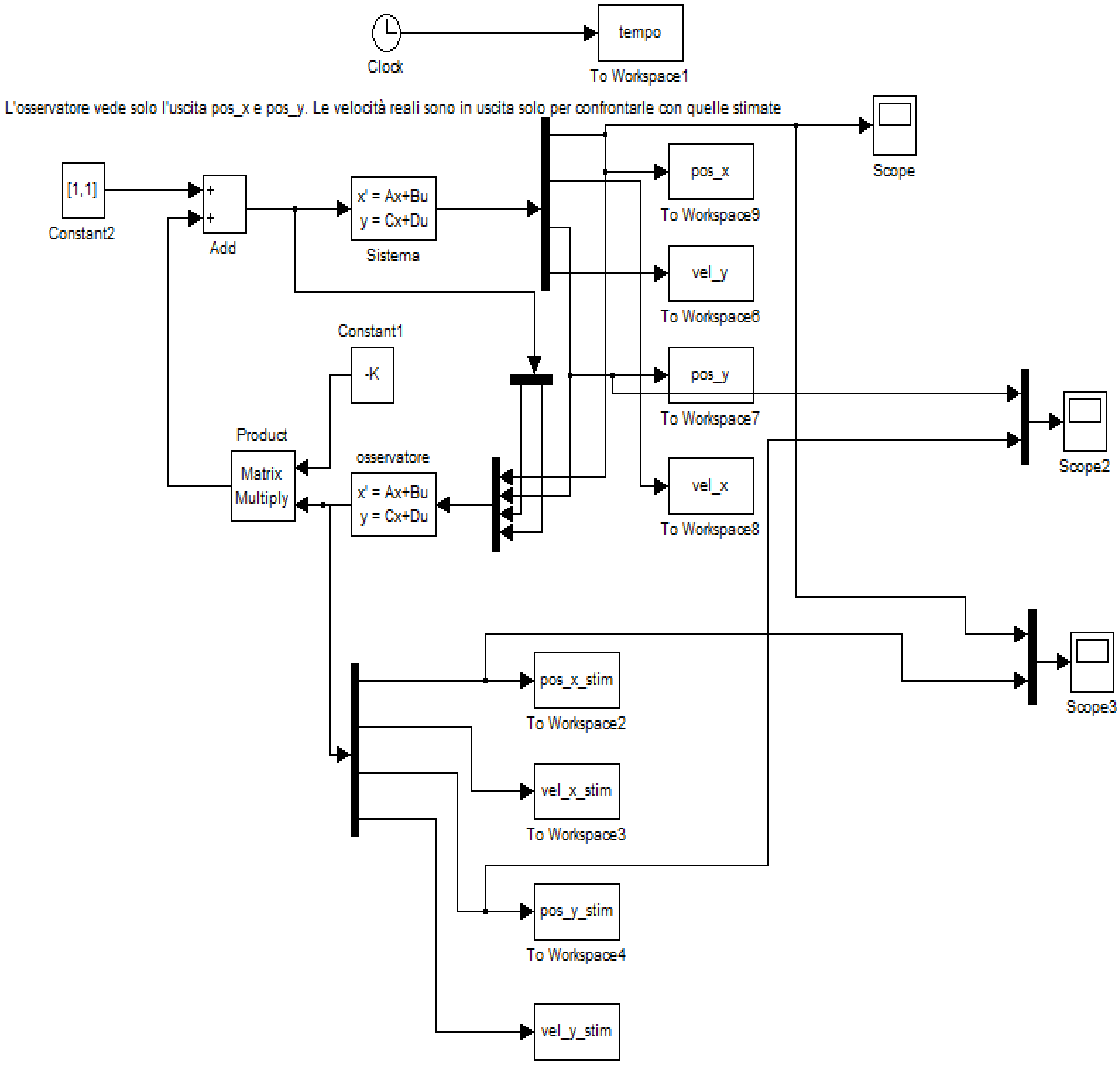
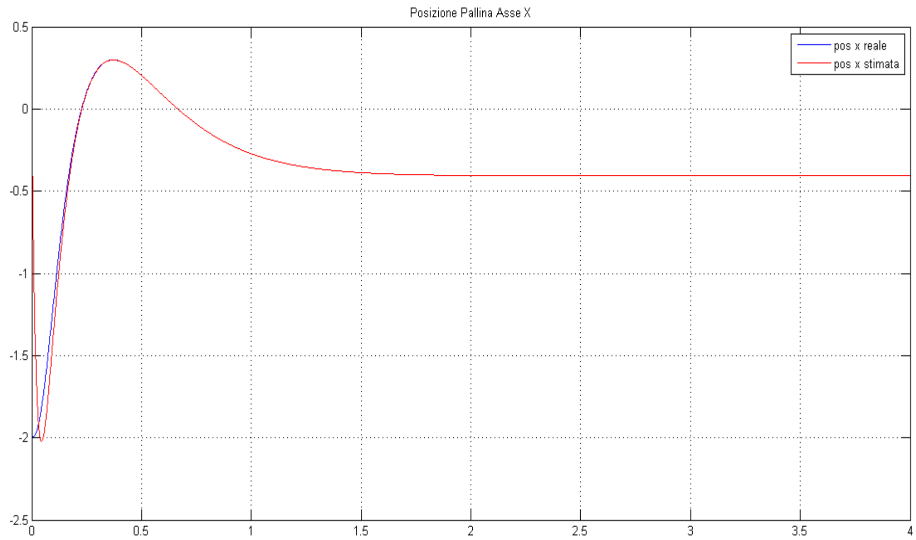
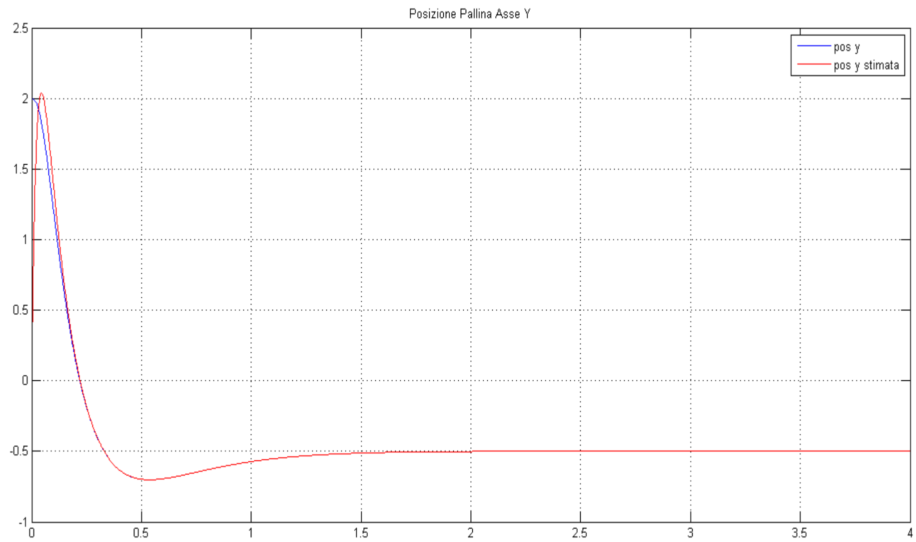


Figura 4.1: Schema Simulink - Osservatore

Gli andamenti temporali dello stato reale e di quello stimato sono riportati in figura:





Appendice A

Appendice

Acquisizione Immagine

```
function [pos_col,pos_rig]  = fcn(IM)

s=size(IM);
col=s(2);
rig=s(1);

not_found=true;
i=1;
pos_col=0;
while (not_found && i<col)
    if(any(IM(:,i)<50))
        %%320    la dimensione in mm del piano in direzione y
        pos_col=(i-col/2)/col*320;
        not_found=false;
    end
    i=i+1;
end

not_found=true;
i=1;
pos_rig=0;
while (not_found && i<rig)
    if(any(IM(i,:)<50))
        %%220    la dimensione in mm del piano in direzione x
        pos_rig=(i-rig/2)/rig*220;
        not_found=false;
    end
    i=i+1;
end

end
```

Controllo LQR

```
dT=1/20;    %passo di campionamento

A=[0 1 0 0; 0 0 0 0; 0 0 0 1; 0 0 0 0];
B=[0 0; -(5/7*9.8) 0;0 0;0 -(5/7*9.8)];
C=[1 0 0 0; 0 0 1 0];
O=obsv(A,C);    %matrice di osservabilita'
rank(O)
C=(ctrb(A,B));  %matrice di controllabilita'
rank(C)

Q=100*eye(4);
R=diag([50 50]);
K=lqrd(A,B,Q,R,dT);
Kx=[K(1,1) K(1,2)];
Ky=[K(2,3) K(2,4)];

x0=[0.1 0.01 0 0 0.2 0.02 0 0]';    %stato iniziale
```

Osservatore di Luenberger

```
A=[0 1 0 0; 0 0 0 0; 0 0 0 1; 0 0 0 0];
B= [0 0; -(5/7*9.8) 0; 0 0; 0 -(5/7*9.8)];
C_t=[1 0 0 0; 0 0 1 0]; %serve all'osservatore per fare i conti
C=eye(4);               %bisogna conoscere tutto lo stato
D=zeros(4,2);
x0=[-2 0.5 2 -0.3];     %stato iniziale

L=place(A',C_t',[-30 -31 -34 -36])';

Aoss=A-L*C_t;
Boss=[L B];
Coss=eye(4);
Doss=zeros(4,4);

%retroazione di stato con assegnamento degli autovalori
K=place(A,B,[-4 -5 -3 -6]);
```